МИНИСТЕРСТВО ОБРАЗОВАНИЯ МОСКОВСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ МОСКОВСКОЙ ОБЛАСТИ «БАЛАШИХИНСКИЙ ТЕХНИКУМ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ СТУДЕНТОВ ПО ДИСЦИПЛИНАМ И ПМ

основной образовательной программы
по профессии среднего
профессионального образования

09.01.03 «Мастер по обработке цифровой информации»

базовой подготовки

Квалификация: **Оператор электронно-вычислительных и вычислительных машин**

Форма обучения: **очная**

г. Балашиха
2020-2021 учебный год

методические рекомендации по выполнению

ПРАКТИЧЕСКИХ РАБОТ

Дисциплине ОП 01 Основы информационных технологий
по профессии
09.01.03. «Мастер по обработке цифровой информации

г. Балашиха 2020г

Практические работы являются неотъемлемой частью учебной дисциплины ОП.01 «Основы информационных технологий» и составляют по количеству часов не менее половины объема обязательной аудиторной нагрузки.

Современная учебная программа согласно требованиям Федерального государственного образовательного стандарта (ФГОС) предусматривает проведение цикла практических работ. На практических уроках проводятся работы, ориентированные на закрепление изученного теоретического материала, его более глубокому освоению и формированию умения применять теоретические знания в практических целях. Они выполняются в группах студентов совместно с преподавателем, имеют репродуктивный характер. Основными задачами практических работ являются: формирование умений подбирать материалы по их назначению, условиям эксплуатации, применять их при выполнении работ.

Содержание практической работы составляют:

- номер и тема практической работы;
- цель лабораторной работы;
- рекомендации для выполнения практической работы;
- перечень используемого оборудования и программного обеспечения;
- порядок выполнения работы;
- вывод о проделанной работе.

Перед тем как приступить к выполнению практической работы, студент должен пройти инструктаж по технике безопасности, усвоить краткие теоретические сведения по теме, методику выполнения работы, а также способы представления полученных данных.

Правила по технике безопасности для студентов при проведении практических работ

Общие правила:

1. Практические работы проводятся под наблюдением преподавателя. К выполнению практических работ студенты допускаются только после прослушивания инструктажа по технике безопасности и противопожарным мерам. После инструктажа каждый студент расписывается в журнале

2. Строго выполнять правила техники безопасности и санитарно-гигиенические нормы при работе в кабинете.

3. Все практические работы проводятся за компьютерными столами учебного кабинета. Студентам не разрешается без уважительной причины отлучаться из кабинета до полного окончания практических работ.

4. На рабочем месте должны находиться только необходимые для работы оборудование и материалы. Класть сумки необходимо на специально отведенный для этого стол.

5. Бережно обращаться оргтехникой. Входить в класс разрешается только после звонка на урок, спокойно, не торопясь, не задевая столы.

6. Занимать места в кабинете необходимо согласно «Схеме посадочных мест», начиная с первых парт.

7. Студент отвечает за состояние рабочего места и сохранность размещенного на нем оборудования.

8. Соблюдение всех вышеперечисленных рекомендаций по организации учебного процесса с использованием компьютеров и технических средств обучения должно способствовать сохранению оптимального уровня работоспособности и функционального состояния организма, на протяжении всех учебных занятий в колледже и полной безопасности для их жизни и здоровья.

Перед началом работы:

1. Проверить порядок на рабочем месте;
2. Отрегулировать положение монитора так, чтобы расстояние от глаз до экрана составляло не менее 50 см.

Во время работы:

1. Во время работы монитор является источником электромагнитного излучения, которое неблагоприятно действует на зрение. Поэтому надо работать на расстоянии 60-70 см, соблюдая правильную осанку (вертикально прямая спина, плечи опущены и расслаблены, ноги на полу не скрещены, стоят на подставке для ног, локти, запястья и кисти рук на одном уровне).

2. Непрерывное занятие студента за компьютером не должно превышать 30 минут. По истечении данного времени необходим перерыв длительностью 5 минут для снятия напряжения глаз. Для снятия усталости мышц используйте комплекс упражнений по профилактике зрительного утомления, упражнения для рук и плечевого пояса, для туловища и ног.

3. При плохом самочувствии, появлении головной боли, головокружении и др. прекратить работу и сообщить об этом преподавателю.

4. Обо всех неисправностях немедленно сообщать преподавателю;

5. В случае аварийной ситуации выключить компьютер.

По окончании работы:

1. Собрать методические указания к практическим работам и сдать их преподавателю;
2. Выключить ЭВМ после разрешения преподавателя;
3. Навести порядок на рабочем месте.

При работе в компьютерном классе строго запрещается:

1. Находиться в верхней одежде и грязной обуви;
2. Принимать пищу на рабочем месте и в компьютерном кабинете.
3. Удалять и перемещать чужие файлы;
4. Приносить и запускать свое программное обеспечение (программы);
5. Работать на ЭВМ грязными или мокрыми руками;
6. Прикасаться пальцами к мониторам, стучать по ним;
7 Включать и выключать компьютер без разрешения преподавателя;
8. Класть диски, книги, тетради на составляющие компьютера;
9. Подключать к компьютеру свои устройства (сот. телефоны, плееры).
10. Работать на не исправном компьютере;
11. Оставлять вычислительную технику на длительное время без присмотра;
12. Прикасаться к электрическим вилкам, розеткам, проводам, разъемам, задним стенкам системного блока и монитора;
13. Вскрывать корпуса, вынимать и вставлять разъемы, платы;

Правила выполнения практических работ

При домашней подготовке к выполнению практических работ студент должен повторить изученную тему.

Практическая работа выполняется каждым студентом самостоятельно.

Студенты, пропустившие занятия, выполняют практические работы во внеурочное время.

После выполнения каждой практической работы студент демонстрирует результат выполнения преподавателю, отвечает на вопросы. Преподаватель на уроке оценивает работу и выставляет оценку в журнал.

Критерии оценки практических работ

Практическая работа считается выполненной, если студент набрал проходной балл, который составляет половину максимального количества баллов.

Для оценивания работы прилагается эталон и шкала оценок.

Оценка «5» – работа выполнена в полном объеме и без замечаний.

Оценка «4» – работа выполнена правильно с учетом 2-3 несущественных ошибок, исправленных самостоятельно по требованию преподавателя.

Оценка «3» – работа выполнена правильно не менее чем на половину или допущена существенная ошибка.

Оценка «2» – допущены две (и более) существенные ошибки в ходе работы, которые студент не может исправить даже по требованию преподавателя или работа не выполнена.
Во всех случаях оценка снижается, если студент не соблюдает требования безопасности труда.

Перечень практических работ

Практическая работа №1 Технологии сбора, хранения, передачи, обработки информации

Практическая работа № 2Подключение периферийных устройств ПК: интерфейсы, кабели и разъемы. Организация данных в ПК

Практическая работа № 3Настройка операционной системы

Практическая работа №4Работа в текстовом редакторе создание таблиц

Практическая работа №5Работа в текстовом редакторе форматирование документов

Практическая работа №6Работа в текстовом редакторе создание авто оглавления

Практическая работа №7Создание документа в текстовом редакторе по заданным условиям

Практическая работа №8Работа в табличном редакторе автозаполнение

Практическая работа №9Работа в табличном редакторе создание диаграмм

Практическая работа №10Создание документа в табличном редакторе по заданным условиям

Практическая работа №11Работа в табличном редакторе работа с вычислениями

Практическая работа №12Работа в редакторе Pover Point создание презентации

Практическая работа №13Работа в редакторе Pover Point создание анимации

Практическая работа №14Работа в редакторе Pover Point создание дизайна

Практическая работа №15Работа в редакторе баз данных создание БД учащихся группы

Практическая работа №16Создание документа в редакторе баз данных по заданным условиям

Практическая работа №17Поиск, пересылка информации в локальной сети

Практическая работа №18Создание электронной почты

Практическое работа №1 (1час)

Тема: Способы хранения, обработки и передачи информации.

Цель: научиться сохранять, обрабатывать и передавать данные при помощи технических средств информации.

Оборудование: ПК, сканер, фотоаппарат, микрофон, диск.

Теоретические основы

1. **Технологии сбора и хранения информации**

Сбор предполагает получение максимально выверенной исходной информации и является одним из самых ответственных этапов в работе с информацией, поскольку от цели сбора и методов последующей обработки полностью зависит конечный результат работы всей информационной системы.

Технология сбора подразумевает использование определенных методов сбора информации и технических средств, выбираемых в зависимости от вида информации и применяемых методов ее сбора. На заключительном этапе сбора, когда информация преобразуется в данные, т. е. в информацию, представленную в формализованном виде, пригодном для компьютерной обработки, осуществляется ее ввод в систему. Для сбора данных необходимо сначала определить технические средства, позволяющие осуществлять сбор быстро и высококачественно и поддерживающие операции ввода информации и представления данных в электронной форме.

2. **Технологический процесс обработки информации**

Технологический процесс обработки информации — есть строго определенная последовательность взаимосвязанных процедур, выполняемых для преобразования первичной информации с момента ее возникновения до получения требуемого результата.

Технологический процесс призван автоматизировать обработку исходной информации за счет привлечения технических средств базовой информационной технологии, сократить финансовые и трудовые затраты, обеспечить высокую степень достоверности результатной информации. Для конкретной задачи той или иной предметной области технологический процесс обработки информации разрабатывается индивидуально. Совокупность процедур зависит от следующих факторов:

- характер и сложность решаемой задачи;
- алгоритм преобразования информации;
- используемые технические средства;
- сроки обработки данных;
- используемые системы контроля;
- число пользователей и т. д.

В общем случае технологический процесс обработки информации включает процедуры.

3. **Способы обработки информации**

Современные информационные технологии позволяют обрабатывать информацию централизованным и децентрализованным (т. е. распределенным) способами.

- Централизованный способ предполагает сосредоточение данных в информационно-вычислительном центре, выполняющем все основные действия технологического процесса обработки информации. Основное достоинство централизованного способа — сравнительная дешевизна обработки больших объемов информации за счет повышения загрузки вычислительных средств.
- Децентрализованный способ характеризуется рассредоточением информационно-вычислительных ресурсов и распределением технологического процесса обработки информации по местам возникновения и потребления информации. Достоинством децентрализованного способа является повышение оперативности обработки информации и решения поставленных задач за счет автоматизации деятельности на конкретных рабочих местах, применения надежных средств передачи информации, организации сбора первичных документов и ввода исходных данных в местах их возникновения.

- Децентрализованный способ обработки информации может быть реализован автономным или сетевым методом. При автономной обработке информации передача документов и данных на электронных носителях осуществляется по почте либо курьером, а при сетевой — через современные каналы связи.

На практике применяют смешанный способ обработки информации, для которого характерны признаки двух способов одновременно (централизованный с частичной децентрализацией или децентрализованный с частичной централизацией).

4. **Режимы обработки информации на компьютере**

Вычислительные средства участвуют в процессе обработки информации в двух основных режимах: пакетном или диалоговом.

В случае, когда технология обработки информации на компьютере представляет собой заранее определенную последовательность операций, не требующую вмешательства человека, и диалог с пользователем отсутствует, информация обрабатывается в так называемом пакетном режиме. Суть его состоит в том, что программы обработки данных последовательно выполняются под управлением операционной системы как совокупность (пакет) заданий. Операционная система обеспечивает ввод данных, вызов требуемых программ, включение необходимых внешних устройств, координацию и управление технологическим процессом обработки информации.

Задачи, решаемые в пакетном режиме, характеризуются следующими свойствами:

- алгоритм решения задачи формализован, вмешательства пользователя не требуется;
- наличие большого объема входных и выходных данных, в основном хранящихся на устройствах хранения информации (например, жестких дисках компьютеров);
- расчет выполняется для большинства записей входных файлов;
- длительное время решения задачи — как правило, обусловлено большими объемами обрабатываемых данных;
- регламентность — задачи решаются с заданной периодичностью.

Пакетный режим возник первым и широко использовался с середины XX в., когда обработка информации на ЭВМ осуществлялась в специально создаваемых вычислительных центрах. Заказчики подготавливали исходные данные (обычно на перфокартах или перфолентах) и отправляли их в вычислительный центр, где данные обрабатывались и результаты обработки возвращались заказчику. С развитием персональных ЭВМ (начиная с 80-х гг. прошлого века) обработка данных стала осуществляться, в основном, непосредственно потребителями, поэтому в настоящее время пакетный режим используется достаточно редко. Сегодня более распространен диалоговый режим, когда необходимо непосредственное взаимодействие пользователя с компьютером и на каждое свое действие пользователь получает немедленные ответные действия компьютера. Диалоговый режим позволяет пользователю интерактивно управлять порядком обработки информации и получать результатные данные в виде необходимых документов либо файлов.

5. **Технологии передачи и представления информации**

Информационные процессы невозможны без средств передачи и представления информации, поскольку зачастую информация требуется в месте, территориально удаленном от источника ее

возникновения, и должна быть представлена в виде символов, образов и сигналов, пригодных для восприятия потребителем.

Современные средства связи способны передавать информацию в любой форме: телефонные, телевизионные, телеграфные сообщения, массивы данных, печатные материалы, фотографии и т. д. В соответствии со спецификой передаваемых сообщений организуется канал передачи информации — совокупность технических средств, обеспечивающих передачу сигналов от источника к потребителю.

Основная характеристика канала передачи — скорость передачи информации, а ее предельно допустимое значение называют емкостью канала, которая ограничивается шириной полосы канала и шумом.

Канал связи соединяет передатчик и приемник с помощью линии связи, которая может быть проводной, кабельной, радио, микроволновой, оптической или спутниковой. Примерами линий связи являются телефонные и вычислительные сети, сети телевизионного и радиовещания, мобильной связи, спутниковые технологии передачи данных.

В современных цифровых системах связи функции передатчика и приемника выполняет модем. Основное достоинство передачи информации в цифровой форме заключается в возможности использования кодированных сигналов, обеспечения защиты информации и наилучшего способа приема.

Для представления переданной или хранимой информации потребителю используются процессы воспроизведения и отображения.

Воспроизведение информации — это процесс, при котором ранее записанная на носителе информация считывается устройством воспроизведения.

Отображение информации — есть представление информации, т. е. генерация сигналов на основе исходных данных, а также правил и алгоритмов их преобразования в форме, приемлемой для непосредственного восприятия человеком.

Потребителем информации наиболее часто выступает человек, и для принятия решений ему необходимы результаты обработки информации. Тем не менее человек не способен ощутить машинное представление информации, а может воспринимать ее лишь органами чувств (органами зрения, слуха, осязания, обоняния и т. д.), поэтому для организации взаимодействия человека с информационными моделями объектов информационная система должна быть наделена специальными средствами отображения данных.

Поскольку зрение используется для восприятия информации наиболее активно, то средства отображения в современных ИС должны представлять информацию в лучшей форме для визуального наблюдения. Заметим, что мультимедиа-системы позволяют также представлять информацию в форме аудио- и видеосигналов, однако для управленческих информационных систем наиболее характерно отображение информации в текстовой и графической форме, что осуществляется за счет использования мониторов и печатающих устройств (например, принтеров, плоттеров).

Прежде чем, например, на мониторе, появится информация в доступной для человека форме, компьютером будет автоматически осуществлена следующая последовательность операций:

- преобразование данных, представленных в машинной форме, в вид, приемлемый для экранного отображения;
- согласование формы представления данных с параметрами монитора;
- воспроизведение в соответствии с возможностями воспроизводящего устройства (т. е. в данном примере — монитора).

Порядок выполнения работы

1. Создание досье группы. Заранее заготовить материал: фотографии, текст.
2. Сфотографировать своих однокурсников.
3. Включить компьютер.
4. Создать общую папку на сервере.
5. Сканировать фотографии и сохранить в общую папку.
6. Включить текстовый редактор. Создать титульный лист с общей фотографией и названием группы: специальность и год.
7. Оформить каждый лист на одного человека. Записать данные: дата рождения, номер школы (выпускниками которой являются), хобби.
8. Сохранить данные на жесткий диск в свою папку под именем досье группы.

Отчет

Отчет должен быть оформлен в текстовом редакторе и содержать:

- наименование работы;
- цель работы;
- задание;
- последовательность выполнения работы;
- ответы на контрольные вопросы;
- вывод о проделанной работе.

Контрольные вопросы

1. Что такое сбор информации и каково его предназначение?
2. Что понимается под технологией сбора информации?
3. Чем отличаются понятия «информация» и «данные»?
4. Назовите основные требования к сбору данных и к хранимым данным.
5. Перечислите основные средства сбора текстовой, графической, звуковой и видеоинформации. Какие еще средства сбора информации вам известны?
6. Какие еще методы сбора данных вам известны?
7. В чем заключается процедура хранения информации?
8. Перечислите основные требования к структурам хранения.
9. Что такое база данных?
10. В чем различие между базой и банком данных?
11. Что такое резервное копирование и для чего оно осуществляется?
12. Что такое архивное копирование и в чем его отличие от резервного копирования?
13. Что такое базовая информационная технология?

14. В чем заключается различие между централизованным и децентрализованным способами обработки информации?
15. Какие режимы обработки информации вам известны?

Практическая работа № 2

Периферийные устройства PC. Настройка и установка периферийных устройств.

Цель работы:

Изучение основных компонентов персонального компьютера и основных видов периферийного оборудования, способов их подключения, основных характеристик (название, тип разъема, скорость передачи данных, дополнительные свойства). Определение по внешнему виду типов разъемов и подключаемого к ним оборудования

получение сведений по настройке пользовательского интерфейса периферийных устройств средствами операционной системы **Microsoft Windows**

Краткие теоретические сведения

Периферийные устройства *(ПУ) — аппаратура, предназначенная для внешней обработки информации. Другими словами, это устройства, расположенные вне системного блока –* ***внешние устройства.***

1 Подключения устройств к системному блоку

Все периферийные устройства подключаются только к системному блоку. Для работы конкретного устройства в составе конкретного комплекта ПЭВМ необходимо иметь:

1. Контроллер (адаптер) – специальную плату, управляющую работой конкретного периферийного устройства. Например, контроллер клавиатуры, мыши, адаптер монитора, портов и т.п.
2. Драйвер – специальное программное обеспечение, управляющее работой конкретного периферийного устройства. Например, драйвер клавиатуры, драйвер принтера и т.п.

Для управления работой устройств в компьютерах используются электронные схемы – *контроллеры*. Различные устройства используют разные способы подключения к контроллерам:

- ✓ некоторые устройства (дисковод для дискет, клавиатура и т. д.) подключаются к имеющимся в составе компьютера стандартным контроллерам (интегрированным или встроенным в материнскую плату);
- ✓ некоторые устройства (звуковые карты, многие факс-модемы и т. д.) выполнены как электронные платы, т. е. смонтированы на одной плате со своим контроллером;
- ✓ некоторые устройства используют следующий способ подключения: в системный блок компьютера вставляется электронная плата (контроллер), управляющая работой устройства, а само устройство подсоединяется к этой плате кабелем;
- ✓ на сегодняшний день большинство внешних устройств подключаются к компьютеру через USB-порт.

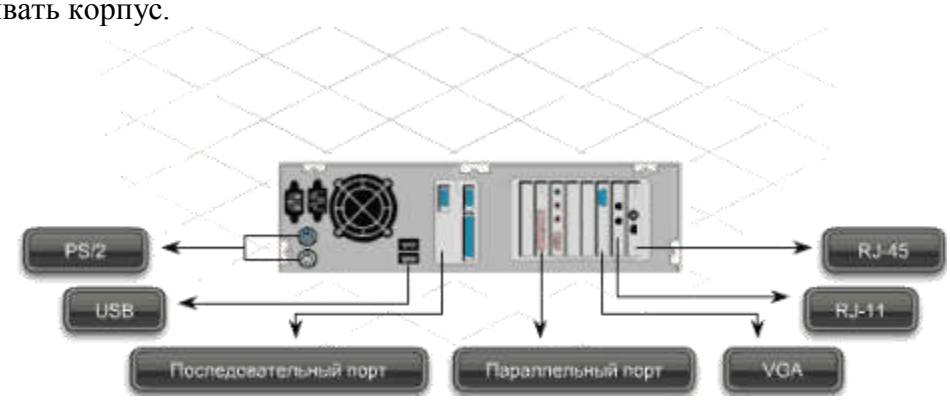
Платы контроллеров вставляются в специальные разъемы (слоты) на материнской плате компьютера.

С помощью добавления и замены плат контроллеров пользователь может модифицировать компьютер, расширяя его возможности и настраивая его по своим потребностям. Например, пользователь может добавить в компьютер факс-модем, звуковую карту, плату приема телепередач и т. д.

Одним из видов контроллеров, которые присутствуют почти в каждом компьютере, является контроллер портов ввода-вывода. Часто этот контроллер интегрирован в состав материнской платы. Контроллер портов ввода-вывода соединяется кабелями с разъемами на задней стенке компьютера, через которые к компьютеру подключаются принтер, мышь и некоторые другие устройства.

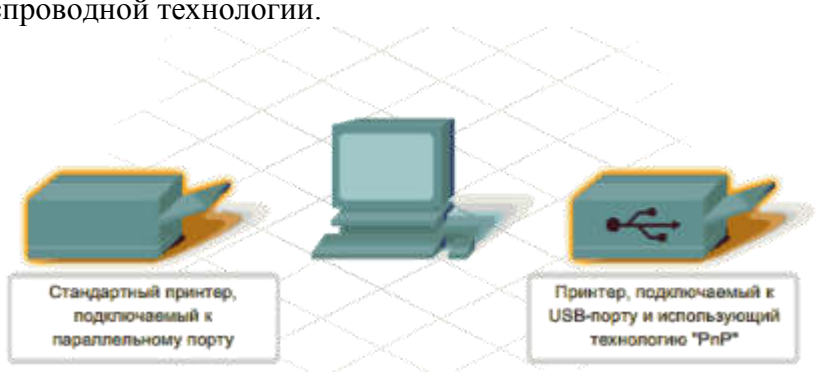
Кроме контроллеров портов ввода-вывода в системном блоке присутствуют разъемы шины USB – универсальной последовательной шины, к которой можно подключить клавиатуру, мышь, принтер, модем, дисковод компакт-дисков, сканер и т. д. Основное требование возможность подключения к данной шине устройства. Особенность шины USB – возможность подключения к ней устройств во время работы компьютера (не выключая его).

В отличие от внутренних компонентов, для установки периферийных устройств не нужно открывать корпус.



Периферийные устройства подключаются к разъемам на внешней части корпуса с помощью проводов или беспроводной связи. Исторически периферийные устройства разработаны в расчете на определенный тип портов. Например, в конструкции принтеров для персональных компьютеров предусмотрено подключение к параллельному порту, который передает от компьютера к принтеру данные в определенном формате.

Разработанный интерфейс универсальной последовательной шины (USB) сильно упростил использование проводных периферийных устройств. USB-устройства не требуют сложных процедур конфигурации. Они просто подключаются к соответствующему порту (при наличии нужного драйвера). Кроме того, все чаще появляются устройства, которые подключаются кузлу с использованием беспроводной технологии.



Установка периферийного устройства выполняется в несколько этапов. Порядок и тип этих шагов зависит от типа физического подключения и от того, относится ли устройство к типу автоматически настраивающихся (PnP). Предусмотрены следующие шаги:

подсоединение периферийного устройства к узлу с помощью соответствующего кабеля или беспроводного соединения;

подключение устройства к источнику питания;

установка соответствующего драйвера.

Некоторые устаревшие устройства, так называемые "обычные устройства", не предусматривают самонастройки. Драйверы таких устройств устанавливаются после того, как устройство подключается к компьютеру и включается питание.

Драйверы самонастраивающихся USB-устройств в системе уже имеются. В таком случае при подключении и включении операционная система распознает устройство и устанавливает соответствующий драйвер.

2 Организация электропитания устройств компьютера

Все устройства, имеющиеся внутри системного блока (как внутренние, так и внешние) питаются через блок питания системного блока.

Монитор поддерживает две схемы подключения:

- через системный блок (в этом случае при включении/выключении системного блока одновременно включается/выключается монитор);

- через отдельную розетку (тогда при включении системного блока монитор нужно включить через отдельную кнопку).

Все остальные периферийные устройства, расположенные в отдельны корпусах, имеют отдельные кабели питания, подключаемые к собственным розеткам.

Несмотря на высокую надежность и безопасность, компьютерное оборудование должно быть заземлено.

Во избежание нежелательных последствий скачков напряжения электросети (потеря и порча информации, выход из строя компьютерного оборудования и пр.), ПК обычно подключают к электросети через источник бесперебойного питания — UPS, который стабилизирует подаваемое на аппаратуру напряжение от сети и, при его полном отключении, поддерживает питание компьютера в течение некоторого (обычно 15—20 минут) времени, чтобы пользователь успел завершить работу запущенных программ, сохранить необходимую информацию и выключить компьютер по стандартной схеме.

Если в составе комплекта компьютерного оборудования нет UPS, то желательно использовать хотя бы обычный стабилизатор напряжения или специальный сетевой фильтр.

2 Разъемы для подключения электропитания и внешних устройств

Расположение разъемов. Обычно разъемы для подключения электропитания и внешних устройств находятся на задней стенке системного блока компьютера. В портативных компьютерах эти разъемы могут находиться и с боковых сторон системного блока.

Подключение электропитания и внешних устройств к компьютеру выполняется с помощью специальных проводов (кабелей). Для защиты от ошибок разъемы для вставки этих кабелей сделаны разными, так что кабель, как правило, просто не вставится в неподходящее гнездо.

Разъемы для электропитания. Обычно на задней стенке системного блока компьютера имеется разъем для подсоединения к электросети и разъем для подачи электропитания на монитор.

Разъемы для подключения внешних устройств. Разъемы для подключения внешних устройств достаточно унифицированы. Распространенны разъемы со штырьками и с гнездами.

Порядок вставки кабелей. Вставлять и вынимать кабели из разъемов можно только при выключенном компьютере, в противном случае компьютер и устройства могут быть испорчены.

Вставлять кабели надо аккуратно, чтобы не погнуть и не поломать содержащиеся в разъемах штырьки.

Некоторые кабели (например, для монитора или принтера) закрепляются с помощью винтов, эти винты надо завернуть рукой или отверткой (только не слишком туго), чтобы кабель не выпадал из разъема во время работы компьютера.

Периферийные устройства.

К внешним или периферийным устройствам персонального компьютера **относится**

1. устройствами, которые предназначены **для ввода информации** в компьютер для обработки, и
2. устройствами, **для вывода информации** из него.
3. Есть еще и устройства **для хранения информации** вне системного блока (внешние накопители).

Клавиатура и мышь

Это основные устройства для ввода информации, необходимость в них очевидна и не обсуждается, пока сенсорные мониторы не вытеснили обычные :)

Рисунок 1. Клавиатура и мышь

Подключаться клавиатура и мышь к компьютеру может с помощью двух видов разъемов:

- ***PS/2***
- ***USB.***

Рисунок 2. Разъемы на задней панели системного блока

Клавиатуру и мышь с разъемом **PS/2** можно **подключать или отключать только при выключенном компьютере**. Если они выполнены с **USB**-разъемами, то их можно **подключать или отключать** в любой момент, **даже если компьютер работает**.

Акустические системы (колонки)

Акустические системы **преобразуют электрический сигнал**, полученный со звуковой карты компьютера в звуковые колебания (т.е. **в звук**) и относятся к устройствам вывода информации. На сегодняшний день акустические системы можно назвать обязательными устройствами ПК. Чтобы слушать музыку, смотреть фильмы, слушать аудиокниги, а также изучать IT-уроки, без «колонок» не обойтись.

Рисунок 3. Акустические системы (колонки)

*Акустические системы подключаются с помощью **разъема «мини-джек» зеленого цвета*** (на фотографии один из разъемов, обозначенных **цифрой 3**). Подключать и отключать можно при работающем компьютере.

Если динамики встроенные в ноутбук не устраивают по качеству звучания, то Вы также можете использовать внешние акустические системы, только подключаются они в разъем для наушников (о нем смотрите ниже.)

Наушники

Наушники, можно считать разновидностью **акустических систем**, только предназначены они **для одного пользователя** ПК. Фактически, наушники – это миниатюрные акустические системы для индивидуального использования.

Наушники

Подключаются наушники к компьютеру так же, как и акустические системы.

На ноутбуках разъем не выделен зеленым цветом, но возле него присутствует значок «наушники».

Микрофон

Микрофон может понадобиться в двух случаях:

- **Общение через Интернет** (например, через Skype);
- **Запись звука** с целью дальнейшего хранения и обработки на ПК.

Как можно догадаться, микрофон является устройством ввода информации.

Микрофон

*Микрофон подключается к ПК с помощью **разъема «мини-джек» розового цвета**.*

Веб-камера (Web-камера)

Веб-камера может понадобиться, если Вы планируете часто общаться через Интернет, и при этом хотелось бы **передавать** не только голос, но и своё **изображение** (следует учитывать, что скорость Интернета должна быть достаточной для передачи видеосигнала).

Веб-камера на ноутбуке (встроенная)

Видео-звонки очень увлекательны, но не обязательны при живом общении. Лично я считаю это периферийное устройство одним из последних в очереди на покупку, но некоторые со мной не согласятся :).

Веб-камера может иметь встроенный микрофон, что освобождает Вас от покупки «лишнего» устройства.

***Подключается** к компьютеру камера через **USB-разъем** (цифра 2 на рисунке 2).*

USB-флэш-накопитель (флэшка)

Флэш-накопители относятся к устройствам хранения данных, т.е. предназначены для **хранения и переноса информации** между компьютерами.

USB-флэш-накопитель (флэшка)

Кстати, рекомендую всем пользователям компьютера использовать флэшку не только по прямому назначению, но и как одно из средств **резервного копирования данных** (об этом мы поговорим в одном из ближайших IT-уроков).

Конечно же, приобрести флэшку желательно уже на первых шагах освоения компьютера.

*Из названия можно догадаться, что **подключается** USB-флэш-накопитель **с помощью USB-разъема**.*

Не забывайте про правило, что извлекать флэшку из разъема можно после программного отключения (значок возле системных часов).

Кардридер (он же картридер)

Если у Вас есть **фотоаппарат или видеокамера** (или **смартфон**), то кардридер значительно упростит работу с фотографиями и видеофайлами при переносе на компьютер.

Внешний кардридер

Не обязательно искать «родной» провод для подключения устройства к USB, достаточно **извлечь карту памяти (флэш-карту)** и вставить в нужный разъем на **кардридере**. Кроме того, некоторые устройства (например мобильные телефоны) не позволяют свободно (напрямую) работать с

файлами на карте памяти, но с помощью кардридера вы вольны **работать с данными как с обычной флэшкой**.

Кардридер может быть:

- **внешний** (указан на фотографии выше);
- **внутренний** (устанавливается в системный блок) или
- **встроенный** (обязателен в современных ноутбуках).

Внешний кардридер *подключается к системному блоку* ***через USB-разъем****.*

Обычно нельзя подключить к компьютерной системе новое устройство, не установив или не настроив соответствующее программное обеспечение. К счастью, операционная система Windows XP обычно делает это автоматически.

Несмотря на автоматическое распознавание, пожалуйста, внимательно изучите инструкции по установке, которые поставляются с каждым устройством. Иногда перед установкой нужно устанавливать специальное программное обеспечение. Как об этом узнать? Прочитать инструкцию!

Способность Windows автоматически распознавать новые устройства называется **plug-and-play** (подключай и играй).

Программа, управляющая работой устройства, называется **драйвером**.

Порядок выполнения работы

Задание 1: Осуществить подключение периферийного оборудования, используя различные интерфейсы: жесткий диск, привод на компакт-дисках, флоппи-дисковод, плату видеоадаптера, сетевую карту, звуковую карту, монитор, принтер, сканер.

1. Изучение разъемов для подключения электропитания и внешних устройств

Посмотрите на оборотную сторону системного блока с подключенными кабелями. Поочередно вынимая кабель конкретного устройства зарисуйте разъемы системного блока. Результаты изучения разъемов занесите в таблицу.

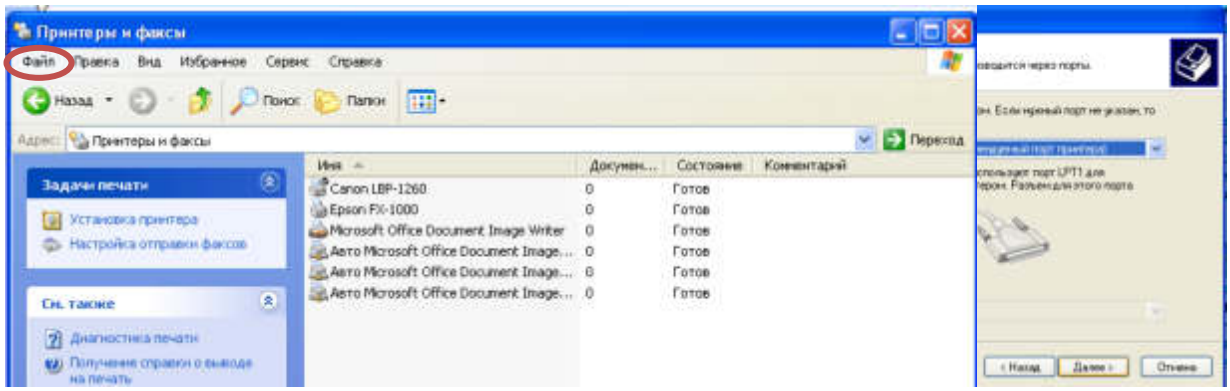
Устройство	Схема разъема, к которому подключено
1. Электропитание	
2. Клавиатура	
3. Мышь	
4. Питание монитора	
5. Монитор	
6. Сетевой кабель	
7. USB порт	

1. Подключение и настройка принтера.

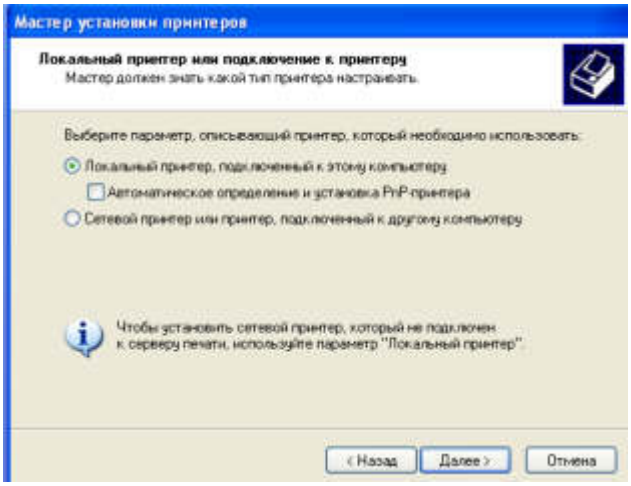
Подключите принтер к системному блоку.

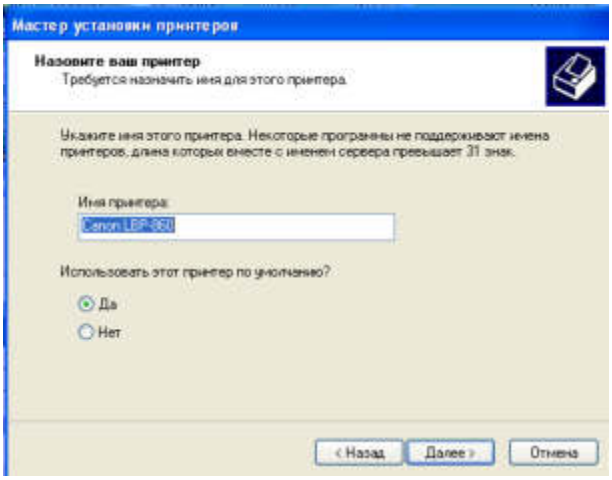
Произведите установку программного обеспечения принтера – драйвера. :

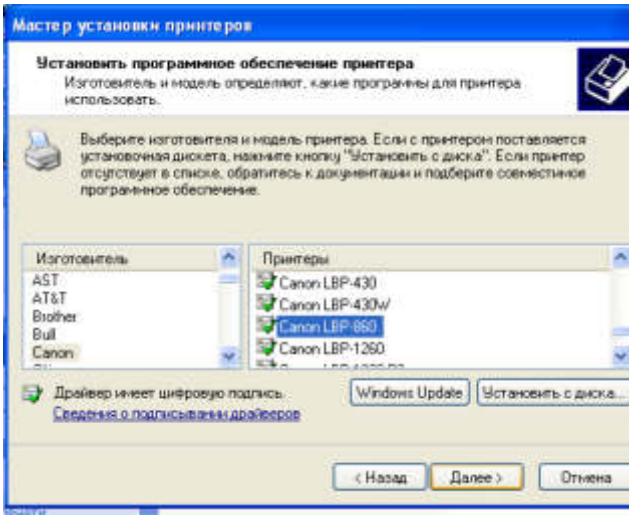
1. Запустите команду Пуск – Панель управления – Принтеры и факсы
2. В открывшемся окне Принтеры и факсы выберите команду Файл - установить принтер

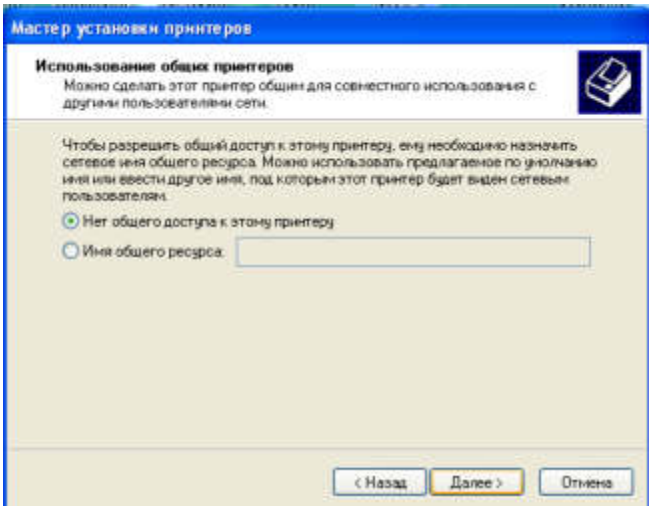


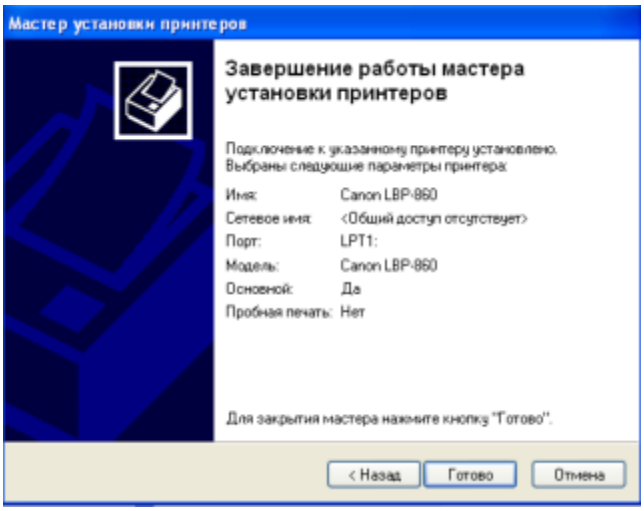
3. Далее действуйте по шагам мастера установки, выбирая варианты согласно рисункам:

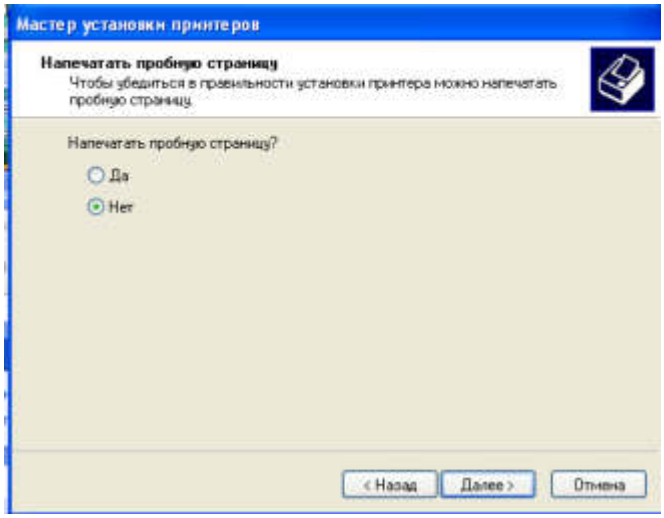






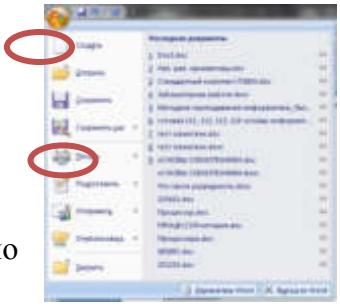






Таким образом, принтер установлен.

Результат можно посмотреть в окне Принтеры и факсы (*Пуск – Панель управления – Принтеры и факсы*). Ваш принтер появился в списке принтеров.



2. Использование свойств принтера при печати

Откройте текстовый редактор
(Пуск – Программы – Microsoft Office - Microsoft Word 2007)

Запустите команду Печать главного меню окна.

Изучите все возможности печати документов (какие изменения можно производить при печати).

Результат зафиксируйте в отчете.

Контрольные вопросы **к заданию**:

1. Расскажите, как осуществляется питание различных устройств ПК.
2. Посчитайте количество необходимых розеток электропитания, если в состав ПК входят сканер и принтер.
3. В чем состоит особенность электропитания мониторов?
4. Какие устройства используются в ПК для стабилизации напряжения в сети?
5. Какое устройство в составе ПК позволяет стандартно завершить работу аппаратуры при внезапном отключении электропитания?
6. Как осуществляется подключение электропитания и внешних устройств в компьютере?

Отчет о практической работе должен содержать:

1. Тему работы
2. Цель
3. Используемые инструменты ИТ.
4. Результаты практической части работы.
5. Ответы на контрольные вопросы.

Практическая работа №3
Установка операционной системы Windows

1. Цель работы: изучить этапы и способы установки Windows 7

2. Оборудование, приборы, аппаратура, материалы: персональный компьютер, видеоматериал, рабочая тетрадь.

3. Краткие теоретические сведения.

Обзор Windows 7

Windows 7 – наиболее новая клиентская ОС фирмы Microsoft, выпущенная в октябре 2009 года. От предыдущих версий Windows она отличается значительно более удобным пользовательским интерфейсом, с которым Вы и познакомитесь в данной работе, и более высоким быстродействием. Система все больше завоевывает популярность пользователей.

Запуск системы

- Включите компьютер с инсталлированной Windows 7.
- Менее чем через минуту система загружается и готова к работе.
- Выводится начальное меню с именами пользователей.

Вход в систему и аутентификация пользователя

Выберите Ваше имя пользователя и кликните мышкой по картинке рядом с именем. Как правило, в систему уже введено стандартное имя User. Если для пользователя установлен пароль, введите его.

После входа в систему на экране визуализируется рабочий стол.

Структура рабочего стола, мой компьютер, панель управления

Рабочий стол состоит из иконок приложений (например, Internet Explorer) и панели задач (taskbar) – обычно серого цвета, в нижней части. В левом нижнем углу расположена кнопка Start, при нажатии на которую пользователь может выбрать начальное действие – запуск какого-либо приложения, создание документа и др.

Вид и фон рабочего стола при разных настройках могут отличаться. Для изменения фона рабочего стола необходимо на фоновом рисунке нажать правую кнопку мыши и в контекстном меню выбрать Properties / Desktop, после чего выбрать нужный рисунок фона в выпадающем списке.

Основные пункты стартового меню, визуализируемого в результате нажатия кнопки Start:

- Computer – информация о компьютере, его ресурсах, устройствах, имени, установеленной на нем ОС
- Documents – стандартная папка для создаваемых документов (Вы можете помещать документы и в любую другую более удобную Вам папку)
- Control Panel – панель управления (рис. 36.3)
- Search programs and files – поиск и запуск программ и открытие файлов
- (в нижней части) Shut down – выход из Вашего пользовательского сеанса, выключение компьютера или перезапуск системы.

Рассмотрите более подробно панель управления. Она позволяет управлять ресурсами компьютера. Например, пункт Programs and Features позволяет инсталлировать новые программы, деинсталлировать или инсталлировать вновь ("ремонтировать") уже установленные.

Выберите в стартовом меню пункт Мой компьютер. При этом в специальном окне визуализируется информация о состоянии компьютера.

В окне Мой компьютер визуализируется информация о дисках и некоторых наиболее важных папках и предлагается набор возможных действий и набор других информационных узлов для перехода к ним (например, Network).

Для визуализации основных свойств компьютера (системной информации) выберите в стартовом меню: Мой компьютер / (Правая кнопка мыши) / Свойства. Возникает окно с системной информацией.

Вы видите информацию об ОС, объеме памяти, типе процессора и ряд ссылок, например, Device Manager, кликнув на которую, получите подробную информацию о составе оборудования компьютера и установленных драйверах. Интерфейс оформлен в виде веб-страницы.

Работа с файлами и папками

Работа с файлами и папками (folders) – хранилищами ссылок на файлы и другие папки – осуществляется с помощью программы Windows Explorer.

Выбор файла или папки в директории выполняется одним кликом мышки, вход в директорию или открытие файла – двойным кликом мышки на имени директории или файла. При этом для файла выполняется действие его открытия, зависящее от его типа, - для текстовых файлов – вызов соответствующего редактора (notepad, WordPad, MS Word и др.), для файлов .pdf – вызов Adobe Acrobat, для исполняемых кодов или командных файлов – запуск соответствующей программы или скрипта и т.д. Поэкспериментируйте на своем компьютере с навигацией по файлам и папкам и открытием файлов с документами.

Есть несколько способов запустить программу:

- из Windows Explorer – дважды кликнуть на имени ее файла;

- из меню Start – выбрать пункт Search programs and files. Это – одно из самых удобных нововведений в пользовательском интерфейсе системы Windows 7. При поиске, по мере набора имени программы, выводятся списки программ и файлов с таким именем (префиксом имени), что позволяет очень комфортно выбирать программу для запуска. Выбрав имя редактора notepad. получите окно для его запуска кликом мышки на имени программы.
- Запуск программы из командной строки (Command Prompt): выберите Start / Search Programs and Files / cmd. После запуска командного процессора визуализируется его окно. В окне командной строки наберите имя программы (например, notepad) и нажмите Enter.

Сетевые установки

Для подсоединения компьютера к локальной TCP/IP - сети необходимо выполнить для него сетевые установки – задать IP-адрес и сетевую маску.

Физическое подсоединение к сети сделайте (проверьте) путем подсоединения к сетевому разъему (RJ45) сетевого кабеля вида twisted pair (витая пара), который соединяет Ваш компьютер с сетевым концентратором (hub) или переключателем (switch). Наличие физического соединения индицируется зеленым световым индикатором (проверьте).

Для соединения в сеть служит сетевая карта (сетевой адаптер). Ваша задача – правильно задать IP-адрес компьютера.

Работа на удаленных компьютерах

При работе в локальной сети очень полезная возможность Windows 7 – удаленный вход на другой компьютер Вашей локальной сети. В Windows такая функция системы называется Remote Desktop Connection (удаленный рабочий стол). Для соединения Вы должны знать имя другого компьютера, например, aphrodite.

Настройка оборудования и звука в Windows 7

В этом меню в панели управления теперь собраны все настройки касающиеся внешнего оборудования Вашего компьютера.

1.Устройства и принтеры. Здесь собрано все оборудование, которое так или иначе можно отключать и подключать к компьютеру через порт или через сетевое соединение. Так же, здесь отображается сам компьютер.

1. Для каждого устройства доступны как общие настройки, такие как обновление драйвера или удаление его, так и индивидуальные. К примеру, если кликнуть правой кнопкой мыши по значку мыши мы увидим, что возможно прямо отсюда изменить параметры мыши и указателей.
2. Если устройство подключено к компьютеру, но не отображается в списке, то Вам следует воспользоваться мастером добавления устройства, благодаря которому новое устройство будет распознано и установлено. Для этого нажмите кнопку "Добавление устройства" в верхнем левом углу.

2. Автозапуск. Выбрав пункт автозапуск Вы сможете настроить автоматический запуск определенных типов устройств или носителей.

3. Звук. В настройках звука есть выбор устройства для воспроизведения и для записи по умолчанию. Здесь, как и в меню персонализации, вы можете настроить и выбрать звуковую схему оформления. Так же, здесь появилась функция, позволяющая автоматически понижать громкость всех системных звуков при поступлении звонка по Voip-каналу. Автоматически громкость может быть понижена на 80 или 50%, а так же Вы можете отключить все системные звуки при поступлении звонка или оставить их без изменения.

Выход из системы

Для выхода из Вашего сеанса пользователя выберите Start / Shut down.

4. Задания

Задание 1. Просмотрите предложенный видеоматериал. Установка и настройка операционной системы Windows 7.

Задание 2. Напишите понятия и классификацию:

- Операционная система (ОС)
- Операционная система как виртуальная машина
- Операционная система как менеджер ресурсов
- Операционная система как защитник пользователей и программ
- Операционная система как постоянно функционирующее ядро
- Классификация ОС

Задание 3. Опишите этапы подготовки к установке Windows 7

Задание 4. Заполните таблицу. Установка и настройка Windows 7

№ Задание Этапы установки Windows 7. Опишите действия на каждом из этапов.

1. Установка Windows 7 Подготовка жесткого диска. Загрузка с установочного диска. Определение параметров установки.

Следование инструкциям мастера установки.

Установка драйверов.

Активация ОС

2. Первичные настройки Windows 7 Основные этапы настройки Windows 7. Опишите действия на каждом из этапов.

Персонализация Свойства компьютера Учетные записи Антивирусное ПО

3.Контекстное меню Windows 7 Опишите работу с контекстным меню Windows 7

4.Оборудование и звук Windows 7 Этапы настройки оборудования и звука Windows 7. Опишите действия на каждом из этапов.Устройства и принтеры Автозапуск Звук

5.Сетевые настройки Windows 7 Этапы настройки домашней сети Windows 7. Опишите действия на каждом из этапов.

Настройка сетевого подключения Windows 7.

В ОС Windows 7 настройка сетевого подключения сводится к следующим этапам:

•Подсоедините компьютер к свитчу, роутеру или концентратору. Для этого подключите коннектор RJ-45 к порту Ethernet. Чтобы быть уверенным в успешном коннекте к сети, щелкните вкладку «Сеть». Откроется окно со значками, свидетельствующими о добавлении устройств в сеть.

- В Windows 7 настройки сетевого подключения в основном вращаются вокруг апплета «Сеть». Эта папка является удобным инструментом доступа к узлам сети. Итак, если не обнаруживается значок, то возможно было отключено обнаружение или общий доступ.

1. Разработка схемы сети
- 2. Выбор необходимого оборудования и поставщика услуг Интернета (дополнительно)
- 3. Настройка маршрутизатора
- 4. Подключение маршрутизатора к Интернету (дополнительно)
- 5. Подключение компьютеров и устройств к сети
- 6. Создание домашней группы или включение общего доступа к файлам и принтерам

5. Содержание отчета

Отчет должен содержать:

1. Название работы.
2. Цель работы.
3. Задание и его решение.
4. Вывод по работе.

6. Контрольные вопросы

1. Что такое операционная система?
2. Как сделать загрузочный диск (флэшнакопитель)?
3. Какие способы запуска программ в Windows 7 применяются?

Практическая работа№ 4
«Создание таблиц в тестовом редакторе WORD»
Построение таблиц

ХОД РАБОТЫ

1. Откройте текстовый редактор MS Word.
2. Создайте таблицу, состоящую из 7 **столбцов** и 7 **строк**.
3. Заполните таблицу следующим образом

<table>
<tr><th></th><th>понедельник</th><th>вторник</th><th>среда</th><th>четверг</th><th colspan="2">пятница</th><th>суббота</th></tr>
<tr><td>1</td><td rowspan="2">Математика</td><td>Математика</td><td>Русский язык</td><td>География</td><td>Математика</td><td rowspan="6">дежурный</td><td>ИЗО</td></tr>
<tr><td>2</td><td>Литература</td><td>Ин. Язык</td><td>Русский язык</td><td rowspan="2">История</td><td rowspan="2">Технология</td></tr>
<tr><td>3</td><td>География</td><td>История</td><td>Биология</td><td rowspan="2">Литература</td></tr>
<tr><td>4</td><td>Русский язык</td><td rowspan="2">ОБЖ</td><td>Математика</td><td>Ин. язык</td><td>Риторика</td></tr>
<tr><td>5</td><td></td><td rowspan="2">Физкультура</td><td>Информатика</td><td></td><td></td></tr>
<tr><td>6</td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

4. Измените цвет шрифта текста по своему усмотрению.
5. Заштрихуйте пустые ячейки таблицы.
6. Залейте ячейки таблицы.
7. Вставьте в вашу таблицу расписание звонков уроков.
8. Сделайте высоту строк таблицы 1 см.

	Расписание звонков	**понедельник**	**вторник**	**среда**	**четверг**	**пятница**		**суббота**
1	8^{00}-8^{40}	Математика	Математика	Русский язык	География	Математика	Дежурный	ИЗО
2	8^{45}-9^{25}		Литература	Ин. Язык	Русский язык	История		Технология
3	9^{35}-10^{15}	География	История	Биология	Литература			
4	10^{25}-11^{05}	Русский язык	ОБЖ	Математика		Ин. язык		Риторика
5	11^{15}-11^{55}			Физкультура	Информатика			
6	12^{00}-12^{40}							

Практикум «тестовый редактор WORD»

Задание 1

Создайте таблицу:

Устройства ввода информации				**Устройства вывода информации**				**Устройство сохранения информации**					**Устройство обработки информации**
					принтер			внутренняя память		внешняя память			
клавиатура	мышка	сканер	микрофон	монитор	матричный	струйный	лазерный	колонки	постоянная	оперативная	винчестер	дискета	диски	процессор

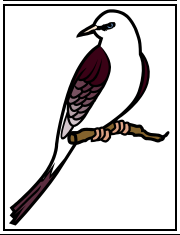
Ячейку **диски** преобразуйте следующим образом:

диски		
CD	DVD	Blu-ray

Задание 2

Создайте таблицу

Схема классификации типа хордовых

Тип	**хордовые**						
Класс	**млекопитающие**						**птицы**
Отряд	**грызуны**				**хищные**		
Семейство	**мышиные**		**беличьи**		**собачьи**	**кошачьи**	
Род	**мыши**	**крысы**	**белки**	**бурундуки**			
Вид							

5. Содержание отчета

Отчет должен содержать:

1. Название работы.
2. Цель работы.
3. Задание и его решение.
4. Вывод по работе.

ПРАКТИЧЕСКАЯ РАБОТА № 5
«Работа в текстовом редакторе форматирование документов

РЕДАКТИРОВАНИЕ И ФОРМАТИРОВАНИЕ ДОКУМЕНТА

Текстовые редакторы — это программы для создания, редактирования, форматирования сохранения и печати документов. Современный документ может содержать, кроме текста, и другие объекты (таблицы, диаграммы, рисунки и т. д.).

Более совершенные текстовые редакторы, имеющие целый спектр возможностей по созданию документов (например, поиск и символов, средства проверки орфографии, вставка таблиц и др.), называют иногда текстовыми процессорами. Примером такой программы является Word из офисного пакета Microsoft Office.

Все действия, которые выполняет пользователь над документом, можно разделить на две большие группы: редактирование и форматирование.

Редактирование – преобразование, обеспечивающее добавление, удаление, перемещение или исправление содержания документа. Редактирование документа обычно производится путем добавления, удаления или перемещения символов или фрагментов текста.

К операциям редактирования относятся следующие действия: набор текста; исправление опечаток; копирование, перестановка, удаление частей текста; вставка рисунков, таблиц и других информационных объектов.

Форматирование — преобразование, изменяющее форму представления документа. В начале работы над документом целесообразно задать параметры страницы: ее формат (размер), ориентацию, размер полей и др.

К операциям форматирования относятся различные способы выделения текста, а именно: изменение свойств символов; изменение свойств абзацев; оформление заголовков и подзаголовков; преобразование текста в список; преобразование текста в табличный вид; вставка колонтитулов, номеров страниц и пр.

Задание 1.2. Наберите текст на казахском языке

АҚПАРАТ

Қоғамның қазіргі кездегі дамуының ең басты - белгісі бұл өндірістің, тұтынудың және адам әрекеттерінің барлық салаларында ақпарат жинаудың артуы болып табылады. Адамның бар өмірі - қалай да болсын - ақпарат алу, жинау және өңдеумен байланысты.

Біз өмір сүріп отырған ғасырға автомобиль, электр, авиация, атом энергиясы, космонавтика, электрондық техника ғылымының жылдам даму, кітап басып шығару мен радиоқабылдағыштың арасы 440 жуық жыл болса, радио мен теледидар арасында 30, ал алғашқы транзистор мен интегралдық схеманың арасы 5 жыл болған.

Ақпарат - адамның өзіне қарағанда әлдеқайда көне құбылыс. (Жартастағы суреттер, "Жазбалар ғибадатханасы", Майя өркениетінің иероглифті тақталары да, ежелгі египеттіктердің папирустары, гректердің абак тастары).

Задание 1.3. Наберите текст на английском языке

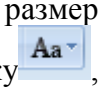
COMPUTERS

Computer is an electronic device that can receive a set of instructions called program and then carry out them. The modern world of high technology could not be possible without computers. Different types and sizes of computers find uses throughout our society. They are used for the storage and handling of data, secret governmental files, information about banking transactions and so on.

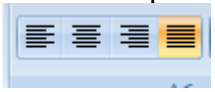
Computers have opened up a new era in manufacturing and they have enhanced modern communication systems. They are essential tools in almost every field of research, from constructing models of the universe to producing tomorrow's weather reports. Using of different databases and computer networks make available a great variety of information sources.

There are two main types of computers, analog and digital, although the term computer is often used to mean only the digital type, because this type of computer is widely used today. That is why I am going to tell you about digital computers.

РАЗДЕЛ 2.

1. После того, как вы набрали текст, полностью выделите его, используя сочетание клавиш ctrl+A,
2. Используя группу команд «Шрифт» установите для текста шрифт Times New Roman, размер (кегль) 14, заголовки (название) текстов оформите прописными буквами, используя кнопку 

, примените к ним полужирное начертание

.
3. Затем, используя группу команд «Абзац», установите для текста междустрочный интервал множитель 1.15, установите абзацный отступ 1,25 см, с помощью кнопок  выровняйте основной текст по ширине, заголовки текстов выровняйте по центру.
4. С помощью вкладки «Разметка страницы», расположенного на ленте главного меню и кнопки «Поля», затем команды «Настраиваемые поля», установите для документа следующие параметры полей: левое- 3 см, правое- 1,5 см, верхнее- 2см, нижнее- 1см.

5. С помощью вкладки «Вставка», используя команду «Номер страницы» добавьте номер страницы в свой документ. Расположите номер страницы внизу страницы по центру.
6. Выделите и скопируйте из текста на русском языке следующий абзац: *«К операциям редактирования относятся следующие действия: набор текста; исправление опечаток; копирование, перестановка, удаление частей текста; вставка рисунков, таблиц и других информационных объектов»*. Вставьте его в конец вашего документа. Затем, используя кнопку «Маркеры» в группе команд «Абзац», маркер «--», произведите форматирование этого фрагмента по следующему образцу:

К операциям редактирования относятся следующие действия:

- набор текста;
- исправление опечаток;
- копирование, перестановка,
- удаление частей текста;
- вставка рисунков, таблиц и других информационных объектов.

РАЗДЕЛ 3.

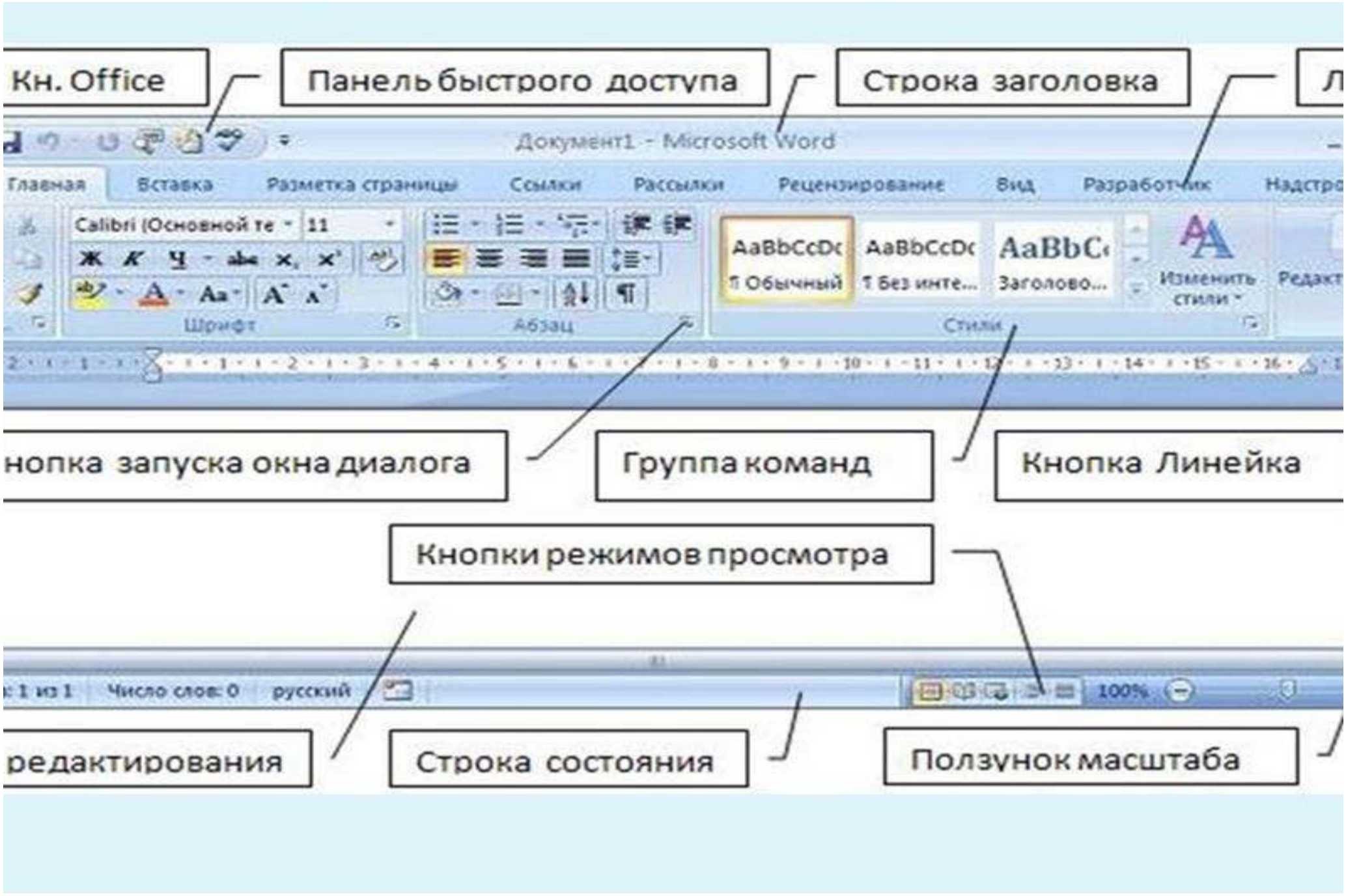
В вашей поименованной папке создайте папку «Практическая работа №5» и сохраните созданный документ в эту папку, дав ему название «Редактирование и форматирование текста». Определите тип, расширение и вес (размер) файла.

Отчет

Отчет должен быть оформлен в текстовом редакторе и содержать:

- наименование работы;
- цель работы;
- задание;
- последовательность выполнения работы;
- ответы на контрольные вопросы;

вывод о проделанной работе


Кн. Office
Панель быстрого доступа
Строка заголовка
Документ1 - Microsoft Word
Главная
Вставка
Разметка страницы
Ссылки
Рассылки
Рецензирование
Вид
Разработчик
Calibri (Основной те
11
Шрифт
Абзац
Стили
Изменить стили
нопка запуска окна диалога
Группа команд
Кнопка Линейка
Кнопки режимов просмотра
Число слов: 0
русский
100%
редактирования
Строка состояния
Ползунок масштаба

Практическое занятие №6

Работа в текстовом редакторе создание авто оглавления

Цель работы – изучение функциональных возможностей текстового процессора Word 2007 и приобретение навыков практической работы по сложному форматированию текстовых документов.

Задания для практического занятия и инструктаж по их выполнению

1. Изучить теоретические сведения. Кратко записать в тетрадь главное
2. Разобрать примеры.
3. Выполнить практические задания по данной теме на компьютере.

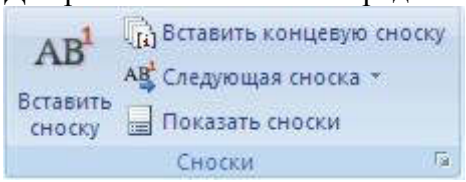
Теоретические сведения

Сложное форматирование документов Word 2007

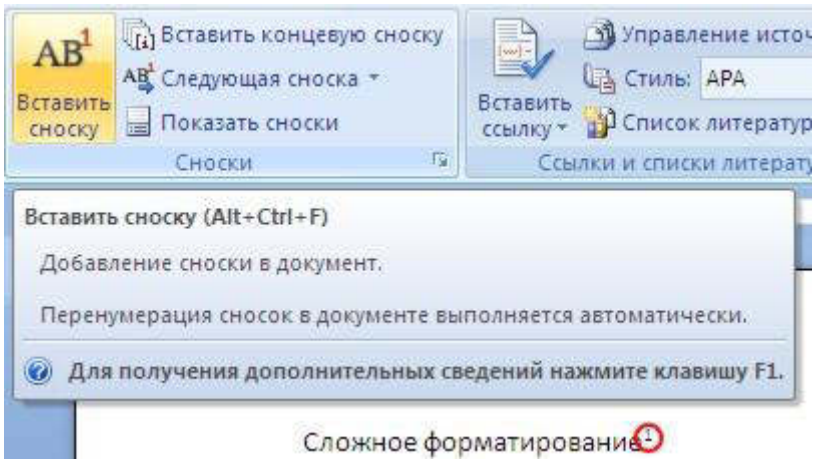
Сноски предназначены для добавления к тексту комментариев, объяснений, указания источника информации.

Сноски бывают обычные (в конце страницы) и концевые (в конце всего текста).

Для работы со сносками предназначена панель "Сноски".

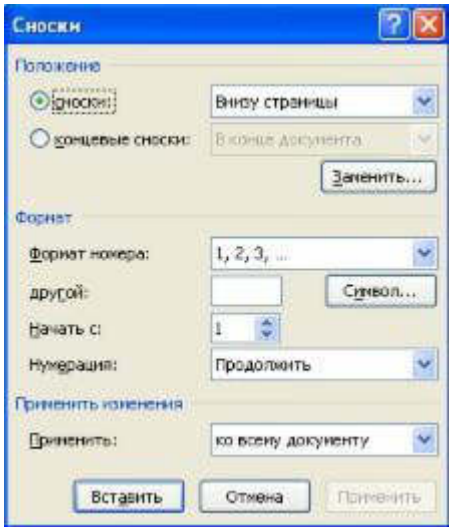


Для вставки обычной сноски необходимо нажать кнопку *"Вставить сноску" (Ctrl+Alt+F)*. В тексте, в том месте где находился курсор появится значок сноски, а внизу страницы - горизонтальная разделительная линия и номер сноски.



Для вставки концевой сноски предназначена кнопка *"Вставить концевую сноску" (Ctrl+Alt+D)*.

Для более тонких настроек сносок служит окно панели *"Сноски"*.



Сноски нумеруются автоматически в соответствии с выбранной пользователем системой нумерации. При добавлении новой сноски или удалении существующей остальные перенумеровываются.

Перемещаться между сносками можно при помощи кнопки "Следующая сноска".

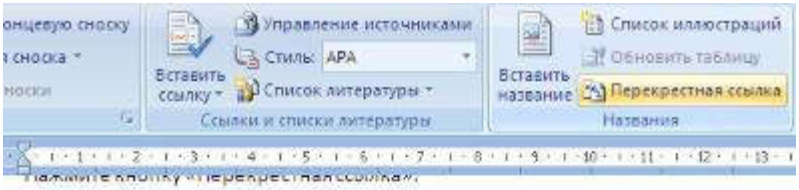
Для удаления сноски необходимо ее выделить, а затем нажать клавишу Delete.

Перекрестные ссылки

Перекрестные ссылки служат для быстрого перехода к нужному элементу документа.

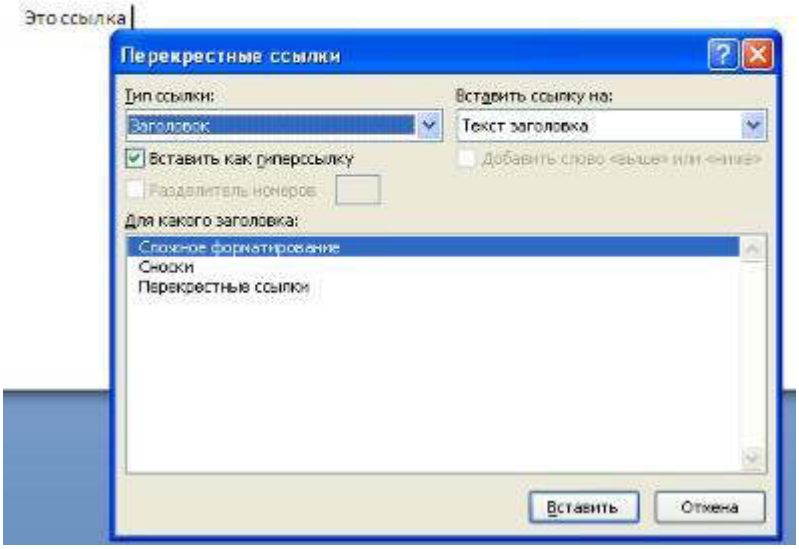
В Ворде 2007 можно создавать перекрестные ссылки на следующие элементы: заголовки, сноски, закладки, названия, нумерованные абзацы. Инструменты для работы с перекрестными ссылками находятся на панели "Названия".

Перекрестные ссылки создаются только между элементами одного документа.



Введите текст, с которого будет начинаться перекрестная ссылка.

Нажмите кнопку "Перекрестная ссылка".



В открывшемся окне в выпадающем списке "Тип ссылки" надо выбрать тип элемента, на который будем ссылаться.

В выпадающем списке "Вставить ссылку на" надо указать данные, которые следует добавить в документ.

Для того, чтобы иметь возможность перехода к ссылаемому элементу флажок "Вставить как гиперссылку" должен быть установлен.

Перекрестные ссылки вставляются в документ в виде полей. Переключаться между режимами отображения кодов полей и значений полей можно при помощи сочетания клавиш Alt+F9.

```
Это ссылка { REF _Ref191718223 \h }
```

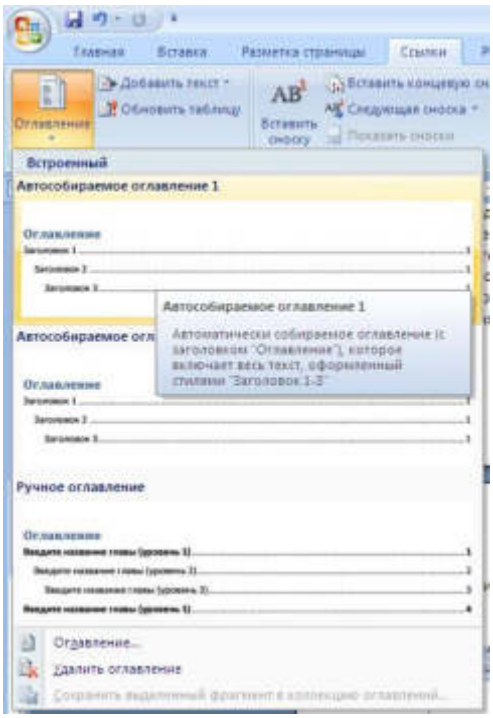
Изменить текст самой ссылки можно прямо в документе.

Оглавление

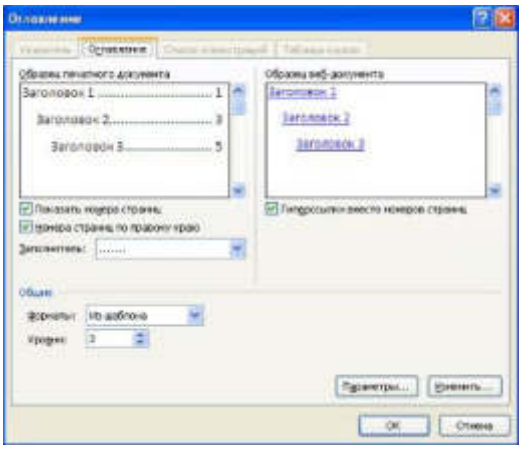
Оглавление - это список заголовков документа.

Для того чтобы быстро сделать оглавление, документ предварительно **должен быть отформатирован согласно встроенным форматам уровней структуры или стилей заголовков.**

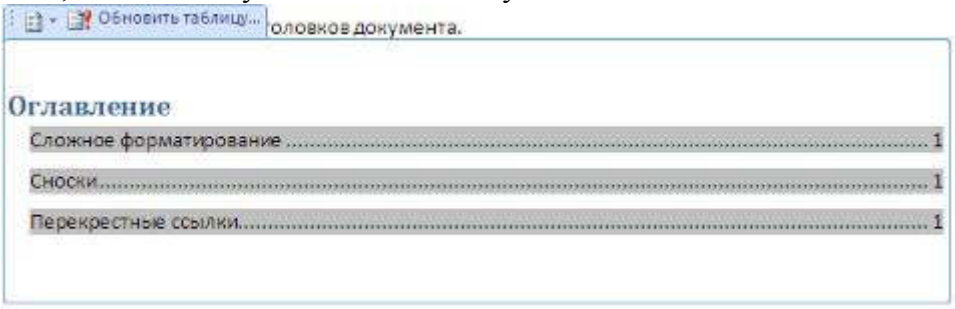
Затем, установив курсор в месте вставки оглавления, нажмите кнопку "Оглавление" панели "Оглавление". В открывшемся окне выберите нужный формат оглавления.



При необходимости тонких настроек оглавления нажмите "Оглавление.."



Для быстрой правки (обновления) уже существующего оглавления сделайте щелчок в поле оглавления, затем кнопку Обновить таблицу.



Предметный указатель

Предметный указатель - это список терминов, встречающихся в документе, с указанием страниц где они расположены.

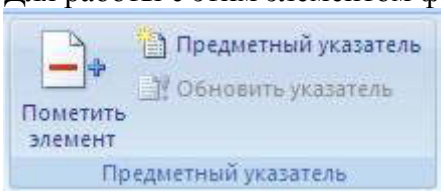
Предметный указатель можно создать для следующих элементов:

отдельных слов, фраз, символов;

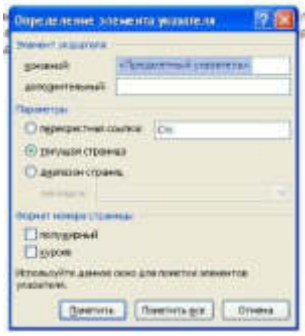
разделов;

ссылок.

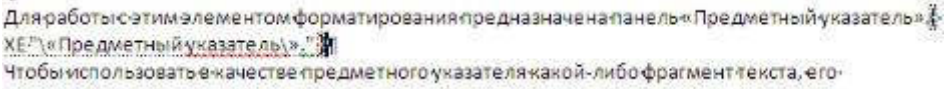
Для работы с этим элементом форматирования предназначена панель "Предметный указатель".



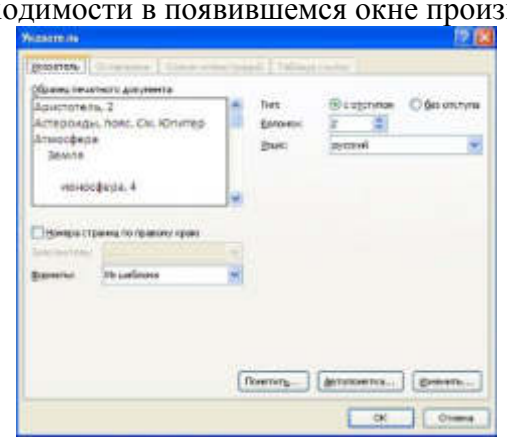
Чтобы использовать в качестве предметного указателя какой-либо фрагмент текста, его необходимо выделить, затем нажать кнопку "Пометить элемент" на панели "Предметный указатель".



При пометке текста в документе добавляется специальное скрытое поле.



Для окончательной сборки предметного указателя нажмите кнопку "Предметный указатель" и при необходимости в появившемся окне произведите окончательные настройки.

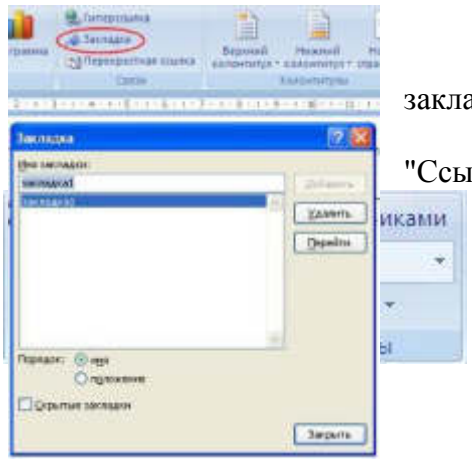


Закладки

Закладки предназначены для быстроты и удобства навигации по документу - они позволяют быстро переходить к ранее помеченным местам в тексте. Для того, чтобы сделать закладку необходимо

установить курсор в нужном месте документа и нажать кнопку "Закладка" на панели "Связи" ленты "Вставка".

В появившемся окне необходимо ввести имя закладки. Следует иметь ввиду, что имя должно начинаться с буквы и не содержать пробелов.



При помощи этой же кнопки и окна можно перемещаться по закладкам, добавлять новые и удалять ненужные.

Для работы с библиографией и цитатами служит панель "Ссылки и списки литературы"

Библиография - это список литературных или других источников, которые использовались при подготовке документа. Как правило, она помещается в конце текста. При вставке цитат надо указывать источник откуда они взяты, поэтому понятия "библиография" и "цитаты" тесно взаимосвязаны.

Данные элементы форматирования присутствуют, как правило, в научных работах, поэтому, более подробно на них останавливаться не будем.

Задание

Откройте документ text1.docx (текст в Приложении)

Для вновь созданного документа, используя выделенные полужирным шрифтом команды, установить следующие параметры:

1.Разметка страницы – Поля - Настраиваемые поля (поле слева: 2,5 см, поле справа: 1,5 см, поле сверху: 1,5 см, поле снизу: 2 см, колонтитул сверху 1 см, колонтитул снизу: 1,2 см);

2.Разметка страницы – Размер (размер бумаги: А4, 21 х 29,7 см);

3.Разметка страницы - Ориентация (ориентация листа: книжная);

4.Главная – Шрифт (шрифт: Times New Roman, размер: 12 пунктов, начертание: обычный);

5.Разметка страницы - Расстановка переносов (установить автоматический перенос слов).

6.Главная-Абзац – Отступ (первая строка на 1 см)

Задание № 2

Перед каждым абзацем набранного Вами текста вставить разрывы, так, чтобы каждый абзац начинался с новой страницы.

Задание №3

Вставьте в документ номера страниц:

Вставка-Номер страницы- Вверху страницы- Страница X из Y- Полужирные номера 3

Задание №4

Вставьте в документ колонтитулы:

Вставка- Нижний колонтитул- Консервативный

В колонтитуле укажите фамилию автора (себя) и наименование документа

Задание № 5

В начале четвертого абзаца установите Буквицу. В результате Вы получите примерно следующий вид текста

Чтобы сохранить документ, воспользуйтесь командой Сохранить кнопки Офис. При первом сохранении документа откроется диалоговое окно Сохранение документа, позволяющее указать имя файла и его положение (папку). Файлы, относящиеся к одному проекту или объединенные по какому- либо иному принципу, рекомендуется хранить в одной папке. Это позволяет упорядочить информацию и упростить поиск данных.

Задание №6

Введите заголовки абзацев:

Введение
Форматирование
Колончатая верстка
Буквица
Связи и ссылки

Задание №7

Установите в пятом абзаце гиперссылки с предыдущими абзацами:

Найдите однокоренные слова в абзацах 5 и 4. Свяжите гиперссылкой эти однокоренные слова текста, и дайте концевую сноску с пояснением того или иного слова.

Задание №8

Примените сложное форматирование к заголовкам абзацев. Для этого выделите последовательно каждый заголовок и на нем установите стиль Заголовок 1 (лента меню ->вкладка Главная->группа Стили-> кнопка со стилем Заголовок 1).

Задание №9

На основе заголовков создайте автособираемое оглавление (см. выше, теоретический материал)

Задание №10

В режиме структуры поменяйте абзацы 2 и 3 местами. (перетаскивая знак ✚)

Задание №11

Обновите Оглавление. Теперь и в Оглавлении заголовки абзацев поменялись местами.

Сохраните результат в файле и отправьте преподавателю.

Отчет

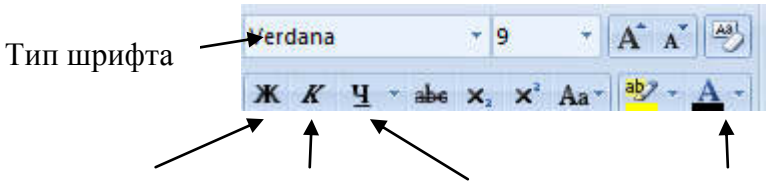
Отчет должен быть оформлен в текстовом редакторе и содержать:

- наименование работы;
- цель работы;
- задание;
- последовательность выполнения работы;
- ответы на контрольные вопросы;

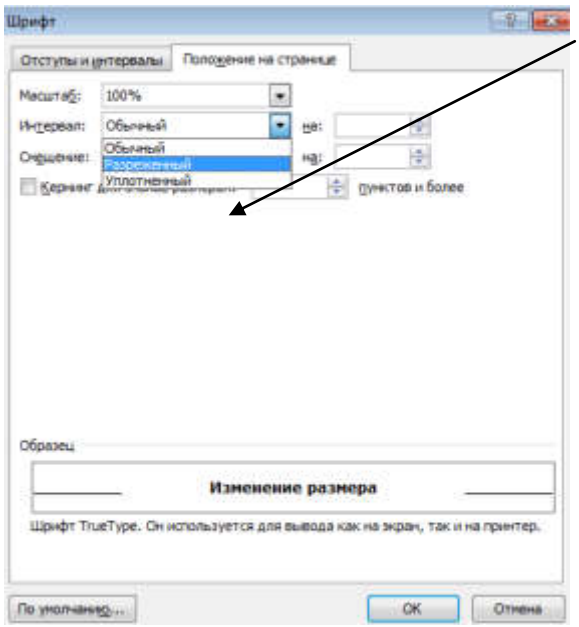
вывод о проделанной работе

Практическая работа №7
Создание документа в текстовом редакторе по заданным условиям.

1. Установка полей и ориентация бумаги.
Поля – области страницы, где не может размещаться текст.
Выполните команду: Разметка страницы → Поля → Настраиваемые поля.
Верхнее – 2 см.
Нижнее – 2 см.
Левое– 3 см.
Правое – 1 см.
Ориентация – книжная (Разметка страницы → Ориентация → книжная)
2. С самого начала приучайтесь работать в режиме специальных символов. Для этого нажмите кнопку **"Непечатаемые символы"** ¶ что очень важно для качественного набора текста и облегчения дальнейших операций по редактированию и форматированию текста.
Важно различать в самом начале курсор текстовый и курсор мыши.
3. Изменение размера, цвета шрифта, межбуквенного интервала в слове.



жирность курсив подчеркивание Цвет шрифта



Установка межбуквенного интервала в слове: 1. Выделить слово.
2. Вызвать контекстное меню (правой кнопкой мыши).
3. В раскрывшемся меню выберите Шрифт → Положение на странице.

4. Форматирование абзацев.

Междустрочный интервал

По центру

Выравнивание по

Выравнивание по левому краю

Выравнивание по правому краю

!! Для того, чтооы текст выровнять по правому, левому краю, по ширине и по центру, текст необходимо выделить, а потом уже нажимать на кнопку на панели Форматирования.

Задание: Напечатайте текст в соответствии с образцом:

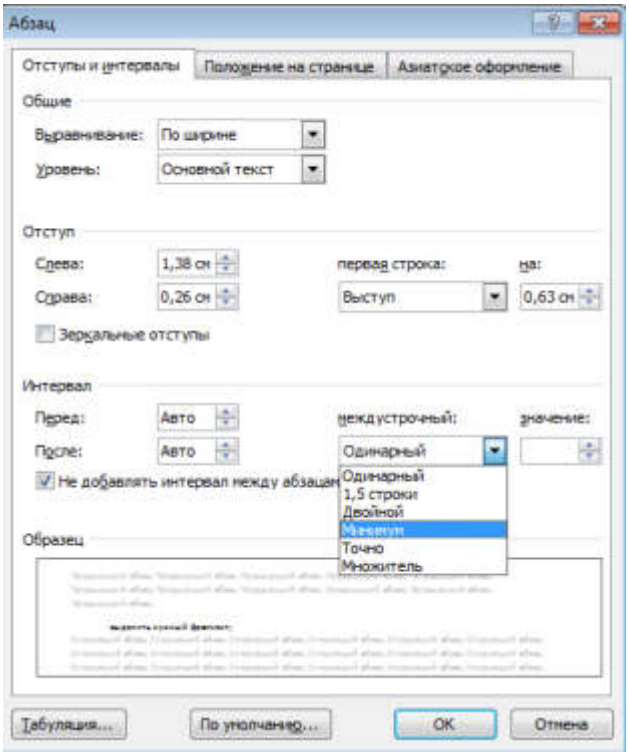
Этот текст выровнен по левому краю. Этот текст выровнен по левому краю. Этот текст выровнен по левому краю. Этот текст выровнен по левому краю. Этот текст выровнен по левому краю. Этот текст выровнен по левому краю. Этот текст выровнен по левому краю.

Этот текст выровнен по правому краю. Этот текст выровнен по правому краю. Этот текст выровнен по правому краю. Этот текст выровнен по правому краю. Этот текст выровнен по правому краю. Этот текст выровнен по правому краю. Этот текст выровнен по правому краю.

Этот текст выровнен по ширине. Этот текст выровнен по ширине. Этот текст выровнен по ширине. Этот текст выровнен по ширине. Этот текст выровнен по ширине. Этот текст выровнен по ширине. Этот текст выровнен по ширине. Этот текст выровнен по ширине. Этот текст выровнен по ширине.

Этот текст выровнен по центру. Этот текст выровнен по центру. Этот текст выровнен по центру. Этот текст выровнен по центру. Этот текст выровнен по центру. Этот текст выровнен по центру. Этот текст выровнен по центру. Этот текст выровнен по центру. Этот текст выровнен

по центру. Этот текст выровнен по центру. Этот текст выровнен по центру. Этот текст выровнен по центру.



!! Для того, чтобы уменьшить или увеличить интервал между строками необходимо:

- **выделить нужный фрагмент;**
- **Вызвать контекстное меню;**
- **Абзац;**
- **Выбираете нужный вам интервал между строк.**

Задание: Установите интервал в 1-м абзаце напечатанного текста – Одинарный; во 2-м абзаце – Двойной; в 3-м абзаце – 1, 5 строки; в 4-м – Точно.

5. Копирование, удаление, перемещение фрагмента.

- **Копирование фрагмента (выделить → контекстное меню → копировать; переместить курсор в место вставки: контекстное меню →** вставить)
- **Удаление фрагмента (выделить → контекстное меню → удалить)**
- **Перемещение (выделить → контекстное меню → вырезать; переместить курсор в место вставки: контекстное меню →** вставить **).**

Задание: Наберите текст по образцу используя операцию копирования.

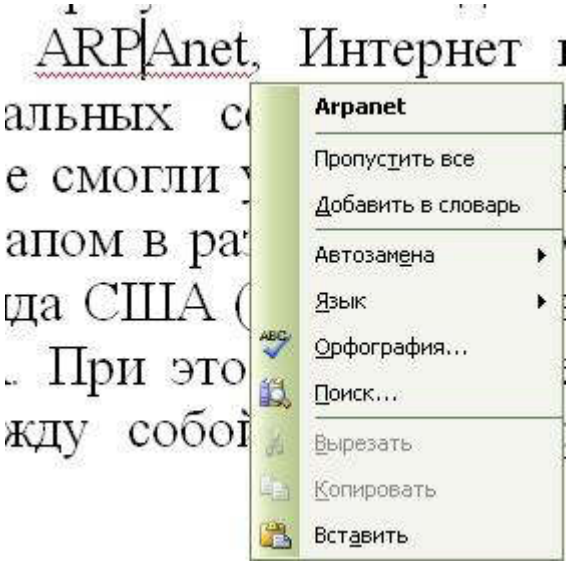
Хозяйка однажды с базара пришла
Хозяйка однажды с базара пришла,
Хозяйка с базара домой принесла:
Картошку,
Капусту,
Морковку,
Горох,
Петрушку и свёклу.
Ох! ..
Вот овощи спор завели на столе
Кто лучше, вкусней и нужней на земле:
Картошка?
Капуста?
Морковка?
Горох?
Петрушка иль свёкла?
Ох!..
Хозяйка тем временем ножик взяла.
И ножиком этим крошить начала:
Картошку,
Капусту,
Морковку,
Горох,

Петрушку и свёклу.
Ох!..
Накрытые крышкой, в душном горшке
Кипели, кипели в крутом кипятке:
Картошка,
Капуста,
Морковка,
Горох,
Петрушка и свёкла.
Ох!..
И суп овощной оказался неплох!

6. Наберите следующий текст: Тип шрифта - Courier New, шрифт разреженный, интервал между строк 1,5 строки. Выравнивание текста – по ширине.

XX век. 1994-й год был годом, когда многие люди впервые услышали о сети Интернет. Этому предшествовало несколько этапов. 2 января 1969 года Управление перспективных исследований (ARPA), являющееся одним из подразделений Министерства обороны США, начало работу над проектом связи компьютеров оборонных организаций. В результате исследований была создана сеть ARPAnet. Но в отличие от ARPAnet, Интернет вырос из множества небольших, независимых локальных сетей, принадлежащих компаниям и другим организациям, которые смогли увидеть преимущества объединения друг с другом. Следующим этапом в развитии Интернет было создание сети Национального научного фонда США (NSF). Сеть, названная NSFnet, объединила научные центры США. При этом основой сети стали пять суперкомпьютеров, соединенных между собой высокоскоростными линиями связи.

!! Обратите внимание, что некоторые слова в тексте подчеркнуты красной волнистой линией или зеленой волнистой линией. Значит Ваш Word настроен на автоматическую проверку орфографии и грамматики. Красная линия - орфографическая ошибка. Щелкаем правой кнопкой по подчеркнутому слову. Встроенный словарь предлагает слова для замены, вы выбираете слово из списка, если же слово набрано верно, и в этом уверены, можете его **Добавить в словарь**. Если сомневаетесь нажмите **Пропустить все.**



7. С помощью различных параметров форматирования приведите текст в соответствие с приведенным ниже образцом. Границу для текста можно выполнить командой: Разметка станицы → Границы страниц→ Поля

Винегрет овощной

Картофель – 3 шт
Морковь -2 шт
Свекла – 1шт
Соленые огурцы – 2 шт
Лук зеленый – 50 г
Масло растительное – 2 ст. ложки
Перец молотый, горчица, укроп – по вкусу
Листья салата

Огурцы, вареный картофель, свеклу, морковь нарезать тонкими ломтиками, лук нашинковать. Овощи выложить в посуду, перемешать, заправить маслом с добавлением перца, соли, горчицы. Готовый винегрет поставить в холодильник.

При подаче на стол винегрет уложить горкой в салатник, украсить зеленым салатом, посыпать укропом.

8. Набрать текст рекламного объявления.

КОМПЬЮТЕРЫ

Ptntium – 200
Ptntium II

- Самые дешевые комплектующие
- Модернизация старых компьютеров
- Курсы компьютерной грамотности

Телефон
2-22-03

Приложение

Чтобы ввести в документ текст, достаточно начать его печатать на клавиатуре компьютера.

Вводимые символы появляются в том месте экрана, где находится курсор, который сдвигается вправо, оставляя за собой цепочку символов. При достижении правого поля страницы курсор автоматически перемещается в следующую строку. Этот процесс называется перетеканием текста, а нажатие на клавишу Enter создает новый абзац, а не новую строку.

Текст, который отображается в окне документа, хранится в оперативной памяти компьютера. Его можно отредактировать и напечатать, но при завершении работы с Word он будет утерян. Поэтому, чтобы сохранить введенный текст, нужно записать документ в файл на жесткий диск компьютера. Тогда его можно будет открыть позже и продолжить работу.

Чтобы сохранить документ, воспользуйтесь командой **Сохранить** кнопки Офис. При первом сохранении документа откроется диалоговое окно *Сохранение документа*, позволяющее указать имя файла и его положение (папку). Файлы, относящиеся к одному проекту или объединенные по какому- либо иному принципу, рекомендуется хранить в одной папке. Это позволяет упорядочить информацию и упростить поиск данных.

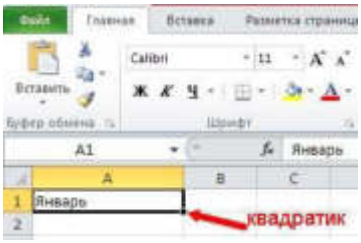
Все последующие версии документа будут сохраняться в том же файле, причем новая версия документа замещает предыдущую. Если требуется сохранить обе версии документа (исходную и содержащую

последние изменения), воспользуйтесь командой Сохранить, указав имя и положение нового файла. Документ можно сохранить в той же папке, открыть другую папку или создать новую.

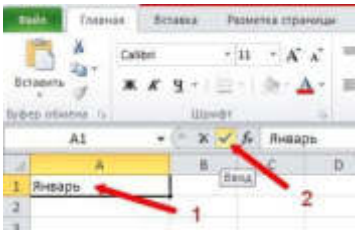
Отчет

Отчет должен быть оформлен в текстовом редакторе и содержать:

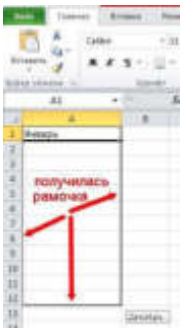
- наименование работы;
- цель работы;



- задание;
- последовательность выполнения работы;
- ответы на контрольные вопросы;

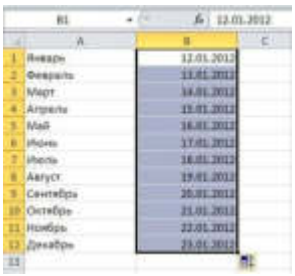


вывод о проделанной работе



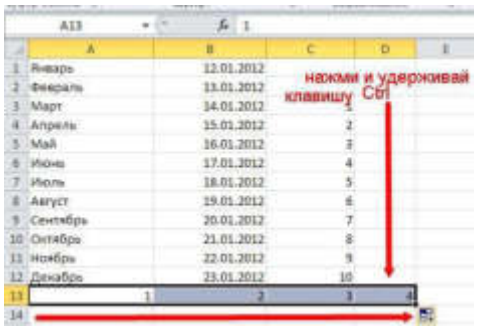
Практическая работа №8
Работа в табличном редакторе автозаполнение

Работа в Excel может оказаться очень трудоемкой, если не использовать встроенные функции, которые существенно облегчают работу с таблицами. Многие операции приходится повторять по несколько раз, а это отнимает много времени и сил. Для автоматизации таких процессов разработчики внедрили в программу очень удобную функцию, которая называется Автозаполнение ячеек в Excel.



АВТОЗАПОЛНЕНИЕ ЯЧЕЕК В EXCEL

1. Возьмем и создадим для примера ***список данных в Excel*** по месяцам.



Пишем в ячейке **A1** слово «**Январь**» (естественно без кавычек и можно с прописной буквы) и щелкаем мышкой по кнопке «**Enter**» (или галочка на панели инструментов в Excel).

Теперь поместите указатель мыши на правый нижний угол ячейки **A1**, туда, где расположен маленький квадратик. Указатель мыши должен измениться с белого креста на маленький черный крестик. Как только это произойдет, нажмите на левую клавишу мыши и, не отпуская её, переместите указатель на ячейку **A12**. У вас получиться вот такая рамочка.

Как только вы отпустите кнопку мышки выделенные ячейки заполняться названиями месяцев.

Теперь попробуем создать ***список данных в Excel*** по дате.

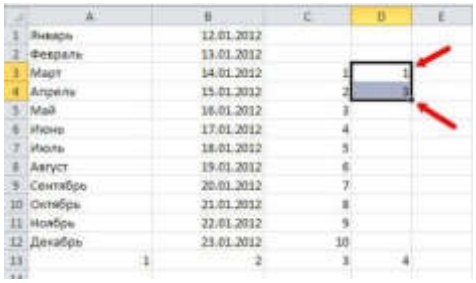
Напишем в ячейке **B1** дату – **12.01.2012** – и, установив указатель мыши на ***маркер заполнения***, выделим диапазон ячеек **B1:B12**.

КАК ПРОНУМЕРОВАТЬ СТРОКИ В EXCEL

Если вы хотите просто ***пронумеровать строки в колонке***, то напишите в ячейке цифру **1**, и выделите необходимый диапазон ячеек при помощи маркера заполнения и нажатой клавиши **Ctrl** (на клавиатуре). Нажимая клавишу **Ctrl**, вы даете команду увеличивать цифру на единицу.
Таким же образом вы можете ***пронумеровать столбцы в Excel***.

КАК ЗАДАТЬ СПИСОК НЕЧЕТНЫХ ИЛИ ЧЕТНЫХ ЧИСЕЛ

- Для этого напишите в ячейке **D3** число (допустим – **1**), а в ячейке **D4** – число – **3**.
- Выделите ячейки **D3** и **D4**.
- Потом установите указатель мыши на маркер заполнения (маленький черный квадратик в нижнем правом углу).
- Удерживая нажатой клавишу **Ctrl**, опускаем выделение до ячейки **D12**.



	A	B	C	D	E
1	Январь	12.01.2012			
2	Февраль	13.01.2012			
3	Март	14.01.2012	1	1	
4	Апрель	15.01.2012	2	3	
5	Май	16.01.2012	3		
6	Июнь	17.01.2012	4		
7	Июль	18.01.2012	5		
8	Август	19.01.2012	6		
9	Сентябрь	20.01.2012	7		
10	Октябрь	21.01.2012	8		
11	Ноябрь	22.01.2012	9		
12	Декабрь	23.01.2012	10		
13	1	2	3	4	

- В результате у вас появится список нечетных чисел.

2. С помощью функции Автозаполнение создайте календарь на 2018г. по предложенному образцу.

КАЛЕНДАРЬ НА 2018 ГОД																						
	ЯНВАРЬ					ФЕВРАЛЬ					МАРТ						АПРЕЛЬ					
Понедельник	1	8	15	22	29		5	12	19	26		5	12	19	26			2	9	16	23	30
Вторник	2	9	16	23	30		6	13	20	27		6	13	20	27			3	10	17	24	
Среда	3	10	17	24	31		7	14	21	28		7	14	21	28			4	11	18	25	
Четверг	4	11	18	25		1	8	15	22		1	8	15	22	29			5	12	19	26	
Пятница	5	12	19	26		2	9	16	23		2	9	16	23	30			6	13	20	27	
Суббота	6	13	20	27		3	10	17	24		3	10	17	24	31			7	14	21	28	
Воскресенье	7	14	21	28		4	11	18	25		4	11	18	25			1	8	15	22	29	
	МАЙ					ИЮНЬ					ИЮЛЬ						АВГУСТ					
Понедельник		7	14	21	28		4	11	18	25		2	9	16	23	30		6	13	20	27	
Вторник	1	8	15	22	29		5	12	19	26		3	10	17	24	31		7	14	21	28	
Среда	2	9	16	23	30		6	13	20	27		4	11	18	25		1	8	15	22	29	
Четверг	3	10	17	24	31		7	14	21	28		5	12	19	26		2	9	16	23	30	
Пятница	4	11	18	25		1	8	15	22	29		6	13	20	27		3	10	17	24	31	
Суббота	5	12	19	26		2	9	16	23	30		7	14	21	28		4	11	18	25		
Воскресенье	6	13	20	27		3	10	17	24		1	8	15	22	29		5	12	19	26		
	СЕНТЯБРЬ					ОКТЯБРЬ					НОЯБРЬ						ДЕКАБРЬ					
Понедельник		3	10	17	24	1	8	15	22	29		5	12	19	26			3	10	17	24	31
Вторник		4	11	18	25	2	9	16	23	30		6	13	20	27			4	11	18	25	
Среда		5	12	19	26	3	10	17	24	31		7	14	21	28			5	12	19	26	
Четверг		6	13	20	27	4	11	18	25		1	8	15	22	29			6	13	20	27	
Пятница		7	14	21	28	5	12	19	26		2	9	16	23	30			7	14	21	28	
Суббота	1	8	15	22	29	6	13	20	27		3	10	17	24			1	8	15	22	29	
Воскресенье	2	9	16	23	30	7	14	21	28		4	11	18	25			2	9	16	23	30	

Отчет

Отчет должен быть оформлен в текстовом редакторе и содержать:

- наименование работы;
- цель работы;
- задание;
- последовательность выполнения работы;
- ответы на контрольные вопросы;

вывод о проделанной работе

Практическая работа №9

Работа в табличном редакторе создание диаграмм

Что собой представляет диаграмма. Диаграмма предназначена для графического представления данных. Для отображения числовых данных, введенных в ячейки таблицы, используются линии, полосы, столбцы, сектора и другие визуальные элементы. Вид диаграммы зависит от её типа. Все диаграммы, за исключением круговой, имеют две оси: горизонтальную – ось категорий и вертикальную – ось значений. При создании объёмных диаграмм добавляется третья ось – ось рядов. Часто диаграмма содержит такие элементы, как сетка, заголовки и легенда. Линии сетки являются продолжением делений, находящихся на осях, заголовки используются для пояснений отдельных элементов диаграммы и характера представленных на ней данных, легенда помогает идентифицировать ряды данных, представленные на диаграмме. Добавлять диаграммы можно двумя способами: внедрять их в текущий рабочий лист и добавлять отдельный лист диаграммы. В том случае, если интерес представляет сама диаграмма, то она размещается на отдельном листе. Если же нужно одновременно просматривать диаграмму и данные, на основе которых она была построена, то тогда создаётся внедрённая диаграмма.

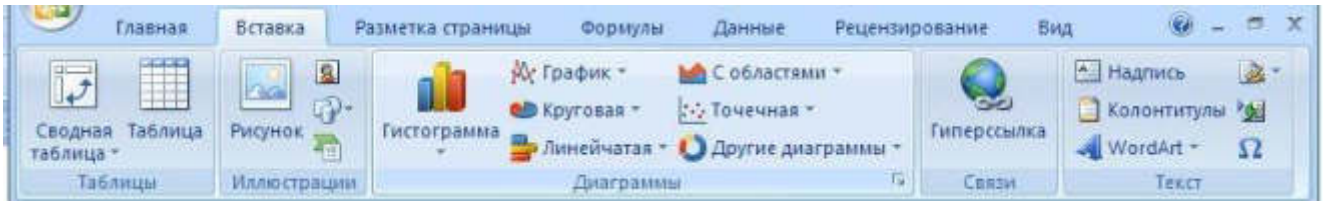


Диаграмма сохраняется и печатается вместе с рабочей книгой.

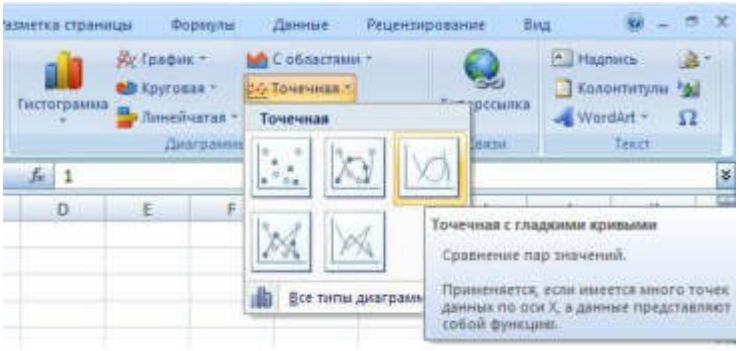
После того, как диаграмма будет сформирована, в неё можно будет внести изменения. Прежде чем выполнять какие либо действия с элементами диаграммы, выделите их, щёлкнув по ним левой кнопкой мыши. После этого вызовите контекстное меню с помощью правой кнопки мыши или воспользуйтесь соответствующими кнопками **панели инструментов Диаграмма**.

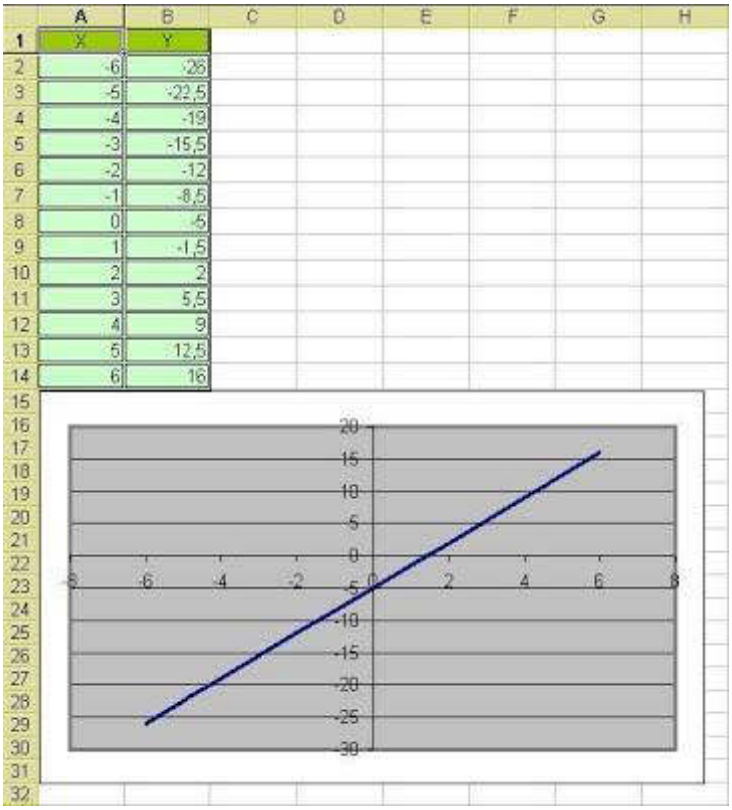
Задача: С помощью электронной таблицы построить график функции Y=3,5x–5. Где X принимает значения от –6 до 6 с шагом 1.

Технология работы:

1. Запустите табличный процессор Excel.
2. В ячейку A1 введите «X», в ячейку B1 введите «Y».
3. Выделите диапазон ячеек A1:B1 выровняйте текст в ячейках по центру.
4. В ячейку A2 введите число –6, а в ячейку A3 введите –5. Заполните с помощью маркера автозаполнения ячейки ниже до параметра 6.
5. В ячейке B2 введите формулу: =3,5*A2–5. Маркером автозаполнения распространите эту формулу до конца параметров данных.
6. Выделите всю созданную вами таблицу целиком и задайте ей внешние и внутренние границы.
7. Выделите заголовок таблицы и примените заливку внутренней области**.**
8. Выделите остальные ячейки таблицы и примените заливку внутренней области другого цвета.
9. Выделите таблицу целиком. Выберите на панели меню Вставка - **Диаграмма**, Тип: точечная, Вид: Точечная с гладкими кривыми.

10. Переместите диаграмму под таблицу.





	A	B
1	X	Y
2	-6	-26
3	-5	-22,5
4	-4	-19
5	-3	-15,5
6	-2	-12
7	-1	-8,5
8	0	-5
9	1	-1,5
10	2	2
11	3	5,5
12	4	9
13	5	12,5
14	6	16

Самостоятельная работа:

1. Постройте график функции y=sin(x)/x на отрезке [-10;10] с шагом 0,5.
2. Вывести на экран график функции: а) y=x; б) $y=x^3$; в) y=-x на отрезке [-15;15] с шагом 1.
3. Откройте файл "Города" (зайдите в папку сетевая - 9 класс-Города).

- Посчитайте стоимость разговора без скидки (столбец D) и стоимость разговора с учетом скидки (столбец F).
- Для нагладного представления постройте две круговые диаграммы. (1- диаграмма стоимости разговора без скидки; 2- диагамма стоимости разговора со скидкой).

ПОСТРОЕНИЕ ГРАФИКОВ И РИСУНКОВ СРЕДСТВАМИ MS EXCEL

1. Построение рисунка **«ЗОНТИК»**

Приведены функции, графики которых участвуют в этом изображении:

	A
1	x
2	-12
3	-11
4	-10
5	-9
6	-8
7	-7
8	-6
9	-5
10	-4
11	-3
12	-2
13	-1
14	0
15	1
16	2
17	3
18	4
19	5
20	6
21	7
22	8

$y1= -1/18x^2 + 12$, $xÎ[-12;12]$

$y2= -1/8x^2 +6$, $xÎ[-4;4]$

$y3= -1/8(x+8)^2 + 6$, $xÎ[-12; -4]$

$y4= -1/8(x-8)^2 + 6$, $xÎ[4; 12]$

$y5= 2(x+3)^2 – 9$, $xÎ[-4;0]$

$y6=1.5(x+3)^2 – 10$, $xÎ[-4;0]$

- Запустить MS EXCEL

· - В ячейке **A1** внести обозначение переменной **x**

· - Заполнить диапазон ячеек A2:A26 числами с -12 до 12.

□ □ □ □□□–□Последовательно для каждого графика функции будем вводить формулы. Для $y1= -1/8x^2 + 12$, $xÎ[-12;12]$, для
$y2= -1/8x^2 +6$, $xÎ[-4;4]$ и т.д.

Порядок выполнения действий:

1. Устанавливаем курсор в ячейку **B1** и вводим **y1**
2. В ячейку **B2** вводим формулу **=(-1/18)*A2^2 +12**
3. Нажимаем **Enter** на клавиатуре
4. Автоматически происходит подсчет значения функции.
5. Растягиваем формулу до ячейки A26
6. Аналогично в ячейку **C10** (т.к значение функции находим только на отрезке x от [-4;4]) вводим формулу для графика функции $y2= -1/8x^2 +6$. И.Т.Д.

В результате должна получиться следующая ЭТ

H19

	A	B	C	D	E	F	G	H	I	J
1	x	y1	y2	y3	y4	y5	y6			
2	-12	4		4,875						
3	-11	5,277778		5,5						
4	-10	6,444444		5,875						
5	-9	7,5		6						
6	-8	8,444444		5,875						
7	-7	9,277778		5,5						
8	-6	10		4,875						
9	-5	10,61111		4						
10	-4	11,11111	4	2,875		-7	-8,5			
11	-3	11,5	4,875			-9	-10			
12	-2	11,77778	5,5			-7	-8,5			
13	-1	11,94444	5,875			-1	-4			
14	0	12	6			9	3,5			
15	1	11,94444	5,875							
16	2	11,77778	5,5							
17	3	11,5	4,875							
18	4	11,11111	4		4					
19	5	10,61111			4,875					
20	6	10			5,5					
21	7	9,277778			5,875					
22	8	8,444444			6					

Лист1 / Лист2 / Лист3 /

После того, как все значения функций подсчитаны, можно **строить графики** этих **функций**

1. Выделяем диапазон ячеек A1:G26
2. На панели инструментов выбираем **меню Вставка → Диаграмма**
3. В окне Мастера диаграмм выберите **Точечная → Выбрать нужный вид→ Нажать Ok.**

В результате должен получиться следующий рисунок:

Отчет

Отчет должен быть оформлен в текстовом редакторе и содержать:

- наименование работы;
- цель работы;
- задание;
- последовательность выполнения работы;
- ответы на контрольные вопросы;

вывод о проделанной работе

Практическая работа № 10
Создание документа в табличном редакторе

Запустите MS Excel: Пуск/Программы/Microsoft Excel.

Задание №1. Создайте таблицу содержащую информацию о планетах солнечных систем, руководствуясь указаниями.

Солнечная система.

Планета	**Период обращения (в земных годах)**	**Расстояние (в млн.км.)**	**Диаметр (в ,тыс.км.)**	**Спутники**
Меркурий	0,241	58	4,9	0
Венера	0,615	108	12,1	0
Земля	1	150	12,8	1
Марс	1,881	288	6,8	2
Юпитер	11,86	778	142,6	16
Сатурн	29,46	1426	120,2	17

Указания:

1) В ячейке *A1* напечатайте заголовок: *Солнечная система*.
2) Расположите заголовок по центру относительно таблицы:
   - Выделите диапазон ячеек *A1 : E1*
   - Щелкните по кнопке *Объединить и поместить в центре* на панели инструментов.
3) В ячейку *A2* внесите текст: *Планета*

4) В диапазон *A3 : A8* введите название планет.
5) В ячейку *B2* внесите текст: *Период обращения (в земных годах).*
6) В ячейку *C2* внесите текст: *Расстояние (в млн. км.).*
7) В ячейку *D2* внесите текст: *Диаметр (в тыс. км.).*
8) В ячейку *E2* внесите текст: *Спутники*.
9) Выделите диапазон ячеек *B2 : D2*, выполните команду *Формат/Ячейки* на вкладке *Выравнивание* активизируйте флажок *Переносить по словам*, нажмите *ОК*.
10) Заполните диапазон *B3 : E3* числами.
11) Отформатируйте текст в таблице
    - Шрифт в заголовке – *Arial* , размер *14*, *синий цвет, жирноее начертание*.
    - Шрифт в таблице – *Times New Roman* , размер *12*, *красный цвет, начертание жирное курсив*
12) Текстовые данные выровняйте по центру.
13) Задайте рамку для таблицы:
    - Выделите таблицу (без заголовка), выполните команду *Формат/Ячейки*, вкладка *Граница*. Установите цвет – *синий*, Тип линии – *двойной* и щелкните по кнопке *Внешние*, затем выберите *Тип линии – пунктир* и щелкните по кнопке *Внутренние*, нажмите ОК.
14) Задайте заливку для второй строки таблицы: Выполните команду *Формат/Ячейки, вкладка Вид.*

Задание №2. Создайте копию своей таблицы на 2 листе и для её оформления используйте готовый стиль.

Указание:

1) Выделите таблицу и выполните команду Правка/Копировать.
2) Перейдите на 2 лист, установите курсор в ячейку А1 и выполните команду Правка/Вставить.
3) Измените высоту строк и ширину столбцов, чтобы данные помещались в ячейку.
4) Выделите таблицу (без заголовка) и выполните команду Формат/Автоформат.
5) Сохраните документ.

Ввод и редактирование данных с помощью формы в MS Excel.

Задание №3. В первую таблицу добавьте новые данные.

Указания:

1) Перейдите на лист 1.
2) Вставьте новый столбец пред столбцом Спутники и заполните его значениями (см. таблицу):
    - Вставьте столбец Спутники и выполните команду *Вставка/Столбцы*.
3) Добавьте информацию ещё о трёх планетах, используя форму:
    - Установите курсор в любую ячейку таблицы.
    - Выполните команду *Данные/Форма* (появится диалоговое окно, состоящие из трёх частей: слева располагаются названия столбцов, в середине – данные, справа – управляющие кнопки)
    - Щёлкните по кнопке *Добавить*, заполните пустые поля новыми сведениями о планете Уран (см. таблицу). Перемещаться между полями можно с помощью клавишей *TAB* или щёлкать мышью. Затем нажмите *Добавить*.
    - Аналогично введите сведения о планетах Нептун и Плутон.
    - Нажмите кнопку *Закрыть*. В результате в таблице появятся новые данные.
5) Сохраните документ.

(таблица) **Солнечная система.**

Планета	**Период обращения (в земных годах)**	**Расстояние (в млн.км.)**	**Диаметр (в тыс.км.)**	**Масса ($*10^{24}$ кг)**	**Спутники**
Меркурий	0,511	58	11,9	0,32	14
Венера	0,615	108	12,1	4,86	0
Земля	1	150	12,8	6	1
Марс	1,881	288	6,8	0,61	2
Юпитер	11,86	300	142,6	1906,98	2
Сатурн	29,46	1426	120,2	570,9	17
Уран	84,01	28,69	49	87,24	14
Нептун	164,8	4496	50,2	103,38	2

Плутон	247,7	5900	2,8	0,1	1

Сортировка данных в таблице (по одному столбцу).

Задание №4. Расположите планеты в порядке увеличения их массы:
Указания:
1) Установите курсор в любую ячейку столбца Масса
2) Щелкните по кнопке на панели инструментов *Сортировка по возрастанию.*

Задание №5. Отсортируйте записи в таблице по расстоянию, начиная с самого большого.

Задание №6. Используя встроенные функции Excel, найти минимальное расстояние, максимальный диаметр и общее число спутников.
Указания:
1) Перейдите на 2 лист.
2) В ячейку *C10* нужно ввести формулу, позволяющую определять минимальное расстояние автоматически. Для этого можно использовать функцию *МИН.* В Excel имеется множество различных функций, которые существенно упрощают расчеты.
- ❖ Установите курсор в ячейку *C10,* выполните команду ***Вставка/Функция*** или щелкните по кнопке ***Вставка функций*** на панели инструментов.
- ❖ В диалоговом окне *Мастера функций* в группе ***Категория*** выберите ***Статические,*** в группе ***Функция*** выберите *МИН* (используйте линейку прокрутки для нахождения данной функции), нажмите *ОК.*
- ❖ В следующем диалоговом окне в строке ***Число 1*** проверьте правильность указания диапазона ячеек ***C3:C8,*** при необходимости исправьте его, нажмите ***ОК.***
- ❖ В результате в ячейке C10 окажется число 58, а в строке формул: =МИН(C3:C8).

3) Аналогично в ячейку *E10* вставьте формулу, позволяющую найти максимальный диаметр.
   - Используйте функцию ***МАКС*** из категории ***Статистические.***
4) В ячейке *E10* подсчитайте общее число спутников:
   - Для вставки формулы можно пользоваться предыдущим способом, для этого используйте функцию *СУММ* из категории *Математические.* Но удобнее воспользоваться кнопкой *Автосумма* на панели инструментов *Стандартная,* выделите мышью диапазон ячеек *E3:E8* и нажмите *ENTER.*
5) В ячейках *C11,* ***D11, E11*** сделайте соответствующие подписи: Минимальное расстояние, Максимальный диаметр, Количество спутников.
6) Сохраните документ.

Отчет

Отчет должен быть оформлен в текстовом редакторе и содержать:

- наименование работы;
- цель работы;
- задание;
- последовательность выполнения работы;
- ответы на контрольные вопросы;

вывод о проделанной работе

Практическая работа №11
Работа в табличном редакторе работа с вычислениями.

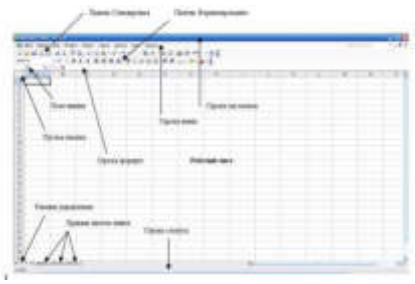
Цель выполнения задания: *создать документ согласно заданию. Закрепить навыки работы в табличном редакторе Excel.*

Необходимо знать: *практические навыки создания и редактирования электронных таблиц, ввода формул в таблицу.*

Необходимо уметь: *работать с табличным редактором Microsoft Excel. Создавать, редактировать и форматировать электронные таблицы разной сложности. Уметь производить необходимые расчеты по формулам.*

Оборудование (приборы, материалы, дидактическое обеспечение) *Инструкционные карты, мультимедийные презентации, персональные компьютеры.*

ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ



1. Запустите программу Microsoft Excel. Внимательно рассмотрите окно программы.

2. Для того чтобы ввести текст в одну из ячеек таблицы, необходимо ее выделить и сразу же (не дожидаясь появления столь необходимого нам в процессоре Word текстового курсора) "набирать".

3. Основным отличием работы электронных таблиц от текстового процессора является то, что после ввода данных в ячейку, их необходимо зафиксировать, т.е. дать понять программе, что вы закончили вводить информацию в эту конкретную ячейку.

Зафиксировать данные можно одним из способов:

- нажать клавишу {Enter};
- щелкнуть мышью по другой ячейке;
- воспользоваться кнопками управления курсором на клавиатуре (перейти к другой ячейке).

Зафиксируйте введенные вами данные. Выделите ячейку таблицы, содержащую день недели, и воспользуйтесь кнопками выравнивания абзацев . Каким образом происходит выравнивание? Сделайте вывод! .

ХОД ВЫПОЛНЕНИЯ ЗАДАНИЯ, МЕТОДИЧЕСКИЕ УКАЗАНИЯ

№ 1. На листе 1 – создайте и рассчитайте ведомость выполнения плана товарооборота киоска №5 по форме:

№	Месяц	Отчетный год			Отклонение от плана
		план	фактически	выполнение, %	
i	Mi	Pi	Fi	Vi	Oi
1	Январь	7 800,00 р.	8 500,00 р.		
2	Февраль	3 560,00 р.	2 700,00 р.		
3	Март	8 900,00 р.	7 800,00 р.		
4	Апрель	5 460,00 р.	4 590,00 р.		
5	Май	6 570,00 р.	7 650,00 р.		
6	Июнь	6 540,00 р.	5 670,00 р.		
7	Июль	4 900,00 р.	5 430,00 р.		
8	Август	7 890,00 р.	8 700,00 р.		
9	Сентябрь	6 540,00 р.	6 500,00 р.		
10	Октябрь	6 540,00 р.	6 570,00 р.		
11	Ноябрь	6 540,00 р.	6 520,00 р.		
12	Декабрь	8 900,00 р.	10 000,00 р.		

1. Заполнение столбца **Mi** можно выполнить протяжкой маркера.
2. Значения столбцов **Vi** и **Oi** вычисляются по формулам: **Vi=Fi / Pi; Oi=Fi – Pi**
3. Переименуйте **ЛИСТ1** в **Ведомость**.

№ 2. На листе 2 – Создать таблицу и отформатировать ее по образцу.

1. Столбец **«Количество дней проживания»** вычисляется с помощью функции ДЕНЬ и значений в столбцах «Дата прибытия» и «Дата убытия»

2. Столбец **«Стоимость»** вычисляется по условию: от 1 до 10 суток – 100% стоимости, от 11 до 20 суток – 80% стоимости, а более 20 – 60% общей стоимости номера за это количество дней.

4. Переименуйте **ЛИСТ2** в **Ведомость регистрации**.

ВЕДОМОСТЬ РЕГИСТРАЦИИ ПРОЖИВАЮЩИХ В ГОСТИНИЦЕ «ЗВЕЗДА»

ФИО	Номер	Стоимость номера в сутки	Дата прибытия	Дата убытия	Количество дней проживания	Стоимость
Иванов И.И.	1	10 грн	2.09.2004	2.10.2004		
Петров П.П.	2	20 грн	3.09.2004	10.09.2004		
Сидоров С.С.	4	30 грн	1.09.2004	25.09.2004		
Кошкин К.К.	8	40 грн	30.09.2004	3.10.2004		
Мышкин М.М.	13	100 грн	25.09.2004	20.10.2004		
					Общая стоимость	

№ 3. На листе 3 – Составить таблицу умножения

Для заполнения таблицы используются формулы и абсолютные ссылки.

ТАБЛИЦА УМНОЖЕНИЯ

	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	4	6	8	10	12	14	16	18
3	3	6	9	12	15	18	21	24	27
…									
9	9	18	27	36	45	54	63	72	81

Переименуйте **ЛИСТ3** в **Таблица умножения**.

№ 4. На листе 4 –Создать таблицу по образцу. Выполнить необходимые вычисления.

1. Отформатировать таблицу.

Продажа товаров для зимних видов спорта

Регион	Лыжи	Коньки	Санки	Всего
Киев	3000	7000	200	
Житомир	200	600	700	
Харьков	400	400	500	
Днепропетровск	500	3000	400	
Одесса	30	1000	300	
Симферополь	40	500	266	
Среднее				

Переименуйте **ЛИСТ4** в **Продажа товаров**.

№ 5. На листе 5 –Создать таблицу и отформатировать ее по образцу.

1. Данные в столбце Возраст вычисляются с помощью функций СЕГОДНЯ и ГОД

2. Отсортировать данные в таблице по возрасту.

3. С помощью фильтра вывести сведения только о военнообязанных сотрудниках (Пол -м, возраст от 18 до 45 лет).

Сведения о сотрудниках фирмы «Лотос»

ФИО	Должность	Дата рожд.	Пол	Возраст
Арнольдов Тарас Бульбович	Директор	01.12.45	м	
Голубков Леня Мавродиевич	Водитель	20.09.78	м	
Барабуля Сэм Джонович	Снабженец	05.08.68	м	
Симеоненко Жорж Жорикович	Гл. бух.	04.11.84	м	
Рыбак Карп Карпович	Инженер	05.05.55	м	
Графченко Дракул Дракулович	Менеджер	03.06.68	м	
Кара-Мурза Лев Филиппович	Охранник	04.03.79	м	
Сидоров Петр Иванович	Техник	20.10.85	м	
Прекрасная Василиса Ивановна	Секретарь	30.05.80	ж	
Поппинс Мэри Джоновна	Психолог	04.07.68	ж	

Переименуйте ЛИСТ5 **в Сведения о сотрудниках фирмы «Лотос».**

Контрольные вопросы:

1. Как создать электронную таблицу?
2. Как закрепить данные в ячейках?
3. Как вставить строку, столбец в таблицу?
4. Как можно произвести вычисления в табличном редакторе?
5. Как произвести автозаполнение в ячейках**?**

6. Как переименовать листы в рабочей книге?
7. Способы форматирования электронных таблиц.
8. Как производятся вычисления с помощью функций?
9. Как закрыть лист, не закрывая приложения?

Отчет

Отчет должен быть оформлен в текстовом редакторе и содержать:

- наименование работы;
- цель работы;
- задание;
- последовательность выполнения работы;
- ответы на контрольные вопросы;

вывод о проделанной работе

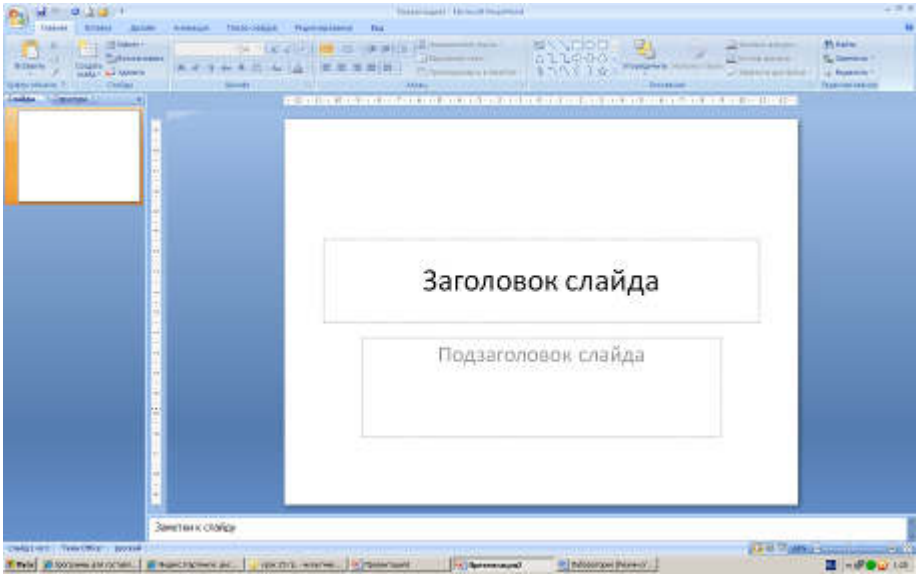
Практическая работа №12
Работа в редакторе Pover Point создание презентации

Выполнив работу, вы научитесь:

- создавать слайды различной структуры;
- задавать необходимый фон;
- добавлять в презентацию рисунки, объекты WordArt и работать с ними;
- создавать гиперссылки и управляющие кнопки;
- вставлять звук;
- настраивать анимацию.

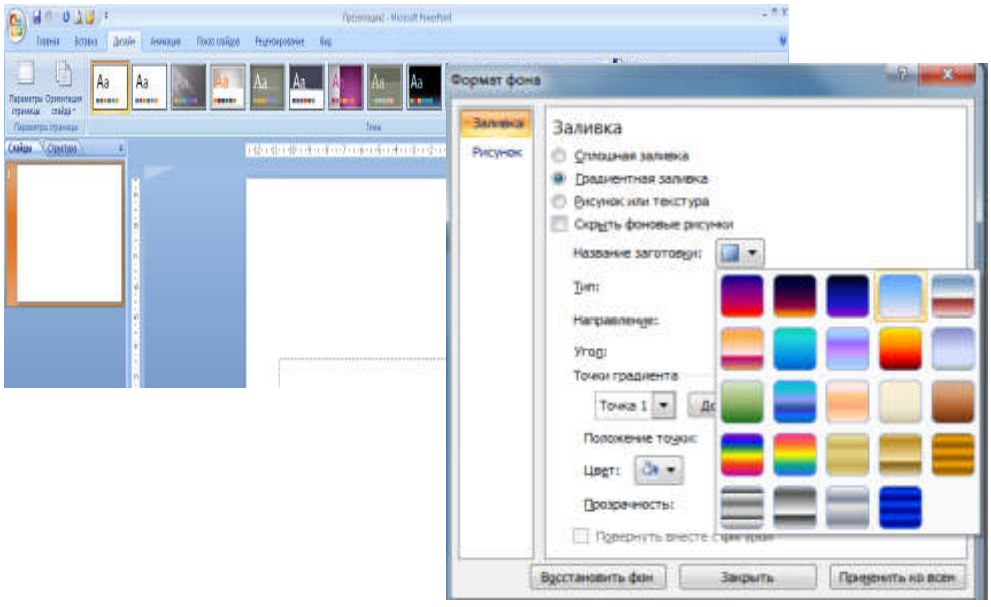
СОЗДАНИЕ СЛАЙДОВ. ВСТАВКА КАРТИНОК.

1. Запустите программу PowerPoint. Для этого выполните ***Пуск/ Все программы/ Microsoft Office/ PowerPoint.*** В открывшемся окне, предназначенном для открытия или выбора презентации, по умолчанию, создается 1 слайд – титульный лист



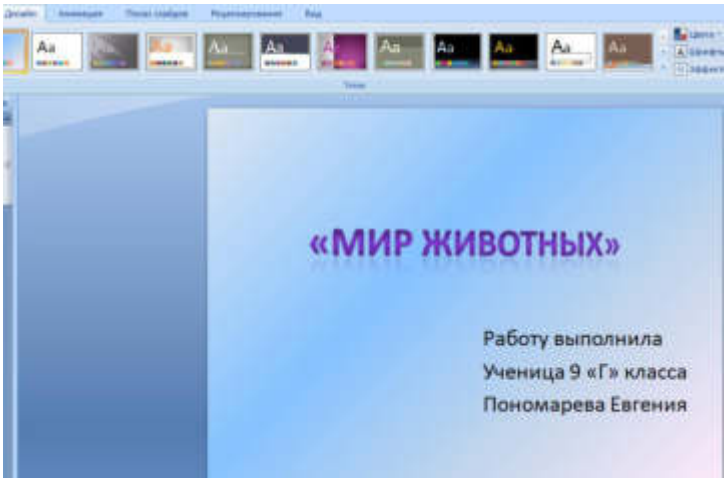
2. . Выберите цветовое оформление слайдов. Вкладка ***Дизайн/ Стили фона/ Формат фона/Градиентная заливка/ Рассвет/ Применять ко всем.***

•

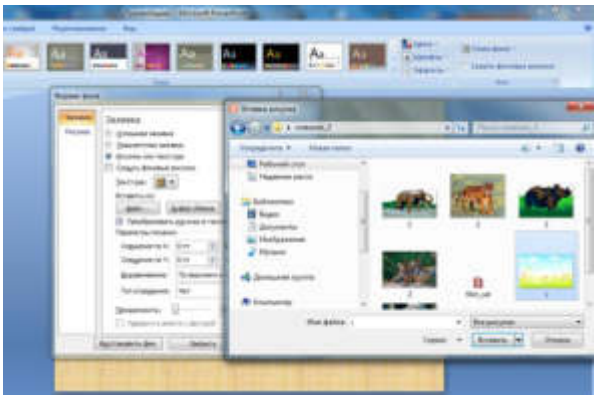


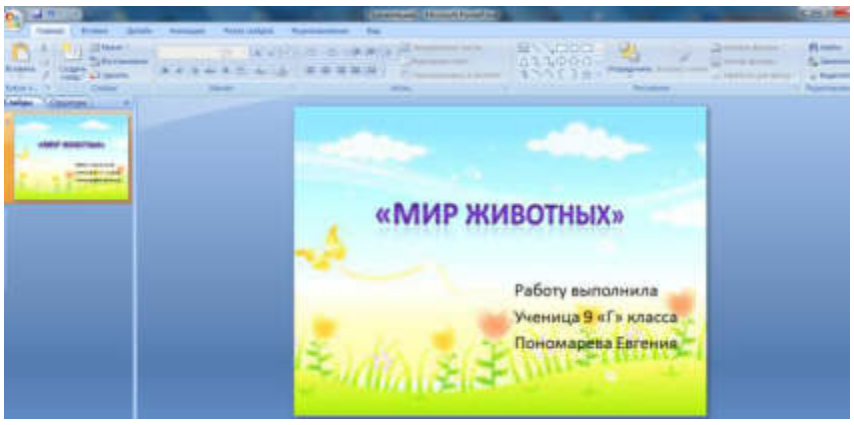
3.Оформление заголовка «Животный мир» - фиолетовым цветом: вкладка ***Вставка/ WordArt.***

Шрифт Calibri, полужирное начертание, размер шрифта 66. В подзаголовке пишем кто выполнил работу

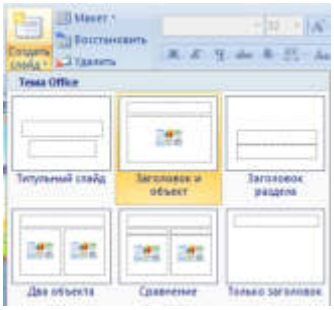


3. Заходим во Вкладку ***Дизайн/ Стили фона/ Формат фона/Рисунок или текстура*** нажимаем по слову **Файл** выбираем путь Рабочий стол/9класс/Презентации/Практическая работа№2/картинку **фон**





4. **2 слайд**: ***Главная/ Создать слайд/Заголовок и объект***



В заголовке с помощью объекта *WordArt написать Содержание. Размер шрифта 54*

3 слайд Главная/ Создать слайд/Два объекта Вставляем картинки, набираем текст

4-6 слайд аналогично 3 слайду

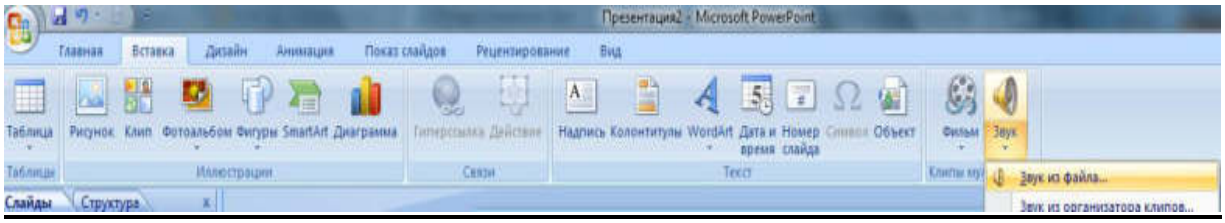
4 слайд

5 слайд

6 слайд

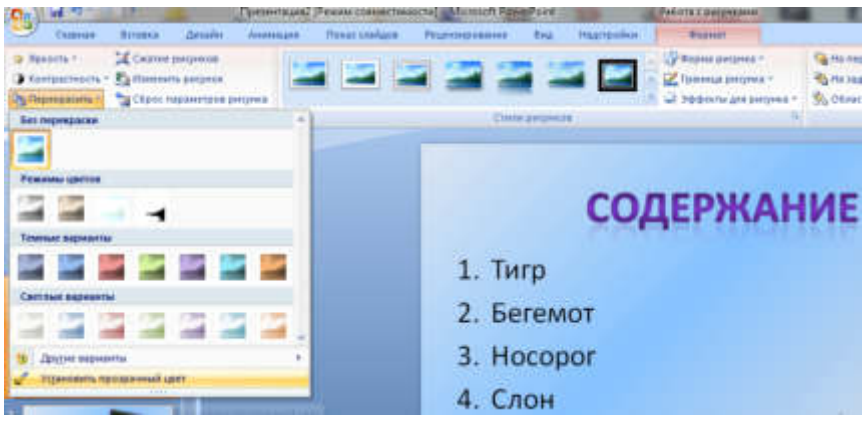
НАСТРОЙКА АНИМАЦИЙ И ВСТАВКА ЗВУКА

1 слайд Вставка звука. Выбираем вкладку **Звук/Звук из файла** выбираем путь Рабочий стол /9класс/ Презентации /Практическая работа№2звуковой файл с названием пение птиц

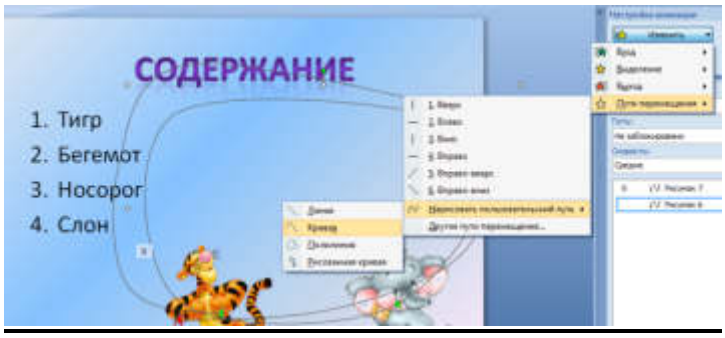


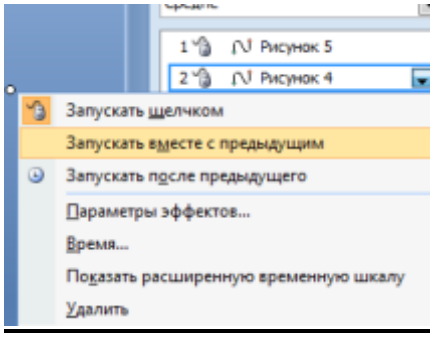
2 слайд

Вставляем картинку с тигром и слоном. **Вставка-Рисунок** выбираем путьи сам файл нажимаем ОК. Чтобы убрать белый фон у слона выделяем картинку, выбираем вкладку **Работа с рисунком - Перекрасить - Установить прозрачный цвет**



Выделяем тигра выбираем вкладку **Анимация/Настройка анимации** справа появляется кнопка **Добавить эффект/Пути перемещения/Нарисовать пользовательский путь/Кривая**. Точно также для слона



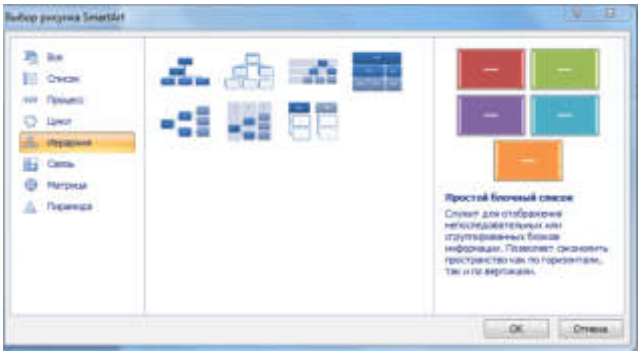


По своему усмотрению настройте Анимацию для картинок на слайдах (**Анимация-Настройка анимации**)

ВСТАВКА SMARTART

7 слайд

Выбираем вкладку **Вставка-SmartArt - Иерархия** выбираем любую схему

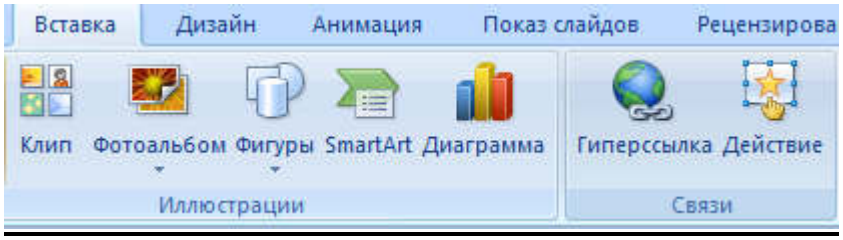


На слайде появляется схема с помощью Конструктора добавляем или удаляем фигуру. Заполняем схему

.ГИПЕРССЫЛКИ. УПРАВЛЯЮЩИЕ КНОПКИ

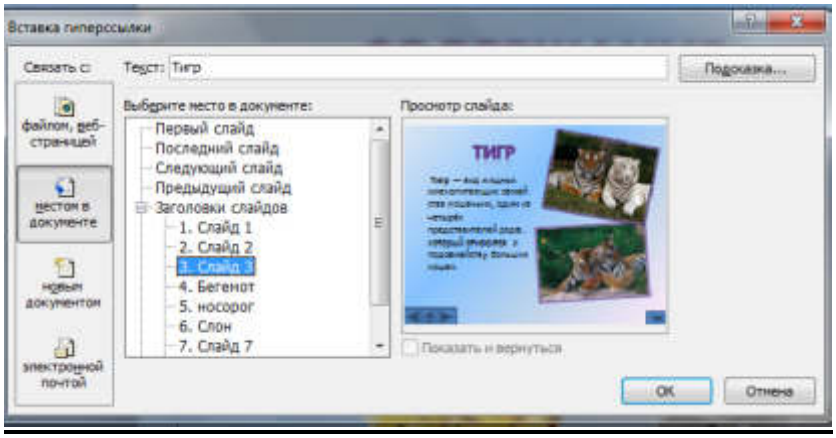
Возвращаемся к второму слайду для создания гиперссылок

1.Выделяем слово ТИГР. Выбираем вкладку **Вставка-Гиперссылка**

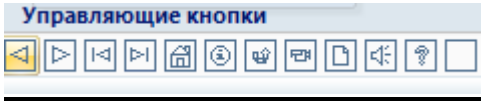


Появляется диалоговое окно выбираем Место в документе и слайд про тигров и нажимаем ОК.

Аналогично для слайдов Бегемот, Носорог, Слон.

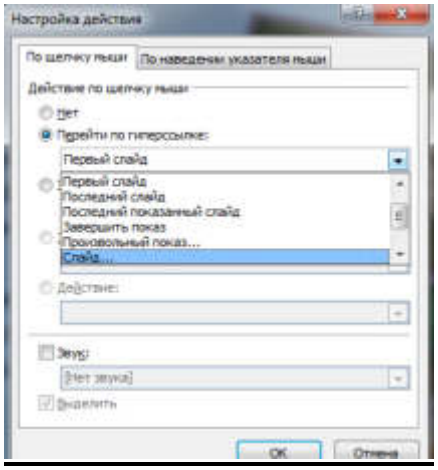


Переходим на третий слайд выбираем **Вставка - Фигуры - Управляющие кнопки.**

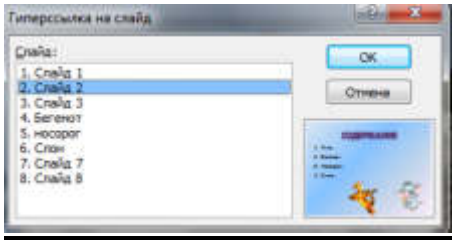


Рисуем в левом нижнем углу слайда три кнопки

Настраиваем гиперссылки. Выделяем кнопку **.** Выбираем в меню **Вставка-Гиперссылка**



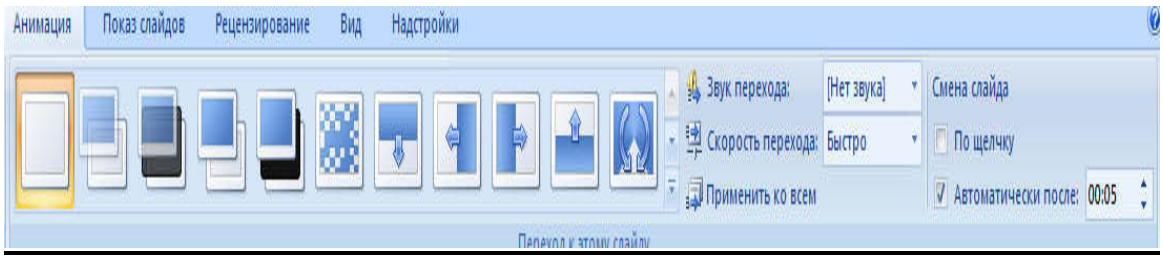
Выбираем **Слайд -Второй слайд - ОК**



Аналогично для 4-6 слайда вставка управляющих кнопок и создание гиперссылок

Создаем последний пустой слайд с помощью объекта WordArt вставляем надпись **Спасибо за внимание**!

Настраиваем презентацию в автоматический режим. Выбираем вкладку **Анимация ставим галочку Автоматически после**, устанавливаем количество секунд и нажимаем на кнопку **Применить ко всем**.



Отчет

Отчет должен быть оформлен в текстовом редакторе и содержать:

- наименование работы;
- цель работы;
- задание;
- последовательность выполнения работы;
- ответы на контрольные вопросы;

вывод о проделанной работе

Практическая работа №13
Работа в редакторе Pover Point создание анимации

Выполнив данную практическую работу, вы научитесь настраивать эффекты анимации для объектов слайда.
Рекомендации: внимательно читайте задание и чётко выполняйте предписания инструкционной карты.

Создайте анимированное изображение морского пейзажа.

Для выполнения работы используйте файл «Заготовка » на рабочем столе.

1. Выделите объект Группа птиц и выполните последовательно команды **Анимация / Настройка анимации / Добавить эффект / Пути перемещения / Нарисовать пользовательский путь / Рисованная кривая**

2. Установите указатель мыши в центр выделенного объекта Группа птиц и, удерживая нажатой левую кнопку мыши, изобразите траекторию полёта по верхней части фонового рисунка за правую границу слайда.
3. В меню **Настройка анимации** установите **Начало** *после предыдущего*, **Скорость** *очень медленно*.
4. Выделите объект Птичка и выполните последовательно команды **Добавить эффект / Пути перемещения / Нарисовать пользовательский путь / Рисованная кривая**
5. Установите указатель мыши в центр выделенного объекта Птичка и, удерживая нажатой левую кнопку мыши, изобразите траекторию полёта по верхней части фонового рисунка, спускаясь на мачту корабля.
6. В меню **Настройка анимации** установите **Начало** *с предыдущим*, **Скорость** *очень медленно*.
7. Выделите объект Птичка и выполните последовательно команды **Добавить эффект / Пути перемещения / Нарисовать пользовательский путь / Рисованная кривая.**
8. Установите указатель мыши в месте окончания траектории полёта маленькой птички на мачте и, удерживая нажатой левую кнопку мыши, изобразите траекторию полёта по верхней части фонового рисунка за правую границу слайда.
9. В меню **Настройка анимации** установите **Начало** *после предыдущего*, **Скорость** *очень медленно*.
10. Выделите объект Осьминог и выполните последовательно команды **Добавить эффект / Вход / Выцветание.**
11. В меню **Настройка анимации** установите **Начало** *с предыдущим*, **Скорость** *средне*.

12. Выделите объект Осьминог и выполните последовательно команды **Добавить эффект / Пути перемещения / Нарисовать пользовательский путь / Рисованная кривая.**
13. Установите указатель мыши в центр выделенного объекта Осьминог и, удерживая нажатой левую кнопку мыши, изобразите траекторию движения по нижней части фонового рисунка за левую границу слайда.
14. В меню **Настройка анимации** установите **Начало** *после предыдущего*, **Скорость** *очень медленно*.
15. Выделите объект Кораблик и выполните последовательно команды **Добавить эффект / Вход / Проявление с увеличением.**
16. В меню **Настройка анимации** установите **Начало** *с предыдущим*, **Скорость** *средне*.
17. Выделите объект Кораблик и выполните последовательно команды **Добавить эффект / Пути перемещения / Нарисовать пользовательский путь / Рисованная кривая.**
18. Установите указатель мыши в центр выделенного объекта Кораблик и, удерживая нажатой левую кнопку мыши, изобразите траекторию движения по нижней части фонового рисунка за пределы левой границы слайда.
19. В меню **Настройка анимации** установите **Начало** *после предыдущего*, **Скорость** *очень медленно*.
20. Сохраните работу под именем «Морской пейзаж. *Фамилия*».

Отчет

Отчет должен быть оформлен в текстовом редакторе и содержать:

- наименование работы;
- цель работы;
- задание;
- последовательность выполнения работы;
- ответы на контрольные вопросы;

вывод о проделанной работе

Практическая работа №14
Работа в редакторе Pover Point создание дизайна

Презентация **PowerPoint** – современный способ представления разной информации. Презентация состоит из серии слайдов, на которых размещается текст, графики, рисунки. Эти слайды можно распечатать на бумаге и пленке или показать их на большом экране в виде электронного слайд-шоу.

Ход работы

Задание 1. О моем друге.

1. Запустить приложение *PowerPoint*.
2. Создать новую презентацию с шаблоном оформления *Совет директоров.*
3. Установить цветовую схему с оранжевым фоном.
4. Подготовить титульный слайд. Для этого в Заголовок слайда вписать текст «О МОЕМ ДРУГЕ». Щелчком мыши выделить рамку «Подзаголовок слайда» и впишите в нем ФИО.
5. Добавить новый слайд. Для этого выполнить команду меню *Вставка → Создать слайд* или на панели инструментов **Форматирование** нажать кнопку

Создать слайд.

6. Установить для этого слайда вариант разметки *Заголовок и объект*. Для этого в *Области задач* в верхней строке нажать на треугольник. При этом раскроется список, выбрать в нем вариант *Разметка слайда* и в списке *Макеты текста* указать *Заголовок и объект*.
7. Подготовить слайды по образцу. Все необходимые картинки вы найдете в папке *Заготовки → О моем друге.*

8. Сохраните работу в личной папке под именем *PowerPoint1* и позовите учителя.

Дополнительные задания*

1. Для презентации настроить эффекты анимации. Для этого в Области задач выбрать команду *Дизайн слайда – Эффекты анимации* и в списке установить вариант *Плавающий*. Нажать кнопку *Применить ко всем слайдам*.
2. Настроить автоматическую смену слайдов. Для этого в Области задач выбрать команду *Смена слайдов* и установить вариант *Объединение по горизонтали*. Убрать флажок *по щелчку*, установить флажок *автоматически после* и установить время смены слайдов 00:01. Нажать кнопку *Применить ко всем слайдам.*
3. Просмотреть полученную презентацию. Для этого нажать клавишу **F5.**
4. Сохранить презентацию.

Задание 1. Подводный мир.

1. Запустите программу *Paint*.
2. Задайте рабочую область шириной 25 см и высотой 20 см. Примените инструмент *Заливка* для создания синего фона.
3. Изобразите в нижней части рабочей области несколько камней (инструмент *Овал*) и морских водорослей (инструмент *Карандаш*). Воспользуйтесь инструментом *Заливка* для их закраски.
4. Сохраните рисунок в личной папке под именем *Фон*.

5. Выполните команду основного меню программы *Paint*. Задайте рабочую область шириной 5 см и высотой 3 см и нарисуйте в ней рыбку. Сохраните рисунок в личной папке под именем *Рыбка*.

6. Запустите программу *PowerPoint* (Пуск → Все программы → Microsoft Office → PowerPoint).

7. Озаглавьте первый слайд: Практическая работа в *PowerPoint* № 2, ФИО.

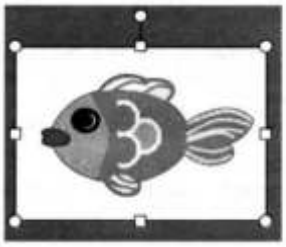
8. Создайте второй слайд, нажав на кнопку *Создать слайд.*

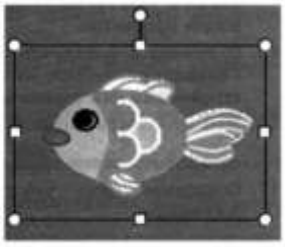
9. На вкладке *Главная* в группе *Слайды* щелкните мышью на кнопке *Макет*. Выберите пустой слайд (щелкните на макете пустого слайда).

10. Вставьте на пустой слайд рисунок *Фон*, созданный в программе *Paint*. Для этого выполните команду *Вставка → Рисунок*, найдите нужный файл в личной папке, выделите его и щелкните на кнопке *Вставить*. Фоновый рисунок появится в рабочем поле (на слайде) программы *PowerPoint.*

11. На этот же слайд поместите изображения рыбки (*файл Рыбка* в личной папке) и осьминога (файл осьминог из папки *Заготовки*).

12. Рыбка и осьминог появились на экране на белом фоне (в белых прямоугольниках). Для того, чтобы сделать фон прозрачным, выделине рыбку, щелкнув на ее изображении, - вокруг рыбки должна появиться рамка с маркерами.





13. Выполните команду Формат → цвет → установить прозрачный цвет.

14. Выполните то же действие для изображения осьминога.

15. Выделите рыбку. Перейдите на вкладку *Анимация*. В группе *Расширенная анимация* выберите Добавить анимацию. В раскрывшемся списке в группе *Пути перемещения* выберите *Пользовательский путь*. Изобразите произвольную траекторию передвижения рыбки.

16. Задайте траекторию передвижения осьминога.

17. Для просмотра анимации воспользуйтесь кнопкой *Просмотр* на вкладке *Анимация*.

18. Сохраните работу в личной папке под именем *PowerPoint2* и позовите учителя.

Дополнительные задания*

1. После завершения просмотра выделите один из объектов анимации и изучите возможности настройки анимации в группе *Время показа слайдов*. По своему усмотрению внесите изменения в параметры анимации. Понаблюдайте за изменениями в ходе просмотра.

2. С помощью инструментов копирования и вставки (вкладка Главная) разместите на слайде несколько копий рыбки и настройте для них параметры анимации. Для просмотра готовой анимации воспользуйтесь функциональной клавишей F5.

3. Сохраните работу

.

Задание. Анимация на свободную тему.

1. Самостоятельно придумайте сюжет для анимации. Дайте ему название и подробно опишите планируемую последовательность событий.

2. Реализуйте свой проект средствами графического редактора и редактора презентаций. Подготовьтесь представить свою работу товарищам по классу.

Задание. Времена года

1. Запустить приложение PowerPoint.

2. Выбрать по собственному желанию шаблон оформления и цветовую схему.
3. Подготовить титульный слайд с названием презентации и ФИО.
4. Добавить 4 слайда с названием времени года в заголовке. Для этого выполнить команду меню *Вставка → Создать слайд* или на панели инструментов **Форматирование** нажать кнопку *Создать слайд*. Для этих слайдов выбрать вариант разметки *Заголовок, текст и объект*.

5. Отключить список в текстовом окне. Для этого на панели инструментов **Форматирование** отжать кнопку **Маркеры.**

6. В текст слайдов вписать стихи:

Снег и снег, и ель в снегу –
В белых пачках – балерины
Снег зажегся на лугу
Ювелирною витриной

Осеев Н.Н.

Ох, лето красное!
Любил бы я тебя,
Когда б не зной, да пыль,
Да комары, да мухи!

Пушкин А.С.

Весна, весна! Как воздух чист!
Как ясен небосклон!
Своей лазурию живой
Слепит мне очи он.

Баратынский Е.А.

Кроет уж лист золотой
Влажную землю в лесу…
Смело топчу я ногой
Вешнюю леса красу.

Майков А.Н.

7. Вставить фотографии с соответствующей порой года. Для этого щелкнуть по миниатюре *добавит рисунок*, в открывшемся окне выбрать необходимый файл.
8. Настроить анимацию заголовка. Для этого выделить заголовок слайда и выполнить команду меню *Показ слайда → Настройка анимации.* Выбрать из списка. *Добавить эффект* вариант *Вход → Другие эффекты… → Общие: вылет* и установить параметр *Направление – Сверху.*
9. Для текста стихотворения выбрать эффект анимации *Вход → Другие эффекты…→Общие: Цветная пишущая машинка.* Установить параметр *Начало: после предыдущего*.
10. Для фотографии выбрать эффект анимации *Вход → Другие эффекты… → Общие: Круговой симметричный.* Установить параметр *Скорость: очень быстро.*
11. Настроить смену слайдов. Для этого выполнить команду *Показ слайдов → Смена слайдов* и в Области задач выбрать вариант *Открывание вниз*, убрать флажок *по щелчку*, установить флажок *автоматически после*. Установить время смены слайдов – 3с. Нажать кнопку *Применить ко всем слайдам*.
12. Подключить к просмотру музыку – Чайковский «Времена года». Для этого перейти на первый слайд, выполнить команду меню *Вставка → Фильмы и звук → Звук из файла…* и указать на файл с мелодией. В появившемся диалоговом окне выбрать вариант *Автоматически.*
13. Реализовать звучание музыки на протяжении всего просмотра. Для этого в Области задач выбрать пункт *Настройка анимации*, правой кнопкой мыши вызвать контекстное меню к звуку, выбрать пункт **Параметры эффектов**… и в открывшемся диалоговом окне установить вариант *Закончить после 5-го слайда,* на вкладке **Параметры звука** установить флажок *скрывать значок звука во время показа.*
14. Просмотреть полученную презентацию. Для этого нажать клавишу **F5.** Сохранить презентацию в личной папке под именем *PowerPoint4*.

Задание

1. Запустить приложение PowerPoint.
2. Создать новую презентацию. Установить произвольный шаблон дизайна слайдов.
3. Оформить титульный слайд по образцу, поместив на него картинку из файла *Винни1.jpg*

4. Добавить к слайдам звуковое сопровождение. Для этого выполнить команду меню *Вставка → Фильм и звук → Звук из файла…* и выбрать файл с песней из мультфильма.

5. Реализовать звучание песни на протяжении 3 слайдов. Для этого в Области задач выбрать пункт *Настройка анимации*, выполнить двойной щелчок на эффекте и в открывшемся окне установить вариант *Закончить после 3-го слайда.*

6. Создать новый слайд с разметкой *Заголовок, текст, объект*. Для этого на панели инструментов *Форматирование* нажать кнопку *Создать слайд.* В Области задач выбрать соответствующую разметку. Ввести текст заголовка и текст слайда. Вставить две картинки из файлов *Винни2.jpg, Винни3.jpg*

7. Создать новый слайд с разметкой *Заголовок и текст.* В области текста поместить список создателей мультфильма. Растянуть текстовую рамку вниз за границу слайда. Для удобства можно уменьшить масштаб отображения слайда. Настроить для текста эффект анимации Вход → *Другие эффекты…→ Сложные: Титры.* В Области задач выполнить двойной щелчок по эффекту, в открывшемся диалоговом окне в дополнительных параметрах в списке. *После анимации* выбрать вариант. *После анимации скрыть*.

Текст для слайда:

Режиссеры
Геннадий Сокольский
Федор Хитрук
Сценаристы
Федор Хитрук
Борис Заходер
Художники-постановщики
Владимир Зуйков
Эдуард Назаров
Аниматоры
Галина Баринова
Эльвира Маслова
Виолетта Колесникова
Анатолий Солин
Елена Малашенкова
Юрий Бутырин
Мария Мотрук
Ольга Орлова
Геннадий Сокольский
Наталия Богомолова
Оператор
Михаил Друян
Директор
Любовь Бутырина
Композитор
Моисей Вайнберг
Звукооператор
Георгий Мартынюк
Редактор
Раиса Фричинская
Роли озвучивали:
Евгений леонов (Винни-Пух)

Зинаида Нарышкина (Сова)
Ия Саввина (Пятачок)
Эраст Гарин (Ослик Иа)
Владимир Осенев

8. Создать четвертый слайд. Поместить на слайд файл с мультфильмом. Для этого выполнить команду меню *Вставка → Фильм и звук → Фильм из файла…* В области задач выполнить двойной щелчок по эффекту. В открывшемся диалоговом окне на вкладке *Параметры фильма* установить флажок *разворачивать во весь экран.*

9. Настроить смену слайдов. Для этого выполнить команду *Показ слайдов → Смена слайдов* и в Области задач выбрать вариант *Открывание вниз*, убрать флажок *по щелчку*, установить флажок *автоматически после*. Установить время смены слайдов – 3с. Нажать кнопку *Применить ко всем слайдам*.
10. Просмотреть полученную презентацию.
11. Сохранить презентацию в личной папке с именем *PowerPoint5*.

Задание 1. Презентация «Часы»

1. Запустите редактор презентаций *PowerPoint*. Выполните титульный слайд. Создайте пустой слайд.
2. С помощью готовых фигур (вкладка *Вставка*) на пустом слайде изобразите циферблат с двумя стрелками. Это может выглядеть примерно так, как показано на рисунке. Чтобы циферблат был симметричным, сначала изобразите в центре экрана пересекающиеся горизонтальную и вертикальную прямые – оси симметрии. Пытайтесь рисовать рационально – копируйте повторяющиеся элементы, при необходимости изменяете их при помощи команд группы *Повернуть*.

3. Скопируйте слайд с часами в буфер обмена. Для этого:
    a. На вкладке *Слайды* левой панели щелкните на слайде с часами правой кнопкой мыши;
    b. Выберите команду *Копировать* в открывшемся меню.
4. Вставьте в презентацию еще 4 копии этого слайда (кнопка *Вставить* на вкладке *Главная*). Всего должно получиться 5 слайдов.
5. Внесите изменения в положение стрелок так, чтобы на них последовательно отмечалось: 12.00, 12.15, 12.30, 12.45, 13.00.
6. Перейдите на вкладку *анимация*. В группе *время показа* слайдов:
    a. Установите минимальную длительность перехода между слайдами (00,01);
    b. Выберите режим смены слайдов *После* и установите время 1 сек (00:01,00);
    c. Щелкните на кнопке *Применить ко всем* слайдам.
7. Запустите показ презентации функциональной клавишей.
8. Сохраните работу в личной папке под именем Часы.

Дополнительное задание*

1. Вставьте в презентацию дополнительные слайды так, чтобы на них отмечалось время 12.05, 12.10 и т.д.

Задание 1.Времена года.

1. Откройте презентацию из практической работы № 4 *«Времена года»*.
2. Вставьте перед вторым слайдом новый слайд. В поле *Заголовок слайда* введите текст «Времена года». Выделите введенный заголовок и перейдите на вкладку *Формат*. С помощью инструментов группы *WordArt* придайте заголовку красочный вид.
3. Перечислите на этом же слайде в поле текст слайда все времена года. На вкладке *Главная* при помощи инструментов группы *Шрифт* придайте списку красочный вид.

4. Необходимо связать слово «зима» с соответствующим слайдом презентации, иначе говоря, создать гиперссылку. Для этого, выделите слово Зима и выполните команду *Вставка → Гиперссылка*.

5. В окне *Вставка* гиперссылки выберите *Связать с: ─ местом в документе*. Выберите *место в документе – слайд Зима.* Подтвердите свой выбор щелчком на кнопке *Ок*. Обратите внимание, как изменилось слово Зима на Слайде № 2.

6. Аналогичным способом создайте гиперссылки от слов «Весна», «Ленто» и «Осень» к соответствующим слайдам презентации.

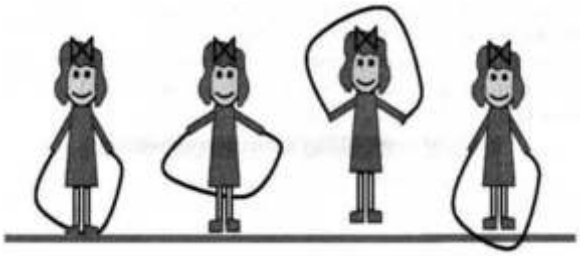
7. Создайте на каждом слайде времен года управляющую кнопку, обеспечивающую переход к оглавлению (к слайду № 2):
   a. Перейдите на вкладку *Вставка*;
   b. В группе *Иллюстрации* выберите инструмент *Фигуры*;
   c. В раскрывшемся списке среди управляющих кнопок выберите кнопку *В начало*;
   d. Вставьте на слайд выбранную кнопку подходящего размера;
   e. В окне *Настройка действия* установите переход по гиперссылке на слайд № 2 и подтвердите свой выбор щелчком по кнопке *Ок.*
8. Запустите презентацию. Просмотрите слайды презентации с использованием гиперссылок.
9. Сохраните изменения, выполненные в презентации «Времена года».

Задание 1. Скакалочка

1. Запустите редактор презентаций *PowerPoint*. Выполните титульный слайд. Создайте пустой слайд.

2. С помощью готовых фигур на пустом слайде изобразите девочку, прыгающую через скакалку. Это может выглядеть примерно так, как показано на рисунке.

3. Добавьте в презентацию еще три копии слайда с изображением девочки. Внесите в изображение изменения, соответствующие основным фазам прыжка через скакалку. Это может выглядеть примерно следующим образом:



4. Настройте циклическую (повторяющуюся) демонстрацию слайдов. Для этого:
   a. На вкладке Показ слайдов в группе Настройка щелкните на кнопке *Настройка демонстрации*;
   b. Установите флажок *Непрерывный цикл* до нажатия клавиши *Esc*;
   c. Задайте смену слайдов *Вручную*;
   d. *Ок*.
5. Запустите презентацию.
6. Сохраните презентацию в личной папке под именем Скакалочка.

Дополнительное задание *

1. Запустите презентацию в режиме автоматической смены слайдов.
2. Дополните слайды по своему усмотрению. Например, можно использовать фоновый рисунок, созданный в графическом редакторе.

Отчет

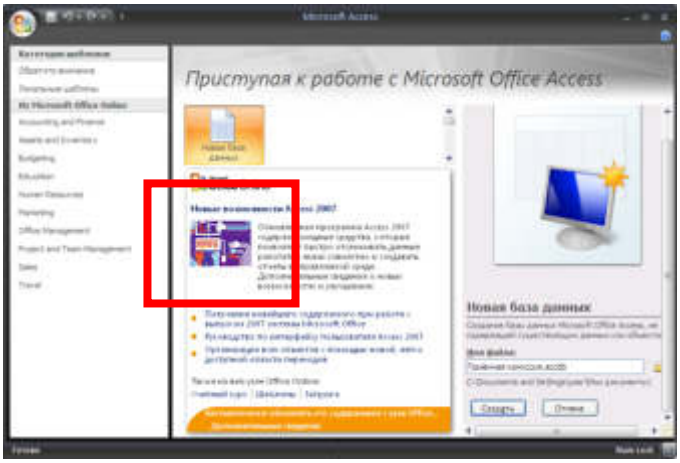
Отчет должен быть оформлен в текстовом редакторе и содержать:

- наименование работы;
- цель работы;
- задание;
- последовательность выполнения работы;
- ответы на контрольные вопросы;

вывод о проделанной работе

Практическая работа №15
Работа в редакторе баз данных создание БД учащихся группы
Ход работы

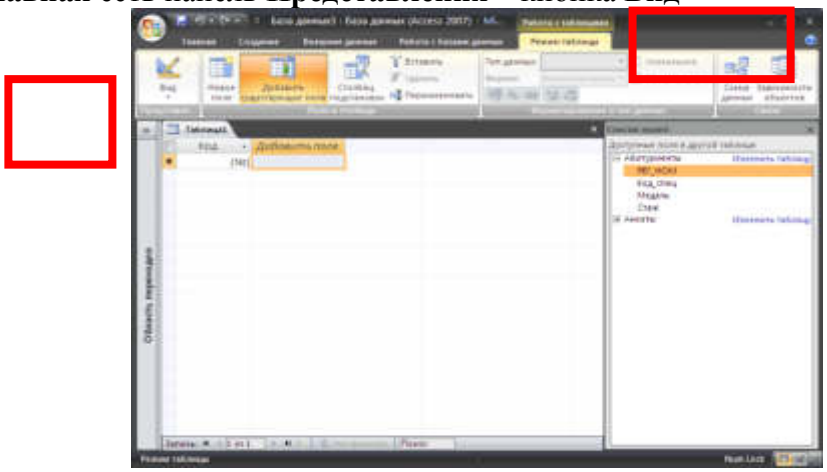
1. С помощью программы MS Access 2007 ***создайте базу данных*** «**Приемная комиссия**» в (***Пуск – Все программы – Microsoft Office – Microsoft Access 2007***) в вашей папке.



В программе MS Access 2007 очень удобно работать при создании баз данных. Помогает тот факт, что при наведении мыши на любую кнопку среды показывается ее назначение! При создании базы данных первая таблица создается автоматически. Когда создана новая таблица на главной панели появляется вкладка **Работа с таблицами**.

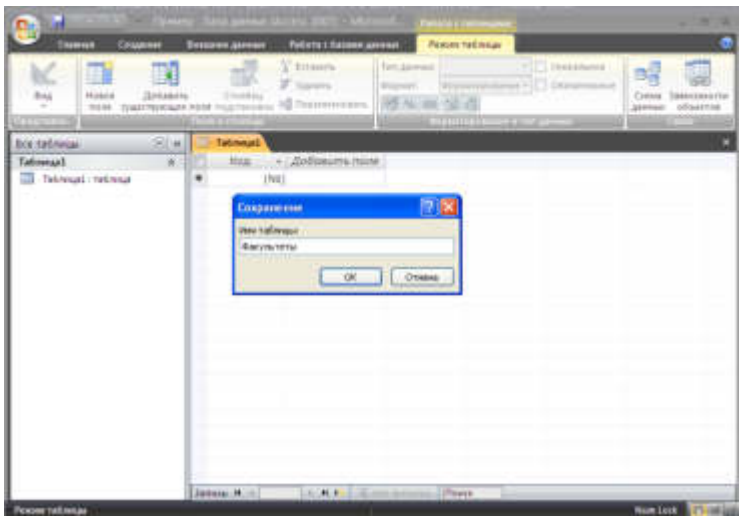
Для каждого объекта Microsoft Access существуют режимы работы.

Для перехода с одного режима работы на другой в Microsoft Access 2007 на вкладке **Главная** есть панель **Представления** – кнопка **Вид**

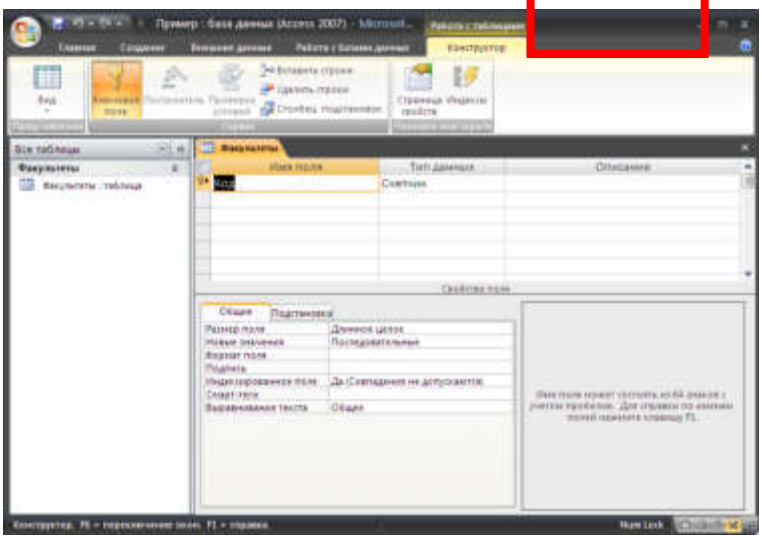


2. Создайте с помощью режима конструктора 2 таблицы «**Факультеты**» и «**Специальности**».

1) Чтобы «Таблицу 1» переименовать в таблицу «Факультеты» и перейти в режим конструктора ***нужно нажать*** **кнопку Вид**, программа попросит сохранить таблицу и ***ввести имя таблицы***



После нажатия **кнопки ОК** появится окно конструктора таблицы «Факультеты»:

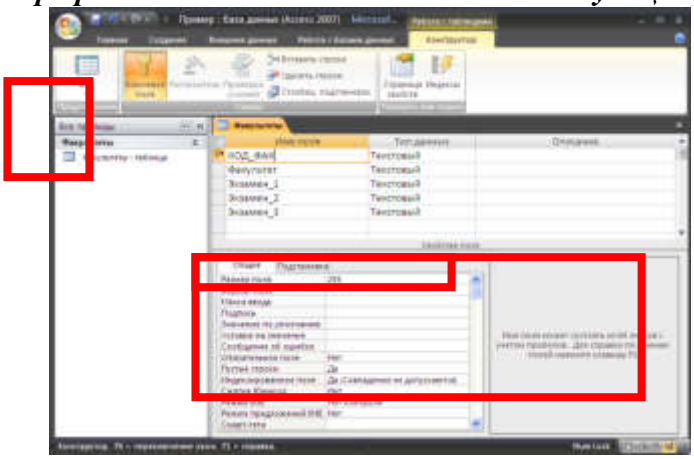


Для создания таблицы необходимо, как минимум, заполнить графы «Имя поля» и «Тип данных».

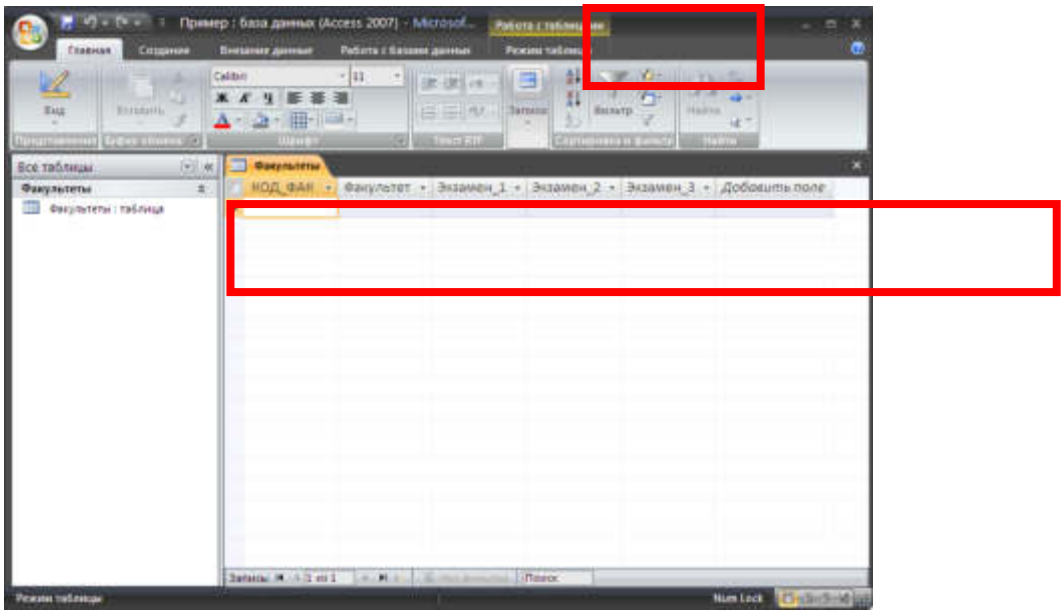
Графа «Описание» не является обязательной и предназначена для текста подсказки, который выводится в нижней строке экрана во время работы с таблицей.

Для того чтобы задать тип поля, надо щелкнуть в соответствующей ячейке в столбце «Тип данных». Это приведет к появлению символа списка справа в выбранной ячейке. Надо открыть его и выбрать нужный тип.

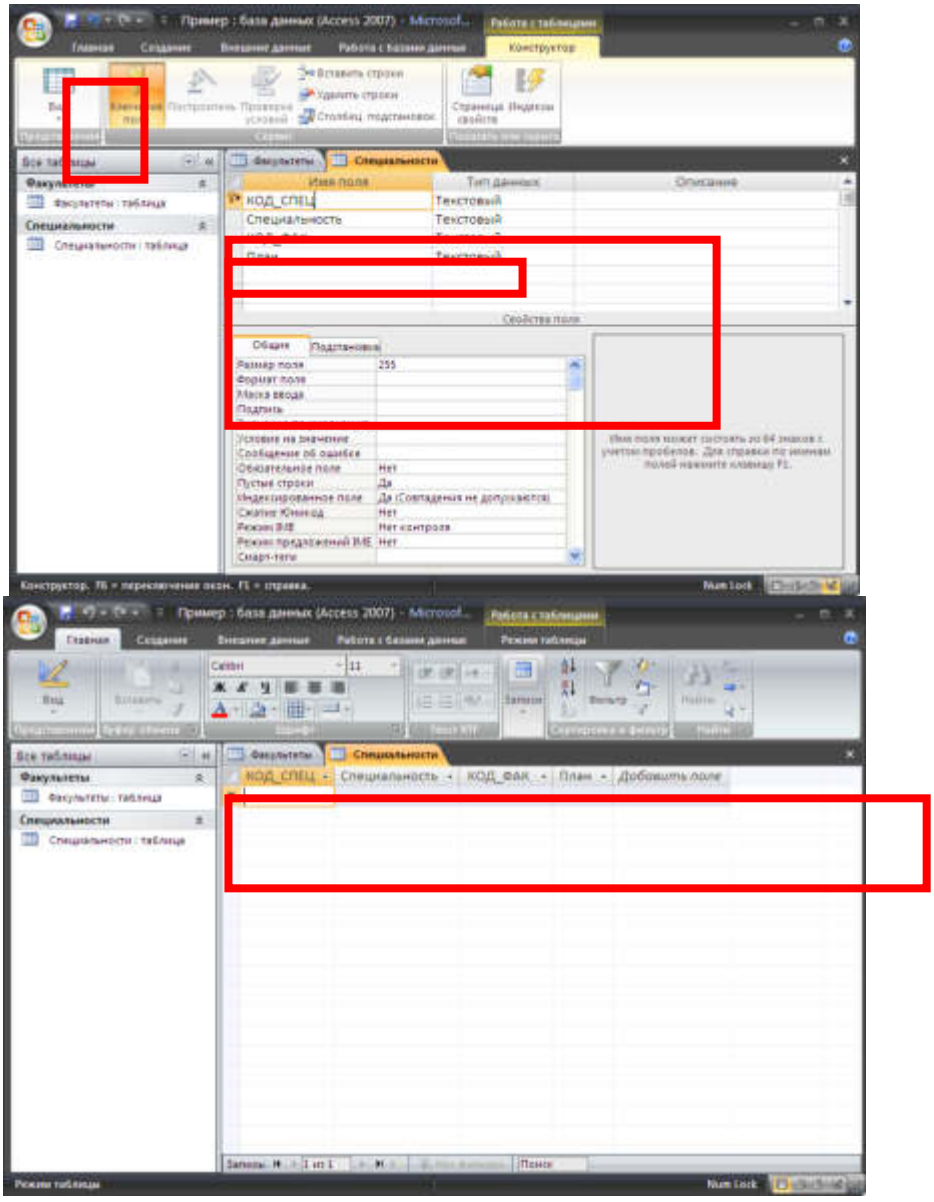
Заполните графы «Имя поля» и «Тип данных» следующим образом:



После этого вернитесь в режим таблицы (**кнопка Вид**), таблица «Факультеты» будет выглядеть так:



2) ***Создайте таблицу «Специальности»*** **(вкладка Создание - Таблица),** ***проделав те же шаги что и при создании таблицы «Факультеты».***



3. ***Заполните таблицы следующим образом:***

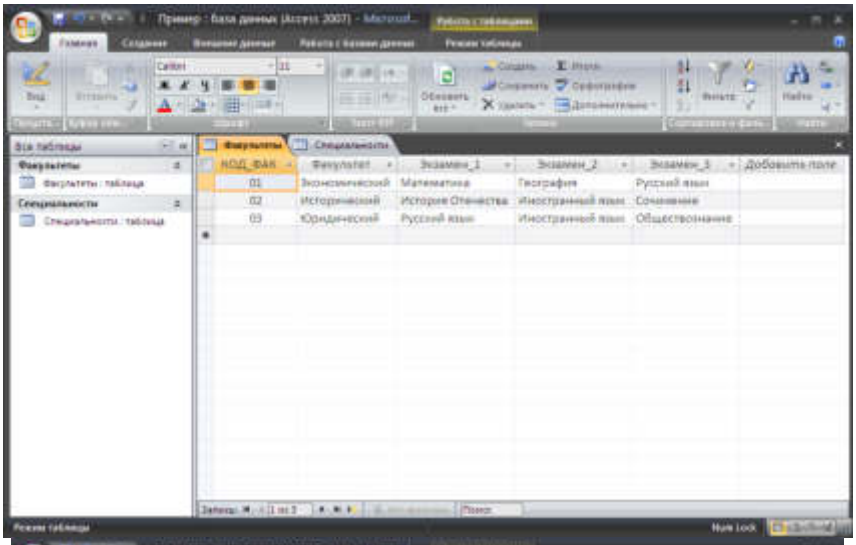
Таблица «Факультеты»:

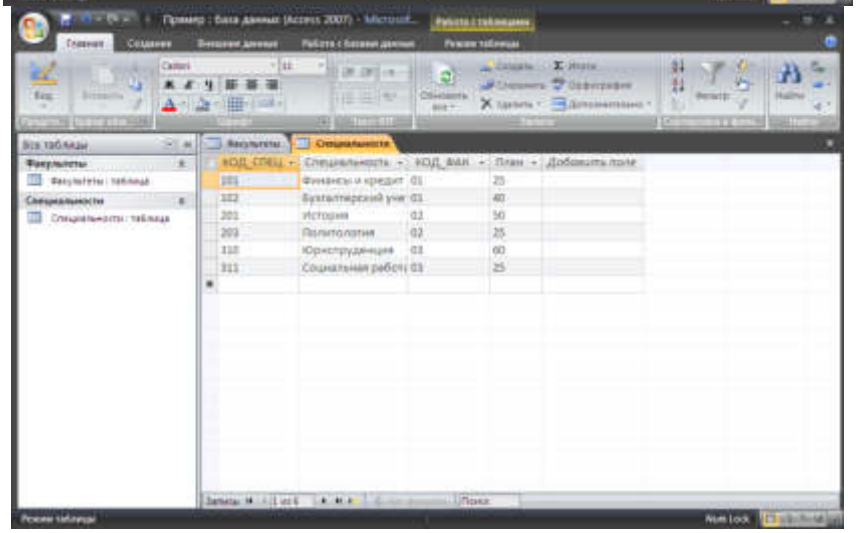
КОД_ФАК	Факультет	Экзамен_1	Экзамен_2	Экзамен_3
01	Экономический	Математика	География	Русский язык
02	Исторический	История Отечества	Иностранный язык	Сочинение
03	Юридический	Русский язык	Иностранный язык	Обществознание

КОД_СПЕЦ	Специальность	КОД_ФАК	План
101	Финансы и кредит	01	25
102	Бухгалтерский учет	01	40
201	История	02	50
203	Политология	02	25
310	Юриспруденция	03	60
311	Социальная работа	03	25

Таблица «Специальности»:

После заполнения таблицы должны выглядеть так:

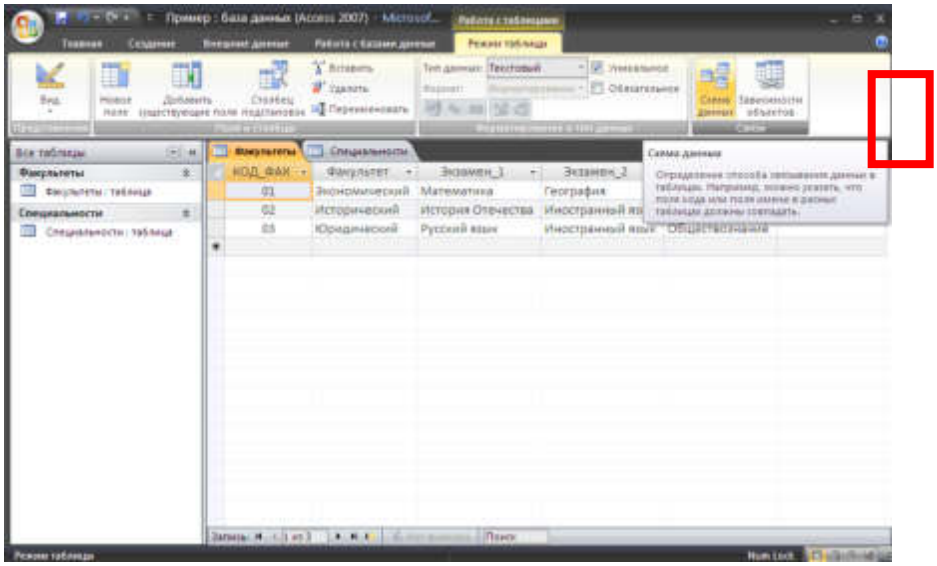




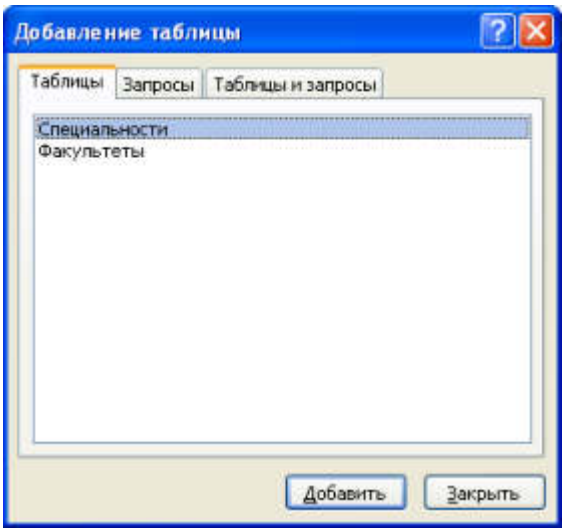
Последнее действие на подготовительном этапе – построение схемы.

4. Создайте схему данных для таблиц «Факультеты» и «Специальность»

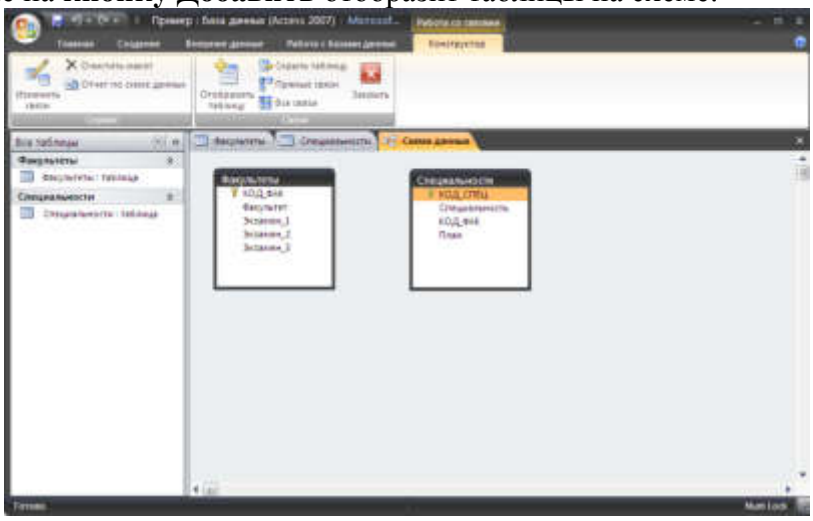
Для этого в режиме таблицы на вкладке **Работа с таблицами** ***нажмите кнопку*** **Схема данных.**



Появится окно:



Нажатие на **кнопку Добавить** отобразит таблицы на схеме:

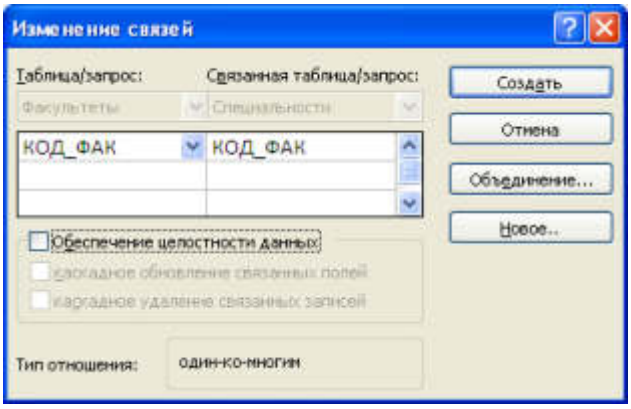


Для установки связи между двумя таблицами:

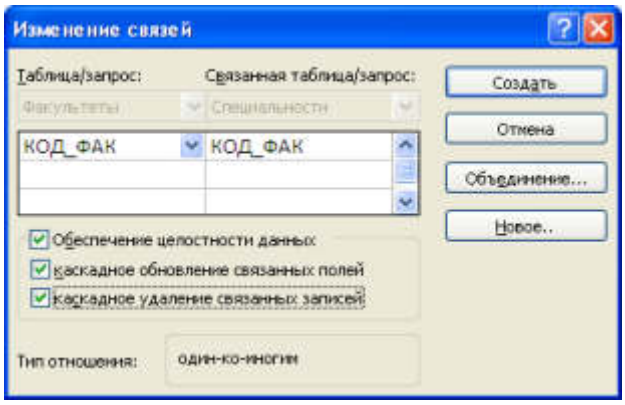
1. ***Сохраните и закройте созданные таблицы.***
2. ***Выберете в таблице «Факультеты» поле КОД_ФАК, для которого хотите создать связь.***

3. Установив курсор мыши на этом поле, нажмите левую кнопку и переместите это поле в таблицу «Специальности» к полю КОД_ФАК.

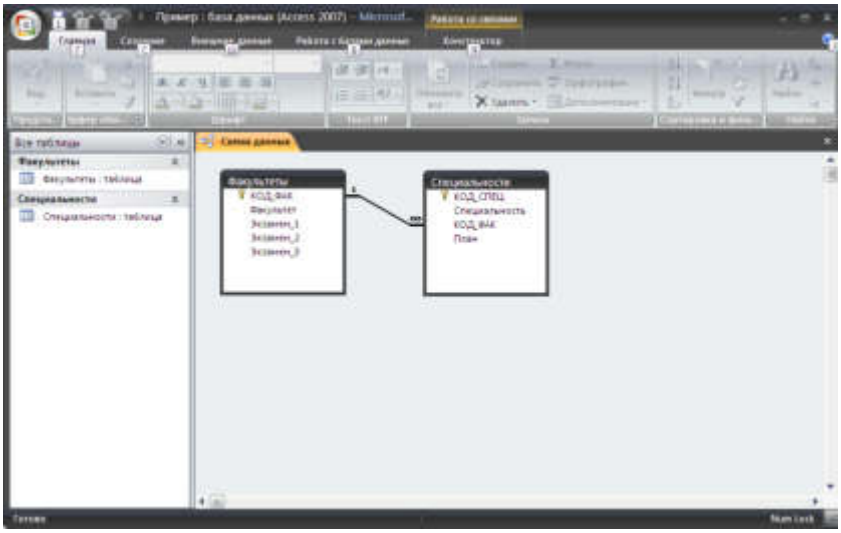
4. Появится окно



5. Последовательно установите флажки «Обеспечить целостность данных», «Каскадное обновление связанных полей» и «Каскадное удаление связанных записей»



Тип связи «один ко многим» будет выбрана автоматически. ***Нажмите кнопку Создать***. Схема должна выглядеть так:



В тех случаях, когда пользователей не удовлетворяет представление данных в виде таблиц, они могут построить форму, которая позволит расположить данные на экране так, как это требуется. Кроме того, форма может использоваться для поиска данных. Если пользовательская задача требует документов определенного вида, то без форм обойтись практически невозможно. Форма позволяет вывести на экран больше данных, нежели таблица. Это достигается при помощи объектов, обеспечивающих "трехмерность" данных, например, раскрывающихся списков.

Т.о. формы используют для ввода, просмотра, редактирования, удаления данных; для создания интерфейса приложения; для поиска и выборки данных, удовлетворяющих определенным условиям, заданным пользователем.

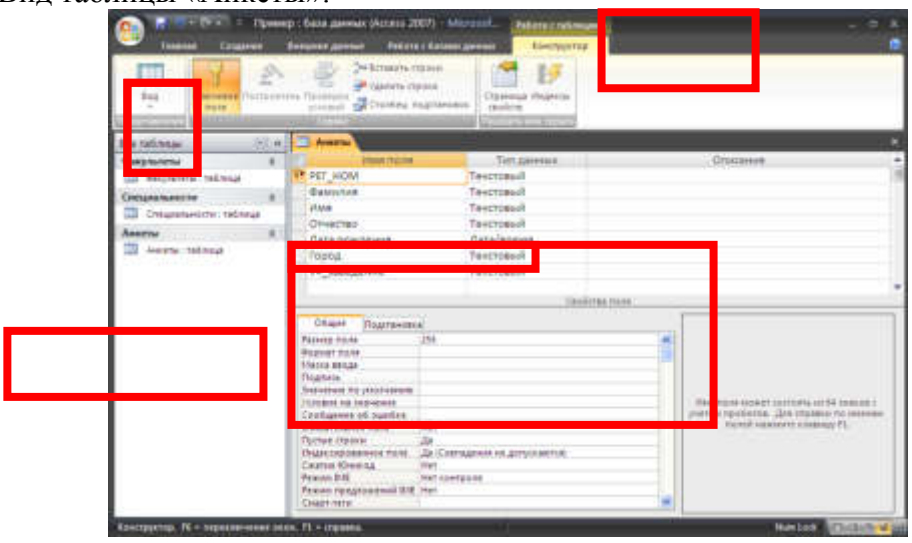
Способы создания форм.

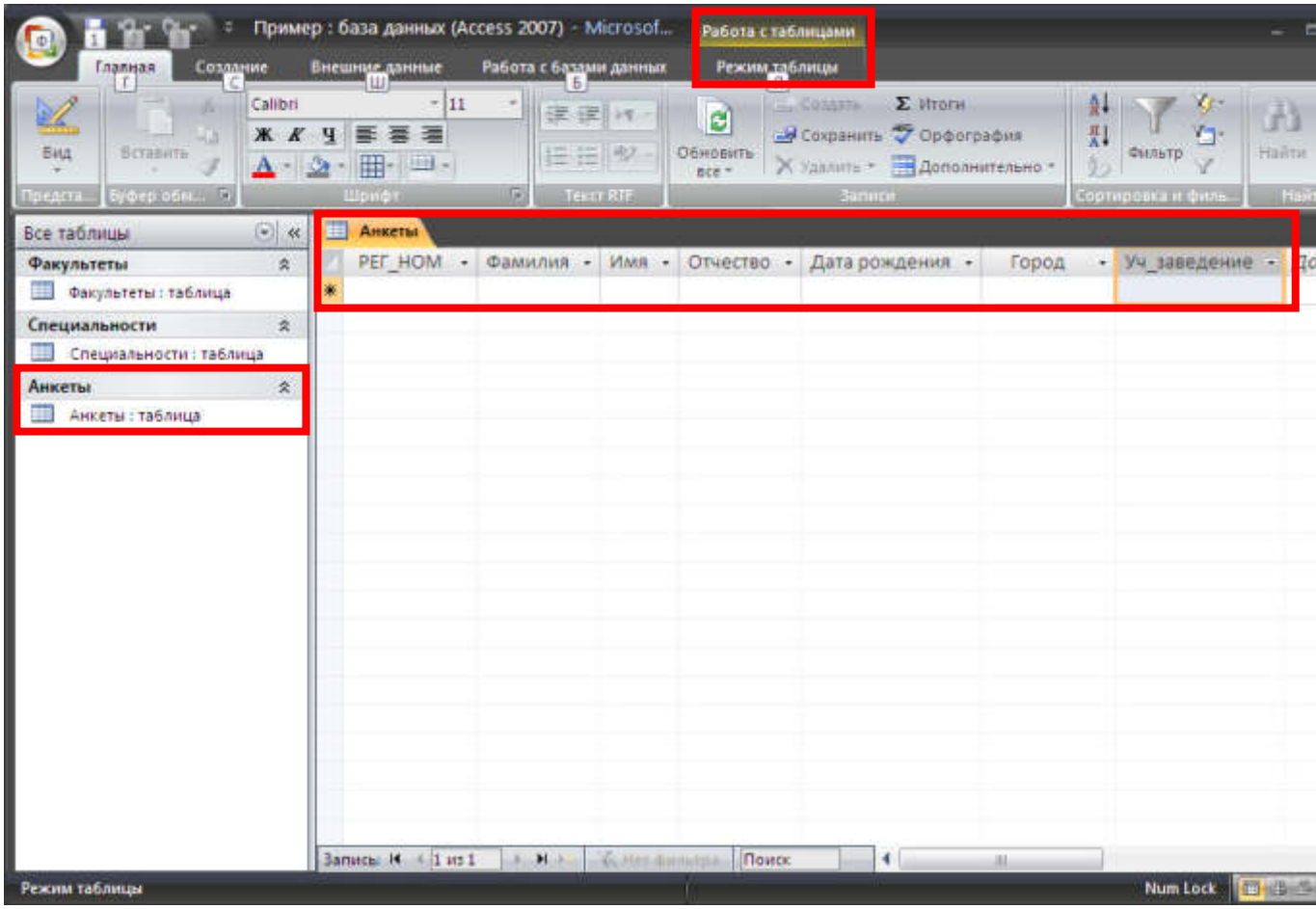
- Конструктор форм. Создание новой пустой формы в режиме конструктора.
- Мастер форм. Создание формы с расширенными инструментами.
- Пустая форма. Создание формы для ввода данных по одной записи за раз.
- Разделенная форма. В верхней части отображается таблица, а в нижней – форма для ввода данных в запись, выделенную в таблице.
- Несколько элементов. Создание формы, в которой отображается сразу несколько записей.
- Режим таблицы. Она выводит поля в виде строк и столбцов, подобно таблицам.
- Сводная диаграмма. Этот вид форм включает графические представления. Для ее создания нужно выбрать таблицу, содержащую числовые значения, которые можно представить в графическом виде.
- Сводная таблица.

Ход работы

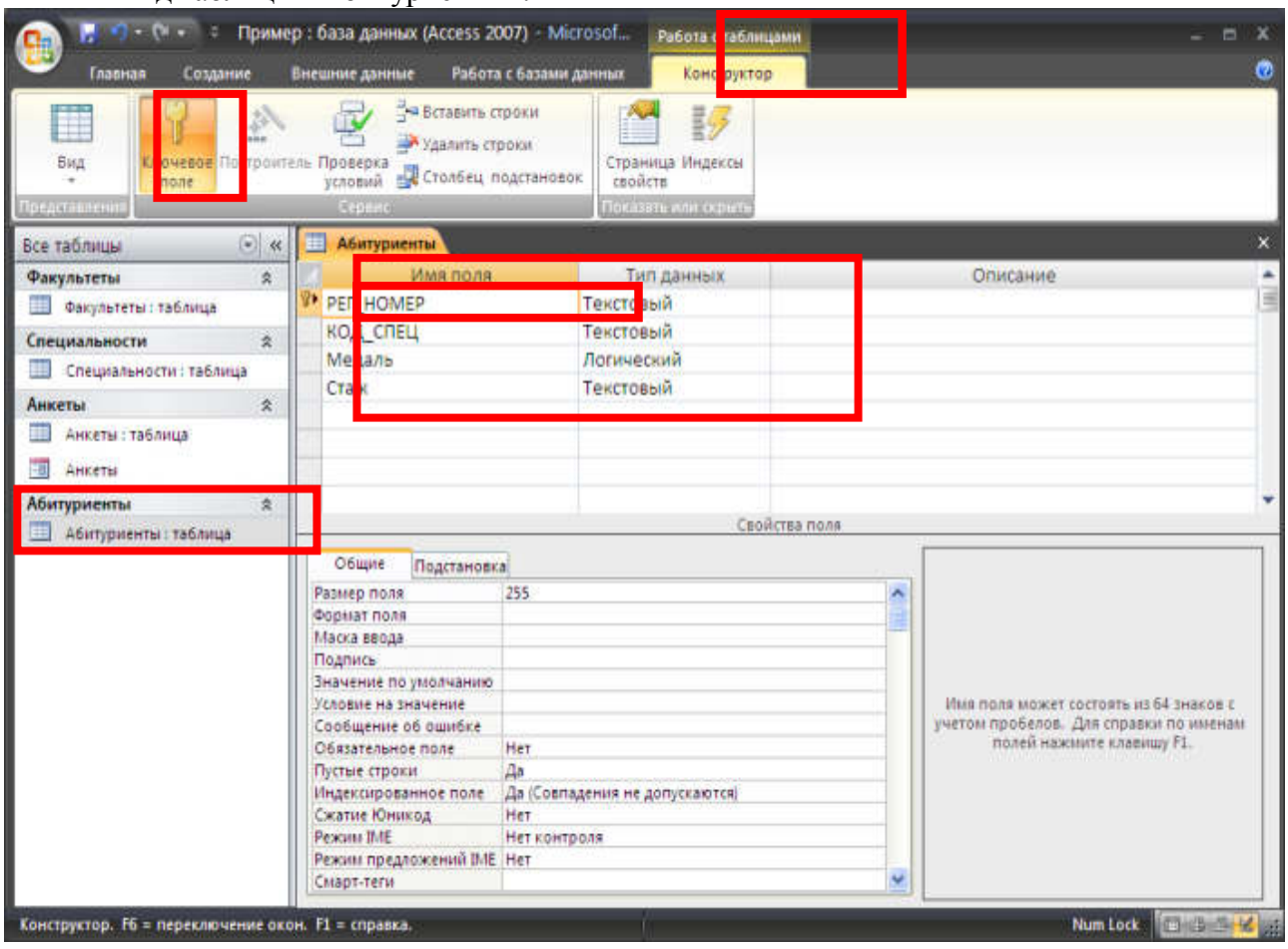
1. Сформируйте в информационной системе еще 2 таблицы: «Анкеты» и «Абитуриенты» (см. часть 1)

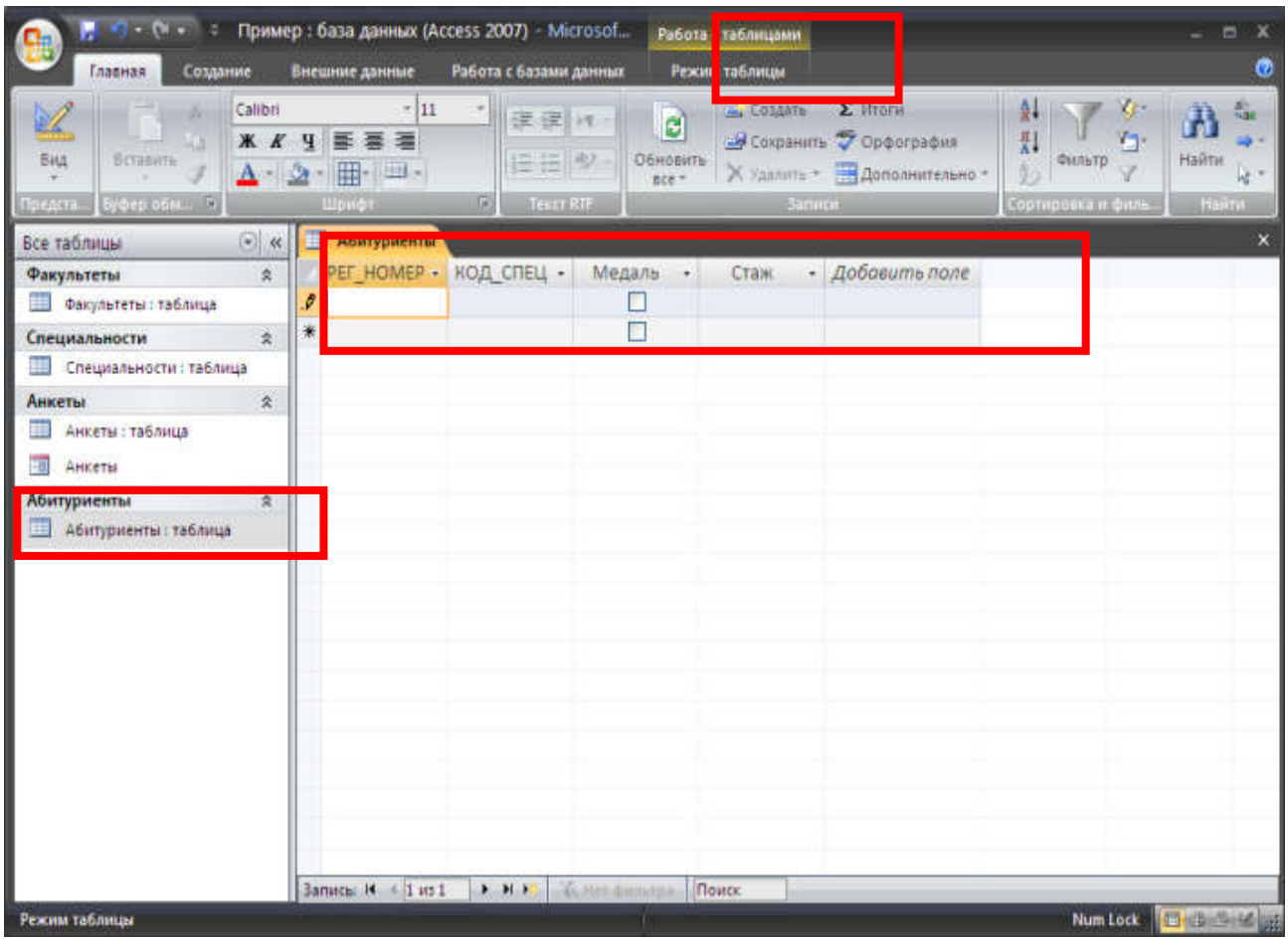
Вид таблицы «Анкеты»:





Вид таблицы «Абитуриенты»:



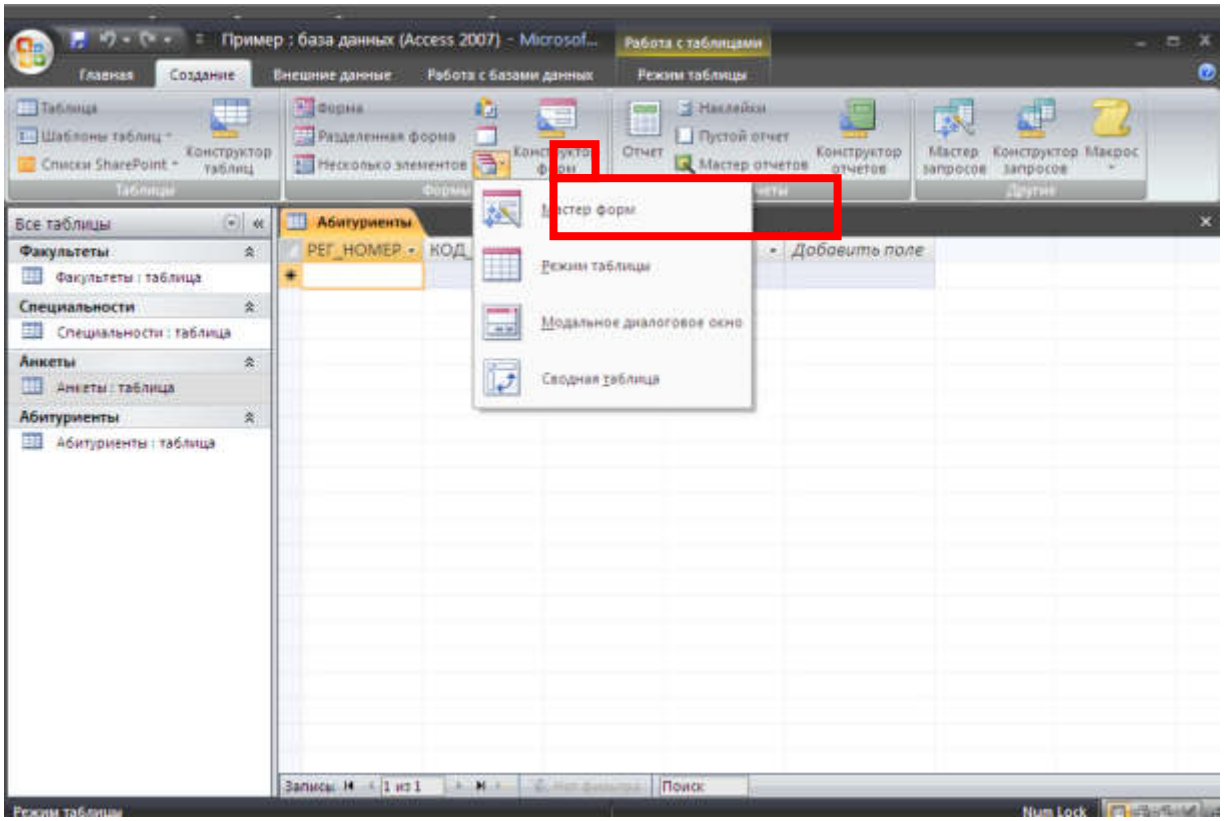


2. Описав с помощью конструктора таблицы «Анкеты» и «Абитуриенты», ***нужно ввести в них данные***.

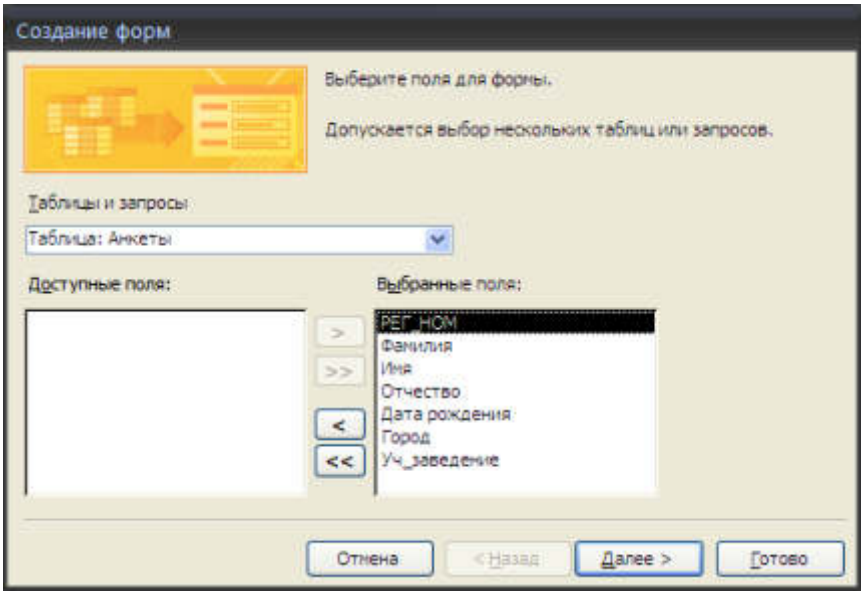
Таблицы содержат такое количество полей, которые в заполненном виде не помещаются на экране. Поэтому в данном случае для заполнения таблиц удобно использовать форму.

Алгоритм создания формы следующий:

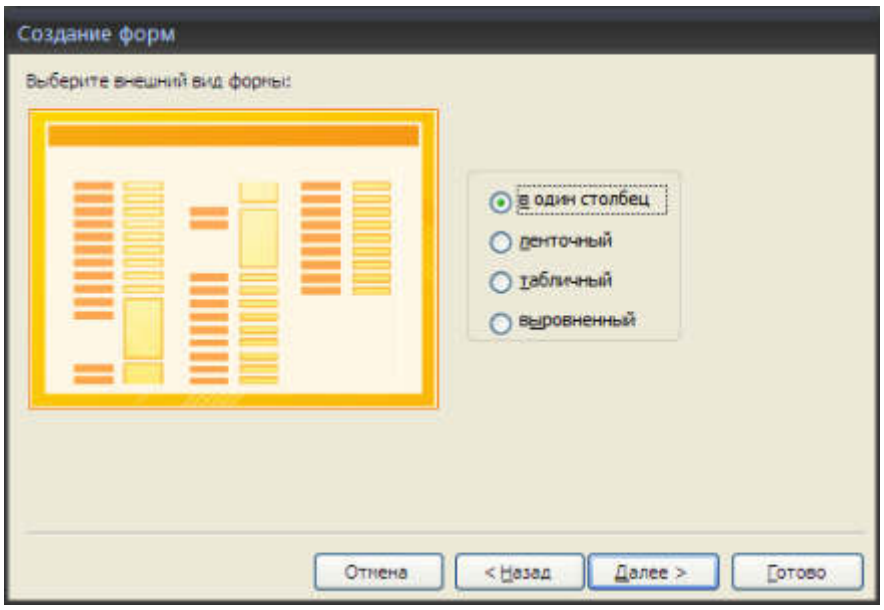
1) *Перейдите на* вкладку Создание *и выберите на* панели Формы кнопку Другие формы – Мастер форм



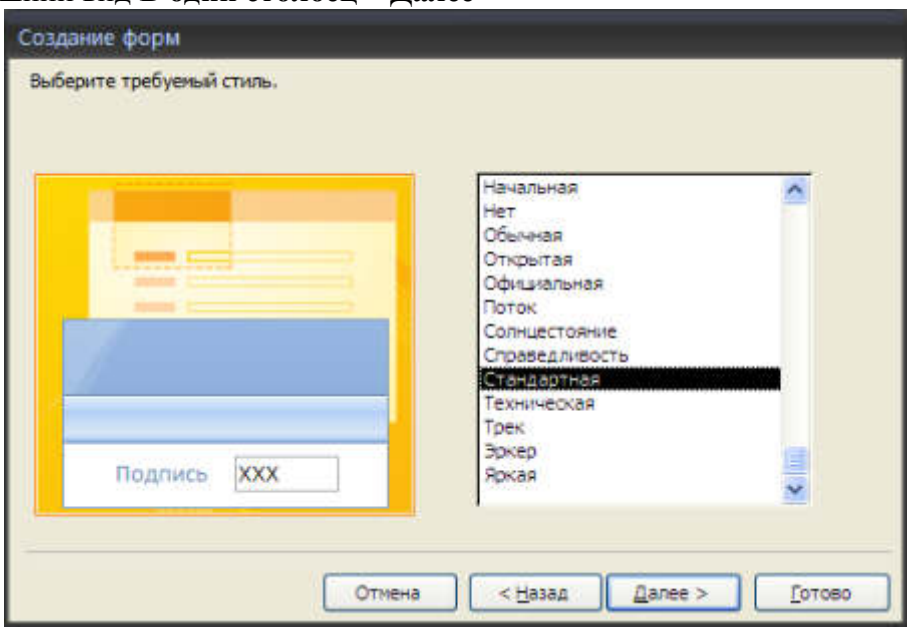
2) В появившемся окне в **раскрывающемся списке Таблицы и запросы** ***выберите*** **таблицу «Анкеты»** и все **Доступные поля** этой таблицы ***переместите*** в **Выбранные поля**, чтоб окно выглядело так:



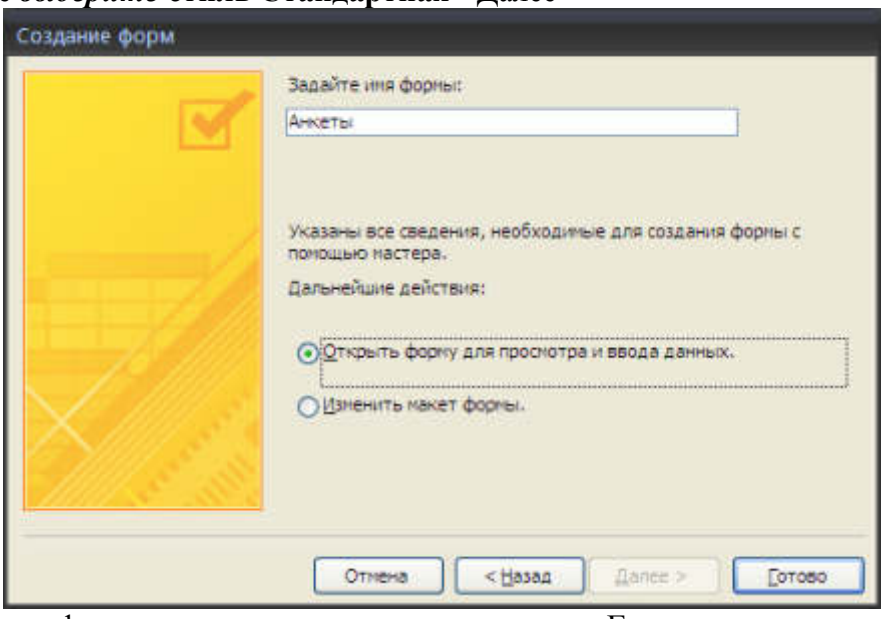
Нажмите **кнопку Далее**



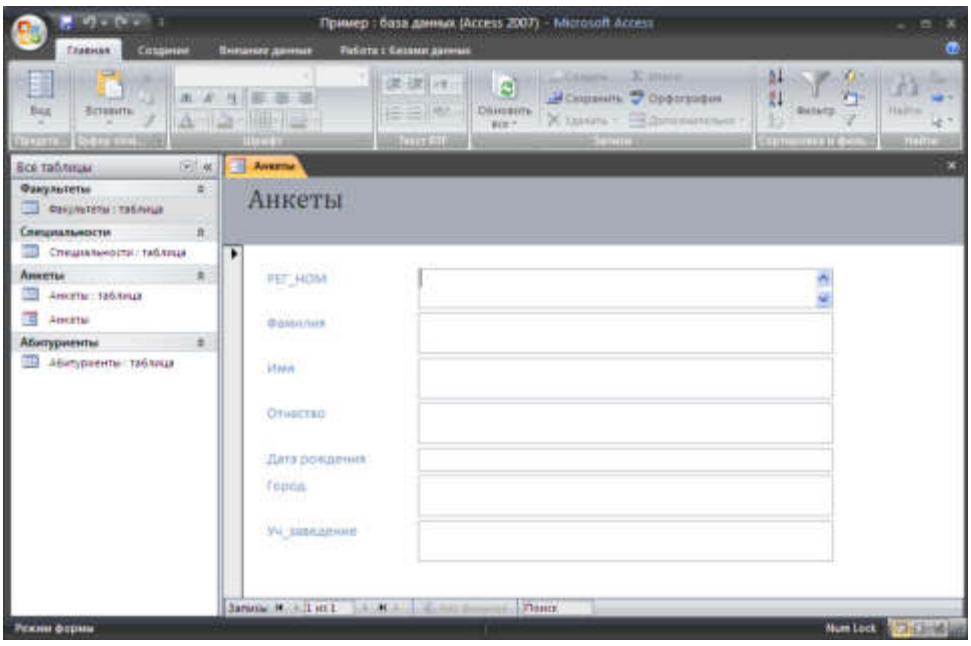
Внешний вид **В один столбец – Далее**



Далее ***выберите*** **стиль Стандартная** - Далее

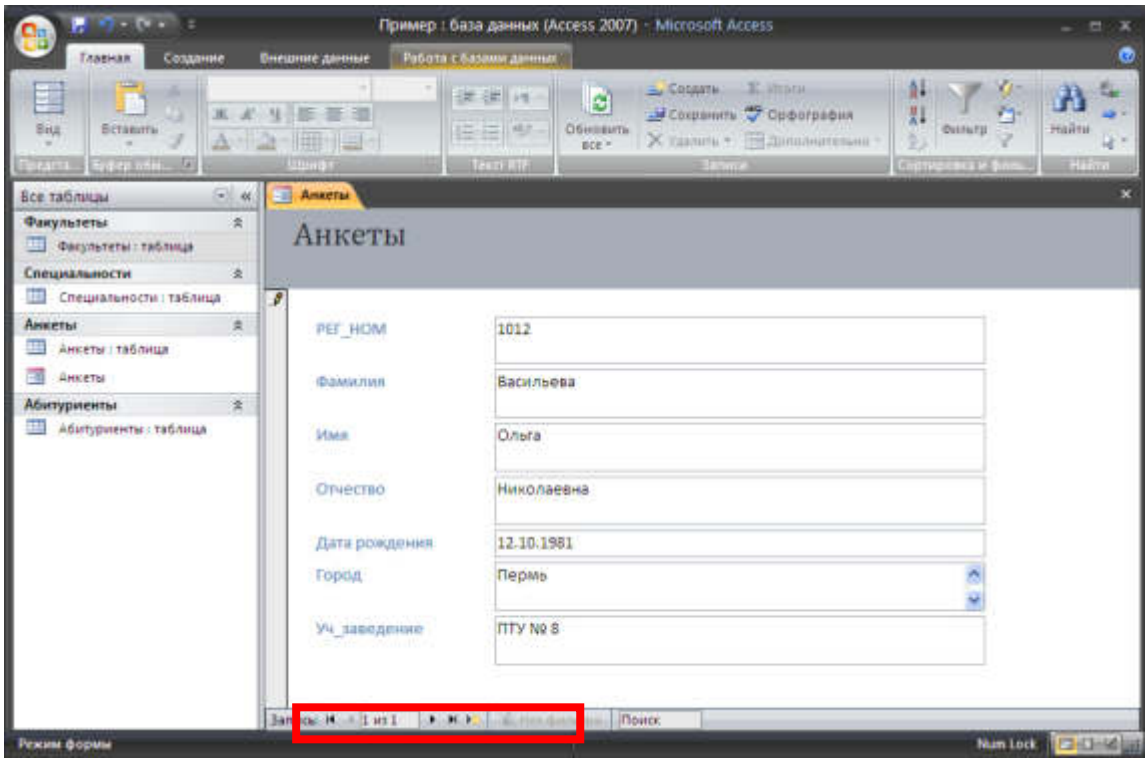


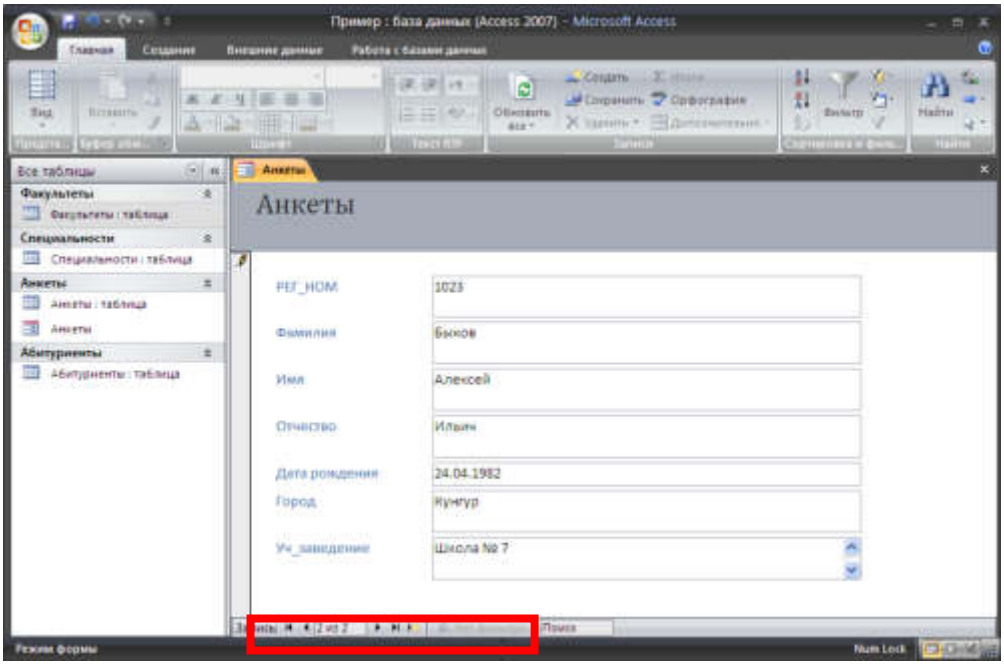
Открыть форму для просмотра и ввода данных – Готово:



3) Теперь через данную форму ***можно последовательно вводить записи в*** **таблицу «Анкеты»**.

Для перехода по записям после создания первой до последней ***используйте*** **кнопки перехода по записям:**





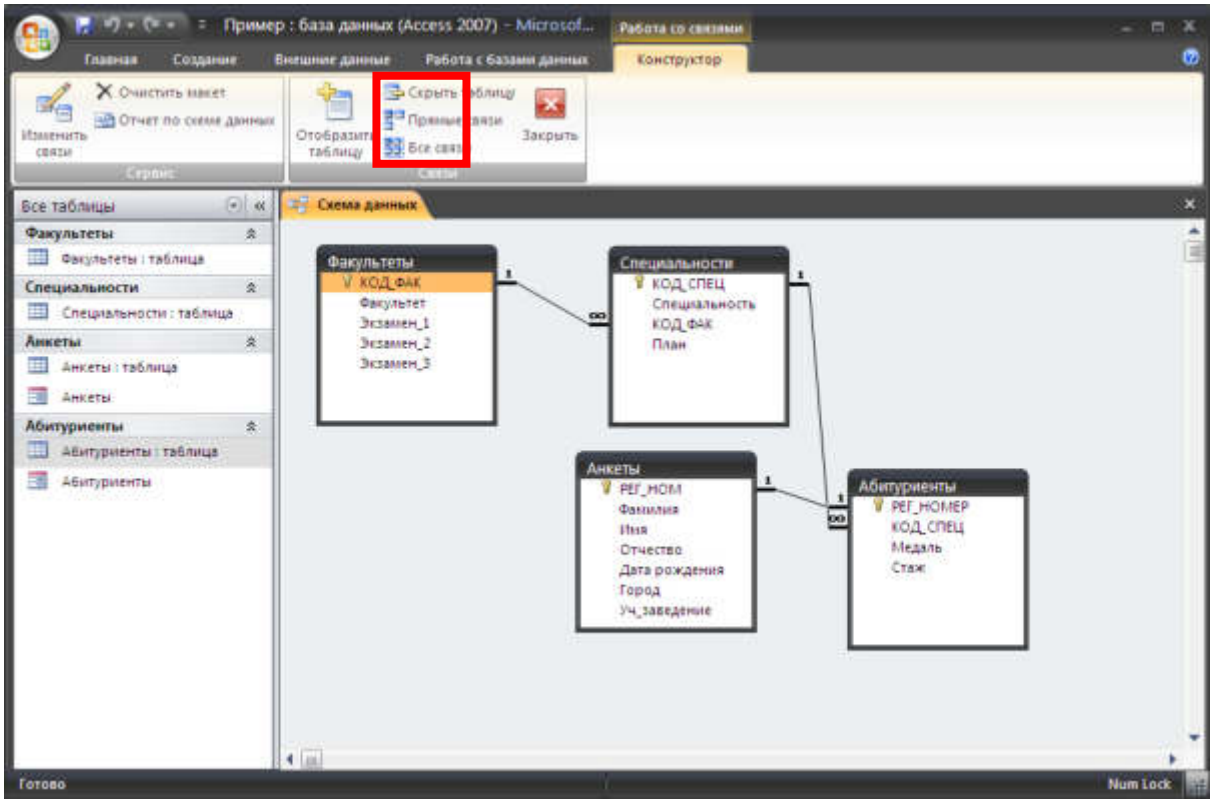
С помощью формы ***заполните*** **таблицу «Анкеты»** следующим образом:

РЕГ_НОМ	Фамилия	Имя	Отчество	Дата рождения	Город	Уч_заведение
1012	Васильева	Ольга	Николаевна	12.10.2004	Пермь	ПТУ № 8
1023	Быков	Алексей	Ильич	24.04.2005	Кунгур	Школа № 7
1119	Круг	Борис	Моисеевич	18.09.2005	Пермь	Школа № 102
1120	Листьев	Дмитрий	Владимирович	01.12.2004	Березники	Школа № 5
2010	Елькин	Виктор	Алексеевич	20.07.2005	Лысьва	ПТУ № 1
2015	Мухин	Олег	Иванович	25.03.2001	Пермь	Школа № 77
2054	Григорьева	Наталья	Дмитриевна	14.02.2003	Березники	Школа № 3
2132	Зубова	Ирина	Афанасьевна	22.11.2004	Пермь	Школа № 96
3005	Анохин	Сергей	Петрович	30.03.2005	Пермь	Школа № 12
3034	Жакин	Николай	Якимович	19.10.2004	Пермь	Школа № 12
3067	Дикий	Илья	Борисович	28.12.1987	Березники	Школа № 3
3118	Ильин	Петр	Викторович	14.07.2003	Кунгур	ПТУ № 8

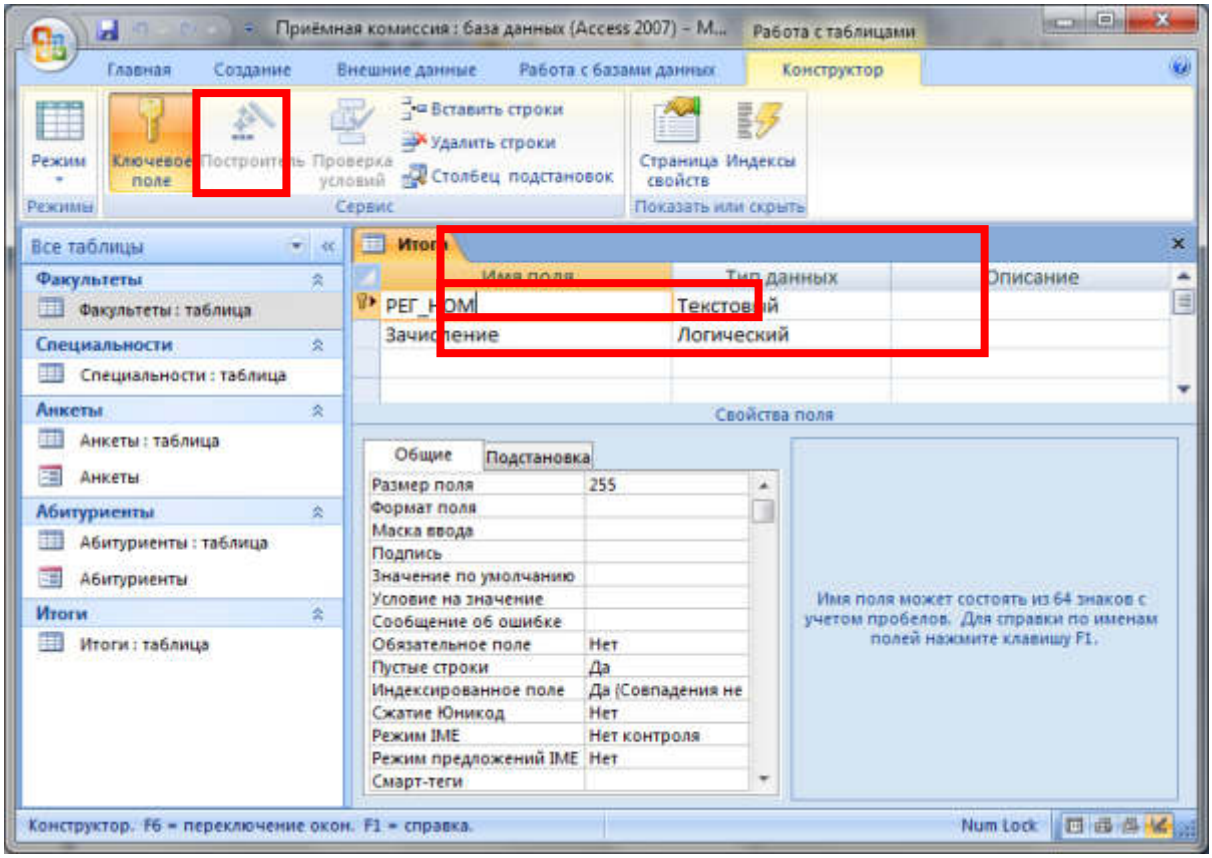
Повторите алгоритм создания и заполнения формы для **таблицы «Абитуриенты»**, ***заполнив следующим образом:***

РЕГ_НОМ	КОД_СПЕЦ	Медаль	Стаж
1012	101	■	1
1023	101	□	0
1119	102	■	0
1120	102	■	0
2010	201	□	0
2015	203	□	3
2054	203	■	2
2132	201	□	0
3005	310	□	0
3034	311	□	1
3067	310	□	3
3118	310	□	2

3. ***Присоедините*** на **схеме данных** таблицы «**Анкеты**» и «**Абитуриенты**» (для добавления новых таблиц в схему данных воспользуйтесь кнопкой Отобразить таблицу) к таблицам «**Факультеты**» и «**Специальности**» так, ***чтобы схема выглядела следующим образом***:



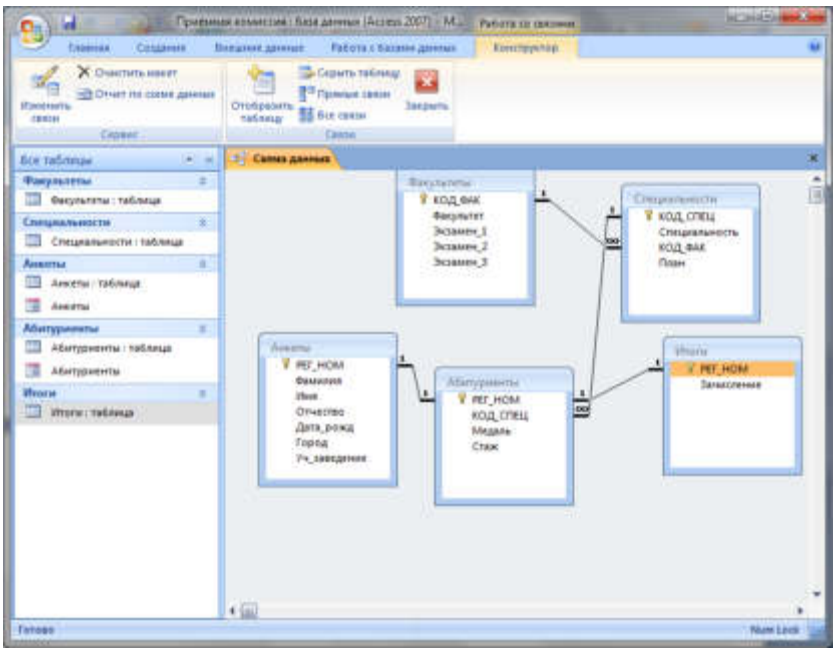
После окончания приема документов можно подготовить таблицу для будущего подведения итогов приема. ***Создайте ее с помощью конструктора***, структура таблицы такая:



Заполните таблицу:

РЕГ_НОМ	Зачисление
1012	☐
1023	☐
1119	☐
1120	☐
2010	☐
2015	☐
2054	☐
2132	☐
3005	☐
3034	☐
3067	☐
3118	☐

Осталось подключить эту таблицу к схеме через **поле РЕГ_НОМ**:



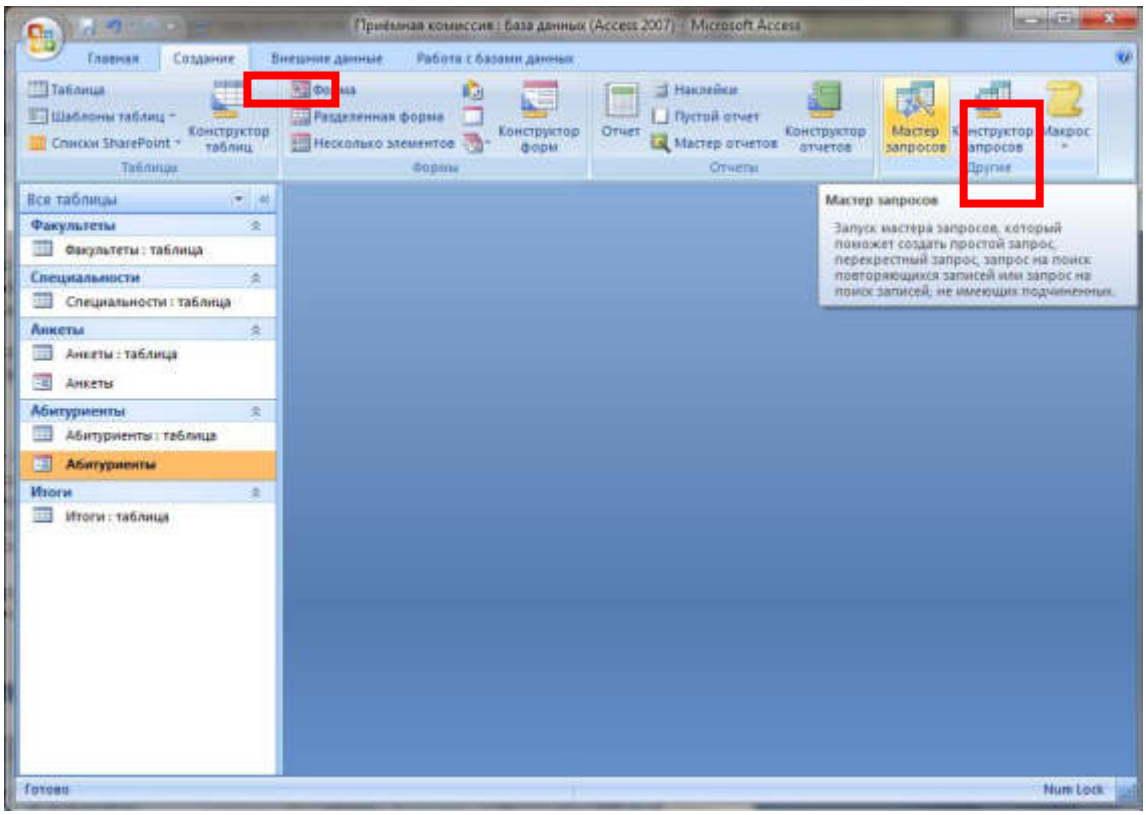
Запрос (query) – это средство выбора необходимой информации из базы данных. Вопрос, сформированный по отношению к базе данных, и есть запрос. Применяются два типа запросов: по образцу (QBE – Query by example) и структурированный язык запросов (SQL – Structured Query Language).

QBE - запрос по образцу – средство для отыскания необходимой информации в базе данных. Он формируется не на специальном языке, а путем заполнения бланка запроса в окне Конструктора запросов.

SQL – запросы – это запросы, которые составляются (программистами) из последовательности SQL – инструкций. Эти инструкции задают, что надо сделать с входным набором данных для генерации выходного набора. Все запросы Access строит на основе SQL – запросов, чтобы посмотреть их, необходимо в активном окне проектирования запроса выполнить команду Вид/SQL.

После того как созданы некоторые таблицы базы данных приёмной комиссии, к ней можно обращаться с любыми запросами, которые могут возникнуть в период приёма документов. ***Сформулируйте несколько таких запросов***.

Чтобы сформулировать запрос ***необходимо перейти*** на **вкладку Создание** и на **панели Другие** выбрать **Мастер запросов**.

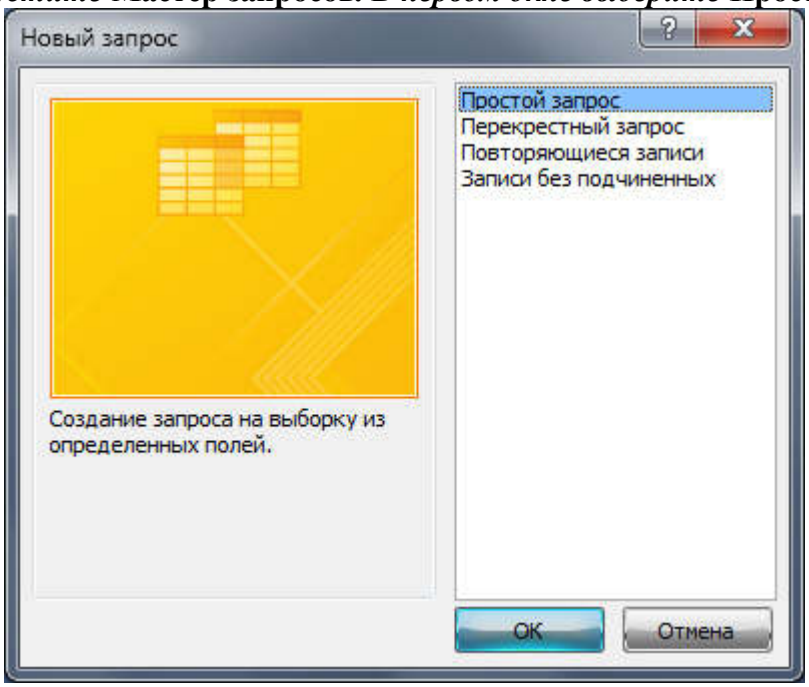


1. Запрос 1. *Получить список всех абитуриентов, живущих в Перми и имеющих медали. В списке указать фамилии, номера школ и факультеты, на которые поступают абитуриенты. Отсортировать список в алфавитном порядке по фамилиям.*

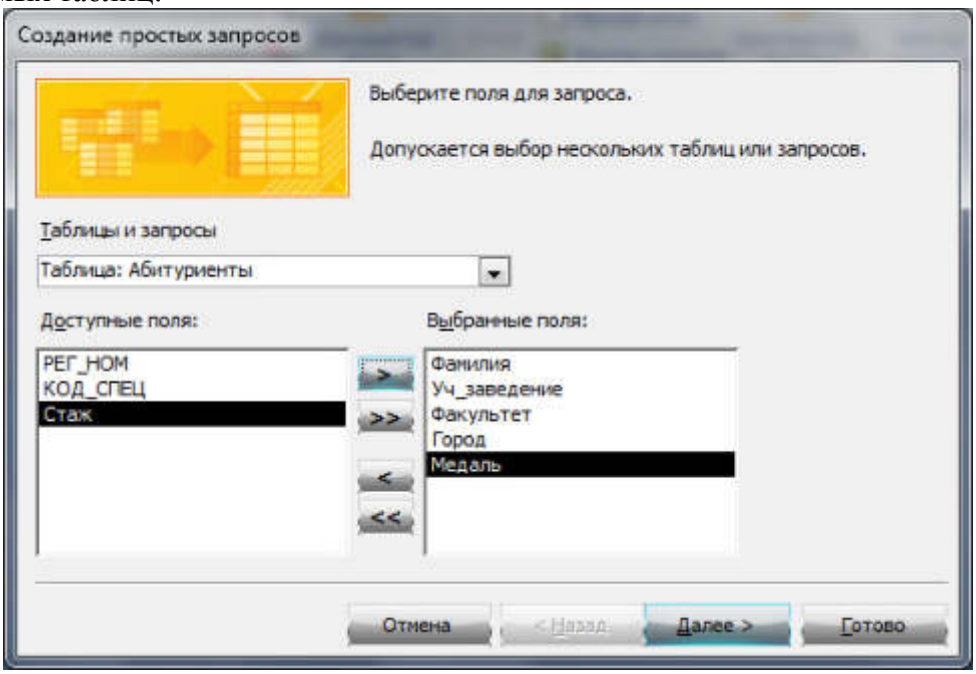
Для реализации данного запроса информация должна извлекаться из трех таблиц: «Анкеты», «Факультеты», «Абитуриенты». На гипотетическом языке команда будет выглядеть так:

.выбор Анкеты.Фамилия, Анкеты.Уч_заведение, Факультеты.Факультет для Анкеты.Город=Пермь и Абитуриенты.Медаль=Да сортировать Анкеты.Фамилия по возрастанию

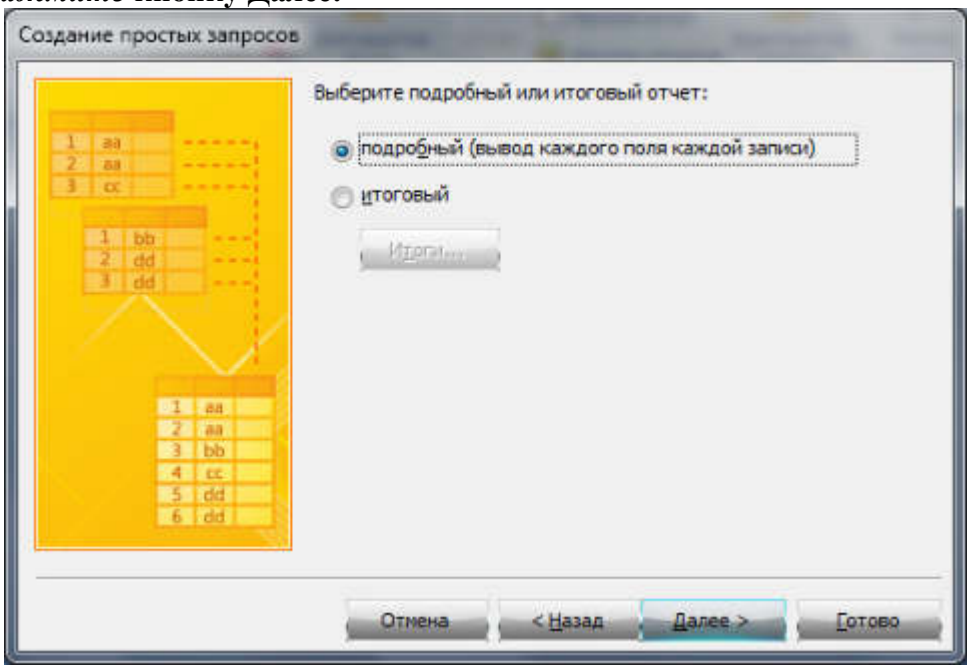
1) ***Запустите*** **Мастер запросов.** ***В первом окне выберите*** **Простой запрос:**



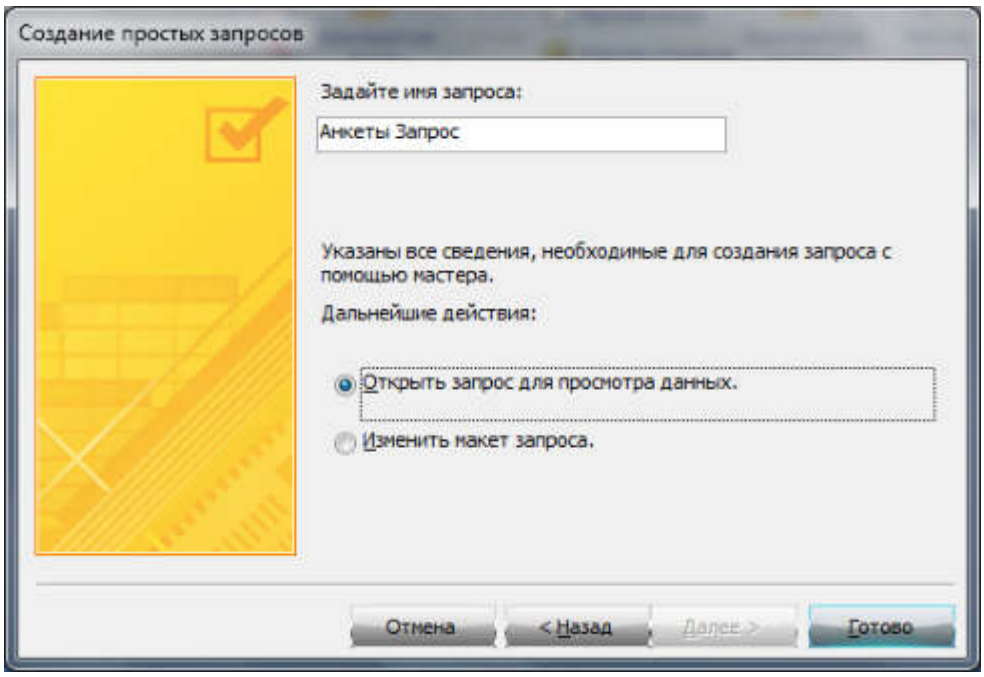
Нажмите* кнопку ОК.** В появившемся окне ***выберите все необходимые поля из необходимых таблиц:



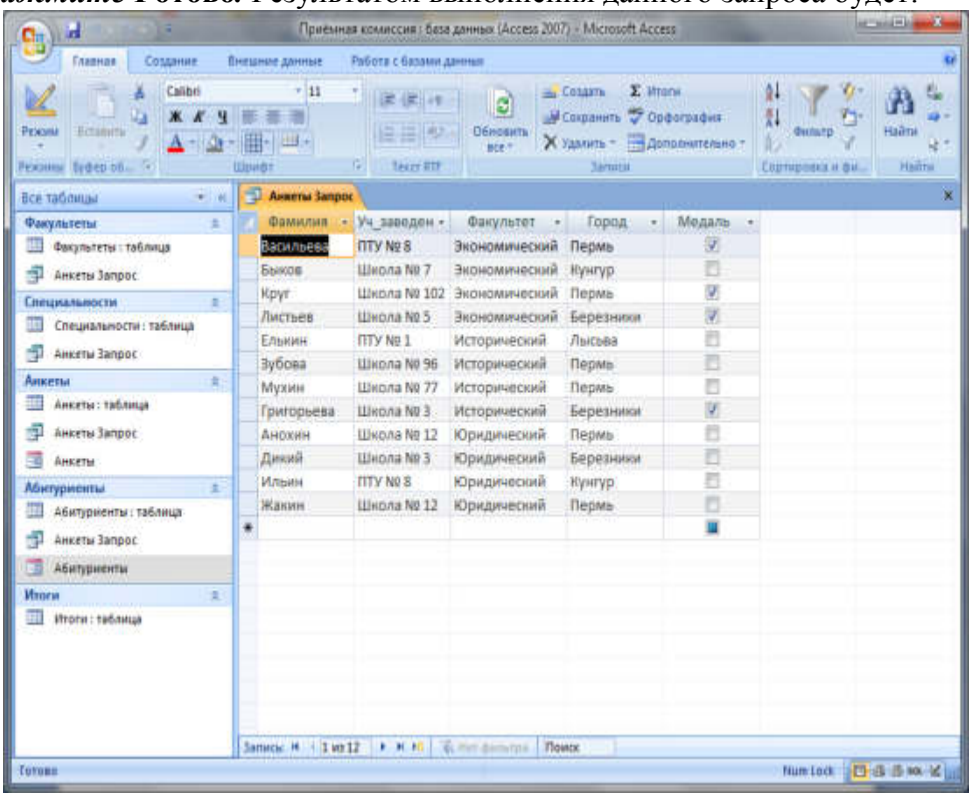
***Нажмите* кнопку Далее:**



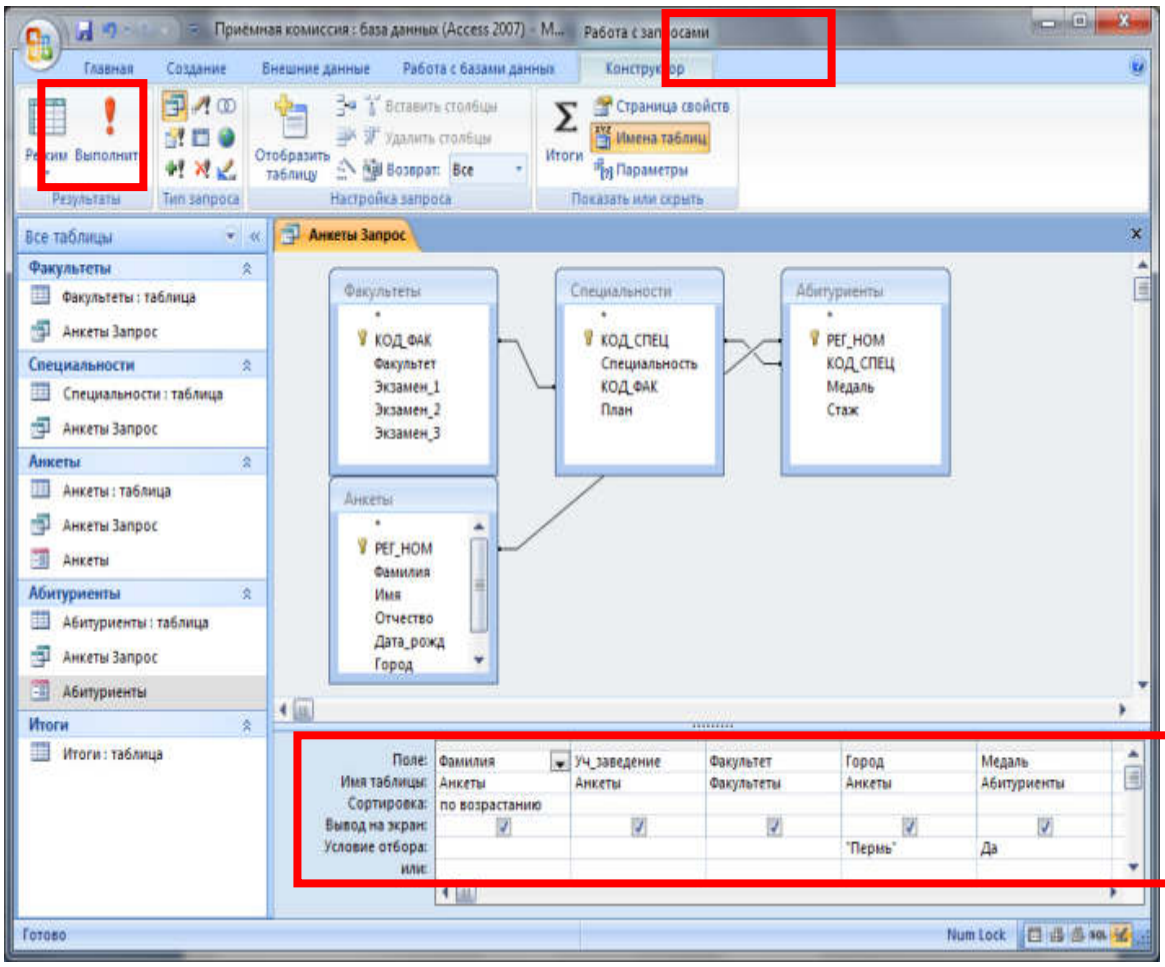
***Нажмите* кнопку Далее:**



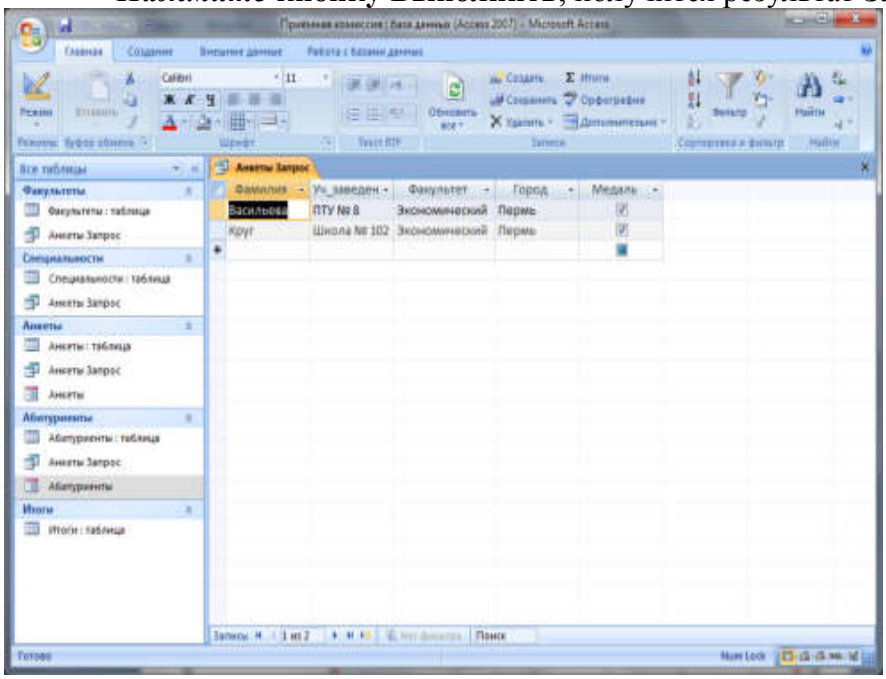
Нажмите **Готово.** Результатом выполнения данного запроса будет:



2) Теперь чтобы получить результат на наш заданный запрос ***необходимо перейти в*** **режим Конструктора запросов** и ***настроить*** данный запрос следующим образом:

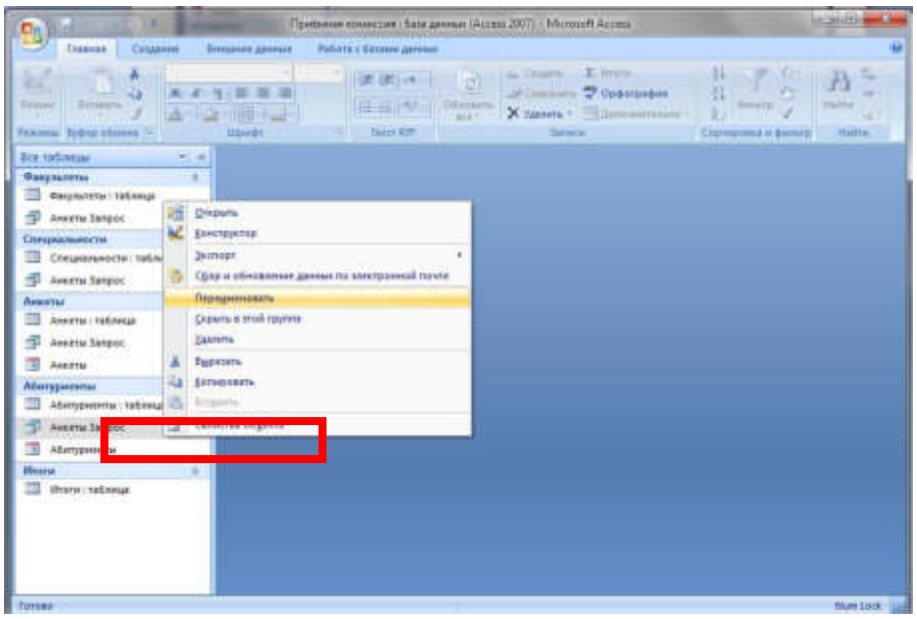


Нажмите **кнопку Выполнить**, получится результат **Запроса 1**:



2. *Переименуйте* Анкеты Запрос *в* Пермяки-медалисты.

Для этого сначала ***сохраните и закройте*** полученный результат, затем в списке объектов правой кнопкой щелкните по «**Анкеты Запрос**» и выберите **функцию Переименовать**:



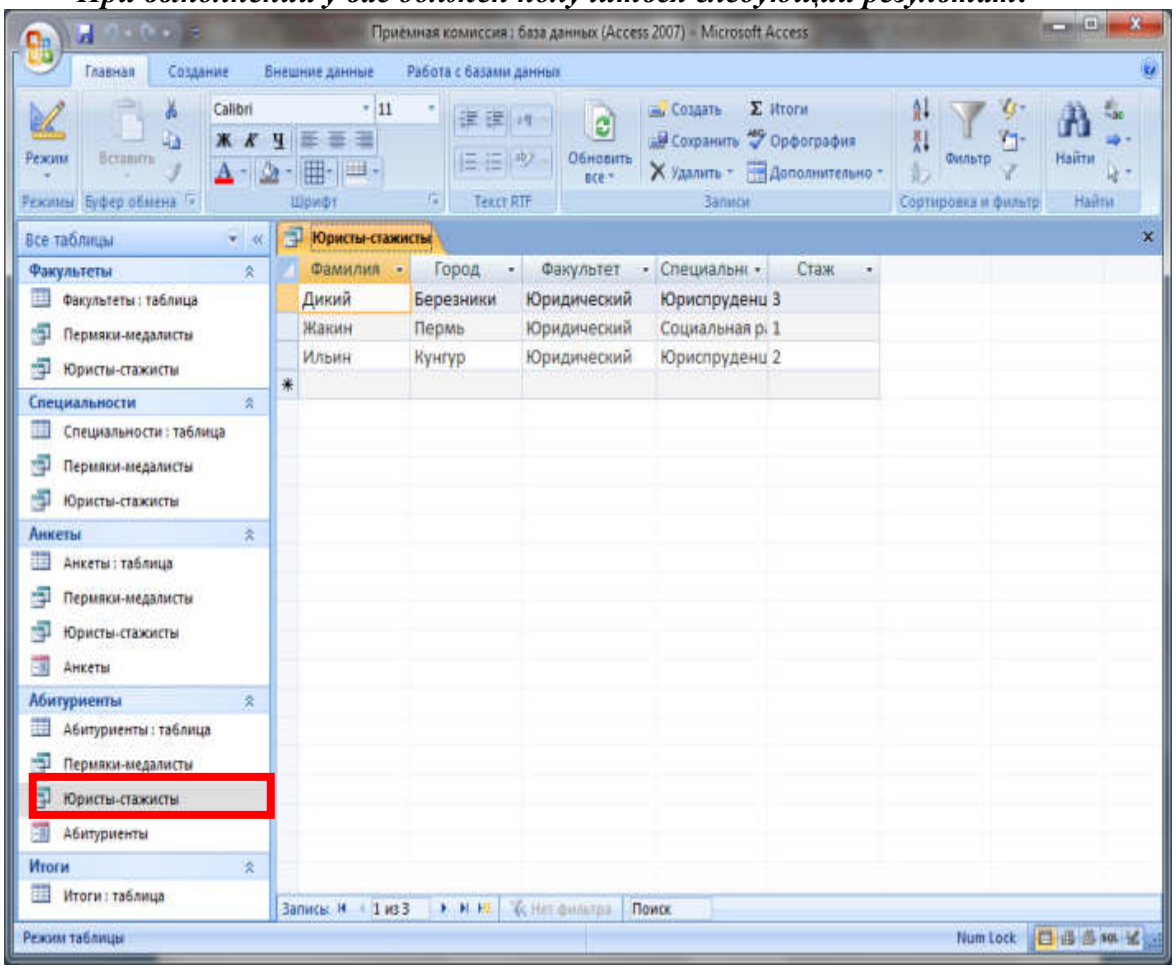
3. ***Создайте запрос*** **Юристы-стажисты.**

Запрос 2. *Получить список всех абитуриентов, поступающих на юридический факультет, имеющих производственный стаж. Указать фамилии, названия городов, специальности и стаж. Упорядочить список по фамилиям в алфавитном порядке.*

На гипотетическом языке запросов команда выглядит так:

.выбор Анкеты.Фамилия, Анкеты.Город, Специальности.Специальность, Абитуриенты.Стаж для Факультеты.Факультет=Юридический и Абитуриенты.Стаж>0 сортировать Анкеты.Фамилия по возрастанию.

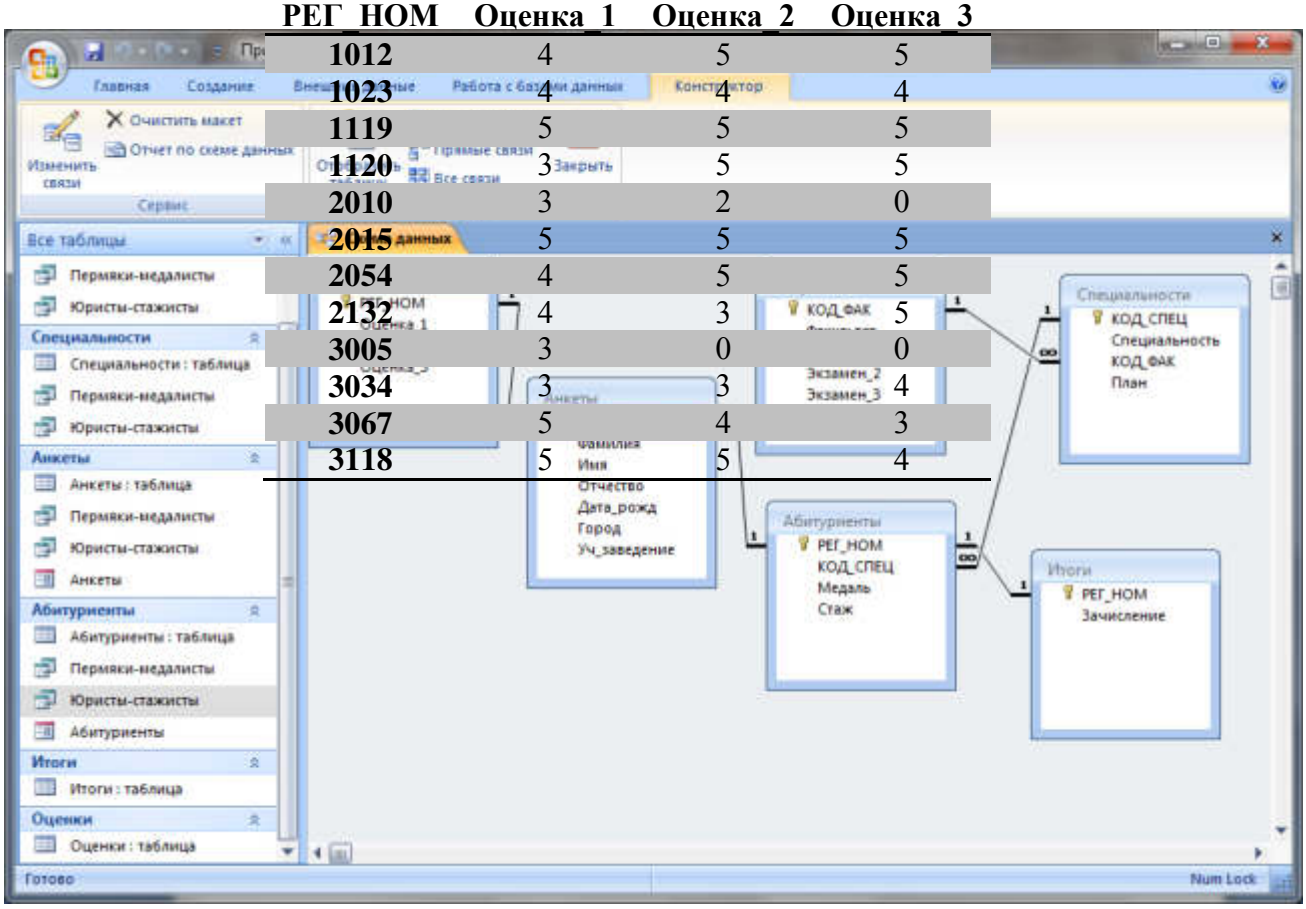
При выполнении у вас должен получиться следующий результат:



Наступил самый трудный этап для абитуриентов – сдача вступительных экзаменов.

1. ***Пора создать*** в базе данных **таблицу «Оценки»** *(тип данных: РЕГ_НОМ – текстовый, Оценка_1,Оценка_2, Оценка_3 - числовой)*:

РЕГ_НОМ	Оценка_1	Оценка_2	Оценка_3
1012	4	5	5
1023	4	4	4
1119	5	5	5
1120	3	5	5
2010	3	2	0
2015	5	5	5
2054	4	5	5
2132	4	3	5
3005	3	0	0
3034	3	3	4
3067	5	4	3
3118	5	5	4



Ноль баллов в таблице выставляется за неявку на экзамен. Теперь базу данных нужно подготовить к проведению зачисления.

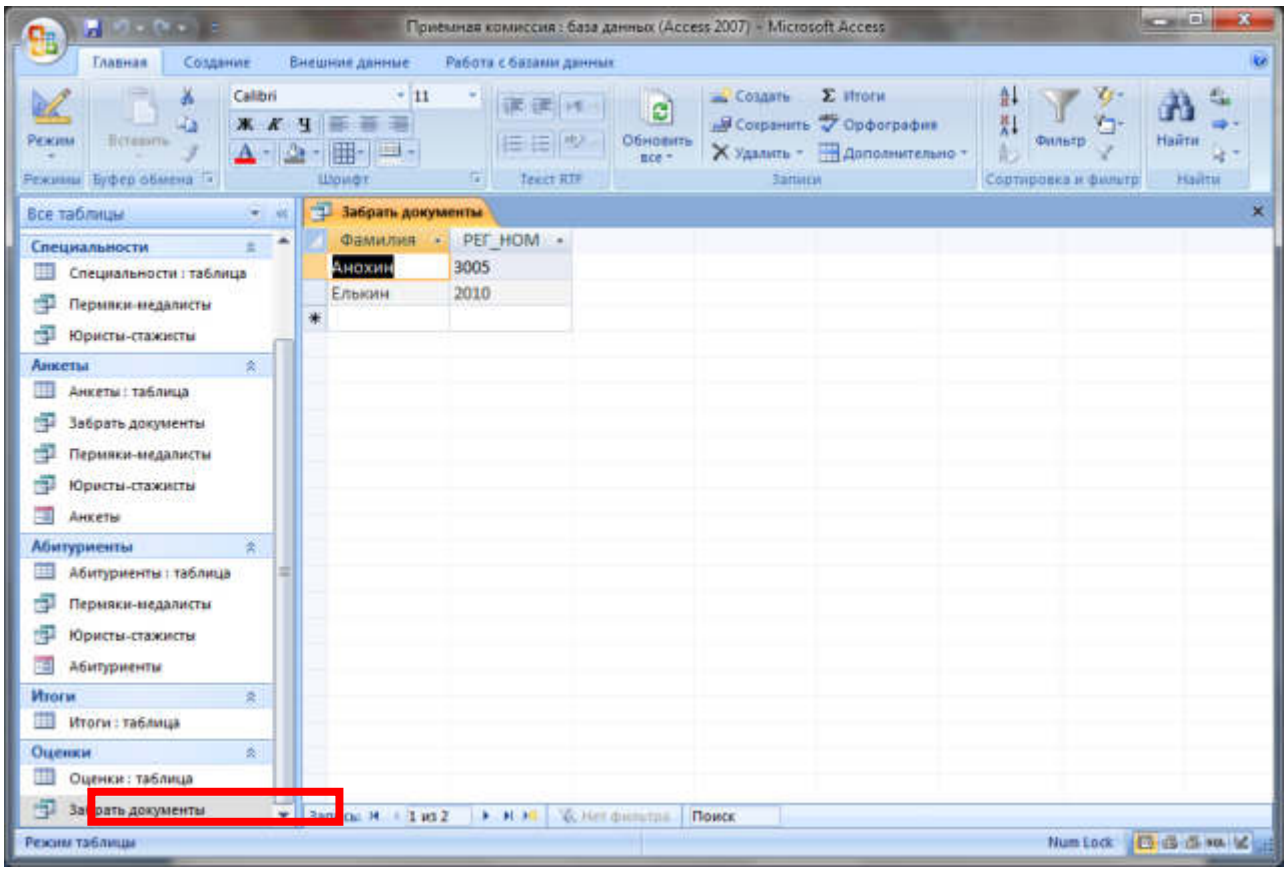
2. ***Создайте запрос*** **Забрать документы.**

Запрос 3. *Получим список абитуриентов, которые получили двойки или не явились на экзамены. Они в дальнейшем конкурсе принимать участия не будут. им следует забрать свои документы из приёмной комиссии.*

На языке запросов команда будет выглядеть так:

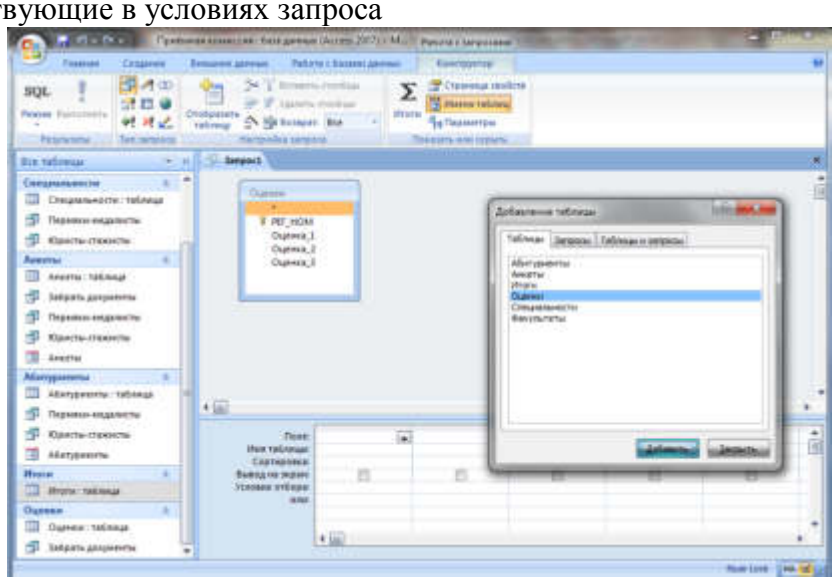
.выбор Анкеты.Фамилия, Оценки.РЕГ_НОМ, для Оценки.Оценка_1<3 или Оценки.Оценка_2<3 или Оценки.Оценка_3<3 сортировать Анкеты.Фамилия по возрастанию

При выполнении у вас должен получиться следующий результат:



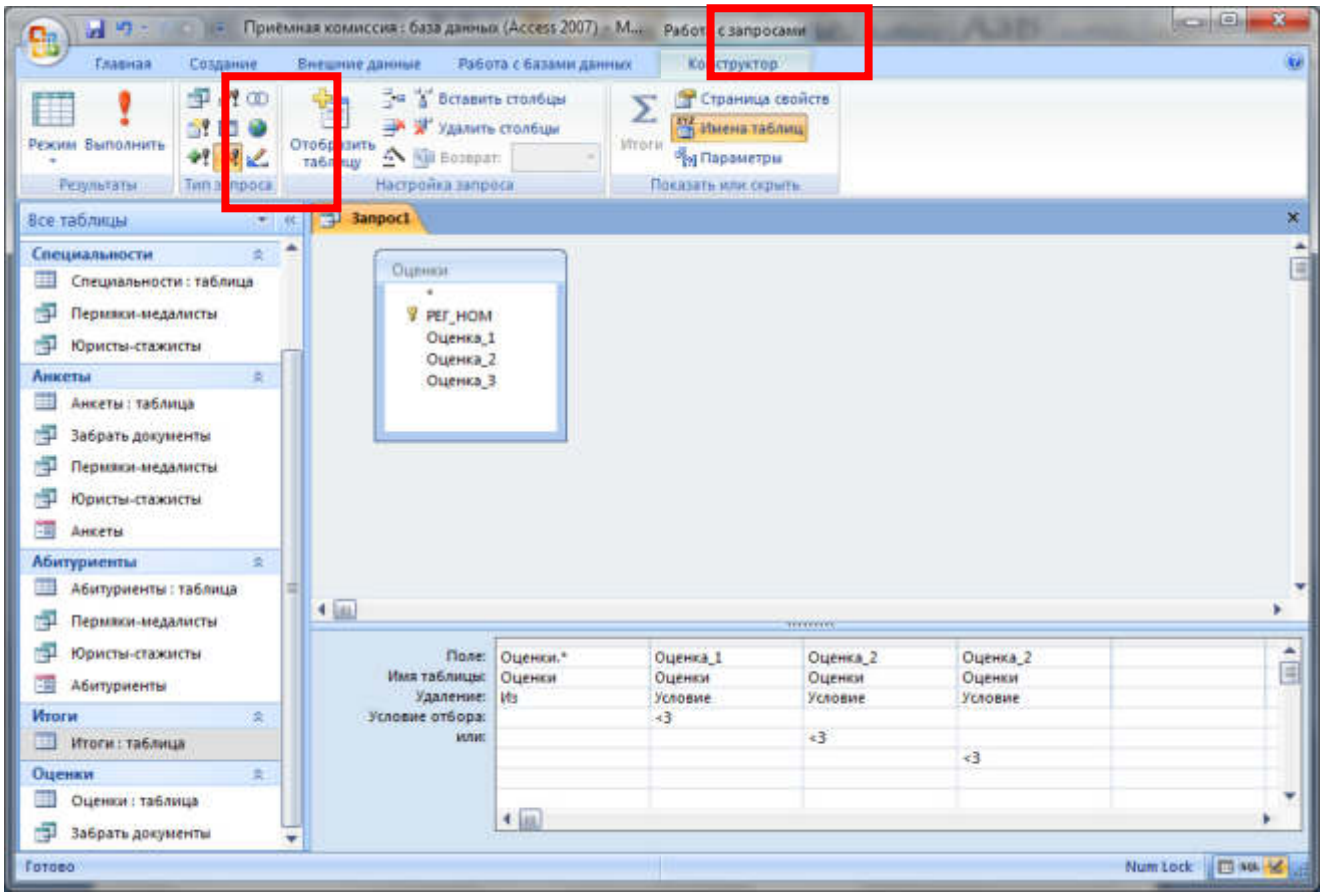
Запросы можно реализовать и с помощью **Конструктора запросов** *(вкладка Создание – панель Другие – Конструктор запросов).*

Нажав на кнопку Конструктор запросов вам предлагается выбрать таблицу или таблицы, участвующие в условиях запроса

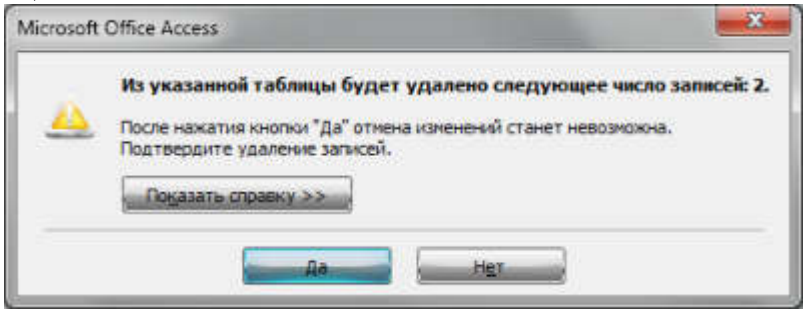


1. Выполните **запрос 5 на удаление** из **таблицы «Оценки»** сведения об абитуриентах, получивших двойки или не явившихся на экзамены.

Для этого ***создайте запрос*** с помощью **Конструктора запросов**, бланк данных которого выглядит так:



При правильно введенных данных после нажатия **кнопки Выполнить** должно появится следующее окно:

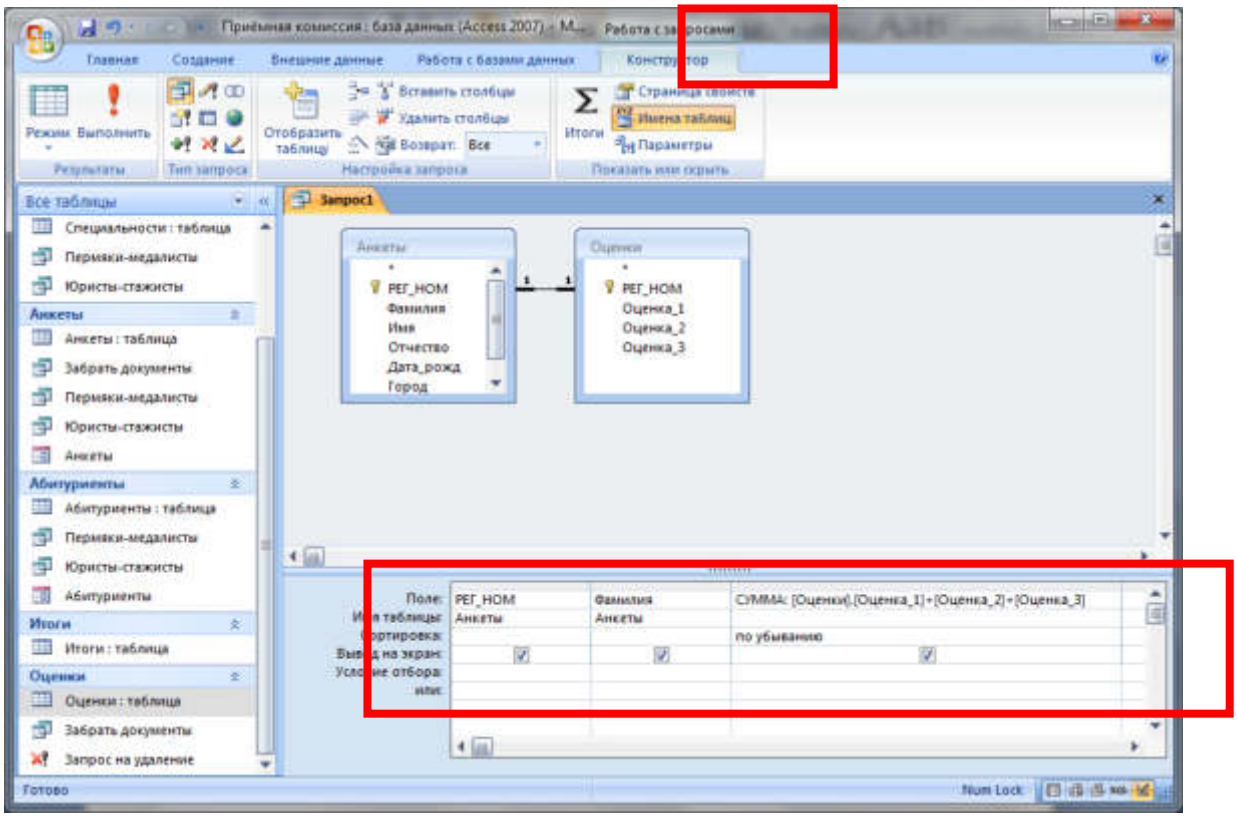


Нажмите* Да.** И ***посмотрите что произошло с **таблицей Оценки.**

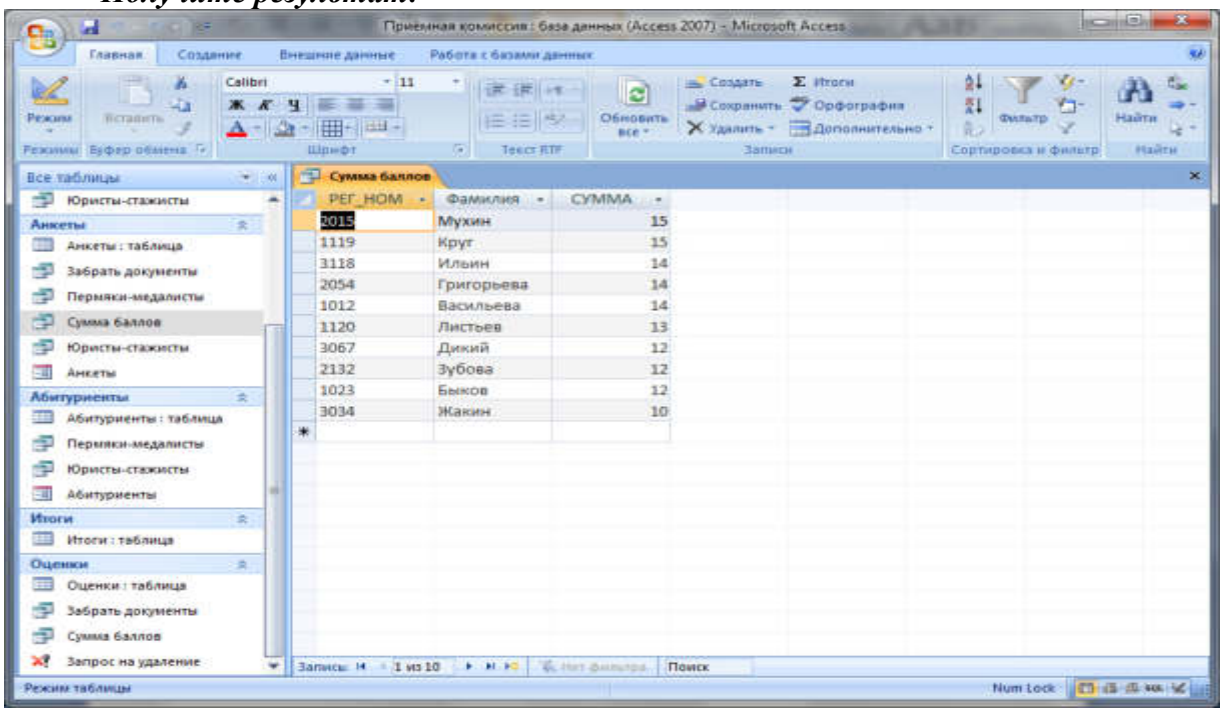
2. ***Выполните* вычисляемый запрос.**

Запрос 6. *Вывести таблицу со значениями сумм баллов, включив в нее регистрационные номера, фамилии и суммы баллов. Отсортировать таблицу по убыванию суммы.*

В **Конструкторе** запрос должен иметь вид:



Получите результат:

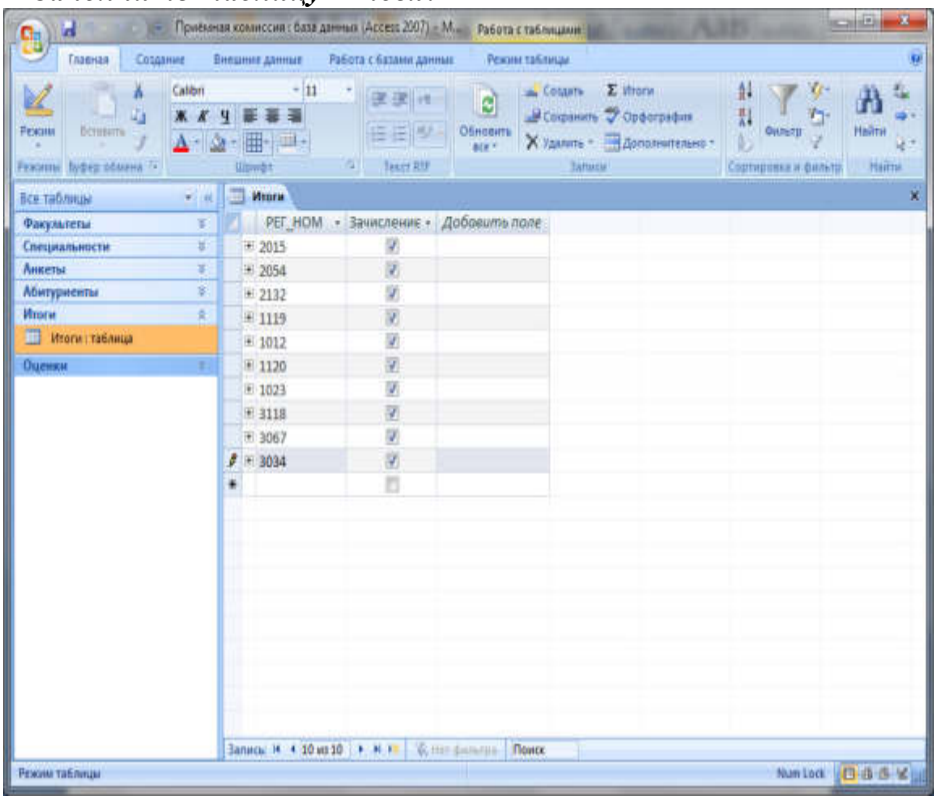


Отчет – итоговый документ. Средства построения отчета позволяют гибко управлять его оформлением.

Информацию, заносимую в отчет, можно извлекать из таблиц базы данных, а также из таблиц запросов.

В СУБД MS Access отчет – четвёртый тип объекта, с которым вы познакомитесь. Наиболее гибкий способ создания отчета – использование конструктора. Однако сейчас мы пойдем простым путем и воспользуемся **Мастером отчетов**, имеющимся в MS Access.

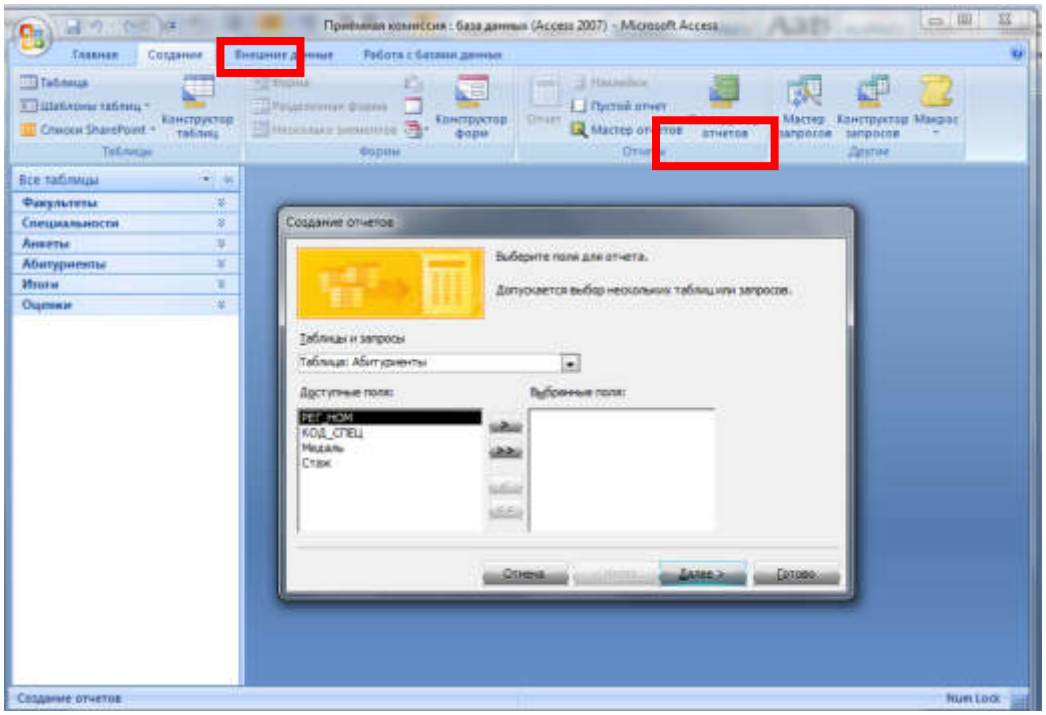
1. ***Заполните таблицу Итоги***:



2. ***Создайте Отчет об итогах вступительных экзаменов.***

Для этого необходимо выполнить следующее:

1) Перейти на **вкладку Создание** ***и на*** **панели Отчеты** ***выбрать*** **Мастер отчетов*:***

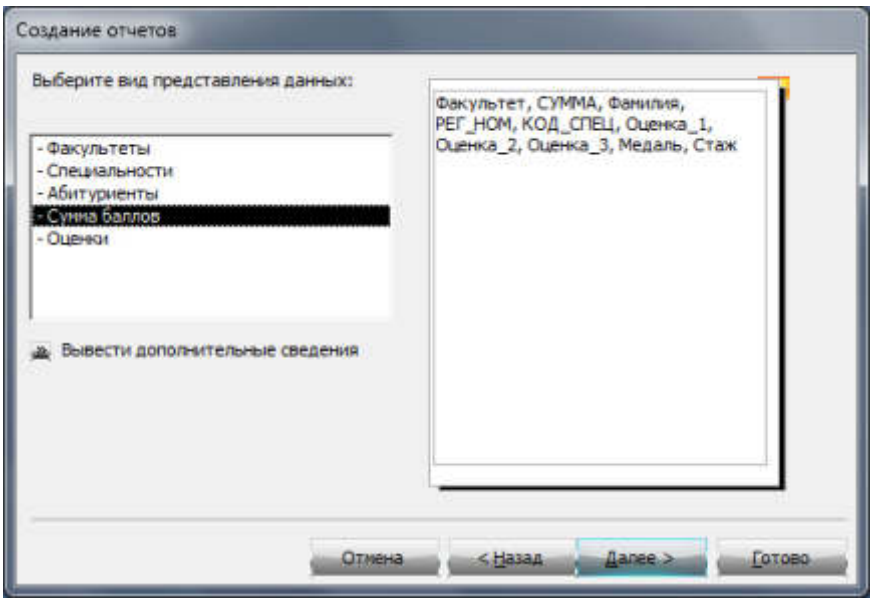


2) В появившемся окне выбрать поля с указанием таблиц и запросов в строгом соответствии с их последовательностью, согласно **таблице «К зачислению»*:***

К зачислению									
Факультет	**Сумма баллов**	**Фамилия**	**РЕГ_НОМ**	**КОД_СПЕЦ**	**Экз. 1**	**Экз. 2**	**Экз. 3**	**Медаль**	**Стаж раб.**
Исторический									
	15	Мухин	2015	203	5	5	5	□	3
	14	Григорьева	2054	203	4	5	5	■	2
	12	Зубова	2132	201	4	3	5	□	0
Экономический									
	15	Круг	1119	102	5	5	5	■	0
	14	Васильева	1012	101	4	5	5	■	1
	13	Листьев	1120	102	3	5	5	■	0
	12	Быков	1023	101	4	4	4	□	0
Юридический									
	14	Ильин	3118	310	5	5	4	□	2
	12	Дикий	3067	310	5	4	3	□	3
	10	Жакин	3034	311	3	3	4	□	1

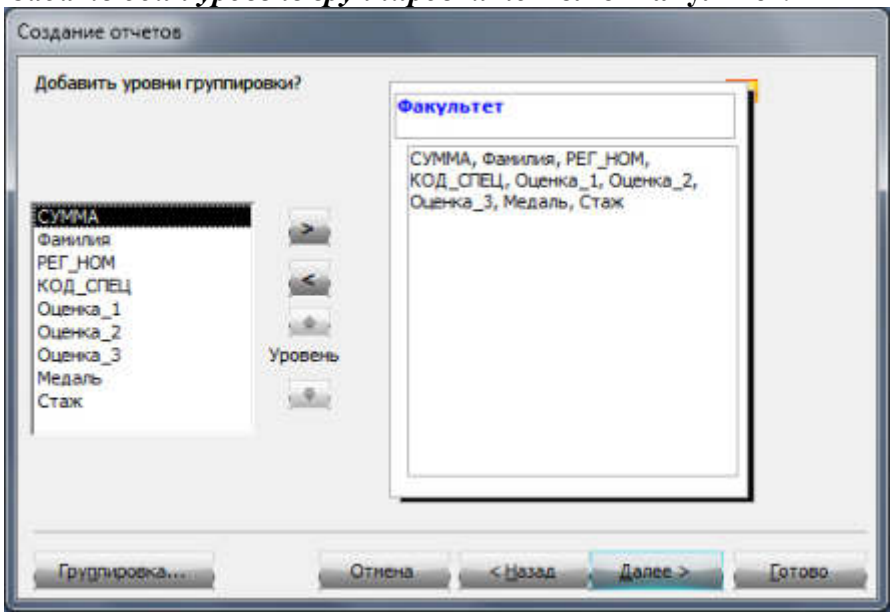
Далее

3) В следующем окне выбрать вид представления данных



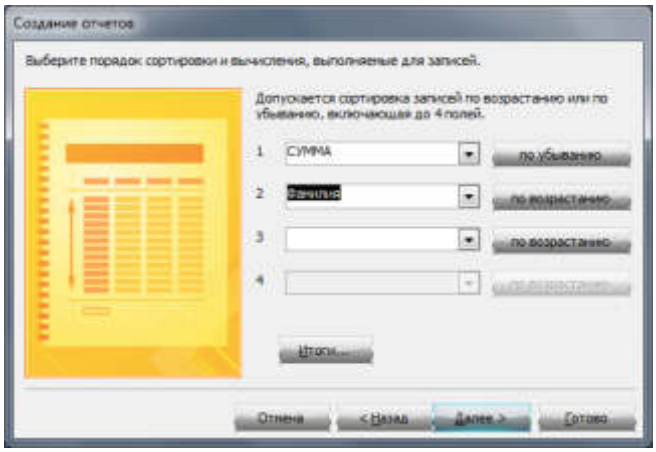
Далее

4) Задать один уровень группировки по* полю Факультет*:



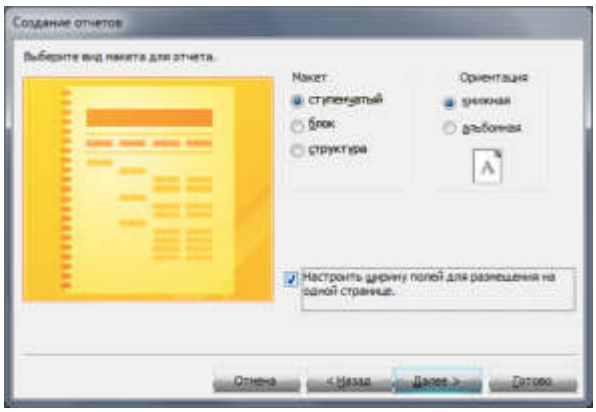
Далее

5) Указать порядок сортировки:* СУММА – по убыванию, *или* Фамилия – по возрастанию*:



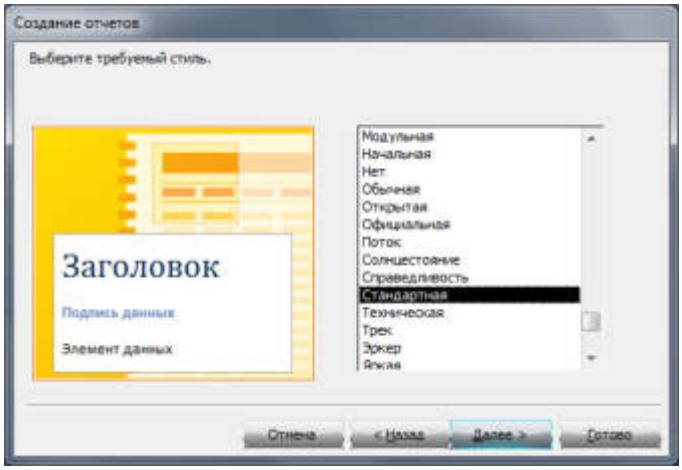
Далее

6) Выбрать вид макета для отчета:



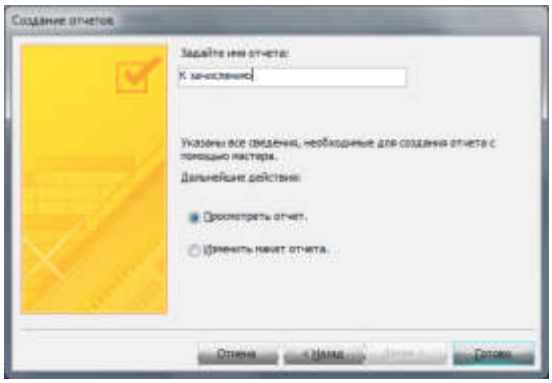
Далее

7) Выбрать стиль:

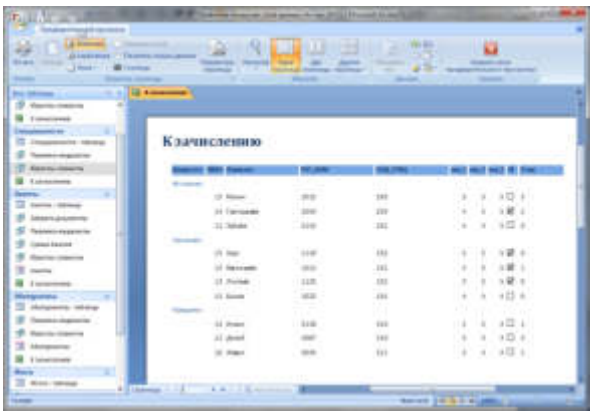


Далее

8) Указать имя отчета «К зачислению»:

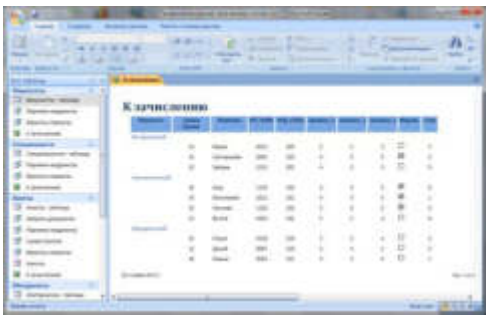


Готово:

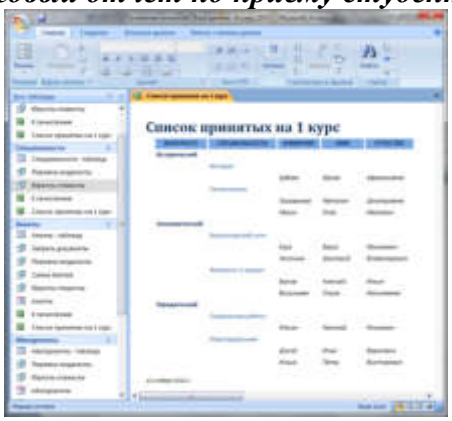


***9) Нажать* кнопку Закрыть окно предварительного просмотра.**

10) Перейдите в* режим Макета *и приведите отчет к такому виду:



3. ***Создайте итоговый отчет по приему студентов на первый курс:***



Отчет

Отчет должен быть оформлен в текстовом редакторе и содержать:

- наименование работы;
- цель работы;
- задание;
- последовательность выполнения работы;
- ответы на контрольные вопросы;

вывод о проделанной работе

Практическая работа №16

Создание документа в редакторе баз данных по заданным условиям

Создать файл базы данных ***Akademik.mdb.*** С помощью конструктора подготовить таблицу ***Члены-корреспонденты Академии наук.*** Заполнить таблицу данными, включив следующие поля: ***ФИО, Дата рождения, Специализация, Пол, Год присвоения звания.***

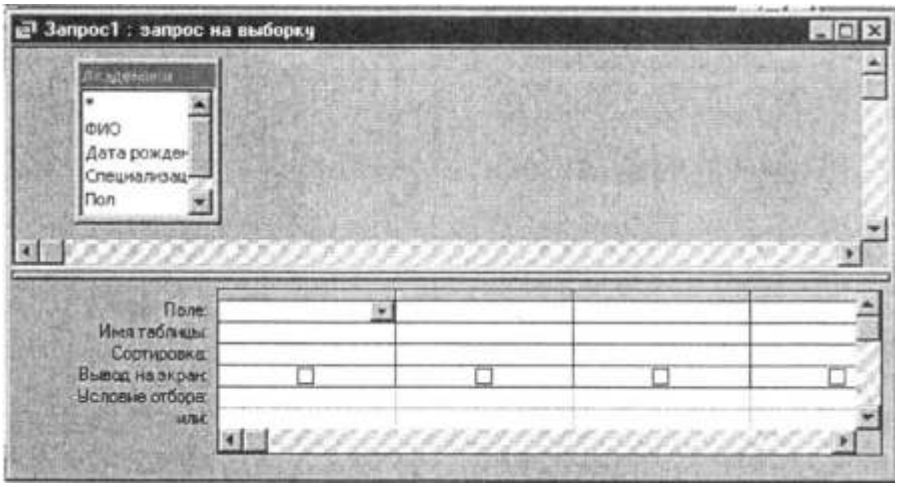
ФИО	Дата рождения	Специали-зация	Пол	Год присвоения звания
Александрович А. И.	22.01.1906	поэт	м	1930
Амбросов А. Л.	16.06.1912	фитопатолог-вирусолог	м	1970
Аринчин М. И.	28.02.1914	физиолог	м	1966
Бабосов Е. М.	23.02.1931	философ	м	1977
Бирич Т. В.	10.01.1905	офтальмолог	ж	1972
Бокуть Б. В.	27.10.1926	физик	м	1974
Бондарчик В. К.	01.08.1920	этнограф	м	1972
Будыка С. Х.	17.03.1909	гидролог	м	1972
Гуринович Г. П.	26.04.1933	физик	м	1970
Иванов А. П.	29.12.1929	физик	м	1974
Каменская Н. В.	10.01.1914	историк	ж	1959
Комаров В. С.	29.01.1923	химик	м	1970
Кулаковская Т. Н.	17.02.1919	агрохимик-почвовед	ж	1969
Мацкевич Ю. Ф.	27.07.1911	языковед	ж	1969
Пилипович В. А.	05.01.1931	физик	м	1977
Сикорский В. М.	10.10.1923	историк	м	1972
Старобинец Г. Л	14.05.1910	химик	м	1969
Судник М. Р.	08.11.1910	языковед	м	1970
Ткачев В. Д.	19.02.1939	физик	м	1974
Хотылева Л. В.	12.03.1928	генетик	ж	1972
Шабуня К. И.	28.10.1912	историк	м	1969
Широканов Д. И.	20.05.1929	философ	м	1974

Задание 1

Для таблицы Члены-корреспонденты Академии наук файла базы данных Akademik.mdb с помощью запросов на выборку изменить порядок следования полей,

сделать невидимыми указанные поля, отсортировать записи по предложенному критерию, вывести на экран записи, отвечающие данным условиям.

1. Открыть файл базы данных ***Akademik.mdb.***



2. В списке **Объекты** выбрать **Запросы** и нажать кнопку **Создать.** В диалоговом окне **Новый запрос** выбрать вариант **Конструктор.**

3. В диалоговом окне **Добавление таблицы** указать на таблицу ***Академики*** и нажать кнопку **Добавить,** а затем кнопку **Закрыть.**

4. Выполнить отбор полей. Для этого из списка полей ***Академики*** двойным щелчком разместить поля на бланке запроса в строке ***Поле.***

5. На панели инструментов **Конструктор запросов** нажать кнопку **Вид-Режим таблицы.**

6. Вернуться в режим конструктора.

7. Установить критерии отбора для вывода на экран полей: ***ФИО, Специализация, Год присвоения звания.*** Для этого на бланке в строке

8. ***Вывод на экран*** отключить флажки для полей: ***Дата рождения, Пол.***

9. Перейти в режим таблицы. Убедиться, что видимыми остались только отобранные поля.

10.Вернуться в режим конструктора.

10. Изменить порядок следования столбцов, переместив столбец ***Пол*** на второе место. Для этого выделить столбец ***Пол*** и перетащить его в нужное место.

11. Вернуть видимость всем полям таблицы.

12. Отобрать записи, которые содержат информацию об академиках-женщинах. Для этого в условии отбора в поле ***Пол*** ввести ***ж.***

13. Упорядочить по возрастанию отобранные данные по значению поля ***Дата рождения.*** Для этого на бланке запроса в строке ***Сортировка*** в поле ***Дата рождения*** выбрать способ сортировки ***По возрастанию.***

14. Просмотреть отобранные записи. Убедиться, что записи расположены в возрастающем порядке значений поля ***Дата рождения.***

15. Создать запросы для вывода на экран записей, отображающих информацию:

- ***об академиках-физиках;***

- об академиках-физиках и химиках;

- об ученых, получивших звание академика до 1970 года;

об академиках-философах и языковедах;

- обо всех академиках, кроме физиков.

Отчет

Отчет должен быть оформлен в текстовом редакторе и содержать:

- наименование работы;
- цель работы;
- задание;
- последовательность выполнения работы;
- ответы на контрольные вопросы;

вывод о проделанной работе

Практическая работа №17

Поиск, пересылка информации в локальной сети

Цель:

- изучение и освоение на практике технологии обмена файлами между пользователями локальной компьютерной сети.

Содержание

Часть 1. Теоретическая

Передача информации между компьютерами существует с самого момента возникновения ЭВМ. Она позволяет организовать совместную работу отдельных компьютеров, решать одну задачу с помощью нескольких компьютеров, совместно использовать ресурсы и решать множество других проблем.

Под ***компьютерной сетью*** понимают комплекс аппаратных и программных средств, предназначенных для обмена информацией и доступа пользователей к единым ресурсам сети.

Основное назначение компьютерных сетей - обеспечить совместный доступ пользователей к информации (базам данных, документам и т.д.) и ресурсам (жесткие диски, принтеры, накопители CD-ROM, модемы, выход в глобальную сеть и т.д.).

Абоненты сети – объекты, генерирующие или потребляющие информацию.

Абонентами сети могут быть отдельные ЭВМ, промышленные роботы, станки с ЧПУ (станки с числовым программным управлением) и т.д. Любой абонент сети подключён к станции.

Станция – аппаратура, которая выполняет функции, связанные с передачей и приёмом информации.

Для организации взаимодействия абонентов и станции необходима физическая передающая среда.

Физическая передающая среда – линии связи или пространство, в котором распространяются электрические сигналы, и аппаратура передачи данных.

Одной из основных характеристик линий или каналов связи является скорость передачи данных (пропускная способность).

Скорость передачи данных **–** количество бит информации, передаваемой за единицу времени.

Обычно скорость передачи данных измеряется в битах в секунду (бит/с) и кратных единицах Кбит/с и Мбит/с.

Соотношения между единицами измерения: 1 Кбит/с =1024 бит/с; 1 Мбит/с =1024 Кбит/с; 1 Гбит/с =1024 Мбит/с.

На базе физической передающей среды строится коммуникационная сеть. Таким образом, компьютерная сеть – это совокупность абонентских систем и коммуникационной сети.

Различают два **вида** компьютерных сетей:

одноранговые и

клиент-сервер (многоранговые)

В одноранговой сети все PC равноправны: нет иерархии среди PC и нет выделенного (dedicated) сервера. Обычно каждый PC функционирует и как *клиент*, и как *сервер* - иначе говоря, нет PC, ответственного за всю сеть.

Если к одноранговой сети, где PC выступают в роли и клиентов и серверов, подключить более 10 PC, то она может не справиться с объемом работы. Поэтому большинство сетей имеют другую конфигурацию - они работают на основе выделенного сервера.

По типу сети делят на **локальные** и **глобальные** в данной классификации выделяют также региональные и мобильные сети.

Основные *компоненты коммуникационной сети:*

- передатчик;
- приёмник;
- сообщения (цифровые данные определённого формата: файл базы данных, таблица, ответ на запрос, текст или изображение);
- средства передачи (физическая передающая среда и специальная аппаратура, обеспечивающая передачу информации).

Топология локальных сетей.Под топологией компьютерной сети обычно понимают физическое расположение компьютеров сети относительно друг друга и способ соединения их линиями.

Топология определяет требования к оборудованию, тип используемого кабеля, методы управления обменом, надежность работы, возможность расширения сети. **Существует три основных вида топологии сети: шина, звезда и кольцо.**

Шина (bus), при которой все компьютеры параллельно подключаются к одной линии связи, и информация от каждого компьютера одновременно передается ко всем остальным компьютерам. Согласно этой топологии создается одноранговая сеть. При таком соединении компьютеры могут передавать информацию только по очереди, так как линия связи единственная.

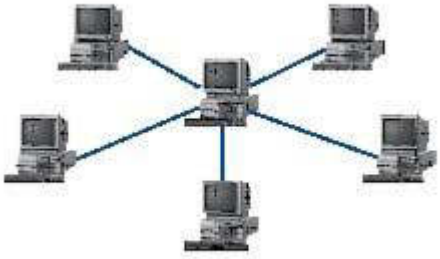
Достоинства:

- простота добавления новых узлов в сеть (это возможно даже во время работы сети);

- сеть продолжает функционировать, даже если отдельные компьютеры вышли из строя;
- недорогое сетевое оборудование за счет широкого распространения такой топологии.

Недостатки:

- сложность сетевого оборудования;
- сложность диагностики неисправности сетевого оборудования из-за того, что все адаптеры включены параллельно;
- обрыв кабеля влечет за собой выход из строя всей сети;
- ограничение на максимальную длину линий связи из-за того, что сигналы при передаче ослабляются и никак не восстанавливаются.



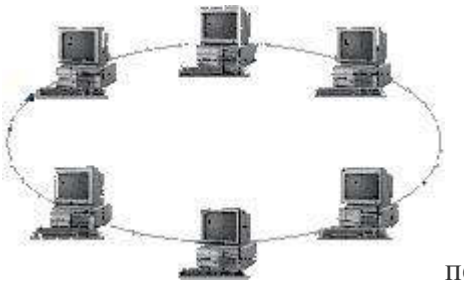
Звезда (star), при которой к одному центральному компьютеру присоединяются остальные периферийные компьютеры, причем каждый из них использует свою отдельную линию связи. Весь обмен информацией идет исключительно через центральный компьютер, на который ложится очень большая нагрузка, поэтому он предназначен только для обслуживания сети.

Достоинства:

- выход из строя периферийного компьютера никак не отражается на функционировании оставшейся части сети;
- простота используемого сетевого оборудования;
- все точки подключения собраны в одном месте, что позволяет легко контролировать работу сети, локализовать неисправности сети путем отключения от центра тех или иных периферийных устройств;
- не происходит затухания сигналов.

Недостатки:

- выход из строя центрального компьютера делает сеть полностью неработоспособной;
- жесткое ограничение количества периферийных компьютеров;
- значительный расход кабеля.



Кольцо (ring), при котором каждый компьютер передает информацию всегда только одному компьютеру, следующему в цепочке, а получает информацию только от предыдущего в цепочке компьютера, и эта цепочка замкнута. Особенностью кольца является то, что каждый компьютер восстанавливает приходящий к нему сигнал, поэтому затухание сигнала во всем кольце не имеет никакого значения, важно только затухание между соседними компьютерами.

Достоинства:

- легко подключить новые узлы, хотя для этого нужно приостановить работу сети;
- большое количество узлов, которое можно подключить к сети (более 1000);
- высокая устойчивость к перегрузкам.

Недостатки:

- выход из строя хотя бы одного компьютера нарушает работу сети;
- обрыв кабеля хотя бы в одном месте нарушает работу сети.

В отдельных случаях при конструировании сети используют комбинированную топологию. Например, **дерево** (tree)– комбинация нескольких звезд.

Каждый компьютер, который функционирует в локальной сети, должен иметь **сетевой адаптер (сетевую карту).** Функцией сетевого адаптера является передача и прием сигналов, распространяемых по кабелям связи. Кроме того, компьютер должен быть оснащен сетевой операционной системой.

При конструировании сетей используют следующие виды кабелей:

*неэкранированная витая пара.*Максимальное расстояние, на котором могут быть расположены компьютеры, соединенные этим кабелем, достигает 90 м. Скорость передачи информации - от 10 до 155 Мбит/с; *экранированная витая пара.* Скорость передачи информации - 16 Мбит/сна расстояние до 300 м.

коаксиальный кабель. Отличается более высокой механической прочностью, помехозащищённостью и позволяет передавать информацию на расстояние до 2000 м со скоростью 2-44 Мбит/с;

волоконно-оптический кабель. Идеальная передающая среда, он не подвержен действию электромагнитных полей, позволяет передавать информацию на расстояние до 10 000 м со скоростью до 10 Гбит/с.

Часть 2. Практическая

Задание №1. Ответить на вопросы:

1. Укажите основное назначение компьютерной сети.

2. Напишите, какие преимущества дает пользователю работа в сети?

3. Назовите объект, который является абонентом сети.

4. Назовите основную характеристику каналов связи.

5. Что такое локальная сеть, глобальная сеть?

6. Что понимается под топологией локальной сети?

7. Какие существуют виды топологии локальной сети?

8. Охарактеризуйте кратко топологию «шина», «звезда», «кольцо».

Решите задачу. Максимальная скорость передачи данных в локальной сети 100 Мбит/с. Сколько страниц текста можно передать за 1 сек, если 1 страница текста содержит 50 строк и на каждой строке - 70 символов

Отчет

Отчет должен быть оформлен в текстовом редакторе и содержать:

- наименование работы;
- цель работы;
- задание;
- последовательность выполнения работы;
- ответы на контрольные вопросы;

вывод о проделанной работе

Практическая работа №18
Создание электронной почты

Цели:

1. Формирование основных приёмов создания почтового ящика. Адресация в Интернете.
2. Развитие способностей планирования последовательности своих действий
3. Воспитание потребности осуществлять самоконтроль учебного процесса

Материально техническое и методическое обеспечение

ПК, Интернет, методические указания по практической работе.

Краткие сведения из теории

Электронная почта – (самая распространенная услуга сети Internet) обмен письмами в компьютерных сетях. Само письмо представляет собой обычный файл, содержащий текст письма и специальный заголовок, в котором указано, от кого письмо направлено, кому предназначено, какая тема письма и дата отправления.

Электронно-почтовый Internet-адрес имеет следующий формат: пользователь@машина

Пример адреса электронной почты: Ivanov@softpro.saratov.ru

Ivanov – имя почтового ящика.

softpro.saratov – название почтового сервера

ru – код Российской Федерации

Порядок выполнения работы (занятия)

Задание 1. Создание электронного почтового адреса на сервере www. mail. ru.

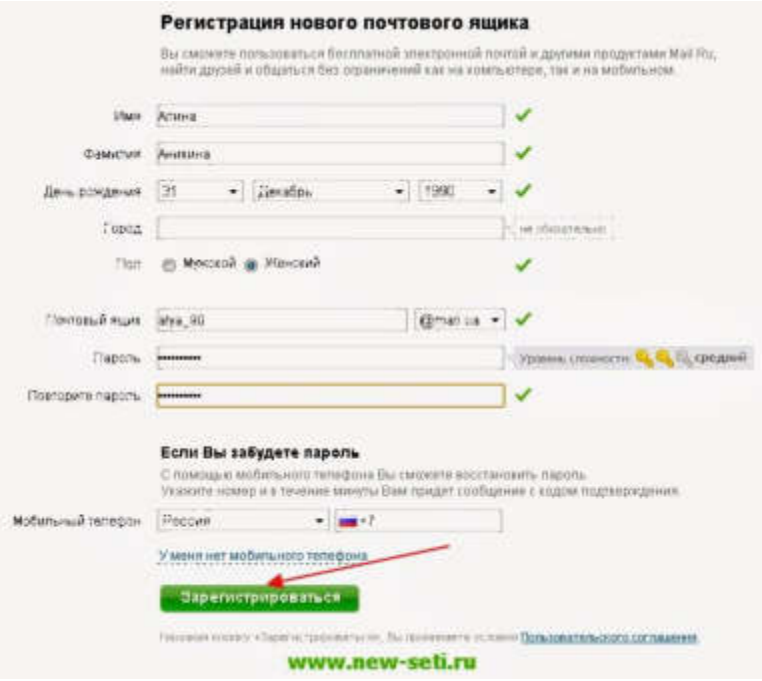
1. Установить соединение с Интернет.
2. «Зайти» на сайт почтового сервера www. mail. ru.
3. В разделе «Почта» щелкнуть ссылку «Регистрация»

4. Заполнить анкету, в которой нужно придумать имя почтового ящика, ввести свои персональные данные, пароль доступа к ящику и т. д.

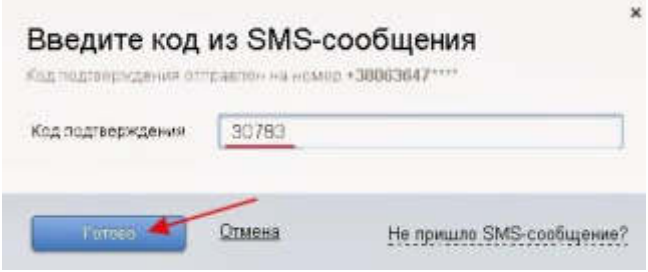
После нажатия вы попадаете на регистрационную форму, которую надо обязательно заполнить. Вот названия пунктов для регистрационного заполнения:

1. Ваше имя.
2. Фамилия, ваша же.
3. День рождения.
4. Город – эта графа необязательна к заполнению, решать вам.
5. Указываете свой пол, всего два варианта.
6. Вводите желаемое название почтового ящика.
7. И, на этом сервисе, конечно же, никак вам не обойтись без пароля.

Можно еще указать и свой мобильный телефон, но если вы этого не желаете, тогда перейдите по надписи «у меня нет / мобильного телефона». Но знайте, что телефон сможет вам помочь восстановить вашу почту, при утере или взломе.



И если вы подтверждаете регистрацию своего mail ru почтового ящика при помощи мобильного, тогда вы увидите вот такое вот окошко, в которое надо будет ввести код (он состоит из пяти цифр), пришедший на ваш телефон, и нажать на кнопку «Готово».



5. В ходе регистрации часто случается, что имя, выбранное вами для ящика, уже используется на этом сервере. В этом случае нужно выбрать из предложенных имён или придумать самому другое имя.
6. Внимательно читайте инструкции мастера и старайтесь следовать им.
7. После того, как ящик создан, рекомендуется записать в сохранном месте его адрес и пароль.

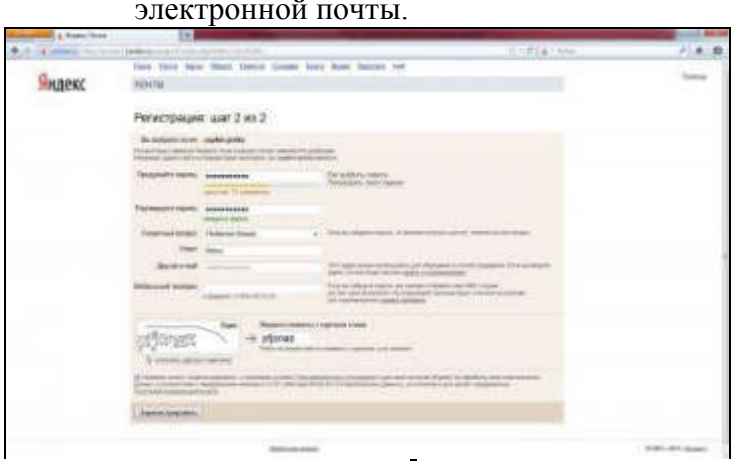
Задание 2. Создание электронного почтового адреса на Яндексе

Чтобы создать **бесплатный почтовый ящик на яндексе** необходимо зайти на сайт *yandex.ru* и нажать пункт меню *«Завести почтовый ящик»*.

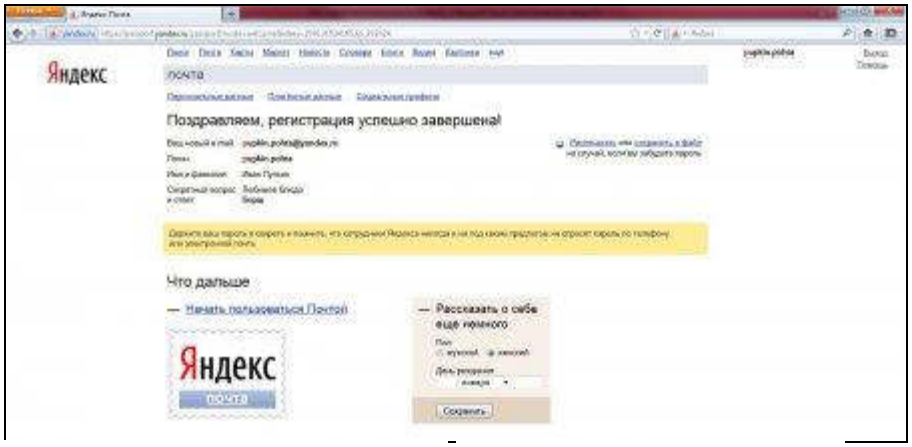
Также компания разработала возможность зайти на **почтовый ящик yandex** через существующие у Вас аккаунты популярных социальных сетей и почтовых сервисов.

При заполнении формы регистрации каждый столкнется с проблемой подбора логина. Это связано с тем, что *электронная почта на яндексе* пользуется большой популярностью и практически все простые логины уже заняты.

Для примера просто выберем один из предлагаемых вариантов и продолжим регистрацию электронной почты.



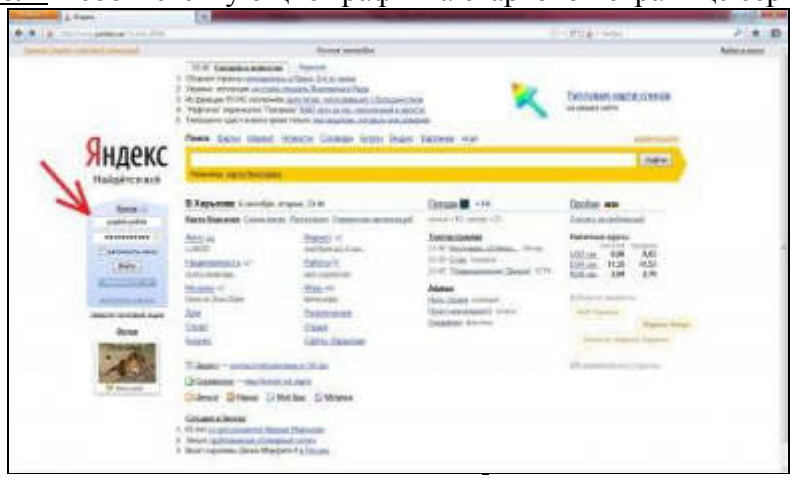
После заполнения всех обязательных полей Ваша **электронная почта на Яндексе** готова.



Теперь, чтобы начать использовать Ваш **почтовый ящик в яндексе**, нажимаем *«Начать пользоваться Почтой»*.

При необходимости войти в свою электронную почту, нужно

ввести логин и пароль в соответствующие графы на стартовой странице сервиса.



Или ввести данные на страничке, пройдя по ссылке *«Войти в почту»*, где также будет предложена возможность **создать почтовый ящик в яндексе**.

Задание 3. Создание и отправление электронного письма

Отправьте письмо по указанному адресу:

Ильиных Ольге Алексеевне – ilyinh@mail. ru

Перфиловой Ирине Александровне – irina_270387@mail. Ru

- Найдите ссылку **Написать письмо** и щелкните на ней.
- В поле Кому: введите адрес. В поле **Тема**: напишите свою фамилию и инициалы.
- Для того, что бы прикрепить фото или другого типа файл:

· При написании письма воспользуйся кнопкой ***Обзор.***

· В открывшемся окне выбери нужный архив и нажми кнопку ***Открыть.***

· Далее нажми кнопку ***Прикрепить.***

· Письмо с прикреплённым файлом можно отсылать. Щелкните на кнопке **Отправить**. Если сообщение отправлено вы увидите на экране подтверждение: «**Ваше сообщение для … отправлено**».

Задание 4. Удаление почтового ящика

Удалить почтовый ящик может только пользователь, имеющий к нему доступ. Для того чтобы удалить ящик, воспользуйтесь специальным интерфейсом. Вам будет необходимо ввести имя ящика и верный пароль к нему, а также указать причину, повлекшую удаление ящика.

Удаляя свой почтовый ящик, вы также удаляете всю сопутствующую информацию на других проектах — персональную страницу на проекте МойМир@Mail.Ru, фотографии с проекта Фото@Mail.Ru и пр.

Обратите внимание, что в течение трех месяцев с момента удаления, вы можете восстановить почтовый ящик, но вы не сможете восстановить информацию, хранящуюся в почтовом ящике, а также информацию с сопутствующих проектов.

Контрольные вопросы

1. Технология создания почтового ящика?
2. Что такое логин?
3. Как можно удалить почтовый ящик?
4. Адресация в системе электронной почты?

Отчет

Отчет должен быть оформлен в текстовом редакторе и содержать:

- наименование работы;
- цель работы;
- задание;
- последовательность выполнения работы;
- ответы на контрольные вопросы;

вывод о проделанной работе

Методические рекомендации по выполнению

Лабораторных работ

для обучающихся

по ОП 02 «Основы электротехники»

по программам подготовки квалифицированных рабочих,

служащих по профессии

09.01.03 (230103.02) Мастер по обработке цифровой информации

г. Балашиха

Пояснительная записка

Цель лабораторных работ - помочь обучающимся систематизировать, закрепить и углубить знания теоретического характера; научить студентов приемам решения лабораторных задач; научить их работать с книгой, нормативно-правовой и распорядительной документацией, пользоваться справочной и научной литературой; формировать умение учиться самостоятельно, т. е. овладевать методами, способами и приемами самообучения, саморазвития и самоконтроля.

Лабораторные работы направлены на овладение студентами умений решения стандартных задач и приобретение навыков практических действий.

Основные требования к содержанию лабораторных работ:

-соответствие содержания практических заданий изученному теоретическому материалу учебной дисциплины; максимальное приближение содержания практических заданий к реальной действительности; поэтапное формирование умения, т.е. движение от знания к умению, от простого умения к сложному и т.д.; использование типовых документов, инструкций, бланков и т.п.

-использование последних версий программного обеспечения.

Результатом освоения программы учебной дисциплины ОП 02 Основы электротехники», является овладение студентами видом профессиональной деятельности, в том числе результатом освоения программы профессионального дисциплины является овладение студентами общими (ОК) и профессиональными (ПК)компетенциями:

ПК 1.1.	Подготавливать к работе и настраивать аппаратное обеспечение, периферийные устройства, операционную систему персонального компьютера и мультимедийное оборудование
ПК 1.2.	Выполнять ввод цифровой и аналоговой информации в персональный компьютер с различных носителей.
ПК 1.3.	Конвертировать файлы с цифровой информацией в различные форматы.
ПК 1.4.	Обрабатывать аудио и визуальный контент средствами звуковых, графических и видео-редакторов.
ПК 1.5.	Создавать и воспроизводить видеоролики, презентации, слайд-шоу, медиафайлы и другую итоговую продукцию из исходных аудио, визуальных и мультимедийных компонентов средствами персонального компьютера и мультимедийного оборудования.
ОК 1	Понимать сущность и социальную значимость своей будущей профессии,

	проявлять к ней устойчивый интерес.
ОК 2	Организовывать собственную деятельность, исходя из цели и способов ее достижения, определенных руководителем.
ОК 3	Анализировать рабочую ситуацию, осуществлять текущий и итоговый контроль, оценку и коррекцию собственной деятельности, нести ответственность за результаты своей работы.
ОК 4	Осуществлять поиск информации, необходимой для эффективного выполнения профессиональных задач.
ОК 5	Использовать информационно-коммуникационные технологии в профессиональной деятельности.
ОК 6	Работать в команде, эффективно общаться с коллегами, руководством, клиентами.
ОК 7	Исполнять воинскую обязанность, в том числе с применением полученных профессиональных знаний (для юношей).

Уметь:

У1 эксплуатировать электроизмерительные приборы

У2 контролировать качество выполняемых работ;

У3 производить контроль различных параметров электрических приборов

У4 работать с технической документацией

Знать

З.1 Основные законы электротехники: электрическое поле, электрические цепи постоянного тока, физические процессы в электрических цепях постоянного тока

З.2 Расчет электрических цепей постоянного тока

З.3 Магнитное поле, магнитные цепи

З.4 Магнитное поле, магнитные цепи

З. 5 Электромагнитная индукция, электрические цепи переменного тока

З.6 Основные сведения о синусоидальном электрическом токе, линейные электрические цепи синусоидального тока;

З.7 Общие сведения об электросвязи и радиосвязи

З.8 Основные сведения об электроизмерительных приборах, электрических машинах, аппаратуре управления и защиты;

Методические рекомендации по подготовке к лабораторным занятиям

Выполнению лабораторной работы предшествует проверка знаний студентов - их теоретической готовности к выполнению задания.

Лабораторная работа предполагает проявление активности и самостоятельности студента при выполнении заданий.

- Прослушайте внимательно инструктаж преподавателя
- Продумайте этапы выполнения задания, распределите время, отберите необходимые обеспечивающие средства.

Подготовка к проведению лабораторных работ предполагает владение студентами умениями и навыками умственного труда и информационной культуры:

- составлять перечни книг, систематизировать их,
- делать выписки,
- осуществлять различные виды чтения,
- проводить аналитико-синтетическую переработку информации
- Систематизировать и оформлять записи в виде аннотаций, тезисов, цитат, конспектов.

Подготавливая технические средства и оборудования для выполнения лабораторных работ, следует строго следовать указаниям преподавателя, а также следить за последовательностью деятельности, прописанной в лабораторной работе.

Для обеспечения безопасности работы с персональным компьютером необходимо:

перед началом работы убедиться в исправности электропроводки, выключателей, штепсельных розеток, при помощи которых оборудование включается в сеть, наличии заземления компьютера, его работоспособности.

Оценка индивидуальных образовательных достижений по результатам текущего контроля производится в соответствии с универсальной шкалой (таблица).

Процент результативности (правильных ответов)	**Качественная оценка индивидуальных образовательных достижений**	
	балл (отметка)	**вербальный аналог**
90 ÷ 100	5	отлично
80 ÷ 89	4	хорошо
70 ÷ 79	3	удовлетворительно
менее 70	2	не удовлетворительно

Критерии оценки лабораторной работы

При проверке усвоения материала выявляется полнота, прочность усвоения студентом теории и умение применять ее на практике в знакомых и незнакомых ситуациях.

Отметка зависит также от наличия и характера погрешностей, допущенных студентами.

• ***грубая ошибка*** – полностью искажено смысловое значение понятия, определения;

• ***погрешность*** отражает неточные формулировки, свидетельствующие о нечетком представлении рассматриваемого объекта;

• ***недочет*** – неправильное представление об объекте, не влияющего кардинально на знания, определенные программой обучения;

• ***мелкие погрешности*** – неточности в устной и письменной речи, не искажающие смысла ответа или решения, случайные описки и т.п.

Исходя из норм (пятибалльной системы), заложенных во всех предметных областях выставляете отметка:

«5» ставится при выполнении всех заданий полностью или при наличии 1-2 мелких погрешностей;

«4» ставится при наличии 1-2 недочетов или одной ошибки:

«3» ставится при выполнении 2/3 от объема предложенных заданий;

«2» ставится, если допущены существенные ошибки, показавшие, что студент не владеет обязательными умениями поданной теме в полной мере (незнание основного программного материала):

«1» – отказ от выполнения учебного задания

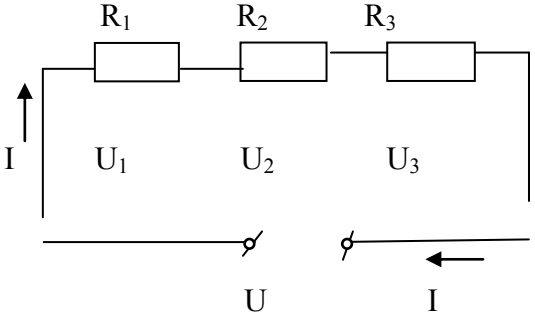
ЛАБОРАТОРНАЯ РАБОТА № 1

Тема : Расчет эквивалентного сопротивления в сложной резистивной схеме. Расчеты эквивалентной емкости и предельного напряжения на ней в сложной конденсаторной схеме.

Цель работы: Определить общий ток и токи в ветвях при смешанном соединении приемников электрической энергии. Определить эквивалентное сопротивление. Проверить баланс мощности рассматриваемой цепи.

Пояснения к работе.

Неразветвленная электрическая цепь это последовательное соединение приемников электрической энергии.



Последовательным называется такое соединение приемников электрической энергии, при котором по всем элементам протекает один и тот же ток.

$\mathbf{U = U_1 + U_2 + U_3}$

Эквивалентное сопротивление цепи равно сумме сопротивлений последовательно включенных резисторов : $\mathbf{R_{экв} = R_1 + R_2 + R_3}$

Эквивалентным называется такое сопротивление, которое будучи включенным вместо данных резисторов, не изменяет режима работы электрической цепи.

Закон Ома для всей замкнутой цепи имеет вид:

$\mathbf{I = U/ (R_1 + R_2 + R_3)}$ $\quad$ $\mathbf{I = U/R_{экв}}$

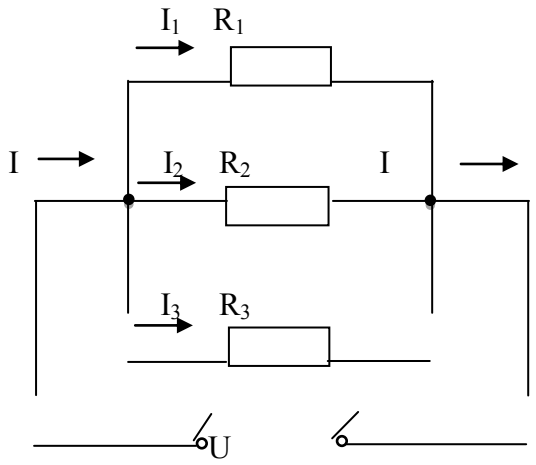
$\mathbf{P = P_1 + P_2 + P_3}$ – уравнение баланса мощностей.

Общая мощность равна сумме мощностей последовательно включенных резисторов.

Мощности на последовательно включенных резисторах распределяются прямо пропорционально сопротивлениям резисторов.

Напряжение на последовательно включенных резисторах распределяется прямо пропорционально сопротивлениям резисторов.

Разветвленная электрическая цепь это параллельное соединение приемников электрической энергии.



Параллельным называется такое соединение приемников электрической энергии, при котором на зажимах всех элементов имеется одно и то же напряжение.

$\mathbf{U = U_1 = U_2 = U_3}$

Согласно первого закона Кирхгофа: $\mathbf{I = I_1 + I_2 + I_3}$

Обратная величина эквивалентного сопротивления равна сумме обратных величин сопротивлений резисторов, включенных параллельно: $\mathbf{1/R_{экв} = 1/R_1 + 1/R_2 + 1/R_3}$

Величина обратная сопротивлению является проводимостью.

$\mathbf{G_{экв} = 1/R_{экв}}$ $\mathbf{G_{экв} = G_1 + G_2 + G_3}$

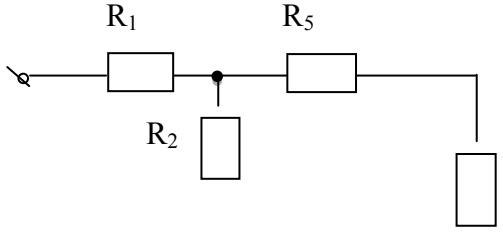
Эквивалентное сопротивление двух резисторов, включенных параллельно, определяется по формуле: $\mathbf{R_{экв} = R_1 \cdot R_2 / (R_1 + R_2)}$

Смешанное соединение – это такое соединение, при котором в электрической схеме имеются одновременно участки с последовательно и параллельно включенными элементами. К этим участкам применяются формулы последовательного и параллельного соединения приемников электрической энергии, а данный метод называется методом эквивалентного сопротивления или методом «свертывания».

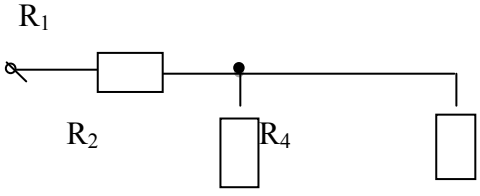
Задание:

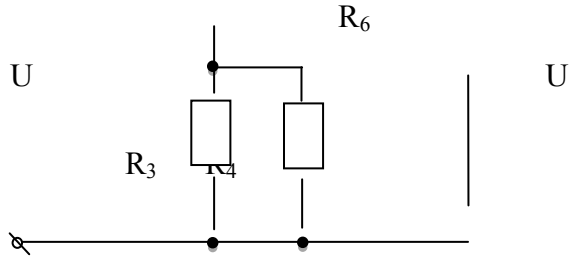
1. В практической работе необходимо определить общий ток и токи в ветвях при смешанном соединении приемников электрической энергии. Определить эквивалентное сопротивление. Проверить баланс мощности рассматриваемой цепи.
2. Начертить принципиальную схему своего варианта.

варианты № 4, 8, 12, 16

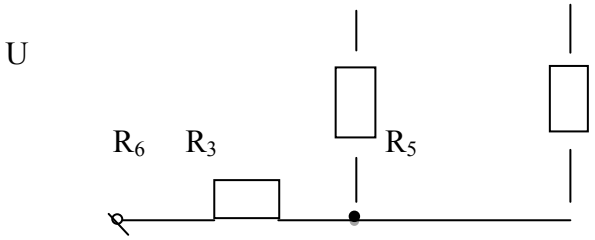


варианты № 3, 7, 11, 15

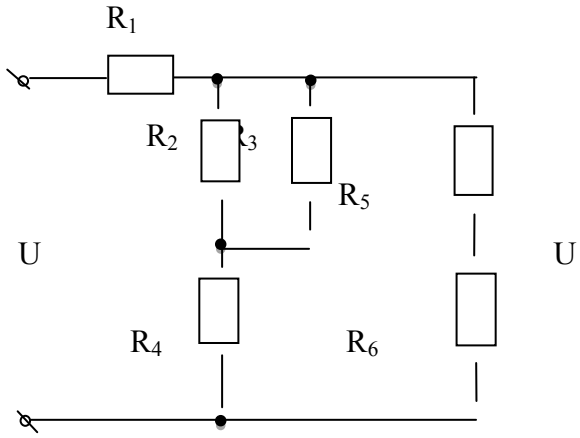




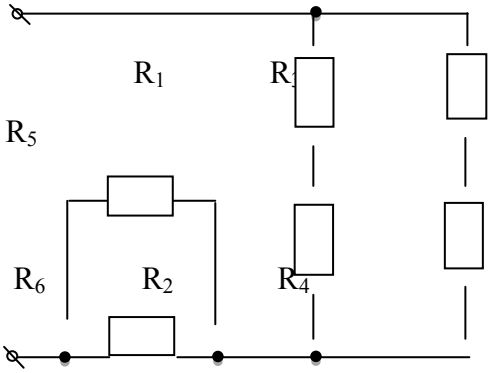
варианты № 1, 5, 9, 13



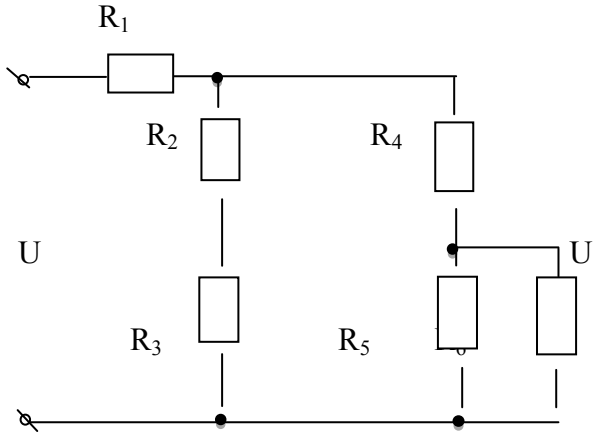
варианты № 2, 6, 10, 14



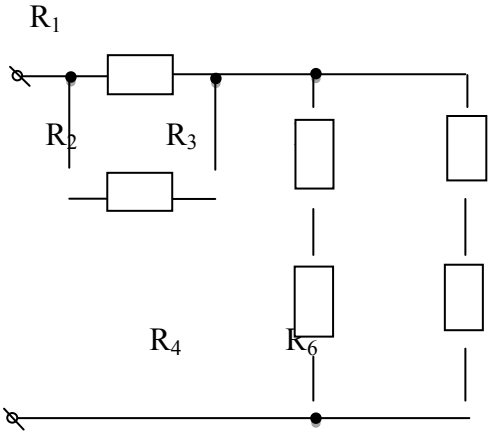
варианты № 20, 24, 28, 32



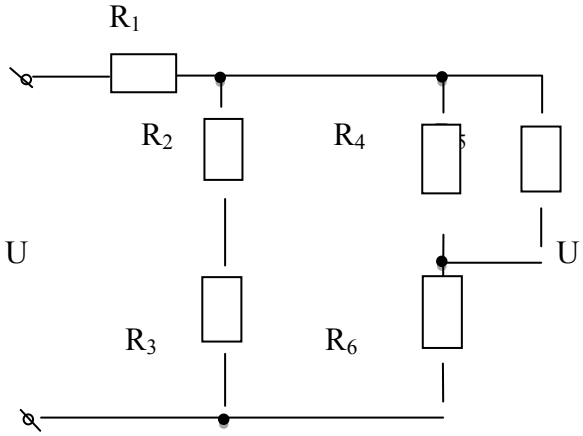
варианты № 19, 23, 27, 31

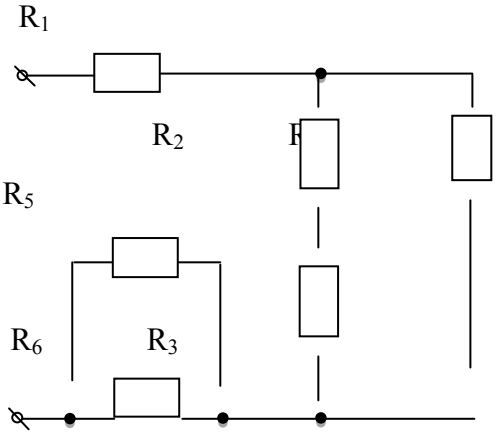


варианты № 17, 21, 25, 29



варианты № 18, 22, 26, 30





3. Переписать из таблицы данные своего варианта.

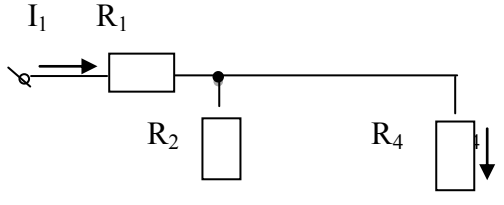
4. Выполнить расчет, в соответствии с предлагаемым методом, применяя следующие формулы: последовательное соединение - $\mathbf{U = U_1 + U_2 + U_3}$; $\mathbf{R_{экв} = R_1 + R_2 + R_3}$; $\mathbf{P = P_1 + P_2 + P_3}$; $\mathbf{I = U/R_{экв}}$; параллельное соединение $\mathbf{-U = U_1 = U_2 = U_3}$; $\mathbf{I = I_1 + I_2 + I_3}$;

$\mathbf{1/R_{экв} = 1/R_1 + 1/R_2 + 1/R_3}$; $\mathbf{G_{экв} = 1/R_{экв}}$; $\mathbf{G_{экв} = G_1 + G_2 + G_3}$; $\mathbf{R_{экв} = R_1R_2/(R_1 + R_2)}$.

5. Исходные данные.

ИСХОДНЫЕ ДАННЫЕ															
№ варианта	$\mathbf{I_3}$	$\mathbf{I_4}$	$\mathbf{U_1}$	$\mathbf{U_2}$	$\mathbf{U_4}$	$\mathbf{U_6}$	$\mathbf{P_2}$	$\mathbf{P_5}$	$\mathbf{R_1}$	$\mathbf{R_2}$	$\mathbf{R_3}$	$\mathbf{R_4}$	$\mathbf{R_5}$	$\mathbf{R_6}$	**Определить все токи цепи, $R_{экв}$, U. Составить баланс мощностей.**
	А	**А**	**В**	**В**	**В**	**В**	**Вт**	**Вт**	**Ом**	**Ом**	**Ом**	**Ом**	**Ом**	**Ом**	
1,2,3,4	0,3								10	20	80	24	50	70	
5,6,7,8								10	12	150	100	40	10	50	
9,10,11, 12					18				10	60	40	36	20	10	
13,14,15,16			20						20	20	60	35	25	50	
17,18,19,20	0,1								20	300	100	300	100	25	
21,22,23,24							0,4		14	10	80	50	75	30	
25,26,27,28						24			10	15	10	30	60	80	
29,30,31,32				9					5	15	5	80	120	12	

<u>Образец решения задачи (вариант №32).</u>

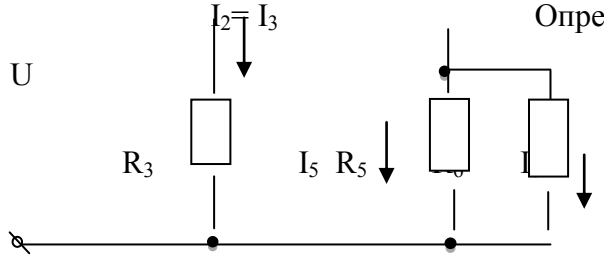


Дано: $R_1 = 5$ Ом; $R_2 = 15$ Ом; $R_3 = 5$ Ом;

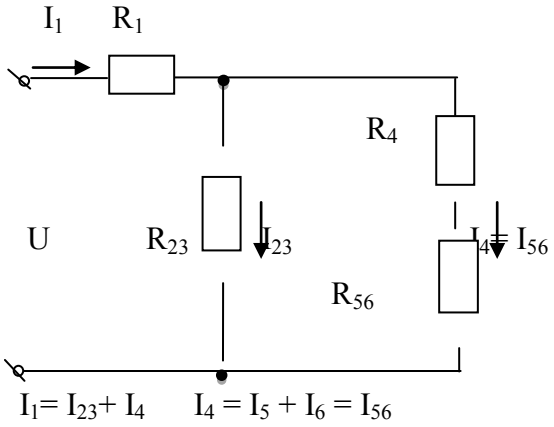
$R_4 = 80$ Ом; $R_5 = 120$ Ом; $R_6 = 12$ Ом;

$U_2 = 9$ В.

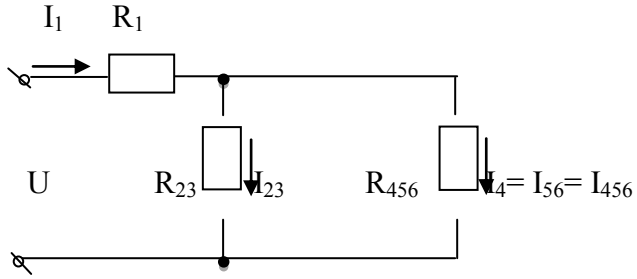
Определить: I_1, $I_2 = I_3$, I_4, I_5, I_6, U, $R_{экв}$.



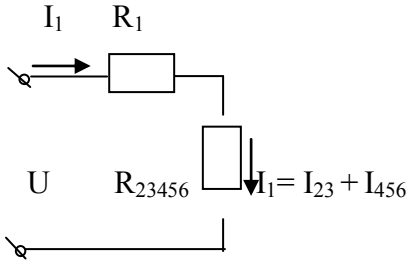
1. $R_{56} = R_5 \cdot R_6 / (R_5 + R_6) = 120 \cdot 12 / (120 + 12) = 10{,}9$ Ом
2. $R_{23} = R_2 + R_3 = 15 + 5 = 20$ Ом



3. $R_{456} = R_4 + R_{56} = 80 + 10{,}9 = 90{,}9$ Ом



4. $R_{23456} = R_{23} \cdot R_{456} / (R_{23} + R_{456}) = 20 \cdot 90{,}9 / (20 + 90{,}9) = 16{,}4$ Ом



5. $R_{экв} = R_1 + R_{23456} = 5 + 16{,}4 =$ **21,4 Ом**
6. $I_2 = U_2 / R_2 = 9/15 = 0{,}6$ А $I_3 = I_2 =$ **0,6 А**
7. $U_3 = I_3 \cdot R_3 = 0{,}6 \cdot 5 = 3$ В
8. $U_{23} = U_2 + U_3 = 9 + 3 = 12$ В $U_{23} = U_{456} = 12$ В
9. $I_4 = U_{456} / R_{456} = 12 / 90{,}9 =$ **0,13 А**
10. $U_4 = I_4 \cdot R_4 = 0{,}13 \cdot 80 = 10{,}6$ В
11. $U_{56} = U_{456} - U_4 = 12 - 10{,}6 = 1{,}4$ В

    $U_5 = U_6 = U_{56} = 1{,}4$ В

12. $I_5 = U_5 / R_5 = 1,4/120 =$ **0,01 А**

13. $I_6 = U_6 / R_6 = 1,4/ 12 =$ **0,12 А**

$I_4 = I_5 + I_6 = 0,01 + 0,12 = 0,13$ А

14. $I_1 = I_2 + I_4 = 0,6 + 0,13 =$ **0,73 А**

15. $U_1 = I_1 \cdot R_1 = 0,73 \cdot 5 = 3,7$ В

16. $U = U_1 + U_{23} = 3,7 + 12 = 15,7$ В

Проверить баланс мощности:

$P_1 = U_1 \cdot I_1 = 3,7 \cdot 0,73 = 2,7$ Вт;

$P_2 = U_2 \cdot I_2 = 9 \cdot 0,6 = 5,4$ Вт;

$P_3 = U_3 \cdot I_3 = 3 \cdot 0,6 = 1,8$ Вт;

$P_4 = U_4 \cdot I_4 = 10,6 \cdot 0,13 = 1,4$ Вт;

$P_5 = U_5 \cdot I_5 = 1,4 \cdot 0,01 = 0,014$ Вт;

$P_6 = U_6 \cdot I_6 = 1,4 \cdot 0,12 = 0,168$ Вт;

$P = U \cdot I = 15,7 \cdot 0,73 = 11,46 \approx 11,5$ Вт

$P = P_1 + P_2 + P_3 + P_4 + P_5 + P_6$.

11,5 Вт = 2,7 + 5,4 + 1,8 + 1,4 + 0.01 + 0,17 = 11,48 ≈ 11,5 Вт.

11,5 Вт = 11,5 Вт.

Работа на занятии.

1. В соответствии с принципиальной схемой своего варианта, используя исходные данные, произвести расчет токов во всех ветвях цепи, определить эквивалентное сопротивление. Проверить баланс мощности рассматриваемой цепи.

2. При решении применить предлагаемые формулы и образец решения подобной задачи.

Содержание отчета.

1. Цель работы.
2. Принципиальная электрическая схема.
3. Исходные данные
4. Формулы, необходимые для расчета.
5. Решение задачи.
6. Вывод по работе.

Лабораторная работа №2

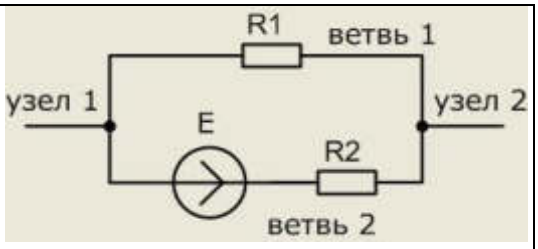
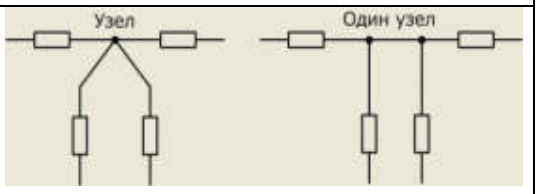
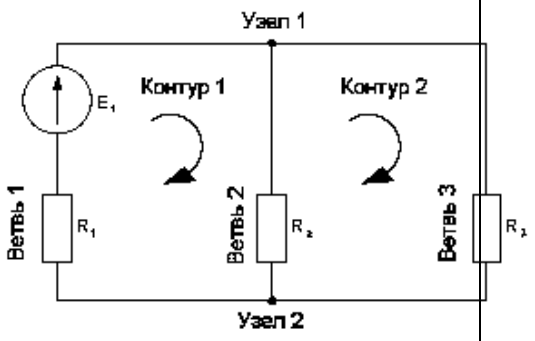
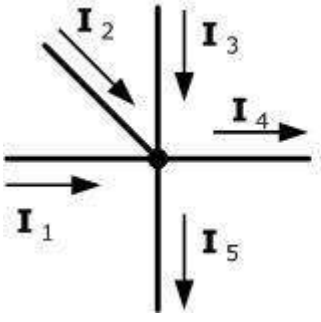
Тема Расчет разветвленной цепи с помощью уравнений Кирхгофа

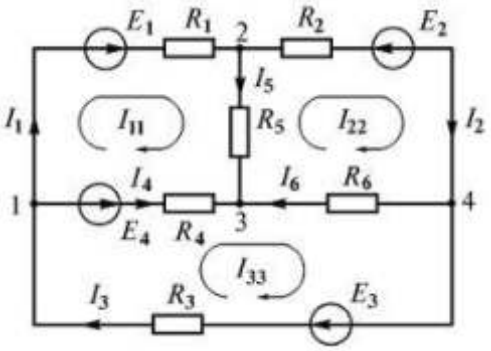
Теория

Зако́ны Кирхго́фа (правила Кирхгофа) — соотношения, которые выполняются между токами и напряжениями на участках любой электрической цепи. Правила Кирхгофа позволяют рассчитывать любые электрические цепи постоянного и квазистационарного тока и имеют особое значение в электротехнике из-за своей универсальности, так как пригодны для решения любых электротехнических задач. Применение правил Кирхгофа к цепи позволяет получить систему линейных уравнений относительно токов, и соответственно, найти значение токов на всех ветвях цепи. Сформулированы Густавом Кирхгофом в 1845 году.

Для расчета разветвленной сложной электрической цепи существенное значение имеет число ветвей и узлов.

Основные характеристики электрической цепи. Правила Кирхгофа

Ветвь электрической цепи	Участок, состоящий только из последовательно включенных источников ЭДС и приемников с одним и тем же током.	
Узел электрической цепи	Место или точка соединения трех и более ветвей	
Замкнутый контур	Замкнутый путь, проходящий по нескольким ветвям, при этом каждый узел в рассматриваемом контуре встречается не более одного раза.	
Первый закон Кирхгофа	Применяется к узлам и формулируется следующим образом: алгебраическая сумма токов в узле равна нулю. Знаки определяются в зависимости от того, направлен ток к узлу или от него. Токи, подходящие к узлу, считаются положительными,	 $I_1+I_2+I_3-I_4-I_5=0$ или $I_1+I_2+I_3=I_4+I_5$

	отходящие от узла-отрицательными	$\sum I = 0$
Второй закон Кирхгофа	В любом замкнутом контуре алгебраическая сумма напряжений на всех участках с сопротивлениями, входящих в этот контур, равно алгебраической сумме ЭДС.	$\sum R \cdot I = \sum E$
Правила применения второго закона Кирхгофа	1.Произвольно выбрать положительные направления токов во всех ветвях 2.Направление обхода контуров выбрать по часовой стрелке. 3.Записать первый закон Кирхгофа для токов произвольно выбранных узлов электрической цепи(токи, походящие к узлу, считаются положительными. Токи, отходящие от узла-отрицательными 4.Записать второй закон Кирхгофы для каждого замкнутого контура, учитывая. что а) ЭДС записывается со знаком «+», если направление обхода контура совпадает с направлением эдс, в противоположном случае ставится зак «-» б)Если ток, протекающий через данное сопротивление, совпадает с направлением обхода контура, то напряжение на этом сопротивлении берется со знаком “+”, в противном случае ставится знак “-”.	 $I_1 + I_4 - I_3 = 0;$ $E_1 - E_4 = I_1 R_1 + I_5 R_5 - I_4 R_4$ $I_2 + I_5 - I_1 = 0;$ $-E_2 = I_2 R_2 + I_6 R_6 - I_5 R_5;$ $I_4 + I_5 + I_6 = 0;$ $E_4 + E_3 = I_4 R_4 + I_3 R_3 - I_6 R_6.$

Баланс мощностей .	*Это выражение закона сохранения энергии, в электрической цепи. Сумма мощностей потребляемых приемниками, равна сумме мощностей отдаваемых источниками.*	$P_{источника} = P_{приёмника}$ $P_{источника} = E * I$ $U = I * R$ $P_{приёмника} = R * I^2$ $E * I = R * I^2$

Содержание отчета:

Цель работы : Экспериментально проверить справедливость законов Кирхгофа , научиться рассчитывать баланс мощностей в разветвленной электрической цепи.

Перечень необходимого оборудования:

Элементы схем: R_1, R_2 — реостаты с сопротивлениями от 10 до 50 Ом Электроизмерительные приборы: амперметр постоянного тока с пределом измерения 2А; вольтметр постоянного тока с пределом измерения 4 В.

Источники питания: источник постоянного нерегулируемого напряжения 4 В

Контрольные вопросы:

1. Какую роль играют законы Кирхгофа в электрической цепи?
2. Что называется ветвью?
3. Что называется контуром?
4. Сформулируйте 1 закон Кирхгофа
5. Какие токи считаются положительными?
6. Какие токи считаются отрицательными?
7. Сформулируйте 2 закон Кирхгофа
8. Запишите правила использования законов Кирхгофа для замкнутого контура
9. Как выбирают направление обхода контура?
10. Что называется балансом мощностей?

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1.Соберите электрическую цепь по схеме (рис. 1)

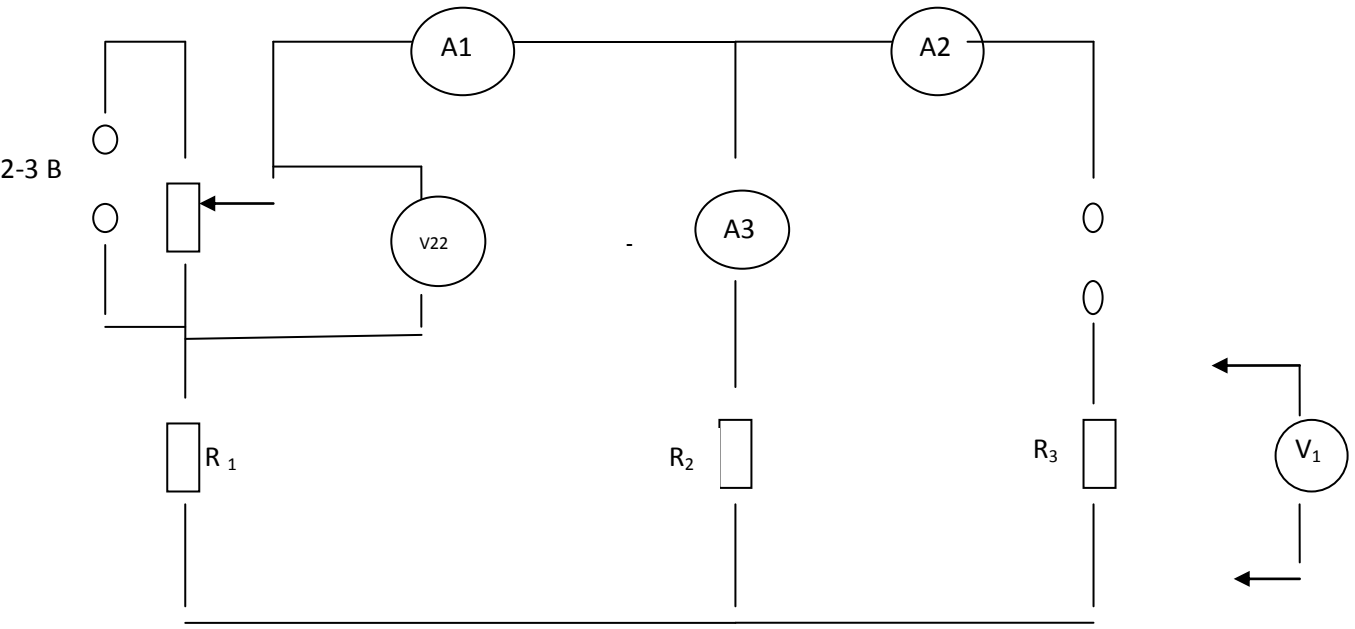


Рис 1.

2.Подключите, соблюдая полярность, вольтметр V_1 к клеммам источника 4 В (правый контур).

3.Включите блок питания левого контура и с помощью реостата установите напряжение 1-3 В . (левый контур). В процессе включения источников и установки нужных напряжений стрелки миллиамперметров могут отклоняться влево от нуля и за последнюю оцифрованную отметку шкалы.

4.После включения обоих источников и установки нужных напряжений стрелки всех приборов должны находится в пределах их диапазонов измерений.

5.С помощью амперметров измерьте токи I_2, I_3, I_4 на соответствующих резисторах, запишите показания приборов в таблицу.

6. С помощью переносных вольтметров измерьте напряжения я источников U_1, U_5..Запишите их показания в таблицу.

7.Измерить напряжение U_2, U_3, U_4 на резисторах R_2, R_3, R_4 соответственно, соблюдая полярность и поочередно подключая к клеммам резисторов. Результаты измерения запишите в таблицу

8.Выключите блок питания.

9. Заполните таблицу

Таблица

Из эксперимента							
U_1, B	U_2, B	U_3, B	U_4, B	U_5, B	I_2, B	I_3, B	I_4, B

Из расчета			
P_2	P_3	P_4	$P_2 + P_3 + P_4$

10. Запишите в общем виде и с числовыми результатами уравнение для токов по первому закону Кирхгофа

11. запишите в общем виде и с числовыми результатами уравнение второго закона Кирхгофа для левого и правого замкнутого контура

12. Рассчитайте мощность источников и мощность, выделяющуюся на резисторах:

мощность источника U_1

$P_1 = U_1 \times I_2, мВт$_____

мощность источника U_5

$P_5 = U_5 \times I_4, мВт$_____

мощность, выделяющаяся на резисторе R_2

$P_2 = U \times I_2, мВт$_____

мощность, выделяющаяся на резисторе R_3

$P_3 = U_3 \times I_3, мВт$_____

мощность, выделяющаяся на резисторе R_4

$P_4 = U_4 \times I_4, мВт$_____

В приведенных отношениях:

$U_1, U_5 -$ *напряжение соответствующих источников, В*;

$U_2, U_3, U_4 -$ *напряжение на резисторах* R_2, R_3, R_4 соответственно,

$I_2, I_3, I_4 -$ токи в ветвях цепи, А

13.Результаты запишите в таблицу

14.Запишите уравнение баланса мощностей в электрической цепи.

15.Сделайте вывод о подтверждении законов Кирхгофа в рассматриваемой электрической цепи и объясните причину неполного совпадения экспериментальных данных с теоретическими.

Содержание отчета.

1. Цель работы.
2. Принципиальная электрическая схема.
3. Исходные данные
4. Формулы, необходимые для расчета.
5. Решение задачи.

6. Вывод по работе

Лабораторная работа № 3

Тема: Расчет цепи с помощью метода узловых потенциалов

Цели:

1. Закрепление теоретических знаний по теме;
2. Формирование практических навыков расчета параметров конденсаторов и энергии электрического поля;
3. Формирование общих компетенций: ОК 2, ОК4.

Студент должен знать:

1. Назначение и основной параметр конденсаторов;
2. Основные расчетные формулы нахождения общей емкости при смешанном соединении конденсаторов.
3. Формулу расчета энергии электрического поля.

Студент должен уметь:

Рассчитывать основные параметры конденсаторов и сворачивать схемы со смешанным соединением, рассчитывать энергию поля.

Теоретическая часть: основные расчетные формулы

Электрическая емкость конденсатора:

$$C = \frac{Q}{U}(Ф) \qquad C = \frac{\varepsilon_0 \cdot \varepsilon_r \cdot S}{d}$$

1. Емкость параллельного соединения конденсаторов: $C_{общ} = C_1 + C_2$

2. Емкость последовательного соединения конденсаторов: $C_{общ} = \frac{C_1 \cdot C_2}{C_1 + C_2}$

3. Энергия электрического поля: $W_{эл} = \frac{C \cdot U^2}{2}(Дж)$

Расчетные задания:

1. Определить емкость конденсатора, если он был заряжен до напряжения 250В, при этом заряд составил $2 \cdot 10^{-4}$Кл.
2. Конденсатор заряжен от источника 100В. Энергия поля конденсатора $6 \cdot 10^{-3}$Дж. Определить его емкость.
3. Определить емкость плоского воздушного конденсатора, имеющего площадь пластин 20см^2, расстояние между ними 0,8см.

4. Три конденсатора емкостями 47пФ, 18пФ и 75пФ соединены параллельно и к ним параллельно подключен конденсатор в 75 пФ, найти общую емкость соединения.
5. Общая емкость двух последовательно включенных конденсаторов 1,2мкФ. Емкость одного из них 3мкФ, найти емкость другого.

Контрольные вопросы:

1. Дать понятие электрической емкости, указав единицы измерения.
2. Что такое конденсатор, для чего его используют в электрических цепях?
3. Как на схеме обозначается конденсатор?
4. Записать формулы последовательного и параллельного соединения конденсаторов и энергии электрического поля.

Содержание отчета.

1. Цель работы.
2. Принципиальная электрическая схема.
3. Исходные данные
4. Формулы, необходимые для расчета.
5. Решение задачи.
6. Вывод по работе

Лабораторная работа № 4

Тема: Составление уравнений состояния для расчета цепи методом контурных токов.

Цель работы: приобрести умения решения сложных цепей постоянного тока методом контурных токов.

Методика расчета цепи методом контурных токов

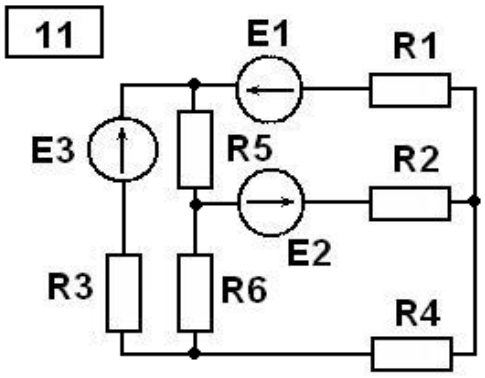
В методе контурных токов за неизвестные величины принимаются расчетные (контурные) токи, которые якобы протекают в каждом из независимых контуров. Таким образом, количество неизвестных токов и уравнений в системе равно числу независимых контуров цепи.

Расчет токов ветвей по методу контурных токов выполняют в следующем порядке:

1 .Вычерчиваем принципиальную схему цепи и обозначаем все элементы.

2 .Определяем все независимые контуры.

3. Произвольно задаемся направлением протекания контурных токов в каждом из независимых контуров (по часовой стрелке или против). Обозначаем эти токи. Для нумерации контурных токов можно использовать арабские сдвоенные цифры (I11, I22, I33 и т. д.) или римские цифры.



4. По второму закону Кирхгофа, относительно контурных токов, составляем уравнения для всех независимых контуров. При записи равенства считать, что направление обхода контура, для которого составляется уравнение, совпадает с направлением контурного тока данного контура. Следует учитывать и тот факт, что в смежных ветвях, принадлежащих двум контурам, протекают два контурных тока. Падение напряжения на потребителях в таких ветвях надо брать от каждого тока в отдельности.

5. Решаем любым методом полученную систему относительно контурных токов и определяем их.

6. Произвольно задаемся направлением реальных токов всех ветвей и обозначаем их. Маркировать реальные токи надо таким образом, чтобы не путать с контурными. Для нумерации реальных токов можно использовать одиночные арабские цифры (I1, I2, I3 и т. д.).

7 .Переходим от контурных токов к реальным, считая, что реальный ток ветви равен алгебраической сумме контурных токов, протекающих по данной ветви.

При алгебраическом суммировании без изменения знака берется контурный ток, направление которого совпадает с принятым направлением реального тока ветви. В противном случае контурный ток умножается на минус единицу.

Пример расчёта сложной цепи методом контурных токов

В цепи, изображённой на рисунке 1, рассчитать все токи методом контурных токов. Параметры цепи: E1 = 24 В, E2 = 12 В, r1 = r2 = 4 Ом, r3 = 1 Ом, r4 = 3 Ом.

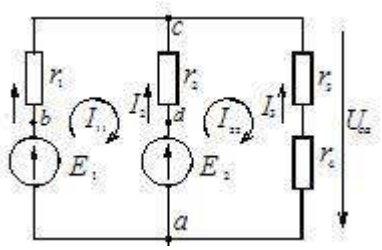


Рис. 1. Схема электрической цепи для примера расчета по методу контурных токов

Решение. Для расчета сложной цепи этим методом достаточно составить два уравнения, по числу независимых контуров. Контурные токи направляем по часовой стрелке и обозначаем I11 и I22 (см. рисунок 1).

По второму закону Кирхгофа относительно контурных токов составляем уравнения:

$$\begin{cases} I_{11}(r_1+r_2)-I_{22}r_2=E_1-E_2, \\ -I_{11}r_2+I_{22}(r_2+r_3+r_4)=E_2, \end{cases} \qquad \begin{cases} I_{11}8-I_{22}4=12, \\ -I_{11}4+I_{22}8=12. \end{cases}$$

Решаем систему и получаем контурные токи I11 = I22 = 3 А.

Произвольно задаемся направлением реальных токов всех ветвей и обозначаем их. На рисунке 1 такими токами являются I1, I2, I3. Направление у этих токов одинаковое – вертикально вверх.

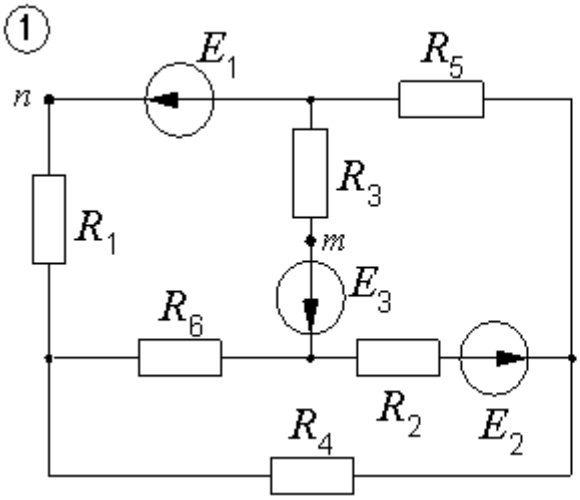
Переходим от контурных токов к реальным. В первой ветви протекает только один контурный ток I11. Направление его совпадает с направлением реального тока ветви. В таком случае реальный ток I1 + I11 = 3 А.

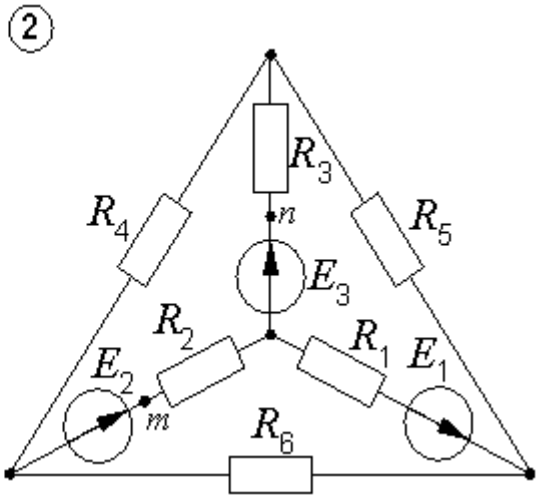
Реальный ток второй ветви формируется двумя контурными I11 и I22. Ток I22 совпадает по направлению с реальным, а I11 направлен навстречу реальному. В результате I2 = I22 - I11 = 3 - 3 = 0А.

В третьей ветви протекает только контурный ток I22. Направление этого тока противоположно направлению реального, поэтому для I3 можно записать I3 = -I22 = -3А.

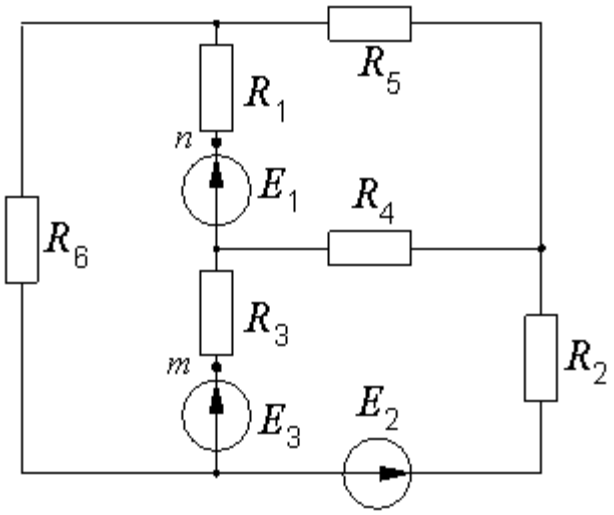
Следует отметить, как положительный факт, что в методе контурных токов по сравнению с решением по законам Кихгофа приходится решать систему уравнений меньшего порядка. Однако этот метод не позволяет сразу определять реальные токи ветвей.

Варианты схем цепи постоянного тока

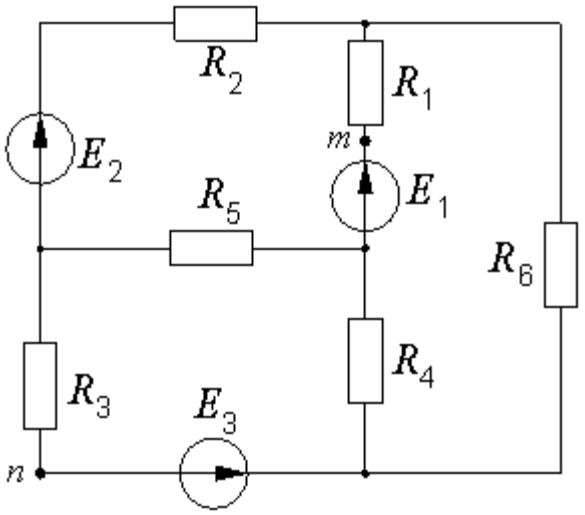




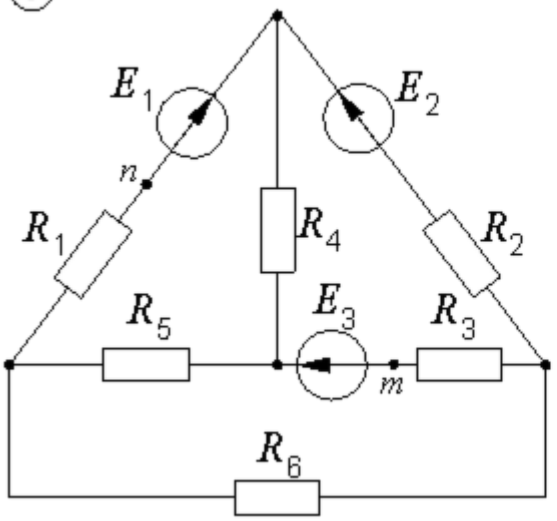
③


R1
R5
n
E1
R4
R6
R3
R2
m
E3
E2

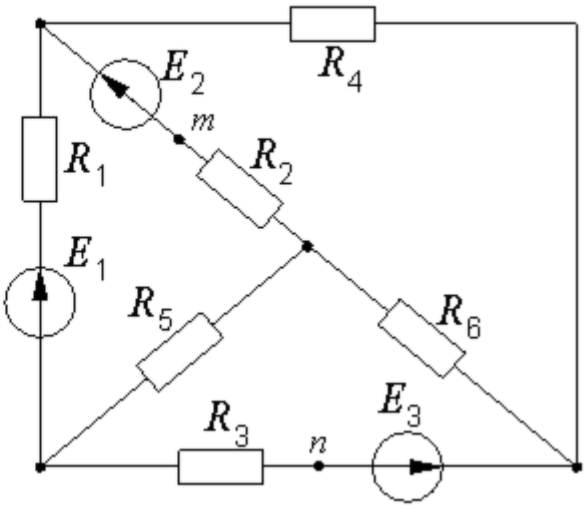
④


R2
R1
E2
m
E1
R5
R6
R3
R4
E3
n

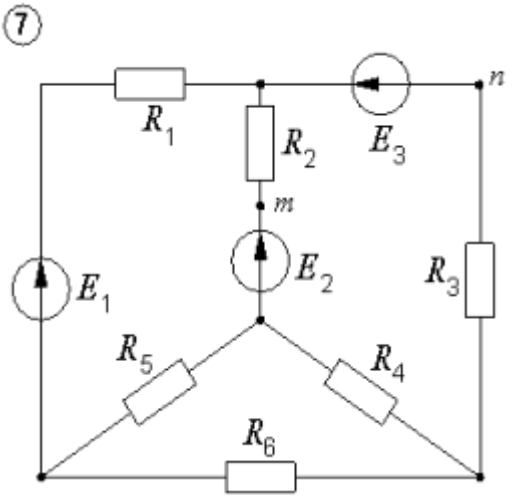
⑤


E1
E2
n
R1
R4
R2
R5
E3
R3
m
R6

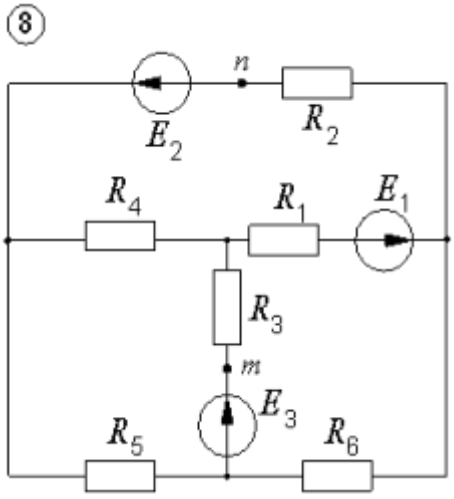
⑥

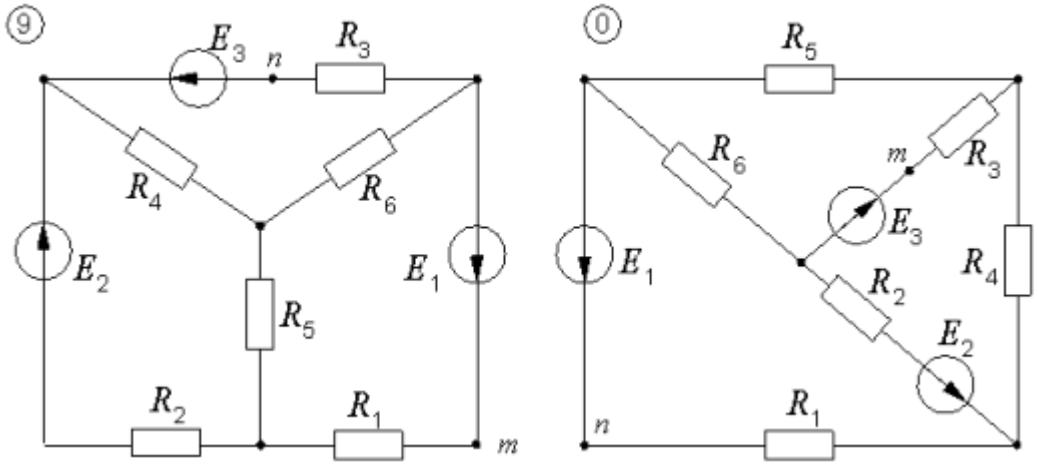

R4
E2
m
R1
R2
E1
R5
R6
R3
E3
n

⑦


R1
R2
E3
n
m
E1
E2
R3
R5
R4
R6

⑧


n
E2
R2
R4
R1
E1
R3
m
E3
R5
R6



1.5. Численные значения параметров цепи

Первая цифра варианта	E_1, В	R_1, Ом	R_4, Ом	Вторая цифра варианта	E_2, В	R_2, Ом	R_5, Ом	Третья цифра варианта	E_3, В	R_3, Ом	R_6, Ом
1	50	10	5	1	80	40	25	1	120	40	100
2	75	25	10	2	120	20	10	2	80	10	20
3	100	50	50	3	30	5	40	3	30	5	50
4	25	25	40	4	40	10	50	4	75	25	10
5	80	20	25	5	50	25	40	5	90	10	50
6	120	40	20	6	75	25	100	6	110	10	25
7	30	10	50	7	80	10	20	7	45	5	25
8	100	5	40	8	90	5	25	8	70	10	40
9	150	50	5	9	60	20	50	9	125	25	40
0	40	20	20	0	25	5	50	0	50	50	20

№ по журналу	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
№ варианта для выбора задания	121	111	211	222	323	333	444	434	555	523	666	624	777	756	888	854	856

№ по журналу	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
№ варианта для выбора задания	999	965	907	000	190	265	785	945	367	863	549	675	189	239	378	320	136	610

Контрольные вопросы:

1.Что такое контур?

2.Что такое контурный ток?

3.Назовите основные пункты расчета цепи методом контурных токов?

4.Как осуществить переход от контурных токов к реальным?

Содержание отчета.

1. Цель работы.
2. Принципиальная электрическая схема.
3. Исходные данные
4. Формулы, необходимые для расчета.
5. Решение задачи.

6. Вывод по работе

ЛАБОРАТОРНАЯ РАБОТА № 5

Тема: Расчет разветвленной цепи с помощью уравнений Кирхгофа. Расчет цепи с помощью метода узловых потенциалов. Составление уравнений состояния для расчета цепи методом контурных токов

Цель работы: приобрести умения решения сложных цепей постоянного тока методом контурных токов.

Методика расчета цепи методом контурных токов

В методе контурных токов за неизвестные величины принимаются расчетные (контурные) токи, которые якобы протекают в каждом из независимых контуров. Таким образом, количество неизвестных токов и уравнений в системе равно числу независимых контуров цепи.

Расчет токов ветвей по методу контурных токов выполняют в следующем порядке:

1 .Вычерчиваем принципиальную схему цепи и обозначаем все элементы.

2 .Определяем все независимые контуры.

3. Произвольно задаемся направлением протекания контурных токов в каждом из независимых контуров (по часовой стрелке или против). Обозначаем эти токи. Для нумерации контурных токов можно использовать арабские сдвоенные цифры (I11, I22, I33 и т. д.) или римские цифры.

4. По второму закону Кирхгофа, относительно контурных токов, составляем уравнения для всех независимых контуров. При записи равенства считать, что направление обхода контура, для которого составляется уравнение, совпадает с направлением контурного тока данного контура. Следует учитывать и тот факт, что в смежных ветвях, принадлежащих двум контурам, протекают два контурных тока. Падение напряжения на потребителях в таких ветвях надо брать от каждого тока в отдельности.

5. Решаем любым методом полученную систему относительно контурных токов и определяем их.

6. Произвольно задаемся направлением реальных токов всех ветвей и обозначаем их. Маркировать реальные токи надо таким образом, чтобы не путать с контурными. Для нумерации реальных токов можно использовать одиночные арабские цифры (I1, I2, I3 и т. д.).

7 .Переходим от контурных токов к реальным, считая, что реальный ток ветви равен алгебраической сумме контурных токов, протекающих по данной ветви.

При алгебраическом суммировании без изменения знака берется контурный ток, направление которого совпадает с принятым направлением реального тока ветви. В противном случае контурный ток умножается на минус единицу.

Пример расчёта сложной цепи методом контурных токов

В цепи, изображённой на рисунке 1, рассчитать все токи методом контурных токов. Параметры цепи: E1 = 24 В, E2 = 12 В, r1 = r2 = 4 Ом, r3 = 1 Ом, r4 = 3 Ом.

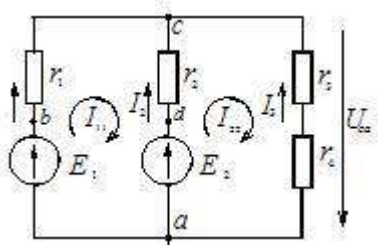


Рис. 1. Схема электрической цепи для примера расчета по методу контурных токов

Решение. Для расчета сложной цепи этим методом достаточно составить два уравнения, по числу независимых контуров. Контурные токи направляем по часовой стрелке и обозначаем I11 и I22 (см. рисунок 1).

По второму закону Кирхгофа относительно контурных токов составляем уравнения:

$$\begin{cases} I_{11}(r_1+r_2)-I_{22}r_2=E_1-E_2, \\ -I_{11}r_2+I_{22}(r_2+r_3+r_4)=E_2, \end{cases} \qquad \begin{cases} I_{11}8-I_{22}4=12, \\ -I_{11}4+I_{22}8=12. \end{cases}$$

Решаем систему и получаем контурные токи I11 = I22 = 3 А.

Произвольно задаемся направлением реальных токов всех ветвей и обозначаем их. На рисунке 1 такими токами являются I1, I2, I3. Направление у этих токов одинаковое – вертикально вверх.

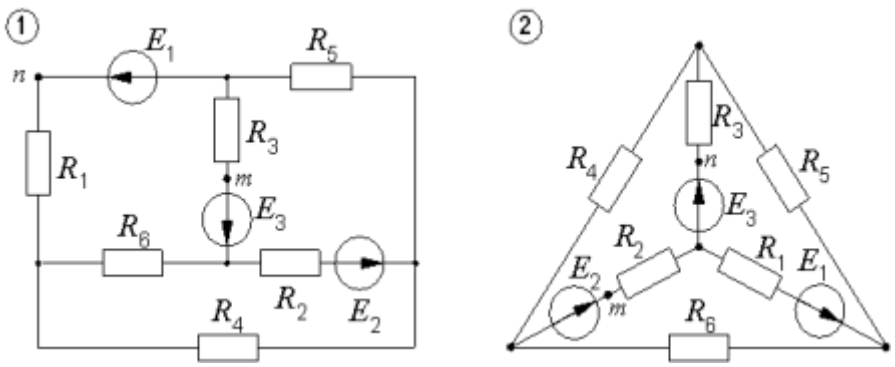
Переходим от контурных токов к реальным. В первой ветви протекает только один контурный ток I11. Направление его совпадает с направлением реального тока ветви. В таком случае реальный ток I1 + I11 = 3 А.

Реальный ток второй ветви формируется двумя контурными I11 и I22. Ток I22 совпадает по направлению с реальным, а I11 направлен навстречу реальному. В результате I2 = I22 - I11 = 3 - 3 = 0А.

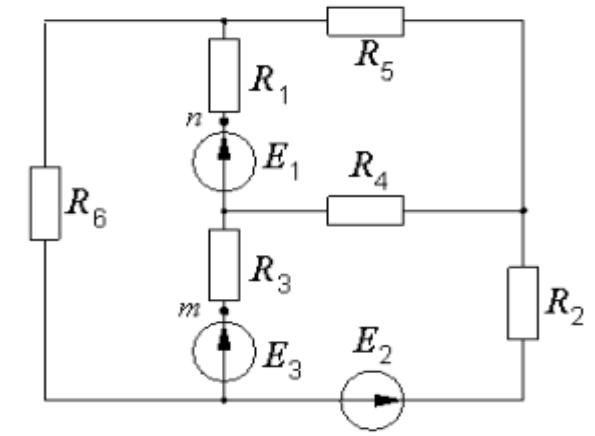
В третьей ветви протекает только контурный ток I22. Направление этого тока противоположно направлению реального, поэтому для I3 можно записать I3 = -I22 = -3А.

Следует отметить, как положительный факт, что в методе контурных токов по сравнению с решением по законам Кихгофа приходится решать систему уравнений меньшего порядка. Однако этот метод не позволяет сразу определять реальные токи ветвей.

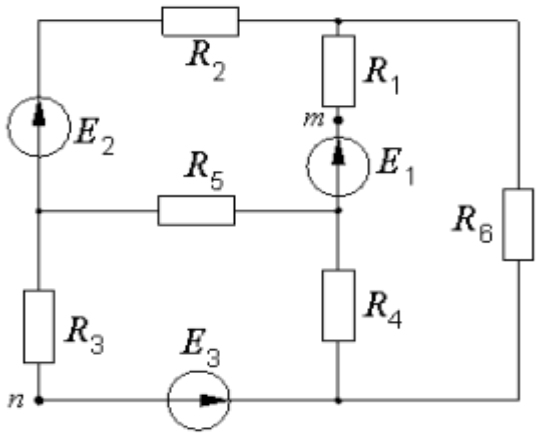
Варианты схем цепи постоянного тока



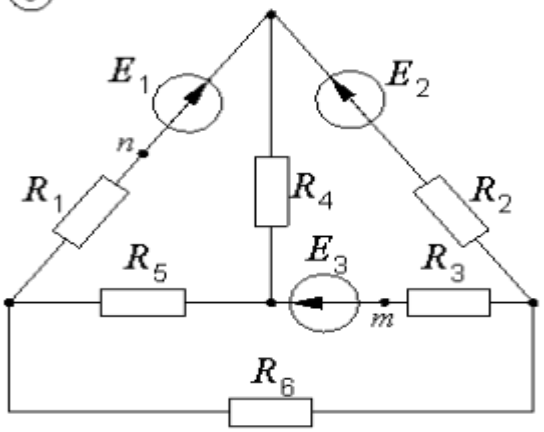
③


R1
R5
n
E1
R4
R6
R3
R2
m
E3
E2

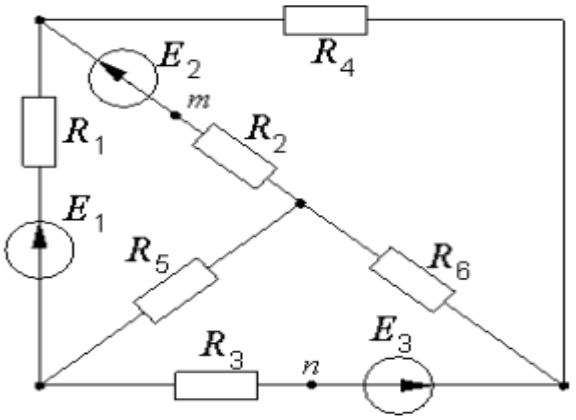
④


R2
R1
E2
m
E1
R5
R6
R3
R4
E3
n

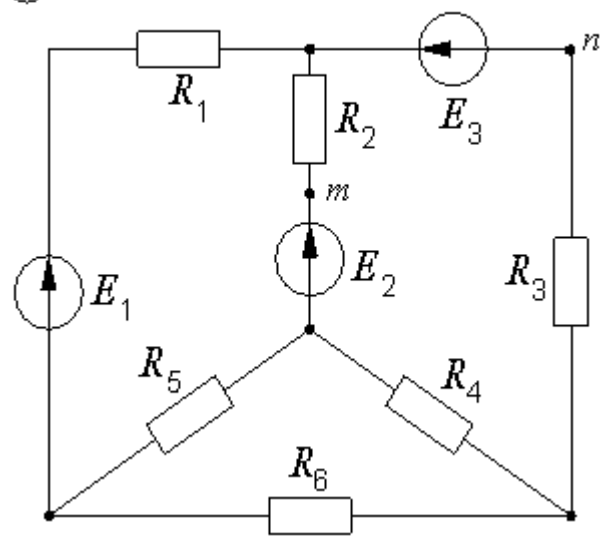
⑤


E1
E2
n
R1
R4
R2
R5
E3
R3
m
R6

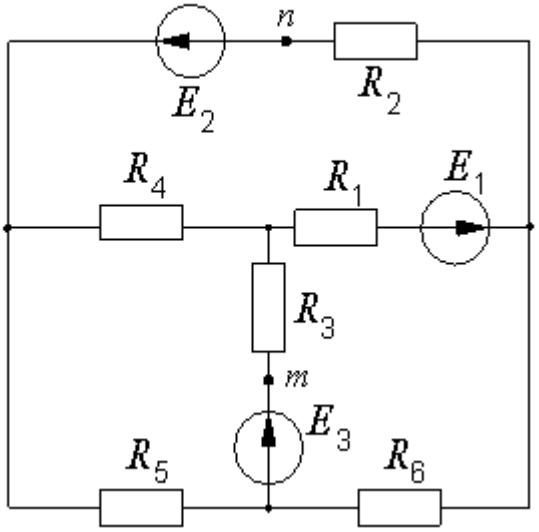
⑥

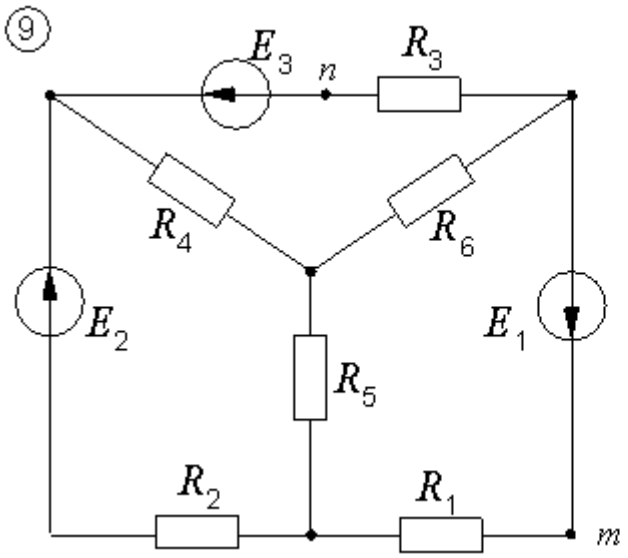

R4
E2
m
R1
R2
E1
R5
R6
R3
E3
n

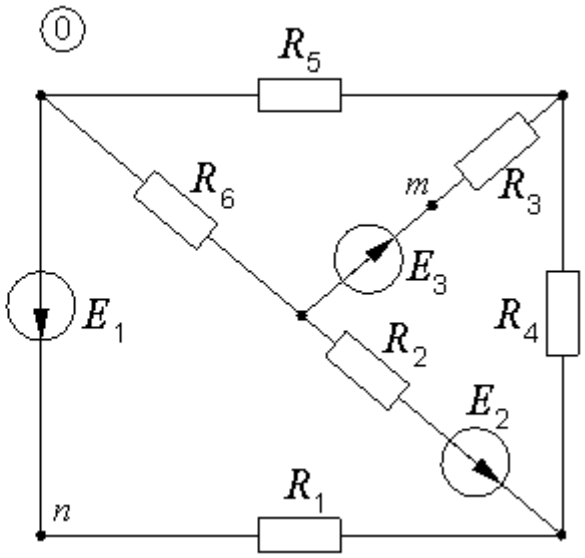
⑦


R1
R2
E3
n
m
E2
E1
R3
R5
R4
R6

⑧


n
E2
R2
R4
R1
E1
R3
m
E3
R5
R6





1.5. Численные значения параметров цепи

Первая цифра варианта	E_1, В	R_1, Ом	R_4, Ом	Вторая цифра варианта	E_2, В	R_2, Ом	R_5, Ом	Третья цифра варианта	E_3, В	R_3, Ом	R_6, Ом
1	50	10	5	1	80	40	25	1	120	40	100
2	75	25	10	2	120	20	10	2	80	10	20
3	100	50	50	3	30	5	40	3	30	5	50
4	25	25	40	4	40	10	50	4	75	25	10
5	80	20	25	5	50	25	40	5	90	10	50
6	120	40	20	6	75	25	100	6	110	10	25
7	30	10	50	7	80	10	20	7	45	5	25
8	100	5	40	8	90	5	25	8	70	10	40
9	150	50	5	9	60	20	50	9	125	25	40
0	40	20	20	0	25	5	50	0	50	50	20

№ по журналу	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
№ варианта для выбора задания	121	111	211	222	323	333	444	434	555	523	666	624	777	756	888	854	856

№ по журналу	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
№ варианта для выбора задания	999	965	907	000	190	265	785	945	367	863	549	675	189	239	378	320	136	610

Контрольные вопросы:

1.Что такое контур?

2.Что такое контурный ток?

3.Назовите основные пункты расчета цепи методом контурных токов?

4.Как осуществить переход от контурных токов к реальным?

Содержание отчета.

1. Цель работы.
2. Принципиальная электрическая схема.
3. Исходные данные
4. Формулы, необходимые для расчета.
5. Решение задачи.
6. Вывод по работе

ЛАБОРАТОРНАЯ РАБОТА № 6

Тема : Расчет цепей. Резонансные явления в электроцепях переменного тока.

Цель работы: Определить токи в ветвях при смешанном соединении приемников электрической энергии и источников электрической энергии, используя четыре метода расчета сложных электрически цепей.

Пояснения к работе.

Электрической схемой называется графическое изображение электрической цепи, содержащие условные обозначения её элементов и показывающие их соединение. Элементами электрической схемы являются ветвь, узел, контур.

Место соединения трех и большего числа ветвей называется узлом электрической цепи.

Участок, вдоль которого ток имеет одно и то же значение, называется ветвью электрической цепи или участок электрической цепи, заключенный между двумя узлами, называется ветвью.

Любой замкнутый путь, проходящий по нескольким ветвям, называется контуром электрической цепи.

Ветви, содержащие источники электрической энергии, называются активными, а не содержащие – пассивными.

Первый закон Кирхгофа - сумма токов притекающих к узлу равна сумме токов вытекающих из узла или алгебраическая сумма токов в узле равна нулю.

Второй закон Кирхгофа - алгебраическая сумма ЭДС, действующих в замкнутом контуре электрической цепи, равна алгебраической сумме падений напряжений на всех участках этого контура.

Для того чтобы составить уравнение по второму закону Кирхгофа необходимо проделать следующее:

1. Задаться во всех ветвях цепи предполагаемыми направлениями токов. Если после расчета предполагаемые токи окажутся отрицательными, то действительное направление этих токов будет противоположным (изображается пунктирной линией).
2. Выбрать направление обходов в каждом контуре, либо по часовой стрелке, либо против часовой стрелки.
3. Записать алгебраическую сумму ЭДС в контуре, при этом все ЭДС контура, направление которых совпадают с направлением обходов в данном контуре, берутся со знаком плюс, а все ЭДС, направление которых противоположны направлению обхода, со знаком минус.
4. Записать алгебраическую сумму падений напряжений на всех элементах контура.

При этом падения напряжений, вызванные токами, направление которых совпадают с направлением обхода, пишутся со знаком плюс, а не совпадают - со знаком минус.

Расчет сложной электрической цепи постоянного тока можно осуществить, применяя следующие методы:

1. Метод расчета с применением законов Кирхгофа.

1. Определить количество ветвей в заданной цепи.
2. Задаться предполагаемыми направлениями токов во всех ветвях и направлениями обходов в контурах (указать на схеме).
3. По первому закону Кирхгофа составить **(n-1)** уравнений, где **n** – количество узлов в цепи.
4. Определить количество элементарных контуров цепи **m.** Составить по второму закону Кирхгофа **m** уравнений.
5. Сумма **(n-1)+ m** уравнений равна количеству ветвей цепи, для которой данные уравнения составляются и, следовательно, количеству токов, которые подлежат расчету.
6. В получившуюся систему уравнений подставить числовые значения и определить токи в ветвях.

2. Метод контурных токов.

1. Произвольно выбрать направление контурных токов, обозначить их на схеме и считать таким же направление обходов в контурах.
2. Составить уравнения по второму закону Кирхгофа с контурными токами. Если на участке цепи, входящем в рассматриваемый контур действуют два контурных тока, то падение напряжения на этом участке равно алгебраической сумме падений напряжений, созданных каждым контурным током.
3. Определить величину и направление реальных токов, применяя правило:

   - если на участке цепи действует один контурный ток, то действительный ток равен контурному току и имеет такое же направление;

   - если на участке цепи действуют два контурных тока в противоположных направлениях, то действительный ток равен их разности и имеет направление в сторону большего;

   - если на участке цепи действуют два контурных тока одинаковых направлений, то действительный ток равен их сумме и совпадает по направлению с ними.

3. Метод узлового напряжения.

1. Выбрать направление токов в ветвях произвольно, но одинаково во всех ветвях.
2. Определить проводимости ветвей по формуле **G=1/R**, где R общее сопротивление каждой ветви.
3. Определить узловое напряжение по формуле $\mathbf{U= \sum EG / \sum G}$. Знак ЭДС в алгебраической сумме определяется в соответствии с направлением токов в ветвях, если направление тока и ЭДС совпадают, берут знак «плюс», а не совпадают – знак «минус».
4. Определить токи в ветвях по формуле $\mathbf{I = (E - U)\cdot G}$. Если направление ЭДС не совпадает с направлением тока ветви, то знак «минус» переносится в данную формулу перед ЭДС.

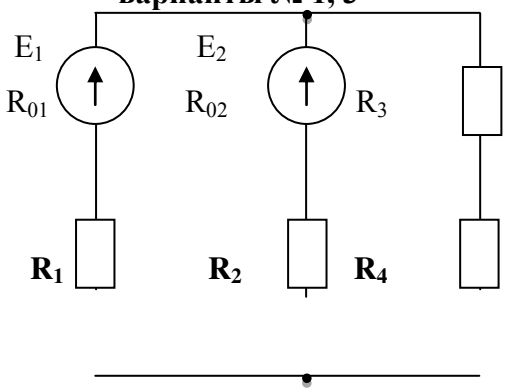
4. Метод наложения.

1. Исключить из электрической цепи все источники кроме одного, учитывая при этом внутренние сопротивления исключенных источников.
2. Указать направления частичных токов на схеме.
3. Определить величину частичных токов в ветвях методом «свертывания».
4. Таким способом определить поочередно частичные токи, созданные каждым источником ЭДС отдельно.
5. Определить величину и направление действительных токов в ветвях как алгебраическую сумму соответствующих частичных токов.

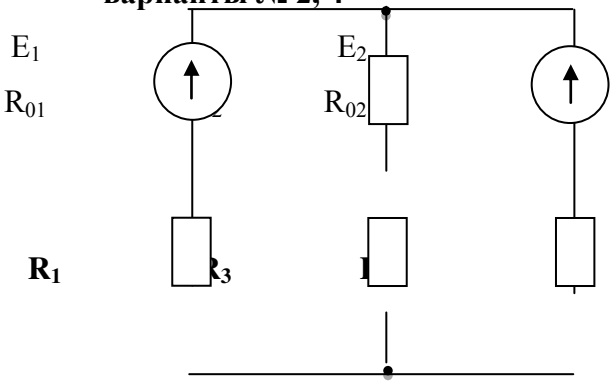
Задание:

1. Необходимо определить токи в ветвях, применяя четыре метода расчета цепей.
2. Начертить принципиальную схему своего варианта.

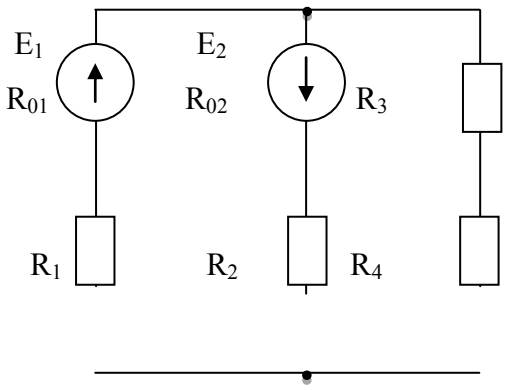
варианты № 1, 3



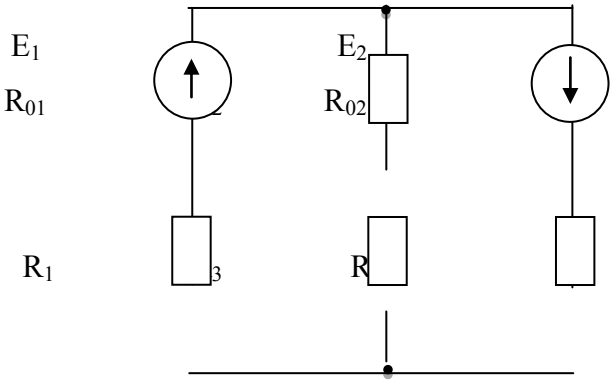
варианты № 2, 4



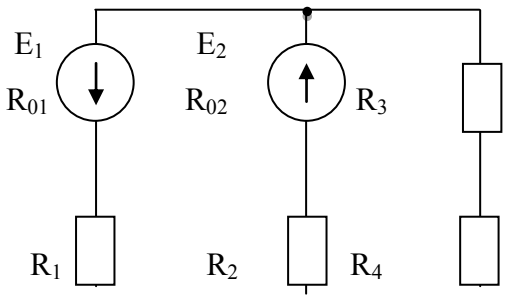
варианты № 5, 7



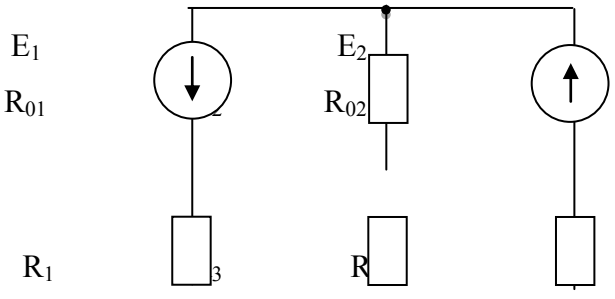
варианты № 6, 8

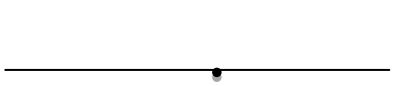


варианты № 9, 11

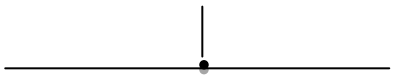


варианты № 10, 12

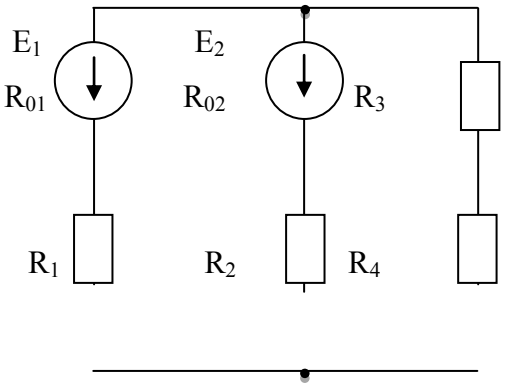


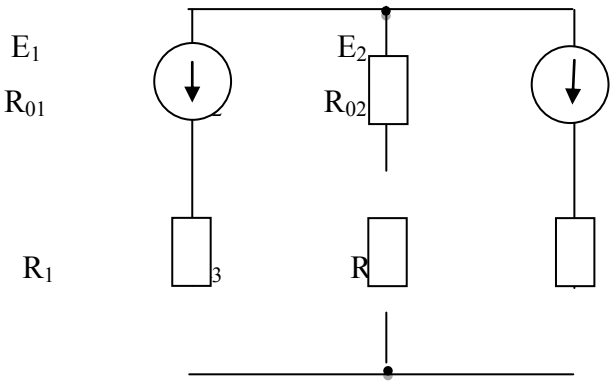


варианты № 13, 15



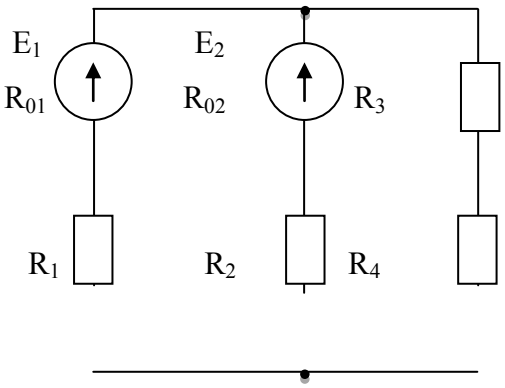
варианты № 14, 16

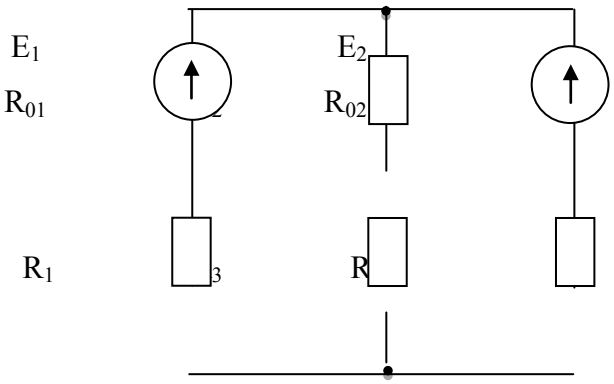




варианты № 17, 19

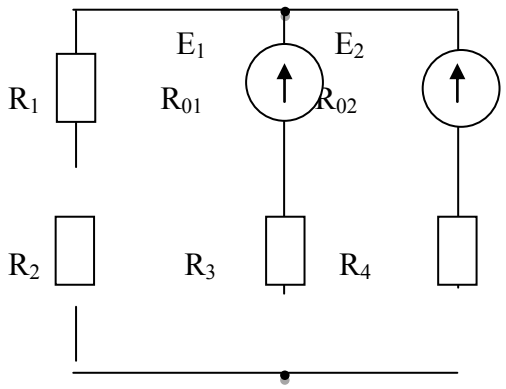
варианты № 18, 20

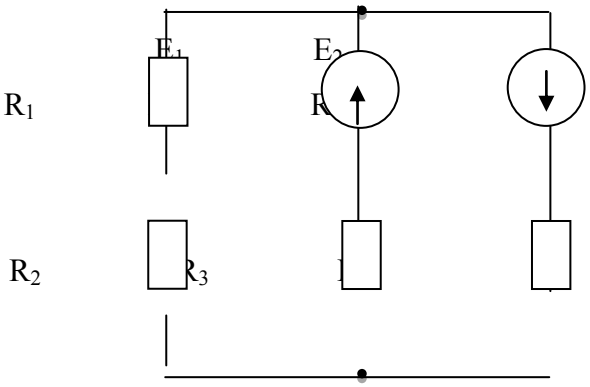




варианты № 21, 23

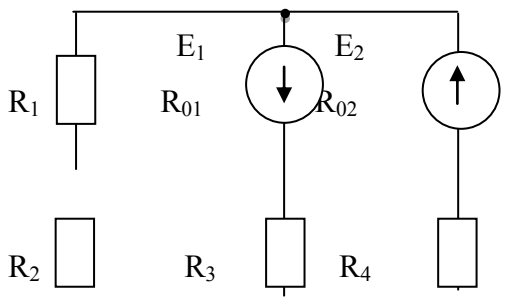
варианты № 22, 24

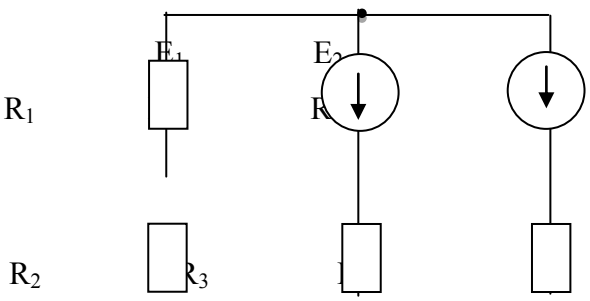




варианты № 25, 27

варианты № 26, 28





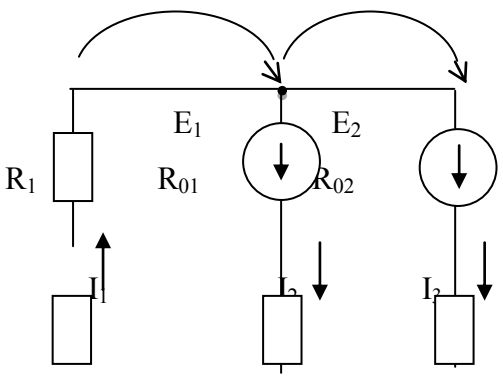
3. Переписать из таблицы данные своего варианта.

№ варианта	E_1	E_2	R_1	R_2	R_3	R_4	R_{01}	R_{02}
	В	**В**	**Ом**	**Ом**	**Ом**	**Ом**	**Ом**	**Ом**
1, 2	52	27	30	10	20	15	1	2
3, 4	46	47	25	15	30	40	2	3
5, 6	35	40	26	28	20	18	3	2
7, 8	50	36	24	32	28	16	4	2
9, 10	48	37	23	26	30	35	2	3
11, 12	36	47	15	20	25	30	3	4
13, 14	40	38	30	18	20	40	3	5
15, 16	28	32	30	19	22	30	4	3
17, 18	30	40	38	28	32	40	5	2
19, 20	25	41	35	25	42	50	1	3
21, 22	39	48	28	30	41	49	1	2
23, 24	50	36	37	29	32	25	5	6
25, 26	48	45	32	28	31	29	6	5
27, 28	36	43	30	41	25	35	2	3

4. Выполнить расчет, применяя четыре метода. При этом значения токов и их направления должны получиться одинаковыми, независимо от применяемого метода расчета. При расчете применить следующие формулы: последовательное соединение -$\mathbf{R_{12} = R_1 + R_2}$; параллельное соединение - $\mathbf{R_{12} = R_1 \cdot R_2 / (R_1 + R_2)}$; $\mathbf{G = 1/R}$; $\mathbf{G_{12} = G_1 + G_2}$; $\mathbf{I = U/R}$; $\mathbf{U = I \cdot R}$; $\mathbf{R = U/I}$. При решении методом узлового напряжения – $\mathbf{U = \sum EG / \sum G}$;

$\mathbf{I = (E - U) \cdot G}$.

Образец решения задачи .

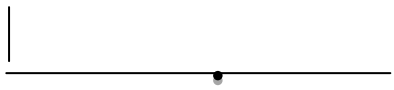


Дано: $E_1 = 36В$, $E_2 = 43В$,

$R_1 = 30$ Ом, $R_2 = 41$Ом, $R_3 = 25$Ом, $R_4 = 35$Ом,

$R_{01} = 2$Ом, $R_{02} = 3$Ом

R_2 R_3 R_4 I_1 - ? I_2 - ? I_3 - ?



1.Метод расчета с применением законов Кирхгофа

$$\begin{cases} I_1 = I_2 + I_3 \\ E_1 = I_1(R_1 + R_2) + I_2(R_{01} + R_3) \\ -E_1 + E_2 = -I_2(R_{01} + R_3) + I_3(R_{02} + R_4) \end{cases}$$

$$\begin{cases} I_1 = I_2 + I_3 \\ 36 = I_1(30+41) + I_2(2+25) \\ -36+43 = -I_2(2+25) + I_3(3+35) \end{cases}$$

$$\begin{cases} I_1 = I_2 + I_3 \\ 36 = 71\,(I_2 + I_3) + 27\,I_2 \\ 7 = -27\,I_2 + 38\,I_3 \end{cases}$$

$$\begin{cases} 36 = 71\,(I_2 + I_3) + 27\,I_2 \\ 7 = -27\,I_2 + 38\,I_3 \end{cases}$$

$$\begin{cases} 36 = 71\,I_2 + 71\,I_3 + 27\,I_2 \\ 7 = -27\,I_2 + 38\,I_3 \end{cases}$$

$$\begin{cases} 36 = 98\,I_2 + 71\,I_3 \\ 7 = -27\,I_2 + 38\,I_3 \end{cases}$$

$36 = 98\,I_2 + 71\,I_3$ $\quad I_3 = (36 - 98\,I_2) / 71$

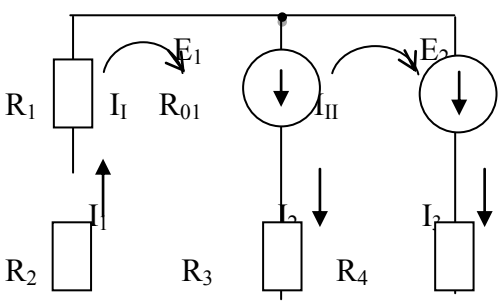
$7 = -27\,I_2 + 38\,I_3$ $\quad 7 = -27\,I_2 + 38\,(36 - 98\,I_2) / 71$ $\quad / \cdot 71$ $\quad 7 \cdot 71 = -27 \cdot 71 I_2 + 38\,(36 - 98\,I_2)$

$497 = -1917\,I_2 + 1368 - 3724\,I_2$ $\quad 5641\,I_2 = 871$ $\quad I_2 = 871/5641 = 0{,}15\text{A}$ $\quad \mathbf{I_2 = 0{,}15A}$

$I_3 = (36 - 98\,I_2) / 71 = (36 - 98 \cdot 0{,}15) / 71 = (36 - 14{,}7) / 71 = 0{,}3\text{A}$ $\quad \mathbf{I_3 = 0{,}3A}$

$I_1 = I_2 + I_3 = 0{,}15 + 0{,}3 = 0{,}45\text{A}$ $\quad \mathbf{I_1 = 0{,}45A}$

2.Метод расчета с применением контурных токов.



$$\begin{cases} E_1 = I_I(R_1 + R_2 + R_{01} + R_3) - I_{II}(R_{01} + R_3) \\ -E_1 + E_2 = I_{II}(R_{01} + R_3 + R_{02} + R_4) - I_I(R_{01} + R_3) \end{cases}$$

$$\begin{cases} 36 = I_I(30 + 41 + 2 + 25) - I_{II}(2 + 25) \\ -36 + 43 = I_{II}(2 + 25 + 3 + 35) - I_I(2 + 25) \end{cases}$$

$$\begin{cases} 36 = 98\ I_I - 27\ I_{II} \ / \cdot 27 \\ 7 = 65\ I_{II} - 27\ I_I \ / \cdot 98 \end{cases}$$

$$\begin{cases} 972 = 2646\ I_I - 729\ I_{II} \\ 686 = -2646\ I_I + 6370\ I_{II} \end{cases}$$

$I_{II} = 1658 / 5641 = 0{,}3$А

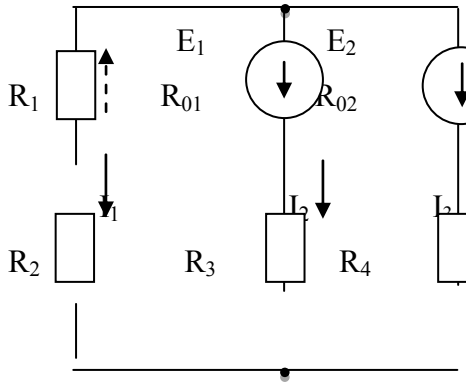
$1658 = 5641\ I_{II}$

$36 = 98\ I_I - 27\ I_{II}$ $\quad I_I = (36 + 27\ I_{II}) / 98 = (36 + 27 \cdot 0{,}3) / 98 = 0{,}45$А

$I_1 = I_I =$ **0,45А** $\quad I_3 = I_{II} =$ **0,3А** $\quad I_2 = I_I - I_{II} = 0{,}45 - 0{,}3 =$ **0,15А**

3.Метод расчета с применением узлового напряжения.

Направления токов во всех ветвях выбираются к одному узлу.



$G_1 = 1/(R_1 + R_2) = 1/(30+41) = 0{,}014$ См

$G_2 = 1/(R_{01} + R_3) = 1/(2 + 25) = 0{,}037$ См

$G_3 = 1/(R_{02} + R_4) = 1/(3 + 35) = 0{,}026$ См

$U = (E_1 \cdot G_2 + E_2 \cdot G_3) / (G_1 + G_2 + G_3)$

$U = (36 \cdot 0{,}037 + 43 \cdot 0{,}026) / (0{,}014 + 0{,}037 + 0{,}026) =$

$= 2{,}45 / 0{,}077 = 31{,}8$ В

$I_1 = - U \cdot G_1 = - 31{,}8 \cdot 0{,}014 =$ **- 0,45А**

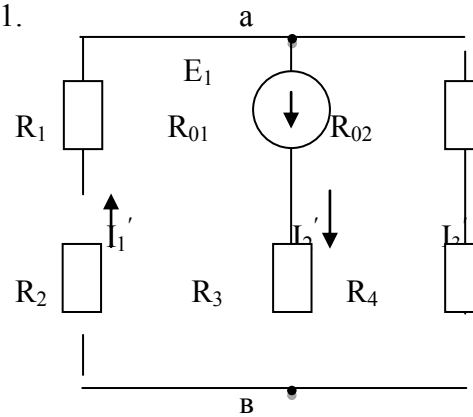
Направление тока I_1 меняем на противоположное, так как ток получился со знаком «минус»

$I_2 = (E_1 - U) \cdot G_2 = (36 - 31{,}8) \cdot 0{,}037 =$ **0,15А**

$I_3 = (E_2 - U) \cdot G_3 = (43 - 31{,}8) \cdot 0{,}026 =$ **0,3А**

4.Метод расчета с применением принципа наложения.

1.



$R'_{экв} = (R_{01}+R_3) + (R_1+R_2) \cdot (R_{02}+R_4) / (R_1+R_2+R_{02}+R_4)$

$R'_{экв} = (2 + 25) + (30+41) \cdot (3 + 35) / (30+41+3 + 35) =$

$= 27 + 24{,}75 = 51{,}75$ Ом $\quad R'_{ав} = 24{,}75$ Ом

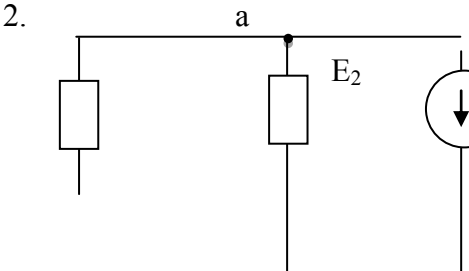
$I_2' = E_1 / R'_{экв} = 36/51{,}75 = 0{,}696$ А

$U_{ав}' = I_2' \cdot R'_{ав} = 0{,}696 \cdot 24{,}75 = 17{,}2$ В

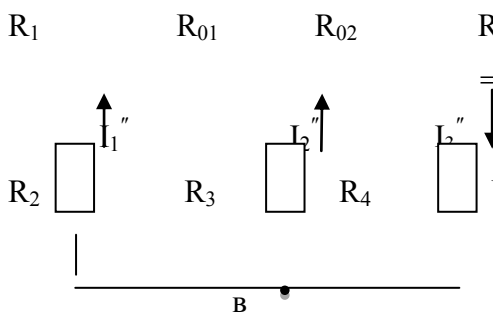
$I_1' = U_{ав}' / (R_1+R_2) = 17{,}2/(30+41) = 0{,}242$А

$I_3' = U_{ав}' / (R_{02}+R_4) = 17{,}2/(3 + 35) = 0{,}453$А

2.



$R''_{экв} = (R_{02}+R_4) + (R_1+R_2) \cdot (R_{01}+R_3) / (R_1+R_2+R_{01}+R_3)$



$R''_{экв} = (3 + 35) + (30+41)\cdot(2 + 25) /(30+41+2 + 25) =$

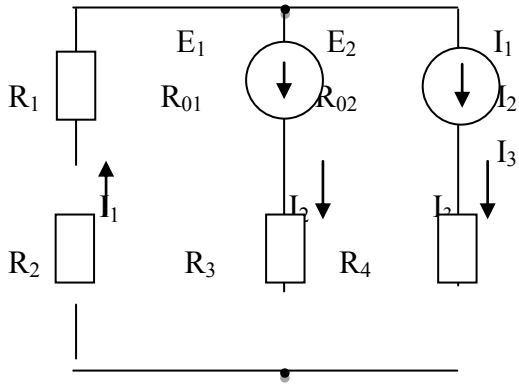
$= 38 + 19,56 = 57,56$ Ом $\quad R''_{ав} = 19,56$ Ом

$I_3'' = E_2 / R''_{экв} = 43/57,56 = 0,747$ А

$U_{ав}'' = I_3''\cdot R''_{ав} = 0,747\cdot 19,56 = 14,6$ В

$I_1'' = U_{ав}''/(R_1+R_2) = 14,6/(30+41) = 0,206$А

$I_2'' = U_{ав}''/(R_{01}+R_3) = 14,6/(2 + 25) = 0,541$А



$I_1 = I_1' + I_1'' = 0,243 + 0,206 =$ **0,45 А**

$I_2 = I_2' - I_2'' = 0,696 - 0,541 =$ **0,15 А**

$I_3 = I_3'' - I_3' = 0,747 - 0,453 =$ **0,3 А**

Работа на занятии.

1. В соответствии с принципиальной схемой своего варианта, используя исходные данные, произвести расчет токов во всех ветвях цепи, применяя четыре метода расчета. Независимо от применяемого метода решения, ответы должны получиться одинаковыми.
2. При решении применить предлагаемые формулы и образец решения подобной задачи.

Содержание отчета.

1. Цель работы.
2. Принципиальная электрическая схема.
3. Исходные данные
4. Формулы, необходимые для расчета.
5. Решение задачи, с применением четырех методов расчета.
6. Вывод по работе.

Лабораторная работа № 7

Тема : **Собрать и испытать выпрямители на полупроводниковых диодах. Собрать и испытать усилители на транзисторах**

Цель: Изучение характеристик и параметров диодов, экспериментальное исследование их вольтамперных характеристик (ВАХ).

Теоретический материал для подготовки и выполнения лабораторной работы:

Полупроводниковым диодом называют электропреобразовательный полупроводниковый прибор с одним *p-n*-переходом.

В зависимости от основного назначения и вида используемого явления в *p-n*-переходе различают шесть основных функциональных типов полупроводниковых диодов: выпрямительные, высокочастотные, импульсные, туннельные, стабилитроны, варикапы.

Одним из свойств p–n-перехода является способность изменять свое сопротивление в зависимости от полярности напряжения внешнего источника. Причем разница сопротивлений при прямом и обратном направлениях тока через p–n-переход может быть настолько велика, что в ряде случаев, например, для силовых диодов, можно считать, что ток протекает через диод только в одном направлении – прямом, а в обратном направлении ток настолько мал, что им можно пренебречь. Прямое направление – это когда электрическое поле внешнего источника направлено навстречу электрическому полю p–n-перехода, а обратное – когда направления этих электрических полей совпадают. Полупроводниковые диоды, использующие вентильное свойство p–n-перехода, называются выпрямительными диодами и широко используются в различных устройствах для выпрямления переменного тока.

Основной характеристикой полупроводниковых диодов служит вольт-амперная характеристика (рис.1)

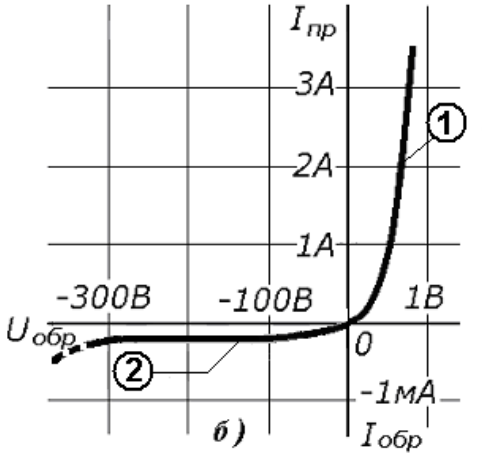


Рис.1

В области обратных напряжений можно пренебречь падением напряжения в объёме полупроводника. При достижении обратным напряжением определённого критического значения ток диода начинает резко возрастать. Это явление называют пробоем диода.

Различают два основных вида пробоя электронно-дырочного перехода: электрический и тепловой. В обоих случаях резкий рост тока связан с увеличением числа носителей заряда в переходе. Электрический пробой бывает двух видов - лавинный и туннельный.

Основными параметрами выпрямительных диодов являются:

- Максимально допустимое обратное напряжение $U_{обр.\ max}$ – наибольшее постоянное (или импульсное) обратное напряжение, при котором диод может длительно и надежно работать обычно $U_{обр\max} \le U_{проб}$;
- Максимально допустимый выпрямленный ток $I_{вп.\ ср\ max}$ — средний за период ток через диод (постоянная составляющая), при котором обеспечивается его надежная длительная работа
- Прямое и обратное сопротивления диода постоянному току, определяемые по его ВАХ с использованием следующих соотношений:

$$\boldsymbol{R}д\ пр = Uпр / Iпр; \qquad \boldsymbol{R}д\ обр = Uобр / Iобр;$$

- Максимальная частота f_{max} — наибольшая частота подводимого напряжения, при которой выпрямитель на данном диоде работает достаточно эффективно, а нагрев диода не превышает допустимой величины.
- Постоянное прямое напряжение $U_{пр}$ – падение напряжения на диоде при протекании через него постоянного прямого тока $I_{пр}$ – заданного ТУ.
- Постоянный обратный ток $I_{обр}$ — ток через диод при постоянном обратном напряжении ($U_{обр\ мах}$). Чем меньше $I_{обр}$, тем качественнее диод.

Превышение максимально допустимых величин ведет к резкому сокращению срока службы или пробою диода.

При коротких импульсах необходимо учитывать инерционность процессов включения и выключения диода, что характеризуется следующими параметрами:

1) Время установления прямого напряжения на диоде ($t_{уст}$) – время, за которое напряжение на диоде при включении прямого тока достигает своего стационарного значения с заданной точностью.

2) Время восстановления обратного сопротивления диода ($t_{восст.}$) определяется как время, в течение которого обратный ток диода после переключения полярности приложенного напряжения с прямого на обратное достигает своего стационарного значения с заданной точностью, обычно 10% от максимального обратного тока.

Там, где требуется малое время переключения, используют диоды Шотки. Они имеют переход металл — полупроводник, который обладает выпрямительным эффектом. Накопление заряда в переходе этого типа выражено слабо. Поэтому время переключения может быть уменьшено до

значения порядка 100 пс. Другой особенностью этих диодов является малое (по сравнению с обычными кремниевыми диодами) прямое напряжение, составляющее около 0,3 В.

Порядок выполнения работы:

1. *Собрать схему для исследования выпрямительного диода на постоянном токе в соответствии с принципиальной схемой (рис. 2). Для измерения анодного тока включить миллиамперметр постоянного тока с пределом 100 мА. Для измерения анодного напряжения использовать мультиметр. Последовательно с диодом включить токоограничивающий резистор Rн (RP2 на стенде).*

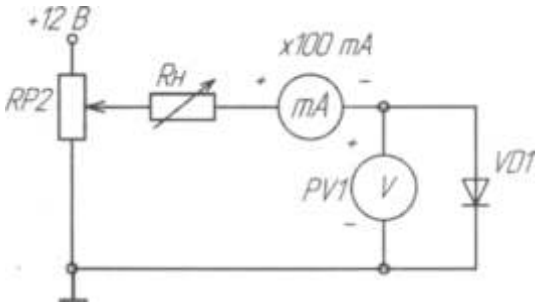


Рис.2

2. После проверки схемы преподавателем включить источник питания.

3. Снять вольтамперную характеристику выпрямительного диода на постоянном токе для прямой ветви. Для снятия характеристики регулировать напряжение на выходе потенциометра. Результаты измерений занести в таблицу 1. Выключить источник питания.

Таблица 1

I, мА	10	20	30	40	50	60	70	80	90	100
U, В										

4. *Собрать схему для исследования диода Шотки на постоянном токе в соответствии с принципиальной схемой рис. 3 заменив диод VD1 на диод VD2. Для измерения анодного тока включить миллиамперметр постоянного тока с пределом 100 мА. Для измерения анодного напряжения использовать мультиметр. Последовательно с диодом включить токоограничивающий резистор Rн (RP2 на стенде)*

5. После проверки схемы преподавателем включить источник питания.

6.Снять вольтамперную характеристику диода Шотки на постоянном токе для прямой ветви. Для снятия характеристики регулировать напряжение на выходе потенциометра. Результаты измерений занести в таблицу 2. Выключить источник питания.

Таблица 2

I, мА	10	20	30	40	50	60	70	80	90	100
U, В										

7. Собрать схему для снятия обратной ветви ВАХ VD1, подключив к PR2 источник -12В и заменив миллиамперметр, поменяв также его полярность подключения (рис.3).

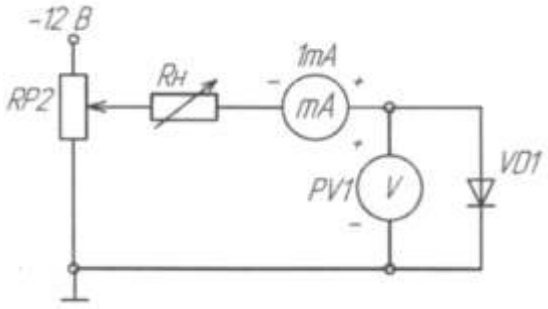


Рис.3

8. После проверки схемы преподавателем включить источник питания.

9. Снять обратную ветвь ВАХ выпрямительного диода на постоянном токе. Для снятия характеристики регулировать напряжение на выходе потенциометра. Результаты измерений занести в таблицу 3. Выключить источник питания.

Таблица 3

I, мА					
U, В					

10. Собрать схему для снятия обратной ветви ВАХ диода Шотки, заменив VD1 на VD2 (рис.3).

11. После проверки схемы преподавателем включить источник питания.

12. Снять обратную ветвь ВАХ выпрямительного диода на постоянном токе. Для снятия характеристики регулировать напряжение на выходе потенциометра. Результаты измерений занести в таблицу 4. Выключить источник питания.

Таблица 4

I, мА					
U, В					

13. Построить для каждого диода вольтамперную характеристику (прямую и обратную ветвь совместить на одном графике (смотри теоретический материал).

14. Разобрать электрическую цепь и сдать стенд преподавателю.

15. Сделать вывод по работе.

Контрольные вопросы:

1. Сколько выводов имеет диод?
2. Что называют прямым включением диода?

3. Какой ток протекает через диод при его обратном включении, и чем он вызван?
4. Какое явление называется пробоем диода?
5. В чем особенность диодов Шотки, их достоинства и область применения.

Содержание отчета.

1. Цель работы.
2. Принципиальная электрическая схема.
3. Исходные данные
4. Формулы, необходимые для расчета.
5. Решение задачи.
6. Вывод по работе

ЛАБОРАТОРНАЯ РАБОТА № 8

Тема Пуск и остановка двигателей

Цель работы: научить подключению трёхфазного асинхронного эл. двигателя по схемам «звезда» и «треугольник»; измерению токов и напряжений при работе двигателя.

МТО :

1. Асинхронный двигатель на напряжение питания 220/380 В или 127/220 В
2. Вольтметр
3. Амперметр электромагнитный 2шт. (сх. Звезда 1А; сх. треугольник 5А)

Порядок выполнения работы.

1. Составить таблицу измерительных приборов.

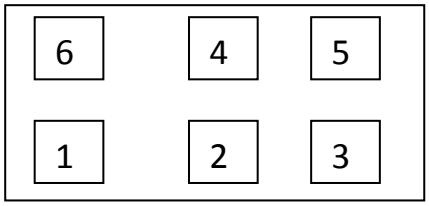
№ п\п	Наименование прибора.	Предел измерения.	Количество делений шкалы.	Цена деления шкалы.

2. Ознакомиться с двигателем и записать данные шильды (паспорта) в таблицу.

№ п/п	Наименование параметра	**Параметры.**
1	**Название и тип двигателя**	
2	Число пар полюсов	
3	Напряжение питания	
4	Схемы включения	

	напряжений питания	
5	Частота тока	
6	Мощность двигателя	
7	Число оборотов ротора	
8	КПД	

3. Подключить к монтажному щитку выводы обмоток двигателя по схеме:



4. Согласно схеме указанной на шильде (паспорте) подключить питание двигателя к клеммам щитка на наименьшее напряжение .

В один из линейных проводов схемы включить амперметр.

4.1. Режим холостого хода.

С разрешения преподавателя включить схему. Выполнить пробный запуск двигателя.

При аварийной ситуации выключить питание и сообщить преподавателю.

При нормальной работе двигателя выполнить лабораторную работу.

В момент включения схемы зафиксировать величину пускового тока.

Затем, в процессе холостого хода двигателя, зафиксировать ток холостого хода.

Измерить напряжение питания на всех фазах.

Произвести переключение фаз двигателя с целью изменения направления вращения ротора.

Данные измерений записать в таблицу измерений.

Отключить питание.

4.2. Режим нагрузки.

С разрешения преподавателя включить схему.

Нагрузку двигателя выполняет преподаватель!

При нагруженном двигателе, зафиксировать ток в режиме нагрузки.

Измерить напряжение питания на всех фазах.

Данные измерений записать в таблицу измерений.

Отключить питание.

5. Согласно схеме указанной на шильде (паспорте) подключить питание двигателя к клеммам щитка на наибольшее напряжение .

В один из линейных проводов схемы включить амперметр.

5.1. Режим холостого хода.

С разрешения преподавателя включить схему. Выполнить пробный запуск двигателя.

При аварийной ситуации выключить питание и сообщить преподавателю.

При нормальной работе двигателя выполнить лабораторную работу.

В момент включения схемы зафиксировать величину пускового тока

Затем, в процессе холостого хода двигателя, зафиксировать ток холостого хода.

Измерить напряжение питания на всех фазах.

Произвести переключение фаз двигателя с целью изменения направления вращения ротора.

Данные измерений записать в таблицу измерений.

Отключить питание.

5.2. Режим нагрузки.

С разрешения преподавателя включить схему.

Нагрузку двигателя выполняет преподаватель!

В момент включения схемы зафиксировать величину пускового тока

Затем, при нагруженном двигателе, зафиксировать ток в режиме нагрузки.

Измерить напряжение питания на всех фазах.

Данные измерений записать в таблицу измерений.

Отключить питание.

6. Таблица измерений.

Наименование Схемы соединения	$I_{пуск}$	$I_{хол. ход.}$	Ua	Uв	Uc	Uав	Uас	Uвс	$I_{нагр}$	$U_{нагр}$	$P_{нагр}$
Звезда											
Треугольник											

7. Отключить двигатель от питания.

С разрешения преподавателя закончить работу. Отсоединить выводы от щитка питания. Разобрать монтажную схему. Подготовить к сдаче преподавателю рабочее место.

Примечание:

Для асинхронных двигателей: УАД 32

C1; C2; C3; - начала фазных обмоток

C4; C5; C6; - концы фазных обмоток.

Соединение звездой без нулевого провод**а**.

Содержание отчета.

1. Цель работы.
2. Принципиальная электрическая схема.
3. Исходные данные
4. Формулы, необходимые для расчета.
5. Решение задачи.
6. Вывод по работе.

методические рекомендации по выполнению

ПРАКТИЧЕСКИХ РАБОТ

дисциплина ОП 03 Основы электроники и цифровой схемотехники
по профессии
09.01.03. «Мастер по обработке цифровой информации

г. Балашиха 2020г

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Учебная дисциплина ОП 03 «Основы электроники и цифровой схемотехники» входит в цикл общепрофессиональных дисциплин для профессии 09.01.03 «Мастер по обработке цифровой информации». Данные методические рекомендации предназначены для обучающихся, изучающие данную дисциплину, и включает в себя 12 практических работ.

Данные методические рекомендации входят в состав учебно-методического комплекса по дисциплине и предполагают формирование у учащихся знаний и умений в области основ электроники и цифровой схемотехники.

2.ОБЩИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ

1. Практические работы выполняются в соответствии с графиком учебного процесса, который доводится до студентов в начале каждого семестра.

2. К выполнению практических работ допускаются студенты, прошедшие инструктаж по правилам техники безопасности.

3. Формы организации обучающихся при проведении практических работ: фронтальная, групповая и индивидуальная.

При фронтальной форме организации занятий все обучающиеся выполняют одновременно одну и ту же работу. При групповой форме организации занятий одна и та же работа выполняется бригадами по 2 - 5 человек. При индивидуальной форме организации занятий каждый обучающийся выполняет индивидуальное задание.

Практические работы выполняются с использованием предоставленных преподавателем технических средств и программного обеспечения.

4. Каждый обучающийся должен заранее подготовиться к очередной практической работе: изучить или повторить теоретический материал, рассмотреть требуемые схемы, повторить необходимые формулы, таблицы и пояснения по выполнению практической работы необходимо занести в тетрадь или оформить их в электронном варианте на ПК. Неподготовленные студенты к выполнению практических работ могут быть не допущены.

5. Перед выполнением работы старший в бригаде (или каждый обучающийся) получает у преподавателя комплект наборных элементов и необходимую электроизмерительную аппаратуру, за которую несет полную ответственность, и после окончания работы сдает их преподавателю.

Преподаватель выдает инструкционные карты обучающимся и проводит вводный инструктаж с пояснением всего процесса выполнения практической работы. Обучающиеся могут задавать вопросы по процессу выполнения практических работ. Только после отсутствия вопросов обучающиеся могут приступать к выполнению практической работы.

6. Во время занятий в учебном кабинете (лаборатории) должны поддерживаться надлежащий порядок и деловая обстановка. Не разрешается перемещение обучающихся по учебному классу (лаборатории) и подключение используемой аппаратуры или компьютеров без разрешения преподавателя. Ответственность за поддержание порядка несет преподаватель, староста группы и дежурный студент.

7. При выполнении практических работ требуется неукоснительное выполнение правил техники безопасности.

8. По окончании работы в черновой тетради каждого обучающегося преподаватель делает отметку о правильном и полном выполнении работы. В случае неправильности полученных результатов работа повторяется.

9. По выполненным практическим работам каждый обучающийся оформляет и защищает отчет. Эта работа планируется в конце текущего практического занятия. Однако если подготовка обучающегося к занятию недостаточна, и практическая работа потребовала слишком много времени, то защита отчета о выполненной работе переносится на следующее практическое занятие или на консультацию. Отчет по выполненной практической работе может быть выполнен как в электронном, так и в рукописном варианте.

10. Пропущенные практические работы отрабатываются во внеурочное время согласно установленному графику.

11. Правила техники безопасности.

Практически все напряжения, с которыми приходится иметь дело в учебном кабинете (лаборатории) электроники и цифровой схемотехники, являются опасными для жизни человека, поэтому при выполнении практических работ необходимо строго выполнять следующие правила безопасности.

1. Электрические цепи должны собираться только при отключенном источнике питания.

2. Собранная схема должна быть проверена преподавателем. Включать схему можно только с разрешения преподавателя.

3. Перед включением электрической цепи все её элементы (реостаты, регулируемые источники, реактивные катушки и т.д.) должны находиться в таком положении, чтобы ток при включении схемы был наименьшим.

4. Все обучающиеся должны быть ознакомлены с системой аварийного отключения источников питания.

5. В учебном кабинете (лаборатории) запрещается:

5.1. Включать схему или персональные компьютеры с программным обеспечением без предупреждения. Перед включением необходимо громко сказать: «Подаю напряжение!».

5.2. Прикасаться к неизолированным частям установки, находящейся под напряжением.

5.3. Производить присоединения в электрической цепи, находящейся под напряжением. Всякие присоединения выполняются при отключенном питании и после каждого присоединения схема вновь должна быть проверена преподавателем.

5.4. Загромождать лабораторные столы посторонними предметами: приборами и аппаратами, не предназначенными для выполнения данной работы, лишними соединительными проводами, сумками, книгами и т.д.

5.5. Оставлять без наблюдения установку, находящуюся под напряжением.

5.6. Разбирать самостоятельно без преподавателя электрическую цепь или персональный компьютер.

5.7. Включать электрическую цепь в случае, когда один из концов монтажных проводов остается свободным.

5.8. Самому ремонтировать приборы, аппараты и другое электрооборудование.

5.9. Работать одному без преподавателя с электроустановками.

6. В случае аварии на рабочем месте (повреждение приборов, перегорание предохранителей и т.п.) обучающийся обязан немедленно отключить питание на рабочем месте и сообщить преподавателю о случившемся.

7. О всех неисправностях срочно сообщать преподавателю.

3. ПОРЯДОК СОСТАВЛЕНИЯ ОТЧЕТА О ПРАКТИЧЕСКОЙ РАБОТЕ

Составление отчета о практической работе имеет своей целью выработать у обучающегося навыки четкого и грамотного изложения результатов любого технического исследования.

Отчет о выполненной практической работе составляется обучающимся на основе записей, сделанных в личной черновой тетради в процессе подготовки и выполнения работы. Отчет оформляется на скрепленных листах стандартного формата А4. Допускается оформление отчета на развернутых листах ученических тетрадей в клетку.

Отчет должен содержать:

- титульный лист, на котором будет записано наименование практической работы, фамилия и инициалы обучающегося, дата проведения практической работы;
- изложение цели работы и её реализации в данной практической работе, включая исследуемые схемы, таблицы результатов экспериментов, требуемые расчеты, графики и диаграммы;
- заключение или краткие выводы по итогам работы.

Цель работы и её реализация должны быть изложены кратко, последовательно, с разбивкой на завершенные в смысловом отношении части, соответствующие выполненным в работе экспериментам или решенным задачам. Пояснения к работе, имеющиеся в методических указаниях, в отчете приводить не следует. Отчет должен быть составлен именно по выполнению работы.

Рабочие схемы, таблицы результатов, графики и диаграммы должны иметь наименования и краткие пояснения типа: “по данным табл. 2 построена
зависимость $U_{вых}(I_н)$ - см. рис. 2” и т.д.

Особое внимание следует уделить графической части отчета. Схемы, графики, диаграммы рекомендуется выполнять карандашом, применяя чертежные принадлежности и соблюдая требования ГОСТ и ЕСКД. Изображение элементов на схемах, как правило, должно быть вертикальным или горизонтальным. Размеры схем и графиков не следует чрезмерно растягивать или, напротив, выполнять их слишком мелкими. Вполне удобны и достаточны для наглядности рисунки размером 8∗10 см, т.е. в четверть тетрадного листа. В некоторых случаях графики можно выполнять на отдельных листах миллиметровой бумаги стандартных форматов. Рекомендуемые по ГОСТ масштабы: 1:1, 1:2, 1:5, 1:10. Так же схемы, графики и другие чертежи могут быть выданы преподавателем или выполнены с помощью компьютерных программ для обучающихся с ограниченными возможностями здоровья.

Итоги проделанной работы могут быть изложены в форме заключения по работе или в виде кратких выводов, которые бы конкретно, со ссылками на таблицы или графики, отвечали пунктам цели работы.

Отчет должен быть подписан обучающимся.

Перечень практических работ

Практическая работа №1 «Замер и расчет ВАХ характеристик электронных приборов». -2 часа

Практическая работа №2 «Расчет пропускной способности канала». -2 часа

Практическая работа №3 «Элементарные логические функции и формы их представления». -1 час

Практическая работа №4 «Алгебра логики. Построение комбинационных схем в заданном базисе». -1 час

Практическая работа №5 «Шифраторы. Дешифраторы». -1 час

Практическая работа №6 «Мультиплексоры. Демультиплексоры». -1 час

Практическая работа №7 «Цифровые компараторы. Сумматоры. Триггеры». -1 час

Практическая работа №8 «Регистры. Счетчики». -1 час

Практическая работа №9 «Классификация и характеристики запоминающих устройств на основе микросхем». -1 час

Практическая работа №10 «Современные модули памяти ОЗУ». -1 час

Практическая работа №11 «Аналого-цифровые преобразователи». -2 часа

Практическая работа №12 «Применение цифро-аналоговых и аналого-цифровых преобразователей». -1 час

Практическая работа №1(2часа)

Ознакомление с электроизмерительными приборами и измерениями электрических величин.

1. Цель работы.

Изучение электроизмерительных приборов, используемых в лабораторных работах, выполняемых на стенде. Получение представлений о пределе измерения и цене деления, абсолютной и относительной погрешности, условиях эксплуатации и других характеристиках стрелочных электроизмерительных приборов, получение навыков работы с цифровыми измерительными приборами.

2. Краткие теоретические сведения.

Контроль работы электрооборудования осуществляется с помощью разнообразных электроизмерительных приборов. Наиболее распространенными электроизмерительными приборами являются приборы непосредственного отсчета. По виду отсчетного устройства различают аналоговые (стрелочные) и цифровые измерительные приборы.

На лицевой стороне стрелочных приборов изображены условные обозначения, определяющие классификационную группу прибора. Они позволяют правильно выбрать приборы и дают некоторые указания по их эксплуатации.

В цепях постоянного тока для измерений токов и напряжений применяются в основном приборы магнитоэлектрической системы. Принцип действия таких приборов основан на взаимодействии магнитного поля постоянного магнита и измеряемого тока, протекающего по катушке. Угол поворота стрелки α прямо пропорционален измеряемому току I: $\alpha = K \times I$. Шкалы магнитоэлектрических приборов равномерные.

В измерительных механизмах электромагнитной системы, применяемых для измерений в цепях переменного и постоянного тока, вращающий момент обусловлен действием магнитного поля измеряемого тока в неподвижной катушке прибора на подвижный ферромагнитный якорь. Угол поворота стрелки α здесь пропорционален квадрату тока: $\alpha = K \times 2I$. Поэтому шкала электромагнитных приборов обычно неравномерная, что является недостатком этих приборов. Начальная часть шкалы не используется для измерений. Для измерений токов и напряжений в цепях переменного тока применяются также приборы выпрямительной системы. Такие приборы содержат выпрямительный преобразователь и магнитоэлектрический измерительный механизм. Они имеют более линейную шкалу, чем приборы электромагнитной системы и достаточно широкий частотный диапазон.

Для практического использования стрелочного измерительного прибора необходимо знать его предел измерений (номинальное значение) и цену деления (постоянную) прибора. **Предел измерений** – это наибольшее значение электрической величины, которое может быть измерено данным прибором. Это значение обычно указано на лицевой стороне прибора. Один и тот же прибор может иметь несколько пределов измерений. Ценой деления прибора называется значение измеряемой величины, соответствующее одному делению шкалы прибора. Цена деления прибора - C легко определяется как отношение предела измерений $A_{НОМ}$ к числу делений шкалы N:

$C = A_{НОМ} / N$.

На лицевой стороне стрелочных прибора указывается класс точности, который определяет приведенную относительную погрешность прибора $\gamma_{ПР}$.

Приведенная относительная погрешность прибора – это выраженное в процентах отношение максимальной для данного прибора абсолютной погрешности ΔA к номинальному значению прибора (пределу измерений) $A_{НОМ}$:

$$\gamma_{ПР} = \frac{\Delta A}{A_{НОМ}} \cdot 100\%$$

Промышленность в соответствии с ГОСТ выпускает приборы с различными классами точности (0,05; 0,1; 0,2; 0,5; 1,5; 2,5; 4,0).

Зная класс точности прибора, можно определить абсолютную ΔA и относительную погрешности измерения $\gamma_{ИЗМ}$, а также действительное значение измеряемой величины $A_Д$:

$$\Delta A = \frac{\gamma_{ПР} \cdot A_{ном}}{100};\ \gamma_{ИЗМ} = \frac{100 \cdot \Delta A}{A};\ A_Д = A \pm \Delta A.$$

Расчетную относительную погрешность измерения в любой точке шкалы прибора можно определить, полагая, что его допустимая абсолютная погрешность ΔA известна и постоянна:

$$\gamma_{ИЗМ} = \frac{100 \cdot \Delta A}{A_{ИЗМ}},$$

где $A_{ИЗМ}$ – условное измеренное значение величины, задаваемое в пределах

шкалы прибора от минимального значения до номинального значения данного прибора. Обратить внимание на значение относительной погрешности измерения, соответствующее предельному значению измеряемой величины, и сравнить его с классом точности прибора.

Нетрудно сделать вывод, что относительная погрешность измерения тем больше, чем меньше измеряемая величина по сравнению с номинальным значением прибора. Поэтому желательно не пользоваться при измерении начальной частью шкалы стрелочного прибора.

Для обеспечения малой методической погрешности измерения необходимо, чтобы сопротивление амперметра было значительно меньше сопротивления нагрузки, а сопротивление вольтметра было значительно больше сопротивления исследуемого участка.

В табл. 1 приведены некоторые условные обозначения, приводимые на лицевых панелях стрелочных измерительных приборов, определяющие их свойства и условия эксплуатации.

При проведении измерений в электрических цепях широкое применение получили цифровые измерительные приборы, например мультиметры – комби-

нированные цифровые измерительные приборы, позволяющие измерять постоянное и переменное напряжение, постоянный и переменный ток, сопротивления, проверять диоды и транзисторы. Представление результата измерения происходит на цифровом отсчетном устройстве в виде обычных удобных для считывания десятичных чисел. Наибольшее распространение в цифровых отсчетных устройствах мультиметров получили жидкокристаллические и светодиодные индикаторы. В лабораторном стенде используются цифровые приборы для измерения постоянных и переменных токов, а также цифровой измеритель мощности. Для переключения режима работы цифровых амперметров стенда (PA1, PA2, PA3 и PA4) на его передней панели установлен тумблер, который для измерения постоянного тока следует установить в позицию «=», для измерения действующих значений переменных токов – в позицию «~». Для измерения постоянного тока входная клемма (+) цифрового амперметра выделена красным цветом.

Цифровой измеритель мощности предназначен для измерения параметров электрической цепи:
– действующего значения напряжения U (True RMS) в диапазоне 0…30 В;

– действующего значения тока I (True RMS) в диапазоне 0…300 мА;

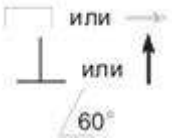
– активной мощности P в диапазоне 0…600 Вт;

– частоты f в диапазоне 35…400 Гц;

– cos ϕ;

– угла сдвига фаз ϕ (Fi) между током и напряжением.

Таблица 1

Условное графическое обозначение	**Содержание условного обозначения**
A, V, W, Ω, Hz, cosφ, F, H	Наименование измеряемой величины (ампер, вольт, ватт, ом, герц, коэффициент мощности, фарада, генри)
	Магнитоэлектрический измерительный механизм
	Электромагнитный измерительный механизм
	Магнитоэлектрический измерительный механизм с выпрямителем
0,05; 0,1; 0,2; 0,5; 1,5; 2,5; 4,0	Класс точности прибора
	Рабочее положение шкалы прибора: горизонтальное; вертикальное; под углом, например 60°
	Прибор предназначен для работы в цепи постоянного тока;

— ~ ≂ ≋	переменного тока; постоянного и переменного; в трехфазной цепи переменного тока
А Б В1; В2; В3	А (или отсутствие буквы) – прибор для сухих отапливаемых помещений с температурой +10°С …+35°С и влажности до 80% при 30°С; Б – прибор для закрытых не отапливаемых помещений с температурой - 30°С …+40°С и влажности до 90% при 30°С; В – приборы для полевых и морских условий: В1 – при температуре -40°С … +50°С и В2 – при температуре -50°С … +60°С и влажности до 95% при 35°С; В3 – при температуре -40°С … +50°С и влажности до 98% при 40°С
	Измерительная цепь прибора изолирована от корпуса и испытана напряжением, например, 2 кВ
30 – 200 Hz	Рабочий частотный диапазон прибора

Прибор содержит:

– клеммы подачи входного измеряемого сигнала (генератора): клемму «Вх» и общую клемму, клеммы подключения потребителя (нагрузки): клемму «Вых» и общую клемму. Шунт для измерения тока нагрузки подключен между клеммами «Вх» и «Вых»;

– жидкокристаллический четырехстрочный индикатор для вывода информации;

– кнопку «f/cosϕ/ϕ» изменения вывода информации в четвертой строке индикатора (соответственно, частоты, коэффициента мощности cosϕ или угла сдвига фаз Fi между током и напряжением).

С задней стороны прибора установлены розетка для подключения питания сети и колодка предохранителя.

С помощью кнопки «f/cosϕ/ϕ» можно изменять вывод информации в четвертой строке индикатора. Для вывода требуемого параметра в четвертой строке индикатора кнопку необходимо нажать на 1…2 секунды.

Изменения схемы подключения прибора и лабораторной установки выполнять при выключенном питании прибора. В противном случае возможны изменения показаний прибора, а также возникновение нарушений в работе индикатора прибора.

3. Порядок выполнения работы.

3.1. Изучение паспортных характеристик стрелочных электроизмерительных приборов. Для этого внимательно рассмотреть лицевые панели стрелочных амперметров и заполнить табл. 2.

Таблица 2

Характеристика электроизмерительного прибора		
Наименование прибора	**Вольтметр №1**	**Вольтметр №2**
Система измерительного механизма		
Предел измерения		
Цена деления		
Минимальное значение измеряемой величины		
Класс точности		
Допустимая максимальная абсолютная погрешность		
Род тока		
Нормальное положение шкалы		
Прочие характеристики		

3.2. Построить график зависимости относительной погрешности измерения от измеряемой величины $\gamma_{изм} = f(A_{изм})$ для прибора, указанного преподавателем. Сделать вывод о величине относительной погрешности измерения в начальной и конечной части шкалы, о характере изменения погрешности вдоль шкалы прибора.

3.3. Измерить величину сопротивления, заданного преподавателем, методом амперметра и вольтметра. Для этого собрать электрическую цепь по рис. 1. Установить тумблер режима работы измерителя тока в позицию «=».

После проверки схемы, включить электропитание и занести полученные данные в табл. 3. Выключить электропитание. Рассчитать, используя закон Ома, величину заданного сопротивления R. Результат занести в табл. 3.

Таблица 3

U, В	**I, мА**	**R, Ом**

4. Содержание отчета

Отчет по работе должен содержать:

а) наименование работы и цель работы;

б) технические данные измерительных приборов;

в) график зависимости относительной погрешности измерений

$\gamma_{изм} = f(A_{изм})$;

г) результаты измерений;

д) выводы по работе.

5. Контрольные вопросы

1. Каков принцип действия приборов магнитоэлектрической и электромагнитной систем?

2. Что такое предел измерения?

3. Как определяется цена деления прибора?

4. Что такое абсолютная и относительная погрешности измерения?

5. Что характеризует класс точности прибора?

6. В какой части шкалы прибора измерение точнее и почему?

7. Каковы основные достоинства цифровых измерительных приборов?

8. Как можно измерить величину сопротивления резистора?

Практическая работа №2 «Расчет пропускной способности канала».(2 часа)

Цель работы: Выполнить расчет пропускной способности канала передачи, в зависимости от установок передатчика. Выполнить расчет зоны покрытия при определенных требованиях к пропускной способности канала передачи.

1. Методика проведения эксперимента

При проведении данной работы используются лабораторный макет, формирователь DVB-T, аттенюатор 40дБ для ослабления сигнала формирователя, сумматор для суммирования сигнала формирователя с сигналами эфирного вещания, ступенчатый аттенюатор для регулируемого ослабления сигналов, анализатор сигналов IT-15T2, ПК (рисунок 1).

Рисунок 1. Схема эксперимента

Главное содержание лабораторной работы состоит в следующем:

а) во-первых, научиться рассчитывать пропускную способность сигнала DVB-T в зависимости от установок формирователя.

б) во-вторых, провести эксперимент и рассчитать зону покрытия сигнала DVB-T при определенных условиях по пропускной способности и помехоустойчивости канала передачи.

Расчет пропускной способности канала

Каждый символ COFDM состоит из несущих, каждая из которых кодирует 2, 4 или 6 бит (QPSK, 16QAM или 64QAM).

Число несущих, передающих полезную информацию в стандарте DVB-T, зависит только от режима и равно 1512 для режима 2k и 6048 для режима 8k.

Длительность передачи символа COFDM в режиме 8k составляет 896 мкс, в режиме 2k длительность символа в 4 раза меньше и составляет 224 мкс.

Таким образом, поскольку число "полезных" несущих в обоих режимах отличается ровно в четыре раза и длительность символа COFDM при переходе от режима к режиму также меняется в четыре раза, то такой важный параметр, как частота следования символов данных RS, оказывается в двух режимах одинаковым и равным 6,75 миллионам символов в секунду (RS=1512/224мкс=6048/896мкс=6,75МГц=6,75Мегасимвол/с).

Используя величину RS, нетрудно найти скорость передачи данных в разных режимах и при различных сочетаниях параметров системы DVB-T: RSU = RS * b * FEC * CRRS * (TU/TS) (здесь b - количество битов, передаваемых в одном символе

с помощью одной несущей, FEC - скорость внутреннего сверточного кода, CRRS - скорость внешнего кода Рида-Соломона, равная 188/204, (TU/TS) - отношение длительности полезного интервала к общей длительности символа.

Таблица 1. Длительность символов для разных параметров модуляции

Режим	Защитный интервал	Длительность символа, мкс	Длительность защитного интервала, мкс	Суммарная длительность символа и защитного интервала, мкс
8к	1/4	896	224	1120
	1/8		112	1008
	1/16		56	952
	1/32		28	924
2к	1/4	224	56	280
	1/8		28	252
	1/16		14	238
	1/32		7	231

Расчет зоны покрытия

В свободном пространстве напряжённость поля E уменьшается линейно с удалением от источника энергии. Если, предположим, источник энергии даёт на расстоянии 1 км напряжённость E=1000 мкВ/м, то на расстоянии 10 км напряжённость составит 100 мкВ/м, на расстоянии 1000 км - 1 мкВ/м».

Для расчета расстояния, на котором прием сигнала формирователя будет удовлетворять минимальным требованиям по значению PostVBER примем следующие условия.

Пусть текущий уровень сигнала формирователя без ослаблений является уровнем сигнала на расстоянии 1 километре от передатчика. Таким образом, зная начальный уровень сигнала на расстоянии 1 км от передатчика и вносимое ослабление ступенчатым аттенюатором, можем привязать значение ослабления к расстоянию.

Пример: На расстоянии 1 км имеем уровень сигнала 60дБмкв, на расстоянии 2км уровень сигнала станет ниже в 2 раза или на 6дБ по напряжению. Таким образом, на расстоянии 2 км уровень сигнала будет 60 – 6 = 54дБмкв. При ослаблении сигнала на 12дБ, получим расчетное расстояние около 4 км.

Таблица 2. Перевод значений децибел в отношении напряжений и мощности

дБ	**U2/U1**	**P2/P1**	**дБ**	**U2/U1**	**P2/P1**
0	1	1	28	25,12	631
1	1,12	1,26	29	28,17	794
2	1,26	1,59	30	31,64	1000
3	1,41	2	31	35,46	1257
4	1,59	2,51	32	39,84	1587

5	1,78	3,16	33	44,64	1993
6	2	3,98	34	48,08	2312
7	2,24	5,01	35	56,82	3165
8	2,51	6,31	36	63,29	4006
9	2,82	7,94	37	70,92	5030
10	3,16	10	38	79,36	6298
11	3,55	12,59	39	89,29	7973
12	3,98	15,85	40	100	10000
13	4,47	19,96	41	112,23	12596
14	5,01	25,12	42	125,94	15861
15	5,62	31,65	43	141,24	19949
16	6,31	39,84	44	158,48	25116
17	7,08	48,08	45	177,94	31663
18	7,94	63,29	46	199,60	39840
19	8,91	79,36	47	223,71	50046
20	10	100	48	251,26	63132
21	11,22	125,94	49	281,69	79349
22	12,59	158,48	50	316,5	100000
23	14,12	199,60	60	1000	1000000
24	15,85	251,26	70	3165	10000000
25	17,79	316,50	80	10000	100000000
26	19,96	398,40	90	31650	1000000000
27	22,37	500,42	100	100000	10000000000

2. Порядок выполнения работы

2.1.Запустить на ПК программу ItToolsT2.

2.1.1. Перевести анализатор IT-15T2 в режим «работа с ПК».

2.1.2. Выбрать основной сигнал формирователя.

2.1.3. Установить параметры формирователя для минимальной пропускной способности сигнала.

2.1.4. Зафиксировать в отчете параметры, основного сигнала формирователя (уровень, MER, PreVBER, PostVBER, защитный интервал, кодовую скорость, тип модуляции, количество несущих).

2.1.5. Рассчитать скорость передачи полезных данных формирователем в режиме минимальной пропускной способности, занести расчетное значение в отчет.

2.1.6. При помощи ступенчатого аттенюатора вносить постепенное ослабление в сигнал, по достижению уровня сигнала 38 дБмкв, зафиксировать параметры основного сигнала формирователя (уровень, MER, PreVBER, PostVBER) и значение ослабления, вносимое ступенчатым аттенюатором.

2.1.7. Выполнить расчет предполагаемого расстояния, на котором значение уровня сигнала будет равно 38 дБмкв. Зафиксировать в отчете расстояние, на которое может быть передано расчетное значение полезной информации из п.2.1.5 в режиме минимальной пропускной способности формирователя.

2.1.8. Установить параметры формирователя для максимальной пропускной способности сигнала

2.1.9. Рассчитать скорость передачи полезных данных формирователем в режиме максимальной пропускной способности, занести расчетное значение в отчет.

2.1.10. При помощи ступенчатого аттенюатора вносить постепенное ослабление в сигнал, по достижению значения PostVBER в интервале от $1*10^{-5}$ до $5*10^{-4}$, зафиксировать параметры основного сигнала формирователя (уровень, MER, PreVBER, PostVBER) и значение ослабления, вносимое ступенчатым аттенюатором.

2.1.11. Выполнить расчет предполагаемого расстояния, на котором значение PostVBER будет в интервале от $1*10^{-5}$ до $5*10^{-4}$. Зафиксировать в отчете расстояние, на которое может быть передано расчетное значение полезной информации.

2.2. Необходимо выполнить передачу 20 Мбит полезной информации на максимальное расстояние.

2.2.1. Расчетным и экспериментальным путем определить наиболее оптимальные параметры формирователя для наилучшего соотношения пропускная способность/помехоустойчивость сигнала при условии значения PostVBER <= в интервале от $1*10^{-5}$ до $5*10^{-4}$.

2.2.3. При помощи ступенчатого аттенюатора вносить постепенное ослабление в сигнал с шагом 3дБ, фиксируя параметры принимаемого канала в таблицу 3.

Таблица 3. Оптимальные параметры формирователя для передачи 20 Мбит информации на максимальное расстояние

Параметры формирователя	Тип модуляции		Количество несущих		Защитный интервал		Кодовая скорость		Пропускная способность	
ослабление дБ	0	3	6	9	12	15	18	21	24	27
d, расстояние от передатчика										
U, уровень, дБмкВ										
MER, дБ										
preVBER										
postVBER										

3. Содержание отчёта

1. Схема макета.
2. Методика проведения эксперимента с краткими пояснениями.
3. Расчетные и экспериментальные данные.
4. Выводы по каждому пункту выполненной работы.

1. Контрольные вопросы

1. Как количество несущих в стандарте DVB-T влияет на пропускную способность канала?
2. Как значение кодовой скорости влияет на пропускную способность канала?
3. Какая максимальная пропускная способность доступна для передачи в стандарте DVB-T?

Практическая работа №3 «Элементарные логические функции и формы их представления».

Цель занятия:

- учить составлять таблицы истинности для логических функций;
- учить строить функциональной схемы по логической формуле;
- учить составлять логическую формулу по функциональной схеме.

Пояснение к работе.

На основе анализа логической связи между высказываниями делается логический вывод. Для получения логического вывода составляется *таблица истинности*, в которой записывают все возможные комбинации каждого простого высказывания.

Работа ЭВМ как автоматических устройств основана исключительно на математически строгих правилах выполнения команд, программ и интерпретации данных. Тем самым работа компьютеров допускает строгую однозначную проверку правильности своей работы в плане заложенных в них процедур и алгоритмов обработки информации. Это позволяет использовать математический аппарат для анализа и разработки логических устройств вычислительной техники.

Функцией логических переменных называют взаимосвязь логических переменных по законам логики. Значения входных переменных и выходных функций связаны некоторым преобразованием, которое реализует логическую функцию.

Логическая операция – способ построения сложного высказывания из данных высказываний, при котором значение истинности сложного высказывания полностью определяется значениями истинности исходных высказываний.

Рассмотрим три базовых логических операций – инверсию, конъюнкцию, дизъюнкцию и дополнительные – импликацию и эквивалентность (таблица №1).

В алгебре логики любую логическую функцию можно выразить через основные логические операции, записать ее в виде логического выражения и упростить ее, применяя законы логики и свойства логических операций. По формуле логической функции легко рассчитать ее таблицу истинности. Необходимо только учитывать порядок выполнения логических операций (приоритет) и скобки. Операции в логическом выражении выполняются слева направо с учетом скобок.

Приоритет выполнения логических операций:

1. Действие в скобках; 2. Действие под отрицанием; 3. Инверсия; 4. Конъюнкция; 5. дизъюнкция.

Таблица №1

Логическая операция	Название	**Соответствует союзу**	**Обозначение знаками**	**Таблица истинности**	**Логическая операция**
Инверсия (от лат. inversion – переворачиваю)	отрицание	не А	$\neg A$	A | $\neg A$ 1 | 0 0 | 1	Опр. 8 Инверсия логической переменной истина, если переменная ложна, и, наоборот, инверсия ложна, если переменная истинна.
Конъюнкция (от лат. conjunction – связываю)	Логическое умножение	А и В	$A \wedge B$	A B | $A \wedge B$ 1 1 | 1 1 0 | 0 0 1 | 0 0 0 | 0	Опр.9Конъюнкция двух логических переменных истинна тогда и только тогда, когда оба высказывания, истинны.
Дизъюнкция (от лат. disjunction – различаю)	Логическое сложение	А или В	$A \vee B$	A B | $A \vee B$ 1 1 | 1 1 0 | 1 0 1 | 1 0 0 | 0	Опр. 10 Дизъюнкция двух логических переменных ложна тогда и только тогда, когда оба высказывания ложны.
Импликация (от лат. implication – тесно связывать)	Логическое следование	Если А, то В; Когда А, тогда В	$A \rightarrow B$ А–условие В-следствие	A B | $A \rightarrow B$ 1 1 | 1 1 0 | 0 0 1 | 0 0 0 | 1	Опр. 11 Импликация двух логических переменных ложна тогда и только тогда, когда из истинного основания следует ложное следствие.
Эквивалентность (от лат. equivalents - равноценность)	Логическое равенство	А тогда и только тогда, когда В	$A \Leftrightarrow B$	A B | $A \Leftrightarrow B$ 1 1 | 1 1 0 | 0 0 1 | 0 0 0 | 1	Опр. 12 Эквивалентность двух логических переменных истинна тогда и только тогда, когда оба высказывания одновременно либо ложны, либо истинны

Пример1.

Построить таблицу истинности для логической функции $F = A \wedge (B \vee \overline{A})$

Решение:

1. Определить *количество строк* в таблице истинности, которое равно количеству возможных комбинаций значений логических переменных, входящих в логическое выражение: количество строк равно 2^n, где n – количество переменных

Количество логических переменных – 2 (*A*, *B*) поэтому количество строк – $2^n = 4$

A	**B**
0	0
0	1
1	0
1	1

2. Определить *количество столбцов*:
   количество столбцов=количество переменных + количество операций, используя приоритет операций.

Количество логических операций -3 (умножение – 1, сложение – 1, отрицание – 1), поэтому количество столбцов 2+3=5

3. *Построить таблицу истинности* с указанным количеством строк и столбцов, обозначить столбцы и внести возможные наборы значений исходных логических переменных и заполнить таблицу истинности по столбцам, выполняя базовые логические операции в необходимой последовательности и в соответствии с их таблицами истинности:

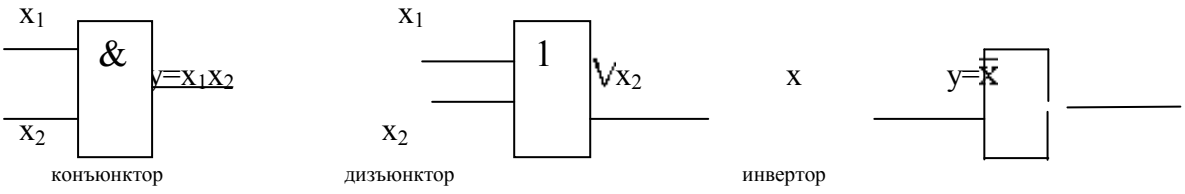
A	**B**	$\overline{A}$	$B \vee \overline{A}$	F
0	0	1	1	0
0	1	1	1	0
1	0	0	0	0
1	1	0	1	1

Арифметико-логические основы работы компьютера

Все логические схемы ЭВМ строятся на логических элементах, реализующих логические функции. Под логическим элементом понимают техническое устройство, реализующее одну элементарную булеву функцию.

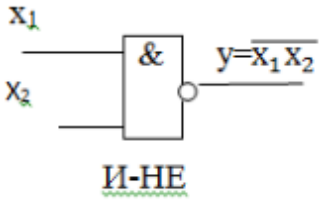
Обычно для построения схем используются логические элементы, реализующие основные логические функции И, ИЛИ, НЕ: конъюнктор (схема И); дизъюнктор (схема ИЛИ), инвертор (схема НЕ). Стандартные обозначения этих элементов выглядят следующим образом:

элементы

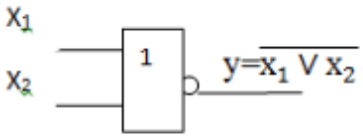


Конструктивно логические объединяются в единые корпуса – интегральные микросхемы (ИМС). В одном корпусе ИМС могут быть один, две и более логических элементов.

Схемы И, ИЛИ, НЕ образуют базис, то есть с их помощью можно реализовать любую булеву функцию на основе суперпозиции. Применяемые на практике комплексы ИМС имеют в своём составе такие логические элементы, как И-НЕ, ИЛИ-НЕ, которые образуют универсальный базис. Стандартные схемы этих элементов имеют следующий вид:

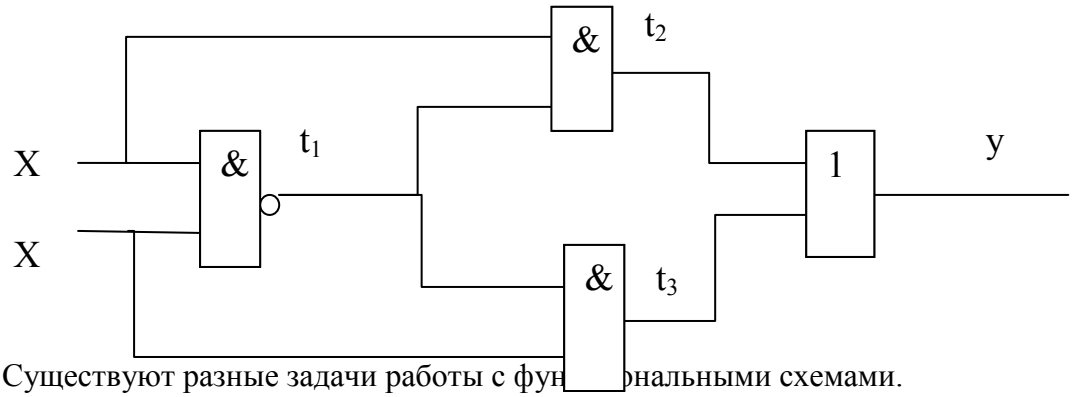


И-НЕ



ИЛИ-НЕ

Схема, показывающая связи между различными логическими элементами, называется **функциональной схемой**. Рассмотрим на рисунке функциональную схему.



Существуют разные задачи работы с функциональными схемами.

Задача1. Найти по функциональной схеме логическую функцию, описывающую работу этой схемы с помощью аппарата алгебры логики. В качестве примера построим логическую функцию, представленной на рисунке выше.

Для этого каждому логическому элементу схемы поставим в соответствие логический оператор:

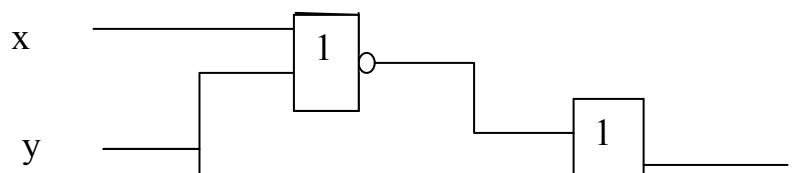
$$t_1 = \overline{x_1 x_2}\,; \qquad t_2 = x_1 \cdot t_1\,; \qquad t_3 = x_2 \cdot t_1\,; \qquad y = t_2 \vee t_3$$

Этим установлено однозначное соответствие между элементами схемы и её математическим описанием. Затем выполним последовательно операции подстановки и преобразования до тех пор, пока не получится функция, выраженная через входные переменные:

$$y = t_2 \vee t_3 = x_1 \cdot t_1 \vee x_2 \cdot t_1 = x_1 \cdot \overline{x_1 x_2} \vee x_2 \cdot \overline{x_1 x_2} = x_1 \cdot (\overline{x_1} \vee \overline{x_2}) \vee x_2 \cdot (\overline{x_1} \vee \overline{x_2})$$
$$= x_1 \overline{x_2} \vee \overline{x_1} x_2$$

Задача 2. Построить функциональную схему по логической функции $f = \overline{x \vee y} \vee \overline{y \cdot z}$. Затем построить схему, используя как базисные, так и дополнительные вентили, следуя по приоритету операций.

Построим схему:



Вопросы для размышления

1. Для чего нужно изучать функциональные схемы в информатике?

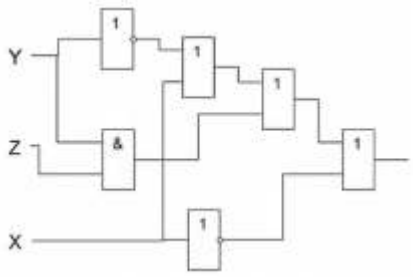
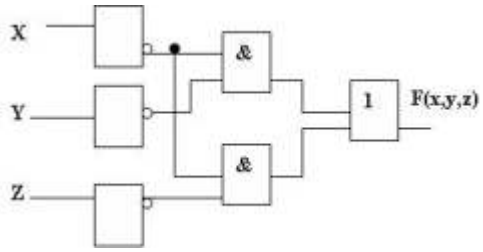
2. Как правильно написать логическую функцию по функциональной схеме?

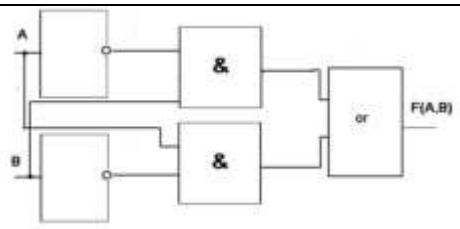
3. Как правильно составить функциональную схему?

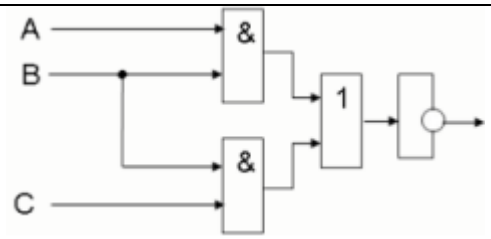
4. В чем процесс составления таблицы истинности по функциональной схеме?

Задания

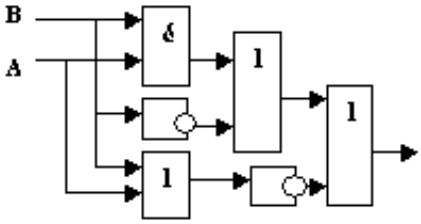
1. Построить таблицу истинности для логической функции;
2. Проверить справедливость равенств;
3. Построить по логической функции функциональную схему;
4. Построить по логической функции функциональную схему;
5. По логической схеме записать логическую функцию.

Вариант 1	**Вариант 2**
1. $F = (A \vee B) \cdot (\overline{A} \vee \overline{B})$ 2. $A \vee (\overline{A} \cdot B) = A \vee B$ 3. $(\mathrm{A} \cdot \overline{\mathrm{B}}) \vee \mathrm{B}$ 4. $\overline{\overline{\mathrm{A}} \vee \mathrm{B}} \cdot (\overline{\mathrm{B}} \vee \mathrm{A})$ 5. 	1. $F = (\overline{A} \cdot \overline{B}) \vee (A \cdot B)$ 2. $\overline{A} \vee (A \cdot B) = \overline{A} \vee B$ 3. $(\overline{\mathrm{A}} \vee \overline{\mathrm{B}}) \cdot \mathrm{B}$ 4. $\mathrm{B} \cdot \overline{(\mathrm{A} \vee \mathrm{B})} \cdot \overline{\overline{\mathrm{B}} \cdot \mathrm{A}}$ 5. 
Вариант 3	**Вариант 4**
1. $F = A \vee \overline{B} \cdot (\overline{A \vee B})$ 2. $A \cdot (\overline{A} \vee B) = A \cdot B$ 3. $(\overline{\mathrm{A}} \vee \mathrm{B}) \cdot \mathrm{A}$ 4. $\overline{(\mathrm{A} \cdot \mathrm{B})} \cdot ((\overline{\mathrm{A}} \cdot \mathrm{B}) \vee (\mathrm{A} \vee \mathrm{B}))$ 5. .	1. $F = \overline{A} \cdot B \vee (\overline{A \cdot B})$ 2. $\overline{A} \cdot (A \vee B) = \overline{A} \cdot B$ 3. $(\overline{\mathrm{A}} \cdot \overline{\mathrm{B}}) \vee \mathrm{A}$ 4. $(\mathrm{A} \cdot \overline{\mathrm{B}}) \vee (\overline{\overline{\mathrm{A}} \vee \mathrm{B}})$ 5. .

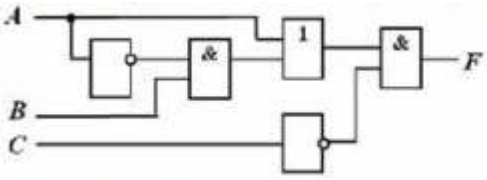




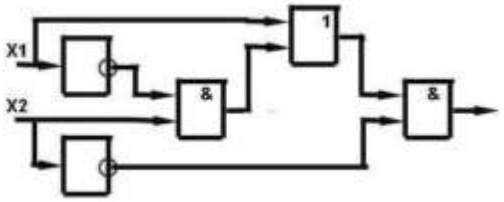
Вариант 5

1. $F = \overline{A} \vee (\overline{B \cdot A}) \vee B$
2. $A \vee (\overline{B} \cdot A \vee B) = A \vee B$
3. $(\overline{A} \cdot B) \vee A$
4. $\overline{\overline{B} \vee A} \cdot (\overline{A} \vee B)$
5. 

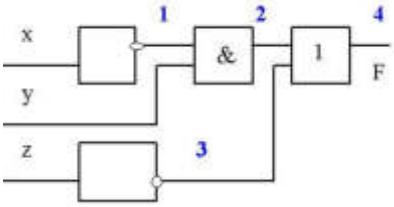
Вариант 6

1. $F = \overline{A} \vee (A \cdot B) \cdot \overline{B}$
2. $\overline{A} \vee (B \vee \overline{A} \cdot B) = \overline{A} \vee B$
3. $(A \vee \overline{B}) \cdot B$
4. $B \cdot \overline{(A \vee B)} \cdot \overline{\overline{A} \cdot B}$
5. 

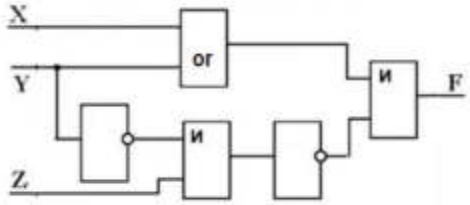
Вариант 7

1. $F = (A \vee B) \cdot (\overline{A} \vee \overline{B})$
2. $A \vee (\overline{A} \cdot B) = A \vee B$
3. $A \cdot (\overline{B} \vee A)$
4. $((\overline{B} \cdot A) \vee (B \vee A)) \cdot \overline{(B \cdot A)}$
5. 

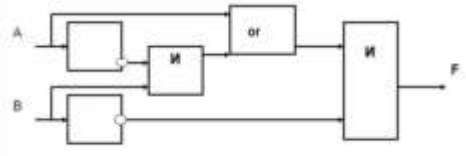
Вариант 8

1. $F = (\overline{A} \cdot \overline{B}) \vee (A \cdot B)$
2. $\overline{A} \vee (A \cdot B) = \overline{A} \vee B$
3. $B \vee (\overline{B} \cdot \overline{A})$
4. $(\overline{\overline{B} \vee A}) \vee (B \cdot \overline{A})$
5. 

Вариант 9

1. $F = A \vee \overline{B} \cdot (\overline{A \vee B})$
2. $A \cdot (\overline{A} \vee B) = A \cdot B$
3. $(\overline{A \cdot B}) \vee A$
4. $\overline{B \vee A} \cdot (\overline{A} \vee B)$
5. 

Вариант 10

1. $F = \overline{A} \cdot B \vee (\overline{A \cdot B})$
2. $\overline{A} \cdot (A \vee B) = \overline{A} \cdot B$
3. $(\overline{A \vee B}) \cdot B$
4. $B \cdot (A \vee \overline{B}) \cdot \overline{\overline{A} \cdot B}$
5. 

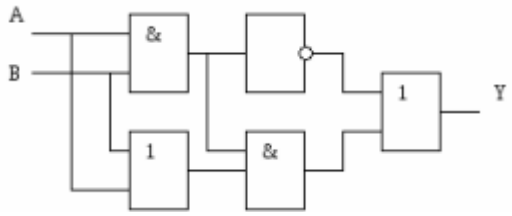
Вариант 11

1. $F = \overline{A} \vee (\overline{B \cdot A}) \vee B$
2. $A \vee (\overline{B} \cdot A \vee B) = A \vee B$

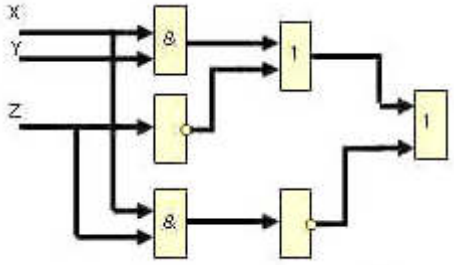
Вариант 12

1. $F = \overline{A} \vee (A \cdot B) \cdot \overline{B}$
2. $\overline{A} \vee (B \vee \overline{A} \cdot B) = \overline{A} \vee B$

3. $A \cdot (\overline{B} \vee \overline{A})$
4. $((\overline{B} \cdot A) \vee (\overline{B \vee A})) \cdot (B \cdot A)$
5.

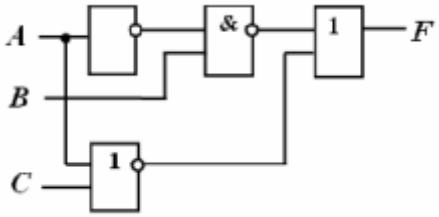


3. $B \vee (\overline{B \cdot A})$
4. $(\overline{\overline{B} \vee A}) \vee (\overline{B \cdot \overline{A}})$



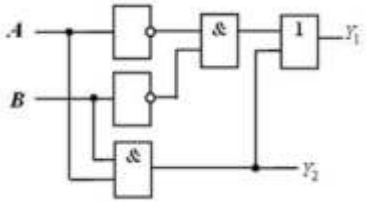
Вариант 13

1. $F = (A \vee B) \cdot (\overline{A} \vee \overline{B})$
2. $A \vee (\overline{A} \cdot B) = A \vee B$
3. $(A \cdot \overline{B}) \vee \overline{A}$
4. $(\overline{A} \vee B) \cdot \overline{B} \cdot \overline{B \vee A}$
5.



Вариант 14

1. $F = (\overline{A} \cdot \overline{B}) \vee (A \cdot B)$
2. $\overline{A} \vee (A \cdot B) = \overline{A} \vee B$
3. $\overline{B} \cdot (B \vee \overline{A})$
4. $(\overline{A} \cdot B) \vee \overline{B} \vee \overline{B \cdot A}$
5.



Отчет

Отчет должен быть оформлен в текстовом редакторе и содержать:

- наименование работы;
- цель работы;
- задание;
- последовательность выполнения работы;
- ответы на контрольные вопросы;
- вывод о проделанной работе.

Практическая работа №4 «Алгебра логики. Построение комбинационных схем в заданном базисе».

Цель работы:

- изучение основных понятий алгебры логики, составление таблиц истинности

Порядок выполнения работы:

1. Изучение методических рекомендаций
2. Выполнение практических заданий

Алгебра логики **— это раздел математики, изучающий высказывания, рассматриваемые со стороны их логических значений (истинности или ложности) и логических операций над ними.**

Алгебра логики возникла в середине XIX века в трудах английского математика **Джорджа Буля**. Ее создание представляло собой попытку решать традиционные логические задачи алгебраическими методами.

Что же такое логическое высказывание?

Логическое высказывание — это любое повествовательное предложение, в отношении которого можно однозначно сказать, истинно оно или ложно.

Высказывания, образованные из других высказываний с помощью логических связок, называются **составными.** Высказывания, не являющиеся составными, называются **элементарными.**

Каждая логическая связка рассматривается как операция над логическими высказываниями и имеет свое название и обозначение:

НЕ Операция, выражаемая словом *"не",* называется **отрицанием** и обозначается чертой над высказыванием (или знаком ¬). Высказывание $\overline{A}$ истинно, когда А ложно, и ложно, когда А истинно.

И Операция, выражаемая связкой *"и",* называется **конъюнкцией** (лат. conjunctio — соединение) или логическим умножением и обозначается точкой **" · "** (может также обозначаться знаками ∧ или **&**). Высказывание **A · B** истинно тогда и только тогда, когда оба высказывания **A** и **B** истинны.

ИЛИ Операция, выражаемая связкой *"или"* (в неисключающем смысле этого слова), называется **дизъюнкцией** (лат. disjunctio — разделение) или логическим сложением и обозначается знаком **v** (или плюсом). Высказывание **A v B** ложно тогда и только тогда, когда оба высказывания А и В ложны.

Таким образом, **операций отрицания, дизъюнкции и конъюнкции достаточно, чтобы описывать и обрабатывать логические высказывания.**

Порядок выполнения логических операций задается круглыми скобками. Но для уменьшения числа скобок договорились считать, что сначала выполняется операция отрицания ("не"), затем конъюнкция ("и"), после конъюнкции — дизъюнкция ("или") и в последнюю очередь — импликация.

Связь между алгеброй логики и двоичным кодированием

Математический аппарат алгебры логики очень удобен для описания того, как функционируют аппаратные средства компьютера, поскольку основной системой счисления в компьютере является двоичная, в которой используются цифры 1 и 0, а значений логических переменных тоже два: “1” и “0”.

Из этого следует два вывода:

1. одни и те же устройства компьютера могут применяться для обработки и хранения как числовой информации, представленной в двоичной системе счисления, так и логических переменных;
2. на этапе конструирования аппаратных средств алгебра логики позволяет значительно упростить логические функции, описывающие функционирование схем компьюте-

ра, и, следовательно, уменьшить число элементарных логических элементов, из десятков тысяч которых состоят основные узлы компьютера.

Логический элемент компьютера — *это часть электронной логической схемы, которая реализует элементарную логическую функцию*

Логическими элементами компьютеров являются электронные схемы И, ИЛИ, НЕ, И—НЕ, ИЛИ—НЕ и другие (называемые также **вентилями**), а также **триггер.**

С помощью этих схем можно реализовать любую логическую функцию, описывающую работу устройств компьютера. Обычно у вентилей бывает от двух до восьми входов и один или два выхода.

Чтобы представить два логических состояния — “1” и “0” в вентилях, соответствующие им входные и выходные сигналы имеют один из двух установленных уровней напряжения. Например, +5 вольт и 0 вольт.

Высокий уровень обычно соответствует значению “истина” (“1”), а низкий — значению “ложь” (“0”).

Каждый логический элемент имеет свое условное обозначение, которое выражает его логическую функцию, но не указывает на то, какая именно электронная схема в нем реализована. Это упрощает запись и понимание сложных логических схем.

Работу логических элементов описывают с помощью таблиц истинности

Таблица истинности **это табличное представление логической схемы (операции), в котором перечислены все возможные сочетания значений истинности входных сигналов (операндов) вместе со значением истинности выходного сигнала (результата операции) для каждого из этих сочетаний.**

Схемы И, ИЛИ, НЕ, И—НЕ, ИЛИ—НЕ

С х е м а И

Схема И реализует конъюнкцию двух или более логических значений. Условное обозначение на структурных схемах схемы **И** с двумя входами представлено на рис. 5.1.

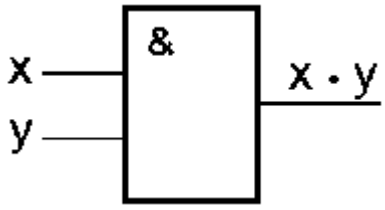


Рис. 5.1

Таблица истинности схемы **И**

x	y	x · y
0	0	0
0	1	0
1	0	0
1	1	1

Единица на выходе схемы И будет тогда и только тогда, когда на всех входах будут единицы. Когда хотя бы на одном входе будет ноль, на выходе также будет ноль.

Связь между выходом z этой схемы и входами x и y описывается соотношением: **z = x · y** (читается как **"x и y"**). Операция конъюнкции на структурных схемах обозначается знаком **"&"** (читается как **"амперсэнд"**), являющимся сокращенной записью английского слова

and.

С х е м а ИЛИ

Схема ИЛИ реализует дизъюнкцию двух или более логических значений. Когда хотя бы на одном входе схемы **ИЛИ** будет единица, на её выходе также будет единица.

Условное обозначение на структурных схемах схемы **ИЛИ** с двумя входами представлено на рис. 5.2. Знак **"1"** на схеме — от устаревшего обозначения дизъюнкции как **">=1"** (т.е. значение дизъюнкции равно единице, если сумма значений операндов больше или равна 1). Связь между выходом **z** этой схемы и входами **x** и **y** описывается соотношением: **z = x v y** (читается как **"x или y"**).

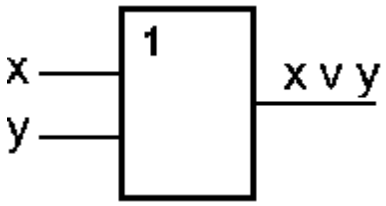


Рис. 5.2

Таблица истинности схемы **ИЛИ**

x	y	x v y
0	0	0
0	1	1
1	0	1
1	1	1

С х е м а НЕ

Схема **НЕ** (инвертор) реализует операцию отрицания. Связь между входом **x** этой схемы и выходом **z** можно записать соотношением **z =** $\overline{x}$ **,** где $\overline{x}$ читается как **"не x"** или **"инверсия x".**

Если на входе схемы **0,** то на выходе **1.** Когда на входе **1,** на выходе **0.** Условное обозначение на структурных схемах инвертора — на рисунке 5.3

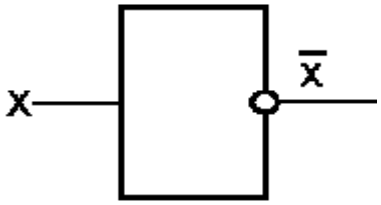


Рис. 5.3

Таблица истинности схемы **НЕ**

x	$\overline{x}$
0	1
1	0

С х е м а И—НЕ

Схема **И—НЕ** состоит из элемента **И** и инвертора и осуществляет отрицание результата схемы **И.** Связь между выходом **z** и входами **x** и **y** схемы записывают следующим образом: $z = \overline{x \cdot y}$, где $\overline{x \cdot y}$ читается как **"инверсия x и y".** Условное обозначение на структурных схемах схемы **И—НЕ** с двумя входами представлено на рисунке 5.4.

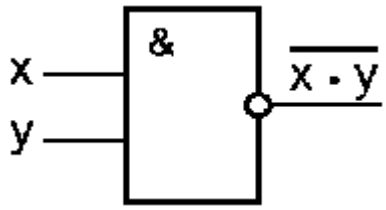


Рис. 5.4

Таблица истинности схемы И—НЕ

x	y	$\overline{x \cdot y}$
0	**0**	**1**
0	**1**	**1**
1	**0**	**1**
1	**1**	**0**

С х е м а ИЛИ—НЕ

Схема **ИЛИ—НЕ** состоит из элемента **ИЛИ** и инвертора и осуществляет отрицание результата схемы **ИЛИ.** Связь между выходом **z** и входами **x** и **y** схемы записывают следующим образом: $z = \overline{x \vee y}$, где $\overline{x \vee y}$, читается как **"инверсия x или y ".** Условное обозначение на структурных схемах схемы **ИЛИ—НЕ** с двумя входами представлено на рис. 5.5.

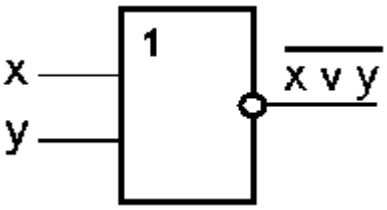


Рис. 5.5

Таблица истинности схемы ИЛИ—НЕ

x	y	$\overline{x \vee y}$
0	**0**	**1**
0	**1**	**0**
1	**0**	**0**
1	**1**	**0**

Основные законы алгебры

В алгебре логики выполняются следующие основные законы, позволяющие производить *тождественные преобразования логических выражений*:

ОСНОВНЫЕ ЗАКОНЫ АЛГЕБРЫ ЛОГИКИ

Закон	Для ИЛИ	Для И
Переместительный	$x \vee y = y \vee x$	$x \cdot y = y \cdot x$
Сочетательный	$x \vee (y \vee z) = (x \vee y) \vee z$	$x \cdot (y \cdot z) = (x \cdot y) \cdot z$
Распределительный	$x \cdot (y \vee z) = x \cdot y \vee x \cdot z$	$x \vee (y \cdot z) = (x \vee y) \cdot (x \vee z)$
Правила де Моргана	$\overline{x \vee y} = \bar{y} \cdot \bar{x}$	$\overline{x \cdot y} = \bar{y} \vee \bar{x}$
Идемпотенции	$x \vee x = x$	$x \cdot x = x$
Поглощения	$x \vee (x \cdot y) = x$	$x \cdot (x \vee y) = x$
Склеивания	$(x \cdot y) \vee (\bar{x} \cdot y) = y$	$(x \vee y) \cdot (\bar{x} \vee y) = y$
Операция переменной с ее инверсией	$x \vee \bar{x} = 1$	$x \cdot \bar{x} = 0$
Операция с константами	$x \vee 0 = x;\ x \vee 1 = 1$	$x \cdot 1 = x;\ x \cdot 0 = 0$
Двойного отрицания	$\bar{\bar{x}} = x$	

Примеры.

1. **Составим таблицу истинности для формулы** $x \cdot y \vee \overline{x \vee y} \vee x$**,** которая содержит две переменные x и y. В первых двух столбцах таблицы запишем четыре возможных пары значений этих переменных, в последующих столбцах — значения промежуточных формул и в последнем столбце — значение формулы. В результате получим таблицу:

Переменные		Промежуточные логические формулы					Формула
x	y	$\bar{x}$	$\bar{x} \cdot y$	$x \vee y$	$\overline{x \vee y}$	$x \cdot y \vee \overline{x \vee y}$	$x \cdot y \vee \overline{x \vee y} \vee x$
0	0	1	0	0	1	1	1
0	1	1	1	1	0	1	1
1	0	0	0	1	0	0	1
1	1	0	0	1	0	0	1

Из таблицы видно, что **при всех наборах значений переменных x и y формула** $x \cdot y \vee \overline{x \vee y} \vee x$ **принимает значение 1**, то есть является ***тождественно истинной*.**

2. **Таблица истинности для формулы** $\overline{x \vee y} \cdot (x \cdot \bar{y})$**:**

Переменные	Промежуточные логические формулы	Формула

x	y	$x \vee y$	$\overline{x \vee y}$	$\bar{y}$	$x \cdot \bar{y}$	$\overline{x \vee y} \cdot (x \cdot \bar{y})$
0	0	0	1	1	0	0
0	1	1	0	0	0	0
1	0	1	0	1	1	0
1	1	1	0	0	0	0

Из таблицы видно, что **при всех наборах значений переменных x и y формула** $\overline{x \vee y} \cdot (x \cdot \bar{y})$ **принимает значение 0**, то есть является ***тождественно ложной***.

3. **Таблица истинности для формулы** $\overline{x \vee \bar{y}} \vee \bar{x} \cdot z$**:**

Переменные			Промежуточные логические формулы					Формула
x	y	z	$\bar{y}$	$x \vee \bar{y}$	$\overline{x \vee \bar{y}}$	$\bar{x}$	$\bar{x} \cdot z$	$\overline{x \vee \bar{y}} \vee \bar{x} \cdot z$
0	0	0	1	1	0	1	0	0
0	0	1	1	1	0	1	1	1
0	1	0	0	0	1	1	0	1
0	1	1	0	0	1	1	1	1
1	0	0	1	1	0	0	0	0
1	0	1	1	1	0	0	0	0
1	1	0	0	1	0	0	0	0
1	1	1	0	1	0	0	0	0

Из таблицы видно, что **формула** $\overline{x \vee \bar{y}} \vee \bar{x} \cdot z$ **в некоторых случаях принимает значение 1, а в некоторых — 0**, то есть является **выполнимой**.

Равносильные преобразования логических формул имеют то же назначение, что и преобразования формул в обычной алгебре. Они служат для упрощения формул или приведения их к определённому виду путем использования основных законов алгебры логики.

Под *упрощением формулы*, не содержащей операций импликации и эквиваленции, понимают *равносильное преобразование*, приводящее к формуле, которая либо содержит по сравнению с исходной меньшее число операций конъюнкции и дизъюнкции и не содержит отрицаний неэлементарных формул, либо содержит меньшее число вхождений переменных.

Некоторые преобразования логических формул похожи на преобразования формул в обычной алгебре (вынесение общего множителя за скобки, использование переместительного и сочетательного законов и т.п.), тогда как **другие преобразования основаны на свойствах, которыми не обладают операции обычной алгебры** (использование распределительного закона для конъюнкции, законов поглощения, склеивания, де Моргана и др.).

Покажем на примерах некоторые **приемы и способы, применяемые при упрощении логических формул:**

1) $\overline{x \vee y} \cdot (x \cdot \bar{y}) = \bar{x} \cdot \bar{y} \cdot (x \cdot \bar{y}) = \bar{x} \cdot x \cdot \bar{y} \cdot \bar{y} = 0 \cdot \bar{y} \cdot \bar{y} = 0 \cdot \bar{y} = 0$

(законы алгебры логики применяются в следующей последовательности: правило де Моргана, сочетательный закон, правило операций переменной с её инверсией и правило операций с константами);

2) $\bar{x}\cdot y\vee\overline{x\vee y}\vee x=\bar{x}\cdot y\vee\bar{x}\cdot\bar{y}\vee x=\bar{x}\cdot(y\vee\bar{y})\vee x=\bar{x}\vee x=1$

(применяется правило де Моргана, выносится за скобки общий множитель, используется правило операций переменной с её инверсией);

3) $(x\vee y)\cdot(\bar{x}\vee y)\cdot(\bar{x}\vee\bar{y})=(x\vee y)\cdot(\bar{x}\vee y)\ \ (\bar{x}vy)\cdot(\bar{x}\vee\bar{y})=y\cdot\bar{x}$

(**повторяется** второй **сомножитель**, что разрешено законом идемпотенции; затем комбинируются два первых и два последних сомножителя и используется закон склеивания);

4)
$$x\cdot\bar{y}\vee\bar{x}\cdot y\cdot z\vee x\cdot z=x\cdot\bar{y}\vee\bar{x}\cdot y\cdot z\vee x\cdot z\cdot(y\vee\bar{y})=$$
$$=x\cdot\bar{y}\vee\bar{x}\cdot y\cdot z\vee x\cdot y\cdot z\vee x\cdot\bar{y}\cdot z=(x\cdot\bar{y}\vee x\cdot\bar{y}\cdot z)\vee(\bar{x}\cdot y\cdot z\vee x\cdot y\cdot z)=$$
$$=x\cdot\bar{y}\vee y\cdot z$$

(**вводится вспомогательный логический сомножитель** ($y\vee\bar{y}$); затем комбинируются два крайних и два средних логических слагаемых и используется закон поглощения);

5) $\overline{x\cdot y\vee z}=\overline{x\cdot y}\cdot\bar{\bar{z}}=(\bar{x}\vee\bar{y})\cdot z$

(сначала добиваемся, чтобы знак отрицания стоял только перед отдельными переменными, а не перед их комбинациями, для этого дважды применяем правило де Моргана; затем используем закон двойного отрицания);

6) $x\cdot y\vee x\cdot y\cdot z\vee x\cdot z\cdot p=x\cdot(y\cdot(1\vee z)\vee z\cdot p)=x\cdot(y\vee z\cdot p)$

(выносятся за скобки общие множители; применяется правило операций с константами);

7)
$$x\vee\overline{y\cdot\bar{z}}\vee\overline{\bar{x}\vee y\vee\bar{z}}=x\vee\bar{y}\vee\bar{\bar{z}}\vee\bar{\bar{x}}\cdot\bar{y}\cdot\bar{\bar{z}}=x\vee\bar{y}\vee z\vee x\cdot\bar{y}\cdot z=$$
$$=x\vee z\vee(\bar{y}\vee x\cdot\bar{y}\cdot z)=x\vee z\vee\bar{y}$$

(к отрицаниям неэлементарных формул применяется правило де Моргана; используются законы двойного отрицания и склеивания);

8)
$$x\cdot\bar{y}\vee x\cdot y\cdot z\vee x\cdot\bar{y}\cdot z\vee x\cdot\overline{y\cdot z}=x\cdot(\bar{y}\vee y\cdot z\vee\bar{y}\cdot z\vee\overline{y\cdot z})=$$
$$=x\cdot((\bar{y}\vee\overline{y\cdot z})\vee(y\cdot z\vee\overline{y\cdot z}))=x\cdot(\bar{y}\vee\overline{y\cdot z}\vee1)=x\cdot1=x$$

(общий множитель x выносится за скобки, комбинируются слагаемые в скобках — первое с третьим и второе с четвертым, к дизъюнкции $y\cdot z\vee\overline{y\cdot z}$ применяется правило операции переменной с её инверсией);

9)
$$(x\cdot\bar{y}\vee z)\cdot(\bar{x}\vee y)\vee\bar{z}=x\cdot\bar{y}\cdot\bar{x}\vee x\cdot\bar{y}\cdot y\vee z\cdot\bar{x}\vee z\cdot y\vee\bar{z}=0\vee0\vee$$
$$\vee z\cdot\bar{x}\vee z\cdot y\vee\bar{z}=z\cdot\bar{x}\vee(z\vee\bar{z})\cdot(y\vee\bar{z})=z\cdot\bar{x}\vee1\cdot(y\vee\bar{z})=z\cdot\bar{x}\vee y\vee$$
$$\vee\bar{z}=(z\cdot\bar{x}\vee\bar{z})\vee y=(z\vee\bar{z})\cdot(\bar{x}\vee\bar{z})\vee y=1\cdot(\bar{x}\vee\bar{z})\vee y=\bar{x}\vee\bar{z}\vee y$$

(используются распределительный закон для дизъюнкции, правило операции переменной с ее инверсией, правило операций с константами, переместительный закон и распределительный закон для конъюнкции);

10)
$$x\cdot y\cdot(\bar{x}\cdot z\vee\overline{\overline{x\cdot y}\cdot z}\vee z\cdot t)=x\cdot y\cdot(\bar{x}\cdot z\vee\overline{\overline{x\cdot y}}\vee\bar{z}\vee z\cdot t)=$$
$$=x\cdot y\cdot(\bar{x}\cdot z\vee x\cdot y\vee\bar{z}\vee z\cdot t)=x\cdot y\vee x\cdot y\cdot\bar{z}\vee x\cdot y\cdot z\cdot t=x\cdot y$$

(используются правило де Моргана, закон двойного отрицания и закон поглощения).

Из этих примеров видно, что при упрощении логических формул не всегда очевидно, какой из законов алгебры логики следует применить на том или ином шаге. Навыки приходят с опытом.

Задания:

1. Упростите следующие формулы, используя законы склеивания:

- **а)** $a\cdot b\cdot c\ v\ \bar{a}\cdot b\cdot c$

- **б)** $a \cdot b \cdot c \vee a \cdot \overline{b \cdot c}$
- **в)**
- **г)** $(a \vee b \vee c) \cdot (a \vee \overline{b} \vee c)$
- **д)** $(\overline{a} \vee \overline{b} \vee \overline{c}) \cdot (\overline{a} \vee b \vee c)$

Решение: $(\overline{a} \vee \overline{b} \vee \overline{c}) \cdot (\overline{a} \vee b \cdot c) = (\overline{a} \vee \overline{b \cdot c}) \cdot (\overline{a} \vee b \cdot c) = \overline{a}$.

2. Упростите следующие формулы, используя законы поглощения:

- **а)** $a \vee a \cdot b \vee a \cdot b \cdot c \vee a \cdot d \cdot f$
- **б)** $a \cdot b \vee a \cdot b \cdot c \vee a \cdot b \cdot d$
- **в)** $a \cdot (a \vee b) \cdot (a \vee c)$
- **г)** $a \cdot b \cdot (a \cdot c \vee a \cdot b)$

3 Постройте таблицы истинности для логических формул и упростите формулы, используя законы алгебры логики:

- **а)** $\overline{a \cdot \overline{c} \vee c \cdot (b \vee \overline{c})} \vee (a \vee \overline{b}) \cdot c$
- **б)** $a \cdot (b \vee \overline{c}) \vee \overline{a} \cdot b$
- **в)** $(\overline{a} \vee c) \cdot \overline{a \cdot c} \cdot (b \vee \overline{c}) \cdot \overline{b \cdot c}$
- **д)** $a \cdot b \cdot c \vee a \cdot \overline{b} \cdot c \vee a \cdot b \cdot \overline{c} \cdot d$
- **е)** $a \vee b \vee \overline{b} \cdot c \cdot d \vee \overline{b} \cdot \overline{c} \cdot \overline{d} \vee \overline{b} \cdot \overline{c} \cdot d$
- **ж)** $a \vee d \vee \overline{a} \cdot b \cdot c \vee \overline{a \cdot \overline{b} \cdot c} \vee \overline{a} \cdot \overline{b \cdot \overline{c}}$
- **з)** $\overline{a \vee b \vee c} \vee \overline{b} \vee \overline{(a \vee \overline{b} \vee c \cdot \overline{a} \vee b \vee c)} \vee \overline{a} \cdot \overline{b}$
- **и)** $a \cdot b \cdot c \vee a \cdot b \cdot \overline{c} \vee a \cdot \overline{b} \cdot c \cdot d \vee a \cdot b \cdot c \cdot \overline{d} \vee a \cdot b \cdot c \cdot d$
- **к)** $a \cdot d \cdot (\overline{a} \vee \overline{c} \cdot b \vee d) \vee a \cdot \overline{c} \vee \overline{a} \cdot b \cdot \overline{c}$

Отчет

Отчет должен быть оформлен в текстовом редакторе и содержать:

- наименование работы;
- цель работы;
- задание;
- последовательность выполнения работы;
- ответы на контрольные вопросы;
- вывод о проделанной работе.

Практическая работа №5 «Шифраторы. Дешифраторы».

Используя простейшие логические элементы, можно сконструировать более сложные устройства, реализующие соответствующие функции. Такими устройствами являются, например, шифраторы и дешифраторы.

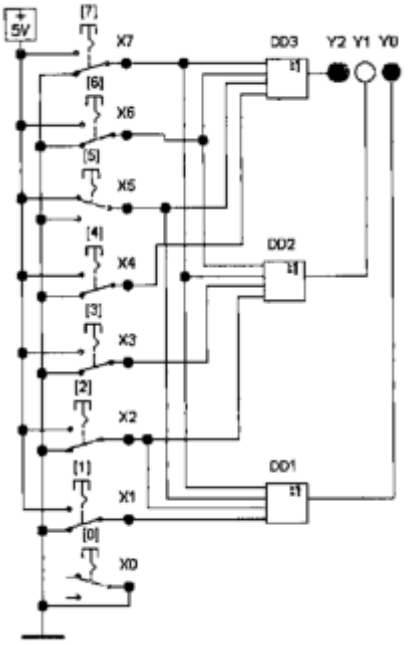
Шифраторы, называемые также кодерами, могут осуществлять преобразование десятичных чисел (позиционный код) в двоичную систему счисления. Шифратор работает следующим образом: шифратор имеет n входов, в текущий момент времени только на один из которых подается сигнал (это вход будет активным); по номеру активного входа на выходах дешифратора формируется двоичных код, соответствующей позиции активного входа. Например, если активным был пятый вход, то на выходах будет комбинация (за исключением старших нулей): $5_{10}=101_2$.

Дешифратор или декодер выполняет обратную по отношению к шифрованию операцию, т.е. преобразует двоичный код в десятичный. Входы дешифратора служат для подачи двоичных числе, а выходы последовательно нумеруются десятичными числами. При

подаче на входы двоичного числа выходной сигнал появляется на выходе, который имеет номер соответствующего десятичного числа.

Существует два типа дешифраторов: логические дешифраторы и дисплейные дешифраторы/формирователи. Логические дешифраторы представляют собой схемы средней интеграции (микросхемы, имеющие с своем составе до 100 ЛЭ), управляемые адресом. Они выбирают и приводят в активное состояние конкретный выход определяемый адресом. Дешифраторы применяются в структурах выборки адреса запоминающих устройства, разуплотнения маршрутизации данных и т.п.

1. Шифратор.



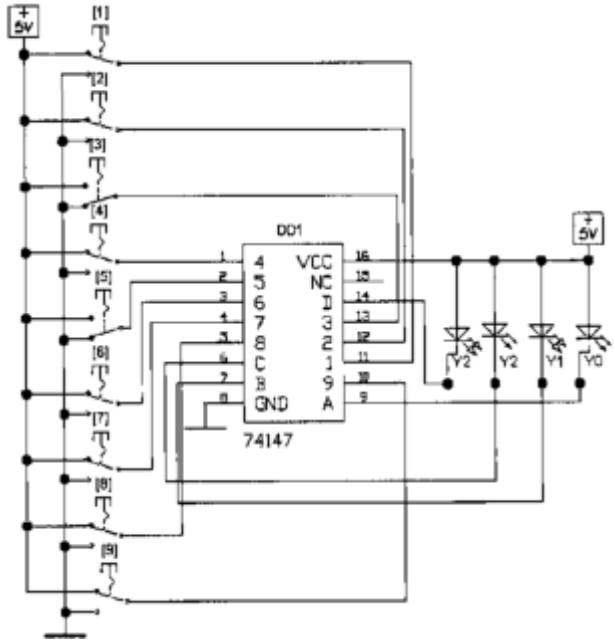
№	Позиционный код								Двоичный код		
	X_7	X_6	X_5	X_4	X_3	X_2	X_1	X_0	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1	0	0	0	1
2	0	0	0	0	0	1	0	0	0	1	0
3	0	0	0	0	1	0	0	0	0	1	1
4	0	0	0	1	0	0	0	0	1	0	0
5	0	0	1	0	0	0	0	0	1	0	1
6	0	1	0	0	0	0	0	0	1	1	0
7	1	0	0	0	0	0	0	0	1	1	1

Рис.1. Восьмивходовый шифратор на логических элементах «ИЛИ» и его таблица истинности.

Собрать схему, представленную на рис.1, и составить таблицу истинности для нее. Убедиться, что она реализует соответствующую функцию (вырабатывает на выходах двоичных код, соответствующий номеру заданного входа).

2. Приоритетный шифратор на микросхеме.

В микросхемном исполнении имеются так называемые приоритетные шифраторы, в которых при нажатии на несколько клавиш на выход будет выеден номер большей из них. Для этого микросхема внутри дополняется логикой выбора приоритета.



74147 (10-to-4 Priority Enc)

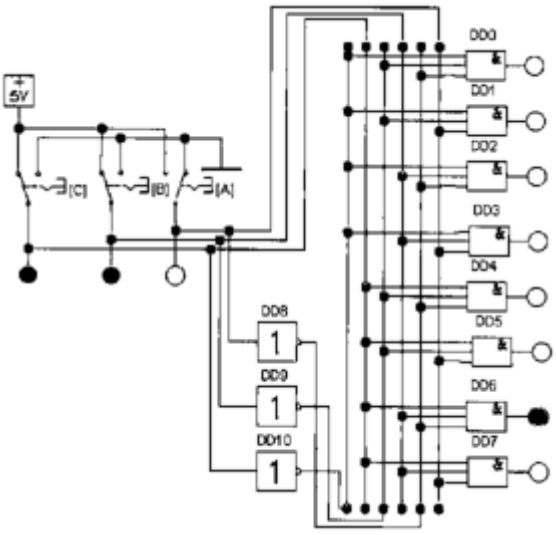
10-Line to 4-Line Priority Encoder truth table:

Inputs										Outputs			
1	2	3	4	5	6	7	8	9		D	C	B	A
1	1	1	1	1	1	1	1	1		1	1	1	1
X	X	X	X	X	X	X	X	0		0	1	1	0
X	X	X	X	X	X	X	0	1		0	1	1	1
X	X	X	X	X	X	0	1	1		1	0	0	0
X	X	X	X	X	0	1	1	1		1	0	0	1
X	X	X	X	0	1	1	1	1		1	0	1	0
X	X	X	0	1	1	1	1	1		1	0	1	1
X	X	0	1	1	1	1	1	1		1	1	0	0
X	0	1	1	1	1	1	1	1		1	1	0	1
0	1	1	1	1	1	1	1	1		1	1	1	0

Рис.2. Приоритетный шифратор и его таблица истинности.

Собрать схему, представленную на рис.2, и составить таблицу истинности. Убедиться, что она соответствует приведенной на рисунке.

3. Дешифратор на логических элементах



№	Двоичный код			Позиционный код							
	X_2	X_1	X_0	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
2	0	1	0	0	0	0	0	0	1	0	0
3	0	1	1	0	0	0	0	1	0	0	0
4	1	0	0	0	0	0	1	0	0	0	0
5	1	0	1	0	0	1	0	0	0	0	0
6	1	1	0	0	1	0	0	0	0	0	0
7	1	1	1	1	0	0	0	0	0	0	0

Рис.3. Трехвходовый дешифратор на логических элементах «И» и «НЕ». Его таблица истинности.

Соберите схему, показанную на рис.3, и составьте для нее таблицу истинности. Проверьте совпадение с таблицей, приведенной на рисунке.

4. Преобразователь кода для семисегментного индикатора.

Дешифраторы и дисплейные дешифраторы/формирователи формируют цифровые коды для семи сегментного индикатора, и затем обеспечивают передачу кода на формирователь или непосредственно на дисплей. В семи сегментном индикаторе десятичных цифр каждый сегмент (их семь) представляет собой отдельный светоизлучающий элемент (используется также буквенная идентификация сегментов, соответственно от a до g). Светящееся изображение цифр или знаков получается при подаче напряжения на определенные сегменты:

Такой преобразователь кода должен удовлетворять таблице истинности, приведенной ниже:

№	X_K				Y_K						
	X_3	X_2	X_1	X_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
					a	*b*	*c*	*d*	*e*	*f*	*g*
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	0	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1
A	1	0	1	0	1	1	1	0	1	1	1
B	1	0	1	1	0	0	1	1	1	1	1
C	1	1	0	0	1	0	0	1	1	1	0
D	1	1	0	1	0	1	1	1	1	0	1
E	1	1	1	0	1	0	0	1	1	1	1
F	1	1	1	1	1	0	0	0	1	1	1

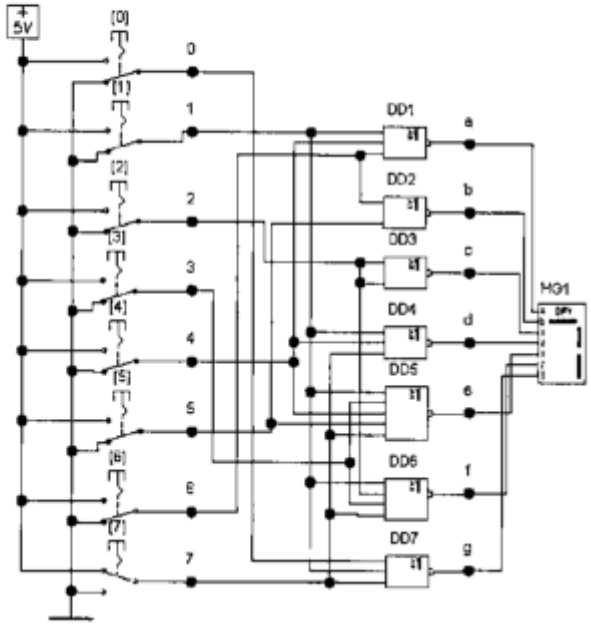


Рис.6 Дешифратор для 7-сегментного индикатора на логических элементах.

Собрать схему, показанную на рис.6, и проверить ее работу в соответствии с представленной таблицей истинности.

5. Дешифратор для 7-сегментного индикатора на микросхеме.

Данный дешифратор преобразует двоично-десятичный код (binary-decimal code), подаваемый на входы A,B,C,D, в код управления 7-сегментным индикатором. Двоично-десятичный код представляет собой упорядоченный по разрадам набор двоичных чисел, в котором разрядам приписаны следующие «веса» в порядке уменьшения старшинства. D – 8, C – 4, B – 2, A – 1. Поэтому данный код называют также кодом 8-4-2-1. Фактически в этом коде записаны десятичные числа от 0 до 15 во входных переменных таблицы истинности:

7447 (BCD-to-Seven-Segment Dec)

No.	$\overline{LT}$	$\overline{RBI}$	D C B A	$\overline{BI}/\overline{RBO}$	a b c d e f g	
0	1	1	0 0 0 0	1	1 1 1 1 1 1 0	
1	1	X	0 0 0 1	1	0 1 1 0 0 0 0	
2	1	X	0 0 1 0	1	1 1 0 1 1 0 1	
3	1	X	0 0 1 1	1	1 1 1 1 0 0 1	
4	1	X	0 1 0 0	1	0 1 1 0 0 1 1	
5	1	X	0 1 0 1	1	1 0 1 1 0 1 1	
6	1	X	0 1 1 0	1	0 0 1 1 1 1 0	
7	1	X	0 1 1 1	1	1 1 1 0 0 0 0	
8	1	X	1 0 0 0	1	1 1 1 1 1 1 1	
9	1	X	1 0 0 1	1	1 1 1 0 0 1 1	
10	1	X	1 0 1 0	1	0 0 0 1 1 0 1	< I
11	1	X	1 0 1 1	1	0 0 1 1 0 0 1	< N
						< V
12	1	X	1 1 0 0	1	0 1 0 0 0 1 1	< A
13	1	X	1 1 0 1	1	1 0 0 1 0 1 1	< L
14	1	X	1 1 1 0	1	0 0 0 1 1 1 1	< I
15	1	X	1 1 1 1	1	0 0 0 0 0 0 0	< D
BI	X	X	X X X X	0	0 0 0 0 0 0 0	
RBI	1	0	0 0 0 0	0	0 0 0 0 0 0 0	
LT	0	X	X X X X	1	1 1 1 1 1 1 1	

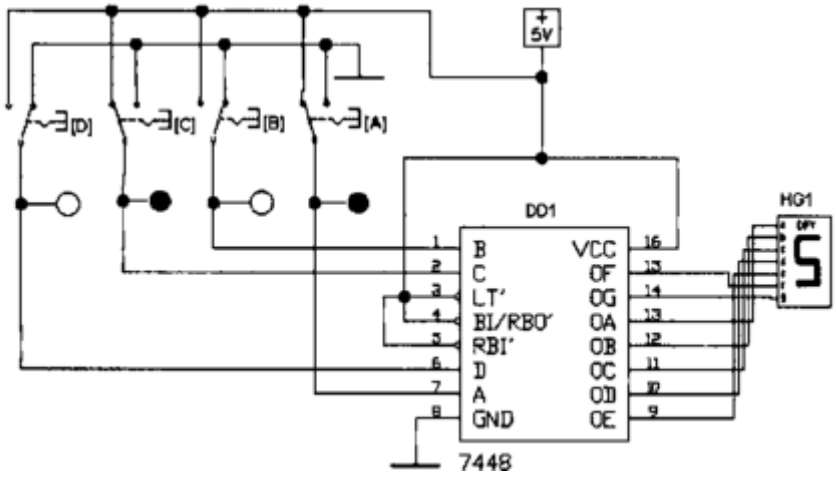


Рис.7. Дешифратор кода для 7-сегментного индикатора на микросхеме 7448. Собрать схему на рис.7 и проверить ее работу (соответствие двоичного кода на входах дешифратора и показаний индикатора).

Содержание отчета:

А) наиболее важные схемы

Б) таблицы истинности для всех схем

В) выводы по работе

Вопросы:

1. Объясните принцип действия шифратора.
2. Объясните принцип действия дешифратора.
3. Какие особенности имеет дешифратор кода для 7-сегментного индикатора?
4. Где применяются шифраторы и дешифраторы?

Практическая работа №6 «Мультиплексоры. Демультиплексоры».

Цель работы: Построение мультиплексоров и демультиплексоров.

1. Приборы и оборудование.

Лабораторная работа выполняется на лабораторном стенде, описание которого дано в лабораторной работе №1.

2. Пояснения и порядок выполнения работы.

2.1 Построить восьмивходовой мультиплексор (8→1)

Мультиплексором называется комбинационное логическое устройство, предназначенное для передачи информации с одного из нескольких информационных входов на один выход. Выбор информационного входа осуществляется с помощью специальных входов, называемых адресными входами. Число адресных входов, определяет число информационных входов. Если n – разрядность адресных входов, то число информационных входов должно равняться $N=2^n$.

Условное обозначение мультиплексора приведено на рис. 1. $D_0 \div D_7$ – информационные входы; $A_2 \div A_0$ – адресные входы.

2.1.1. По таблице истинности 1, задающей закон функционирования мультиплексора, составить Булево выражение в СДНФ.

2.1.2. По полученному выражению начертить функциональную схему мультиплексора.

2.1.3. Собрать схему на стенде и проверить её работоспособность. Значение на информационных входах D_i набираются на переключательном регистре. Значение функции контролируется с помощью светодиодов.

2.2. Построить демультиплексор на 4 выхода (1→4).

Условное обозначение демультиплексора приведено на рис.2. D – информационный вход; $y_0 \div y_3$ – выходы демультиплексора.

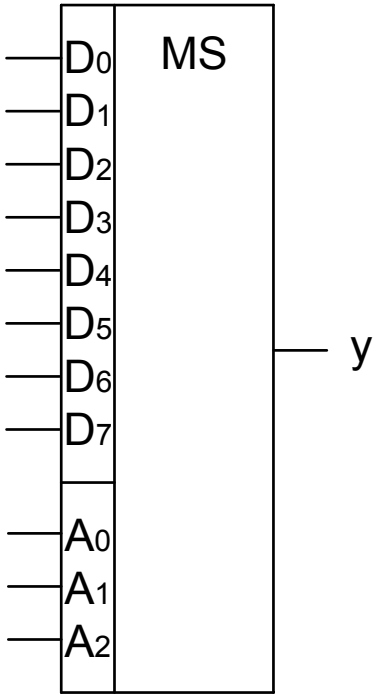


Рис.1 Условное графическое обозначение мультиплексора.

Таблица 1

Информационные входы								Адресные входы			y
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	A_2	A_1	A_0	
1	x	x	x	x	x	x	x	0	0	0	1
0	x	x	x	x	x	x	x	0	0	0	0
x	1	x	x	x	x	x	x	0	0	1	1
x	x	1	x	x	x	x	x	0	1	0	1
x	x	x	1	x	x	x	x	0	1	1	1
x	x	x	x	1	x	x	x	1	0	0	1
x	x	0	x	x	1	x	x	1	0	1	1
x	x	x	x	x	x	1	x	1	1	0	1
x	x	x	x	x	x	x	1	1	1	1	1

Примечание: Символ x – произвольное состояние.

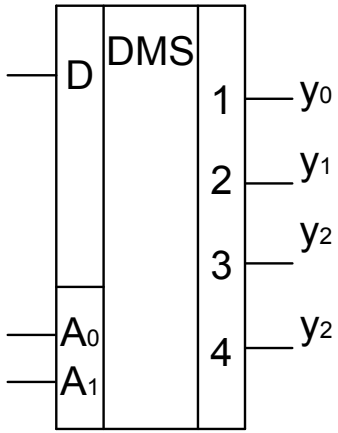


Рис.2 Условное графическое обозначение демультиплексора.

Таблица 2

D	Адресные входы		Выходы			
	A_1	A_1	y_0	y_1	y_2	y_4
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

Демультиплексором называется комбинационное логическое устройство,
предназначенное для передачи информации с информационного входа D на один из выходов. Выбор необходимого выхода осуществляется с помощью адресных входов A_i , число разрядов n которых определяет число выходов m демультиплексора: $m=2^n$.

2.2.1. По таблице истинности 2, задающей закон функционирования демультиплексора, составить Булевы выражения в СДНФ.

2.2.2. По полученным выражениям начертить функциональную схему демультиплексора.

2.2.3. Собрать схему на стенде и проверить её работоспособность. Значения аргументов набираются на переключательном регистре. Значение функций контролируются по светодиодам.

Построить схему сумматора для сложения двух одноразрядных чисел a и b.

Сумматором называется комбинационное логическое устройство, предназначенное для выполнения операции арифметического сложения чисел, представленных в виде двоичных кодов.

Условное графическое обозначение одноразрядного сумматора приведено на рис. 1.

Одноразрядный сумматор имеет три входа: a – для подачи первого слагаемого; b – вход для подачи второго слагаемого; p_0 – вход для подачи сигнала переноса из младших разрядов. И два выхода: S – выход, на котором фиксируется результат сложения a+b+ p_0; p_1 – выход, на котором фиксируется сигнал переноса в старший разряд. Такой сумматор называется полным. Логика работы полного сумматора сводится к сложению двух одноразрядных переменных a и b с учётом переноса из предыдущего разряда и формированию сигнала суммы S и переноса в старший разряд p_1.

2.1.1. По таблице истинности 1, задающей закон функционирования дешифратора, составить булевы выражения в СДНФ.

2.1.2. Минимизировать булевы выражения.

2.1.3. По полученному выражению начертить функциональную схему одноразрядного сумматора.

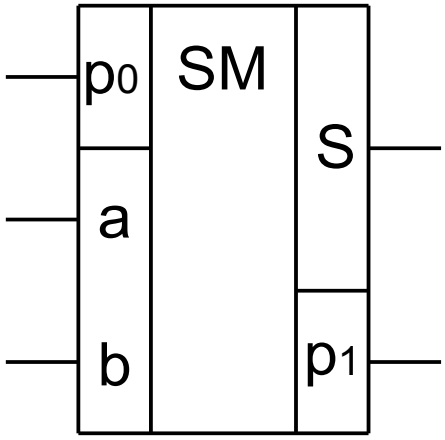
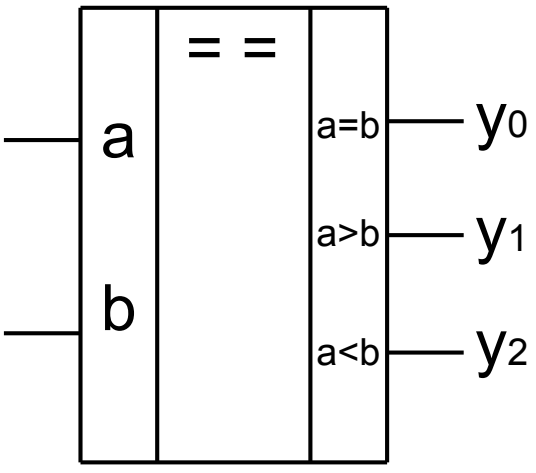


Рис.1. Условное графическое обозначение полного одноразрядного сумматора.



Ри

с.2. Условное графическое обозначение одноразрядного компаратора.

Таблица 1

p_0	a	b	S	p_1
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Таблица 2

a	b	y_2	y_1	y_0
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

2.1.4. Собрать схему на стенде и проверить её работоспособность. Значение входных переменных набираются на переключательном регистре. Значение функции контролируется с помощью светодиодов.

2.2. Построить компаратор для сравнения по величине двух одноразрядных чисел a и b.

Цифровым компаратором называется комбинационное устройство, предназначенное для сравнения чисел, представленных в виде двоичных кодов. Число входов компаратора определяется разрядностью сравниваемых кодов. На выходе компаратора обычно формируются три сигнала: y_0 фиксирует равенство кодов ($a = b$); y_1 фиксирует признак $a > b$; y_2 фиксирует признак $a < b$. При этом активный сигнал 1 формируется только на одном из выходов. Условное графическое обозначение компаратора приведено на рис. 2.

2.2.1. По таблице истинности 2, задающей закон функционирования компаратора, составить булевы выражения в СДНФ.

2.2.2. По полученному выражению начертить функциональную схему одноразрядного компаратора.

2.2.3. Собрать схему на стенде и проверить её работоспособность. Значение входных переменных набираются на переключательном регистре. Значение функции контролируется по светодиодам.

2.3. Построить компаратор для сравнения двух двухразрядных чисел a_1a_0 и b_1b_0.

Для построения такого компаратора используется метод блочного конструирования, в котором за основу берётся схема одноразрядного компаратора.

2.3.1. Записать алгоритм сравнения.

2.3.2. Начертить функциональную схему двухразрядного компаратора.

2.3.3. Собрать схему на стенде и проверить её работоспособность.

3. Содержание отчёта.

1. Условные графические обозначения сумматора, компараторов.

2. Таблицы истинности сумматора, компараторов.
3. Минимизирующие карты Карно и Булевы выражения.
4. Функциональные схемы сумматора и компараторов.
4. Контрольные вопросы.
1. Построить на базе одноразрядного сумматора трёх-, четырёхразрядные сумматоры.
2. Как заменить операцию вычитания двоичных кодов операцией сложения.
3. Каким образом можно перевести сумматор в режим вычитания?
4. Начертить условное графическое обозначение сумматора, работающего в режиме вычитания, с использованием: а) обратного; б) дополнительного кодов. Показать на примере трёхразрядного сумматора.
5. Как на базе двоичного сумматора построить сумматор для сложения двоично-десятичных чисел?
6. Постройте схему одноразрядного полного сумматора на базе трёхразрядного дешифратора.
7. Каковы назначения и логическая схема компаратора?
8. Построить на базе трёхразрядного сумматора компаратор для сравнения двух двухразрядных чисел.

Отчет

Отчет должен быть оформлен в текстовом редакторе и содержать:

- наименование работы;
- цель работы;
- задание;
- последовательность выполнения работы;
- ответы на контрольные вопросы;
- вывод о проделанной работе.

Практическая работа №7 «Цифровые компараторы. Сумматоры. Триггеры

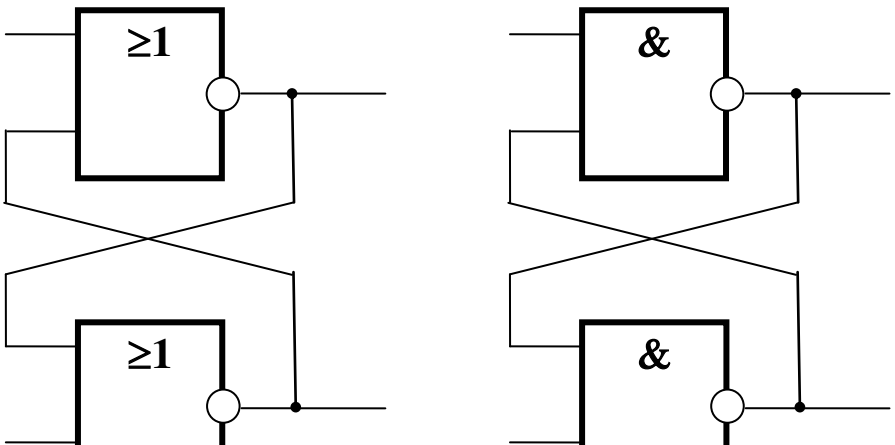
Тема работы: Моделирование и исследование логики работы триггеров и регистров.

Цель работы: ознакомление с принципом работы триггеров и регистров, получение практических навыков в построении и контроле работоспособности триггеров и регистров, а также исследование логики работы триггеров и регистров в различных режимах методом моделирования с использованием программы Electronics Workbench.

Теоретическая часть

Общие сведения об элементах памяти бортовых цифровых вычислительных устройств

Для построения цифровых устройств кроме логических элементов требуются элементы памяти, предназначенные для хранения двоичных кодов в течение требуемого времени.



В качестве статического элемента памяти используются бистабильные ячейки (БЯ), имеющие два устойчивых состояния. Бистабильные ячейки могут быть построены на двух логических элементах И-НЕ или ИЛИ-НЕ, соединенных перекрёстными связями (см. рисунок 1).

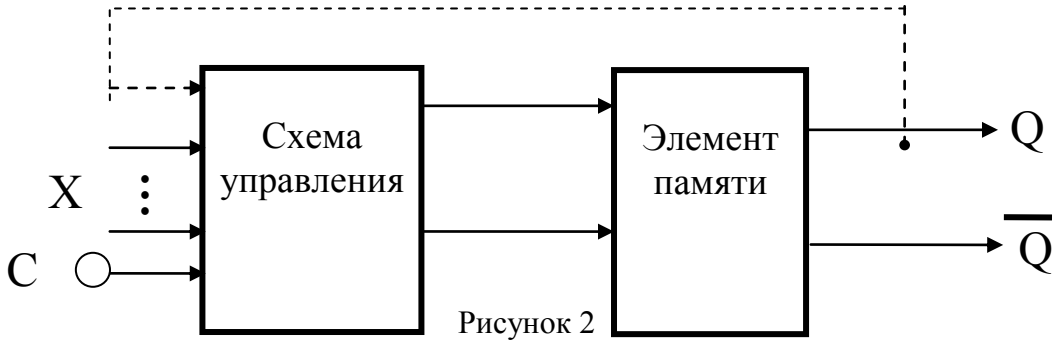


Рисунок 2

В качестве элементов памяти используются так называемые триггеры. Триггер - это цифровая электронная схема с двумя устойчивыми состояниями, которые устанавливаются при подаче соответствующей комбинации входных сигналов и сохраняются после снятия этих сигналов. Структурная схема триггера показана на рисунке 2. Триггер имеет несколько входов и два выхода - прямой и инверсный . Сигналы на выходах триггера всегда имеют различные значения. Если на прямом выходе сигнал равен 1, то на инверсном (Q и $\overline{Q}$) - 0 и наоборот. Состояние триггера определяется значением сигнала на прямом выходе (Q). Если сигнал на прямом выходе равен 1, то триггер находится в состоянии 1.

Триггеры могут быть синхронными или асинхронными. Если изменения сигнала Q происходит только при наличии специального сигнала C, являющегося сигналом синхронизации, то такой триггер называется синхронным триггером. Синхронизация триггера может происходить либо по уровню сигнала, либо по фронту сигнала (переднему или заднему).

Асинхронный триггер не имеет входа синхронизации, поэтому переключение триггера происходит только при поступлении на вход информационных входных сигналов X.

Логика переключения триггера из одного состояния в другое зависит от количества и назначения входов. Наиболее часто используются в цифровой технике следующие типы триггеров: RS-триггеры, JK-триггеры, D-триггеры и T-триггеры. Буквами R, S, J, K, D и T обозначаются информационные входы триггеров (X).

Асинхронные и синхронные триггеры разных типов

Асинхронные RS-триггеры

Асинхронный RS-триггер имеет два информационных входа - R и S. Вход S используется для установки триггера в состояние 1, а вход R - для установки в состояние 0.

Работа триггера описывается таблицей переходов, которая имеет вид таблицы 1.

Таблица 1

Входы		Состояния	
R	**S**	**Q(0)**	**Q(1)**

0	0	0	1
0	1	1	1
1	0	0	0
1	1	Не определено	

Из таблицы 1 может быть получено уравнение переходов триггера. После минимизации (например, с использованием карт Карно) уравнение переходов примет вид:

$$Q_{t+1}=S+\bar{R}\cdot Q_t$$

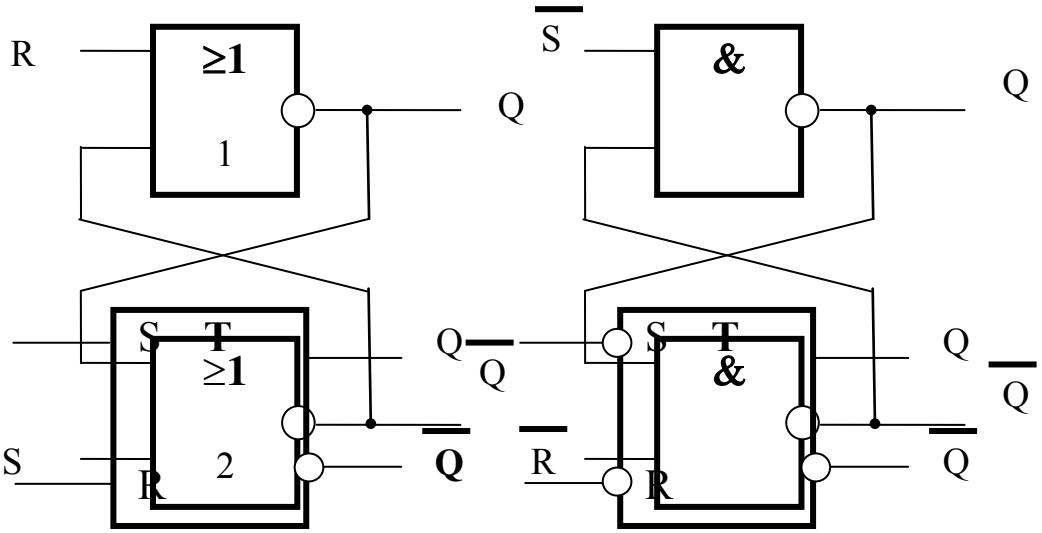


Рисунок 3

Из уравнения следует, что при S=1, R=0 всегда $Q_{t+1}=1$, при S=0, R=1 всегда $Q_{t+1}=0$, а при S=0, R=0 $Q_{t+1}=Q_t$. Комбинация сигналов S=1, R=1 является запрещенной, так состояние триггера не определено.

Для построения триггера на элементах И-НЕ уравнение необходимо преобразовать (двойным инвертированием) к другому виду:

$$Q_{t+1}=\overline{\overline{S}\cdot\overline{\bar{R}\cdot Q_t}}$$

Для построения триггера на элементах ИЛИ-НЕ уравнение имеет вид:

$$Q_{t+1}=\overline{\bar{S}+\overline{R+Q_t}}$$

Функциональные схемы асинхронных RS-триггеров, построенные на элементах ИЛИ-НЕ (слева) и И-НЕ (справа), и их условные графические обозначения (УГО) показаны на рисунке 3.

Как видно из рисунка 3, асинхронный RS-триггер представляет собой бистабильную ячейку, построенную на элементах И-НЕ или ИЛИ-НЕ.

При построении RS-триггера на элементах И-НЕ действующими установочными сигналами являются инверсные значения информационных сигналов R и S.

Синхронные RS-триггеры

Синхронный триггер дополнительно имеет вход синхронизации C, на который поступает синхросигнал. Информационные сигналы R и S воздействуют на состояние триггера только при значении синхросигнала C=1.

Таблица переходов синхронного RS-триггера состоит из двух частей. Первая часть таблицы описывает переходы триггера при C=1 и совпадает с таблицей переходов асинхронного триггера (см. таблицу 1), а вторая – при C=0.

При C=0 триггер не меняет своего состояния при любой комбинации сигналов на информационных входах R и S. В этом случае всегда $Q_{t+1}= Q_t$.

Уравнение синхронного RS-триггера имеет вид:

$$Q_{t+1}=S\cdot C+(\overline{R}+\overline{C})\cdot Q_t$$

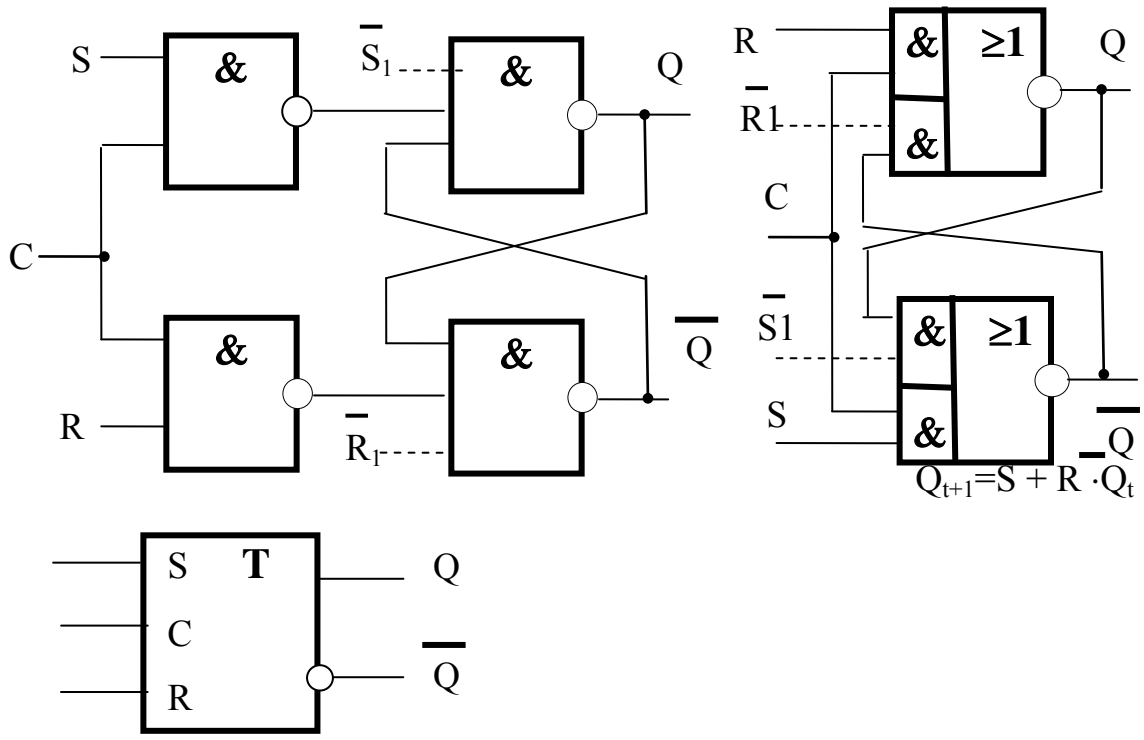


Рисунок 4

Из уравнения следует, что при C=0 $Q_{t+1}=Q_t$, а при C=1 $Q_{t+1}=S+\overline{R}\cdot Q_t$, т.е. работа описывается уравнением асинхронного триггера. На рисунке 6.4 приведены функциональные схемы синхронных RS-триггеров, реализованных на элементах И - НЕ для уравнения

$$\overline{Q}_{t+1}=\overline{S\cdot C}+\overline{\overline{R\cdot C}\cdot Q_t}$$

и на элементах И-ИЛИ-НЕ для уравнения

$$\overline{Q}_{t+1}=\overline{S\cdot C+\overline{R\cdot C}\cdot Q_t}$$

На рисунке 4, кроме основных входов R и S, показаны дополнительные инверсные асинхронные входы R_1 и S_1.

Двухтактные RS-триггеры

Триггеры в ЭВМ используются в различных узлах, между которыми осуществляется передача информации. Устойчивая работа цепочки триггеров возможна только в том случае, если запись новой информации в триггер осуществляется после считывания ранее записанной информации и передачи её в следующий по цепочке триггер. Это возможно при использовании двух серий синхроимпульсов, сдвинутых относительно друг друга на 180°. Такой принцип управления и синхронизации применяется в двухтактных триггерах.

Простейшая схема двухтактного RS-триггера может быть построена на двух однотактных триггерах, причём синхроимпульсы на входы С первого и второго триггеров должны подаваться в противофазе. Это делается с помощью инвертора (см. рисунок 5).

При поступлении на вход первого однотактного триггера импульса C=1 информация

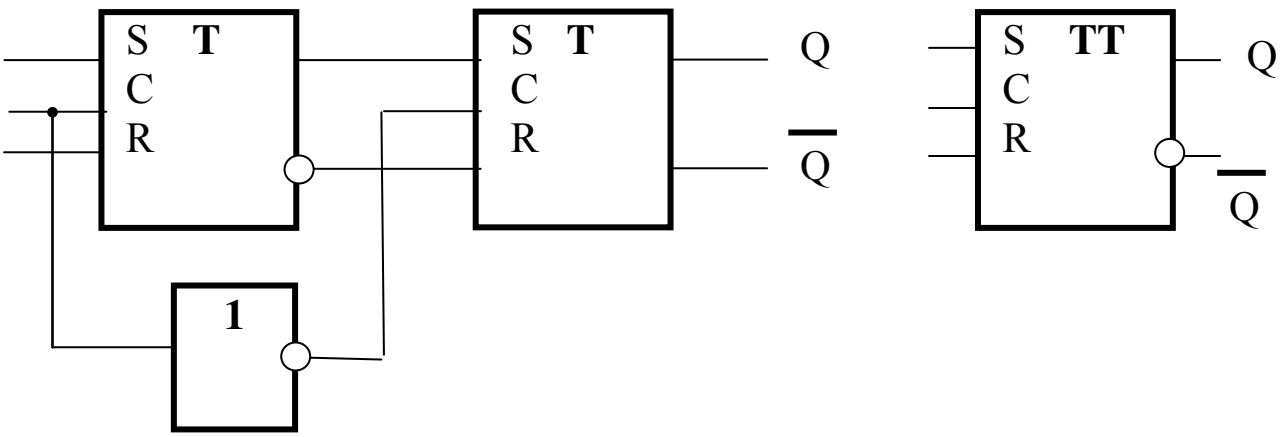


Рисунок 5

на входах R и S устанавливает триггер в соответствующее новое состояние Q_{t+1}, а второй однотактный триггер хранит информацию о предыдущем состоянии Q_t, так как на его входе С сигнал равен нулю. По окончании действия синхроимпульса, т.е. при С=0, первый триггер переходит в режим хранения, а информация Q_{t+1}, записанная в первом триггере, передается во второй, так как на его входе С сигнал становится равным единице. В результате к началу следующего такта на выходе двухтактного RS-триггера появится сигнал, определяемый состоянием Q_{t+1} первого триггера. В таком триггере выходной сигнал формируется по заднему фронту синхроимпульса.

Двухтактный синхронный RS-триггер может быть использован для построения других типов триггеров, таких как D-, T- и JK-триггеров.

Для установки RS-триггера в 0 или 1 независимо от присутствия сигнала на входе С в схему вводят прямые или инверсные входы R и S асинхронной установки, как показано на рисунке 6

Асинхронный и синхронный D-триггеры

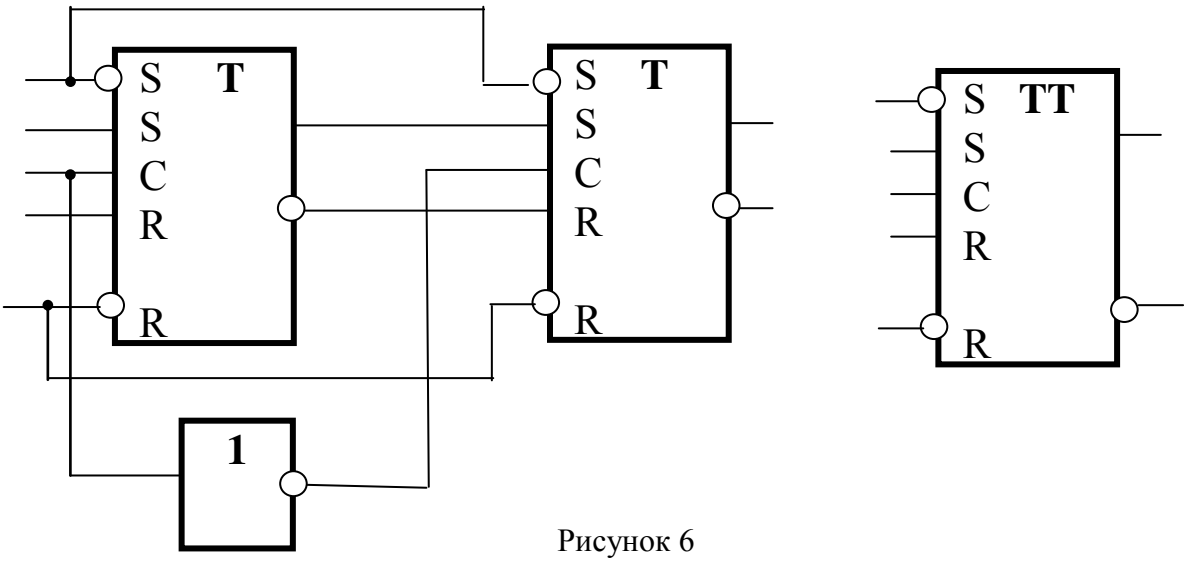


Рисунок 6

В вычислительной технике широко применяется D-триггер, который реализует функцию временн*о*й задержки входного сигнала. D-триггер имеет один информационный вход. Логика работы асинхронного D -триггера описывается таблицей переходов, которая имеет вид таблицы 2.

По таблице 2 может быть записано уравнение переходов D-триггера:

$Q_{t+1} = D_t$,

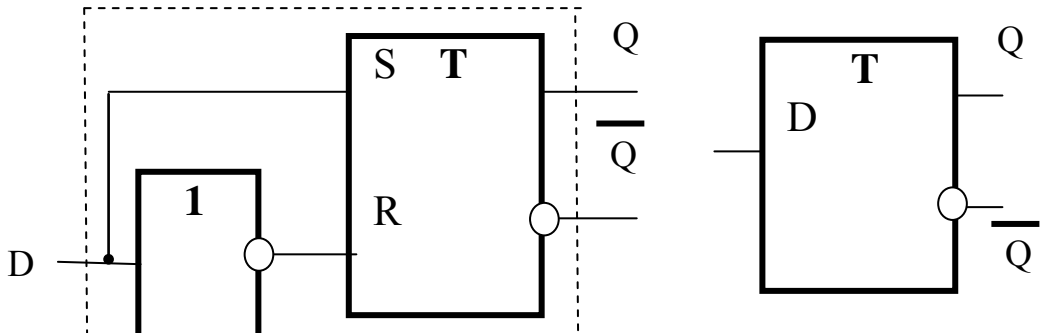
где: t - текущий момент времени; t+1 - последующий момент времени.

Таблица 2

Вход	Состояния	
D	Q(0)	Q(1)
0	0	0
1	1	1

Как видно из уравнения, в асинхронном D-триггере состояние (выходной сигнал) $\mathbf{Q_{t+1}}$ повторяет значение входного сигнала $\mathbf{D_t}$. Поэтому асинхронный D-триггер по существу является не элементом памяти, а элементом задержки, и рассматривается только как основа для построения синхронного D-триггера.

Функциональная схема и УГО асинхронного D-триггера, построенного на основе асинхронного RS-триггера, показаны на рисунке 7.



Для построения счётчиков, регистров и других цифровых схем используются синхронные D-триггеры как однотактные, так и двухтактные. Логика работы синхронного D-триггера описывается таблицей переходов, которая имеет вид таблицы 3.

Входы		Состояния	
C	D	Q(0)	Q(1)
1	0	0	0
1	1	1	1
0	0	0	1
0	1	0	1

Таблица 3

Уравнение переходов синхронного триггера, записанное по таблице 6.3, имеет следующий вид:

В соответствии с уравнением синхронный D-триггер при C=0 сохраняет свое состояние, а при C=1 работает как асинхронный.

Функциональная схема синхронного D-триггера на элементах ИЛИ-НЕ приведена на рисунке 8.

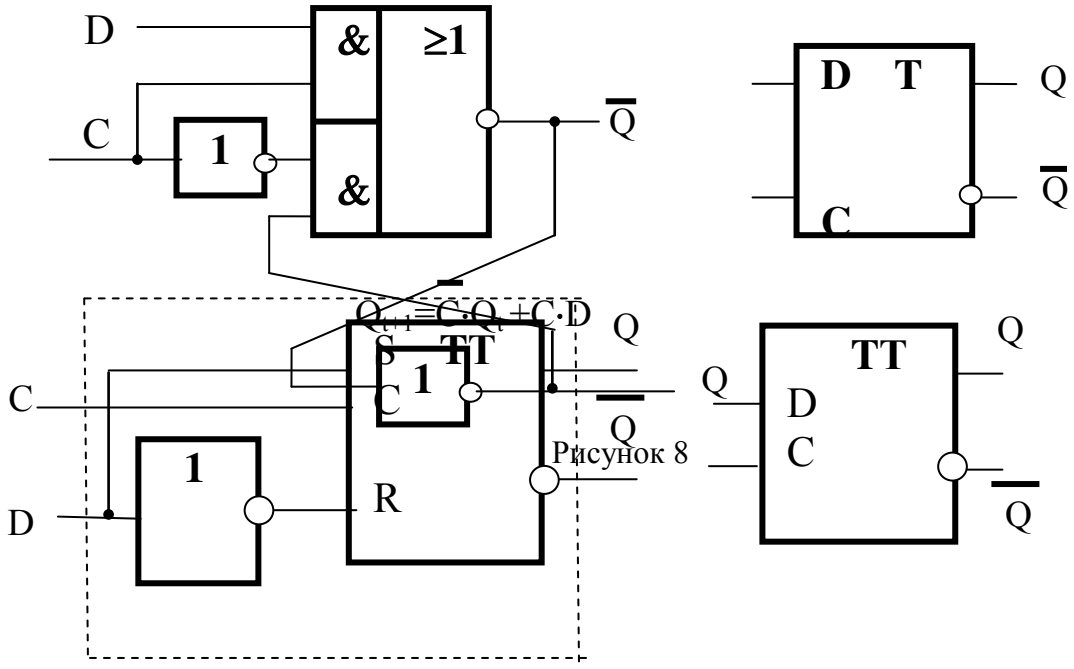


Рисунок 8

Рисунок 9

Функциональная схема двухтактного D-триггера, построенного на основе двухтактного RS- триггера, приведена на рисунке 9.

Асинхронный и синхронный Т-триггеры

Т-триггер имеет один информационный вход. Логика работы асинхронного Т-триггера может быть описана таблицей переходов, которая имеет вид таблицы 4.

Таблица 4

Вход	Состояния	
T	Q(0)	Q(1)
0	0	1
1	1	0

По таблице 4 может быть получено следующее уравнение асинхронного Т-триггера**:**

$$Qt+1 = \overline{T}Q_t + T\overline{Q}_t$$

Как видно из таблицы 4 и уравнения триггера, при Т=1 асинхронный Т-триггер меняет свое состояние на противоположное, а при Т=0 состояние триггера не изменяется.

Так как Т-триггер суммирует (или подсчитывает) по модулю два количество единиц, поступающих на его информационный вход, то Т-триггер называют также триггером со счетным входом.

Логика работы синхронного Т-триггера описывается таблицей переходов, которая имеет вид таблицы 5.

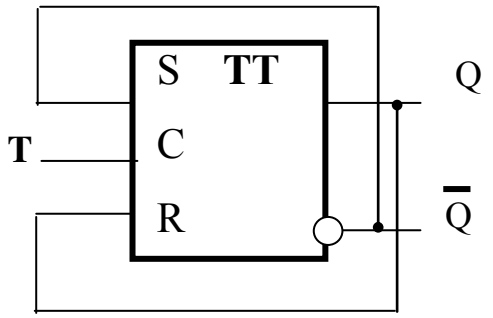
Таблица 5

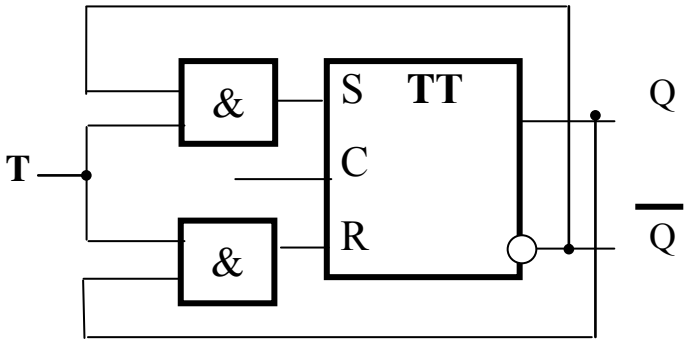
Входы		Состояния	
C	T	Q(0)	Q(1)
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Из таблицы 5 видно, что при С=0 триггер не изменяет своего состояния, а при С=1 работает как асинхронный Т-триггер.

Функциональная схема Т-триггера может быть построена на основе синхронного RS-триггера (однотактного или двухтактного).

Схема асинхронного Т-триггера приведена на рисунке 10, а синхронного Т-триггера - на рисунке 11. Обе схемы построены на основе синхронного двухтактного RS-триггера. Аналогичные схемы можно строить на основе однотактного RS-триггера. В двухтактных асинхронных Т-триггерах выходной сигнал формируется по заднему фронту входного сигнала Т, а в однотактных - по переднему фронту. В двухтактных синхронных Т-триггерах выходной сигнал формируется по заднему фронту сигнала С.





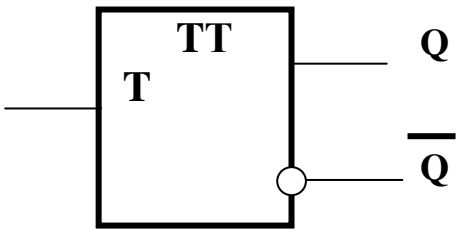


Рисунок 10

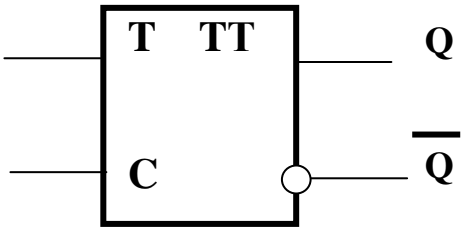


Рисунок 11

Схему асинхронного Т-триггера, в свою очередь, можно получить из D-триггера простой коммутацией входов и выходов (см. рисунок 12).

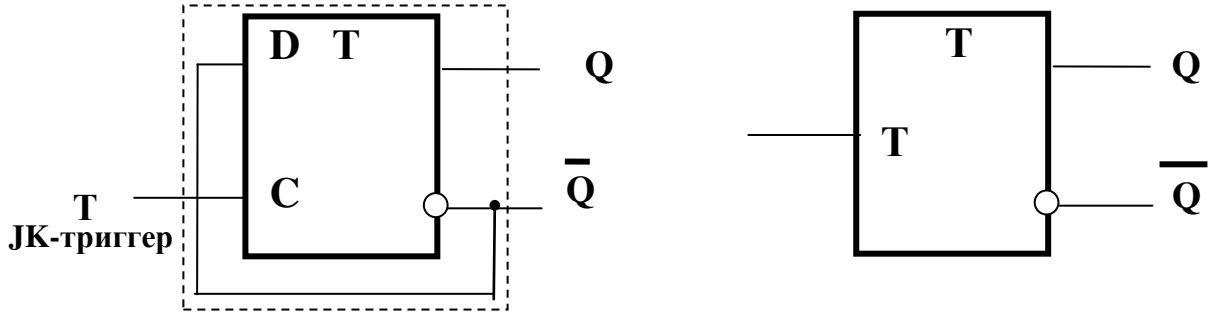


Рисунок 12

JK-триггер называется также универсальным триггером. Универсальность схемы JK-триггера состоит в том, что простой коммутацией входов и выходов можно получать схемы других типов триггеров.

JK-триггер имеет два информационных входа. Вход J используется для установки триггера в состояние 1, а вход К -для установки в состояние 0, т.е. входы J и К аналогичны входам R и S RS-триггера. Отличие заключается в том, что на входы J и К могут одновременно поступать сигналы 1. В этом случае JK- триггер изменяет свое состояние на противоположное.

Таблица переходов JK-триггера при C=1 имеет вид таблицы 6.

Таблица 6

Входы		Состояния	
J	K	Q(0)	Q(1)
0	0	0	1
0	1	0	0
1	0	1	1
1	1	1	0

Из таблицы 6 можно получить следующее уравнение JK-триггера:

$$Q_{t+1}=J\overline{Q_t}+\overline{K}Q$$

Следовательно, при J=1, K=0 всегда $Q_{t+1}=1$, а при J=0, K=1 всегда $Q_{t+1}=0$, т.е. JK-триггер работает как RS-триггер, если рассматривать входы J и K как входы S и R.

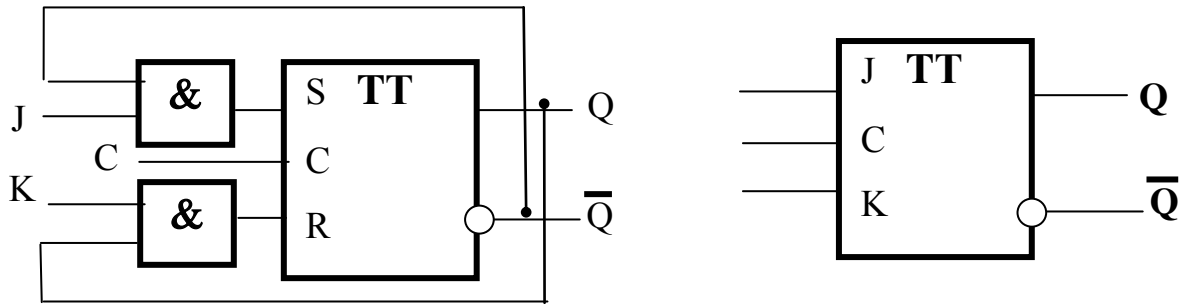


Рисунок 13

В свою очередь, при J=1, K=1 _$Q_{t+1}=Q_t$, т.е. триггер переходит в противоположное состояние (работает как Т-триггер).

Функциональная схема двухтактного JK-триггера и УГО триггера показаны на рисунке 13. Примеры получения других типов триггеров на основе JK-триггера представлены на рисунок 14.

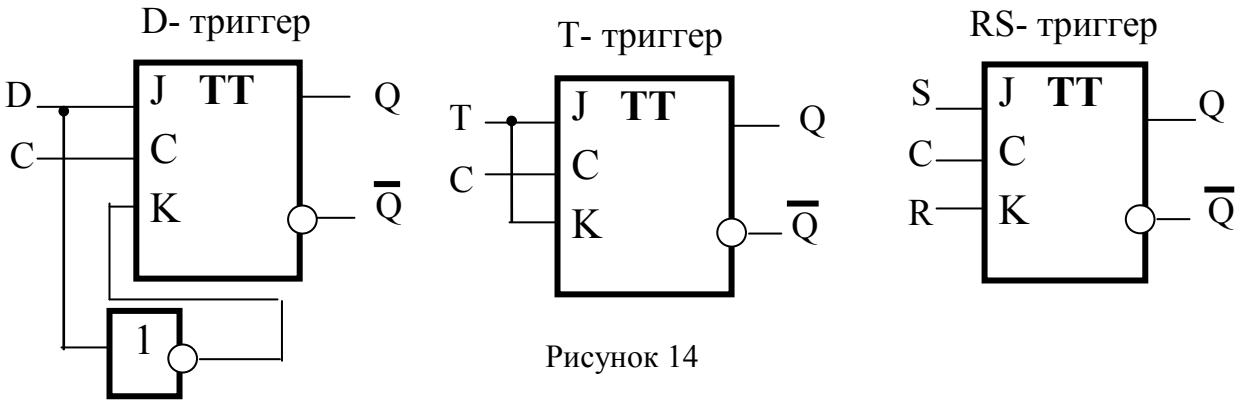


Рисунок 14

JK-триггер, кроме основных информационных входов и входа синхронизации, может иметь также дополнительные информационные входы, например, дополнительные инверсные асинхронные входы R и S, которые используются для установки триггера в 0 или 1 независимо от значения сигнала на входе синхронизации. Кроме того, триггер может иметь несколько входов J или K, объединенных по схеме И.

Регистры.

Наиболее распространенным узлом цифровой техники и устройств автоматики являются *регистры*. Регистры строятся на базе синхронных одно- и двухступенчатых RS и D-триггеров. Регистры могут быть реализованы также на базе JK-триггеров.

Регистры с *параллельным* приемом и выдачей информации служат для хранения информации и называются регистрами *памяти* или *хранения*. Запись новой информации в регистр осуществляется после установки на входах D0 ... Dm новой цифровой комбинации при поступлении синхроимпульса C. Количество разрядов записываемой цифровой информации определяется разрядностью регистра, которая, в свою очередь, определяется количеством триггеров, образующих этот регистр. Регистры памяти могут быть реализованы

на D-триггерах, если информация поступает на входы регистра в виде однофазных сигналов и на RS-триггерах, если информация поступает в виде парафазных сигналов. В некоторых случаях регистры могут иметь вход для установки выходов в состояние “0”. Этот асинхронный вход называют входом R “сброса” триггеров регистра. На рис. 15 приведены схемы четырехразрядных регистров памяти на D- и RS-триггерах, синхронизируемых уровнем и фронтом синхроимпульсов (обычно четыре триггера объединены в одном корпусе ИМС). На рисунке 15 показаны регистры хранения на D-триггерах, синхронизируемых фронтом (а) и на RS-триггерах, синхронизируемых фронтом (б). На рисунке 15, в показано УГО регистра.

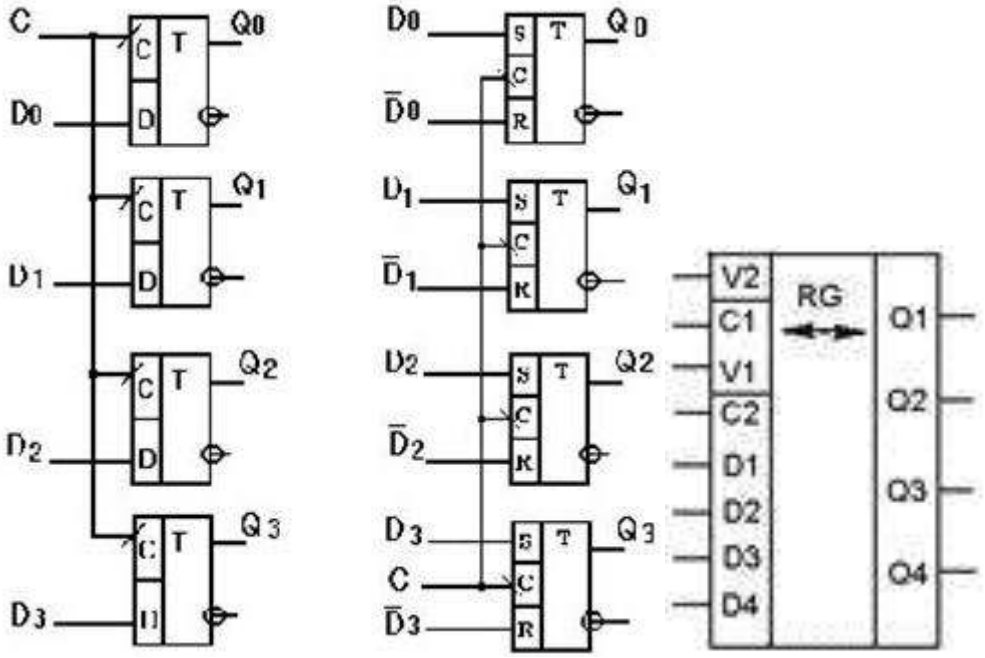


Рисунок 15 а б в

Регистры с последовательным приемом или выдачей информации называются сдвиговыми регистрами или регистрами *сдвига*. Они могут выполнять функции хранения и преобразования информации (умножение и деление чисел двоичной системы счисления, преобразование параллельного кода в последовательный и наоборот и т.д.).

На рисунке 16, а и 16,б приведены схемы четырехразрядных регистров сдвига, реализованных на D- и RS-триггерах, а временные диаграммы, поясняющие работу регистра сдвига, приведены на рисунке 17.

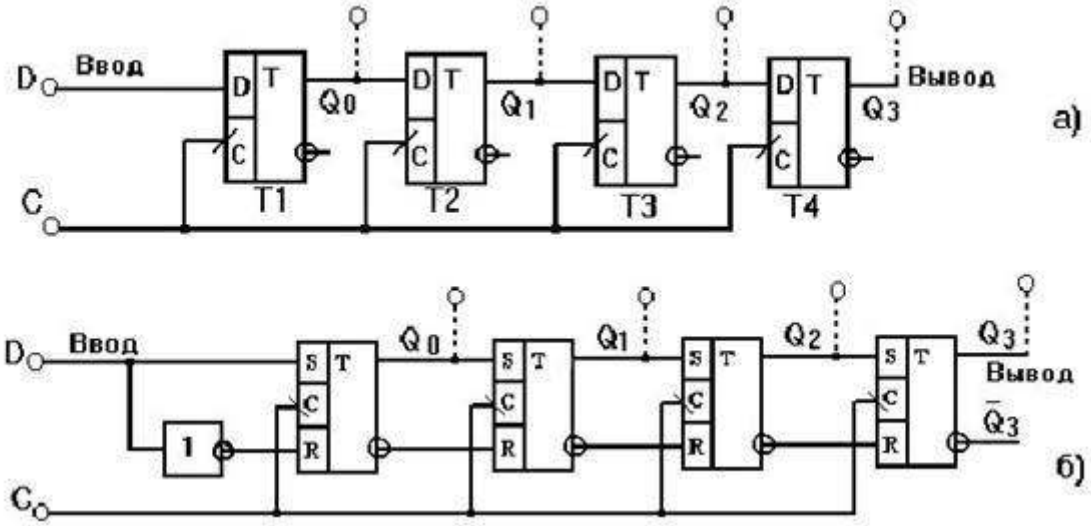


Рисунок 16

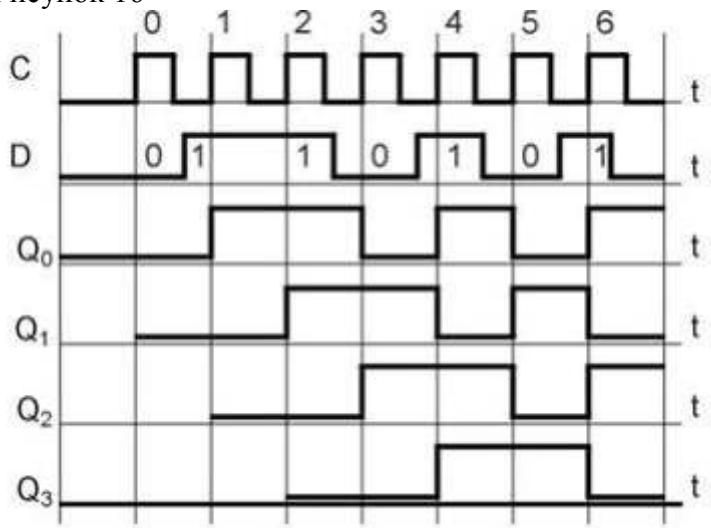


Рисунок 17

Порядок выполнения работы

Задание 1. Построить на элементах 2И-НЕ и 2ИЛИ-НЕ схемы асинхронных RS-триггеров (см. рисунок 3) и исследовать логику их работы в статическом режиме. Для этого собрать схемы с использованием пробников и переключателей.

Путем моделирования работы триггеров получить таблицы переходов и сравнить их с таблицей 1. Образцы схем для моделирования приведены на рисунке 18. Исследуемые схемы и таблицы занести в отчет.

Задание 2. Построить на элементах 2И-НЕ и 2-2И-2ИЛИ-НЕ схемы синхронных RS-триггеров (см. рисунок 4) и исследовать логику их работы в статическом режиме. Образцы схем для моделирования приведены на рисунке 19 и 20. В качестве элементов 2-2И-2ИЛИ-НЕ использована микросхема 7455, в которой располагается элемент 4-4И-2ИЛИ-НЕ. Исследуемые схемы и таблицы занести в отчет.

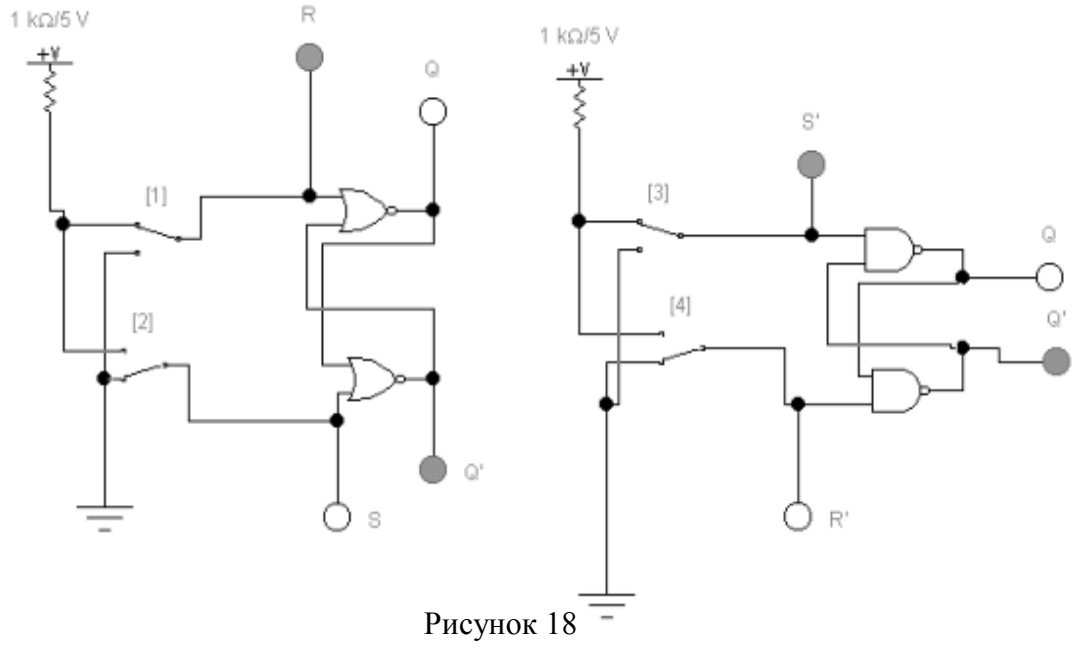


Рисунок 18

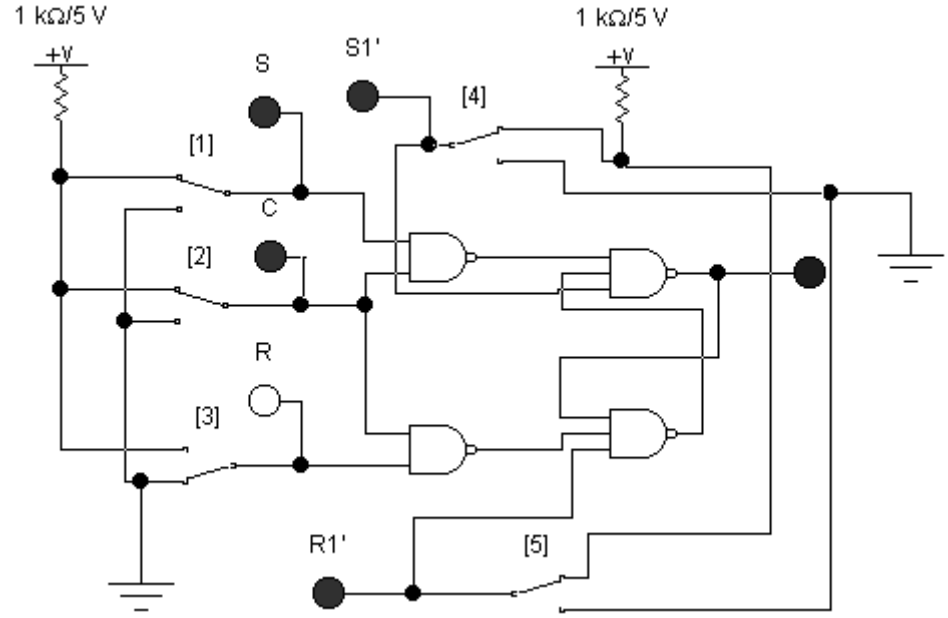


Рисунок 19

Задание 3. Исследовать в статическом режиме логику работы RS-триггера, который имеется в библиотеке программы. Для этого собрать схему, показанную на рисунке 21. Получить таблицу переходов триггера и сравнить ее с таблицей 1. Исследуемую схему и таблицу занести в отчет.

Задание 4. Исследовать в статическом режиме логику работы двухтактного RS-триггера. Для этого собрать схему, показанную на рисунке 22. Получить таблицу переходов триггера и сравнить ее с таблицей 1. Исследуемую схему и таблицу занести в отчет.

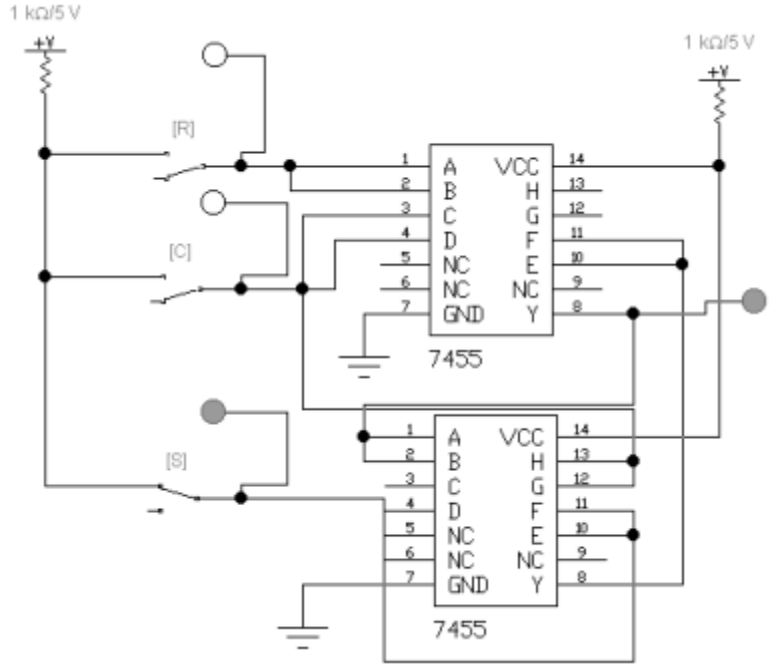


Рисунок 20

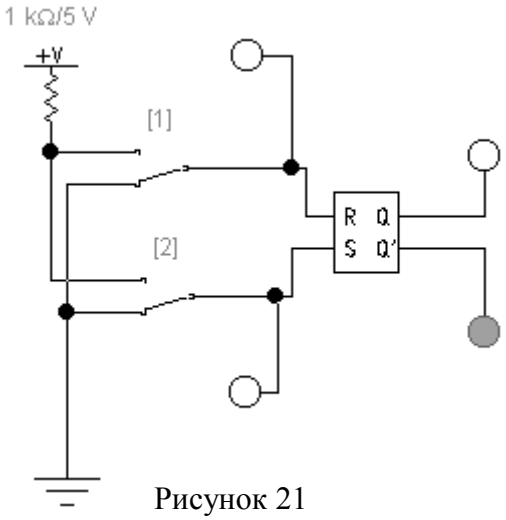


Рисунок 21

Задание 5. Исследовать в статическом режиме логику работы асинхронного D-триггера. Для этого собрать схему, показанную на рисунке 23. Получить таблицу переходов триггера и сравнить ее с таблицей 3. Исследуемую схему и таблицу занести в отчет.

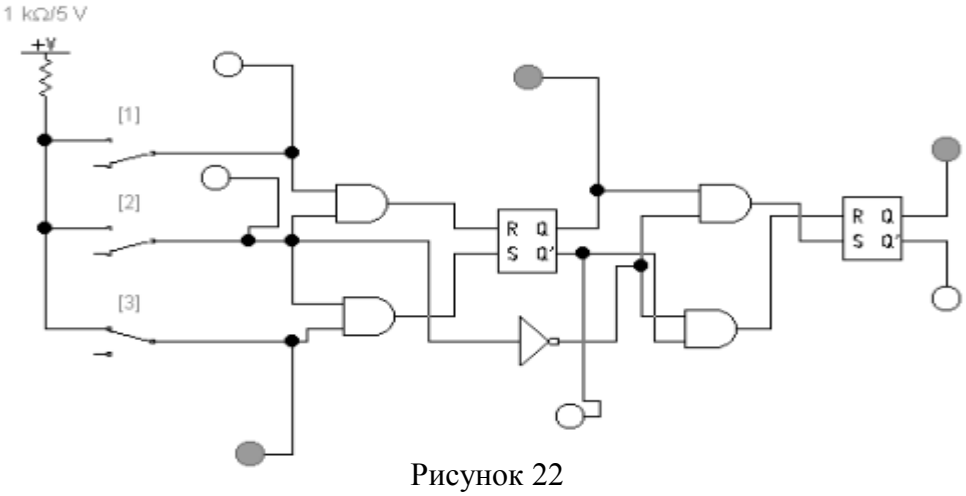


Рисунок 22

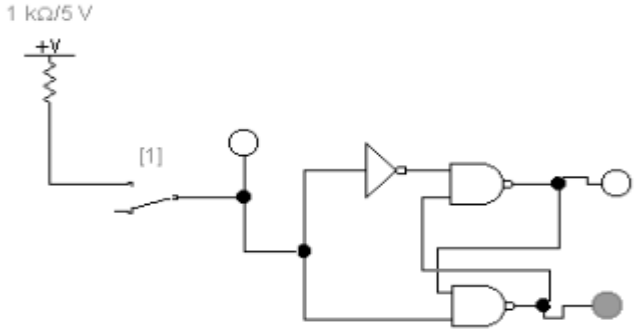


Рисунок 23

Задание 6. Исследовать в динамическом режиме логику работы асинхронного D-триггера. Для этого собрать схему, показанную на рисунке 24. Для визуального наблюдения работы схемы установить частоту генератора 1 Гц. Зарисовать полученную осциллограмму. Исследуемую схему и таблицу занести в отчет.

Задание 7. Собрать и исследовать в статическом режиме схему синхронного D- триггера на элементе 2И-2И-2ИЛИ-НЕ, в качестве которого использовать микросхему 7451 с 2-мя элементами 2И-2И-2ИЛИ-НЕ. Схема для исследования показана на рисунке 25. Результаты исследования занести в отчет.

Задание 8. Собрать и исследовать микросхему 7474, состоящую из 2-х синхронных D-триггеров. Схема показана на рисунке 26. Результаты исследования занести в отчет.

Задание 9. Собрать схему и исследовать работу асинхронного T-триггера, построенного на базе синхронного D-триггера в статическом режиме. Соответствующая схема показана на рисунке 27. В качестве синхронного D-триггера использовать микросхему 7474 с дополнительными асинхронными входами установки и сброса (инверсные входы R и S). Результаты исследования занести в отчет.

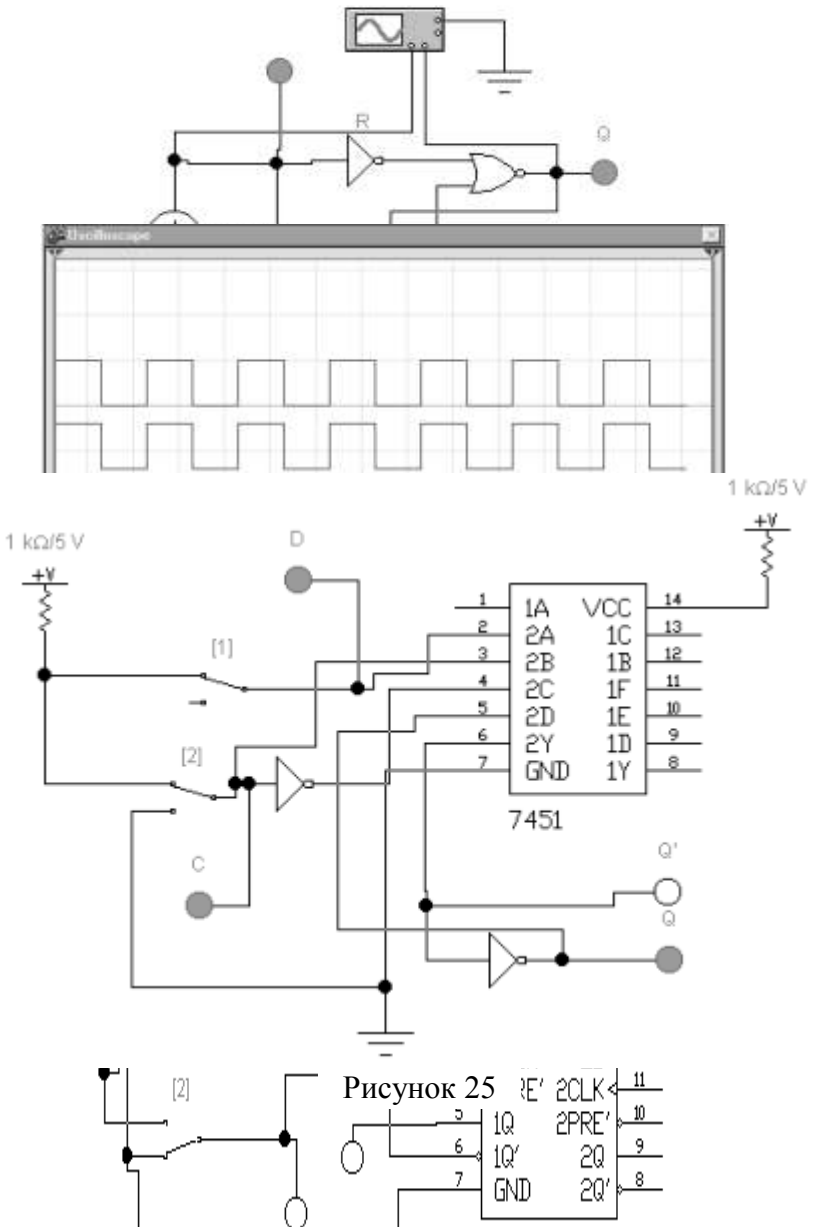


Рисунок 25

Задание 10. Исследовать работу синхронного JK-триггера в динамическом режиме. Для этого собрать схему, показанную на рисунке 28. При подаче на входы J и K сигналов высокого уровня, а на вход синхронизации импульсов от генератора, триггер будет работать в режиме переключения с частотой в два раза ниже, чем частота генератора. Для визуальной индикации подключить осциллограф к выходам генератора и триггера.

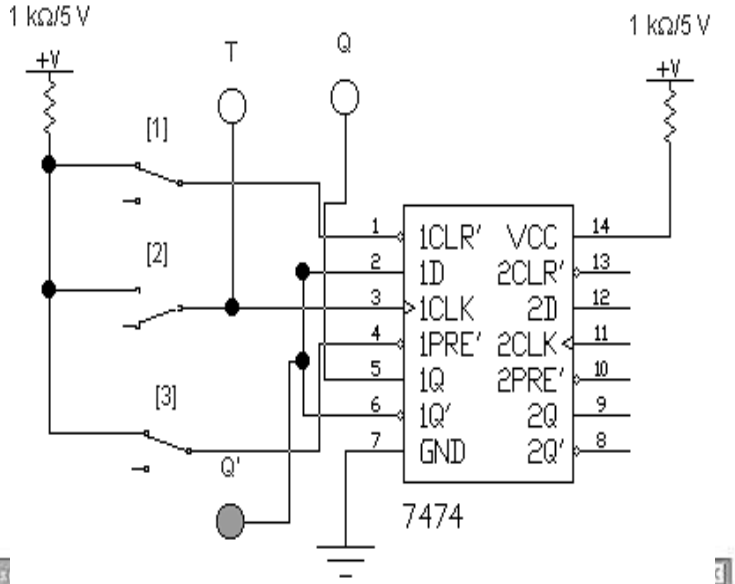


Рисунок 27

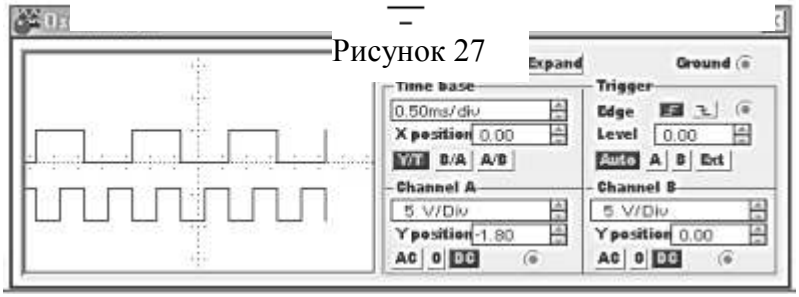


Рисунок 28

Задание 11. Собрать схему и исследовать работу синхронного JK-триггера в статическом режиме. Соответствующая схема показана на рисунке 29. В качестве синхронного JK-триггера использовать микросхему 74112. Результаты исследования занести в отчет.

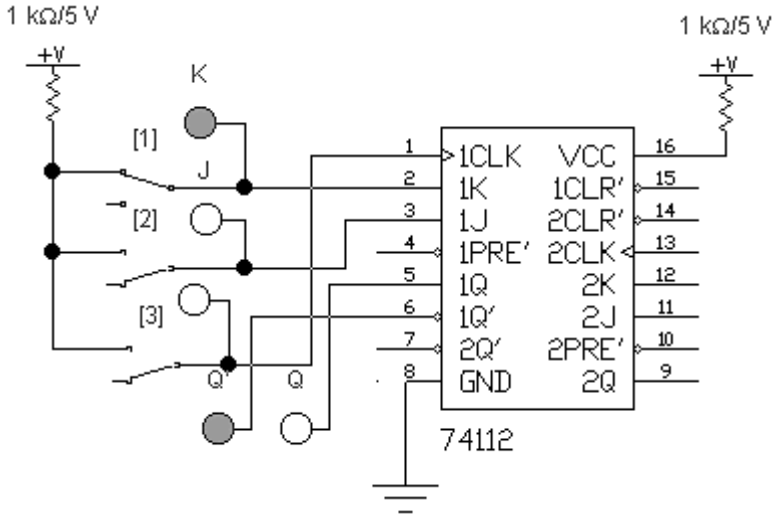


Рисунок 29

Задание 12*. **1)** Разработать и начертить схему электрическую функциональную рехразрядного параллельного регистра на базе D-триггеров синхронизиру

руемых фронтом для четных вариантов или на базе RS-триггеров, синхронизируемых фронтом для нечетных вариантов.

2) Разработать и начертить схему электрическую функциональную четырехразрядного регистра сдвига на базе на RS-триггеров, синхронизируемых фронтом, для четных вариантов или на базе D-триггеров, синхронизируемых фронтом, для нечетных вариантов.

3) Смоделировать параллельный регистр, разработанный в п. 12.1, в среде Electronics Workbench. Поочередно подать на входы D0 ... D3 код, соответствующий четырем младшим разрядам двоичного числа, равного номеру вашего варианта, и код на единицу меньший с помощью соответствующих ключей. Подать синхроимпульс С с помощью генератора слов Word Generation, включив его в ручном режиме *Step*, и убедиться в правильной работе параллельного регистра по состоянию логических пробников на его выходах.

4) Смоделировать регистр сдвига, разработанный в п. 12.2, в среде Electronics Workbench. Для имитации работы схемы подключить ее синхровход к генератору слов Word Generation, включив его в циклическом режиме *Sycle*. Подать на входы D0 ... D3 регистра код, соответствующий четырем младшим разрядам двоичного числа, равного номеру вашего варианта плюс три. Получить временные диаграммы входных и выходных сигналов сдвигающего регистра на экране логического анализатора Logic Analizer.

Содержание отчета

В отчет о выполненной работе включить следующие материалы:

1. тему и цель работы;
2. результаты выполнения заданий: исследуемые схемы, полученные таблицы переходов, временные диаграммы;
3. анализ полученных результатов;
4. выводы по работе.

Контрольные вопросы:

1. Из каких логических элементов можно построить схему триггера?

2. Чем отличаются синхронные триггеры от асинхронных триггеров?

3. Можно ли построить схему D-триггера на основе RS- триггера?

4. Как построить схему Т-триггера, если использовать схему RS- триггера и логические элементы?

5. В каких случаях таблица переходов JK-триггера совпадает с таблицей переходов RS-триггера, в каких случаях отличается?

6. Почему JK-триггер называется универсальным триггером?

7. Почему Т-триггер называется триггером со счетным входом?

8. На какое время может быть задержана установка синхронного D-триггера по отношению к сигналу на его входе?

9. На какое время может быть задержана установка в 1 асинхронного D-триггера по отношению к сигналу на его входе?

10. Чем отличается двухтактный триггер от однотактного триггера?

11. Каково назначение регистров?

12. По каким признакам классифицируют регистры?

13. Чем определяется разрядность регистров?

14. Как работает параллельный регистр?

15. Каким образом осуществить операции умножения и деления в двоичной системе счисления в реверсивном регистре?

16. Как произвести с помощью регистра преобразование последовательного кода числа в параллельный код и обратно?

17. Как обозначаются регистры на схемах электрических функциональных и принципиальных?

Практическая работа №8 «Регистры. Счетчики».

Краткие теоретические сведения

Регистрами называются последовательностные схемы с различными комбинациями последовательного и параллельного способов записи и считывания информации. Информация поступает и хранится в регистре в виде n-разрядных двоичных чисел. Регистры выполняются на основе триггеров. Основными видами регистров являются параллельный и последовательный.

Регистры с параллельной записью называются, также регистрами памяти. Они могут быть построены на триггерах «защелках». На рисунке 1 приведена схема 8-ми разрядного регистра памяти с общим входом управления записью информацией и ее условное обозначение. Высокий уровень на входе С переписывает информацию с входа на выход ($Q_i=D_i$), а низкий уровень – защелкивает данные.

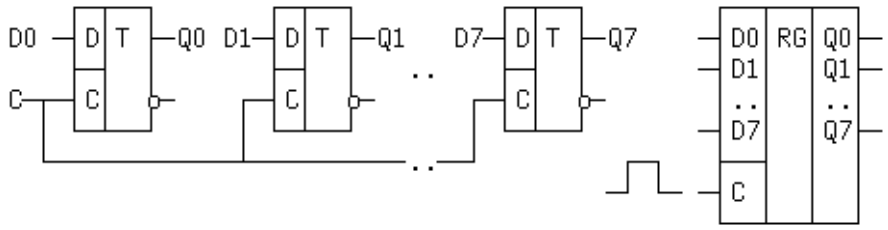


Рисунок 1 – Схема регистра памяти и его условное обозначение

Последовательные регистры или, как их еще называют, регистры сдвига (рисунок 2) выполняются на основе триггеров с динамическим синхровходом (справедливо для регистров с одним тактирующим сигналом).

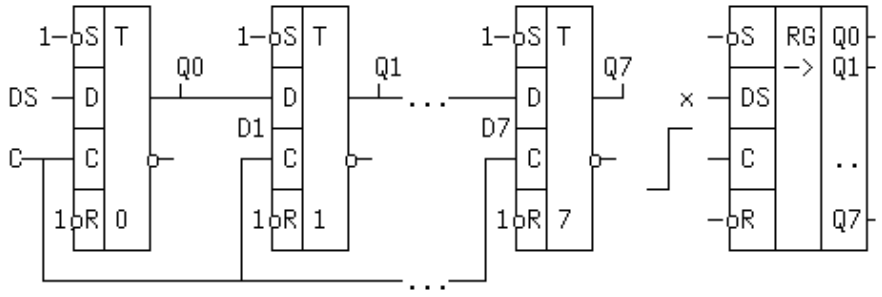


Рисунок 2 – Схема регистра сдвига

В двухтактных можно использовать и прозрачные регистры – «защелки». Функция записи в n-разрядном регистре сдвига на D-триггерах задается в виде условий: $D_0 = D_S = x, D_i = Q_{(i-1)}$, где $i = 1,2,\ldots,n-1$. D_S – вход для последовательной записи. С приходом очередного синхроимпульса С, сигнал с входа i-го триггера поступит на вход следующего (i+1)-го триггера. Однако на его выход эта информация не перепишется. На этом процесс сдвига данных на один разряд закончится до прихода следующего тактового сигнала. Каждый раз при С = 1 вся цепочка окажется прозрачной от входа DS до выхода Q_7 и значение D_S = x будет записано во все триггеры. В обозначениях регистров сдвига направление стрелки, указывающей сдвиг, условно. В разных справочниках, ее направление различно. Условно принимается, что сдвиг производится от младшего разряда к старшему.

Одно из применений регистров сдвига с параллельной загрузкой кода заключается в преобразовании параллельного формата данных в последовательный, передаче этих данных по однопроводной линии связи (вторая линия должна быть, как минимум «землёй» или экраном) и в обратном преобразовании последовательной информации в параллельную.

Указания по выполнению практической работы

1. Составить и исследовать схему трёхразрядного параллельного регистра на

основе синхронных D-триггеров.

2. Составить и исследовать схему трёхразрядного последовательного регистра на основе синхронных D-триггеров.

Контрольные вопросы:

1. Что называют регистрами?
2. Назовите основные виды регистров и охарактеризуйте каждый из них.

Практическая работа №9 «Классификация и характеристики запоминающих устройств на основе микросхем».

Цель работы.

Работа содержит сведения по установке и модернизации модулей оперативной памяти. Целью работы является облегчение учащимся освоения основных принципов установки модулей памяти на системную плату компьютера.

В результате обучающиеся должны знать:

- основные модификации модулей памяти, их отличие друг от друга;
- основные характеристики (параметры) модулей памяти;
- правила установки модулей памяти на системную плату

В результате обучающиеся должны уметь:

- правильно устанавливать модуль памяти на системную плату.

Задание.

1. Ознакомиться и получить навыки работы по установке и модернизации модулей оперативной памяти.

2. Ознакомиться и получить навыки измерение быстродействия оперативной памяти с помощью тестовых программ.

Теоретическая часть

3.1 Общие сведения

Оперативная память (рисунок 3.1) – это рабочая область для процессора компьютера. В ней во время работы хранятся программы и данные, которые сохраняются в ней только при включенном компьютере или до нажатия кнопки Reset. Поэтому перед выключением компьютера все данные, подвергнутые изменениям во время работы, необходимо сохранять на запоминающее устройство (например винчестер). При новом включении питания сохраненная информация вновь может быть загружена в память.

Рисунок 3.1 Модуль оперативной памяти

Память, применяемая для временного хранения инструкций и данных в компьютерной системе, получила название RAM (Random Access Memory – память с произвольной выборкой), потому что обращение происходит в любой момент времени к произвольно выбранной ячейке. Память этого класса подразделяется на два типа – память с динамической (Dynamic RAM, DRAM) и статической (Static RAM, SDRAM) выборкой.

В персональных компьютерах используется следующие типы памяти:

DRAM – Динамическая память. Широко использовался в ПК семейства 386 и 486, а так же первых поколениях Pentium. На сегодняшний момент самый медленный тип памяти.

EDO DRAM – Являлась основной для персональных компьютеров с процессором Pentium. Представляет собой память типа DRAM с расширенными возможностями вывода. Память этого типа работала на частоте шины не более 66 МГЦ. Время доступа к данным: от 50 до 70 нс. В настоящее время эти модули памяти используются для модернизации встроенной памяти на некоторых моделях внешних устройств (например, лазерных принтерах)

SDRAM – В настоящее время они используются в современных компьютерах с процессорами Pentium II/III. Память этого типа значительно быстрее EDO – время доступа к данным от 6 до 9 нс. Пропускная способность от 256 до 1000 Мбайт/с. Эти модули работают на частоте системной шины 66, 100 и 133 МГц.

DDR SDRAM – Улучшенная модификация памяти SDRAM. Время доступа к данным 5-6 нс. Пропускная способность – до 2,5 Гбайт. Поддерживаемая частота системной шины до 700 МГц.

RDRAM – Тип памяти разработанный для персональных компьютеров с процессором Pentium 4. Поддерживает рабочую частоту шины до 800 МГЦ. Время доступа к данным составляет 4 нс. Скорость передачи данных до 6 Гбайт/с.

В современных компьютерах вместо отдельных микросхем памяти используются модули памяти. SIMM (Single In Memory Module), DIMM (Dual In Line Memory Module) и RIMM (Rambus In Line Memory Module), представляющие собой небольшие платы, которые устанавливаются в специальные разъемы на системной плате или плате памяти. Отдельные микросхемы так припаяны к плате модуля, что выпаять и заменить их практически невозможно. При появлении неисправности заменяется весь модуль.

Модули SIMM изготавливаются 30 или 72-контактные. Первые из них меньше по размерам. 30-и контактные модули SIMM использовались в компьютерах с процессорами 386 и 486. 72-х контактные модули (рисунок 3.2) применялись с процессорами Pentium. В настоящее время данные модули практически не используются в современных компьютерных системах.

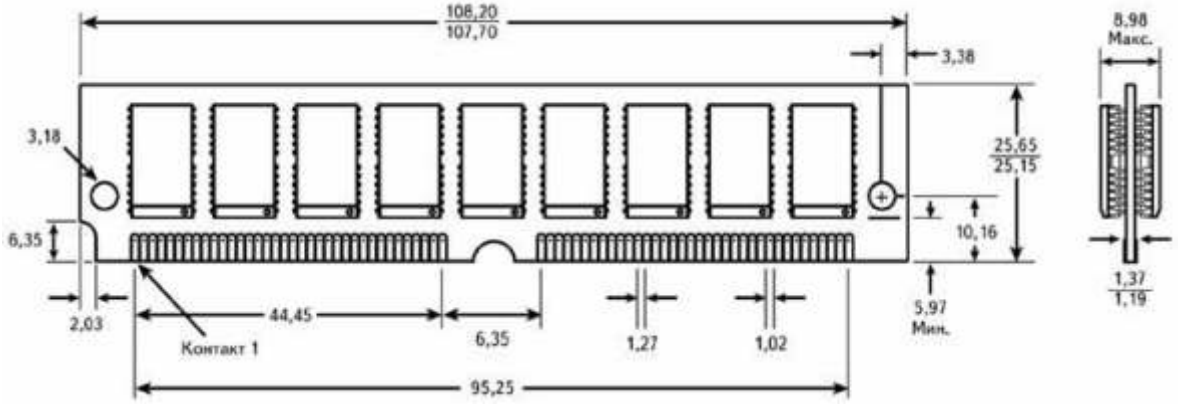


Рисунок 3.2 72-х контактный модуль памяти SIMM

Поэтому в новых системах с процессорами Pentium II/III используются 168-контактные модули DIMM (рисунок 7.3). В настоящее время для памяти DIMM SDRAM действуют спецификации PC100 и PC133, где цифры обозначают частоту синхронизации, при которой гарантированы работоспособность.

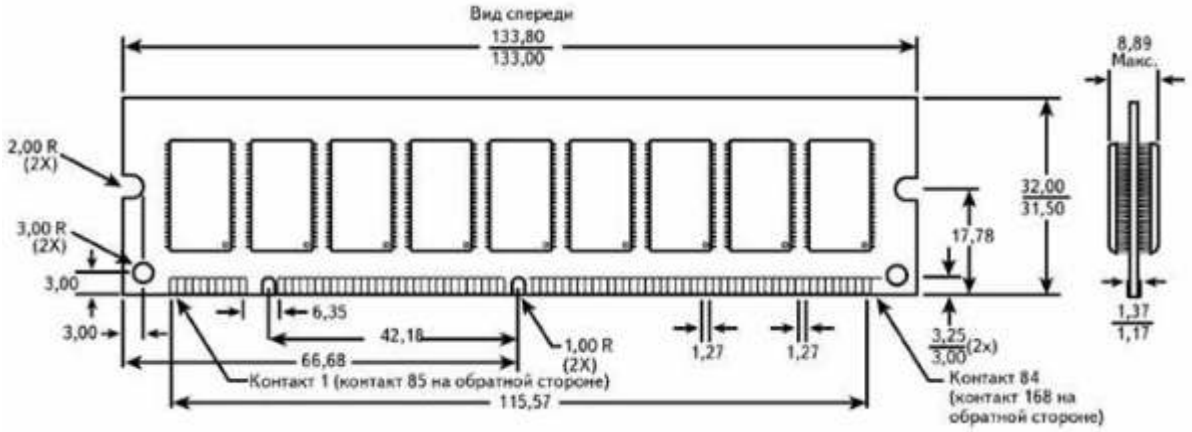


Рисунок 3.3 168-и контактный модуль памяти DIMM с микросхемами SDRAM

В системах с процессором Pentium IV широко используется модификация модуля DIMM – 184-х контактный модуль DDR SDRAM (Double Data Rate SDRAM) со спецификацией PC2100 или PC2700, которые работают на частоте шины более 266МГц. Модули DDR SDRAM имеют такие же размеры, как и модули DIMM, однако с существующими разъемами DIMM они полностью не совместимы.

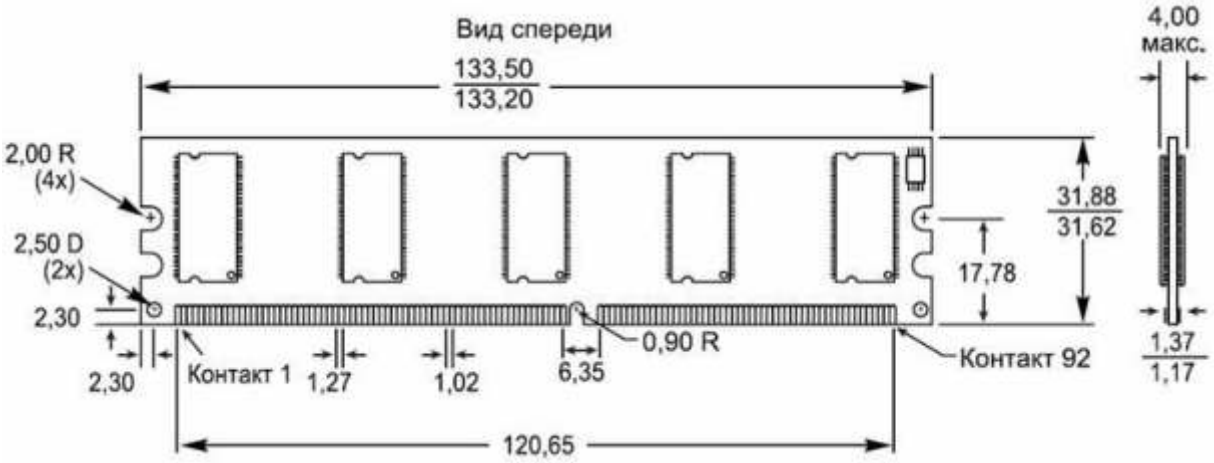


Рисунок 3.4 184-х контактный модуль памяти DDR SDRAM

Новой разработкой памяти для компьютерных систем является технология Rambus DRAM, который используется при производстве модулей памяти RIMM (рисунок 3.5). Данные модули могут работать на частоте 800МГц.

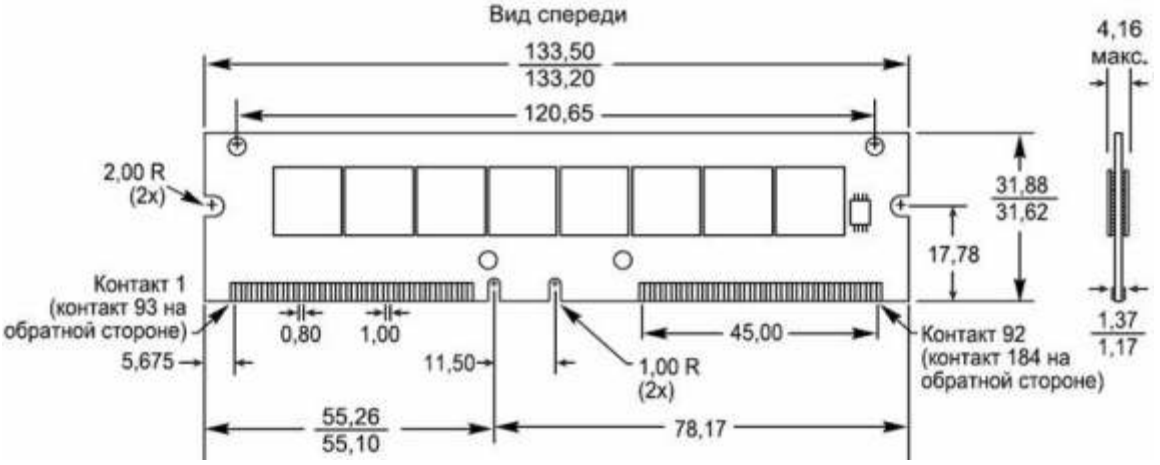


Рисунок 3.5 184-х контактный модуль памяти RIMM

3.2 Спецификации и характеристики оперативной памяти

3.2.1 Спецификации оперативной памяти

Стандарты на оперативную память устанавливаются ассоциацией JEDEC, которая устанавливает набор требований к модулям памяти для гарантированного обеспечения их работы в требуемых условиях. Регламентируется длина проводников в модуле памяти, ширина дорожек и расстояние между ними, электрические и другие параметры. В настоящее время для памяти SDRAM действуют спецификации PC100 и PC133, где цифры означают частоту синхронизации, при которой гарантирована работоспособность. Для модулей памяти DDR принято обозначать спецификацию по частоте передачи данных (например PC200 или PC333) или по пропускной способности – PC2100, PC 2700 (измеряется в Мбайт/с).

3.2.2 Характеристики оперативной памяти.

Быстродействие памяти

Быстродействие оперативной памяти и его эффективность выражается следующими характеристиками: временем доступа к данным и максимальная рабочая частота шины.

Время доступа микросхем памяти колеблется от 4 до 200 нс. (1нс – это время, за которое свет преодолеет расстояние в 30 см.) При замене неисправного модуля или микросхемы памяти новый элемент должен быть такого же типа, а его время доступа должно быть меньше или равно времени доступа заменяемого модуля. Таким образом, заменяющий элемент может иметь и более высокое быстродействие.

Регенерация данных

Для исключения утраты данных периодически производятся циклы регенерации данных с определенной частотой.

Контроль четности

Ранее для всех модуле памяти применялся контроль четности с целью проверки достоверности информации. Для этого при записи байта вычисляется сумма по модулю 2 всех информационных битов и результат записывается как дополнительный контрольный разряд. При чтении бита снова вычисляется контрольный разряд и сравнивается с полученным ранее.

Коррекция ошибок

Выявление и коррекция ошибок (ECC – Error Checking and Correction) – это специальный алгоритм, который заменил контроль четности в современных модулях памяти. Каждый бит данных включается более чем в одну контрольную сумму, поэтому при возникновении в нем ошибки можно восстановить адрес и исправить сбойный бит. При сбое в двух и более битах ошибка лишь фиксируется, но не исправляется.

3.3 Маркировка оперативной памяти

При приобретении модуля памяти необходимо обращать внимание на его маркировку. Корпус микросхемы и модуль памяти всегда имеют специальные обозначения, включающие наименование и знак фирмы изготовителя, дату выпуска, специальный код (рисунок 3.6).

Рисунок 7.6
Маркировка модулей памяти

Маркировка модулей памяти тесно связана с особенностями их технологии производства и тестирования. Все произведенные микросхемы делятся на три класса: A, B и C – в порядке понижения результатов.

Класс A – готовые микросхемы, прошедшие полный цикл тестирования, которые гарантированно работают в соответствии с заявленными характеристиками и имеют суще-

ственный запас по параметрам. Они также и самые дорогие, поскольку гарантируют работу в любых условиях.

Класс B – гарантировано отвечают заданным параметрам, но имеют меньший «запас прочности»

Класс C – модули памяти с небольшими дефектами, на этапе тестирования которых были выявлены ошибки. Данные модули могут быстро и хорошо работать в «домашних» системах, но использовать их в системах, где требуется высокая надежность – не рекомендуется.

Существует еще одна группа модулей памяти, чипы которой вообще не тестировались производителем на скорость и надежность. Это самые низкие по стоимости модули оперативной памяти. Зачастую на данных модулях на маркировке не указывается фирма производитель, либо маркировка отсутствует. Стабильность работы таких модулей памяти вызывает большие сомнения.

3.4 Установка модулей памяти

При установке и удалении памяти возможны следующие проблемы:

- накопление электростатических зарядов;
- повреждение выводов микросхем;
- неправильная установка модулей;

Чтобы предотвратить накопление электростатических зарядов при установке микросхем памяти, не надевайте одежду из синтетических тканей или обувь на кожаной подошве. Удалите все накопленные статистические заряды, прикоснувшись к корпусу системы до начала работы, или оденьте специальный браслет.

Каждая микросхема (или модуль) памяти должна быть установлена соответствующим образом. На одном конце микросхемы имеется маркировка. Это может быть вырез, круглое углубление или и то и другое. Гнездо микросхемы может иметь соответствующую маркировку. Ориентация выреза указывает положение первого вывода микросхемы.

3.4.1 Установка модулей памяти SIMM

a. Выключите питание компьютера и отсоедините сетевой шнур.

b. Возьмите модуль SIMM за верхний край платы и под небольшим углом осторожно вставьте микросхему в гнездо.

c. Убедитесь, что каждый вывод совпал с отверстием разъема, а затем надавливайте на микросхему двумя большими пальцами до тех пор, пока она полностью не войдет в разъем.

d. Надавив на края модуля, установите его вертикально (рисунок 3.7). При этом срабатывает механизм фиксации модуля (рисунок 3.8).

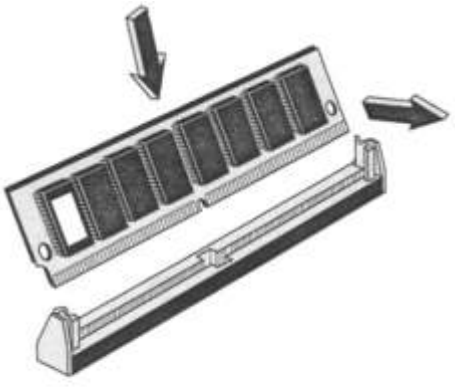


Рисунок 7.7
Установка модуля SIMM

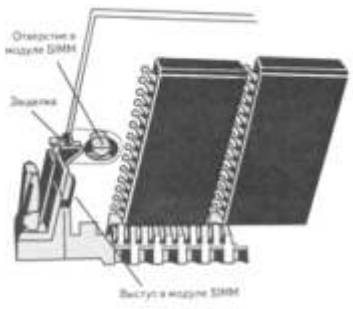


Рисунок 7.8
Механизм фиксации модуля SIMM

Ориентация модуля SIMM определяется вырезом, расположенным только с одной стороны модуля. В гнезде есть выступ, который должен совпасть с вырезом на одной стороне SIMM. Благодаря выступу установить модуль SIMM «наоборот» можно только в случае повреждения гнезда

3.4.2 Установка модулей DIMM и RIMM

Модули DIMM устанавливать легче, чем модули SIMM.

Подобно микросхемам SIMM, микросхемы DIMM имеют по краям ключи-вырезы, которые смещены от центра так, чтобы микросхемы могли быть однозначно ориентированы.

a. Выключите питание компьютера и отсоедините сетевой шнур.

b. Установите модуль в гнездо в вертикальном положении.

c. Надавите на верхнее ребро модуля памяти, так, чтобы он плотно вошел в гнездо и сомкнулись защелки, фиксирующие модуль в гнезде. Защелки DIMM находятся в прижатом состоянии, когда модуль вставлен в слот, и откинуты в стороны, когда модуль вынут (рисунок 3.9)

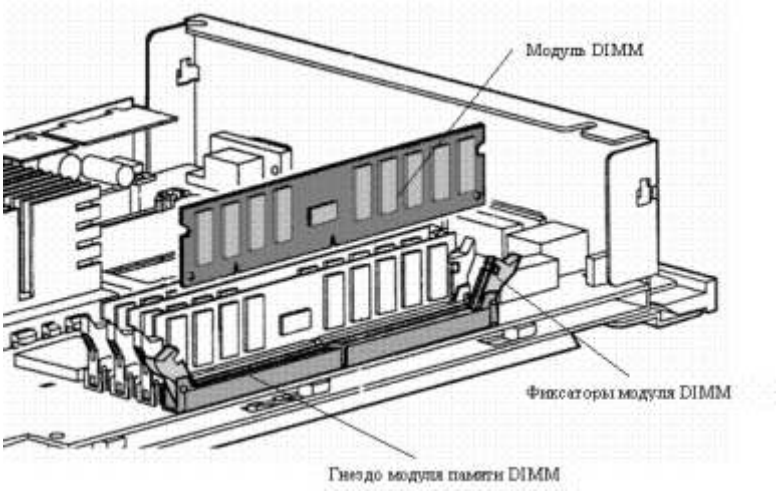


Рисунок 3.9 Установка модуля памяти DIMM

Для того, чтобы извлечь модуль DIMM из слота, нужно отжать защелки наружу (и вниз), и модуль выталкивается из слота.

Если модуль не проскальзывает легко в разъем и затем не фиксируется на своем месте, значит, он неправильно ориентировании не выровнен. Если к модулю приложить значительные усилия, можно сломать модуль или разъем. Если сломаны зажимы разъема, память не будет установлена на своем месте. В этом случае возможны сбои памяти.

Практическая часть.

1. Ход работы.

2. Провести тестирование оперативной памяти с помощью Memtest86 3.5(или использовав другую утилиту)

2. Протокол работы №1. Определение основных характеристик оперативной памяти.

3. Протокол работы №2. Сравнительная характеристика оперативной памяти

Протокол работы №1

Определение основных характеристик оперативной памяти

Тип памяти	Рабочая частота	разрядность	Время доступа	Время рабочего цикла	Пропускная способность

подсчитать максимально достижимую пропускную способность при обработке зависимых данных можно используя для этого следующую формулу:

$$C = \frac{N}{T*8*1.024}$$

здесь: C - пропускная способность (Мегабайт/с), N - разрядности памяти (бит), T - полное время доступа (нс.)

Протокол работы №2

Сравнительная характеристика оперативной памяти

Тип памяти	Рабочая частота	разрядность	Время доступа	Время рабочего цикла	Пропускная способность

Практическая работа №10 «Современные модули памяти ОЗУ».

Цель работы: ознакомиться с распределением адресного пространства памяти. Исследовать обмен данными процессора с памятью.

2. Порядок выполнения работы:

2.1. Повторить теоретический материал.

Распределение адресного пространства памяти

Все адресное пространство памяти компьютера разделяется на несколько областей, что связано, в первую очередь, с необходимостью обеспечения совместимости с первыми компьютерами семейства. В компьютере IBM PC XT на процессоре i8088 процессор мог адресовать 1 Мбайт памяти (20 адресных разрядов). Но все программные и аппаратные средства строились исходя из предположения, что доступное адресное пространство — только младшие 640 Кбайт (тогда это казалось вполне достаточным). Данная область памяти получила название стандартной памяти (Conventional memory). Именно в пределах этих 640 Кбайт (адреса 0…9FFFF) работает операционная система MS DOS и все ее прикладные программы.

Первые 1024 байта (адреса 0…3FF) хранят таблицу векторов прерывания (Interrupt Vectors) объемом 256 двойных слов, формируемую на этапе начальной загрузки. Однако если процессор работает в защищенном режиме, таблица векторов может располагаться в любом другом месте памяти.

Адреса 400…4FF отводятся под область переменных BIOS (BIOS Data Area). Подробнее о BIOS будет рассказано в следующем разделе.

Адреса 500…9FFFF включают в себя область операционной системы DOS (DOS Area) и память пользователя (User RAM).

Оставшиеся от 1 Мбайта памяти 384 Кбайта (адреса A0000…FFFFF), зарезервированные под другие системные нужды, называются UMA (Upper Memory Area) — область верхней памяти или UMB (Upper Memory Blocks) — блоки **верхней памяти** или High DOS Memory.

Пространство видеопамяти (адреса A0000…BFFFF) содержит области для хранения текстовой и графической информации видеоадаптера.

Пространство памяти с адресами E0000…FFFFF отведено под системную постоянную память компьютера ROM BIOS.

В этой же области выделено окно размером в 64 Кбайта (page frame) с адресами D0000…DFFFF, через которое программы могли получать доступ к **дополнительной (отображаемой) памяти** (Expanded memory) объемом до 32 Мбайт, оставаясь в пределах того же 1 Мбайта адресуемой памяти. Это достигается путем поочередного отображения четырех страниц по 16 Кбайт из дополнительной памяти в выделенное окно. При этом положение страниц в дополнительной памяти можно изменять программным путем. Понятно, что работать с дополнительной памятью менее удобно, чем с основной, так как в каждый момент компьютер «видит» только окно в 64 Кбайт. Поэтому сейчас она применяется довольно редко.

В настоящее время область памяти с адресами C0000…DFFFF чаще используется для оперативной и постоянной памяти , входящей в состав различных адаптеров и плат расширения компьютера.

В результате логическая организация адресного пространства в пределах 1 Мбайт получилась довольно сложной (рис.1). И такую же организацию должны поддерживать все персональные компьютеры семейства IBM PC для обеспечения совместимости с предшествующими моделями.

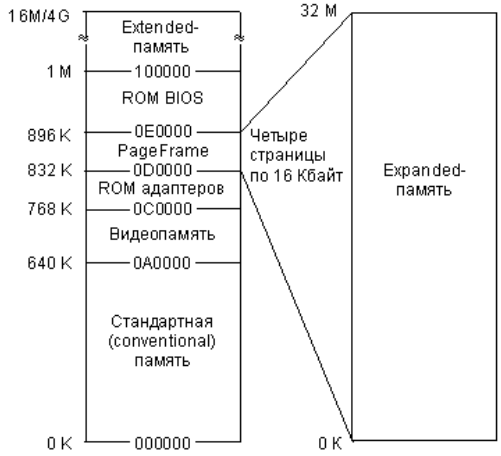


Рис. 1. Распределение адресного пространства памяти.

При дальнейшем расширении адресуемого пространства памяти в последующих моделях компьютеров вся память объемом свыше 1 Мбайт получила название **расширенной памяти** (Extended memory). Для доступа к ней микропроцессор должен переходить из реального режима в защищенный и обратно. Общий объем памяти персонального компьютера (верхняя граница расширенной памяти) может доходить до 16 Мбайт (24 разряда адреса) или до 4 Гбайт (32 разряда адреса).

Особого упоминания заслуживает так называемая **теневая** память (Shadow RAM), представляющая собой часть оперативной памяти , в которую при запуске компьютера переписывается содержание постоянной памяти , и заменяющая эту постоянную память на время работы компьютера. Необходимость данной процедуры вызвана тем, что даже сравнительно медленная динамическая оперативная память оказывается все-таки быстрее, чем постоянная память . Постоянная память часто заметно сдерживает быстродействие компьютера. Поэтому было предложено выделять часть оперативной памяти для исполнения обязанностей как системной постоянной памяти ROM BIOS, так и постоянной памяти , входящей в состав дополнительных адаптеров, которые подключаются к компьютеру. Переписывание информации обычно предусмотрено в программе начального пуска.

В связи с особенностями работы динамической памяти для сокращения времени доступа к ней применяются специальные режимы работы оперативной памяти : режим расслоения (интерливинг) и страничный режим.

Использование режима интерливинга предполагает не совсем обычное разбиение памяти на банки (части). Если при обычном разбиении (последовательной адресации) адреса следующего банка начинаются после окончания адресов предыдущего, то при интерливинге адреса банков чередуются. То есть, например, после первого адреса первого банка следует первый адрес второго банка, затем второй адрес первого банка и второй адрес второго банка и т.д. Получается, что в одном банке четные слова, а в другом — нечетные. Таких чередующихся банков может быть не только два, а четыре, восемь, шестнадцать. Объемы банков при этом должны быть одинаковыми. В результате такого подхода появляется возможность начинать обращение к следующему слову еще до окончания процесса доступа к предыдущему.

Страничный режим предполагает постраничную работу микросхем памяти, когда выбор страницы производится один раз на всю страницу, а выбор ячейки внутри страницы может происходить гораздо быстрее. Для поддержки обоих режимов применяются специальные технологические решения.

Здесь же отметим, что при замене памяти компьютера или при установке дополнительных банков памяти надо строго следовать рекомендациям изготовителей системных плат, так как порядок заполнения банков может быть далеко не очевиден.

Практическая часть работы.

- Как процессор обменивается данными с памятью

Как процессор обменивается данными с памятью

1. Краткие сведения об устройстве памяти

Каждому, кто изучал хотя бы минимальный курс информатики, известно, что все данные и программы их обработки хранятся в компьютере в **дискретном двоичном** виде. Согласно классическим принципам, минимальной допустимой информацией является 1 **бит** и именно бит служит основой компьютерной памяти, ее минимальным конструктивным элементом. В настоящее время благодаря успехам в технологии производства миниатюрных электронных схем сформулированный тезис не имеет столь очевидных (в прямом смысле этого слова!) доказательств, но, тем не менее, своей актуальности не утратил.

Бит слишком маленькая единица информации, чтобы быть достаточной для представления практически полезных данных. Известно, например, что для сохранения одного символа требуется 8 бит, стандартного целого числа – 16, а разрядность целочисленных данных в современных процессорах достигла 32 бит. Следовательно, обеспечивать доступ к каждому отдельному биту памяти едва ли нецелесообразно. Начиная с третьего поколения, в ЭВМ фактически сложился стандарт организации памяти, при котором минимальной считываемой порцией информации является 1 **байт**. Кроме того, для работы с более крупными данными современные процессоры способны одновременно считывать несколько байт, начиная с заданного (как правило, два или четыре).

Итак, минимальной единицей обмена информацией с памятью в современных компьютерах является 1 байт. Каждый байт имеет свой идентификационный номер, по которому к нему

можно обращаться – его принято называть **адресом**. Адреса соседних байтов отличаются на единицу, зато для двух 32-разрядных чисел, хранящихся в памяти «друг за другом», эта разница по понятным причинам равняется четырем. Практически при обращении к памяти задается **адрес начального байта** и **их требуемое количество** (см. 3.3 и 3.2 соответственно).

2. Задание размера данных в команде

В семействе процессоров Intel количество считываемых или записываемых байт определяется кодом машинной инструкции. Учитывая, что первые представители этого семейства имели разрядность 16, и лишь начиная с модели 80386 перешли к 32 разрядам, система задания требуемого количества байт выглядит немного запутано. Так, коды команд обращения к байту или слову (вполне естественным образом) отличаются одним битом. Например, байтовая команда *MOV AL,1* имеет код *B0 01*, а двухбайтовая *MOV AX,1* кодируется *B8 01 00* (длина команды увеличилась из-за размера константы!); легко убедиться, что коды операций *B0* и *B8* действительно имеют отличие в единственном бите. Что касается четырехбайтовой команды *MOV EAX,1*, то код ее операции абсолютно такой же, как и у двухбайтовой команды (только константа 1 еще «длиннее» – 4 байта). Оказывается, что две эти одинаковые по кодам команды процессор различает по установленному режиму: при обработке 32-разрядного участка памяти константа заносится в полный регистр EAX, а 16-разрядного – в его младшую половину AX. Для изменения «режима по умолчанию» служит специальный префиксный код (так называемый *префикс переключения разрядности слова*, равный 66h), который для следующей за ним инструкции изменяет стандартный режим на противоположный. В результате при работе в 16-разрядном сегменте код *66 B8 01 00 00 00* реализует именно 32-разрядную операцию записи в EAX единицы.

Примечание. Не пытайтесь проверить данное утверждение путем ввода в Debug команды *MOV EAX,1* – к сожалению, отладчик «не понимает» мнемоник, содержащих расширенные регистры!

Таким образом, мы видим, что разрядность команды определяется ее кодом (и, может быть, некоторыми «внешними» по отношению к программе факторами). Полученный вывод позволяет нам в дальнейшем ограничиться рассмотрением методов адресации для данных какой-либо фиксированной длины: для остальных достаточно будет просто написать другой код операции.

3. Задание адреса данных в команде

В предыдущем разделе было показано, что для понимания методов адресации достаточно рассмотреть способы задания начального адреса при фиксированной длине данных. Поэтому в дальнейшем мы везде будем полагать, что данные имеют двухбайтовый размер. Для упрощения понимания идей адресации в большинстве примеров ограничимся также одной (самой простой!) инструкцией переписи данных, которая имеет мнемоническое обозначение MOV. Подчеркнем, что принятые допущения нисколько не ограничат общности наших знаний по описываемому вопросу.

Итак, перейдем к рассмотрению основных принципов обращения к данным, принятых в компьютерах семейства IBM PC на базе процессоров фирмы Intel.

3.1. Адресация простых данных

Под простыми данными принято понимать такие, которые хранят в себе только одно значение, например, целое число. Сложные данные, напротив, включают в себя несколько значений, причем даже не обязательно одного типа; простейшим примером «однородных» сложных данных является массив, о котором мы будем говорить позднее. Не следует путать сложные структуры, состоящие из нескольких более простых значений, в частности из набора целых чисел, с простыми данными, занимающими в памяти нескольких байт, например, с отдельно взятым 32-разрядным целым числом.

Будем пока рассматривать адресацию простых данных. Как выбрано выше, это будут целые 16-разрядные числа.

Простейшие и очень часто используемые инструкции обработки данных, которые выполняются в регистрах микропроцессора, настолько естественны, что легко понимаются на интуитивном уровне. В частности, речь идет о случаях, когда в регистр заносится копия содержимого другого регистра (*MOV AX,BX*) или константа (*MOV AX,30*). Тем не менее, с теоретической точки зрения это уже простейшие методы адресации данных, которые для процессоров семейства Intel принято называть ***регистровой*** и ***непосредственной*** адресацией соответственно.

Чуть более сложным является случай, когда данные извлекаются из конкретной ячейки памяти, например, *MOV AX,[30]*; такой метод получил название ***прямой*** адресации. Приведенная в качестве примера команда считывает из памяти два байта начиная с адреса 30 и помещает их в 16-разрядный регистр *AX*. Обязательно обратите внимание на наличие в записи квадратных скобок, которые всегда появляются при ссылке на содержимое памяти.

Подобно тому, как заключение в квадратные скобки константы приводит нас к появлению прямой адресации, аналогичный прием для регистра (например, *MOV AX,[BX]*) также порождает новый (и очень важный!) метод адресации – ***косвенный регистровый***. Его суть заключается в том, что содержимое регистра рассматривается не как данные, а как адрес памяти, где эти данные расположены.

Для понимания сущности косвенной адресации можно рассмотреть следующую аналогию: при подготовке к экзамену ученик открывает свой конспект по требуемой теме, а там вместо текста для ответа видит запись: «материал по этому вопросу прочитать в § 25». Иными словами, вместо готовой информации дается ссылка на нее.

Или еще одна аналогия из книги: «косвенная адресация похожа на операцию "для передачи (кому-то)", выполняемую почтовой службой США, когда указанный адрес не является реальным адресом получателя, а является адресом друга или родственника».

Чтобы закрепить четыре изученных метода адресации, рассмотрим несложный пример, который складывает два целых числа с адресами 200 и 204. Его выполнение зафиксировано в протоколе 1 и состоит из нескольких действий: ввод программы (команда ***a***) и ее вывод для контроля набора (***u***); ввод (***e200***) и вывод чисел (***d200***); контроль состояния регистров до выполнения программы (***r***); пошаговое выполнение четырех команд с выводом значений регистров после каждого шага (***t4***).

Надеемся, что вы без труда самостоятельно разберут приводимый протокол 1 и повторят соответствующий диалог в Debug. Не забудьте только о цели эксперимента – поработать с четырьмя различными способами адресации. Поэтому при анализе программы обязательно обратите внимание на то, какие методы использованы в каждой из команд.

Протокол 1

```
-a
1423:0100                       mov                     ax,[204]
1423:0103                       mov                     bx,200
1423:0106                       mov                     cx,[bx]
1423:0108                       add                     ax,cx
1423:010A
-u
1423:0100           A10402              MOV             AX,[0204]
1423:0103           BB0002              MOV             BX,0200
1423:0106           8B0F                MOV             CX,[BX]
1423:0108           01C8                ADD             AX,CX
1423:010A           0000                ADD             [BX+SI],AL
...
-e200       5       0       0       0       3       0
-d200
1423:0200  05 00 00 00 03 00 00 00-00 00 00 00 00 00 00 00  ................
...
-r
AX=0000  BX=0000 CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=1423 ES=1423  SS=1423  CS=1423  IP=0100   NV UP EI PL NZ NA PO NC
1423:0100    A10402          MOV         AX,[0204]                DS:0204=0003
-t4

AX=0003 BX=0000 CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=1423 ES=1423  SS=1423  CS=1423  IP=0103   NV UP EI PL NZ NA PO NC
1423:0103           BB0002              MOV             BX,0200

AX=0003  BX=0200 CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=1423 ES=1423  SS=1423  CS=1423  IP=0106   NV UP EI PL NZ NA PO NC
1423:0106    8B0F            MOV         CX,[BX]                  DS:0200=0005

AX=0003  BX=0200 CX=0005  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=1423 ES=1423  SS=1423  CS=1423  IP=0108   NV UP EI PL NZ NA PO NC
1423:0108           01C8                ADD             AX,CX

AX=0008 BX=0200  CX=0005  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=1423 ES=1423  SS=1423  CS=1423  IP=010A   NV UP EI PL NZ NA PO NC
1423:010A 0000        ADD     [BX+SI],AL                DS:0200=05
```

Примечание. В очередной раз напоминаем, что по принятому в IBM PC соглашению байты данных хранятся в памяти «задом наперед», т.е. байты 05 00 следует расшифровывать как 0005.

Задание

Составить и реализовать в Debug свою программу (аналогичную протоколу 1) в соответствии с вариантом таблицы 1

Таблица 1

Сложение целых чисел с адресами		
№ варианта	**Числа**	**Адреса**
1	6 и 4	300 и 304
2	7 и 5	400 и 404
3	8 и 6	500 и 504
4	9 и 7	600 и 604
5	A и 8	700 и 704
6	B и 9	800 и 804
7	C и A	900 и 904
8	D и B	A00 и A04
9	E и C	B00 и B04
10	F и D	C00 и C04
11	10 и E	D00 и D04
12	11 и F	E00 и E04
13	12 и 10	F00 и F04
14	13 и 11	310 и 314
15	14 и 12	410 и 414
16	15 и 13	510 и 514
17	16 и 14	610 и 614
18	17 и 15	710 и 714
19	18 и 16	810 и 814
20	19 и 17	910 и 914
21	1A и 18	A10 и A14
22	1B и 19	B10 и B14
23	1C и 1A	C10 и C14
24	1D и 1B	D10 и D14
25	1E и 1C	E10 и E14
26	1F и 1D	F10 и F14
27	20 и 1E	320 и 324
28	21 и 1F	420 и 424
29	22 и 20	520 и 524
30	23 и 21	620 и 624
31	24 и 22	720 и 724
32	25 и 23	820 и 824
33	26 и 24	920 и 924
34	27 и 25	A20 и A24

2.3. Содержание отчета:

- Тема и цель работы;
- Рис. 1. Распределение адресного пространства памяти.
- Листинг введеной программы в соответствии с вариантом.

3. Проверка знаний:

- тест (допуск к выполнению теста - проверка выполнения практической части работы).

Практическая работа №11,12 «Аналого-цифровые преобразователи». **(3 часа)**
«Аналого - цифровые»

«Цифро - аналоговые преобразователи»

1. Цель работы:

1.1 Ознакомиться с принципами построения и действия АЦП, ЦАП

1.2 Исследовать работу АЦП, ЦАП

2. Литература

2.1 Калабеков Б.А., Мамзелев И.А. Цифровые устройства и микропроцессорные системы. М.: Радио и связь, 1997

3. Подготовка к работе

3.1 Повторить материал по теме "Цифро-аналоговые и аналого-цифровые преобразователи информации" 2.1 с.156-173.

3.2 Ответить на вопросы для допуска к работе:

1) Назначение ЦАП
2) Назначение АЦП
3) Этапы аналого-цифрового преобразования.

4. Основное оборудование

4.1. Персональный компьютер

4.2. пакет программ Electronic Workbench

5. Задание

5.1 Исследовать работу схемы ЦАП.

5.2 Исследовать работу АЦП.

6. Порядок выполнения работы

Исследовать библиотечные АЦП, ЦАП

6.1 Соберите схему, представленную на рисунке 6.3.1, используя библиотечные АЦП, ЦАП

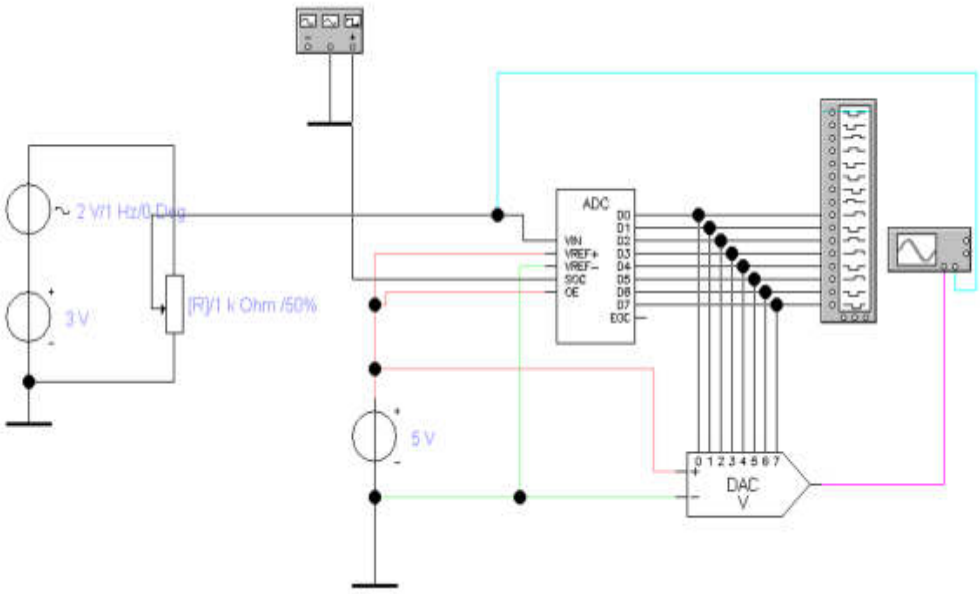


Рисунок 6.1 - схема для исследования библиотечных АЦП, ЦАП

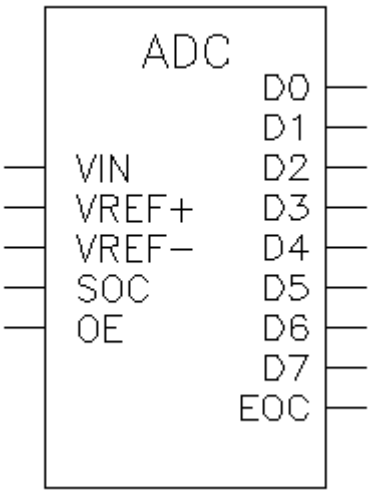
6.2 В отчете опешите назначение каждого элемента схемы, ответьте на вопросы:

Какой элемент схемы задает амплитуду и частоту аналогового входного сигнала?

Какой элемент схемы определяет «точность» преобразования аналогового сигнал в цифровой:

Примечание для пункта 6.2:

Обозначение АЦП в программном пакете Electronic workbench



Назначение входов:
VIN - вход для источника преобразуемого сигнала;
VREF+; VREF- - вход для источника опорного напряжения;
SOC - Вход синхронизации;
OE - разрешение на выдачу двоичной комбинации на выходы D0...D7;
EOC - сигнал готовности данных (например, при выдаче данных на ЭВМ)
6.3 Установите параметры входного сигнала: Амплитуда: 5 Вольт, частота 1 Герц
Установите частоту, с которой сигнал будет оцифровываться, равной 1 Гц (вход синхронизации)
Получите осциллограмму цифрового и аналогового сигнала на осциллографе, занесите её в отчет.
Увеличивая частоту на входе синхронизации АЦП получите осциллограммы для частот:
2, 4, 8, 16, 32, 64, 128 Гц (рисунок 6.3.2)

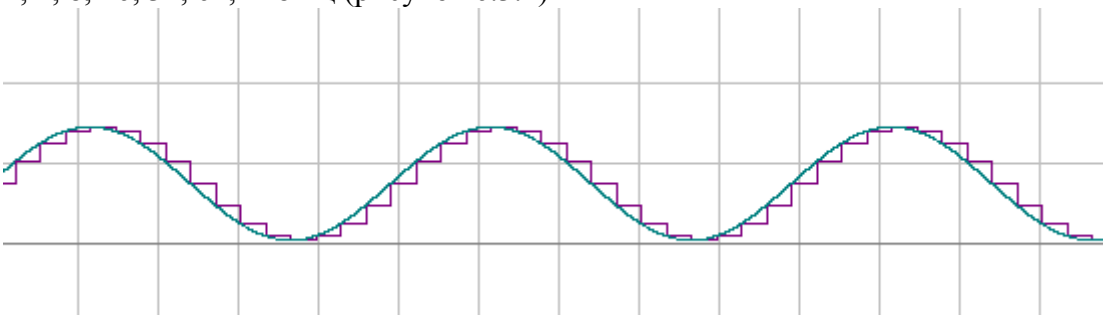


Рисунок 6.3.2 - пример оцифрованного сигнала

7. Содержание отчета

Наименование работы
Цель работы
Исследуемая схема ЦАП и АЦП.
Осциллограммы и графики согласно задания.
Выводы по работе ЦАП и АЦП
Ответы на контрольные вопросы в письменном виде.

8. Контрольные вопросы

Назначение ЦАП и АЦП.

Этапы аналого-цифрового преобразования.

Принцип действия АЦП последовательного приближения и АЦП параллельного типа, схемы, отличия, преимущества (письменно).

Схемы ЦАП, основные виды, принцип действия?

Преобразование между аналоговыми и цифровыми величинами – основная операция в вычислительных и управляющих системах, поскольку физические параметры, такие, как температура, перемещение, и напряженность магнитного поля, являются аналоговыми, а большинство практических методов обработки, вычисления и визуального представления информации – цифровыми. Путем преобразования в цифровую форму с помощью АЦП,

расположенных в оконечном устройстве, реализуются высокоскоростные, малошумящие, устойчивые и дешевые системы передачи данных на большие расстояния.

Так как АЦП и ЦАП по существу являются устройствами сопряжения, основную схему преобразования необходимо приспосабливать для удовлетворения требований различных применений. Для адаптации необходимы суммирующие регистры, буферы, цифровой узел синхронизации и источник опорного напряжения: их часто предусматривают внешними по отношению к преобразователю. Очевидно, что точное определение того, что содержится в модуле преобразователя, существенно влияет на цену и затрудняет сравнение конкурирующих моделей.

ЦАП можно представить как цифровой управляемый потенциометр, который создают на выходе аналоговый сигнал (напряжение или ток), отображающий нормированную часть его заданной полной шкалы. Выходное напряжение или ток зависят от значения опорного напряжения, выбираемого для задания полной шкалы выходного сигнала. Если опорное напряжение изменяется в соответствии с аналоговым сигналом, то выходной сигнал пропорционален произведению цифрового числа и аналогового входного сигнала. Полярность произведения зависит от полярности аналогового сигнала, цифровой системы кодирования и характера преобразования. Если ЦАП воспринимает опорные сигналы как положительной, так и отрицательной полярности и цифровой сигнал биполярный, то происходит четырехквадрантное умножение.

В АЦП цифровое число на выходе зависит от отношения преобразуемого входного сигнала к опорному сигналу, соответствующему полной шкале. Если опорный сигнал изменяется согласно изменению второго входного аналогового сигнала, то цифровой сигнал на выходе будет пропорционален отношению аналогового и опорного сигналов. Таким образом «измеритель отношений» АЦП может быть представлен как делитель аналоговых сигналов с цифровым выходом.

Для согласования работы ЦАП и АЦП необходимы разнообразные элементы как аналоговой, так и цифровой техники.

Благодаря широкому внедрению в современную электронную аппаратуру микропроцессоров, ЦАП, выпущенные в виде БИС, стали необходимыми элементами при проектировании блоков вычислительной техники, в робототехнике, в системах отображения информации, в системах цифровой связи, в измерительных приборах, системах синтеза аналоговых сигналов и т.д.

Преобразование аналоговой величины в цифровой код применяется довольно часто: в цифровых приборах с индикацией результатов измерения в привычном десятичном счис-

лении, для ввода в цифровой форме параметров технологического процесса в ЭВМ, которая не допускает их выхода за установленные пределы, при передаче информации по линии с целью повышения её помехозащищенности и т.д.

Обратное – цифроаналоговое – преобразование в ряде случаев сопровождает аналогово-цифровое. Кроме того, их сочетание позволяет осуществлять цифровую обработку аналоговой величины, предварительно преобразованной в цифровую форму, и последующее преобразование к исходному аналоговому виду.

Аналого-цифровые преобразователи (АЦП) являются устройствами, которые принимают входные аналоговые сигналы и генерируют соответствующие им цифровые сигналы, пригодные для обработки микропроцессорами и другими цифровыми устройствами.

Принципиально не исключена возможность непосредственного преобразования различных физических величин в цифровую форму, однако эту задачу удается решить лишь в редких случаях из-за сложности таких преобразователей. Поэтому в настоящее время наиболее рациональным признается способ преобразования различных по физической природе величин сначала в функционально связанные с ними электрические, а затем уже с помощью преобразователей «напряжение-код» – в цифровые. Именно эти преобразователи имеют обычно в виду, когда говорят об АЦП.

Цифро-аналоговый преобразователь (ЦАП) предназначен для преобразования числа, определенного, как правило, в виде двоичного кода, в напряжение или ток, пропорциональные значению цифрового кода. Схемотехника цифро-аналоговых преобразователей весьма разнообразна.

Схемы применения цифро-аналоговых преобразователей относятся не только к области преобразования код – аналог. Пользуясь их свойствами можно определять произведения двух или более сигналов, строить делители функций, аналоговые звенья, управляемые от микроконтроллеров, такие как аттенюаторы, интеграторы. Важной областью применения ЦАП являются также генераторы сигналов, в том числе сигналов произвольной формы.

Теоретические сведенья

1. Цифро-аналоговые преобразователи (ЦАП) служат для преобразования информации из цифровой формы в аналоговый сигнал – суммирование токов и напряжений. ЦАП широко применяется в различных устройствах автоматики для связи цифровых ЭВМ с аналоговыми элементами и системами.

Принцип работы ЦАП состоит в суммировании аналоговых сигналов, пропорциональных весам разрядов входного цифрового кода, с коэффициентами, равными нулю или единице в зависимости от значения соответствующего разряда кода.

ЦАП преобразует цифровой двоичный код $Q_4Q_3Q_2Q_1$ в аналоговую величину, обычно напряжение $U_{вых.}$. Каждый разряд двоичного кода имеет определенный вес i-го разряда вдвое больше, чем вес (i-1)-го. Работу ЦАП можно описать следующей формулой:

$U_{вых}$=e*(Q_1 1+Q_2*2+Q_3*4+Q_4*8+…), (1)

где e - напряжение, соответствующее весу младшего разряда, Q_i - значение i -го разряда двоичного кода (0 или 1).

Например, числу 1001 соответствует

$U_{вых}$=y*(1*1+0*2+0*4+1*8)=9*e, а числу 1100

$U_{вых}$=e*(0*1+0*2+1*4+1*8)=12*e.

На рисунке 1 приведена схема цифро - аналогового преобразователя.

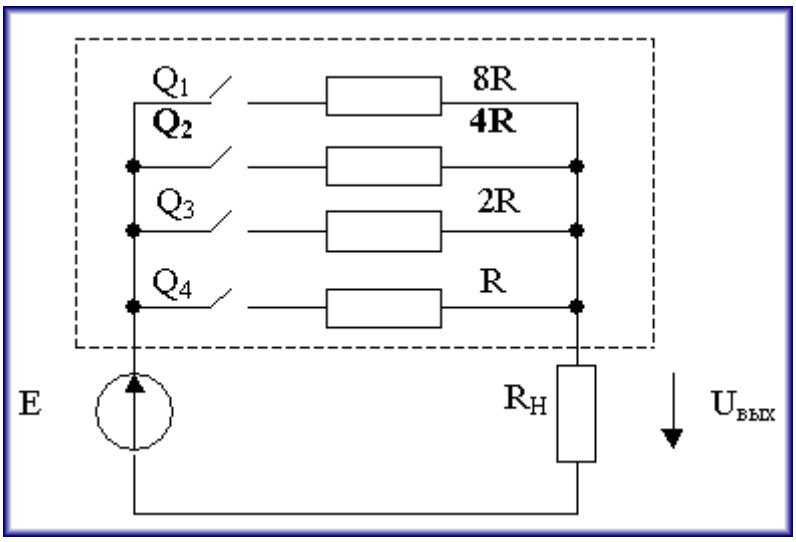


Рисунок 1 - Схема цифро-аналогового преобразователя

Упрощенная схема реализации ЦАП представлена на рис1. В схеме i – й ключ замкнут при Q_i=1, при Q_i=0 – разомкнут. Регистры подобраны таким образом, что R>>Rн.

Эквивалентное сопротивление обведенного пунктиром двухполюсника Rэк и сопротивление нагрузки Rн образуют делитель напряжения, тогда

Uвых = E Rн / Rэк + Rн » E*Rн / Rэк (2)

Проводимость двухполюсника 1 / Rэк равна сумме проводимостей ветвей (при Q_i=1 i – ветвь включена, при Q_i=0 – отключена):

1 / Rэк = Q1 / 8R + Q2 / 4R + Q3 / 2R + Q4 / R (3)

Подставив (3) в (2), получаем выражение, идентичное (1)

Uвых = (8E Rн / R)*(Q_1*1 + Q_2*2 + Q_3*4 + Q_4*8)

Очевидно, что e = 8E Rн / R. Выбором e можно установить требуемый масштаб аналоговой величины.

2. Аналогово-цифровые преобразователи. В информационных и управляющих системах часть (или вся) информация от датчиков бывает представлена в аналоговой форме. Для ее ввода в цифровые ЭВМ и цифровое управляющее устройство широко применяются аналогово-цифровые преобразователи (АЦП). В большинстве случаев АЦП выполняют преобразование входного напряжения или тока в двоичный цифровой код.

Существуют различные типы АЦП. Мы остановимся лишь на тех типах, которые получили в настоящее время наибольшее распространение.

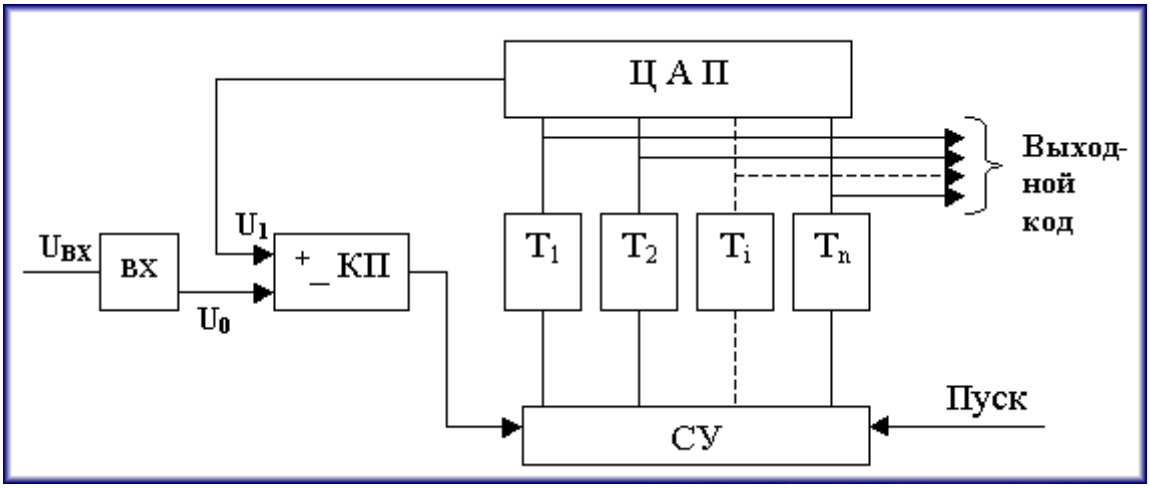


Рисунок 2 - Схема АЦП последовательного приближения

2.1. АЦП последовательного приближения (АЦППП). Структурная схема АЦППП приведена на рисунке 2. Схема работает следующим образом. Входной аналоговый сигнал Uвх перед началом преобразования запоминается схемой выборки – хранения ВХ, что необходимо, так как в процессе преобразования необходимо изменение аналогового сигнала. Далее по команде “Пуск” с помощью сдвигового регистра последовательно во времени каждый триггер Ti, начиная со старшего разряда, переводит в положение 1 соответствующий разряд ЦАП. Напряжение U1 (или ток) с выхода ЦАП сравнивается с входным аналоговым сигналом с помощью компаратора КП. Если U0 > U1, на выходе компаратора сохраняется низкий уровень и в триггере сохраняется единица, при U0 < U1 срабатывает компаратор и переводит триггер в положение 0. После окончания цикла на выходах триггеров получается двоичный код, соответствующий (при идеальных элементах) U0 с точностью до половины младшего разряда.

Погрешность АЦППП определяется неточностью ЦАП, зоной нечувствительности и смещением нуля компаратора, а также погрешностью схемы выборки – хранения.

Поскольку в такой схеме ошибка в каком – либо разряде в дальнейшем не корректируется, необходимо, чтобы время на “взвешивание” каждого разряда было достаточно для затухания переходного процесса до уровня, соответствующего половине младшего разряда, и чтобы при разбалансе U1 – U0 на это значение компаратора успел сработать. Общее время преобразования

$$t_{пр}=t_{вх}+n(t_{з,к}+t_{у}+t_{ц})+t_{сб},$$

где tвх – время, необходимое для фиксации Uвх схемой ВХ; n – число разрядов; tз,к – время задержки, вносимое компаратором; ty – время установления U1 на входе ЦАП; tц – время задержки цифровых элементов в схеме управления и срабатывания триггера; tсб – время, необходимое для сброса ЦАП в исходное состояние, включая время, необходимое для синхронизации с началом такта.

Наибольшую долю в tпр обычно вносит ty, наибольшая величина которого может быть оценена следующим образом:

$t_y=(1+n)T_э\ln 2$,

где Тэ – эквивалентная постоянная времени на входе ЦАП. Если на его выходе включен ОУ, который полезен для уменьшения выходного сопротивления и ускорения тем самым переходного процесса, то Тэ» 1/2p fcp (fcp – частота среза ОУ по контуру обратной связи).

При 12 – разрядном АЦП и использовании быстродействующего ЦАП с ty=100 нс время tпр близко к 1,5 мкс. В большинстве случаев tпр такого преобразователя достигает 10 – 100 мкс.

2.2. АЦП параллельного типа (АЦПП). Существенное уменьшение tпр удается получить в АЦП параллельного типа. Его структурная схема приведена на рис3. Здесь входная аналоговая величина U_0 с выхода схемы ВХ сравнивается с помощью 2^{n+1} – 1 компараторов с $2(2^n-1)$ эталонными уровнями, образованными делителями из резисторов равного сопротивления. При этом срабатывают m младших компараторов, образующих на выходах схем И-НЕ нормальный единичный код, затем который с помощью специального дешифратора ДШ преобразуется в двоичный выходной сигнал.

Погрешность АЦПП определяется неточностью и нестабильностью эталонного напряжения, резистивного делителя и погрешностями компараторов. Значительную роль могут играть входные токи компараторов, если делитель недостаточно низкоомный. На рисунке 3.3.4.3 приведена структурная схема АЦП параллельного типа.

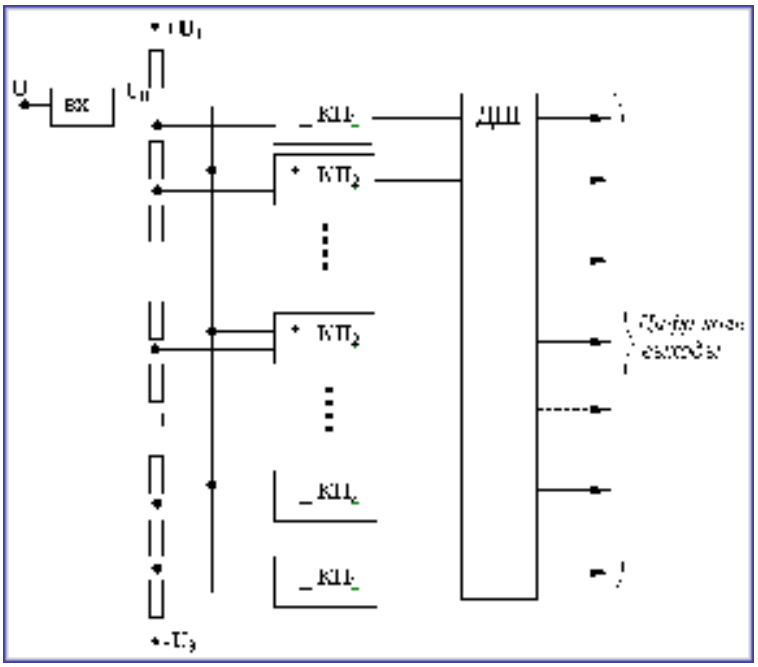


Рисунок 3.3.4.3 - Схема АЦП параллельного типа.

Время преобразования складывается из следующих составляющих:

$$\mathbf{t_{пр} = t_{вх} + t_{з,к} + a\ t_{л,сi}\ ,}$$

где $\mathbf{t_{л,сi}}$ – Время задержки логичесих схем; k – число последовательно включенных логических схем.

При использовании компаратров со стробированием АЦПП может быть без схемы ВХ. При этом он обеспечивает наибольшее быстродействие по сравнению с любыми другими АЦП.

Отчет

Отчет должен быть оформлен в текстовом редакторе и содержать:

- наименование работы;
- цель работы;
- задание;
- последовательность выполнения работы;
- ответы на контрольные вопросы;
- вывод о проделанной работе.

Комплект лабораторных работ

учебной дисциплины
«Охрана труда и техника и безопасности»

основной профессиональной образовательной программы
СПО на базе основного общего образования
при подготовке квалифицированных рабочих по профессии

по профессии:
09.01.03 "Мастер по обработке цифровой информации"

г. Балашиха

Введение

Настоящий сборник лабораторных работ предназначен в качестве методического пособия при проведении лабораторных работ по программе учебной дисциплины «Охрана труда и техника безопасности» при подготовке квалифицированных рабочих.

Требования к знаниям и умениям при выполнении лабораторных работ

В результате выполнения лабораторных работ, предусмотренных программой по данной учебной дисциплине, проводится текущий контроль индивидуальных образовательных достижений.

В результате освоения учебной дисциплины обучающийся

должен знать:

- правила техники безопасности и охраны труда при работе с электрооборудованием;
- нормативные документы по использованию средств вычислительной техники и видеотерминалов;
- виды и периодичность инструктажа по технике безопасности и охране труда (ТБиОТ)

должен уметь:

- выполнять санитарно-технологические требования на рабочем месте и в производственной зоне, нормы и требования к гигиене и охране труда.

Правила выполнения лабораторных работ

1. Обучающийся должен выполнить практическую работу в соответствии с полученным заданием.
2. Каждый обучающийся после выполнения работы должен представить отчет о проделанной работе с анализом полученных результатов и выводом по работе.
3. Отчет о проделанной работе следует выполнять в тетрадях для практических (лабораторных) работ.
4. Содержание отчета указано в описании лабораторных работ.
5. Таблицы и рисунки следует выполнять с помощью чертежных инструментов (линейки, циркуля) карандашом с соблюдением ЕСКД.
6. Расчет следует проводить с точностью до двух значащих цифр.
7. Если обучающийся не выполнил практическую работу или часть работы, то он может выполнить работу или оставшуюся часть во внеурочное время, согласованное с преподавателем.
8. Оценку по практической работе обучающийся получает, с учетом срока выполнения работы, если: - расчеты выполнены правильно и в полном объеме;
- сделан анализ проделанной работы и выводы по результатам работы;
- обучающийся может пояснить выполнение любого этапа работы;
- отчет выполнен в соответствии с требованиями к выполнению работы.

Зачет по лабораторным работам обучающийся получает при условии выполнения всех предусмотренных программой работ, после сдачи отчетов по работам при получении удовлетворительных оценок.

Пояснительная записка

Методические указания по выполнению лабораторных работ предназначены для организации работ по дисциплине " Охрана труда и техника безопасности "
и составлены в соответствии с Федеральным государственным стандартом СПО, рабочей программой дисциплины и предназначены для реализации требований к минимуму содержания и уровню подготовки выпускников по профессиям начального профессионального образования (НПО) и специальностям среднего специального образования (СПО).

Основной целью выполнения практических работ является отработка и закрепление навыков работы

- С инструментом и приспособлениями, используемыми на станках
- С методами контроля и измерительным инструментом
- С типовыми техпроцессами

А также формирование умений анализировать, обощать, выделять особенности и делать выводы, используя схемы, чертежи, текстовой материал.

Выполнение лабораторных работ направлено на:

- Развитие познавательной установки личности на создание целостных образов представлений о технологии выполнения ремонтных работ и объединяет теоретические знания с практическими умениями и навыками.
- Обобщение, систематизацию, углубление, закрепление полученныхзнаний, выработку способностей и готовность использовать теоретические знания на практике.
- Развитие интеллектуальных умений.
- Выработку при решении поставленных задач таких профессионально значимых компетенций как самостоятельность, точность, творческая инициатива.

Лабораторные работы позволяют реально применять свои теоретические знания для выполнения поставленных задач

Перечень лабораторных работ.

1.Лабораторная работа № 1
Аттестация и сертификация рабочих мест
2.Лабораторная работа №1
"Классификация, расследование, учет и оформление несчастных случаев.
3.Лабораторная работа №3
«Защита от теплового излучения»
4.**Лабораторная работа № 4**
Пожарная безопасность
5.**Лабораторная работа №5**
Изучение первичных средств тушения
6.Лабораторная работа №6
"Оформление проведения инструктажей. «Цвета сигнальные и знаки безопасности»
7.Лабораторная работа №7
«Первая помощь пострадавшим»
8.Лабораторная работа №8
Оказание первой помощи при переломе
9.Лабораторная работа № 9
Способы иммобилизации

Лабораторная работа № 1

Аттестация и сертификация рабочих мест

Цель работы: изучить методику проведения аттестации рабочих мест по условиям труда и разработать мероприятия по их улучшению.

Задание по работе: ознакомиться с требованиями нормативных документов по изучаемому вопросу, провести оценку условий труда на основе данных, приведенных в характеристике условий труда, по варианту и заполнить карту условий труда на рабочем месте, определить право работников на пенсию по возрасту за работу с особыми условиями труда, размер доплат за вредные условия труда, право на дополнительный отпуск за работу с вредными и (или) опасными условиями труда и его продолжительность, право на сокращенную продолжительность рабочего времени за работу с вредными и (или) опасными условиями труда, предложить мероприятия по улучшению условий труда на производстве.

Теоретические сведения.

Общие положения. В настоящее время работникам, занятым на работах с вредными и тяжелыми условиями труда, которые при современном техническом уровне производства и существующей организации труда не могут быть устранены, законодательством устанавливается ряд льгот и компенсаций.

Для совершенствования системы управления охраной труда на предприятии необходима аттестация рабочего места по условиям труда, предусматривающая оценку степени вредности и тяжести условий труда на конкретном рабочем месте. Аттестация рабочих мест является прогрессивной мерой на пути изменения системы предоставления компенсаций, приведение их в соответствие с фактическим состоянием условий труда.

Аттестация рабочих мест по условиям труда – это система учета, анализа и комплексной оценки на конкретном рабочем месте всех факторов производственной среды и трудового процесса, воздействующих на здоровье и трудоспособность человека в процессе трудовой деятельности.

В настоящее время порядок проведения аттестации рабочих мест по условиям труда определен постановлением Совета Министров Республики Беларусь от 22.02.2008 № 253 «Об аттестации рабочих мест по условиям труда» (в ред. постановления Совмина от 12.01.2009 N 25), а оценка условий труда при аттестации – постановлением Министерства труда и социальной защиты Республики Беларусь от 22.02.2008 № 35 «Об утверждении Инструкции по оценке условий труда при аттестации рабочих мест по условиям труда и предоставлению компенсаций по ее результатам и признании утратившими силу некоторых постановлений Министерства труда Республики Беларусь, Министерства труда и социальной защиты Республики Беларусь» (в ред. постановления Минтруда и соцзащиты от 13.01.2009 N 7), а также Санитарными нормами, правилами и гигиеническими нормативами 13-2-2007 «Гигиеническая классификация условий труда», утвержденными постановлением Министерства здравоохранения Республики Беларусь от 20.12.2007 № 176.

Процедура аттестация предусматривает оценку условий труда на конкретном рабочем месте, а именно определение неблагоприятных факторов
и показателей производственной среды, показателей тяжести и напряженности
трудового процесса.

Целью проведения аттестации рабочих мест по условиям труда (далее - аттестация) является оценка условий труда на конкретном рабочем месте для разработки и реализации плана мероприятий по улучшению условий труда, определения обязанностей нанимателя по профессиональному пенсионному страхованию работников, а также определения права работника:

- на пенсию по возрасту за работу с особыми условиями труда;
- на дополнительный отпуск;
- на сокращенную продолжительность рабочего времени;
- на оплату труда в повышенном размере путем установления доплат.

Пенсия по возрасту за работу с особыми условиями труда, дополнительный отпуск за работу с вредными и (или) опасными условиями труда, сокращенная продолжительность рабочего времени за работу с вредными и (или) опасными условиями труда, оплата труда в повышенном размере путем установления доплат за работу с вредными и (или) опасными условиями труда по результатам аттестации предоставляются работникам, занятым на работах с вредными и (или) опасными условиями труда в течение полного рабочего дня.

Под полным рабочим днем понимается выполнение работы с вредными и (или) опасными условиями труда не менее 80 процентов от продолжительности ежедневной работы (смены), установленной законодательством. В продолжительность ежедневной работы при этом включается подготовительно-заключительное время, оперативное время (основное и вспомогательное) и время обслуживания рабочего места в пределах установленных нормативов времени, а также время регламентированных перерывов, включаемых в рабочее время.

Работу по аттестации необходимо начать с издания приказа о проведении аттестации, в котором утверждается состав аттестационной комиссии, ее полномочия, назначается председатель аттестационной комиссии, а также лицо ответственное за ведение и хранение документации по аттестации, устанавливается сроки и график проведения работ по аттестации. При необходимости следует создать аттестационные комиссии в структурных подразделениях.

В состав аттестационной комиссии следует включать работников служб охраны труда, кадровой, юридической, организации труда и заработной платы, промышленно-санитарной лаборатории, руководителей структурных подразделений организации, медицинских работников, представителей профсоюза (профсоюзов). Председателем аттестационной комиссии назначается, как правило, заместитель руководителя либо главный инженер.

В функции аттестационной комиссии входит:

- организационное, методическое руководство и контроль над ходом аттестации;
- формирование необходимой для проведения аттестации нормативно правовой базы и организация ее изучения;
- определение перечня рабочих мест, подлежащих аттестации;
- установление соответствия наименования профессий рабочих и должностей служащих требованиям законодательства. Подготовка предложения о внесении изменений в штатное расписание, трудовые книжки работников и другие документы (при необходимости);
- определение исполнителей для измерения и исследования уровней вредных и опасных факторов производственной среды, для оценки условий труда по показателям тяжести и напряженности трудового процесса;

- проведение обследования рабочих мест в целях проверки на соответствие производственного оборудования и технологических процессов требованиям охраны труда и принятие мер по устранению выявленных недостатков;
- организация проведения фотографии рабочего времени, и оформление карты фотографии рабочего времени; составление карты аттестации рабочего места по условиям труда
- ознакомление работников с результатами аттестации.

При определении перечня рабочих мест подлежащих аттестации аттестационная комиссия руководствуется Списками производств, работ, профессий, должностей и показателей, дающих право на пенсию по возрасту за работу с особыми условиями труда (Список №1, Список №2), утвержденными Постановлением Совета Министров Республики Беларусь от 25 мая 2005г. №536, с последующими изменениями и дополнениями, а также Списком производств, цехов, профессий и должностей с вредными и (или) опасными условиями труда, работа в которых дает право на сокращенную продолжительность рабочего времени, утвержденным Постановлением Министерства труда и социальной защиты Республики Беларусь 10 декабря 2007 г. №170.

Аттестации подлежат рабочие места, на которых работа с вредными и (или) опасными условиями труда занимает не менее 80 процентов от продолжительности ежедневной работы (смены).

Аттестационная комиссия должна включить в перечень также те рабочие места, на которых работникам ранее устанавливались дополнительный отпуск
и сокращенная продолжительность рабочего времени. При этом могут учитываться предложения специалистов и работников организации. Такие предложения должны быть проанализированы аттестационной комиссией и при наличии оснований включены в перечень рабочих мест подлежащих аттестации.

Для определения уровней воздействия опасных и вредных производственных факторов (содержания вредных веществ в воздухе, уровней шума и вибрации и т. д.) привлекаются организации, имеющие государственную лицензию, дающую право проводить инструментальные замеры. Для получения лицензии организация должна иметь соответствующее оборудование и специалистов, владеющих методикой проведения измерений. Инструментальные замеры и лабораторные исследования проводятся в присутствии членов аттестационной комиссии в соответствии с действующими нормативно-методическими документами. Применяемые для измерения приборы и устройства должны соответствовать метрологическим требованиям. Они подвергаются проверке в сроки, установленные Государственным комитетом по стандартизации, метрологии и сертификации.

Оценка вредных и опасных факторов производственной среды проводится с учетом продолжительности воздействия этих факторов в течение смены. Для постоянных рабочих мест, если работа осуществляется в одном пункте рабочей зоны, продолжительность времени воздействия фактора производственной среды определяется на основании фотографий рабочего дня.

Аттестация проводится не реже чем один раз в пять лет. Внеочередная аттестация проводится:
- при изменении условий труда в связи с реконструкцией производственного объекта, внедрением новой техники и технологии, применением новых видов сырья и материалов;

- по требованию органов Государственной экспертизы условий труда. Ответственность за своевременное и качественное проведение аттестации возлагается на нанимателя.

Контроль качества проведения аттестации возложен на Государственную экспертизу условий труда.

Оценка фактического состояния условий труда на рабочем месте при аттестации производится на основании гигиенической классификации условий труда, утвержденной Министерством здравоохранения.

Принципы классификации условий труда

Условия труда, исходя из гигиенических критериев, подразделяются на 4 класса:

оптимальные условия труда (1 класс) – такие условия, при которых сохраняется здоровье работников и создаются предпосылки для поддержания высокого уровня работоспособности. Оптимальные нормативы установлены для микроклиматических параметров и факторов трудового процесса. Для других факторов условно за оптимальные принимаются такие условия труда, при которых факторы условий труда отсутствуют либо не превышают уровни, принятые в качестве безопасных для населения;

допустимые условия труда (2 класс) характеризуются такими уровнями факторов среды и трудового процесса, которые не превышают установленных гигиенических нормативов для рабочих мест, а возможные изменения функционального состояния организма, возникающие под их воздействием, восстанавливаются во время регламентированного отдыха или к началу следующей смены и не оказывают неблагоприятного действия в ближайшем и отдаленном периоде на состояние здоровья работников и их потомство;

вредные условия труда (3 класс) характеризуются наличием вредных производственных факторов, выходящих за пределы гигиенических нормативов и оказывающих неблагоприятное действие на организм работника и/или его потомство;

опасные условия труда (4 класс) характеризуются уровнями производственных факторов, воздействие которых в течение рабочей смены (или ее части) может создать угрозу для жизни, высокий риск развития острых профессиональных поражений, в том числе и тяжелых форм. При этом работа должна проводиться в соответствующих средствах индивидуальной защиты и при строгом соблюдении режимов, регламентированных для такого вида работ и обеспечивающих безопасность для здоровья работников.

Вредные условия труда по степени отклонения параметров факторов от гигиенических нормативов и выраженности изменений в организме работников подразделяются на 4 степени вредности:

1 степень 3 класса (3.1) – условия труда характеризуются такими отклонениями уровней вредных факторов от гигиенических нормативов, которые вызывают функциональные изменения, восстанавливающиеся, как правило, при более длительном (чем к началу следующей смены) прерывании контакта с вредными факторами и увеличивают риск повреждения здоровья;

2 степень 3 класса (3.2) – уровни вредных факторов, вызывающие стойкие функциональные изменения, приводящие в большинстве случаев к

увеличению производственно обусловленной заболеваемости (что проявляется повышением уровня заболеваемости с временной утратой трудоспособности и,
в первую очередь, теми болезнями, которые отражают состояние наиболее уязвимых органов и систем для данных вредных факторов), появлению начальных признаков или легких (без потери профессиональной трудоспособности) форм профессиональных заболеваний, возникающих после продолжительной экспозиции (часто после 15 и более лет);

3 степень 3 класса (3.3) – условия труда, характеризующиеся такими уровнями вредных факторов, воздействие которых приводит к развитию, как правило, профессиональных болезней легкой и средней степеней тяжести (с потерей профессиональной трудоспособности) в периоде трудовой деятельности, росту хронической (производственно обусловленной) патологии, включая повышенные уровни заболеваемости с временной утратой трудоспособности;

4 степень 3 класса (3.4) – условия труда, при которых могут возникать тяжелые формы профессиональных заболеваний (с потерей общей трудоспособности), отмечается значительный рост числа хронических заболеваний и высокие уровни заболеваемости с временной утратой трудоспособности.

По результатам аттестации устанавливается класс (степень) вредности или опасности условий труда на рабочем месте. В зависимости от гигиенической оценки рабочего места определяется перечень льгот и компенсаций за работу с особыми условиями туда.

Компенсации, предоставляемые по результатам аттестации

При оценке условий труда, соответствующих 3-му классу третьей степени вредности (3.3) и выше, подтверждаются особые условия труда на рабочих местах работников, профессии, должности, показатели работ которых предусмотрены списком производств, работ, профессий, должностей и показателей на подземных работах, на работах с особо вредными и особо тяжелыми условиями труда, занятость в которых дает право на пенсию по возрасту за работу с особыми условиями труда (Список N 1).

При оценке условий труда, соответствующих 3-му классу второй степени вредности (3.2), для указанных работников подтверждаются условия труда, соответствующие требованиям списка производств, работ, профессий, должностей и показателей на работах с вредными и тяжелыми условиями труда, занятость в которых дает право на пенсию по возрасту за работу с особыми условиями труда (Список N 2).

При оценке условий труда, соответствующих 3, 4 классам, на рабочих местах работников в зависимости от класса и степени вредности условий труда устанавливаются доплаты за работу с вредными и (или) опасными условиями труда (таблица 14).

Доплаты за работу с вредными и(или) опасными условиями труда к тарифным ставкам и должностным окладам работников устанавливаются в процентах от тарифной ставки первого разряда, установленной в организации. Наниматель не имеет права устанавливать размер доплат ниже минимально гарантированных.

Если в организации применяется несколько тарифных ставок первого разряда, доплата исчисляется исходя из тарифной ставки первого разряда, принятой для начисления заработной платы работнику.

Продолжительность дополнительного отпуска за работу с вредными и (или) опасными условиями труда устанавливается в зависимости от класса (степени) вредности или опасности условий труда согласно постановлению Совета Министров Республики Беларусь от 19 января 2008 г. № 73 .

При оценке условий труда, соответствующих 3, 4 классам подтверждается право на сокращенную продолжительность рабочего времени на рабочих местах работников, профессии, должности которых предусмотреныСписком производств, цехов, профессий и должностей с вредными и (или) опасными условиями труда, работа в которых дает право на сокращенную продолжительность рабочего времени.

Порядок выполнения работы

При выполнении работы для заполнения Карты условий труда на рабочем месте (таблица 1) необходимо:

1. В соответствии с вариантом задания (таблица 2) занести в графу 3 характерные для конкретного рабочего места санитарно-гигиенические условия факторов производственной среды.
2. Гигиенические нормативы факторов производственной среды (графа 2) определить согласно техническим нормативным правовым актам:

– химический фактор в соответствии с таблицей 3;
– пыли и аэрозоли (таблица 4);
– шум – по таблице 5 определяется допустимый уровень звука дБА;
– вибрация – определяется нормативный уровень виброскорости дБ (таблица 6) в соответствии с среднегеометрической частотой (по варианту задания);
– микроклимат – в зависимости от категории работы по тяжести определить допустимые значения температуры, относительной влажности, скорости движения воздуха (таблица 7);
– освещенность определяется в зависимости от разряда зрительной работы, характеристики фона и контраста объекта с фоном по таблице 8.

3. Оценка факторов производственной среды проводится путем сопоставления полученных в результате измерений и исследований их фактических величин с гигиеническими нормативами. В зависимости от отклонения каждого фактора производственной среды от нормативного значения в соответствии с таблицами 9…13 устанавливается класс условий труда для каждого фактора производственной среды. Результаты оценки (класс и степень условий труда) без учета времени воздействия фактора производственной среды заносятся в графу 4.
4. Время воздействия фактора (графа 5). Среднее время воздействия факторов производственной среды при выполнении работы в различных рабочих зонах рассчитывается на основании нескольких фотографий рабочего времени. Время воздействия для каждого фактора определяется в соответствии с вариантом задания.
5. Оценка с учетом времени воздействия фактора производственной среды (графа 6). Если влияние вредного и (или) опасного фактора производственной среды на работника составляет менее 50 и до 10 процентов (включительно) от продолжительности рабочего времени, класс условий труда по данному фактору снижается на одну степень; при продолжительности воздействия фактора производственной среды на работника менее 10 процентов от продолжительности рабочего времени производится снижение класса условий труда на две степени.

6. Итоговая оценка фактора по микроклиматическим условиям производится по

Факторы и показатели производственной среды	Гигиенические нормативы (ПДК, ПДУ)	Фактические величины	Класс (степень) условий	Время воздействия фактора	Класс (степень) условий

наиболее высокому классу и степени вредности. Например, если два фактора оценены классом 3.1 и один классом 3.2, то итоговая оценка условий труда будет оценена классом 3.2.;

7. Общая оценка условий труда по классу (степени) проводится на основании оценок по всем факторам производственной среды, тяжести и напряженности трудового процесса. Общая оценка условий труда на рабочем месте устанавливается по наиболее высокому классу и степени вредности.

При наличии 3-х и более факторов производственной среды, тяжести и напряженности трудового процесса, относящихся к классу 3.1, общая оценка условий труда соответствует классу 3.2.

При наличии 2-х и более факторов производственной среды, тяжести и напряженности трудового процесса, относящихся к классам 3.2, 3.3 и 3.4 – условия труда оцениваются соответственно на одну степень выше.

8. В зависимости от общей оценки условий труда определить:

– право работников на пенсию по возрасту за работу с особыми условиями труда (Список N 1) или (Список N 2);

При оценке условий труда, соответствующих 3-му классу третьей степени вредности (3.3) и выше, подтверждаются особые условия труда на рабочих местах работников с особо вредными и особо тяжелыми условиями труда, занятость в которых дает право на пенсию по возрасту за работу с особыми условиями труда (Список N 1).

При оценке условий труда, соответствующих 3-му классу второй степени вредности (3.2), подтверждаются особые условия труда на рабочих местах работников, профессии, должности, показатели работ которых предусмотрены Списком № 2.

– минимально гарантированный размер доплат за каждый час работы во вредных и тяжелых условиях труда (таблица 14). С учетом действующей в Республике Беларусь тарифной ставке 1-го разряда определить величину доплат при условии, что месячный фонд рабочего времени равен 176 часов;

– право на сокращенную продолжительность рабочего времени;

– право на дополнительный отпуск за работу с вредными и (или) опасными условиями (таблица 15);

– с учетом влияния факторов на итоговую оценку условий труда разработать комплекс мероприятий по улучшению условий труда.

Таблица 1 – Карта условий труда на рабочем месте
(Извлечение из «Методики проведения аттестации рабочих мест по условиям труда»)

			труда		труда с учетом времени воздействия фактора
1.Химический фактор, мг/м^3:					
2. Пыли и аэрозоли, мг/м^3					

3. Шум, дБА					
4. Вибрация локальная дБ:					
5. Микроклимат:					
5.1. температура воздуха, ^{0}С					
5.2. относительная влажность воздуха, %					
5.3. скорость движения воздуха, м/с					
Итоговая оценка фактора					
5. Освещенность, лк					

Результаты аттестации рабочего места

1. Общая оценка условий труда_____

(указать класс условий труда)

2. Право на следующие виды компенсаций:

_____

_____

(указать: Список № 1, Список № 2, право на пенсию не подтверждено результатами аттестации)

дополнительный отпуск за работу с вредными и (или) опасными условиями труда

_____

(указать количество календарных дней)

сокращенная продолжительность рабочего времени за работу с вредными и (или) опасными условиями труда

_____

(указать количество часов)

доплата за работу с вредными и (или) опасными условиями труда

_____

(указать процент доплат)

Лабораторная работа №2

ТЕМА: «Классификация, расследование, учет и оформление несчастных случаев»

ЦЕЛЬ ЗАНЯТИЯ: Ознакомиться с порядком расследования несчастных случаев на производстве. Изучить порядок оформления и учета несчастных случаев.

ПРИОБРЕТАЕМЫЕ УМЕНИЯ И НАВЫКИ: Научиться правилам организации расследования и оформления несчастных случаев на производстве.

НОРМА ВРЕМЕНИ: 2 часа

ОСНАЩЕНИЕ РАБОЧЕГО МЕСТА: инструкционные карты - задания, акт по форме Н-1.

ЗАДАНИЯ:

Задание № 1. Изучить положение о порядке расследования и учета несчастных случаев на производстве. Описать несчастные случаи, которые подлежат расследованию и учету.

Задание № 2. Ознакомиться с обязанностями работодателя у которого произошёл несчастный случай, порядком расследования несчастного случая.

Задание № 3. Изучить порядок заполнения акта по несчастным случаям на производстве по форме Н-1. Заполнить акт по форме Н-1 на примере.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ:

Производственная травма - это травма, полученная работающим на производстве, или вызвана не соблюдением ТБ, или внезапно возникшей аварийно-стрессовой ситуацией.

Несчастный случай - это случай с работающим, связанный с воздействием на него опасного производственного фактора.

В соответствии с положением о порядке расследования и учета несчастных случаев на производстве расследованию и учету подлежат несчастные случаи (травма, в том числе полученная в результате нанесения телесных повреждений другим лицом, острое отравление, тепловой удар, ожог, обморожение, утопление, поражение электрическим током, молнией и ионизирующем излучением, укусы насекомых и пресмыкающихся, телесные повреждения, нанесенные животными, повреждения, полученные в результате взрывов, аварий, разрушения зданий, сооружений и конструкций, стихийных бедствий и других чрезвычайных ситуаций), повлекшие за собой необходимость перевода работника на другую работу, временную или стойкую утрату им трудоспособности либо, его смерть и происшедшее при выполнении работником своих трудовых обязанностей (работ) на территории организации или вне ее, а также при следовании к месту работы или с работы на предоставленном работодателем транспорте, либо на личном транспорте при соответствующем договоре или распоряжении работодателя о его использовании в производственных целях; при следовании к месту командировке и обратно; при привлечении работника в установленном порядке к участию в ликвидации последствий катастрофы, аварий и других чрезвычайных происшествий природного и техногенного характера; при осуществлении не входящих в трудовые обязанности работника действий, но совершаемых в интересах работодателя или направленных на предотвращение аварии или несчастного случая и в некоторых других случаях. Действие Положения распространяется на:

1. работников, выполняющих работу по трудовому договору (контракту);
2. граждан, выполняющих работу по гражданско-правовому договору;
1. студентов образовательных учреждений высшего и среднего профессионального образования, студентов и учащихся образовательных учреждений высшего, среднего и начального профессионального образования и образовательных учреждений основного общего образования, проходящих

производственную практику в организациях; лиц, осужденных к лишению свободы и привлекаемых к труду администрацией организации;

2. других лиц, участвующих в производственной деятельности организации или индивидуального предпринимателя.

Работодатель или лицо, им уполномоченное (далее именуется работодатель), обязан:

1. Обеспечить незамедлительное оказание пострадавшему первой помощи, а при необходимости доставку его в учреждение скорой медицинской помощи или другое иное лечебно-профилактическое учреждение;

1. Организовать формирование комиссии по расследованию несчастного случая:

1. Обеспечить сохранение до начала расследования обстоятельств и причин несчастного случая обстановки на рабочем месте и оборудования такими, какими они были на момент происшествия (если это не угрожает жизни и здоровью работников и не приведет к аварии);

2. Сообщать в течении суток по форме, установленной Министерством труда РФ, о каждом групповом несчастном случае (два и более пострадавших), несчастном случае с возможным инвалидном исходом и несчастном случае со смертельным исходом:

1. государственную инспекцию труда по субъекту РФ;
2. прокуратуру по месту, где произошел несчастный случай;
3. орган исполнительной власти субъекта РФ;
4. соответствующий федеральный орган исполнительной власти;

1. орган государственного надзора, если несчастный случай произошёл в организации (на объекте), подконтрольной этому органу;

1. организацию, направившую работника, с которым произошёл несчастный случай; - соответствующий профсоюзный орган.

Расследование несчастных случаев проводится комиссией, образуемой из представителей работодателя, а также профсоюзного органа или иного уполномоченного работниками представительного органа. Состав комиссии утверждается приказом. Руководитель, непосредственно отвечающий за безопасность производства, в расследовании не участвует.

По требованию пострадавшего (а при его смерти его родственников) в расследовании несчастного случая может принимать участие его доверенное лицо.

Несчастные случаи, происшедшие с работниками, направленными сторонними организациями, в том числе со студентами и учащимися, проходящими производственную практику, расследуются с участием представителя направившей их организацией.

Комиссия по расследованию несчастного случая обязана в течении трех суток с момента происшествия расследовать обстоятельства и причины, при которых произошел несчастный случай; при случаях, вызвавших потерю у работника трудоспособности на период не менее одного календарного дня или необходимость перевода его на тот же срок с работы по основной профессии на другую работу (согласно медицинскому заключению), или его смерть, составить акт по форме Н-1

в двух экземплярах (если несчастный случай произошел с работником другой организации, то акт составляют в трех экземплярах), разработать мероприятия по предупреждению несчастных случаев и направить их работодателю для утверждения. Подписанный и утвержденный акт заверяют печатью организации.

Руководитель предприятия (главный инженер) обязан немедленно принять меры к устранению причин, вызвавших несчастный случай. После окончания расследования в течении трех суток один экземпляр утвержденного акта по форме Н-1 должен быть передан пострадавшему (или его представителю).

Несчастный случай, о котором пострадавший не сообщил администрации предприятия, цеха в течении рабочей смены или от которого потеря трудоспособности наступила не сразу, должен быть расследован по заявлению пострадавшего или заинтересованного лица в срок не более месяца со дня подачи заявления. Вопрос о составлении акта по форме Н-1 решается после всесторонней проверки заявления о происшедшем несчастном случае с учетом всех обстоятельств, медицинского заключения о характере травмы и возможной причины потери трудоспособности, показаний очевидцев и других доказательств.

Специальному расследованию несчастных случаев на производстве подлежат; групповой несчастный случай, несчастный случай с возможным инвалидным исходом, несчастный случай со смертельным исходом. Расследование производится комиссией в составе государственного инспектора труда органа исполнительной власти соответствующего субъектам РФ, представителей работодателя, профсоюзного или иного уполномоченного работниками представительного органа в течение 15 дней. Акт Н-1 с материалами расследования хранится 45 лет. Опросы очевидцев и лиц, допустивших нарушения нормативных требований по охране труда, оформляются в производной форме и подписываются опрашиваемыми. При групповом несчастном случае акт Н-1 составляется на каждого пострадавшего отдельно. Каждый акт по форме Н-1 регистрируется в журнале регистрации несчастных случаев.

Порядок заполнения акта несчастного случая на производстве по форме Н-1.

Акт по форме Н-1 заполняется текстовой и цифровой информацией, которая должна записываться и кодироваться в соответствии с общепринятыми терминами и специально разработанным классификатором. Кодирование проводит организация, где произошел несчастный случай.

В пункте 1 в первой строке указывается дата и время прошедшего несчастного случая. Число месяца кодируется двумя цифрами, месяц - его порядковым номером в году, год -последними двумя цифрами. *В третей строке* пункта следует указать и кодировать через сколько полных часов от начала работы с пострадавшим произошел несчастный случай.

Во пункте 2 в первой строке указывается наименование организации, где произошел несчастный случай. Наименование организации кодируется классификатором отраслей народного хозяйства. Наименование цеха организации, где произошел несчастный случай должно проводиться в соответствии с утвержденным перечнем структурных подразделений организации.

Пункте 3 заполняется текстовой информацией и не кодируется.

В пункте 4 указывается наименование адрес организации, направивший работника. Организация кодируется по классификаторам народного хозяйства.

В пункте 5 в первой строке полностью записывается Ф.И.О. пострадавшего. Пол кодируется цифрой (1-мужчина; 2-женщина); в третей строке указывается и кодируется возраст (числом полных лет, исполнившихся пострадавшему на момент происшедшего с ним несчастного случая').

В четвёртой строке профессия кодируется по общероссийскому классификатору профессий рабочих, должностей служащих и тарифных разрядов. Если у пострадавшего несколько профессий, то указывается та, при работе на которой произошёл несчастный случай.

В 5 строке указывается и кодируется стаж работы (числом полных лет работы, при выполнении которой произошёл несчастный случай), (меньше года -00).

Пункт 6-ой заполняется в соответствии с ГОСТом и не кодируется.

В пункте 7 при описании обстоятельств несчастного случая следует:

1. дать краткую характеристику условий труда и действий пострадавшего;
2. изложить последовательность событий, предшествующих несчастному случаю;
3. описать как протекал процесс труда;
4. указать, кто руководил работой, организовывал её, обеспечен ли был пострадавший средствами индивидуальной защиты и применял их или нет.

Во 2-ой строке указывается и кодируется вид происшествия в соответствии с классификатором.

В третьей строке указывается и кодируются причины несчастного случая.

В 4-ой строке в текстовой части приводится полное наименование оборудования, использование которого привело к несчастному случаю и который кодируется по классификатору оборудование, машины, механизмы, являющиеся источником травмы.

В 5-й строке указывается и кодируется возможное нахождение пострадавшего в состоянии опьянения.

Например - алкогольное опьянение кодируется цифрой -20, наркотическое-21.

В пункте 8 указываются лица, допустившие нарушение государственных нормативныхтребований по охране труда, действие или бездействие которых стали причиной несчастного случая. Организация, работниками которых допущены нарушения кодируется по общероссийскому классификатору предприятий и организаций. Если количество организаций, работниками которых допущены нарушения, две и более, то они в акт вносятся текстом и не кодируются. В случае, если нарушение допустило конкретное лицо, то оно указывается только в текстовой части акта.

Пункте 9 заполняется текстовой информацией и не кодируется.

В пункте 10 указывается каждое мероприятие по устранению причин несчастного случая отдельно. Не следует вносить в данный пункт наложенные взыскания на лиц, допустивших нарушения государственных нормативных требований по охране труда. Не кодируется.

КОНТРОЛЬНЫЕ ВОПРОСЫ:

1. Как оформляют несчастные случаи?
2. Комиссия в каком составе может расследовать несчастный случай?
3. Сколько хранится акт по форме Н-1?

Лабораторная работа №3

Тема: «Защита от теплового излучения»

Цель работы: изучение методов измерения количества тепла от излучающих поверхностей и оценка эффективности и условий применения защитных средств.

Задание для самостоятельной работы: изучить рекомендованную литературу и методические указания по выполнению данной работы. Подготовить бланк отчета, в котором указать параметры, входящие в понятие «инфракрасное излучение», перечислить методы снижения опасности перегрева, подготовить таблицы протоколов экспериментов.

Лабораторная работа проводится в течение 2 академических часов.

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1. ФИЗИЧЕСКАЯ ХАРАКТЕРИСТИКА ТЕПЛОВОГО ИЗЛУЧЕНИЯ

Тепловое излучение (инфракрасное излучение (ИКИ)) представляет собой невидимое электромагнитное излучение с длиной волны от 0.76 до 420 мкм, обладающее волновыми и световыми свойствами.

По длине волны инфракрасные лучи делятся на коротковолновую ИКИ-А (менее 1.4 мкм), средневолновую ИКИ-В (1.4-3 мкм), длинноволновую ИКИ-С (3 мкм-1 мм) область.

В производственных условиях гигиеническое значение имеет более узкий диапазон 0.76-70 мкм.

Источником инфракрасного излучения в производственных условиях являются нагретые поверхности слитков, чушек, листов, поковок, разливаемый жидкий металл, открытое пламя печей, сварочное пламя (при электро и газосварке) и т.п.

По характеру излучения производственные источники тепла и лучистой энергии подразделяются на четыре основные группы:

1) Источники с температурой до 500°С - спектр содержит исключительно длинноволновое ИКИ;

2) Источники с температурой от 500°С до 1200°С - в спектре содержится ИКИ-А, ИКИ-В, ИКИ-С, но появляется также видимое излучение слабой интенсивности, сначала красное, а затем белое;

3) Источники с температурой от 1200°С до 2000°С - спектр содержит как все виды ИКИ, так и видимое излучение высокой яркости;

4) Источники с температурой от 2000°С до 4000°С - спектр наряду с инфракрасным и видимым излучением содержит ультрафиолетовое излучение.

Основные законы физики инфракрасного излучения следующие:

Закон Кирхгофа: лучеиспускание обуславливается только состоянием излучающего тела и не зависит от окружающей среды. Лучеиспускательная способность любого тела пропорциональна его лучепоглощающей способности. Тело, поглощающее все падающие на него лучи (абсолютно черное тело), обладает максимальным излучением. На этом законе основано применение поглощающей защитной одежды, светофильтров, окраска оборудования, устройство приборов для измерения теплового излучения.

Закон Стефана-Больцмана: с повышением температуры излучающего тела мощность излучения увеличивается пропорционально 4-й степени его абсолютной температуры:

$$\mathbf{E = \sigma \cdot T^4\ [Вт/м^2]}\ (1)$$

E - мощность излучения; **σ** - постоянная Стефана-Больцмана, равная $5.67032 \cdot 10^{-8}$

Вт·м$^{-2}$·К$^{-4}$; **Т** - абсолютная температура, К (Кельвин).

В соответствии с этим законом даже небольшое повышение температуры тела приводит к значительному росту отдачи тепла излучением. Используя этот закон можно определить величину теплообмена излучением в производственных условиях.

Количество тепловой энергии, передаваемое излучением, определяется законом Стефана-Больцмана по формуле:

$$\mathbf{E = C_1C_2 \cdot \sigma\ (T_1^4 - T_2^4)}\ (2)$$

E - теплоотдача, Вт/м , $\mathbf{C_1}$ и $\mathbf{C_2}$ - константы излучения с поверхностей, **σ** - постоянная Стефана-Больцмана; $\mathbf{T_1}$ и $\mathbf{T_2}$ - температуры поверхностей (°К), между которыми происходит теплообмен излучением.

При расчете теплоотдачи излучением учитывают температуру стен и других поглощающихтепловую радиацию поверхностей (среднерадиационная температура).

Закон Вина: произведение абсолютной температуры излучающего тела на длину волны излучения (**λ** макс) с максимальной энергией - величина постоянная

$$\lambda_{макс} \cdot T = C \quad (3)$$

где: **C=2880**; **T** - абсолютная температура °К; **λ** - длина волны в мкм.

Исходя из закона Вина, длина волны максимального излучения нагретого тела обратно пропорциональна его абсолютной температуре:

$$\lambda_{макс} = C / T \quad (4)$$

Интенсивность теплового излучения на рабочих местах может колебаться от 175 Вт/м^2 до 13956 Вт/м^2. К горячим цехам относят цеха, в которых тепловыделение превышает 23 Дж/м^2.

В литейных цехах (нагрев и обработка деталей) интенсивность излучения составляет 1392-3480 Вт/м^2.

В производственных помещениях с большим тепловыделением (горячие цеха) доля тепла, приходящее на инфракрасное излучение, может составлять до 2/3 выделяемого тепла и только 1/3 составляет конвекционное тепло, т.е. тепло, передающееся при контакте с нагретым воздухом.

Основная физическая характеристика инфракрасного излучения - интенсивность излучения (плотность потока) **E** (Вт/м^2) зависит от температуры излучателя, его площади и расстояния до исследуемой точки пространства и определяется по следующим формулам:

При $R \geq \sqrt{S}$

$$E_u = \quad (5)$$

При $R < \sqrt{S}$

$$E_u = \quad (6)$$

где **S** - площадь поверхности излучателя, м^2,

T_u - абсолютная температура излучателя, °К,

R - расстояние от излучателя до точки замера, м

1. *1.2. БИОЛОГИЧЕСКОЕ ДЕЙСТВИЕ ТЕПЛОВОГО ИЗЛУЧЕНИЯ*

Лучистое тепло имеет ряд особенностей. ИКИ, помимо усиления теплового воздействия среды на организм работающего, обладает и специфическим влиянием, зависящим от интенсивности энергии излучения отдельных участков его спектра. Существенное влияние на лучистый теплообмен организма оказывают оптические свойства кожного покрова с его избирательной характеристикой коэффициентов отражения, поглощения и пропускания по отношению к различным участкам спектра инфракрасной радиации.

Воздействие ИКИ на организм человека проявляется как общими, так и местными реакциями. Местная реакция - выражается сильнее при длинноволновом облучении, поэтому при одной и той же интенсивности облучения время переносимости при длинноволновом облучении короче, чем при коротковолновой радиации. За счет большой глубины проникновения в ткани тела коротковолновая область спектра ИКИ обладает выраженным общим действием на организм человека, вызывая повышение температуры глубоколежащих тканей: например, при длительном облучении глаза может привести к помутнению хрусталика (профессиональная катаракта).

Под влиянием ИКИ в организме человека возникают биохимические сдвиги и изменения функционального состояния центральной нервной системы: образуются специфические биологически активные вещества типа гистамина, холина, повышается уровень фосфора и натрия крови, усиливается секреторная функция желудка, поджелудочной и слюнной желез, в центральной нервной системе развиваются тормозные процессы, уменьшается нервно-мышечная возбудимость, понижается общий обмен.

При инфракрасном облучении кожи повышается ее температура, изменяется тепловое ощущение. При значительных интенсивностях возникают ощущения жжения, боль. Время переносимости тепловой радиации уменьшается с увеличением длины волны и ее интенсивности, (табл.1)

Таблица 1. Время переносимости (в секундах) инфракрасной радиации в зависимости от ее интенсивности и длины волны.

Интенсивность радиации, Вт/м2	Длина волны, мкм	
	3.6	1.07
1400 2800 4200 5600	159 27.3 12.9 9.5	305 37.9 21.2 14.5

Участки кожи, подвергающиеся инфракрасному облучению, получают большое количества тепла, перегреваются. При сильном перегреве происходит омертвление тканей, так называемый термический ожог. Перегрев сначала носит местный характер, но вследствие циркуляции крови он вскоре охватывает весь организм и самочувствие работающего значительно ухудшается.

При длительном воздействии высокой температуры и лучистой энергии температура тела человека может повыситься на 1-2 °С. Из организма тогда усиливается выделение пота, причем пот содержит значительное количество поваренной соли, вследствие чего происходит обеднение крови солью и самочувствие человека ухудшается. При прекращении работы и переходе в помещение с нормальной температурой спустя 20-30 мин. Восстанавливается нормальное самочувствие.

Влияние радиационного тепла различно в зависимости от зоны облучения: наибольший эффект наблюдается при облучении шейной области верхней половины туловища, наименьший - при облучении ног (области бедра). Выносливость к облучению возрастает с увеличением периода облучения, при котором наблюдаются процессы приспособления (адаптация), сохраняющиеся довольно долго.

В довольно редких случаях, когда перегрев достигает 40,5 °С и выше и организм не
в состоянии справиться с ним и нарушениями, которые перегрев вызывает, может наступить тепловой удар. Человек тогда впадает в чрезвычайно болезненное состояние,
которое при определенных условиях может привести к смерти.

Для защиты от инфракрасного излучения применяются следующие экраны: непрозрачные, полупрозрачные и прозрачные.

Непрозрачные экраны могут быть теплоотражающими, теплопоглощающими и теплоотводящими. В теплоотражающих экранах используется алюминиевая фольга-альфоль в листовой или гофрированной форме. Теплопоглощающие экраны изготовляются из асбеста, металла, футерованного теплоизоляционным материалом - заслонки, щиты и др. Асбестовые экраны применяются при интенсивности потока до 3500 Вт/м^2, футерованные - до 10000 Вт/м^2. Теплоотводящие экраны представляют собой сварные или литые конструкции, охлаждаемые протекающей внутри водой.

Полупрозрачные экраны изготовляют из металлической сетки, цепей, армированного стальной сеткой стекла и применяются: сетки - при интенсивности излучения 350 - 1000 Вт/м^2, цепные завесы и армированное стекло - 700 - 5000 Вт/м^2.

Прозрачные экраны могут быть теплопоглощающими и теплоотводящими. Теплопоглощающие экраны изготовляют из силикатных, кварцевых и органических стекол, бесцветных, окрашенных или металлизированных тонкими пленками.

Теплоотводящие экраны - *водяные завесы* - образуются слоем воды или распыленной водой. Водяные завесы применяются при интенсивности излучения 350 -1400 Вт/м^2

Кратность ослабления теплового потока, т, защитным экраном определяется по формуле:

m = , где (7)

$\mathbf{E_u}$ - интенсивность потока излучателя, Вт/м^2

$\mathbf{E_э}$ - интенсивность потока теплового излучения экрана, Вт/м^2

2. Кратность снижения температуры излучающей поверхности, **μ**, определяется по формуле:

μ = , где (8)

3. $\mathbf{t_u}$ - температура излучателя, °С,

$\mathbf{t_э}$ - температура экрана, °С

Коэффициент пропускания экраном теплового потока , **τ**, равен:

τ = 1/m (9)

Коэффициент эффективности экрана, **η**, равен:

η = 1 — τ = (10)

4. ***1.2. НОРМИРОВАНИЕ ВОЗДЕЙСТВИЯ ТЕПЛОВОГО ИЗЛУЧЕНИЯ***

Интенсивность теплового облучения работающих от нагретых поверхностей технологического оборудования, осветительных приборов, инсоляции на постоянных и непостоянных рабочих местах не должна превышать:

-35 Вт/м^2 при облучении 50% поверхности тела и более,

-70 Вт/м^2 - при величине облучаемой поверхности от 25 до 50% и

-100 Вт/м^2 - при облучении не более 25% поверхности тела.

Интенсивность теплового облучения работающих от открытых источников (нагретый металл, стекло, "открытое" пламя и др.) не должна превышать 140 Вт/м^2, при этом облучению не должно подвергаться более 25%) поверхности тела и обязательным являе гея использование средств индивидуальной защиты, в том числе средств защиты лица и глаз (ГОСТ 12.1.005-88

Контрольные вопросы:

1. Назовите основные источники излучения вредные для человека?

Вопросы	Ответы
1. Возник пожар -	А-Ползи к выходу
2. В комнате много дыма	Б-Звони 01
3. На тебе горит одежда –	В-Дыши через мокрую тряпку
4. Задымился телевизор-	Г-Забросай землей
5. Горит старая трава-	Д-Открой окно, балкон
6. Почувствовал запах газа-	Е-Отключи, накрой одеялом
7. Трудно дышать от едкого дыма-	Ж- Падай на землю, катайся

2. Где возникают источники электромагнитных излучений?

3. Что является источником инфракрасного излучения?

Лабораторная работа № 4

Пожарная безопасность

Цель :

1. Перечислите все возможные средства тушения пожара.
2. Каковы должны быть ваши действия, если из работающего телевизора или из какого-то другого электроприбора пошёл дым?
3. Чем нельзя тушить электропроводку под напряжением? Почему?
4. Что делать при сильном задымлении квартиры едким густым дымом?
5. Соотнесите

6. .Составить пары из фраз и пословиц поговорок, одинаковых по смыслу

Лабораторная работа № 5.

«Изучение первичных средств пожаротушения»

Цель проведения урока:

Обучающая:

- ознакомить студентов с основными способами, средствами и правилами пожаротушения,
- сформировать знания и умения по классификации, устройству и применению огнетушителей,
- научить правилам пользования огнетушителями;

Воспитательная:

- развитие у обучающихся коммуникативной культуры,
- воспитание чувства ответственности за собственную и другие жизни;

Развивающая:

- развитие мышления и мировоззрения

Методы обучения:

Практическое занятие,

Материально-техническое обеспечение урока:

Наглядные пособия

Методические указания, инструкционные карты

Дидактический материал: обучающий конспект, нормативные документы

Межпредметные связи:

Учебные дисциплины: Обществознание, Химия, Физика, Охрана труда,

Ход урока.

Организационный момент:

Отметить присутствующих, проверить готовность группы к уроку: – 5 мин.

1.Сам сможешь сделать	А-В красный день, как обгорелый пень
2.Все нарядные, а ты неряха	Б-Не боги горшки обжигают
3.Вещи быстро изнашиваешь	В-Хоть горшком назови, только в печку не ставь
4.Сильно ударился	Г- Высох как спичка
5.С тобой мне надежно	Д- Из огня да в полымя
6.Сильно болеет	Е- Из глаз искры посыпались
7.Правду не скроешь	Ж- Сгореть со стыда
8.И тут плохо и там тоже	З- За тобой – как за каменной стеной
9.Мне стыдно	И- Правда в огне не горит и в воде не тонет
10Мнем все равно, как меня называют, лишь бы относились хорошо	К- Как на огне горит

План изложения нового материала:

1. Задание 1 (Учебник стр. 286-288).

Записать основные термины и понятия, заполнить таблицу 1,

Наименование огнетушащих веществ	*Характеристика воздействия на пожар*	*Положительное действие*	*Отрицательное действие*	*Запрещено применять при тушении*

2. Задание 2 (Учебник стр. 288-296). Записать основные термины и понятия, заполнить таблицу 2.

Марка огнетушителя	*Продолжительность действия, с*	*Дальность струи, м*	*Огне гася-щая способ-ность, $м^2$*	*Область применения*	*Хранение*

3. Изучение общих правил пожаротушения.

4. Ответить на тестовые вопросы.

Вопросы на закрепление изучаемого материала:

1. Назовите основные способы пожаротушения.
2. В каких случаях воду для тушения пожара использовать нельзя.
3. Что относится к первичным средствам пожаротушения.
4. Назовите основные виды огнетушителей.
5. Назовите основные правила тушения пожаров при помощи огнетушителей.

Домашнее задание для выполнения самостоятельной работы:

Изучить тему по учебнику стр.286-294. Прочитать конспект лекций. Оформить практическую работу.

Используемая литература:

Косолапова Н.В., Прокопенко Н.А. Основы безопасности жизнедеятельности: Учебник для учреждений нач. и сред. проф. образования, М.: Издательский центр «Академия», 2016

Инструкционная карта для практической работы № 5
«Изучение первичных средств пожаротушения»

Задание 1. Изучение основных способов пожаротушения и различных видов огнегасящих веществ.

Пожар - Неконтролируемое горение, причиняющее материальный ущерб, вред жизни и здоровью граждан, интересам общества и государства.
Огнегасящее вещество - Вещество, обладающее физико-химическими свойствами позволяющими создать условия для прекращения горения.

Горение — это сложный физико-химический процесс взаимодействия горючего вещества и окислителя, сопровождающийся выделением большого количества тепла и свечением.

Выбор способов и средств пожаротушения зависит от объекта (категории помещений по взрывопожарной и пожарной опасности), характеристики горящих материалов и класса пожара.

Для того, чтобы произошло возгорание, необходимо наличие четырёх условий:

- Горючий материал,
- Окислитель,
- Источник огня,
- Наличие путей распространения пожара.

Горючие вещества и материалы могут быть в трех агрегатных состояниях: жидком, твердом и газообразном. Все горючие вещества на молекулярном уровне содержат углерод и водород, т. е. основные составляющие газо-воздушной смеси, необходимые для реакции горения.

Окислителем обычно бывает кислород воздуха.

Источник зажигания - открытый огонь, пламя, искра электрического или механического происхождения, молния, лучистая энергия, накаленные тела,

химическая реакция, электроток, механическая энергия.

Возникновение и продолжение горения возможно при определенном количественном соотношении горючего вещества и кислорода, а также при определенной температуре и тепловой энергии источника воспламенения. Чтобы началось горение, горючая среда должна быть нагрета до определенной температуры при помощи источника зажигания.

Виды пожаров в зависимости от вида горящих веществ и материалов

Классификация по виду материалов, вовлечённых в пожар, важна для правильного выбора средств тушения, в первую очередь, ручных огнетушителей.

- класс A — пожары твердых горючих веществ и материалов.
- класс B — пожары горючих жидкостей или плавящихся твердых веществ и материалов.
- класс C — пожары газов.
- класс D — пожары металлов.
- класс E — пожары горючих веществ и материалов электроустановок, находящихся под напряжением.
- класс F — пожары ядерных материалов, радиоактивных отходов и радиоактивных веществ.

Способы прекращения процесса горения.

Для прекращения горения необходимо либо снизить тепловыделение в зоне горения фронта пламени, либо увеличить из фронта пламени теплоотвод. Цель – понизить температуру горения до критической температуры гашения.

№ п/п	*Способ прекращения горения*	*Применяемые огнетушащие вещества*
1	Охлаждение зоны горения и поверхности горящих веществ	Вода (до 1700^0C на пожаре сплошными струями и тонкораспыленной водой), вода со смачивателями и загустителями, водные растворы солей, твердый CO_2, снег, перемешиванием.
2	Изоляция очага горения от атмосферного воздуха	Химическая и воздушно-механическая пены, огнетушащие порошковые составы, аэрозоли, негорючие сыпучие вещества (песок, земля, шлаки и т.п.), листовые негорючие материалы.
3	Химическое торможение (ингибирование) реакций горения.	Галоген углеводороды (хладоны, фреон в 10 раз эффективнее CO_2) огнетушащие порошковые составы, аэрозоли, (соли металлов)
4	Сбивание пламени.	Сильной струей газа или воды, слоем продуктов взрыва, подрывом в горючего вещества.
	Изоляция горящих веществ от зоны горения.	Негорючие газы (CO, N 42 0, дымовые газы), водяной пар, тонкораспыленная вода, газо-водяные смеси, аэрозоль.

Огнетушащие вещества

К огнетушащим веществам относятся вода, песок, земля, порошок, газ, пена.

Вода является основным средством тушения пожара, вследствие своей доступности, простоты и дешевизны. Вода, обладая высокой теплоемкостью и скоростью испарения, оказывает на очаг горения сильное охлаждающее действие. При этом, в процессе испарения образуется большое количество пара, который поглощает много тепла и оказывает изолирующее действие на очаг пожара.

К недостаткам воды следует отнести плохую смачиваемость и проникающую способность по отношению к большинству материалов и веществ.

Водой нельзя тушить:

- легковоспламеняющиеся и горючие жидкости (нефть, керосин, бензин, масла), т.к. плотность воды больше их плотности, вода тяжелее, поэтому горючая горящая жидкость поднимается вверх и продолжает гореть;
- электрические сети и установок, находящихся под напряжением, т.к. вода электропроводна и при попадании воды на электропровода может возникнуть короткое замыкание или возможно поражение человека электрическим током.

Песок, земля - одни из простейших средств тушения небольших очагов пожара. Их огнегасящее действие заключается в том, что они препятствует проникновению к очагу пожара кислорода воздуха, а также могут охлаждать место очага пожара.

Пена образуется в результате смешивания воды и пенообразователя в установках тушения. Применяют также химические и воздушно-пенные огнетушители. Раствор пенообразователя, проходя через сетку насадка огнетушителя, смешивается с засасываемым воздухом и образует воздушно-механическую пену, которая, попадая на горящее вещество, охлаждает его и изолирует от кислорода. Пеной можно тушить различные вещества и материалы, за исключением щелочных и щелочноземельных элементов, а также электроустановок, находящихся под напряжением.

Порошок для тушения применяют в составе огнетушителей и установок для тушения. Порошок, попадая на горящее вещество, изолирует его от кислорода, содержащегося в воздухе, выделяет огнегасящие вещества, попав в зону огня.

Огнетушащие средства

Ломы, багры, топоры применяют для разборки горящих деревянных или металлических конструкций для проникновения в здание, спасение людей и окончательной ликвидации пожара.

Лопата - применяется для забрасывания места очага пожара песком, землей.

Кошма - плотная брезентовая ткань - предназначена для изоляции очага горения от доступа воздуха, применяется лишь при небольшом очаге горения.

Пожарная автоматическая сигнализация

Пожарная сигнализация является важной мерой предотвращения крупных пожаров.

Основная задача автоматической пожарной сигнализации

- обнаружение начальной стадии пожара,
- передача извещения о месте и времени его возникновения,
- при необходимости включения автоматических систем пожаротушения

и дымоудаления.

Функционально автоматическая пожарная сигнализация состоит из приёмно – контрольной станции, которая через сигнальные линии соединена с пожарными извещателями и оповещателями.

Пожарный извещатель - техническое средство, предназначенное для формирования сигнала о пожаре.

Пожарный оповещатель - техническое средство, предназначенное для оповещения людей о пожаре.

Задачей сигнальных извещателей является преобразование различных проявлений пожара в электрические сигналы. Приёмно – контрольная станция после получения сигнала от первичного извещателя включает световую и звуковую сигнализацию и при необходимости автоматические установки пожаротушения и дымоудаления. В настоящее время наиболее часто используют тепловые, дымовые, световые и звуковые пожарные извещатели.

Тепловые пожарные извещатели устанавливают под потолком в таком положении, чтобы тепловой поток, обтекая чувствительный элемент извещателя, нагревал его. Тепловые пожарные извещатели не обладают высокой чувствительностью, поэтому обычно не дают ложных сигналов срабатывания в случае увеличения температуры в помещении при включении отопления, выполнения технологических операций.

Дымовые пожарные извещатели обладают меньшей инерционностью по сравнению с тепловыми. Они бывают точечными и линейно- объемными. Точечные дымовые извещатели используют ионизационный эффект. При попадании дыма в открытую камеру происходит уменьшение электрического тока, в результате чего включается цепь электрического реле. Линейно – объемный дымовой извещатель оптического типа работает по принципу изменения силы света при задымлении.

Световые извещатели работают на принципе регистрации инфракрасного или ультрафиолетого излучения пламени. Они обладают высокой чувствительностью и включают сигнализацию почти немедленно после появления небольшого источника радиационной теплоты в пределах прямой видимости извещателя.

Звуковые пожарные оповещатели представляют собой приемопередатчик ультразвуковых колебаний, который настраивают на форму стоячей волны в пределах защищаемого объема. Принцип действия в результате изменения скорости звука в воздушном пространстве из-за влияния образующихся при пожаре конвективных потоков.

Автоматические средства пожаротушения

Эффективно справиться с огнем, сохранить жизнь человека, а также частное и

коммерческое имущество позволяют автоматические установки пожаротушения. Автоматические системы пожаротушения (АСПТ) справляются со своей задачей оперативно и результативно

Автоматические стационарные установки пожаротушения в зависимости от используемых огнетушащих веществ подразделяют на водяные, пенные, газовые и порошковые. Наиболее широкое распространение получили установки водяного и пенного тушения двух типов спринклерные и дренчерные.

Спринклерная установка – наиболее эффективное средство тушения обычных горючих материалов в начальной стадии развития пожара. Спринклерные установки включаются в работу автоматически при повышении температуры в защищаемом объеме выше заданного предела. Вся система состоит из трубопроводов, прокладываемых под потолком помещения и спринклерных оросителей, размещаемых на трубопроводах с заданным расстоянием друг от друга.

Дренчерные установки отличаются от спринклерных отсутствием клапана в оросителе. Дренчерный ороситель всегда открыт. Включение дренчерной системы в действие производится вручную или автоматически по сигналу автоматического извещателя с помощью контрольно – пускового узла, размещаемого на магистральном пожарном трубопроводе. Спринклерная установка срабатывает над очагом пожара, а дренчерная орошает водой весь защищаемый объект.

Задание 2. Изучение назначения, устройства и принципа действия первичных средств тушения пожаров.

Виды огнетушителей и их применение

Огнетушителем *называется устройство, предназначенное для тушения пожара в начальной стадии его возникновения. Тушение производится за счет содержащегося в огнетушителе вещества, способного быстро загасить пламя.*
Потому-то в каждом офисе, средстве передвижения, а по-хорошему, и в каждом доме необходим **противопожарный огнетушитель**.
Обычно они представляют собой баллон, преимущественно красного цвета (традиционная маркировка противопожарного оборудования), с соплом или трубкой. Вещество, которым предполагается гасить огонь, находится в баллоне под давлением и при необходимости выходит через сопло или трубку.

Классификация огнетушителей

1. По способу срабатывания:

Ручные. Запускаются человеком при обнаружении возгорания. К этому типу относится большая часть переносных огнетушителей, используемых в гражданских целях при работе в небольших помещениях или предназначенных для хранения в автомобилях.

Автоматические, то есть такие устройства, которые не требуют присутствия человека. Это стационарные виды, имеющие в своей конструкции датчики дыма. Их помещают обычно в местах повышенной пожарной опасности. Срабатывают тогда, когда температура превышает заданную величину.

2. В зависимости от объема баллона и мобильности огнетушители делятся на:

Ручные малолитражные (до 5 л). Зачастую используются в автомобилях, где они являются обязательными согласно правилам дорожного движения и в офисных помещениях;

Промышленные ручные (5-10-20 л*).* Они применяются чаще всего в отдельно стоящих небольших цехах, где подобные способы тушения огня будут иметь эффективность;

Передвижные и стационарные (от 20 кг и до 400 кг). Этот тип предназначен для эксплуатации в больших складских или промышленных помещениях, где находится скопление легковоспламеняющихся материалов. Обычно передвижные огнетушители перевозят на тележке.

3. От способа подачи содержимого наружу. Содержимое выходит:

- под собственным давлением закачанного внутрь огнетушащего средства;
- под давлением газов, закачанных в корпус;
-под давлением газов, которые находятся в баллончике, встроенном внутри огнетушителя;
- под давлением газов, которые образуются в результате химической реакции всех присутствующих в огнетушителе компонентов.

4. По виду пускового устройства, которые бывают:

- пистолетного типа;
- с вентильным затвором;
-с пуском, который зависит от давления;

5. По веществу, содержащемуся внутри баллона:

- водные (обозначаются аббревиатурой ОВ);
- воздушно-эмульсионные огнетушители (ОВЭ).
- воздушно-пенные огнетушители (ОВП);
-газовые или углекислотные (ОУ или Х);
-порошковые (ОП).

Характеристика огнетушителей

1. Водные огнетушители (ОВ). Данный огнетушитель пригоден для пожаров класса А – тушение твердых горючих веществ. Если в состав воды входят специальные добавки, то этот огнетушитель также можно будет использовать и для тушения жидких горючих веществ, это уже называется пожар класса В. Сразу стоит сказать, что кроме указанных выше ситуаций водными огнетушителями ничего тушить не стоит, т.к. вода может вступить в реакцию с этими предметами. Водные огнетушители являются самыми экологически чистыми и безопасными из всех представленных далее видов огнетушителей.

2. Воздушно-эмульсионные огнетушители (ОВЭ), в своем составе гасящие вещества такого типа огнетушителей имеют фторосодержащие компоненты. По своему назначению этот вид огнетушителей также подходит для тушения пожаров **класса А и В.**

3. Воздушно-пенные огнетушители (ОВП). В основе своего действия имеют процесс химического или воздушно-механического образования пены из водных растворов. При химическом образовании пены применяются различные кислоты и щелочи, а в воздушно-механической технологии за основу пенообразования берется воздействие различными газами, чаще всего азотом или углекислотой. Такими огнетушителями можно гасить пожары класса А, В, а иногда и класса **С** (при котором возникает горение газообразных веществ).

К недостаткам этого типа огнетушителей относится в первую очередь невозможность тушить ими пожары в электросетях из-за того, что пена является хорошим проводником электричества. Также малоэффективны они в тушении спиртов и активных металлов, поскольку те вступают в химическую реакцию с пеной. Еще один существенный недостаток воздушно-пенных огнетушителей – их характеристики, а именно низкий температурный режим, который позволяет работать с ними только в диапазоне от +5 до +45 градусов С.

4. Газовые огнетушители. Довольно большая группа огнетушителей. К ним относятся: углекислотные; аэрозольные; углекислотно-бромэтиловые.

Выпускаются такие огнетушители как ручные, так и передвижные. Принцип их работы состоит в переведении гасящего вещества из жидкого состояния в газообразное при распылении. В процессе та же углекислота, к примеру, увеличивается в объеме в 400 раз, что сопровождается значительным понижением температуры (до -70-80 градусов). Также эффект тушения достигается за счет вытеснения кислорода из области возгорания углекислым газом.

Главное что стоит знать, при использовании углекислотных ручных огнетушителей

1) В момент выхода углекислоты из раструба не прикасаться к нему рукой, открытыми частями тела, т.к. можно получить обморожение,(т.к. углекислота охлаждает раструб при выходе до t - 79,4 С),

2) Нельзя тушить такие вещества, которые могут продолжать гореть и без доступа кислорода (различные сплавы магния, алюминия, натрия и пр.).

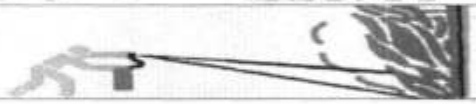
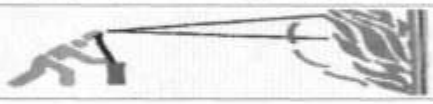
5. Порошковые огнетушители (ОП) являются наиболее универсальным видом. Они применяются практически во всех классах пожаров и в связи с этим подразделяются на ОП общего назначения, предназначенные для тушения пожаров класса А, В, С, Е (возгорание электрооборудования, находящегося под напряжением), ОП специального применения, которые используются в сложных случаях классов В, С, Е.

Основу гасящего порошка в ОП составляют различные минеральные соли. К ним добавляются дополнительные вещества,

препятствующие комкованию и увлажнению порошка.
Основными недостатками этого вида противопожарного оборудования являются высокая степень загрязнения порошком помещения, где производились мероприятия по тушению пожара, а также высокая степень запыленности воздуха при их применении. Также из-за предрасположенности порошков к отсыреванию и комкованию необходимо строго соблюдать условия хранения подобных огнетушителей, чтобы сохранить их в рабочем состоянии.

Общие принципы тушения пожаров при помощи ручных огнетушителей.

1. Огнетушители располагают у выходов из помещений в доступном месте, на высоте1,5 м от пола.
2. При приведении в действие огнетушитель необходимо поднести к месту очага пожара не ближе, чем на 1 м. Сорвать пломбу, выдернуть чеку, направить раструб или ствол (для порошкового) на очаг пожара, нажать на рычаг, приступить к тушению.
3. Тушить очаг пожара с наветренной стороны (чтобы ветер или воздушный поток бил в спину) на расстояние не меньше минимальной длины струи ОТВ огнетушителя (2-4 метра). Необходимо учитывать, что сильный ветер мешает тушению, снося с очага пожара огнетушащее вещество и интенсифицируя горение.
4. На ровной поверхности тушение начинают с передней стороны очага, углубляясь постепенно, по мере тушения
5. При тушении горящего масла запрещается направлять струю заряда сверху вниз.
6. При проливе ЛВЖ тушение начинать с передней кромки, направляя струю порошка на горящую поверхность, а не на пламя
7. Жидкие вещества тушат сверху вниз.
8. Горящую стену или вертикальную поверхность тушат снизу вверх.
9. При тушении электроустановок порошковым огнетушителем подавай заряд порциями через 3-5 секунд.
10. Не подноси огнетушитель ближе 1м к горящей электроустановке.
11. Не берись голой рукой за раструб углекислотного огнетушителя во избежание обморожения.
12. При наличии нескольких огнетушителей следует применять все одновременно.
13. Следите, чтобы потушенный очаг не вспыхнул снова (никогда не поворачивайтесь к нему спиной)
14. Первичные средства пожаротушения всегда должны быть исправны и люди должны уметь ими пользоваться.
15. Запрещается использовать первичные средства пожаротушения не по назначению.

	Правильно	Неправильно
Тушить очаг пожара с наветренной стороны		
При проливе ЛВЖ тушение начинать с передней кромки, направляя струю порошка на горящую поверхность, а не на пламя		
Истекающую жидкость тушить сверху вниз		
Горящую вертикальную поверхность тушить снизу вверх		
При наличии нескольких огнетушителей, необходимо применять их одновременно		
Следите, чтобы потушенный очаг не вспыхнул снова (никогда не поворачивайтесь к нему спиной)		
После использования, огнетушители сразу необходимо отправить на перезарядку		

Ответить на тестовые вопросы

1. **Какой класс пожара можно тушить огнетушителем, если на его корпусе имеется обозначение класса «Е»?**
   А) горение газообразных веществ
   Б) горение металла или металлосодержащих веществ
   + В) горение электрооборудования, находящегося под напряжением

2. В каких огнетушителях в качестве огнетушащего средства используется двууглекислая сода?
   А) аэрозольных;
   Б) порошковых;
   + В) пенных;
   Г) углекислотных.

3. С какой целью используют при пожаре в помещении распылённую струю?
   А) для понижения температуры среды;
   Б) для повышения концентрации кислорода
   + В) для более эффективного тушения пламени.

4. Какие действия не допускаются при использовании углекислотного огнетушителя?
   А) срывать пломбу и выдёргивать чеку;
   Б) держать огнетушитель в горизонтальном положении;
   + В) направлять раструб руками непосредственно на пламя.

5. Пожар – это?
   + А) неконтролируемый процесс горения;
   Б) любое возгорание с образованием пламени и дыма;
   В) неконтролируемый процесс горения, сопровождающийся уничтожением материальных средств и создающий опасность для жизни людей.

6. Как потушить загоревшуюся на человеке одежду?

А) направить на него струю огнетушителя

Б) повалить человека на землю и накрыть плотной тканью

В) сорвать с него одежду.

7. Можно ли тушить горящие газы водой и пеной?

А) да,

+ Б) нет,

В) при определенных условиях.

8. **Какие средства пожаротушения относятся к первичным?**

А) пожарные краны, ручные огнетушители, ящики с песком, асбестовые и войлочные полотна, бочки с водой

+ Б) пожарные краны, ручные огнетушители, установки пожаротушения, ящики с песком, асбестовые и войлочные полотна, бочки с водой

В) ручные огнетушители, ящики с песком, асбестовые и войлочные полотна, бочки с водой.

9. Что относится к огнетушащим средствам

А) вода, пена, газ,

Б) песок, земля, порошок,

+ В) ломы, багры, топоры, лопаты, кошма.

10. Переносные малолитражные огнетушители имеют объем гасящего вещества

+ А) до 4-х литров,

Б) от 4-х до 8 –и литров,

В) свыше 8-и литров.

11. Какой класс пожара можно тушить огнетушителем, если на его корпусе имеется обозначение класса «Д»?

А) горение жидких веществ

+ Б) горение металлов и металлосодержащих веществ

В) горение твердых веществ.

12. По способу срабатывания огнетушители подразделяются на:

+ А) ручные, автоматические, комбинированные,

Б) переносные, стационарные, передвижные,

В) водные, пенные, порошковые.

13. Какой класс пожара можно тушить огнетушителем, если на его корпусе имеется обозначение класса «А»?

А) горение жидких веществ,

+ Б) горение твердых веществ,

В) горение газообразных веществ.

14. Первичные средства пожаротушения используют:

+ А) для ликвидации пожара в начальной стадии возникновения

Б) для тушения большого пожара

В) для локализации стихийного бедствия природного характера.

15. Для тушения каких пожаров применяются воздушно-пенные огнетушители

+ А) Пожаров класса А и В

Б) Пожаров класса С и D

В) Пожаров класса Е и F.

16. Какие сведения обязательны для наличия в паспорте огнетушителя?

+ А) дата последней зарядки (проверки) и вес заряда

Б) дата изготовления

В) дата последнего использования (если было)

Г) место изготовления и приобретения огнетушителя.

17. Каким средством невозможно потушить горюче-смазочные материалы?

+ А) водой**,**
Б) песком,
В) пеной.

18. Назвать тип огнетушителя, предназначенного для тушения разнообразных материалов, предметов, веществ:

А) пенный;
+ Б) углекислотный;
В) воздушно-пенный.

19. Пожар какого класса можно потушить водным огнетушителем

+ А) Пожар класса А,
Б) Пожар класса В,
В) Пожар класса С.

20. Назвать огнетушитель, предназначенный для тушения возгораний различный материалов, кроме веществ горение, которых происходит без доступа воздуха:

А) углекислотный;
Б) химический пенный;
В) воздушно-капельный.

Лабораторная работа №6

Тема: Оформление проведения инструктажей.

Цель работы - Закрепить и систематизировать полученные знания по оформлению проведения инструктажей.

Задание: Письменно заполнить инструктаж, ответить на контрольные вопросы. Сделать вывод о проделанной работе.

Студент выбирает из таблицы №1 задание в соответствии с номером положения фамилии в учебном журнале.

Таблица1 - Номера вариантов заданий к практической работе №1

Вариант			
Вид инструктажа	1	2	3
	Вводный	Внеплановый	Целевой

Пояснения к работе:

Виды инструктажей работников по охране труда, порядок их проведения и оформления

Все виды инструктажей следует считать элементами учебы. При инструктаже особое внимание надо уделять рабочим со стажем до 1 года, а также опытным рабочим с большим стажем. Эти категории рабочих наиболее подвержены травматизму. В первом случае - из-за неопытности, во втором - из-за чрезмерной самоуверенности. Разбор несчастных случаев, проработка приказов есть также своеобразная форма обучения. По характеру и времени проведения инструктажи подразделяют на:

1) вводный;

2) первичный на рабочем месте;

3) повторный;

4) внеплановый;

5) целевой.

Вводный инструктаж и первичный на рабочем месте проводятся по утвержденным программам.

Вводный инструктаж

Вводный инструктаж по безопасности труда проводит инженер по охране труда или лицо, на которое возложены эти обязанности, со всеми вновь принимаемыми на работу не зависимо от их образования, стажа работы по данной профессии или должности, с временными работниками, командированными, учащимися и студентами, прибывшими на производственное обучение или практику, а также учащимися в учебных заведениях. О проведении вводного инструктажа делают запись в журнале регистрации вводного инструктажа с обязательной подписью инструктируемого и инструктирующего, а также в документе о приеме на работу или контрольном листе. Проведение вводного инструктажа с учащимися регистрируют в журнале учета учебной работы.

Первичный инструктаж

Первичный инструктаж по охране труда на рабочем месте до начала производственной деятельности проводит непосредственный руководитель работ по инструкциям по охране труда, разработанным для отдельных профессий или видов работ:

- со всеми работниками, вновь принятыми в организацию, и переводимыми из одного подразделения в другое;

- с работниками, выполняющими новую для них работу, командированными, временными работниками;

- со строителями, выполняющими строительно-монтажные работы на территории действующей организации;

- со студентами и учащимися, прибывшими на производственное обучение или практику перед выполнением новых видов работ, а также перед изучением каждой новой темы при проведении практических занятий в учебных лабораториях, классах, мастерских, участках.

Лица, которые не связаны с обслуживанием, испытанием, наладкой и ремонтом оборудования, использованием инструмента, хранением и применением сырья и материалов, первичный инструктаж не проходят.

Перечень профессий и должностных работников, освобожденных от первичного инструктажа на рабочем месте, утверждает руководитель организации по согласованию с профсоюзным комитетом и службой охраны труда. Все работники, в том числе выпускники профтехучилищ, после первичного инструктажа на рабочем месте должны в течение первых 2 - 14 смен (в зависимости от характера работы, квалификации работника) пройти стажировку по безопасным методам и приемам труда на рабочем месте под руководством лиц, назначенных приказом (распоряжением) по предприятию (подразделению, цеху, участку и т.п.). Ученики и практиканты прикрепляются к квалифицированным специалистам на время практики.

Повторный инструктаж

Повторный инструктаж проходят все работающие, за исключением лиц, освобожденных от первичного инструктажа на рабочем месте, не зависимо от их квалификации, образования и стажа работы не реже чем через 6 месяцев. Его проводят с целью проверки знаний правил и инструкций по охране труда, а также с целью повышения знаний индивидуально или с группой работников одной профессии, бригады по программе инструктажа на рабочем месте. По согласованию с соответствующими органами государственного надзора для некоторых категорий работников может быть установлен более продолжительный (до 1 года) срок прохождения повторного инструктажа.

Повторный инструктаж проводится по программам первичного инструктажа на рабочем месте.

Внеплановый инструктаж

Внеплановый инструктаж проводится:

• при введении в действие новых или переработанных стандартов, правил, инструкций по охране труда, а также изменений к ним;

• при изменении, технологического процесса, замене или модернизации оборудования, приспособлений и инструмента, исходного сырья, материалов и других факторов, влияющих на безопасность труда;

• при нарушении работающими и учащимися требований безопасности труда, которые могут привести или привели к травме, аварии, взрыву или пожару, отравлению;

• по требованию органов надзора;

• при перерывах в работе - для работ, к которым предъявляются дополнительные (повышенные) требования безопасности труда, более чем 30 календарных дней, а для остальных работ - более двух месяцев.

Внеплановый инструктаж проводят индивидуально или с группой работников одной профессии. Объем и содержание инструктажа определяют в каждом конкретном случае в зависимости от причин или обстоятельств, вызвавших необходимость его проведения. Внеплановый инструктаж отмечается в журнале регистрации инструктажа на рабочем месте с указанием причин его проведения.

Внеплановый инструктаж проводит непосредственно руководитель работ (преподаватель, мастер).

Целевой инструктаж

Целевой инструктаж проводится:

• при выполнении разовых работ, не связанных с прямыми обязанностями работника по специальности (погрузка, выгрузка, уборка территории, разовые работы вне предприятия, цеха и т.п.);

• при ликвидации последствий аварии, стихийных бедствий, производстве работ, на которые оформляется наряд-допуск, разрешение и другие документы.

Целевой инструктаж проводится непосредственно руководителем работ и фиксируется в журнале инструктажей и необходимых случаях - в наряде-допуске.

Содержание отчета:

1. Оформить титульный лист в соответствии с СТП 1.2 – 2005.
2. В практической работе необходимо отразить следующее:

А) Название практической работы.

Б) Цель работы.

В) Задание.

Г) Выполненная работа в соответствии с заданием.

Д) Ответы на практические работы.

Е) Вывод.

Ж) Отчет необходимо оформить в папку.

Контрольные вопросы:

Перечислите виды инструктажей.

Укажите виды построения вводного инструктажа.

Укажите виды построения внепланового инструктажа.

Укажите виды построения целевого инструктажа.

Лабораторная работа №7
ТЕМА: «Первая помощь пострадавшим»

ЦЕЛЬ ЗАНЯТИЯ: Изучить общие принципы оказания первой помощи пострадавшим и приемы оказания первой помощи.

ПРИОБРЕТАЕМЫЕ УМЕНИЯ И НАВЫКИ: Научиться оказывать первую помощь пострадавшему.

НОРМА ВРЕМЕНИ: 2 часа

ОСНАЩЕНИЕ РАБОЧЕГО МЕСТА: инструкционные карты,

ЗАДАНИЯ:

ЗАДАНИЕ № 1. Изучить общие принципы оказания первой помощи пострадавшим. Перечислить этапы оказания первой помощи пострадавшему.

ЗАДАНИЕ № 2. Ознакомиться с приемами оказания искусственного дыхания и массажа сердца. Описать операции подготовки к искусственному дыханию, перечислите порядок выполнения искусственного дыхания и массажа сердца.

ЗАДАНИЕ № 3 . Ознакомиться с приемами оказания первой помощи. Перечислите способы остановки кровотечения. Заполнить таблицу № 1.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ:

Общие принципы оказания первой помощи пострадавшим.

Первая доврачебная помощь пострадавшему имеет важное значение для спасения жизни и последующего восстановления здоровья человека. Умение безотлагательно проводить ряд простейших действий по оказанию помощи до прибытия медицинского персонала во многих случаях позволяет предотвратить смертельный исход и развитие тяжелых осложнений у пострадавшего.

Первую доврачебную помощь должен уметь оказывать каждый человек. Поэтому необходимо проходить обучение способам оказания первой помощи.

Первая помощь пострадавшему оказывается в несколько последовательных этапов.

1. Оценка обстановки и незамедлительное прекращение действия повреждающего фактора (электрического тока, температуры, излучения, механического воздействия).
1. Удаление пострадавшего из опасной зоны в место, где будет оказываться дальнейшая помощь.
2. Выявление причины тяжелого состояния пострадавшего, характера повреждения, признаков жизни и смерти.
3. Оказание первой помощи пострадавшему с использованием приемов, определяемых характером повреждения и состоянием пострадавшего.
4. Вызов медицинского персонала, скорой медицинской помощи, доставка пострадавшего в лечебное учреждение. Вызов медицинского персонала при тяжелом состоянии пострадавшего должен быть произведен незамедлительно.

Для эффективности доврачебной помощи в каждом подразделении предприятия, организации должна быть медицинская аптечка с набором медикаментов, перевязочных средств, средств остановки кровотечения, плакаты с правилами оказания доврачебной помощи, указатели для облегчения поиска аптечки и медицинского пункта. В каждом подразделении должен быть ответственный за своевременное пополнение аптечки и поддержания ее в надлежащем состоянии.

Перед оказанием первого этапа помощи пострадавшему необходимо быстро оценить обстановку на месте, степень опасности действующего повреждающего фактора и исключить возможность самому попасть под его действие.

Искусственное дыхание. Назначение искусственного дыхания — обеспечить газообмен в организме, т. е. насыщение крови пострадавшего кислородом и удаление из крови углекислого газа.

Способы искусственного дыхания. Существует множество различных способ ов выполнения
искусственного дыхания. Все они делятся на две группы: аппаратные и ручные.

Аппаратные способы требуют применения специальных аппаратов, которые обеспечивают вдувание и удаление воздуха из легких через резиновую трубку, вставленную в дыхательные пути, или через маску, надетую на лицо пострадавшего. Простейшим из аппаратов является ручной портативный аппарат, предназначенный для искусственного дыхания и аспирации (отсасывания)
жидкости и слизи из дыхательных путей. Основными частями его являются небольшой мех, приводимый в действие рукой, и маска, плотно накладываемая на рот и нос пострадавшего.

Ручные способы значительно менее эффективны и несравненно более трудоемки, чем аппаратные. Они обладают, однако, тем важным достоинством, что могут выполняться без каких-либо приспособлений и приборов, т. е. немедленно при возникновении нарушений деятельности дыхания у пострадавшего.

Среди большого числа существующих ручных способов наиболее эффективным является *способ «изо рта в рот».* Он заключается в том, что оказывающий помощь вдувает воздух из своих легких в легкие пострадавшего через его рот или нос. *Подготовка к искусственному дыханию.* Прежде чем приступить к искусственному дыханию, необходимо быстро выполнить следующие операции:

1. освободить пострадавшего от стесняющей дыхание одежды — расстегнуть ворот, развязать галстук, расстегнуть брюки и т. п.;
2. уложить пострадавшего на спину на горизонтальную поверхность — стол или пол;

максимально запрокинуть голову пострадавшего назад, положив под затылок ладонь одной руки, а второй рукой надавливать на лоб пострадавшего до тех пор, пока подбородок его не окажется на одной линии с шеей*).* При этом положении головы язык отходит от входа в гортань, обеспечивая тем самым свободный проход для воздуха в легкие. Вместе с тем при таком положении головы обычно рот раскрывается. Для сохранения достигнутого положения головы под лопатки следует подложить валик из свернутой одежды; пальцами обследовать полость рта, и, если обнаружится инородное содержимое (кровь, слизь и т. п.), необходимо удалить его, вынув одновременно зубные протезы, если они имеются. Для удаления слизи и крови необходимо голову и плечи пострадавшего повернуть в сторону (можно подвести свое колено под плечи пострадавшего), а затем с помощью носового платка или края рубашки, намотанного на указательный палец, очистить полость рта и глотки. После этого необходимо придать голове первоначальное положение и максимально запрокинуть ее назад. *Выполнение искусственного дыхания.* По окончании подготовительных операций оказывающий помощь делает глубокий вдох и затем с силой выдыхает воздух в рот пострадавшего. При этом он должен охватить своим ртом весь рот пострадавшего, а пальцами зажать ему нос*).* Затем оказывающий, помощь откидывается назад, освобождая рот и нос пострадавшего, и делает новый вдох. В этот период грудная клетка пострадавшего опускается и происходит пассивный выдох. Контроль за поступлением воздуха в легкие пострадавшего осуществляется на глаз по расширению грудной клетки при каждом вдувании. Если после вдувания воздуха грудная клетка пострадавшего не расправляется, это свидетельствуето непроходимости дыхательных путей. В этом случае необходимо выдвинуть нижнюю челюсть пострадавшего вперед. Для этого нужно поставить четыре пальца каждой руки позади углов нижней челюсти и, упираясь большими пальцами в ее край, выдвинуть нижнюю челюсть вперед так, чтобы нижние зубы стояли впереди верхних. Легче выдвинуть нижнюю челюсть введенным в рот большим пальцем. Иногда оказывается невозможным открыть рот пострадавшего вследствие судорожного сжатия челюстей. В этом случае искусственное дыхание следует производить *по способу «изо рта в нос»,* закрывая рот пострадавшего при вдувании воздуха в нос.

В одну минуту следует делать 10—12 вдуваний взрослому человеку (т. е. через 5...6 с). При появлении у пострадавшего первых слабых вдохов следует приурочивать искусственный вдох к началу самостоятельного вдоха.

Искусственное дыхание необходимо проводить до восстановления глубокого ритмичного дыхания.

Массаж сердца производится ритмичным надавливанием на грудь, т. е. на переднюю стенку грудной клетки пострадавшего. В результате этого сердце сжимается между грудиной и позвоночником и выталкивает из своих полостей кровь. После прекращения надавливания грудная клетка и сердце распрямляются, и сердце заполняется кровью, поступающей из вен. Кровообращение необходимо для того, чтобы кровь доставляла кислород ко всем органам и тканям организма. Следовательно, кровь должна быть обогащена кислородом, что достигается искусственным дыханием. Таким образом, *одновременно с массажем сердца должно производиться искусственное дыхание.*

Подготовка к массажу сердца является одновременно подготовкой к искусственному дыханию, поскольку массаж сердца должен производиться совместно с искусственным дыханием.

Для выполнения массажа необходимо уложить пострадавшего на спину на жесткую поверхность (скамью, пол или в крайнем случае подложить под спину доску).

Необходимо также обнажить его грудь, расстегнуть стесняющие дыхание предметы одежды.

Для выполнения массажа сердца нужно встать с какой-либо стороны от пострадавшего в такое положение, при котором возможен более или менее значительный наклон над ним. Затем определить прощупыванием место надавливания (оно должно находиться примерно на два пальца выше мягкого конца грудины, и положить на него нижнюю часть ладони одной руки, а затем поверх первой руки положить под прямым углом вторую руку и надавливать на грудную клетку пострадавшего, слегка помогая при этом наклоном всего корпуса Предплечья и плечевые кости рук

Рис 8

оказывающего помощь должны быть разогнуты до отказа. Пальцы обеих рук должны быть сведены вместе и не должны касаться грудной клетки пострадавшего. Надавливать следует быстрым толчком так, чтобы сместить нижнюю часть грудины вниз на 3...4 см, а у полных людей на 5...6 см. Усилие при надавливании следует концентрировать на нижней части грудины, которая более подвижна. Следует избегать надавливания на верхнюю часть грудины, а также на окончания нижних ребер, т. к. это может привести к их перелому. Нельзя надавливать ниже края грудной клетки (на мягкие ткани), поскольку можно повредить расположенные здесь органы, в первую очередь печень.

Надавливание (толчок) на грудину следует повторять примерно *1 раз в секунду*. После быстрого толчка руки остаются в достигнутом положении в течение примерно 0,5 с. После этого следует слегка выпрямиться и расслабить руки, не отнимая их от грудины.

Для обогащения крови пострадавшего кислородом одновременно с массажем сердца необходимо проводить искусственное дыхание по способу «изо рта в рот» (или «изо рта в нос»).

Если помощь оказывает один человек, следует чередовать проведение указанных операций в следующем порядке: *после двух глубоких вдуваний в рот или нос пострадавшего — 15 надавливаний на грудную клетку,* затем снова два глубоких вдувания и 15 надавливаний для массажа сердца и т. д.

Эффективность наружного массажа сердца проявляется в первую очередь в том, что при каждом надавливании на грудину на сонной артерии четко прощупывается пульс. Для определения пульса указательный и средний пальцы накладывают на адамово яблоко пострадавшего и, продвигая пальцы вбок, осторожно ощупывают поверхность шеи до определения сонной артерии (рис. 8). Другими признаками эффективности массажа является сужение зрачков, появление у пострадавшего самостоятельного дыхания, уменьшение синюшности кожи и видимых слизистых оболочек.

Для повышения эффективности массажа рекомендуется на время наружного массажа сердца приподнять (на 0,5 м) ноги пострадавшего. Такое положение ног пострадавшего способствует лучшему притоку крови в сердце из вен нижней части тела.

Искусственное дыхание и наружный массаж сердца следует производить до появления самостоятельного дыхания и восстановления деятельности сердца или до передачи пострадавшего медицинскому персоналу.

О восстановлении деятельности сердца пострадавшего судят по появлению у него собственного, не поддерживаемого массажем регулярного пульса. Для проверки пульса через каждые 2 мин прерывают массаж на 2...3 с. Сохранение пульса во время перерыва свидетельствует о восстановлении самостоятельной работы сердца. При отсутствии пульса во время перерыва необходимо немедленно возобновить массаж.

Отсутствие пульса при появлении других признаков оживления организма (самостоятельного дыхания, сужения зрачков, попытки пострадавшего двигать руками и ногами и др.) служит признаком фибрилляции сердца. В этом случае необходимо продолжать оказание помощи пострадавшему до прибытия врача или до доставки пострадавшего в лечебное учреждение, где будет произведена дефибрилляция сердца. В пути следует беспрерывно оказывать помощь пострадавшему, производя искусственное дыхание и массаж сердца вплоть до момента передачи его медицинскому персоналу.

Приемы оказания первой помощи

Кровотечения. Кровотечение бывает наружным и внутренним. Если кровь вытекает из раны или естественных отверстий наружу, то такое кровотечение называют наружным, если же она скапливается в полостях тела — внутренним. Различают артериальное, венозное и капиллярное кровотечения. Наиболее опасным является артериальное, во время которого кровь изливается под давлением, она ярко-красного (алого) цвета и бьет пульсирующей струей в такт с сокращениями сердечной мышцы. Скорость кровотечения при ранении крупного артериального

сосуда (сонная, плечевая, бедренная артерия, аорта и др.) такова, что буквально в течение считанных минут может произойти потеря крови, несовместимая с жизнью.

Кровь при венозном кровотечении темно-вишневого цвета вытекает медленно, равномерно и непрерывной струей. Оно менее интенсивное, чем артериальное, и поэтому реже приводит к необратимым изменениям. Однако при ранении, например, вен шеи и грудной клетки в момент вдоха в их просвет может поступить воздух. Пузырьки воздуха, попадая с током крови в сердце, могут стать причиной смерти.

Капиллярное кровотечение наблюдается при поверхностных ранах, неглубоких порезах кожи, ссадинах. Кровь из раны вытекает медленно по каплям, и при нормальной свертываемости кровотечение прекращается самостоятельно.

При кровотечении следует временно остановить его, наложив обычную или давящую повязку, жгут.

Для остановки *артериального кровотечения* необходимы энергичные меры, и если кровоточит небольшая артерия, то бывает достаточно наложения давящей повязки. При сильном кровотечении наиболее надежным способом является пережатие кровоточащего сосуда поясным ремнем, резиновой трубкой, прочной веревкой и т. п., которые накладывают выше места кровотечения, сделав 2—3 оборота вокруг конечности по типу наложения жгута.

Следует запомнить, что время пережатия кровоточащего сосуда не должно превышать 1,5...2 ч в теплое время года, а **в** холодное до 1... 1.5 ч, т. к. может произойти омертвение конечности. Поэтому для контроля длительности пережатия сосуда необходимо отметить точное время наложения жгута.

Пережимать сосуд надо до остановки кровотечения. Если это сделано правильно, то пульсация ниже жгута не определяется. В то же время нельзя очень сильно затягивать жгут, т. к. это может вызвать деформацию мышц, повреждение нервов и стать причиной паралича конечности.

До момента наложения жгута для временной быстрой остановки кровотечения прижимают артерию пальцем выше места ее повреждения. После наложения жгута пострадавшего немедленно транспортируют в лечебное учреждение для окончательной остановки кровотечения. Если доставка задерживается, то по истечении критического времени с целью частичного восстановления кровообращения жгут следует на 2...3 мин ослабить, а затем наложить вновь несколько выше или ниже. На период освобождения конечности от жгута артериальное кровотечение сдерживают прижатием пальца. При необходимости ослабление и наложение жгута приходится повторять через каждые 30 мин зимой, через каждые 50...60 мин летом.

Кроме того, для временной остановки кровотечения можно прижать артерию фиксацией конечностей в определенном положении. Так, при повреждении подключичной артерии останавливают кровотечение максимальным отведением рук за спину с фиксацией их на уровне локтевых суставов.

Венозное кровотечение останавливают при помощи плотно наложенной поверх раны давящей повязки, прикрытой чистым бинтом или другой материей.

Капиллярное кровотечение можно легко остановить наложением на рану обычной повязки.

Кровотечение из носа прекращают наложением на область переносицы льда, снегом или емкости с холодной водой, можно использовать смоченный холодной водой платок, бинт, салфетку и др. При продолжении кровотечения нужно прижать пальцами обе половины носа к носовой перегородке. Сжимать нос надо не менее 3...5 мин, а при необходимости и больше. Вместе с тем в носовые наружные ходы можно ввести ватные тампоны, смоченные раствором перекиси водорода, — при этом голову больного следует несколько наклонить вперед.

Ушибы, растяжения, вывихи. При *растяжениях* необходимо создать покой поврежденной части, для чего на сустав надо наложить тугую повязку и по возможности придать ей возвышенное положение, поверх повязки на область повреждения с целью уменьшения боли, уменьшения развития отека тканей приложить пузырь со льдом, с холодной водой и т. д.

При *вывихе* нужно зафиксировать конечность повязкой или косынкой, наложить холод на поврежденную область. Не следует самому пытаться вправлять поврежденную часть конечности, т. к. нередко это может сопровождаться переломом.

Переломы бывают открытые и закрытые. Открытые переломы более опасны, чем закрытые, т. к.

при них происходит беспрепятственное загрязнение и попадание микробов непосредственно в область перелома, что может повлечь за собой серьезные осложнения, которые в дальнейшем резко затрудняют процесс срастания перелома и выздоровление пострадавшего.

Признаками перелома являются резкая боль, усиливающаяся при небольшом движении; неестественные положение и форма конечности; подвижность вне сустава; в области перелома быстро появляются припухлость и кровоподтеки, а нередко заметное на глаз укорочение конечности.

При оказании помощи нужно быстро наложить шины на область перелома, дать обезболивающие средства.

Существуют стандартные шины, однако если их нет, то можно использовать для фиксации костей дощечку, кусок доски, палку и др. При полном отсутствии подходящего материала фиксацию можно выполнить плотным прибинтовыванием поврежденной конечности к здоровой части тела, например верхней конечности к туловищу, нижней конечности — к здоровой ноге.

Фиксация при открытом переломе осуществляется так же, как и при закрытом, но при открытом переломе кожу вокруг раны надо смазывать 3...5%-м раствором йода, а рану закрывать чистой (желательно стерильной) повязкой. При обработке раны не надо пытаться удалять или вправлять торчащие кости.

Черепно-мозговые травмы — сотрясения, ушибы (контузии) головного мозга с возможным разрушением мозговой ткани, при **этом** может произойти потеря сознания (от нескольких секунд до суток и более), возникнуть головная боль, тошнота и рвота, амнезия (потеря памяти), нарушение речи, снижение или потеря чувствительности, отсутствие мимики и т. д.

Первая помощь заключается в наложении повязки (при наличии раны), создания полного покоя. При нарушении дыхания и сердечной деятельности — приступить к проведению искусственного дыхания и массажа сердца.

Раны могут быть резаные, рубленые, колотые, рваные и огнестрельные. Первая помощь заключается в наложении повязки. Перед ее наложением необходимо из раны и вокруг нее убрать видимые на глаз крупные инородные предметы, обработать кожу вокруг раны 3...5%-м раствором йода, не смазывая при этом раневую поверхность и не удаляя инородные тела из глубоких слоев раны. Нельзя также засыпать ее порошком стрептоцида, антибиотиков, антисептическими веществами, накладывать мазь и прикладывать вату, что может усилить нагноение.

Термические ожоги подразделяют на четыре степени. При ожогах I степени появляются покраснение и отек кожи, сопровождающиеся жгучей болью; при ожогах II степени — пузыри на коже, заполненные прозрачной жидкостью; при ожогах III степени верхний слой кожи (эпидермис) практически отсутствует, мягкие покровные ткани отечны, напряжены, поверхность их белесоватой окраски или же покрыта сухой тонкой светло-коричневой коркой, при ожогах IV степени возникает повреждение глубоколежащих тканей, пораженная поверхность черного цвета с признаками обугливания.

При оказании помощи снимать одежду необходимо очень осторожно, с тем чтобы дополнительно не травмировать кожу. Для снятия одежды рекомендуется ее разрезать. Нельзя отрывать обрывки одежды от поверхности ожога — их надо обрезать ножницами, а поверх наложить повязку. При отсутствии стерильного перевязочного материала ожоговую поверхность можно закрыть чистой хлопчатобумажной тканью. Не следует смазывать ожоговую поверхность мазями, животными и растительными маслами, вазелином. Нанесенный жир не улучшит заживление и не снимет боль, а в последующем затруднит хирургическую обработку. Можно наложить повязку с разведенным спиртом, водкой, раствором перманганата калия (марганцовка) — такие повязки уменьшают боль.

При ожогах полезно сразу же поместить обожженное место либо под струю холодной воды из-под крана, либо в емкость с холодной водой на 20...30 мин. Это значительно успокоит боль и уменьшит отечность.

Контрольные вопросы

1. Каковы основные методы и последовательность оказания первой помощи пострадавшему?
2. Как выполняется искусственное дыхание и массаж сердца?
3. Как остановить кровотечение?
4. Перечислите приемы оказания первой помощи при вывихах, переломах и других видах травм.

Список используемой литературы:

1. Л.В. Груманова, В.О. Писарева « Охрана труда и техника безопасности в сфере компьютерных технологий» Г.Н. Бурлак. « Безопасность работы на компьютере: Организация труда на предприятиях информационного обслуживания» 2017г.

2. Ляпина О.П., Безопасность жизнедеятельности, Учебное пособие, Новосибирск, 2017

3.Документ ГОСТ 12.0.003-74* ССБТ. Опасные и вредные производственные факторы. Классификация 1974-11-18 00:00:00

Электронный ресурс «Охрана труда». Форма доступа: www.ohranatruda.ru

Лабораторная работа №8
Оказание первой помощи при переломе

Цель работы: Студенты должны самостоятельно усвоить методику оказания первой медицинской помощи при травмах и переломах

Задание : 1 . Изучить теоретический материал по теме практического занятия

2. Записать в отчет практической работы ответы на следующие вопросы:

-Что такое перелом, виды переломов?

- Признаки перелома?

- Как оказать первую медицинскую помощь при открытом и закрытом переломе?

- Что такое иммобилизация и способы иммобилизации при различных переломах?

- Как оказать первую медицинскую помощь при синдроме длительного сдавливания? - Первая медицинская помощь при травматическом шоке

3. Оформить практическую работу и представить для защиты преподавателю.

ТЕОРЕТИЧЕСКИЙМАТЕРИАЛ ПРИ ПЕРЕЛОМАХ

Переломами принято называть полное или частичное нарушение целости костей. В зависимости от того, как проходит линия перелома по отношению к кости, их подразделяют на: поперечные, продольные, косые, спиральные. Встречаются и оскольчатые, когда кость раздроблена на отдельные части. Переломы могут быть закрытые и открытые (рис. 14). При открытом через рану нередко выступают отломки кости.

В целях грамотного оказания первой медицинской помощи при травме необходимо уметь определить наличие у пострадавшего перелома.

Для перелома характерны резкая боль, усиливающаяся при любом движении и нагрузке на конечность, нарушение ее функции, изменение положения и формы конечности, появление отечности и кровоподтека, укорочение и патологическая подвижность кости.

Обнаружить перелом можно при наружном осмотре поврежденной части тела. Если необходимо, то прощупывают место предполагаемого перелома. При этой травме наблюдаются неровности кости, острые края отломков и характерный хруст при легком надавливании. Ощупывать, особенно для определения подвижности кости вне области сустава, нужно осторожно, двумя руками, стараясь не причинить дополнительной боли и травмы пострадавшему.

Перелом всегда сопровождается повреждением мягких тканей, степень нару-шения которых зависит от вида перелома и характера смещения отломков кости. Особенно опасны повреждения крупных сосудов и нервных стволов, грозными спутниками

которых являются острая кровопотеря и травматический шок. В случае открытого перелома возникает опасность инфицирования раны.

Оказывая первую медицинскую помощь при переломах, ни в коем случае не следует пытаться сопоставить отломки кости — устранить искривление конечности при закрытом переломе или вправить вышедшую наружу кость при открытом. Пострадавшего нужно как можно быстрее доставить в лечебное учреждение.

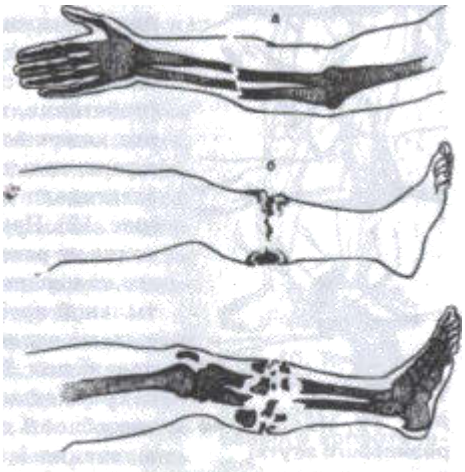


Рис. 14. Закрытый перелом костей предплечья (а) и открытый перелом костей голени (б).

В оказании первой медицинской помощи при переломах и повреждениях суставов главное — надежная и своевременная иммобилизация поврежденной части тела, что приводит к уменьшению боли и предупреждает развитие травматического шока. Устраняется опасность дополнительного повреждения и снижается возможность инфекционных осложнений. Временная иммобилизация проводится, как правило, с помощью различного рода шин и подручных материалов.

Использование шин практиковалось у всех народов с древних времен. Так, раскопки египетских гробниц подтвердили, что за 4-5 тыс. лет до н.э. египтяне применяли различные шины в виде дощечек, коры дерева или кожи животных. В настоящее время предложены сотни различных видов стандартных шин из разнообразных материалов: дерева, пластмассы, металла. При умелом их наложении обеспечивается достаточная неподвижность.

При отсутствии стандартных шин можно использовать подручные средства: доски, палки, фанеру и другие предметы. В исключительных случаях допускается транспортная иммобилизация путем прибинтовывай поврежденной конечности к здоровой части тела: верхней — к туловищу, ни-жней — к здоровой ноге.

Поврежденной конечности необходимо придать наиболее удобное положение, так как последующие исправления часто бывают затруднены из-за болей, воспалительного отека и опасности инфицирования раны. Руку фиксируют в слегка отведенном и согнутом в локтевом суставе под прямым углом положении (рис. 15). Ладонь при этом обращена к животу, пальцы полусогнуты. При переломах нижних конечностей транспортную шину обычно накладывают на выпрямленную ногу. При переломах бедренной кости в нижней трети боль, припухлость и патологическая подвижность отмечаются над коленным суставом. В этих случаях конечность фиксируют согнутой в коленном суставе, а при транспортировке под колени подкладывают валик из одеяла или одежды.

Для придания необходимого положения конечности шину моделируют (придают ей нужную форму) по конечности оказывающего помощь или по здорово й конечности пострадавшего.

Иммобилизирующая повязка должна обеспечивать хорошую фиксацию места перелома, не нарушая существенно кровоснабжения поврежденной конечности. Для выполнения этого требования при наложении транспортной шины нужно обеспечить неподвижность в суставах выше и ниже места перелома, а после ее наложения проверить наличие пульса.

Под шину, обернутую бинтом, в местах костных выступов подкладывают вату или мягкую ткань для предупреждения сильного с давления и боли. При открытом переломе останавливают кровотечение, накладывают асептическую повязку на рану и только после этого приступают к иммобилизации.

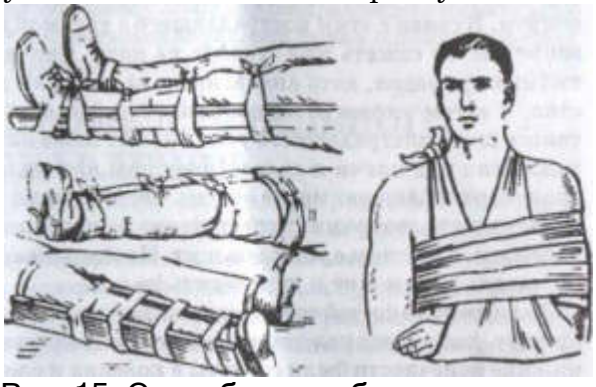


Рис. 15. Способы иммобилизации при переломах.

Переломы позвоночника принадлежат к наиболее тяжелым и болезненным травмам. Основной признак — нестерпимая боль в месте перелома при малейшем движении. Судьба пострадавшего в этих случаях в решающей степени

Рис. 16. Иммобилизация перелома костей таза.

зависит от правильности первой медицинской помощи и способа транспортировки. Даже незначительные смещения отломков костей могут привести к смерти. В связи с этим пострадавшего с травмой позвоночника категорически запрещается сажать или ставить на ноги. Сначала следует ввести из шприц-тюбика промедол, дать анальгин, седалгин или другое обезболивающее средство, а затем уложить на ровный твердый щит или доски. При отсутствии такого щита пострадавшего укладывают лежа на животе на обычные носилки, подложив под плечи и голову подушки или валики. В таком положении его транспортировка наименее опасна. Очень важно помнить, что пострадавшие с переломом позвоночника совершенно н^лереносят перекладывания обычным способом, допустим с земли на щит. Необходимо осторожно уложить их набок, положить рядом щит и перекатить на него.

При переломе костей таза пострадавшего также кладут на спину на твердый щит (фанеру, доски), под колени подкладывают скатанное одеяло или пальто так, чтобы нижние конечности были согнуты в коленях и слегка разведены в стороны (рис. 16). В таком положении конечности фиксируют с помощью распорки и бинтов.

При переломах костей свода черепа пострадавшего укладывают на носилки, под голову подкладывают мягкую подстилку с углублением, а по бокам — мягкие валики, свернутые из одежды или другого подручного материала. Иммобилизацию головы можно осуществить с помощью пращевидной повязки, которая проходит под подбородком и фиксируется к носилкам. При переломах верхней челюсти наиболее простой способ иммобилизации — круговая повязка из бинта или косынки. При ее наложении подтягивают нижнюю челюсть к верхней до смыкания зубов и фиксируют в таком положении вертикальными ходами бинта вокруг головы или косынкой. Можно использовать и пращевидную повязку — для фиксации сомкнутой нижней челюсти. В тех

случаях, когда зубы не смыкаются, между челюстями вводят полоску фанеры или кусочек линейки и прижимают ее к верхней челюсти.

При оказании помощи пострадавшим с переломом нижней челюсти прежде всего принимают меры для устранения или предупреждения асфиксии (удушья). Если человек в результате травмы потерял сознание и лежит на спине, возможно западение языка и немедленное удушье.

Поворот тела облегчает дыхание, в наибольшей степени это достигается в сидячем положении с наклоненной головой или лежа на животе с повернутой набок головой. Иногда прибегают к прошиванию языка булавкой и удержанию его за нитку, фиксированную к одежде или пращевидной повязке, с помощью которой обеспечивают иммобилизацию нижней челюсти.

Первая медицинская помощь при переломе ключицы направлена на обездвижение пояса верхних конечностей. Поврежденную руку лучше уложить на широкую косынку. Транспортировать пострадавшего нужно в положении сидя, слегка откинувшись назад. Не рекомендуется наклоняться вперед, например, садясь в машину, так как при этом возможно дополнительное смещение отломков кости. Для обездвижения применяют и другие способы. Двумя ватно-марлевыми кольцами, связывая которые на спине или крестообразной повязкой, разводят надплечья. При переломе ребер накладывают тугую бинтовую повязку на грудную клетку, делая первые ходы бинта в состоянии выдоха. При отсутствии бинта можно использовать простыню, полотенце или кусок ткани. Транспортируют пострадавшего в положении сидя.

Временная иммобилизация при вывихах и других повреждениях суставов осуществляется так же, как при переломах костей. При этом фиксировать конечность необходимо в положении, которое наиболее удобно для пострадавшего и причиняет ему меньшее беспокойство. Нельзя пытаться вправлять вывих и применять силу для изменения вынужденного положения конечности.

ПРИ СИНДРОМЕ ДЛИТЕЛЬНОГО СДАВЛЕНИЯ

Механические травмы при землетрясениях, оползнях, лавинах, обвалах в шахтах часто имеют специфические особенности. В силу объективных причин значительная часть пострадавших продолжительное время остается под завалами, не имея возможности получить медицинскую помощь. Если придавленная конечность не освобождена от сдавления, то общее состояние пострадавшего может быть удовлетворительным. Боль, которая вначале сдавления была очень сильной, через несколько часов притупляется.

Высвобождение находившейся под завалом конечности без предварительного наложения кровоостанавливающего жгута или закрутки часто приводит к резкому ухудшению состояния пострадавшего с падением артериального давления, потерей сознания, непроизвольным мочеиспусканием. Такое состояние получило название краш-синдром — синдром длительного сдавления (СДС).

Синдром длительного сдавления развивается в результате выброса в кровь миоглобина и других токсических продуктов, которые образовались при некробиотических изменениях в сдавленных тканях (омертвение сдавленных мышц и других тканей). В результате такого выброса развивается тяжелый токсический шок. В дальнейшем миоглобин оседает в почечных канальцах, что приводит к почечной блокаде и тяжелой почечной недостаточности. Пострадавшие погиба-

ют в ранние сроки после травмы от шока, в течение 7 — 10 суток — от почечной недостаточности.

Опыт свидетельствует, что некоторым можно спасти жизнь и после сдавления частей тела в течение нескольких суток, в то же время другие погибают через несколько часов.

Различают легкую, среднюю, тяжелую и крайне тяжелую степень тяжести синдрома длительного сдавления.

Тяжесть синдрома определяется степенью ишемии (нарушение кровоснабжения) тканей при сдавлении, которая независимо от площади и времени сдавления может существенно различаться. От степени нарушения кровоснабжения и ее правильного определения в момент оказания первой медицинской помощи во многом зависит судьба пострадавшего.

Перед высвобождением пострадавшей конечности от сдавления необходимо выше места сдавления наложить жгут (закрутку) — так, как при временной остановке кровотечения. Крайне необходимо ввести обезболивающее средство (промедол, анальгин, седалгин и т.п.).

После высвобождения пострадавшего из-под завала надо определить степень нарушения кровоснабжения тканей, от которой зависит правильность дальнейших действий по оказанию медицинской помощи. Это просто сделать, если знать признаки четырех степеней ишемии.

Первая степень — ишемия компенсированная, которая, несмотря на длительное сдавление, не привела к нарушению кровообращения и обмена веществ в сдавленной конечности. При такой ишемии активные движения сохранены, т.е. пострадавший может самостоятельно двигать пальцами и другими частями сдавленной конечности. Есть тактильная (чувство прикосновения) и болевая чувствительность. В этом случае жгут, наложенный раньше, после высвобождения необходимо срочно снять.

Вторая степень — ишемия некомпенсированная. При такой ишемии тактильная и болевая чувствительность не определяется, активных движений нет, но пассивные свободны, т.е. можно согнуть и разогнуть пальцы и другие части поврежденной конечности легкими усилиями руки оказывающего помощь. Трупного окоченения мышц сдавленной конечности нет. Жгут также нужно срочно снять, так как его пребывание на конечности опасно продолжением ишемии, что может привести к гибели конечности.

Третья степень — ишемия необратимая. Тактильная и болевая чувствительность также отсутствуют. Появляется главный прйзнак — утрата пассивных движений, отмечается трупное окоченение мышц сдавленной конечности. При такой ишемии жгут снимать нельзя.

Четвертая степень — некроз (омертвение) мышц и других тканей, которое заканчивается гангреной. В этом случае жгут также снимать не следует.

После того, как вопрос с жгутом решен, необходимо наложить на имеющиеся раны асептические повязки и произвести иммобилизацию конечности с помощью стандартных шин или подручного материала. По возможности обложить поврежденную конечность пузырями со льдом или грелками с холодной водой, согреть пострадавшего и дать ему щелочное питье. После оказания первой медицинской помощи необходимо принять все меры для быстрейшей эвакуации пострадавшего в лечебное учреждение. Транспортировать его лучше лежа на носилках, желательно в сопровождении медицинского работника.

Д. ПРИ ШОКЕ

Шок представляет собой тяжелую общую реакцию организма, развивающуюся в результате воздействия тяжелой механической или психической травмы, ожо-га, интоксикации и других экстремальных факторов. При шоковом состоянии отмечается расстройство кровообращения и дыхания, нервной и эндокринной регуляции, обмена веществ.

Наиболее часто встречается травматический шок, развивающийся при тяжелых ранениях головы, груди, живота, таза, конечностей.

В начальном периоде шока, особенно если травме предшествовало сильное нервное перенапряжение, пострадавший может быть возбужден, может не сознавать тяжести своего состояния и полученных повреждений. Он мечется, иногда кричит, вскакивает, пытается бежать. Лицо его бледное, зрачки расширены, взгляд беспокойный, дыхание и пульс учащены. В последующем пострадавший становится заторможенным. У него появляется безразличие, полная безучастность к окружающему. Кожа бледная с землистым оттенком, покрыта холодным липким потом, руки и ноги холодные, температура тела снижена. Сознание сохранено. Отмечается учащенное, поверхностное дыхание, пульс частый, иногда не прощупывается. Появляется жажда, иногда рвота.

Шок может возникнуть непосредственно после травмы или через некоторое время после нее. Его появлению способствуют запоздалое и неумелое оказание первой медицинской помощи, небрежная транспортировка пострадавшего.

Первая медицинская помощь заключается прежде всего в прекращении воздействия на пострадавшего травмирующего фактора — основной причины шока. Для этого нужно освободить его из-под завала, погасить горящую одежду, извлечь из воды и т.п. При оказании помощи особое внимание обратить на остановку кровотечения (если это возможно) прижатием сосуда, наложением жгута или другим способом. При переломе или подозрении на него, а также при вывихе обеспечить временную иммобилизацию поврежденной части тела. Все эти действия необходимо производить решительно, но с максимальной осторожностью, не причиняя при этом пострадавшему дополнительных страданий.

Человеку, находящемуся в сознании, дать болеутоляющее средство (анальгин, седалгин, пенталгин и др.). Если у него нет ранения живота, напоить горячим чаем, дать 50 — 100 мл водки. Такого пострадавшего необходимо как можно быстрее доставить в лечебное учреждение.

Лабораторная работа № 9
Способы иммобилизации

Информационный блок

Определение и общие вопросы темы

Иммобилизация *(immobilis* - неподвижный) - комплекс лечебно-профилактических мероприятий, направленный на создание покоя поврежденной анатомической области с целью восстановления анатомических взаимоотношений поврежденных частей тела и профилактики возможных осложнений.

Транспортная иммобилизация - создание неподвижности (покоя) поврежденной части тела с помощью транспортных шин или подручных средств на время, необходимое для транспортировки пострадавшего (раненого) с места получения травмы (поля боя) или этапа медицинской эвакуации в лечебное учреждение.

Различают лечебную и транспортную иммобилизацию. В лечебных учреждениях выполняется лечебная иммобилизация на срок, необходимый для консолидации перелома, восстановления поврежденных структур и тканей.

Транспортная иммобилизация как неотъемлемая часть оказания первой помощи применяется в первые часы и минуты после ранения. Зачастую она играет решающую роль не только в профилактике осложнений, но и в сохранении жизни раненых и пострадавших. С помощью иммобилизации обеспечивается покой, предупреждаются интерпозиция сосудов, нервов, мягких тканей, распространение раневой инфекции и вторичные кровотечения. Кроме того, транспортная иммобилизация является неотъемлемой частью мероприятий по профилактике развития травматического шока у раненых и пострадавших.

Транспортная иммобилизация осуществляется непосредственно на поле боя (месте повреждения) и на этапах медицинской эвакуации. Транспортировка раненого или пострадавшего с переломами и обширными повреждениями без адекватной транспортной иммобилизации опасна и недопустима.

Своевременно и правильно выполненная транспортная иммобилизация является важнейшим мероприятием первой помощи при огнестрельных, открытых и закрытых переломах, обширных повреждениях мягких тканей, повреждениях суставов, сосудов и нервных стволов. Отсутствие иммобилизации во время транспортировки может привести к развитию тяжелых осложнений (травматический шок, кровотечение и др.), а в некоторых случаях и к гибели пострадавшего.

В очаге ЧС в большинстве случаев первая помощь при переломах и обширных ранениях будет оказываться в порядке само- и взаимопомощи. Поэтому медицинская сестра обязана хорошо владеть техникой транспортной иммобилизации.

Показания к транспортной иммобилизации:

- • переломы костей;
- • повреждение суставов: ушибы, повреждения связок, вывихи, подвывихи;
- • разрывы сухожилий;
- • повреждение крупных сосудов;
- • повреждение нервных стволов;
- • обширные повреждения мягких тканей;
- • отрывы конечностей;
- • обширные ожоги, отморожения;
- острые воспалительные процессы конечностей.

Транспортная иммобилизация

Только правильно выполненная, иммобилизация принесет облегчение больному и пострадавшему, поэтому следует неукоснительно придерживаться следующих правил:

1) иммобилизация должна производиться в кратчайшие сроки от, момента повреждения, желательно непосредственно на месте получения травмы;

2) перед наложением иммобилизации необходимо обезболивание пострадавшего;

3) до выполнения транспортной иммобилизации на раны накладывается асептическая повязка;

4) транспортная иммобилизация накладывается поверх одежды и обуви пострадавшего;

5) если необходимо наложить жгут, то его накладывают так, чтобы он был хорошо виден и его можно было бы снять, не нарушая качественную фиксацию конечности в шине;

6) при повреждениях костей конечностей обездвиживаются по меньшей мере два смежных с поврежденным сегментом конечности сустава. При переломах обеих костей голени, бедренной и плечевой костей обязательно обездвиживаются три сустава;

7) средства иммобилизации должны быть заранее выстланы ватно-марлевыми прокладками или подручными материалами (трава, сено и др.) для предотвращения болевых ощущений от давления шины, а в холодное время года — от контактных отморожений;

8) в холодное время года иммобилизованная конечность должна быть утеплена.

При транспортной иммобилизации рекомендуется выполнять правила, которые мы условно назвали «трижды осторожно»:

1) осторожно наложить повязку;

2) осторожно наложить транспортную шину;

3) осторожно перенести, переложить на носилки и транспортировать пострадавшего.

Ошибки при выполнении транспортной иммобилизации.

- *Применени е необоснованно коротких шин из подручных средств*.

В результате средства транспортной иммобилизации не обеспечивают полное обездвиживание области повреждения.

- *Наложение средств транспортной иммобилизации без предварительного обертывания их ватой и марлевыми бинтами.*

Причиной ошибки, как правило, служит поспешность или же отсутствие заранее подготовленных к наложению шин.

- *Невыполненное или недостаточно тщательно выполненное моделирование проволочных шин в соответствии с контурами и положением поврежденной части тела.*
- *Недостаточная фиксация шины к поврежденной части тела бинтом.*

Экономия бинта в таких случаях не позволяет удержать шину в нужном для обездвиживания положении.

- *Концы шины чрезмерной длины либо недостаточно надежно закреплены при бинтовании.*

Это способствует дополнительной травматизации, создает неудобства при транспортировке, не позволяет придать конечности удобное положение.

- *Нечастой, но очень опасной ошибкой является закрытие кровоостанавливающего жгута бинтованием при укреплении шины.* В результате жгут не виден, и его своевременно не снимают, что приводит к омертвению конечности.

Правила транспортной иммобилизации

Транспортная иммобилизация должна выполняться качественно и обеспечивать полный покой поврежденной части тела или ее сегмента. Все действия должны быть продуманными и исполняться в определенной последовательности.

Основные правила при выполнении транспортной иммобилизации.

1. Транспортная иммобилизация поврежденной части тела должна выполняться на месте травмы в максимально ранние сроки после ранения или повреждения. Чем раньше выполнена иммобилизация, тем меньше дополнительное травмирование области повреждения.
2. Перед проведением транспортной иммобилизации необходимо ввести пострадавшему обезболивающее средство (омнопон, морфин, промедол). При этом следует учитывать, что действие обезболивающего препарата наступает только через 5-10 мин. До наступления обезболивающего эффекта наложение транспортных шин недопустимо.
3. Транспортную иммобилизацию на этапах первой и доврачебной помощи выполняют поверх обуви и одежды, так как раздевание пострадавшего является дополнительным травмирующим фактором.
4. Поврежденную конечность иммобилизируют в функциональном положении: верхняя конечность согнута в локтевом суставе под углом 90°, кисть расположена ладонью к животу либо укладывается ладонью на поверхность шины, пальцы кисти полусогнуты, нижняя конечность незначительно согнута в коленном суставе, голеностопный сустав согнут под углом 90°.
5. Гибкие шины необходимо предварительно смоделировать в соответствии с контурами и положением поврежденной части тела (на здоровой конечности или на себе).
6. Перед наложением средств транспортной иммобилизации следует защитить костные выступы (лодыжки, гребни подвздошных костей, крупные суставы) ватно-марлевыми салфетками. Давление жестких шин в области костных выступов приводит к образованию пролежней.
7. При наличии раны на нее накладывается стерильная повязка, и только после этого осуществляется иммобилизация. Противопоказаны наложение повязки и укрепление шины одним и тем же бинтом.
8. В случаях, когда повреждение сопровождается наружным кровотечением, перед транспортной иммобилизацией выполняются его остановка (жгут, давящая повязка), обезболивание, рана укрывается стерильной повязкой.
9. Металлические шины предварительного обертывают ватой и бинтами с целью профилактики пролежней от непосредственного давления на мягкие ткани. При транспортировке в зимнее время металлические шины, охлаждаясь, могут вызвать местное отморожение.
10. Перед транспортировкой в холодное время конечность с наложенной шиной необходимо утеплить, обернув теплой одеждой,одеялом или термопленкой. Если конечность в обуви, то следует расслабить шнуровку. Соблюдение перечисленных общих правил обязательно при выполнении транспортной иммобилизации повреждений любой локализации.

Инструкция

Перед началом выполнения манипуляций Вам необходимо объединится в пары.

Один из студентов будет выступать в роли медицинского работника, другой в роли- пациента.

1. Внимательно прочитайте алгоритм выполнения манипуляции.
2. Повторите показания к выполнению данной манипуляции.
3. Подготовьте все необходимое оснащение.
4. Установите контакт с пациентом.
5. Выполните манипуляцию строго в соответствие с алгоритмом.
   В случае возникновения проблем обратитесь к преподавателю.

После удачного выполнения манипуляции одним из студентов поменяйтесь ролями.

6. После удачного выполнения манипуляции продемонстрируйте свои успехи преподавателю.
7. Для закрепления материала ответьте на вопросы и решите ситуационную задачу.
8. Оцените Ваши действия. Ответив на следующие вопросы:

- Какие проблемы у Вас возникли при выполнении данной манипуляции и почему?
- Какие этапы выполнения манипуляции не вызвали проблем?
- Какие проблемы возникли при решении ситуационной задачи и почему?

9. Оцените Вашу работу на практическом занятии в соответствии со следующими критериями.

КРИТЕРИИ ОЦЕНОК

Критерии оценки при решении задач по оказанию неотложной помощи

5 «отлично» – правильная оценка характера патологии, полное, последовательное перечисление действий с аргументацией каждого этапа;

4 «хорошо» – правильная оценка характера патологии, полное, последовательное перечисление действий, затруднение в аргументации этапов;

3 «удовлетворительно» – правильная оценка характера патологии; неполное перечисление или нарушение последовательности действий, затруднения в аргументации;

2 «неудовлетворительно» – неверная оценка ситуации или неправильно выбранная тактика действий, приводящая к ухудшению состояния пациента.

Критерии оценки выполнения практических манипуляций

5 «отлично» – рабочее место оснащается с соблюдением всех требований к подготовке для выполнения манипуляций; практические действия выполняются последовательно в соответствии с алгоритмом выполнения манипуляций;

соблюдаются все требования к безопасности пациента и медперсонала; выдерживается регламент времени; рабочее место убирается в соответствии с требованиями санэпиднадзора; все действия обосновываются;
4 «хорошо» – рабочее место не полностью самостоятельно оснащается для выполнения практических манипуляций; практические действия выполняются последовательно, но не уверенно; соблюдаются все требования к безопасности пациента и медперсонала; нарушается регламент времени; рабочее место убирается в соответствии с требованиями санэпидрежима; все действия обосновываются с уточняющими вопросами педагога;
3 «удовлетворительно» – рабочее место не полностью оснащается для выполнения практических манипуляций; нарушена последовательность их выполнения; действия неуверенные, для обоснования действий необходимы наводящие и дополнительные вопросы и комментарии педагога; соблюдаются все требования к безопасности пациента и медперсонала; рабочее место убирается в соответствии с требованиями санэпидрежима;
2 «неудовлетворительно» – затруднения с подготовкой рабочего места, невозможность самостоятельно выполнить практические манипуляции; совершаются действия, нарушающие безопасность пациента и медперсонала, нарушаются требования санэпидрежима, техники безопасности при работе с аппаратурой, используемыми материалами.

Критерии оценки работы

-задания выполнены без ошибок- 5 баллов;
-задания выполнены с незначительными замечаниями и дополнениями- 4 балла;
-задания выполнены с ошибками- 3 балла;
-задания не выполнены или выполнено с грубыми ошибками -2 балла.

ТЕХНОЛОГИЯ ВЫПОЛНЕНИЯ ПРОСТОЙ МЕДИЦИНСКОЙ УСЛУГИ ИММОБИЛИЗАЦИЯ ПРИ ПЕРЕЛОМАХ КОСТЕЙ

Код технологии		Название технологии
A15.03.002		Иммобилизация при переломах костей
1.	**Требования к специалистам и вспомогательному персоналу, включая требования**	
1.1	Перечень специальностей/кто участвует в выполнении услуги	1) Специалист, имеющий диплом установленного образца об окончании высшего образовательного учебного заведения по специальностям: Лечебное дело Педиатрия 2) Специалист, имеющий диплом установленного образца об окончании среднего профессионального медицинского образовательного учреждения по специальностям: Лечебное дело Акушерское дело Сестринское дело
1.2	Дополнительные или специальные требования к специалистам и вспомогательному персоналу	Имеются навыки выполнения данной простой медицинской услуги
2.	**Требования к обеспечению безопасности труда медицинского персонала**	
2.1	Требования по безопасности труда при выполнении услуги	До и после проведения процедуры провести гигиеническую обработку рук Использование перчаток во время процедуры.
3.	**Условия выполнения простой медицинской услуги:** Транспортировка в условиях "скорой медицинской помощи" Стационарные Амбулаторно-поликлинические	
4.	**Функциональное назначение простой медицинской услуги:** Профилактическое Лечебное	
5.	**Материальные ресурсы**	
5.1	Приборы, инструменты, изделия медицинского назначения	Транспортные (стандартные) шины: Крамера 60х8 см - 4 шт. 110х11 см Дитерикса Пневматические Деревянный щит Ватно-марлевый воротник или пневматический воротник Кольца Дельбе Ножницы хирургические.

		Стерильная укладка с набором инструментария и перевязочного материала для обработки раны. Пластиковый пакет для отработанного материала.
5.2	Реактивы	Отсутствуют
5.3	Иммунобиологические препараты и реагенты	Отсутствуют
5.4	Продукты крови	Отсутствуют
5.5	Лекарственные препараты	Антисептическое средство для обработки рук. Дезинфицирующее средство Анальгетики (в зависимости от локализации и степени тяжести переломов)
5.6	Прочий расходуемый материал	Бинты (средние, широкие) Вата медицинская Марля Косынка Медицинская клеенка. Стерильный перевязочный материал. Лейкопластырь Перчатки нестерильные

6.	**Характеристика методики выполнения простой медицинской услуги** Иммобилизация при переломах костей осуществляется для создания неподвижности поврежденной части тела, предупреждения смещений отломков и повреждения ими кровеносных сосудов, периферической и центральной нервной системы, внутренних органов и мягких тканей. Основные принципы транспортной иммобилизации. Шину накладывают непосредственно на месте происшествия; Шина обязательно должна захватывать как минимум два сустава (выше и ниже места перелома) при повреждении плеча и бедра - три сустава. Перед применением жесткие стандартные шины необходимо обернуть ватой, фиксировать ее бинтом или надеть чехол из моющейся ткани. Не следует снимать одежду с пострадавшего, если ее невозможно снять, для осмотра места повреждения необходимо разрезать ее по швам. Перед наложением шины необходимо по возможности придать поврежденной конечности средне физиологическое положение, а если это не возможно - такое положение, при котором конечность меньше всего травмируется. Моделирование шины следует проводить, прикладывая ее к поврежденной конечности. При открытых переломах, вправление отломков не производится: останавливают кровотечение временными способами, накладывают стерильную повязку, конечность фиксируют в том положении, в котором она находится. При закрытых переломах, особенно нижних конечностей, фиксируя шину повязкой необходимо проводить легкое и осторожное вытяжение поврежденной конечности по оси. При перекладывании пострадавшего с наложенной шиной на носилки нужно поддерживать поврежденную конечность или часть тела. Алгоритм иммобилизации при переломах костей

	1.Подготовка к процедуре Придать пациенту удобное положение. Объяснить ход предстоящей манипуляции. Обработать руки гигиеническим способом, осушить. Надеть перчатки. Освободить место травмы от одежды. Осмотреть место травмы для определения характера повреждения. Подготовить необходимое количество шин нужной длины и ширины. Обернуть их ватой, фиксировать ее бинтом (марлей) или надеть специальный чехол. 2. Выполнение процедуры 2.1. Иммобилизация конечности при переломе костей кисти и пальцев Подготовить лестничную шину длиной от концов пальцев до локтевого сустава. Наложить шину на поврежденную конечность с ладонной стороны. Придать кисти физиологическое положение, вложить в ладонь ватный валик. Фиксировать шину к конечности бинтом, оставив пальцы открытыми. Иммобилизировать поврежденную конечность с помощью косынки. 2.2. Иммобилизация конечности при переломе костей предплечья Подготовленную проволочную шину, длиной 80 см, смоделировать по контуру здоровой руки (от пястно-фаланговых суставов до середины плеча). Осторожно согнуть поврежденную конечность в локтевом суставе под прямым углом, предплечье привести в среднее положение между пронацией и супинацией, кисть немного разогнуть, вложить в ладонь плотный валик. Уложить поврежденную конечность на подготовленную шину. Зафиксировать шину бинтом. Иммобилизировать поврежденную конечность с помощью косынки. 2.3. Иммобилизация конечности при переломе плечевой кости Подготовить лестничную шину нужной длины (от концов пальцев поврежденной конечности до плечевого сустава противоположной стороны), моделировать по не поврежденной конечности. В подмышечную впадину подкладывают ватно-марлевый валик, который фиксируют лентой бинта, проведенной через грудь и завязанной на здоровом надплечье. Поврежденную конечность согнуть в локтевом суставе под острым углом. Уложить на внутреннюю поверхность шины кисть в среднем положении между супинацией и пронацией, вложить под согнутые пальцы кисти ватно-марлевый валик. Направить шину по задненаружной поверхности поврежденной конечности через плечо на спину до плечевого сустава противоположной стороны. Связать концы шины бинтом через неповрежденное надплечье и подплечную область. Фиксировать шину к конечности спиральными ходами бинта до локтевого сгиба или средней трети плеча. Для лучшей фиксации конечности положить повязку Дезо. 2.4. Иммобилизация конечности при переломе ключицы, лопатки, головки шейки плеча 2.4.1. Иммобилизация с использованием косыночной повязки 1. Взять косынку с двумя длинными концами и одним коротким. Середину косынки

подвести под согнутое в локтевом суставе до 90° предплечье.
2. Один конец косынки расположить между предплечьем и туловищем и провести через здоровое плечо, другой, находящийся спереди от предплечья, - через больное плечо. Оба конца завязать сзади на шее. Вершину косынки (третий ее конец) загнуть в области локтевого сустава кпереди и закрепить булавкой.

2.4.2. Иммобилизация путем бинтования
1. Поврежденную руку слегка отвести в сторону, подложить в подмышечную область ватно-марлевый валик, чтобы поднять плечевой сустав и отодвинуть его от грудной клетки плечо прижимать к туловищу и наложить повязку.
2. Согнуть поврежденную руку под прямым углом и плотно прижать к груди.
3. В таком положении прибинтовать поврежденную руку к туловищу круговыми витками бинта от здоровой стороны грудной клетки по направлению к больной.

2.4.3. Иммобилизация кольцами Дельбе:
1. Смоделировать диаметр стандартных колец по здоровому плечу или изготовить кольца из марли и ваты - валик из ваты завернуть в марлю квадратной формы 60х60 см "с угла на угол", наружные углы марли, свободные от ваты, завязать узлом в виде кольца.
2. Кольца одеть на плечи с обеих сторон - как лямки рюкзака в положении максимального сведения лопаток.
3. Плечи отвести назад, а кольца связать между собой за спиной при помощи бинта.

2.5. Иммобилизация конечности при переломе ребер и грудины
2.5.1. Иммобилизация бинтованием
Пострадавшему выполнить не тугое бинтование грудной клетки или стянуть грудь полотенцем. Тугое бинтование широкими бинтами и бинтование проводят на глубоком выдохе. Или же наложить спиральную повязку на грудную клетку, на выдохе обернуть вокруг грудной клетки ткань (сложенную простынь, марлю, полотенце) зашить край нитками или фиксировать булавками. Убедиться в отсутствии открытого пневмоторакса. Если он обнаружен, то наложить герметизирующую асептическую повязку перед бинтованием грудной клетки.

2.5.2. Иммобилизация с применением лейкопластыря
Наложить на место перелома лейкопластырную черепицеобразную повязку. Широкие полоски лейкопластыря наложить от края грудины до позвоночника так, чтобы последующая полоска прикрывала предыдущую на половину. Повязка должна захватывать область 2-3-х ребер выше и ниже места перелома.

2.6. Иммобилизация конечности при переломе лодыжек и костей стопы.
Подготовить две лестничные шины нужной длины (первая, длиной 80 см, шириной 11 см, располагается от верхней трети голени по задней поверхности ее, под стопу до конца пальцев. Вторая, длиной 120 см, шириной 8 см, моделируется в виде дуги, от верхней трети голени наружной стороны под стопу до верхней трети голени внутренней стороны).
Наложить шины на поврежденную конечность, фиксировать марлевым бинтом.

2.7. Иммобилизация конечности при переломе костей голени
Подготовить три лестничные шины нужной длины (первую, длиной 120 см, шириной 11 см, расположить от ягодичной складки поврежденной конечности по задней поверхности бедра, голени, под стопу до концов пальцев; вторую, длиной 80 см, шириной 8 см, от верхней трети наружной поверхности бедра до наружного края

	стопы и загнуть под стопу; третью шину, такой же длины и ширины как вторая, моделировать так же как вторую и расположить на внутренней поверхности поврежденной конечности. Уложить поврежденную конечность на первую шину, подложить ватные валики в подколенную ямку и над пяткой. Приложить две оставшиеся шины к наружной и внутренней поверхности поврежденной конечности, наложить шины от верхней трети бедра до стопы с фиксацией коленного и голеностопного суставов Фиксировать шины на конечности спиральными турами бинта. 2.8. Иммобилизация конечности при переломе бедра 2.8.1. Иммобилизация проволочными шинами Крамера 1. Подготовить три лестничные шины нужной длины (первая моделируется нижнею угла лопатки поврежденной стороны по задней поверхности бедра, голени, по стопу до конца пальцев. Вторая от подмышечной ямки по наружной поверхности туловища, бедра голени до наружного края стопы и загибается под стопу. Третья располагается от паховой области по внутренней поверхности поврежденной конечности под стопу.) 2. Приподнять пострадавшую конечность и уложить ее на первую шину. Подложить в под коленную ямку и над пяткой ватный валик. Приложить две оставшиеся шины. 3. Фиксировать шины марлевым бинтом. 2.8.2. Иммобилизация с использованием шины Дитерикса 1. Смоделировать шину по здоровой конечности: короткую планку шины смоделировать по внутренней стороне конечности, ее длина должна быть от паховой складки и на 10-15 см длиннее конечности; длинную планку шины смоделировать по наружной стороне конечности и туловища, ее длина должна быть от подмышечной впадины и на 10-15 см. длиннее конечности. 2. Конечность слегка согнуть в тазобедренном и коленном суставах, стопа по отношению к голени должна находиться под углом 90°. 3. В подмышечную впадину и в область промежности укладывают ватно-марлевые валики. 4. Фиксировать подошвенную часть шины, крепление для шнура наружу, к стопе 8 - образной повязкой. 5. Короткую (внутреннюю) планку, нижнюю ее часть, завести во внутреннюю скобу "подошвы" (шина должна выстоять от "подошвы" на 10 см.), валик шины должен прилегать к бедру и упираться в промежность 6. Подвижную часть на торце внутренней планки, с отверстием, установить под углом 90°, в отверстие просунуть веревку от "подошвы". 7. Длинную (наружную) планку шины, нижнюю ее часть, завести в наружную скобу подошвы (шина должна выстоять от подошвы на 10 см.), а шип ее - в паз торцовой, установленной под углом 90 , планки. 8. вложить под костные выступы (лодыжек, коленного сустава, большого вертела и крыла подвздошной кости) прокладку из ваты для предупреждения сдавления и развития некроза. 9. Наружную шину уложить вдоль конечности и туловища до подмышечной впадины, зафиксировать ремнем через плечо противоположной стороны и через талию, другим ремнем - к туловищу. 9. Внутреннюю шину зафиксировать ремнем к бедру. 10. Шину в области голени зафиксировать бинтами. 11. С помощью палочки, закручивая веревку, выполнить натяжение конечности.

	Палочку зафиксировать. 2.9. Иммобилизация конечности при переломе костей таза 1. Переложить пострадавшего на деревянный щит, твердые носилки, на спину с использованием трех помощников (минимум вдвоем), поднимая одновременно все части тела. 2. Под колени пострадавшего подложить валик, изготовленный из одеяла, подушки или одежды так, чтобы конечности были согнуты в коленных и тазобедренных суставах. 3. Бедра развести в стороны на 20-30 см и зафиксировать их ремнем, наложив ремень 8-образно в области нижней трети бедер. 2.10. Иммобилизация конечности при переломе позвоночника 1. Переложить пострадавшего на деревянный щит с использованием трех помощников (минимум вдвоем), поднимая одновременно все части тела. 2. Под позвоночник, в область перелома, подложить валик. 3. При переломе шейного отдела позвоночника необходимо одеть на шею пострадавшего ватно-марлевый или пневматический воротник и закрепить его специальными застежками. 3. Окончание процедуры 1. Сообщить пострадавшему о результатах иммобилизации и дальнейших действиях (если он находится в сознании). 2. Перенести пострадавшего в автомобиль для транспортировки в лечебное учреждение. 3. Снять перчатки, поместить их в контейнер для дезинфекции 4. Обработать руки гигиеническим способом, осушить. 5. Записать о факте иммобилизации в сопроводительный лист, принятый в службе скорой медицинской помощи.
7.	**Дополнительные сведения об особенностях выполнения методики:** Одежду с пострадавшего снимать нельзя! При отсутствии стандартных шин иммобилизацию можно выполнить с помощью подручных средств.
8.	**Достигаемые результаты и их оценка:** В процессе иммобилизации общее состояние не ухудшилось. Шина обеспечивает неподвижность как минимум двум суставам, выше и ниже места перелома, при повреждении плеча и бедра трем суставам.
9.	**Форма информированного согласия пациента при выполнении методики и дополнительная информация для пациента и членов его семьи** Пациент должен быть информирован о предстоящий иммобилизации. Информация о необходимости проведения иммобилизации сообщается медицинским работником с указанием ее цели
10.	**Параметры оценки и контроля качества выполнения методики** - Отсутствие осложнений во время и после проведения процедуры.

	- Отсутствуют отклонения от алгоритма выполнения измерения. - Наличие записи о результатах выполнения назначения в медицинской документации. - Своевременность выполнения процедуры (в соответствии со временем назначения). - Удовлетворенность пациента качеством предоставленной медицинской услуги.
11.	**Стоимостные характеристики технологий выполнения простой медицинской услуги** Коэффициент УЕТ врача - 1,0. Коэффициент УЕТ медицинской сестры - 1,0.
12.	**Графическое, схематическое и табличное представление технологий** Отсутствует
13.	**Формулы, расчеты, нормограммы, бланки и другая документация (при необходимости)** Отсутствует

Техника иммобилизации верхней конечности

Показания: Проводится при ранениях и заболеваниях плечевого сустава и лопатки, закрытых и открытых переломах плеча, обширных повреждениях мягких тканей плеча и плечевого пояса.

Оснащение: Чаще накладывается шина Крамера длиной 120 см., бинт, ножницы, перчатки.

Последовательность действий:

Шина должна выступать на 2—3 см из-за кончиков пальцев (допускается до пястно-фаланговых суставов) и доходить до внутреннего края лопатки здоровой стороны. Фиксируется 3 сустава.

1. Моделирование шины.

Проводится на себе или на здоровой конечности пострадавшего:

а) измеряем расстояние от кончиков пальцев до локтевого сустава и сгибаем шину под прямым углом;

б) затем измеряем расстояние от локтевого сустава до плечевого сустава и сгибаем шину под тупым углом для плечевого сустава;

в) далее моделируем шину по надплечью и спине до здоровой лопатки.

Шина должна плотно прилегать к руке и туловищу.

2. Наложение шины и фиксация с помощью бинтов.

Сначала накладывается крестообразная повязка на лучезапястный сустав, затем спиральная на предплечье и плечо.

3. Рука в шине подвешивается с помощью косынки.

Рука может быть подвешена с помощью тесьмы из бинта, привязанной к верхнему концу шины.

Иммобилизация предплечья

Проводится при повреждениях и заболеваниях предплечья и локтевого сустава.

Используется короткая шина Крамера длиной 80 см. Шина должна выступать из-за кончиков пальцев на 2 — 3 см и доходитъ до середины плеча. Фиксируются 2 сустава.

1. Моделирование шины.

Измеряется расстояние от кончиков пальцев до локтевого сустава и шина сгибается под прямым углом.

2. В кисть вкладывается валик, чтобы пальцы удерживались в полусогнутом положении.

3. Наложение шины и фиксация с помощью бинтов.

Сначала накладывается спиральная повязка на лучезапястный сустав, затем на предплечье и плечо.

4. Рука в шине подвешивается на косынку.

Иммобилизация бедра, коленного и тазобедренного сустава

Проводится при повреждениях и заболеваниях.

Достигается наложением специалъной шины Дитерихса или с помощью 5 шин Крамера. Фиксируются 3 сустава.

Техника иммобилизации с помощью шин Крамера

1. Две длинные шины связываются и укладываются по задней поверхности ноги и туловища. Сгибаются под прямым углом для стопы и должны выступать за пальцы. Далее моделируются по задней поверхности ноги и туловища и должны доходить до ребер.
2. Две другие длинные шины связываются и укладываются от подмышечной впадины до наружного края стопы, где крючкообразно сгибается под углом 90^0 для стопы.
3. Пятая шина (короткая или длинная) идет по внутренней поверхности ноги и у внутреннего края стопы крючкоооразно сгибается кнаружи, поддерживая стопу под углом 90°.
4. Шины фиксируются марлевыми бинтами, мягкими или ремнями. Сначала накладывается крестообразная повязка на голеностопный сустав, затем спиральная на голень и бедро. На туловище шины можно фиксировать лямками или ремнями.

Техника наложение шины Дитерихса

Шина состоит из 4 частей: «подошвы, наружной и внутренней части (планка, костыль, бранша) и палочки-закрутки со шнуром-петлей. В этой шине сочетается фиксация и одновременным вытяжением конечности.

Наложение шины Дитерихса состоит из семи этапов:

1. Подошвенная часть шины тщательно фиксируется бинтами к подошвенной поверхности стопы, по типу восьмиобразной повязки.

2. Наружная часть шины (более длинная) закрепляется в подвижной части на такой длине, чтобы шина, начиналась у подмышечной впадины и на 8—10 см выступала за подошвенную поверхность стопы; затем она вставляется через металлическое ушко в подошвенной части шины.

3. Внутренняя часть шины, упираясь в промежность, укладывается по внутренней поверхности конечности, пройдя через внутреннее металлическое ушко подошвенной части шины;

4. Для предупреждения провисания задней поверхности голени рекомендуется дополнительно использовать шину Крамера, укрепив ее спиральной повязкой.

5. Под костные выступы (лодыжек, коленного сустава, большого вертела и крыла подвздошной кости) подкладывается прокладка из ваты для предупреждения сдавления и развития некроза;

6. Циркулярными ходами бинта и ремнями закрепляется на голени, бедре, животе и грудной клетке;

7. При помощи закрутки осуществляется вытяжение. Палочка-закрутка помещается за выступ наружной части шины.

Вопросы для закрепления темы:

1. Дайте определение транспортной иммобилизации.
2. Перечислите виды транспортной иммобилизации.
3. Перечислите виды транспортных шин.
4. Сколько суставов необходимо иммобилизировать при переломе плеча?
5. Сколько суставов необходимо иммобилизировать при переломе костей предплечья?
6. Сколько суставов необходимо иммобилизировать при переломе костей голени?
7. Назначение шины Дитерихса.
8. Как зафиксировать конечность при переломе костей предплечья используя подручные средства?
9. Протяженность шины Крамера при переломе бедра.
10. Как наложить пневматическую шину при переломе бедра?
11. Последовательность оказания неотложной помощи при открытом переломе.
12. Последовательность оказания неотложной помощи при закрытом переломе.

КОМПЛЕКТ ПРАКТИЧЕСКИХ (ЛАБОРАТОРНЫХ) РАБОТ
ПО ДИСЦИПЛИНЕ

_____ЭКОНОМИКА ОРГАНИЗАЦИИ_____
(наименование учебной дисциплины)

Г.о. Балашиха 2017г.

Введение

Настоящий сборник практических (лабораторных) работ предназначен в качестве методического пособия при проведении практических (лабораторных) работ по программе учебной дисциплины «Экономика организации» при подготовке квалифицированных рабочих и специалистов среднего звена.

Требования к знаниям и умениям при выполнении практических (лабораторных) работ

В результате выполнения практических (лабораторных) работ, предусмотренных программой по данной учебной дисциплине, проводится текущий контроль индивидуальных образовательных достижений.

Результаты обучения:

В результате изучения учебной дисциплины «Экономика организации» обучающийся должен:

уметь:

- воспринимать изменения в условиях производства, рыночной экономики и предпринимательства;
- находить и использовать необходимую экономическую информацию

знать:

- основы экономики, подходы к анализу экономической ситуации в стране и за рубежом, денежно-кредитную и налоговую политику;
- механизмы ценообразования на продукцию (услуги), формы оплаты труда в современных условиях;
- законодательство по охране авторских прав

использовать приобретенные знания и умения в практической деятельности и повседневной жизни:

- для получения и оценки экономической информации;
- составления семейного бюджета;
- оценки собственных экономических действий в качестве потребителя, члена семьи и гражданина.

Правила выполнения практических работ

1. Обучающийся должен выполнить практическую (лабораторную) работу в соответствии с полученным заданием.
2. Каждый обучающийся после выполнения работы должен представить отчет о проделанной работе с анализом полученных результатов и выводом по работе.
3. Отчет о проделанной работе следует выполнять в тетрадях для практических (лабораторных) работ.
4. Содержание отчета указано в описании практических (лабораторных) работы.
5. Таблицы и рисунки следует выполнять с помощью чертежных инструментов (линейки, циркуля и т. д.) карандашом.
6. Расчет следует проводить с точностью до двух значащих цифр.
7. Если обучающийся не выполнил практическую работу или часть работы, то он может выполнить работу или оставшуюся часть во внеурочное время, согласованное с преподавателем.
8. Оценку по практической работе обучающийся получает, с учетом срока выполнения работы, если:

-расчеты выполнены правильно и в полном объеме;

-сделан анализ проделанной работы и вывод по результатам работы;

-обучающийся может пояснить выполнение любого этапа работы;

-отчет выполнен в соответствии с требованиями к выполнению работы.

Зачет по лабораторным (практическим) работам обучающийся получает при условии выполнения всех предусмотренных программой работ, после сдачи отчетов по работам при получении удовлетворительных оценок.

Перечень лабораторных и практических работ

Лабораторная работа №1

«Расчет длительности производственного цикла»

Лабораторная работа №2

«Расчет численности рабочих по категориям»

Лабораторная работа №3

«Расчет показателей производительности труда»

Лабораторная работа № 4

«Расчет заработной платы различных категорий работников»

Лабораторная работа № 5

«Расчет показателей эффективности использования основного капитала»

Лабораторная работа № 6

«Расчет норматива оборотных средств и показателей эффективности использования материальных ресурсов»

Лабораторная работа № 7

«Расчет плановой прибыли предприятия»

Лабораторная работа № 8

«Расчет эффективности капитальных вложений»

Лабораторная работа № 9

«Порядок исчисления налогов задачи»

Лабораторная работа № 10

«Порядок исчисления налогов задачи»

Лабораторная работа № 1

Тема: «Расчет длительности производственного цикла»

Цель: Научиться рассчитывать длительность производственного цикла при последовательном, параллельном и последовательно-параллельном способе календарной организации процесса. Приобретение практических навыков определения длительности производственного цикла.

Ход работы:

Производственный цикл — это календарный период времени, в течение которого материал, заготовка или другой обрабатываемый предмет проходит все операции производственного процесса или определенной его части и превращается в готовую продукцию. Он выражается в календарных днях или при малой трудоемкости изделия — в часах.

Длительность производственного цикла в большой степени зависит от способа передачи детали (изделия) с операции на операцию. Существуют три вида движения детали (изделий) в процессе их изготовления:

-последовательный;

-параллельный;

- параллельно-последовательный.

Методические указания по решению задач:

Задача 1. Рассчитайте длительность операции штифтования, если нормативная трудоемкость штифтования составляет 30 ч, длительность рабочей смены 8 часов, коэффициент выполнения норм 0,95, на операции штифтования занято двое рабочих.

Решение.

По условию задачи t_o = 30ч, C = 2 рабочих, q = 8ч, $Кв$ = 0,95.

$$Тобл = \frac{t_o}{C \times Кв \times q} = \frac{30}{2 \times 0{,}95 \times 8} = 2$$ (дня)

Задача 2. Рассчитайте длительность совокупного цикла механической обработки партии из 20 деталей при последовательном, параллельном и последовательно-параллельном способе календарной организации процесса и следующих значениях плановой трудоемкости операций (в часах): 1) токарная – 6; 2) сверлильная – 1; 3) токарная – 2; 4) фрезерная – 1,5; 5) шлифовальная – 4.

Решение.

По условию задачи m = 5, n = 20, t_1 = 6; t_2 = 1; t_3 = 2; t_4 = 1,5; t_5 = 4.

$Тп = n \sum_{i=1}^{m} t_i = 20 \times (6+1+2+1,5+4) = 290$ (ч).

$Тпп = Тп - (n-1) \sum_{i=1}^{m} t_{наим} = n \sum_{i=1}^{m} t_i - (n-1) \sum_{i=1}^{m} t_{наим} =$

$= 20 \times (6+1+2+1,5+4) - (20-1) \times (1+1+1,5+1,5) = 195$ (ч)

$Тпр = (n-1)\, t_{наим} + \sum_{i=1}^{m} t_i = (20-1) \times 6 + (6+1+2+1,5+4) = 128,5$ (ч).

Контрольные вопросы:

1.Структура производственного цикла

2. Методы расчета производственного цикла

3. Пути и значение сокращения производственного цикла

Вывод

Лабораторная работа № 2

Тема «Расчет численности рабочих по категориям»

Цель: научиться планировать перспективную потребность в персонале и приобрести навыки оценки эффективности использования персонала.

Для достижения поставленной цели необходимо решить ряд задач:

Освоить порядок проведения сравнительного анализа персонала предприятия;

Научиться планировать численность работников предприятия различными методами;

Освоить методы определения уровня производительности труда на предприятии.

Ход работы:

Для проведения сравнительного анализа персонала используются такие показатели, как коэффициент оборота по приему, коэффициент оборота по выбытию, коэффициент восполнения численности работников и другие.

Задача 1

Постановка задачи:

Среднесписочная численность работников предприятия за год составила 1 000 чел. По трудовому договору в этом году на предприятие зачислено 200 чел. Рассчитать коэффициент оборота по приему.

Технология решения задачи:

Коэффициент оборота по приему рассчитывается как отношение количества работников, принятых на работу за данный период $Ч_п$, к среднесписочной численности работников $\overline{Ч}$ за тот же период:

$$k_{пр} = \frac{Ч_п}{\overline{Ч}} = \frac{200}{1000} = 0,2. \quad (1)$$

Ответ: *коэффициент оборота по приему 0,2.*

Задача 2

Постановка задачи:

Среднесписочная численность работников предприятия за год составила 1 000 чел. В течение года уволилось по собственному желанию 75 чел., поступило в учебные заведения 15 чел., ушло на пенсию 25 чел. Рассчитать коэффициент оборота по выбытию.

Технология решения задачи:

Коэффициент оборота по выбытию рассчитывается как отношение количества выбывших работников за данный период $Ч_{выб}$, к среднесписочной численности работников $\overline{Ч}$ за тот же период:

$$k_{выб} = \frac{Ч_{выб}}{\overline{Ч}} = \frac{115}{1000} = 0,115. \quad (2)$$

Ответ: *коэффициент оборота по выбытию 0,115.*

Контрольные вопросы

Дайте определение понятиям: «трудовые ресурсы предприятия», «персонал предприятия», «кадры предприятия».

Назовите основные категории структуры персонала предприятия.

Дайте определение понятия «производительность труда». Какие показатели используются для ее измерения?

Вывод

Лабораторная работа № 3

Тема «Расчет показателей производительности труда»

Цель: Овладеть навыками в экономических расчетах и их анализа. Научится отыскивать пути улучшения использования производственных фондов объединения.

Ход работы:

2.1 Краткие теоретические сведения:

Производственные фонды состоят из основных и оборотных средств. Амортизацией называется денежное возмещение износа основных средств. Расчет амортизации выполняется с помощью следующей формулы:

$$A_{ГОД} = \frac{(Ф1 * 12 + фвв * п1 - Фвыв * п2) * На}{100 * 12}$$

где Ф 1- стоимость основных средств на начало года, руб.,

Ф вв- стоимость основных средств, введенных в эксплуатацию, руб.,

Ф выв- стоимость основных средств, выведенных из эксплуатации, руб**.,**

nl- количество месяцев работы основных средств с момента ввода их

в эксплуатацию до конца года; п2- количество месяцев с момента вывода основных средств из

эксплуатации до конца года; .

На - норма амортизации, %

Показателями использования основных средств на предприятии являются следующие:

фондоотдача -(Фо)

$$Ф_О = \frac{ВП}{С_п} \rightarrow \max$$

где ВП – выпуск продукции на предприятии (цехе, участке) за год, шт.

фондоемкость (Фе)

$$Ф_е = \frac{1}{Ф_о} \rightarrow \min$$

Фондовооруженность (Фв)

$$Фв = \frac{С_п}{Р} \rightarrow \max$$

где Р - число работающих, чел.

Коэффициент сменности (Ксм)

$$К_{см} = \frac{Т_{ч.ф}}{Т_{ч.м}}$$

где $Т_{ч}ф$- фактическое число отработанных станко - часов (машино-часов) в исследуемый период времени

$Т_{чм}$- максимально возможное число станко - часов (машино-часов) при работе в одну смену.

Пример решения:

Задача 1 Стоимость основных средств предприятия на начало года составляет 65000 тыс. руб. С 1 марта введены в эксплуатацию новые средства на 500 тыс. руб.; с 1 июля выведены

из эксплуатации старые средства на 800 тыс. руб. Общая среднегодовая норма амортизации - 10%. Определить сумму амортизации.

Решение:

Общая годовая сумма амортизационных отчислений:

$$A = \frac{(C_{нач} * 12 + C_{вв} * 10 - C_{выв} * 7) * H_a}{100 * 12}$$

$$A = \frac{(65000 * 12 + 500 * 10 - 800 * 7) * 10}{100 * 12} = 6588.3 тыс. руб$$

Ответ: A =6588,3 тыс. руб.

Задача 2. Два предприятия производят одинаковую продукцию, их работа за год характеризуется следующими данными:

Показатели	Варианты	
	Первый	Второй
Среднегодовая стоимость основных средств, тыс. руб.	80000	140000
Численность работающих, чел.	400	500
Выработка продукции на одного работающего, руб	3000	3600

Определить показатели фондоотдачи и фондовооруженности по первому и второму вариантам.

Решение.

Фондовооруженность:

$$V = \frac{C_{оф}}{P}$$

$$V_1 = \frac{80000000}{400} = 200000 руб$$

$$V_2 = \frac{140000000}{500} = 280000 руб$$

Объем продукции:

$$B = П_{тр} * P$$

$$B_1 = 3000 * 400 = 1200000 руб$$

$$B_2 = 3600 * 500 = 1800000 руб$$

Фондоотдача:

$$Ф = \frac{B}{C_{оф}}$$

$$Ф_1 = \frac{1200000}{80000000} = 1,5 руб$$

$$Ф_2 = \frac{1800000}{140000000} = 1,28 руб$$

Контрольные вопросы

Дайте определение понятиям: «трудовые ресурсы предприятия», «персонал предприятия», «кадры предприятия».

Назовите основные категории структуры персонала предприятия.

Дайте определение понятия «производительность труда». Какие показатели используются для ее измерения?

Вывод

Лабораторная работа № 4

Тема «Расчет заработной платы различных категорий работников»

Цель: усвоение методики расчета заработка при сдельной, повременной, бестарифной системе оплаты труда, среднесписочной численности работников.

Ход работы:

Заработная плата является формой вознаграждения за труд и важным стимулом работников предприятия, поскольку выполняет воспроизводственную и стимулирующую (мотивационную) функции. Различают номинальную и реальную заработную плату. *Номинальная заработная плата* – это начисленная и полученная работником заработная плата за его труд за определенный период. *Реальная заработная плата* – это количество товаров и услуг, которые можно приобрести за номинальную заработную плату, т.е. реальная заработная плата – это «покупательная способность» номинальной заработной платы.

В современных условиях на предприятиях применяются различные формы и системы оплаты труда, но наибольшее распространение получили две формы оплаты труда: повременная и сдельная.

Повременной называется такая форма оплаты труда, при которой заработная плата работникам начисляется по установленной тарифной ставке или окладу за фактически отработанное на производстве время

$$Зп = Тарифная\ ставка \times Фактически\ отработанное\ время, \quad (24)$$

При простой повременной системе заработная плата работника (ЗПпп) за определенный отрезок времени рассчитывается как

$$ЗПпп = ТС \times Т, \quad (25)$$

где m – часовая (дневная) тарифная ставка рабочего соответствующего разряда, руб;

Т – фактически отработанное на производстве время, ч. (дни).

Сдельная форма. При сдельной форме оплаты труда заработная плата работникам начисляется по заранее установленным расценкам за каждую единицу выполненной работы или изготовленной продукции, т.е. это оплата труда за количество произведенной продукции. Сдельная заработная плата рассчитывается по формуле

$$ЗП = СР \times V, \quad (26)$$

где СР – сдельная расценка за единицу продукции;

V – количество изготовленной продукции.

При косвенно-сдельной системе заработной платы определяются косвенно-сдельные расценки (Рк.с)

$$Ркс = ТС / В, \quad (27)$$

где ТС - тарифная ставка вспомогательного рабочего

В - норма выработки основных рабочих, обслуживаемых данным вспомогательным рабочим

Пример решения типовой задачи

Рассчитать заработок рабочего сдельщика за месяц, если норма выработки за смену 0,7 тонны продукции. Дневная тарифная ставка 192,8 руб. Премия за месяц составляет 30% от сдельного заработка. За месяц рабочим выпущено продукции 15,2 т.

Решение

1. Определяем дневную заработную плату работника - сдельщика

$$ЗПдн = СР \times V = 192,8 \times 0,7 = 134,96 \text{ руб.}$$

2. Определяем месячную заработную плату

$$ЗП\ мес = ЗПдн \times Vмес + Премия = 134,96 \times 15,2 + 30\% = 2667 \text{ руб.}$$

Задача 1

Рассчитать заработок рабочего - сдельщика за месяц, если норма выработки за смену 0,5 тонны продукции. Дневная тарифная ставка 146,4 руб., премия составляет 30% от сдельного заработка. За месяц рабочим выпущено продукции 16,0 тонн.

Задача 2

Рассчитать месячную заработную плату рабочего - сдельщика, если норма выработки за смену 1,3 тонны продукции. Дневная тарифная ставка 214,4 руб., премия за месяц составляет 30% от сдельного заработка. Выпущено за месяц 20 тонн продукции.

Контрольные вопросы

В чем суть понятия «персонал предприятия».

2. Какова структура персонала предприятия.

3. Охарактеризуйте промышленно-производственный персонал предприятия.

Выводы:

Лабораторная работа № 5

Тема «Расчет показателей эффективности использования основного капитала»

Цель: усвоение методики расчета структуры оборотных средств, показателей, оборачиваемости оборотных средств, норматива оборотных средств, суммы высвобожденных оборотных средств вследствие ускорения их оборачиваемости.

Ход работы:

Под *оборотными фондами* понимается часть средств производства, которые участвуют в производственном процессе и полностью переносят свою стоимость на производимый продукт. Оборотные фонды (средства) определяются как совокупность оборотных производственных фондов и фондов обращения. Оборотные производственные фонды — это предметы труда, находящиеся в сфере производства. К ним относятся сырье, основные и вспомогательные материалы, не законченная производством продукция и другие предметы труда, которые, в отличие от основных, целиком потребляются в каждом производственном цикле. Оборотные средства находятся одновременно на всех стадиях и во всех формах производства, что обеспечивает его непрерывность и бесперебойную работу предприятия.

Продолжительность одного оборота определяется суммой времени производства и времени обращения. От продолжительности оборота средств в сферах производства и обращения во многом зависит величина оборотных средств, необходимых для производственно-хозяйственной деятельности предприятий. Чем быстрее средства будут переходить из одной формы в другую, тем скорее будет происходить оборот, тем меньше будет общая сумма оборотных средств.

Важнейшими показателями использования оборотных средств на предприятии являются коэффициент оборота оборотных средств, коэффициент загрузки и длительность одного оборота в днях.

1. Коэффициент оборачиваемости

$$\text{Ко} = \frac{\text{РП}}{\text{ОС}}, \tag{13}$$

где Р - стоимость реализации продукции за год, тыс. руб;

ОС - среднегодовая сумма оборотных средств;

Ко - коэффициент оборачиваемости.

2. Скорость оборота оборотных средств Д, дни

$$\text{Д} = \frac{360}{\text{Ко}}, \tag{14}$$

где 360 - количество дней в финансовом году.

3. Сумма высвободившихся оборотных средств ВС, тыс. руб., определяется по формуле

$$\text{ВС} = \frac{\text{РП}}{360} \times (\text{Д0} - \text{Д1}), \tag{15}$$

Норматив оборотных средств определяется умножением суточного расхода данного вида оборотных средств в днях.

Суточный расход оборотных средств определяется делением годовой потребности в оборотных средствах на 360 (число дней в финансовом году).

Норматив оборотных средств для предприятия есть сумма оборотных средств по элементам.

Пример решения типовой задачи

Задача

Стоимость реализуемой продукции по годовому плану завода 3200 т.р. Средний остаток 800 т.р. В результате проведенных организационно-технических мероприятий фактическая деятельность одного оборота доведена до 70 дней. Определить коэффициент оборачиваемости, длительность одного оборота до проведения мероприятий и сумму высвободившихся оборотных средств.

Решение

1. Определяем коэффициент оборачиваемости

$$\text{Ко} = \frac{\text{РП}}{\text{ОС}} = \frac{3200}{800} = 4\text{об}$$

2. Определяем скорость оборота оборотных средств

$$\text{Д} = \frac{360}{\text{Ко}} = \frac{360}{4} = 90 \text{ дней}$$

3. Определяем сумму высвободившихся оборотных средств

$$\text{ВС} = \frac{\text{РП}}{360}(\text{Д}_0 - \text{Д}_1) = \frac{3200}{360} \times (90 - 70) = 177{,}8 \text{ тыс.руб.}$$

Задача 1.

В текущем году заводу установили план реализации продукции на год в сумме 20000 т.р., норматив собственных оборотных средств определен в сумме 2000 т.р. Фактически при тех же оборотных средствах за счет ускорения их оборачиваемости выпущено продукции на 24 т.р. Определить эффективность ускорения оборачиваемости оборотных средств, сумму высвободившихся средств.

Задача 2

Предприятие реализовало за год продукции на 40 млн. руб. Среднегодовой остаток оборотных средств 10 млн. руб. Определить коэффициент оборачиваемости оборотных средств, продолжительность их оборота.

Контрольные вопросы

Дайте определение оборотным средствам предприятия.

2. Раскройте состав и структуру оборотных средств предприятия.
3. Что такое кругооборот оборотных средств?
4. Какие пути ускорения оборачиваемости оборотных средств вам известны.

Выводы

Лабораторная работа № 6

Тема «Расчет норматива оборотных средств и показателей эффективности использования материальных ресурсов»

Цель: усвоить методику расчета мощности и выпуска продукции и коэффициентов использования мощности, финансовой устойчивости, ликвидности, деловой активности.

Ход работы:

План производства и реализации продукции – центральный раздел стратегического и текущего планов; должен обеспечить рост выпуска продукции, повышение ее качества, полное удовлетворение спроса потребителей, возможно более полное использование сырья и производственной мощности предприятия.

В плане определяется сколько и какой продукции надо производить.

В производственном плане рассчитываются:

1) объем производства отдельных видов продукции в натуральном выражении:

- *готовая продукция* (прошедшая все стадии обработки, отвечающая ГОСТам, техническим условиям и международным стандартам)
- *полуфабрикат* (продукция отдельных технологических установок, прошедшая часть стадии обработки и требующая дальнейшей обработки)
- *незавершенное производство* (находится в стадии обработки, продукт который не прошел все стадии обработки в пределах цеха или предприятия; то что внутри ТУ)
- *продукция вспомогательных цехов* (электроэнергия, пар, вода, отпущенная на сторону и собственные нужды)

2) объем производства отдельных видов продукции в стоимостном выражении:

- *товарная продукция* (стоимость готовых изделий, выработанных в планируемом году + стоимость полуфабрикатов своей выработки, предназначенные к отпуску на сторону + стоимость работ и услуг сторонним организациям)

$$ТП = \sum Q*P, \quad (42)$$

- *валовая продукция* (стоимость товарной продукции и изменение остатков незавершенного производства и полуфабрикатов собственной выработки)

$$ВП = ТП + \Delta ПФ = ТП – ПФ_{НГ} + ПФ_{КГ,} \quad (43)$$

- *реализованная продукция* (товарная продукция, которая отгружена и подлежит оплате)

$$РП = ТП + \Delta ГП = ТП – ГП_{КГ} + ГП_{НК,} \quad (44)$$

Остатки ГП – то, что находится на складе или в пути или их получили, но не оплатили; не подтверждено документально.

3) показатели производственной мощности предприятия

Производственная мощность – максимально возможный объем выпуска продукции в номенклатуре и ассортименте, установленных планом, при наиболее полном использовании оборудования и площадей.

- Мвх - входящая мощность (мощность на начало планового года)
- Ми - исходящая мощность (мощность на конец года с учетом намеченных плановых реконструкций оборудования, его пополнения, изменения технологии производства)
- $М_{мод}$ – модернизированные мощности
- $М_{рек}$ – реконструированные мощности
- $М_{лик}$ – ликвидированные мощности

$$М_И = М_{ВХ} + М_{РЕК} + М_{МОД} - М_{ЛИК,} \quad (45)$$

- среднегодовая мощность

$$М_{СР} = М_{ВХ} + М_{РЕК}*Т_Р/12 + М_{МОД}*Т_М/12 - М_{ЛИК}(12-Т_Л)/12, \quad (46)$$

Пример решения типовой задачи

1. Определяем выработку по кварталам и в целом за год

По изделию А
1 кв.: гр. 10 = гр. 3 × гр. 5 = 4,97 × 47 = 233,59
2 кв.: гр. 11 = гр. 3 × гр. 6 = 4,97 × 85 = 422,45
3 кв.: гр. 12 = гр. 3 × гр. 7 = 4,97 × 85 = 422,45
4 кв.: гр. 14 = гр. 3 × гр. 8 = 4,97 × 85 = 422,45
Итого за год: гр. 9 = гр. 3 × 4,97 × 302 = 1500,94

2. Определяем производственную мощность на год и по кварталам

По изделию А
1 кв.: гр. 14 = гр. 2 × гр. 5 = 14,5 × 47 = 681,5
2 кв.: гр. 14 = гр. 2 × гр. 6 = 14,5 × 85 = 1232,5
3 кв.: гр. 14 = гр. 2 × гр. 7 = 14,5 × 85 = 1232,5
4 кв.: гр. 14 = гр. 2 × гр. 8 =14,5 × 85 = 1232,5
За год: гр. 14 = гр. 2 × гр. 4 = 14,5 × 302 = 4379

3. Годовая мощность по изделию А определяется как:

$$Мг = В \times Д, \quad (47)$$

$$Мг = 14,5 \times 302 = 4379$$

4. Общий коэффициент использования мощности определяется по изделию А следующим образом:

$$Ки = ВП / Мг = 1500,94 / 4379 = 0,34.$$

По остальным изделиям анализ проводится аналогично.

Наименование изделия	Выработка в сутки в тоннах		Количество дней работы					Выпуск продукции, в тыс.					Годовая производ. мощ-ность	Общий коэффиц. использ. мощности
	выработка по мощности	выработка по плану	всего	в том числе по кв.				всего	в том числе по кв.					
				1	2	3	4		1	2	3	4		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
А	14,5	4,97	302	47	85	85	85	1500,94	233,59	442,45	442,45	442,45	4379	0,34
Б	14,2	13,48	341	86	85	85	85							
В	14,8	14,0	341	86	85	85	85							
ИТОГО														

Задача 1

Рассчитать план производства и использования производственных мощностей по следующим данным

Наименование изделия	Выработка в сутки в тоннах		Количество дней работы					Выпуск продукции, в тыс.					Годовая производ. мощ-ность	Общий коэффиц. использ. мощности
	выработка по мощности	выработка по плану	всего	в том числе по кв.				всего	в том числе по кв.					
				1	2	3	4		1	2	3	4		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Вариант 1														
Йогурт 0%	7,3	6,8	302	47	85	85	85							
Сметана 30%	9,8	9,0	341	86	85	85	85							
Молоко 3,5%	12,0	11,2	341	86	85	85	85							
Вариант 2														
Сырок глазированный с курагой	12,0	11,4	302	47	85	85	85							
Масса творожная с изюмом	12,4	11,6	341	86	85	85	85							
Ряженка	14,5	13,8	341	86	85	85	85							

Контрольные вопросы

1. Что такое мощность.
2. Каковы показатели использования мощности.
3. Расчет среднегодовой мощности.

Выводы

Лабораторная работа № 7

Тема «Расчет плановой прибыли предприятия»

Цель: закрепление знаний, умений и приобретение практических навыков по решению задач.

Ход работы:

Прибыль представляет собой денежное выражение части чистого дохода общества, который создается на предприятиях в результате их производственной деятельности и используется государством и предприятиями для развития производства и повышения жизненного уровня населения.

Основная формула прибыли:

$$\boldsymbol{П = Д - U \text{ руб.},}$$

где Д – доход = выручка в оптовых ценах, руб;
U – издержки (себестоимость), руб.

Прибыль выполняет три основные функции:

1. выступает в качестве одного из важнейших показателей экономической эффективности;
2. используется в качестве оружия распределения части чистого дохода между коллективом предприятия и общества в целом;
3. является источником формирования поощрительных фондов.

Общая сумма прибыли составляет ***балансовую прибыль****, обозначается* $\boldsymbol{П_б}$:

$$\boldsymbol{П_б = П_р + П_{др} - (+) П_в \text{ руб.},}$$

где $П_р$ – прибыль от реализации продукции основной деятельности , руб.
$П_{др}$ – прибыль от выполнения других работ и оказания услуг на сторону.
$П_в$ – прибыль (убытки) от внереализационных операций.

Наибольшая часть общей прибыли составляет ***прибыль от реализации продукции:***

$$\boldsymbol{П_р = В - С \text{ руб;}}$$

$$\boldsymbol{П_р = (Ц_i - Ci) * Q_{год\ I} \text{, руб.},}$$

где В – выручка, руб;
С – себестоимость, руб;
Цi – оптовая цена единицы продукции i – вида в рублях
Сi – полная себестоимость единицы продукции i – вида, руб.
$Q_{год}$ = объём реализуемой продукции i-вида в натуральных показателях
n – число выпускаемых видов продукции, ед.

$$\boldsymbol{Q_{год}i = Q_{нг} + Q_{но} + Q_{пл} - Q_{кг} - Q_{н.пл} \text{, т.},}$$

где $Q_{нг}$ – остаток готовой продукции на складе на начало года, т;
$Q_{но}$ – продукция отгруженная, но неоплаченная покупателем на начало года, т;
$Q_{пл}$ – плановый объём выпуска продукции, т;

$Q_{кг}$ – планируемый остаток готовой продукции на складе на конец года, т;

$Q_{н.пл.}$ – планируемый объём отгруженной но не оплаченной товарной продукции на конец года, т.

$П_{др}$ – прибыль от выполнения других работ и оказания услуг на сторону складывается из прибыли от реализации изделий, работ, услуг подсобных и обслуживающих производств (например, автохозяйств) состоящих на балансе у предприятия, а так же прибыли от реализации электроэнергии, газа, пара, сжатого воздуха.

Расчет прибыли **$П_{др}$** производится также как от реализации продукции основной деятельности.

Прибыль или убытки от внереализационных операций **$П_{в}$** включают: полученные (уплаченные) штрафы, пени, неустойки, убытки от содержания объектов ЖКХ, поступление или списание безнадёжных долгов.

Из прибыли предприятия вносят в госбюджет плату за основные фонды, оборотные средства рентные (фиксированные), осуществляют выплату % за банковский кредит, а оставшаяся часть прибыли, называемая расчётной – остаётся в распоряжении предприятия, и используется на создание фондов экономического стимулирования.

$$П_{расч} = П_{б} - Р_{ф} - Р_{р} - Р_{к}, руб.,$$

где $Р_{ф}$ – сумма платы за производственные фонды, руб;

$Р_{р}$ – фиксированные платежи, руб;

$Р_{к}$ – сумма на выплату %% за банковский кредит, руб.

К внутренним источникам относятся: чистая прибыль, амортизационные отчисления, фонд переоценки имущества в аренду и прочие источники.

К внешним источникам относятся: выпуск акций, безвозмездная финансовая помощь, прочие внешние источники.

Балансовая прибыль распространяется в строго установленном порядке, в первую очередь – вносится плата в бюджетные и внебюджетные фонды, %% за использование кредита.

Остаток прибыли направляются на образование фондов экономического стимулирования:

- ФМП – фонд материального поощрения,
- ФРП – фонд развития производства,
- ФСКМ – фонд социально-культурных мероприятий.

Сверхплановая прибыль образуется за счёт перевыполнения плана по реализации продукции, снижение её себестоимости и распределяется аналогично.

Основой для улучшения экономических показателей и финансового положения предприятия являются мероприятия по дальнейшему развитию и совершенствованию производства, росту ПТ, экономия расходования материальных ресурсов и денежных средств.

Рентабельность предприятия

Рентабельность показывает уровень использования производственных фондов, с помощью которых получена прибыль.

Рентабельность определяется отношением прибыли к затраченным ресурсам. Чем выше уровень рентабельности, тем больше сумма отчислений от прибыли. В хозяйственной практике рентабельность исчисляют по отношению к стоимости производственных фондов и себестоимости.

На предприятиях определяют общую и расчетную рентабельность:

$$R_{общ} = \frac{П_{бал}}{C_{срОПФ} + C_{обСр}} \times 100\%$$ (% дописывается но не выводится на калькулятор).

Общая рентабельность отражает уровень производственных фондов.

$$R_{расч} = \frac{П_{расч}}{C_{срОПФ} + C_{обСр} - Ф_{л}} \times 100\%,$$

где $Ф_{л}$ – льготные основные фонды, освобождённые от платы за их использование, (руб.)

Расчётная рентабельность показывает размер прибыли, остающейся в распоряжении предприятия на 1 рубль производственных фондов.

От платы освобождается:

- ОПФ созданные за счет кредита банка – до полного погашения ссуды;
- Вновь введенные ОПФ – на срок освоения ОФ;
- Законсервированные по решению совета министров;
- ОПФ средозащитного назначения, а так же обеспечивающие охрану труда и промышленную санитарию.

Для оценки эффективности издержек предприятия на производство и реализацию продукции, используется показатель рентабельности продукции:

$$R_{пр} = \frac{прибыль}{себестоимость}$$

Основные направления повышения рентабельности:

- Увеличение объёма производства;
- Реализация продукции;
- Повышение качества выпускаемой продукции;
- Снижение себестоимости за счет внедрения новой техники и технологии;
- Сокращение времени транспортирования выпускаемой продукции до потребителя;
- Ускорение расчетов за отгруженную продукцию;
- Снижение запасов товарно-материальных ценностей на складе.

1. Работа в аудитории

Задание

Студент самостоятельно:

- изучает методические рекомендации по проведению практической работы;
- производит решение задачи с пояснениями;
- отвечает на контрольные вопросы практической работы;
- оформляет отчет.

Содержание отчета:

1. Номер ПР.
2. Название ПР.
3. Цель работы.
4. Решение задач с пояснениями и выводом.
5. Ответы на вопросы.

Задание:

За год шахтой было реализовано 750000 т. угля. Средняя оптовая цена 1 т угля 16,6 руб. Себестоимость реализованного угля 9850000 руб.
Определить: 1) годовую прибыль шахты от реализации угля;
2)прибыль от реализации 1 т угля.
Методические рекомендации:
1.Для расчета годовой прибыли шахты от реализации угля необходимо стоимость 1 т угля умножить на годовой объем угля реализованного шахтой и отнять сумму себестоимости реализованного угля за год:

$$\boldsymbol{П_р = В – С \ руб;}$$

$$П_р = (16,6 * 750000) – 9850000 = 2600000 \text{ руб.}$$

2.Для расчета прибыли от реализации 1 т угля возьмем сумму годовой прибыли от реализации угля разделим на общий объем реализации угля в тоннах:

$$\mathbf{П_{1т} = 2600000 / 750000} = 3,46 \text{ руб.}$$

3. В заключение решения задачи следует проанализировать стоимость единицы продукции, возможные факторы, которые влияют на стоимость единицы продукции, написать ответ.

Задание:
Среднегодовая стоимость ОПФ шахты 11500000 руб. Среднегодовая стоимость нормируемых оборотных средств 550000 руб. Годовая балансовая прибыль 1220000 руб.
Определить общую рентабельность производства.
Методические рекомендации:
1.Для определения общей рентабельности производства воспользуемся формулой:

$$R_{общ} = \frac{П_{бал}}{C_{срОПФ} + C_{обСр}} \times 100\%$$

$$R_{общ} = 1220000 / 11500000 + 550000 * 100\% = 10,12 \%.$$

2.В заключение решения задачи следует проанализировать процент общей рентабельности производства, возможные факторы, которые повлияли этот процент общей рентабельности производства, написать ответ.

Контрольные вопросы

Что такое балансовая прибыль?

2.Как рассчитывается балансовая прибыль предприятия на планируемый год?

3. Как рассчитывается расчетная прибыль предприятия, почему она меньше балансовой прибыли?

4. Как рассчитывается общая и расчетная рентабельность производства на планируемый период?

Вывод

Лабораторная работа № 8

Тема «Расчет эффективности капитальных вложений»

Цель: закрепление знаний, умений и приобретение практических навыков по решению задач.

Ход работы:

Расчеты экономической эффективности – обязательная стадия принятия решения о капитальных вложениях в производство и при внедрении новой техники.

Капитальные вложения – совокупность затрат на строительство новых, реконструкцию и расширение действующих предприятий, на приобретение новой техники.

При определении экономических последствий капитальных вложений и новой техники используют **показатели:**

- **экономического эффекта**,
- **общей экономической эффективности**,
- **сравнительной экономической эффективности затрат.**

Экономический эффект - это величина результата, для получения которого осуществляют капитальные вложения, внедряются новая техника.

Экономический эффект может быть измерен в натуральных показателях (например, вложения в реконструкцию рудника увеличат его мощность на 1000000м^3 горной массы) и стоимостных показателей (под стоимостным выражением подразумевается **сумма экономии за** счет внедрения новшеств).

При определении **общей экономической эффективности КВ** применяются следующие показатели:

а) по предприятиям при использовании собственных средств и кредитов

$$Э_{П} = \text{дельта}\, П \;/\; КВ$$

где дельта П – прирост годовой прибыли или экономический эффект, руб,

КВ – капитальные вложения, вызвавшие прирост прибыли, руб.

б) по вновь строящимся предприятиям

$$Э = (Ц - С) / КВ = П / КВ$$

где Ц – годовой выпуск продукции по плану в оптовых ценах предприятия, руб.

С – себестоимость годового выпуска продукции, руб..

в) по предприятиям, где применяются показатели снижения себестоимости продукции, а также по планово- убыточным предприятиям

$$Э = (C_1 - C_2) / КВ$$

где $C_1 - C_2$ – себестоимость годового выпуска продукции до и после проведения мероприятия, руб.,

КВ - капитальные вложения, вложенные в мероприятия, руб..

Полученные показатели общей экономической эффективности сравниваются с нормативным коэффициентом эффективности

$$E_н = 0{,}15 \quad \text{должно соблюдаться условие} \quad Э > E_н.$$

При расчетах общей экономической эффективности определяются **сроки** окупаемости общих объемов Капитальных вложений - Т:

а) $\mathbf{Т_п = КВ \ / _{дельта} П,\ год}$

б) $\mathbf{Т = КВ \ / \ Ц - С,\ год}$

в) $\mathbf{Т = КВ \ / C_1 - C_2,\ год.}$

Сравнительная экономическая эффективность рассчитывается при выборе одного из нескольких проектных решений и показывает насколько один вариант эффективнее другого.

Рассчитывается коэффициент сравнительной экономической эффективности:

$$\mathbf{Е = C_1 - C_2 \ / \ КВ_2 - КВ_1 > Е_н.}$$

Рассчитывается срок окупаемости сравнительной экономической эффективности:

$$\mathbf{Т = КВ_2 - КВ_1 \ / \ C_1 - C_2 = 1 / Е}.$$

При расчете коэффициента сравнительной экономической эффективности применяется формула « Формула приведенных затрат»:

$$\mathbf{З = С + Е_Н * КВ \sim\sim MIN}, \text{ руб},$$

где З – приведенные затраты единицы продукции, руб.,

С – себестоимость единицы продукции, руб.,

КВ – капитальные вложения на единицу продукции, руб.,

$Е_н$ – нормативный коэффициент эффективности капитальных вложений,

вариант для которого $З_{min}$ признается оптимальным.

Задание:

Объем капитальных вложений по объектам производственного назначения в развитие одной из подотраслей горнодобывающей промышленности составит 5400000000 руб. Прирост прибыли в рассматриваемом периоде по плану должен составить 930000000 руб.

Определить: 1) общую экономическую эффективность капитальных вложений; 2) отклонения общей эффективности капитальных вложений от норматива; 3) срок окупаемости общих капитальных вложений.

Методические рекомендации:

1. Для расчета общей экономической эффективности воспользуемся формулой:

   $$\mathbf{Э_П = _{дельта} П \ / \ КВ}$$

   $$Э_П = 930000000 / 5400000000 = 0,17$$

2. Отклонения общей эффективности капитальных вложений от норматива составляют:

   $$0,17 - 0,15 = 0,02$$

3. Срок окупаемости рассчитываем по формуле:

   $$\mathbf{Т_п = КВ \ / _{дельта} П,\ год}$$

   $$Т_п = 5400000000 / 930000000 = 5,8 \text{ года}.$$

4. В заключении решения задачи следует полученное показание общей экономической эффективности подставить в неравенство:

должно соблюдаться условие $Э > E_{н.}$ $0,17 > 0,15.$
Написать ответ.

Задание:
Предлагается три варианта размещения нового рудника по добыче руд цветных металлов с одинаковой годовой производственной мощностью. **Первый** вариант предусматривает строительство рудника в освоенном районе и не связан с необходимостью дополнительных капитальных вложений в развитие транспортной сети. Осуществление **второго** варианта требует дополнительных вложений в расширение портовых сооружений, а **третий** - в строительство железной дороги. Экономические показатели по вариантам (млн.руб.) приведены в таблице 1. Нормативный коэффициент эффективности капитальных вложений, связанных с расширением портовых сооружений и строительством железной дороги, равен 0,12.
Определить наиболее целесообразный вариант размещения рудника.
Таблица 1 – Экономические показатели по вариантам

Показатели	Вариант		
	1	2	3
Капитальные вложения в строительство рудника	21	21	23
Годовые эксплуатационные расходы на добычу руды	6,1	4,1	3,7
Капитальные вложения в расширение портовых сооружений	-	2	-
Капитальные вложения в строительство железной дороги	-	-	5

Методические рекомендации:

1) Рассчитаем приведенные затраты по первому варианту, используя «формулу приведенных затрат»:

$$\mathbf{З = С + E_н * КВ \sim\sim MIN}, \text{ руб}$$
$$З_1 = 6,1 + 0,15 * 21 = 9,25 \text{ млн.руб.}$$

2) Приведенные затраты по второму варианту:

$$З_2 = 4,1 + (0,15 * 21 + 0,12 * 2) = 7,49 \text{ млн.руб.}$$

3) Приведенные затраты по третьему варианту:

$$З_3 = 3,7 + (0,15 * 23 + 0,12 * 5) = 7,75 \text{ млн.руб.}$$

4) В заключение решения задачи следует написать ответ: наиболее целесообразным является второй вариант размещения рудника, так как при этом варианте минимальные приведенные затраты.

Контрольные вопросы

1.Раскройте сущность и задачи капитальных вложений?
2.Как определяются задания по вводу в действие основных фондов?
3. Как устанавливается целесообразность капитальных вложений?
4. Показатели общей экономической эффективности капитальных вложений?

Выводы

Лабораторная работа № 9
Тема «Порядок исчисления налогов задачи»

Цель: научиться: руководствоваться положениями гл.31 НК РФ; рассчитывать сумму налога на землю.

Ход работы:

Решение задачи по расчету НДФЛ и имущественного вычета

Семенов С. В. продал дом с прилегающим земельным участком и гараж. Доход, полученный в результате продажи дома, составил 2 400 000 рублей. Продажа гаража принесла 170 000 руб.

Сделки по продаже недвижимости правильно оформлены, период владения подтвержден документально, однако отсутствуют документы, подтверждающие фактические расходы на приобретение продаваемых объектов.

Требуется рассчитать размер вычетов, базу налогообложения и сумму НДФЛ, если:

1. объекты недвижимости находились в собственности Семенова С. В. более 3 лет;
2. указанным имуществом Семенов С. В. владел 2 года.

Решение:

1. Если продаваемый объект недвижимости находился в собственности плательщика НДФЛ более 3 лет (а для объектов, полученных в собственность с 2016 года, — более 5 лет), то на полученный от его продажи доход налог не начисляется (п. 17.1 ст. 217 НК РФ, п. 4 ст. 229 НК РФ). Таким образом, НДФЛ по первому варианту нашей задачи определять нет необходимости.

2. По условиям второго варианта задачи общий доход Семенова С. В. составит: 2 400 000 + 170 000 = 2 570 000 рублей.

Сумма имущественного налогового вычета, согласно подп. 1 п. 2 ст. 220 НК РФ, будет равна 1 000 000 + 170 000 = 1 170 000 рублей.

НДФЛ с доходов от продажи имущества, которым Семенов С. В. владел 2 года, будет равен:

(2 570 000 – 1 170 000) × 0,13 = 1 400 000 × 0,13 = 182 000 рублей.

Контрольные вопросы

1.Экономическое содержание и назначение налогов в государстве.

2.Роль налогов в воспроизводственном процессе, распределении и перераспределении национального дохода и внутреннего валового продукта.

3.Определение налога. Признаки налогов, их характеристика. Отличие налогов от сборов и платежей.

4.Функции налогов, их взаимосвязь.

Выводы

Лабораторная работа № 10
Тема «Порядок исчисления налогов задачи»

Цель: научиться: руководствоваться положениями гл.31 НК РФ; рассчитывать сумму налога на землю.

Ход работы:

Задача по НДФЛ с расчетом стандартного вычета

Работница организации Ковалева А. Н., до февраля 2016 года не состоящая в зарегистрированном браке (вдова), содержит 12-летнего ребенка. Ежемесячный доход сотрудницы за период с января по май 2016 года составлял 20 000 рублей. 17 февраля 2016 года был зарегистрирован повторный брак сотрудницы. Требуется рассчитать сумму налога на доходы физлица за указанный период.

Решение:

До момента регистрации повторного брака сотрудница имела право на двойной налоговый вычет на ребенка. С марта 2016 года это правило перестало действовать (подп. 4 п. 1 ст. 218 НК РФ), и вычет на ребенка Ковалевой А. Н. далее должен предоставляться в одинарном размере.

Сумму налога за период с января по май 2016 года следует рассчитывать так:

- НДФЛ за январь = (20 000 – (1 400 × 2)) × 0,13 = 2 236 рублей.
- НДФЛ за февраль = (20 000 – (1 400 × 2)) × 0,13 = 2 236 рублей.
- НДФЛ за март = (20 000 – 1 400) × 0,13 = 2 418 рублей.
- НДФЛ за апрель = (20 000 – 1 400) × 0,13 = 2 418 рублей.
- НДФЛ за май = (20 000 – 1 400) × 0,13 = 2 418 рублей.

А всего за указанный период сумма начисленного налога с дохода Ковалевой А. Н. составит 2 236 + 2 236 + 2 418 + 2 418 + 2 418 = 11 726 рублей.

О последних изменениях по НДФЛ читайте в статье «НДФЛ: изменения 2016».

Задача на НДФЛ с решением: расчет социального вычета

В 2015 году Звягинцев М. К. оплачивал свое лечение в размере 140 000 руб. Лечение его заболевания включено в утвержденный Правительством РФ перечень медуслуг и отнесено к числу дорогостоящих. Медучреждение действует в соответствии с лицензией, а Звягинцев М. К. располагает документами, которые подтверждают его расходы, связанные с лечением и покупкой необходимых лекарств (подп. 3 п. 1 ст. 219 НК РФ).

За 2015 год доход Звягинцева М. К., участвующий в расчете налоговой базы, составил 260 000 руб. Нужно рассчитать базу по НДФЛ за 2015 год с учетом всех вышеизложенных обстоятельств.

Решение:

База по НДФЛ без учета социального вычета на лечение равна 260 000 рублей. По дорогостоящим медуслугам вычет устанавливается в размере произведенных и подтвержденных расходов. Поэтому, несмотря на то, что предельная сумма вычета на лечение составляет 120 000 рублей, в данном случае вычет можно взять в полной сумме, т. е. 140 000 рублей.

Таким образом, база по НДФЛ с доходов Звягинцева М. К. с учетом социального вычета составит: 260 000 – 140 000 = 120 000 рублей.

Контрольные вопросы

1.Основные понятия, используемые в налогообложении: субъект (плательщик) налога, сбора и носитель налога. 2.Элементы налогообложения: объект налога, налоговая база, налоговая ставка, виды ставок, налоговый период, отчетный период, порядок исчисления налогов и сборов, порядок уплаты, сроки уплаты налогов и сборов. 3.Налоговая политика государства, ее место в регулировании экономики. Цель и задачи налоговой политики. Типы налоговой политики.

Выводы

методические рекомендации по выполнению

ПРАКТИЧЕСКИХ РАБОТ

по междисциплинарному курсу
МДК 01.01. Технологии создания и обработки цифровой мультимедийной информации

профессионального модуля
ПМ 01 Ввод и обработка цифровой информации

по профессии
230103.02 Мастер по обработке цифровой информации

Г.Балашиха

Пояснительная записка

Цель практических работ - помочь обучающимся систематизировать, закрепить и углубить знания теоретического характера; научить студентов приемам решения лабораторных задач; научить их работать с книгой, нормативно-правовой и распорядительной документацией, пользоваться справочной и научной литературой; формировать умение учиться самостоятельно, т. е. овладевать методами, способами и приемами самообучения, саморазвития и самоконтроля.

Практические работы направлены на овладение студентами умений решения стандартных задач и приобретение навыков практических действий.

Основные требования к содержанию лабораторных работ:

-соответствие содержания практических заданий изученному теоретическому материалу учебной дисциплины; максимальное приближение содержания практических заданий к реальной действительности; поэтапное формирование умения, т.е. движение от знания к умению, от простого умения к сложному и т.д.; использование типовых документов, инструкций, бланков и т.п.

-использование последних версий программного обеспечения.

Результатом освоения программы профессионального модуля ПМ 01 «Ввод и обработка цифровой информации», является овладение студентами видом профессиональной деятельности, в том числе результатом освоения программы профессионального модуля ПМ 01 является овладение студентами общими (ОК) и профессиональными (ПК)компетенциями:

ПК 1.1.	Подготавливать к работе и настраивать аппаратное обеспечение, периферийные устройства, операционную систему персонального компьютера и мультимедийное оборудование
ПК 1.2.	Выполнять ввод цифровой и аналоговой информации в персональный компьютер с различных носителей.
ПК 1.3.	Конвертировать файлы с цифровой информацией в различные форматы.
ПК 1.4.	Обрабатывать аудио и визуальный контент средствами звуковых, графических и видео-редакторов.
ПК 1.5.	Создавать и воспроизводить видеоролики, презентации, слайд-шоу, медиафайлы и другую итоговую продукцию из исходных аудио, визуальных и мультимедийных компонентов средствами персонального компьютера и мультимедийного оборудования.

ПК 1.6	Набор и редактирование текста Разметка и форматирование документов Сохранение, копирование и резервирование документов. Преобразование и перекомпоновка данных , связанные с изменениями структуры документов, форм и требований к оформлению Сохранение документов в различных компьютерных форматах
ПК 1.7	Подготовка материалов для сканирования Определение параметров сканирования. Сканирование документов, сохранение, копирование и резервирование файлов с изображением Обработка изображений (масштабирование, кадрирование, изменение разрешения и палитры. Сохранение изображений в различных форматах и оптимизирование их для публикации в Интернете
ПК 1.8	Наполнение карточек объектов (товаров, услуг, персоналий) информацией Сверка сведений в базе данных с реальной ситуацией на предприятии и с текущими документами (прайс-листами, каталогами). Формирование запросов для получения недостающей информации
ПК 1.9	Форматирование (визуальное - внесение необходимой и удаление лишней информации) и настройка отображения веб-страниц. Заполнение служебной информации (названий и идентификаторов страниц, ключевых слов, мета-тегов) настройка внутренних связей между информационными блоками, страницами в системе управления контентом Установка прав доступа и других характеристик веб-страниц информационных ресурсов для просмотра и скачивания. Проверка правильности отображения веб-страниц в браузерах. Регулярное обновление (актуализация) информации в базах данных.
ОК 1	Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.
ОК 2	Организовывать собственную деятельность, исходя из цели и способов ее достижения, определенных руководителем.
ОК 3	Анализировать рабочую ситуацию, осуществлять текущий и итоговый контроль, оценку и коррекцию собственной деятельности, нести ответственность за результаты своей работы.
ОК 4	Осуществлять поиск информации, необходимой для эффективного выполнения профессиональных задач.

ОК 5	Использовать информационно-коммуникационные технологии в профессиональной деятельности.
ОК 6	Работать в команде, эффективно общаться с коллегами, руководством, клиентами.
ОК 7	Исполнять воинскую обязанность, в том числе с применением полученных профессиональных знаний (для юношей).

Уметь:

У1. подключать и настраивать параметры функционирования персонального компьютера, периферийного и мультимедийного оборудования;

У2. настраивать основные компоненты графического интерфейса операционной системы и специализированных программ-редакторов;

У3. управлять файлами данных на локальных, съемных запоминающих устройствах, а также на дисках локальной компьютерной сети и в Интернете;

У4. производить распечатку, копирование и тиражирование документов на принтере и других периферийных устройствах вывода;

У5. распознавать сканированные текстовые документы с помощью программ распознавания текста;

У6. вводить цифровую и аналоговую информацию в персональный компьютер с различных носителей, периферийного и мультимедийного оборудования;

У7. создавать и редактировать графические объекты с помощью программ для обработки растровой и векторной графики;

У8. конвертировать файлы с цифровой информацией в различные форматы;

У9. производить сканирование прозрачных и непрозрачных оригиналов;

У10. производить съемку и передачу цифровых изображений с фото- и видеокамеры на персональный компьютер;

У11. обрабатывать аудио, визуальный контент и медиафайлы средствами звуковых, графических и видео-редакторов;

У12. создавать видеоролики, презентации, слайд-шоу, медиафайлы и другую итоговую продукцию из исходных аудио, визуальных и мультимедийных компонентов;

У13. воспроизводить аудио, визуальный контент и медиафайлы средствами персонального компьютера и мультимедийного оборудования;

У14. производить распечатку, копирование и тиражирование документов на принтер и другие периферийные устройства вывода;

У15. использовать мультимедиа-проектор для демонстрации содержимого экранных форм с персонального компьютера;

У16. вести отчетную и техническую документацию.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ

Выполнению практической работы предшествует проверка знаний студентов - их теоретической готовности к выполнению задания.

Практическая работа предполагает проявление активности и самостоятельности студента при выполнении заданий.

- Прослушайте внимательно инструктаж преподавателя
- Продумайте этапы выполнения задания, распределите время, отберите необходимые обеспечивающие средства.

Подготовка к проведению практических работ предполагает владение студентами умениями и навыками умственного труда и информационной культуры:

- составлять перечни книг, систематизировать их,
- делать выписки,
- осуществлять различные виды чтения,
- проводить аналитико-синтетическую переработку информации
- Систематизировать и оформлять записи в виде аннотаций, тезисов, цитат, конспектов.

Подготавливая технические средства и оборудования для выполнения практических работ, следует строго следовать указаниям преподавателя, а также следить за последовательностью деятельности, прописанной в практической работе.

Для обеспечения безопасности работы с персональным компьютером необходимо:

перед началом работы убедиться в исправности электропроводки, выключателей, штепсельных розеток, при помощи которых оборудование включается в сеть, наличии заземления компьютера, его работоспособности.

Оценка индивидуальных образовательных достижений по результатам текущего контроля производится в соответствии с универсальной шкалой (таблица).

Процент результативности (правильных ответов)	Качественная оценка индивидуальных образовательных достижений	
	балл (отметка)	вербальный аналог
90 ÷ 100	5	отлично
80 ÷ 89	4	хорошо
70 ÷ 79	3	удовлетворительно
менее 70	2	не удовлетворительно

КРИТЕРИИ ОЦЕНКИ ПРАКТИЧЕСКОЙ РАБОТЫ

При проверке усвоения материала выявляется полнота, прочность усвоения студентом теории и умение применять ее на практике в знакомых и незнакомых ситуациях.

Отметка зависит также от наличия и характера погрешностей, допущенных студентами.

• *грубая ошибка* – полностью искажено смысловое значение понятия, определения;

• *погрешность* отражает неточные формулировки, свидетельствующие о нечетком представлении рассматриваемого объекта;

• *недочет* – неправильное представление об объекте, не влияющего кардинально на знания, определенные программой обучения;

• *мелкие погрешности* – неточности в устной и письменной речи, не искажающие смысла ответа или решения, случайные описки и т.п.

Исходя из норм (пятибалльной системы), заложенных во всех предметных областях выставляете отметка:

«5» ставится при выполнении всех заданий полностью или при наличии 1-2 мелких погрешностей;

«4» ставится при наличии 1-2 недочетов или одной ошибки:

«3» ставится при выполнении 2/3 от объема предложенных заданий;

«2» ставится, если допущены существенные ошибки, показавшие, что студент не владеет обязательными умениями поданной теме в полной мере (незнание основного программного материала):

«1» – отказ от выполнения учебных обязанностей.

СОДЕРЖАНИЕ.

1. Практическая работа № 1 "Подключение устройств к ПК по заданным условиям" Подключение мультимедийного и сетевого оборудования по заданным условиям" 10

2.Практическая работа № 2 "Кодирование информации в ПК по заданным условиям".... .13

3.Практическая работа № 3 "Настройка среды ОС по заданным условиям" "Работа с объектами ОС (файлами, папками, ярлыками) по заданным условиям" 16

4.Практическая работа № 4 Создание документа. Форматирование символов и абзацев" Создание и форматирование таблиц по заданным условиям..... 21

5.Практическая работа № 5 Обработка числовой информации. Форматы данных. Способы ввода и оформления данных" 27

6.Практическая работа № 6 Создание графических объектов в электронных таблицах"... .30

7.Практическая работа № 7 Организовать расчёты в электронных таблицах. Построение диаграмм по заданным условиям" 32

8.Практическая работа № 8 Владение методами работы с формами, электронными таблицами множеством текстовых документов 40

9.Практическая работа № 9 Использование формул в расчётных операциях с данными таблиц по заданным условиям" 43

10.Практические работы № 10 Наполнение карточек объектов (товаров, услуг, персоналий) информацией 47

11.Практические работы № 11 Формирование запросов для получения недостающей информации 51

12.Практическая работа № 12 "Обработка звука" 54

13.Практическая работа № 13 "Обработка растровых графических изображений" 65

14.Практическая работа № 14 "Создание мультимедийной презентации" 71

15.Практическая работа № 15 "Работа в программе обработки видео файлов" 85

16.Практическая работа №16 Поиск заданной информации в Интернете. Сохранение найденной информации по заданным условиям" 88

17.Практическая работа № 17 Хранение и скачивание файлов из сети Интернет (FTP) 96

18.Практическая работа № 18 Работа с языком разметки HTML" 100

19.Практическая работа №19 Закачивание файлов из сети Интернет на Yandex диск" Скачивание файла по сети" Простые и расширенные поисковые запросы" 110

20.Практическая работа №20 Форматирование (визуальное - внесение необходимой и удаление лишней информации) и настройка отображения веб-страниц..... 138

Практическая работа № 1 (1 час)

***Тема*: Подключение устройств к ПК по заданным устройствам**

Подключение мультимедийного и сетевого оборудования к ПК по заданным устройствам

Цель:

1. Изучить разъемы для подключения электропитания и внешних устройств
2. Научиться подключать и настраивать принтер, сканер

Оснащение работы: системный блок, монитор, мышь, принтер, сканер, мультимедиа проектор, ОС Windows, драйверы.

План:

1. Изучение теоретического материала
2. Выполнение заданий для практической работы
3. Подготовка отчета о лабораторной работе (домашнее задание)

Теоретическая часть

1. Изучение возможностей подключения устройств к системному блоку, организации электропитания устройств.

2. Изучение технологии установки оборудования

3. Изучение технологии настройки оборудования

Практическая часть

1. Изучение разъемов для подключения электропитания и внешних устройств
2. Подключение и настройка принтера.
3. Подключение и настройка сканера.

Контрольные вопросы:

1. Расскажите, как осуществляется питание различных устройств ПК.
2. Посчитайте количество необходимых розеток электропитания, если в состав ПК входят сканер и принтер.
3. В чем состоит особенность электропитания мониторов?
4. Какие устройства используются в ПК для стабилизации напряжения в сети?
5. Какое устройство в составе ПК позволяет стандартно завершить работу аппаратуры при внезапном отключении электропитания?
6. Как осуществляется подключение электропитания и внешних устройств в компьютере?
7. Опишите возможности настройки принтера.
8. Опишите возможности настройки сканера.
9. Опишите возможности настройки мультимедиа проектора.

Отчет о лабораторной работе должен содержать:

1. Тему работы
2. Цель
3. Оснащение
4. Результаты практической части работы.
5. Ответы на контрольные вопросы

Тема: Подключение мультимедийного и сетевого оборудования к ПК по заданным устройствам

Цель:

1. Изучить разъемы для подключения электропитания и внешних устройств
2. Научиться подключать и настраивать принтер, сканер

Оснащение работы: системный блок, монитор, мышь, принтер, сканер, мультимедиа проектор, ОС Windows, драйверы.

План:

1. Изучение теоретического материала
2. Выполнение заданий для практической работы
3. Подготовка отчета о лабораторной работе (домашнее задание)

Теоретическая часть

1. Изучение возможностей подключения устройств к системному блоку, организации электропитания устройств.
2. Изучение технологии установки оборудования
3. Изучение технологии настройки оборудования

Практическая часть

1. Изучение разъемов для подключения электропитания и внешних устройств
2. Подключение и настройка принтера.
3. Подключение и настройка сканера.

Контрольные вопросы:

1. Расскажите, как осуществляется питание различных устройств ПК.
2. Посчитайте количество необходимых розеток электропитания, если в состав ПК входят сканер и принтер.
3. В чем состоит особенность электропитания мониторов?
4. Какие устройства используются в ПК для стабилизации напряжения в сети?
5. Какое устройство в составе ПК позволяет стандартно завершить работу аппаратуры при внезапном отключении электропитания?
6. Как осуществляется подключение электропитания и внешних устройств в компьютере?
7. Опишите возможности настройки принтера.
8. Опишите возможности настройки сканера.

9. Опишите возможности настройки мультимедиа проектора.

Отчет о лабораторной работе должен содержать:

1. Тему работы
2. Цель
3. Оснащение
5. Результаты практической части работы.
6. Ответы на контрольные вопросы

Практическая работа № 2 (1 час)

Тема Кодирование информации в ПК по заданным условиям

Цель: Изучить способы кодирования текстовой, графической, звуковой информации; научиться кодировать и декодировать текстовые сообщения по предложенным правилам

Оснащение работы: системный блок, монитор, мышь, принтер, сканер, мультимедиа проектор, ОС Windows, драйверы.

План:

1. Изучение теоретического материала
2. Выполнение заданий для практической работы
3. Подготовка отчета о лабораторной работе (домашнее задание)

Теоретическая часть

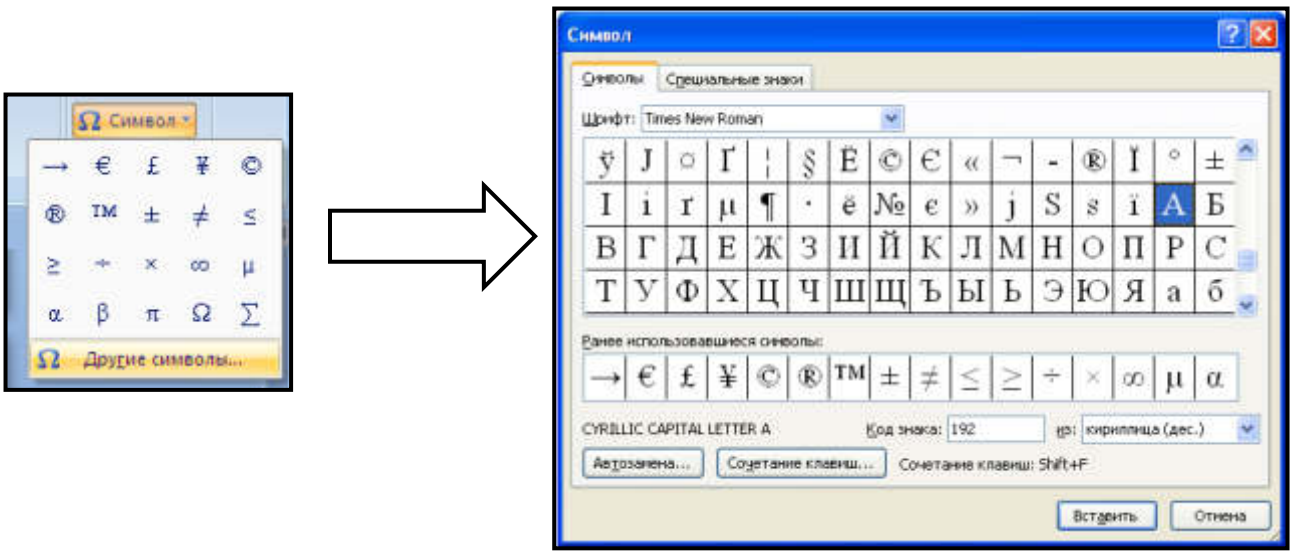
1. Изучение возможностей кодирования информации

2. Изучение технологии кодирования

3.технология решения задач на кодирование информации

Практическая часть

Используя таблицу символов, записать последовательность десятичных числовых кодов в кодировке Windows для своих ФИО, названия улицы, по которой проживаете. Таблица символов отображается в редакторе MS Word с помощью команды: вкладка *Вставка→Символ→Другие символы*



В поле *Шрифт* выбираете Times New Roman, в поле *из* выбираете кириллица. Например, для буквы «А» (русской заглавной) код знака– 192.

Пример:

И	В	А	Н	О	В		А	Р	Т	Е	М

200	194	192	205	206	194		192	208	210	197	204

П	Е	Т	Р	О	В	И	Ч
207	197	210	208	206	194	200	215

Выполнение задания №1

Задание №2. Используя стандартную программу *БЛОКНОТ*, определить, какая фраза в кодировке Windows задана последовательностью числовых кодов и продолжить код. Запустить *БЛОКНОТ*. С помощью дополнительной цифровой клавиатуры при нажатой клавише ALT ввести код, отпустить клавишу ALT. В документе появиться соответствующий символ.

Выполнение задания №2

0255	0243	0247	0243	0241	0252		0226		0225	0232	0234		0239	0238

0241	0239	0229	0246	0232	0235	0224	0252	0237	0238	0241	0242	0232

заполнить верхнюю строку названием группы

Задание №3

Расшифровать следующие записи, используя таблицу азбуки Морзе.

--	---	-.	..	-	---	.-.

.--.	.-.	..	-.	-	.	.-.

-.-.	..	..-.	.-.	.-

-	.	-.-	…	-

Задание №4

Декодировать слова по номерам букв.

23 13 6 2 -

9 15 1 12 -

9 3 21 12 -

14 29 26 30 -

2 21 12 3 1 -

25 10 19 13 16 -

Задание №5

Закодировать текст, используя код Цезаря (на одну букву вперед).

Компьютер –

Клавиатура –

Таблица –

График –

Информатика - Ваши фамилия имя отчество -

Контрольные вопросы

1. Что такое информация?
2. Какие виды информации вы знаете?
3. Как человек хранит информацию?
4. Что такое кодирование?
5. Какие способы кодирования вы знаете?
6. Приведите примеры специальных кодов.
7. Укажите правильный порядок убывания единиц измерения информации: байт, Мбайт, кбайт, Гбайт?

Отчет о практической работе должен содержать:

1. Тему работы
2. Цель
3. Оснащение
5. Результаты практической части работы.
6. Ответы на контрольные вопросы

Практическая работа № 3 (1 час)

***Тема*: Настройка среды ОС по заданным условиям**

Работа с объектами ОС (файлами, папками, ярлыками) по заданным условиям

Цель :сформировать навыки и умения устанавливать операционную систему Windows XP на виртуальную машину Virtual PC Console, а также осуществлять настройку ее параметров
Оснащение работы: системный блок, монитор, мышь, принтер, сканер, мультимедиа проектор, ОС Windows, драйверы.

План:

4. Изучение теоретического материала
5. Выполнение заданий для практической работы
6. Подготовка отчета о лабораторной работе (домашнее задание)

Теоретическая часть

1. Изучение возможностей подключения инастройки ОС.

2. Изучение технологии установки драйверов ОС

3. Изучение технологии настройки оборудования.

Практическая часть:

1. Программное обеспечение Acronis Disk Director Suite для работы с жестким диском (его тоже можно скачать в Интернете). Записываем данное программное обеспечение на диск.
2. Установочный диск с операционной системой. Его вы можете либо купить, либо скачать в сети Интернет.
3. Пакет драйверов для Вашей конфигурации компьютера (часто присутствует на установочных дисках с ОС Windows).
4. Программное обеспечение, которое Вы хотите установить себе на компьютер (часто уже на установочных дисках с ОС Windows присутствует программное обеспечение).

Контрольные вопросы

Что такое операционная система Windows?

2. Какими достоинствами обладает Windows?

3. Как называются программные приложения, работающие только под управлением системы Windows?

4. Что понимается под объектно-ориентированным подходом?

5. Перечислите приемы работы с манипулятором типа «мышь».

6. Какие основные понятия Windows вы знаете?

7. Опишите правила входа в систему Windows.

8. Из каких основных элементов состоит рабочая среда Windows?

9. Для чего предназначено Главное меню Windows?

10.Какие правила завершения работы в операционной системе Windows

Отчет о лабораторной работе должен содержать:

1. Тему работы

2. Цель

3. Оснащение

5. Результаты практической части работы.

6. Ответы на контрольные вопросы

Тема: Работа с объектами ОС (файлами, папками, ярлыками) по заданным условиям

Цель:изучить основы работы в операционной системе Windows: приёмы управления при помощи мыши, операций с файловой системой при помощи системы окон папок и работы с файловой структурой при помощи программы Проводник.

Оснащение работы: системный блок, монитор, мышь, принтер, сканер, мультимедиа проектор, ОС Windows, драйверы.

План:

1. Изучение теоретического материала
2. Выполнение заданий для практической работы
3. Подготовка отчета о лабораторной работе (домашнее задание)

Теоретическая часть

Возможности и основы использования Windows.

2. Интерфейс Windows.

3. Работа с окнами и объектами.

4. Настройка рабочей среды Windows.

Практическая часть:

1. *Получите справку «Что нового » в операционной системе* Windows*?*

2. *Работа с окнами*.

• Установите свойства Панели задач так, чтобы она автоматически убиралась с экрана.

• Убедитесь, что панель появляется на экране при перемещении указателя мыши на Панели задач и исчезает при перемещении указателя за пределы.

• Установите свойства Панели задач так, чтобы она всегда была видна на экране, затем установите ее свойства так, чтобы открытые окна перекрывали Панель задач.

• Переместите Панель задач к верхней границе экрана.

• Отобразите на Панели задач все панели инструментов, затем закройте все Панели инструментов кроме панели Рабочий стол.

• Переместите панель инструментов Рабочий стол с Панели задач на экран рабочего стола, затем переместите панель инструментов Рабочий стол к нижнему краю рабочего стола.

• Уберите заголовок панели инструментов Рабочий стол.

• Отобразите значки на панели Рабочий стол крупными значками, затем уберите подписи под значками.

• Создайте на Панель задач собственную панель инструментов, отображающую диск D, затем закройте панель D: на Панели задач.

• Закройте панель инструментов Рабочий стол.

• Откройте папку Корзина и измените размеры окна по горизонтали, по вертикали, в двух направлениях.

• Откройте папку Мой компьютер и установите ее окна режим ввода крупных значков.

• Активизируйте поочередно окна Мой компьютер и Корзина.

• Расположите окна Каскадом или Мозаикой.

• Разверните окно Мой компьютер , затем восстановите прежние размеры окна .

• Закройте все окна .

• Представьте Рабочий стол со значкам на нем.

• Откройте папку Корзина и переместите папку Корзина так, чтобы она перекрывала Панель задач.

3. *Работа с объектами*:

• Создайте папку с именем Временная на рабочем столе.

• В папке Временная создайте:

– ярлык к программе Norton (C:\NC\nc.exe) с именем Нортон;

– ярлык к программе калькулятор (C\Windows\system32\calc.exe);

– ярлык к файлу Группа И-11 папки Историки диска F;

– установите значок к ярлыку Группа БИ-11, а затем измените его;

– изменить имя программы Norton на NC.

• Создайте на рабочем столе папку Личная и скопируйте в нее из папки Временная ярлык программы NC.

• Закройте папки Историки, Личная.

• Удалите все значки в папке Временная.

• Переместите из папки Личная значок NC в папку Временная и закройте папки Личная и Временная.

• Удалите папки Личная и Временная и закройте все открытые окна.

• Откройте диск D: и создайте папки Рабочая, Личная.

• Скопируйте папку Разное в папку Рабочая.

• Создайте файл Биограф, содержащий Вашу биографию, с помощью текстового редактора Блокнот и сохраните его на диске D: в папке Рабочая.

• Скопируйте папку Рабочая на диск F:.

• Переименуйте файл Биограф в папке Рабочая диска D: в Моя биография. Переместите файл Моя биография из папки Рабочая диска D: в папку Личная диска D:.

• Откройте с помощью текстового редактора Wordpad документ Моя биография из папки Личная диска D: и допишите несколько предложений.

• Сохраните документ Моя биография на диск A:.

• Найдите все текстовые файлы, созданные за последний день с помощью Поиск.

4. *Настройка* Windows.

• Измените цвет различных элементов рабочего стола и окон.

• Измените фон рабочего стола, поместите рисунок на рабочий стол.

• Создайте рисунок с помощью графического редактора Paint и поместите его в центре рабочего стола.

• Закрыть все окна и убедиться в том, что все выполнено правильно.

• Уберите свой рисунок с рабочего стола.

• Удалите все файлы из папки Корзина.

• Установите заставку экрана.

• Установите заставку экрана Бегущая строка.

• Создайте ярлык к программе MS Word в главном меню.

• Создайте задание запуска программы Paint ежедневно в 12^{00}.

• Создайте задание запуска через 5 минут программы Косынка.

5. Обмен данными.

• Создайте в папке Рабочая диска D: несколько текстовых файлов с помощью текстового редактора Блокнот.

• Отсортируете файлы в папке Рабочая на диске D: по имени.

• Скопируйте два любых файла из папки Рабочая диска D: в папку Личная диска D: с помощью буфера обмена.

• Переместите два любых файла из папки Рабочая диска D: в папку Личная диска D: с помощью буфера обмена.

• Откройте Проводник и удалите на диске D: папку Рабочая.

• Очистите Корзину.

• Создайте на диске D: папку с Вашей фамилией, используя программу Проводник.

• Создайте в папке с Вашей фамилией вложенную папку Документы, используя программу Проводник.

• Скопируйте из папки Личная диска D: в папку Документы файл, используя программу Проводник.

Контрольные вопросы:

1. В чем особенности ОС Windows?
2. Перечислите возможности Windows Me.
3. Перечислите возможности Windows 2000 Professional.
4. Перечислите возможности Windows XP.
5. Что называется файлом, каким требованиям должно удовлетворять имя файла?
6. Что называется папкой и маршрутом папки?
7. Какие функции выполняются с помощью мыши?
8. Что включает интерфейс Windows?
9. Какие элементы присутствуют на окне Windows?
10. Как осуществляется перемещение окна, изменение размеров окна?
11. Как осуществляется просмотр содержимого в окне?
12. Как осуществляется свертывание, развертывание и закрытие окон?
13. Как осуществляется переключение между окнами?
14. Как осуществляется упорядочивание окон?
15. Какие элементы могут присутствовать в диалоговых окнах?
16. Как осуществляется создание папок и ярлыков?
17. Как осуществляется копирование и перемещение объектов?
18. Как осуществляется переименование объектов?
19. Как удалить объект и при ошибочном его удалении – восстановить?
20. Что находится папке Мой компьютер?
21. Как работать с папкой Корзина?
22. Для чего служит программа Планировщик заданий?
23. Как работать с программой Планировщик заданий?
24. Как изменить фон и рисунок Рабочего стола?
25. Как установить дату и время?
26. Как настроить раскладку клавиатуры?
27. Для чего служит программа Проводник?
28. Как выглядит окно проводника Windows?

Отчет о практической работе должен содержать:

1. Тему работы
2. Цель
3. Оснащение
5. Результаты практической части работы.
6. Ответы на контрольные вопросы

Практическая работа №4 (1час)

***Тема*: Создание документа. Форматирование символов и абзацев**
Создание и форматирование таблиц по заданным условиям

Цень: приобретение навыков работать с окнами и объектами Windows; производить настройку рабочей среды Windows

Оснащение работы: системный блок, монитор, мышь, принтер, сканер, мультимедиа проектор, ОС Windows, драйверы.

План:

1. Изучение теоретического материала
2. Выполнение заданий для практической работы
3. Подготовка отчета о лабораторной работе (домашнее задание

Теоретическая часть

1. Выбор шрифта и его размера
2. Ввод и редактирование текста
3. Форматирование текста
4. Сохранение и открытие документа
5. Выдача на печать

Практическая часть:

1 Создайте текстовый документ, содержащий краткие сведения о программах для создания и обработки текстовых документов, и выполните его сохранение в личной папке на диске Z, назвав его *Primer.doc*. Документ должен содержать:

а) не менее трех различных шрифтов;

б) состоять не менее чем из трёх абзацев.

2. Выполнить упражнения по перемещению по тексту:

- движение по строке с помощью клавиш управления курсором;
- в конец строки с помощью клавиши *End*;
- в начало строки с помощью клавиши *Home*;
- по ширине листа с помощью горизонтальной линейки прокрутки;
- по страницам документа с помощью бегунка на вертикальной линейке прокрутки;

- на следующую страницу с помощью кнопки на вертикальной линейке прокрутки;
- в конец документа с помощью *Ctrl*+ *End*;
- в начало документа с помощью *Ctrl*+*Home*.

3. Внесите изменения в документ, дополнив его сведениями о файловых операциях, и сохраните его под новым именем *Primer*1. *doc*.

4. Выполнить упражнения по выделению текста:

- Выделить строку, группу строк (используя полосу выделения на левом поле)
- выделить абзац с помощью двойного щелчка левой кнопки «мыши» на полосе выделения;
- установив указатель «мыши» на полосу выделения слева от текста выделить весь текст тройным щелчком;
- выделить весь текст с помощью пункта меню *Правка – Выделить все*;
- выделить различные несвязанные участки текста с помощью клавиши *Ctrl*.

5. Выполните предварительный просмотр документа перед печатью, сохраните документ.

6. Откройте документ *Primer. doc.* Выполнить упражнения по установке параметров страниц:

- с помощью меню *Файл – Параметры страницы* установить новые поля, поменять размер бумаги и ориентацию страницы;
- поменять поля, пользуясь вертикальной и горизонтальной линейками;

7. Откройте второй документ (*Primer1. doc*). Скопируйте фрагмент текста из одного документа и вставьте его во второй документ. Закройте *Primer1. doc.* Удалите его.

8. Выделите первый абзац документа *Primer. doc*, переместите его в конец документа всеми известными вам способами (с использованием буфера обмена *Clipboard* и использую технологию*Drop-and-Drag*. См. таб. 1 приложения).

9. Выполнить упражнения по выведению на экран специальных символов с помощью кнопки *Непечатаемые символы* на панели инструментов.

10. Выполните поиск какого-либо слова по заданному корню (команда Правка-Найти). Выполните поиск определенных символов с заданным форматом, например, задайте поиск словосочетания «текстовый процессор» с форматом: Times New Roman, 10 пт, курсив (команда Правка-Найти-Больше).

11. В документе Word выполните глобальную замену одного слова на другое слово (команда Правка-Заменить)..

12. Найдите в нем часто употребляемые слова и создайте для них элементы автотекста. Убедитесь, что созданный автотекст работает.

13. Проверьте, настроена ли программа Word на автоматическую проверку правописания, разберитесь с назначением каждого элемента настройки на вкладке Правописание.

14. Откройте документ *Primer. doc* и выполните форматирование символов первого абзаца, присвоив им гарнитуру Times New Roman, 12 пт, полужирный. Выполните эту операцию с помощью кнопок на панели инструментов, с помощью клавиш и, наконец, используя диалог Шрифт.

15. В первом абзаце текста переведите последнюю букву каждого слова первой строки в положение нижнего индекса, а первую букву каждого слова – в положение верхнего индекса.

16. Во втором абзаце задайте первой строке цвет шрифта красный, второй строке – синий, а третьей – зеленый.

17. В третьем абзаце задайте первому слову начертание букв с контуром, второму слову – начертание с тенью, третьему слову – приподнятое, а четвертому слову – утопленное. Сравните полученные начертания.

18. В четвёртом абзаце задайте уплотнённый и разреженный интервалы между символами с помощью команды Формат– Шрифт – Интервал. Сохраните документ под именем*Форматирование символов.doc*.

19. Откройте документ *Primer. doc* .

20. В каждом абзаце установите следующие параметры:

- выравнивание - по ширине;
- красная строка (т. е. отступ первой строки) – 1,5 см;
- интервалы: перед – 4 пт, после – 4 пт;
- междустрочный – полуторный.

21. Придумайте название вашего текста и оформите его, выровняв по центру.

22. Внизу поставьте вашу фамилию и инициалы и выровняйте подпись по правому краю.

23. Наберите какое-нибудь известное стихотворение. При этом пользуйтесь комбинацией клавиш Shift+Enter для перехода на новую строку и установите выравнивание абзаца по левому краю, отступ слева – 2 см.

24. Создайте список времён года так, чтобы получился маркированный список, затем скопируйте его и преобразуйте в нумерованный. Попробуйте менять формат списка.

Затем создайте многоуровневый список. Вновь скопируйте нумерованный и добавьте в список входящие в сезоны месяцы, сделав их вторым уровнем нумерации (с помощью клавиши Tab).Поменяйте формат списка (можно менять формат каждого уровня отдельно).

1. Зима
   1.a. Декабрь
   1.b. Январь
   1.c. Февраль
2. Весна
   2.a. Март
   2.b. …..

25. Сохраните документ под именем *Форматирование абзацев. doc*.

Контрольные вопросы:

1.Раскройте понятия редактирования и форматирования текста.
2. Какие существуют способы выделения фрагментов текста?
3. Как провести выделение фрагмента текста с помощью клавиш?
4. Как можно выделить отдельные символы, слова, строки текста?
5. Каким образом выделить прямоугольный фрагмент текста?
6. Как выделить отдельное слово (абзац)?
7. Как провести переключение режимов выделения в Writer?
8. Как образом можно копировать, перемещать и удалять текст?
9. Как включить режим «Непечатаемые символы» и чем он удобен для работы? 10.Как изменить формат символа (абзаца)?
11. Как провести автоматический поиск и замену?

12.Какой командой можно оформить абзац с помощью буквицы?

13.Какой командой производится разбиение текста по колонкам?

Отчет о лабораторной работе должен содержать:

1. Тему работы
2. Цель
3. Оснащение
5. Результаты практической части работы.
6. Ответы на контрольные вопросы

Тема : Создание и форматирование таблиц по заданным условиям

Цель: *Научиться создавать и форматировать таблицы в текстовом редакторе MS Word.*

Оснащение работы: системный блок, монитор, мышь, принтер, сканер, мультимедиа проектор, ОС Windows, драйверы.

План:

1. Изучение теоретического материала
2. Выполнение заданий для практической работы
3. Подготовка отчета о лабораторной работе (домашнее задание

Теоретическая часть:

1. Запустите текстовый редактор: Пуск|Программы|Microsoft Word.
2. Создайте новый документ Word: Файл|Создать.
3. Введите заголовок таблицы (14 пт).
4. Создайте шаблон таблицы 7×2: кнопка Добавить таблицу.
5. Установите, используя перетаскивание, необходимую ширину первого столбца (4 см).

Практическая часть:

1. Запустите текстовый редактор Microsoft Word;
2. Установите параметры страницы используя команду *Файл/ Параметры страницы:* размер бумаги - А4, ориентация книжная; поля: левое - 3 см, правое - 2 см; верхнее - 3 см; нижнее - 2,5 см;
3. Установите формат абзаца: *Формат/Абзац*(первая строка – отступ 1,25, межстрочный интервал - полуторный).
4. Создайте таблицу 2 х 9, используя команду *Таблица/Вставить/Таблица*
5. Измените ширину колонок по образцу табл. 11.1: наведите стрелку мыши на вертикальный разделитель таблицы, при этом стрелка мыши примет вид разделителя; нажатием и

продвижением разделителя левой кнопкой мыши задайте нужную ширину столбцов таблицы

6. Тема курса: Технологии хранения, поиска и сортировки информации
7. Выделите первую строку таблицы (шапку) и задайте тип выравнивания абзаца - по центру.
8. Выделите второй столбец таблицы и задайте тип выравнивания абзаца - по центру.
9. Заполните таблицу, перемещаясь по ней с помощью клавиш [TaЬ], [Shift]-[Tab].
10. Добавьте в таблицу новую строку, для чего поместите курсор в правую ячейку нижней строки таблицы и нажмите клавишу [TaЬ] или воспользуйтесь командой *Таблица/Добавить/Строки выше/ниже,* предварительно установив курсор в любую ячейку нижней строки таблицы.
11. Выделите всю таблицу, для чего установите курсор в любую ячейку таблицы и выполните команду *Таблица/ Выделить/Таблица* или щелкните левой кнопкой мыши по крестообразному указателю мыши в левом верхнем углу таблицы за ее контуром.
12. Произведите обрамление таблицы по образцу, используя команду *Формат/Границы и заливка.*
13. Проведите сортировку
14. Сохраните файл в вашей папке с именем «Таблица 1».
15. Произведите автоформатирование таблицы. Для этого установите курсор внутри таблицы, выберите в меню Таблица команду Автоформат и формат таблицы - Столбцы таблицы 1
16. Сохраните отформатированную таблицу в вашей папке с Именем «Таблица 2»

Контрольные вопросы.

1. Понятие таблицы.

2. Ее структура.

3. Выделение отдельных элементов таблицы.

4.Способы создания таблиц.

5.Редактирование структуры таблицы.

6.Форматирование таблиц и их содержимого. Основные элементыформатирования таблиц.

Отчет о лабораторной работе должен содержать:

1. Тему работы
2. Цель
3. Оснащение
5. Результаты практической части работы.
6. Ответы на контрольные вопросы

Практическая работа № 5 (1 час)

Тема: Обработка числовой информации. Форматы данных. Способы ввода и оформления данных

Цель : Познакомиться и частично освоить программу Excel для Windows

Научиться создавать и редактировать простую таблицу

Вводить данные разного типа в ячейки электронной таблицы

Выполнять операции по копированию, перемещению и автозаполнению отдельных ячеек и диапазонов. Научиться использовать в расчетах встроенные математические и статистические функции Excel.

Оснащение работы: системный блок, монитор, мышь, принтер, сканер, мультимедиа проектор, ОС Windows, драйверы.

План:

1. Изучение теоретического материала
2. Выполнение заданий для практической работы
3. Подготовка отчета о лабораторной работе (домашнее задание)

Теоретическая часть:

1. Запускать электронные таблицы;
2. Познакомитесь с основными понятиями: ячейка, строка, столбец, адрес ячейки;
3. Познакомитесь с командным меню
4. Научитесь вводить данные в ячейку;
5. Научитесь выделять целиком строки, столбцы, несколько ячеек, расположенных рядом, выделять таблицу целиком.

Практическая часть:

1. Запустите программу Microsoft Excel. Внимательно рассмотрите окно программы. Одна из ячеек выделена (обрамлена черной рамкой). Как выделить другую ячейку? Достаточно щелкнуть по ней мышью, причем указатель мыши в это время должен иметь вид светлого креста. *Попробуйте выделить различные ячейки таблицы.* Для перемещения по таблице воспользуйтесь полосами прокрутки.
2. Для того чтобы ввести текст в одну из ячеек таблицы, необходимо ее выделить и сразу же (не дожидаясь появления столь необходимого нам в процессоре Word текстового курсора) "писать".

   Введите в любую выделенную (активную) ячейку сегодняшний день недели, например: Среда.

   Выделите целиком ту строку таблицы, в которой расположено название дня недели. Щелкните мышью по заголовку строки (ее номеру).

Выделите целиком тот столбец таблицы, в котором расположено название дня недели.
Щелкните мышью по заголовку столбца (его имени).

3. Основным отличием работы электронных таблиц от текстового процессора является то, что после ввода данных в ячейку, их необходимо зафиксировать, т.е. дать понять программе, что вы закончили вводить информацию в эту конкретную ячейку.

   Зафиксировать данные можно одним из способов:

   - нажать клавишу {Enter};
   - щелкнуть мышью по другой ячейке;
   - воспользоваться кнопками управления курсором на клавиатуре (перейти к другой ячейке).

   Зафиксируйте введенные вами данные. Выделите ячейку таблицы, содержащую день недели, и воспользуйтесь кнопками выравнивания абзацев . Каким образом происходит выравнивание? Сделайте вывод!

4. Вы уже заметили, что таблица состоит из столбцов и строк, причем у каждого из столбцов есть свой заголовок (A, B, C...), и все строки пронумерованы (1, 2, 3...). Для того, чтобы выделить столбец целиком, достаточно щелкнуть мышью по его заголовку, чтобы выделить строку целиком, нужно щелкнуть мышью по ее заголовку. *Выделите целиком тот столбец таблицы, в котором расположено введенное вами название дня недели.*

   Каков заголовок этого столбца?

   Выделите целиком ту строку таблицы, в которой расположено введенное вами название дня недели.

   Какой заголовок имеет эта строка?

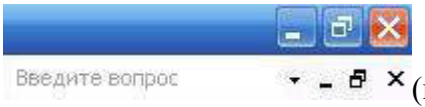
   Определите сколько всего в таблице строк и столбцов? *Воспользуйтесь полосами прокрутки для того, чтобы определить сколько строк имеет таблица и каково имя последнего столбца.*

   Внимание!!! Чтобы достичь быстро конца таблицы по горизонтали или вертикали, необходимо нажать комбинации клавиш: Ctrl+→ - конец столбцов или Ctrl+↓ - конец строк. Быстрый возврат в начало таблицы - Ctrl+Home. *Выделите всю таблицу.*

   Воспользуйтесь пустой кнопкой.

5. *Выделите ту ячейку таблицы, которая находится в столбце C и строке 4.* Обратите внимание на то, что в Поле имени, расположенном выше заголовка столбца A, появился адрес выделенной ячейки C4. Выделите другую ячейку, и вы увидите, что в

Поле имени адрес изменился. *Выделите ячейку D5; F2; A16.* Какой адрес имеет ячейка, содержащая день недели?

6. Определите количество листов в Книге1. Вставьте через контекстное меню Добавить–Лист два дополнительных листа. Для этого встаньте на ярлык листа Лист 3 и щелкните по нему правой кнопкой, откроется контекстное меню выберите опцию Добавить и выберите в окне Вставка Лист. Добавлен Лист 4. Аналогично добавьте Лист 5. Внимание! Обратите внимание на названия новых листов и место их размещения. Измените порядок следования листов в книге. Щелкните по Лист 4 и, удерживая левую кнопку, переместите лист в нужное место.
7. Установите количество рабочих листов в новой книге по умолчанию равное 3. Для этого выполните команду Сервис–Параметры–Общие.
8. Закройте документ, щелкнув по кнопке Закрыть 

(внутренний крестик) на вопрос о сохранении документа, выберите НЕТ. Выйдите из окна приложения: Файл – Выход. Также завершить работу с программой можно любым из стандартных способов:

- Файл - Сохранить …
- Файл - Сохранить как …

Контрольные вопросы:

1.Каким образом можно установить размеры и ориентацию листа?

2. Зачем нужен режим предварительного просмотра документа?

3. Какие форматы числа вы знаете? Перечислите их. Для чего они применяются и в чем их отличия?

4. По какому принципу можно отсортировать текстовые записи или числовые данные? В каком пункте меню документа находится команда сортировки?

5. Каким способом можно вставить дополнительные строки в таблицу?

Отчет о лабораторной работе должен содержать:

1. Тему работы
2. Цель
3. Оснащение
5. Результаты практической части работы.
6. Ответы на контрольные вопросы

Практическая работа №6 (1 час)

Тема: Создание графических объектов в электронных таблицах.

Цель: Освоить основные приемы работы с электронной таблицей Excel. Научиться моделировать простейшие математические объекты и настраивать таблицу для решения конкретных задач

Оснащение работы: системный блок, монитор, мышь, принтер, сканер, мультимедиа проектор, ОС Windows, драйверы.

План:

1. Изучение теоретического материала
2. Выполнение заданий для практической работы
3. Подготовка отчета о лабораторной работе (домашнее задание)

Теоретическая часть

Созданный графический объект можно

1. выделять (щелчок по объекту);
2. перемещать (выделить и выполнить перетаскивание мышью);
3. изменять размеры (выделить и выполнить перетаскивание мышью маркера выделения);
4. форматировать выделить/ Контекстное меню /Формат фигуры.

Практическая часть:

На первом рабочем листе книги «Классный журнал» оформить титульный лист журнала следующим образом (например, см. рис.):

1. В левом верхнем углу листа вставить рисунок ClipArt - книгу.
2. По центру с помощью WordArt оформить надпись «Классный журнал».
3. С помощью WordArt получить надпись на рисунке, соответствующую классу, например: «10 А».
4. Нарисовать личную панель инструментов, содержащую 4 кнопки с различными условными обозначениями.

Контрольные вопросы:.

1. Понятие таблицы.
2. Ее структура.
3. Выделение отдельных элементов таблицы.
4. Способы создания таблиц.
5. Редактирование структуры таблицы.
6. Форматирование таблиц и их содержимого.
7. Основные элементыформатирования таблиц.

Отчет о лабораторной работе должен содержать:

1. Тему работы
2. Цель
3. Оснащение
5. Результаты практической части работы.
6. Ответы на контрольные вопросы

Практическая работа №7 (1 час)

Тема: Организовать расчёты в электронных таблицах.
Построение диаграмм по заданным условиям

Цель: Научиться вводить в ячейки данные разного типа: текстовые, числовые, формулы; научиться работать со строками и столбцами.

Оснащение работы: системный блок, монитор, мышь, принтер, сканер, мультимедиа проектор, ОС Windows, драйверы.

План:

1. Изучение теоретического материала
2. Выполнение заданий для практической работы
3. Подготовка отчета о лабораторной работе (домашнее задание)

Теоретическая часть:

1. Изучить опции меню.
2. Просмотреть назначение кнопок на закладках «Главная», «Вставка», «Разметка страницы», «Формулы».
3. Найти строку формул.
4. Изучить элементы окна документа.

Перемещение указателя ячейки (активной ячейки):

1. в начало строки (HOME);
2. в начало таблицы (CTRL+HOME);
3. в последнюю заполненную ячейку (CTRL+END);
4. с помощью мыши

Практическая работа:

Выделение различных диапазонов:

1. прямоугольного диапазона:
   выделить первую ячейку/ перетаскивание мышью
   или
   выделить первую ячейку/ на последней ячейке SHIFT+ щелчок.
2. столбцов (строк):
   щелчок на имени столбца (строки) / перетаскивание (для нескольких столбцов (строк).
3. всей таблицы:
   щелчок на кнопке в верхнем левом углу таблицы.
4. несмежных диапазонов:

выделить первый диапазон/ выделить следующий при нажатой CTRL.
Обратить внимание на элементы выделенного диапазона. Найти маркер заполнения (квадратик в нижнем правом углу выделенного диапазона).

Ввод и редактирование данных:

Ввести в ячейки текст, число. Обратить внимание на представление данных в ячейках, строке формул. Выполнить редактирование данных можно двумя способами: в ячейке (дв.щелчок) и в строке формул.

Установка ширины столбцов:

Установить указатель мыши на границу столбцов/ перетаскивание мышью.
Для установки ширины столбцов равной 2 символам необходимо*: Выделить столбец/ Контекстное Меню / Ширина столбца.* Повторить изменение ширины столбцов можно с помощью команды меню Главная/Формат

Использование автозаполнения:

Ввести в ячейку A1 значение 1.
Ввести в ячейку A2 значение 2.
Выделить две ячейки A1 и A2. Установить указатель мыши на маркер автозаполнения и перетащить маркер автозаполнения вниз.

Ввод формул для ячеек смежного диапазона:

Пример, ввести формулу =срзнач(C4:F4) в ячейку G4:

1. ввести =срзнач(в ячейку G4;
2. выделить указанный диапазон C4:F4 ячеек в таблице.
3. набрать).

Ввод формул для ячеек несмежного диапазона:

Пример, ввести в ячейку W4 формулу для подсчета среднего балла за год:

1. Ввести =срзнач(в ячейку W4.
2. Выделить ячейки G4,L4,R4,V4, удерживая клавишу CTRL.
3. Набрать)

Копирование формул на смежные ячейки:

Выделить ячейку с формулой, установить указатель мыши на маркер автозаполнения и перетащить маркер автозаполнения вниз.

Копирование формул на несмежные ячейки:

Выделить ячейку, скопировать содержимое в буфер обмена, выделить ВСЕ диапазоны, куда надо скопировать формулы, выполнить вставку из буфера обмена.

Защита данных листа с предварительным указанием диапазонов, не подлежащих блокировке:

Указать диапазоны, не подлежащие защите: Выделить диапазон/ Контекстное Меню / Формат ячеек /Защита/ снять флажок защиты.

Защитить весь лист: Рецензирование /группа Изменения / Защитить лист.

Копирование листов:

Пример, установить указатель мыши на ярлык Лист2 / Перетаскивание на ярлык Лист3 (при нажатой CTRL)

Изменение имени листа в соответствии с предметами: установить указатель мыши на ярлыке листа/ Контекстное Меню / Переименовать

Индивидуальное задание

Задание 1

Подготовить электронный классный журнал, включающий сведения по одному предмету. Предусмотреть 5 оценок в четверти, средние баллы за 4 четверти и за год для 10 учеников, средний балл класса за четверти и за год (см. рисунок ниже), при этом:

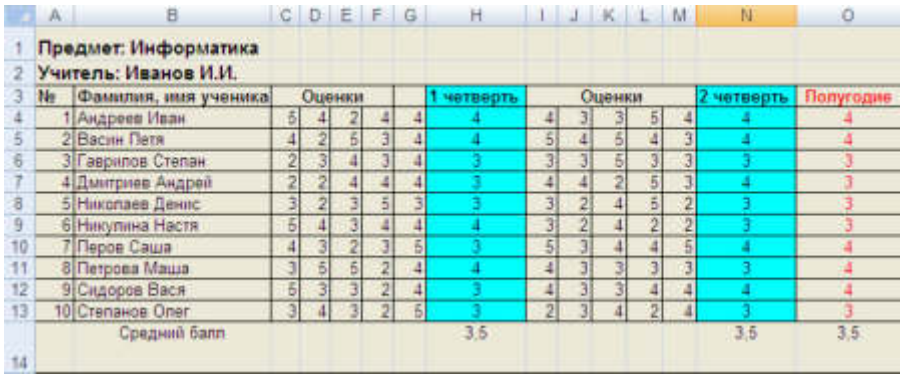
расчет средних баллов за четверти и за год осуществлять строго по формулам!

ввод оценок осуществлять различными способами:

- путем непосредственного ввода чисел в ячейку.
- с помощью функции СЛЧИС()

=ОКРУГЛ(СЛЧИС() *3+2;0)

защитить весь лист от редактирования, оставив диапазоны оценок незащищенными.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Предмет: Информатика														
2	Учитель: Иванов И.И.														
3	№	Фамилия, имя ученика	Оценки					1 четверть	Оценки					2 четверть	Полугодие
4	1	Андреев Иван	5	4	2	4	4	4	4	3	3	5	4	4	4
5	2	Васин Петя	4	2	5	3	4	4	5	4	5	4	3	4	4
6	3	Гаврилов Степан	2	3	4	3	4	3	3	3	5	3	3	3	3
7	4	Дмитриев Андрей	2	2	4	4	4	3	4	4	2	5	3	4	3
8	5	Николаев Денис	3	2	3	5	3	3	3	2	4	5	2	3	3
9	6	Никулина Настя	5	4	3	4	4	4	3	2	4	2	2	3	3
10	7	Перов Саша	4	3	2	3	5	3	5	3	4	4	5	4	4
11	8	Петрова Маша	3	5	5	2	4	4	4	3	3	3	3	3	4
12	9	Сидоров Вася	5	3	3	2	4	3	4	3	3	4	4	4	4
13	10	Степанов Олег	3	4	3	2	5	3	2	3	4	2	4	3	3
		Средний балл						3,5						3,5	3,5
14															

Задание 2

Заполнить классный журнал по трем предметам, выполнив копирование таблицы на Лист2, Лист3. Изменить имена листов в соответствии с предметами.

Задание 3

Для электронного классного журнала получить итоговую таблицу по предметам для класса. Таблица должна содержать информацию о средних баллах класса за четверти, за год по всем предметам. Таблицу разместить на Листе 5.

Данные в таблицу копировать из итоговых строк по предметам, следующим образом:

Выделить диапазон / Контекстное Меню / Копировать

Перейти на другой лист/КМ/ Специальная вставка/ Вставить ссылку.

Обратить внимание на ссылки в получившихся формулах.

Контрольные вопросы:

1. Как устроена рабочая область электронной таблицы?
2. Как переименовать лист в MS Excel?
3. Как добавить новый лист в книгу?

Отчет о практической работе должен содержать:

1. Тему работы
2. Цель
3. Оснащение
5. Результаты практической части работы.
6. Ответы на контрольные вопросы

Тема : Построение диаграмм по заданным условиям

Цель: Научиться строить диаграммы по заданным условиям

Оснащение работы: системный блок, монитор, мышь, принтер, сканер, мультимедиа проектор, ОС Windows, драйверы.

План:

1. Изучение теоретического материала
2. Выполнение заданий для практической работы
3. Подготовка отчета о лабораторной работе (домашнее задание)

Теоретическая часть:

Ввод текста и чисел

В ячейку вводятся типы данных - текст, число, дата, примечания и формула. Текст по умолчанию выравнивается по левому краю ячейки, а числа по правому.

Ввести данные в ячейку можно несколькими способами, например:

- сделать ячейку активной, то есть поместить курсор в нужную ячейку и один раз щёлкнуть левой клавишей мыши;
- ввести данные с клавиатуры;
- нажать клавишу Enter;
- сделать двойной щелчок мышью на нужной ячейке;
- ввести данные с клавиатуры;
- нажать клавишу Enter.

Редактировать данные можно также несколькими способами, например:

Для замены одних данных в ячейке другими следует сделать её активной и ввести новые данные.

- сделать активной ячейку;
- выполнить двойной щелчок на редактируемой ячейке;
- сделать активной ячейку;
- нажать клавишу F2;
- сделать активной ячейку;
- сделать щелчок в строке формул.

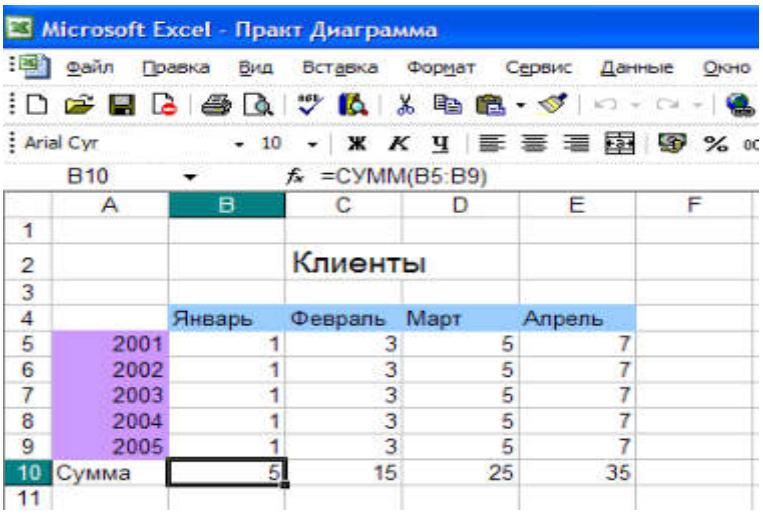
Для удаления содержимого ячейки или диапазона ячеек необходимо выделить очищаемую ячейку или диапазон и нажать клавишу Delete.

Практическая часть:

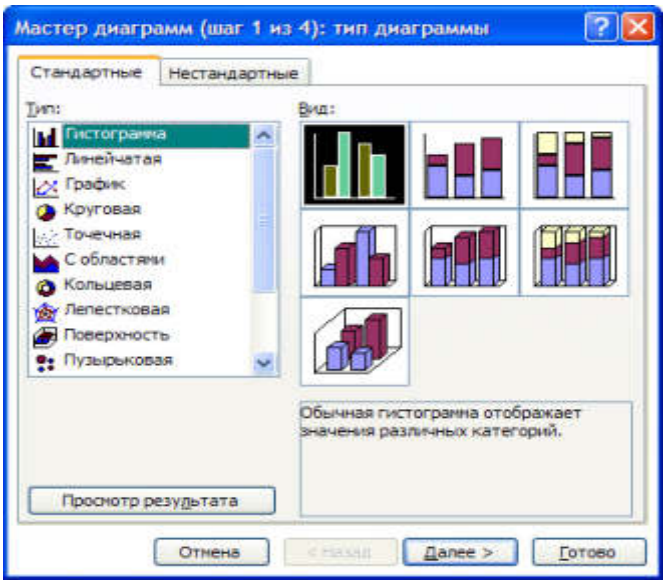
Задание: Построить диаграмму на основе таблицы «Клиенты».

Порядок выполнения практической работы:

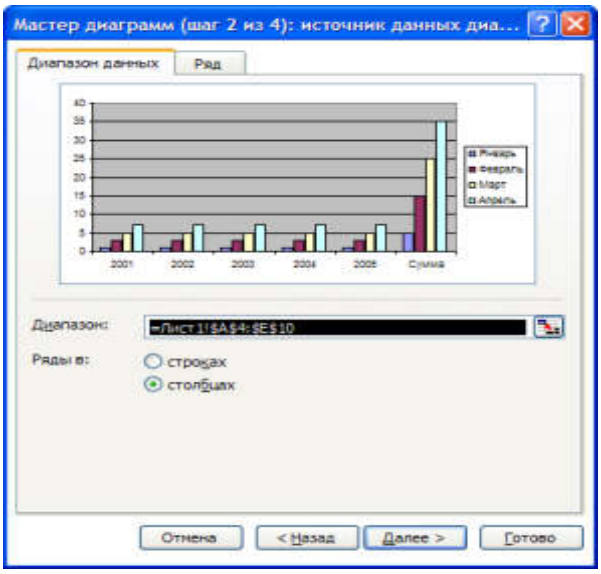
1. Открыть MS Excel и создать таблицу «Клиенты». Сумму посчитать с помощью функции =СУММ().



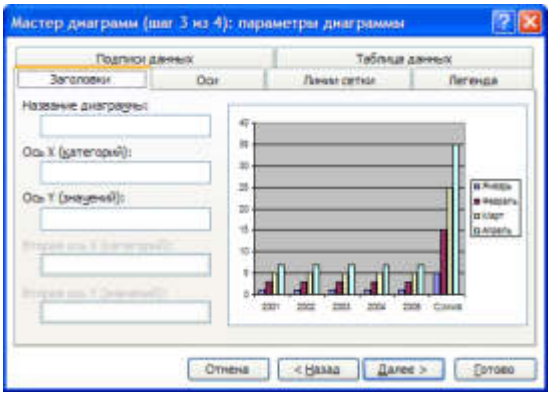
2. Выберите команду Вставка > Диаграмма. Откроется первое окно диалога мастера диаграмм, в котором нужно указать тип диаграммы, задающий ее оформление, и конфигурацию элементов, отображающих данные.



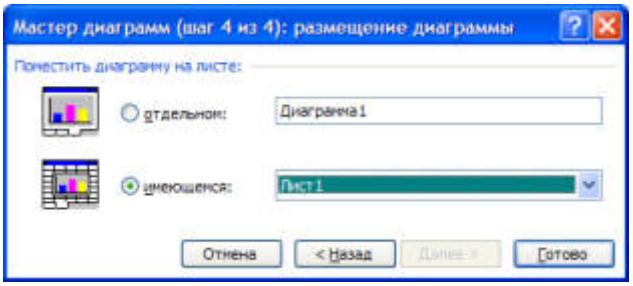
3. В списке Тип выберите Линейчатая. В разделе Вид щелкните на левом верхнем квадрате, задающем подтип Линейчатая диаграмма.
4. Чтобы не выходя из окна мастера оценить будущий вид диаграммы, нажмите мышью кнопку Просмотр результата. В окне диалога вместо списка подтипов появится внешний вид будущей диаграммы. Отпустите кнопку мыши, и окно диалога вернется в прежнее состояние.
5. Нажмите кнопку Далее.



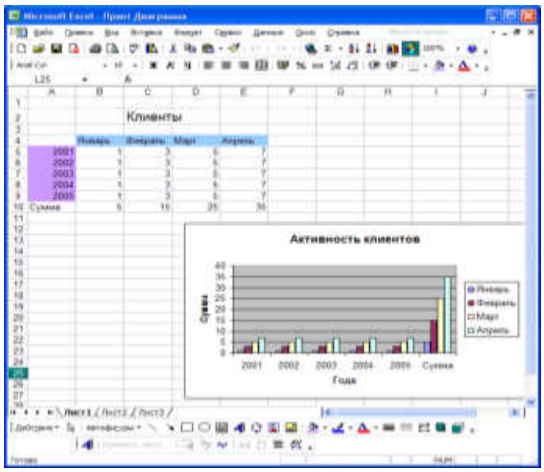
6. Второе окно диалога мастера позволяет выбрать или скорректировать источник данных. Переключатель Ряды позволяет группировать ряды данных по строкам или столбцам таблицы. Чтобы по вертикальной оси диаграммы выводились названия месяцев года, выберите положение Столбцах.
7. Для перехода к окну диалога параметров диаграммы щелкните кнопку Далее.



8. Введите в поле Название диаграммы текст Активность клиентов. В поле Ось X – Года, в поле Ось Y – Сумма.
9. Нажмите кнопку Далее.



10. Выберите положение переключателя Имеющемся.
11. Нажмите кнопку Готово.



Контрольные вопросы:

1. Как устроена рабочая область электронной таблицы?
2. Как переименовать лист в MS Excel?

   Как добавить новый лист в книгу
3. Как создать диаграмму

Отчет о практической работе должен содержать:

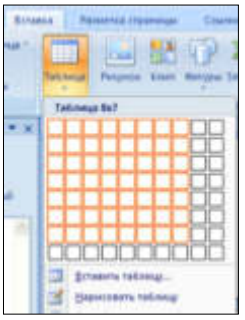
1. Тему работы
2. Цель
3. Оснащение
5. Результаты практической части работы.
6. Ответы на контрольные вопросы

Практическая работа №8 (1 час)
Тема: «Владение методами работы с формами, электронными таблицами множеством текстовых документов

Цель урока: закрепить практические навыки работы в MS WORD с таблицами в текстовых документах.

Задачи урока:

- контроль уровня теоретических знаний по данной теме;
- контроль за умением применять теоретические знания на практике (построение таблиц и работа с графикой);
- Систематизировать способы работы с элементами текстового редактора WORD.
- Научить создавать многообъектные документы в текстовом редакторе WORD.



Любая таблица состоит из *строк* и *столбцов*. Их пересечение образуют *ячейки* таблицы.

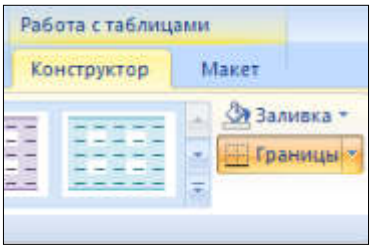
Для вставки таблицы служит кнопка Таблицы, расположенная на панели *Таблицы* лента *Вставка*. При нажатии на эту кнопку можно в интерактивном режиме выбрать необходимое количество строк и столбцов для будущей таблицы.

Создание таблиц

Задание 1. Создайте таблицу «Годовые темпы роста населения мира, в %»
(заголовок таблицы – шрифт *Times New Roman*, размер шрифта - 14 пт, стиль – полужирный, курсив; данные в таблице - шрифт *Times New Roman*, размер шрифта - 14 пт)

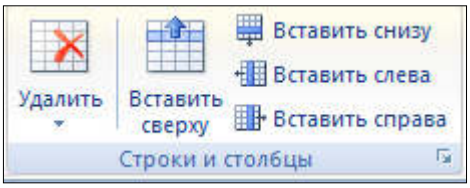
Годовые темпы роста населения мира, в %

Мир, регионы	1990-1995	2005-2010	2010-2015
Мир	1,52	1,16	1,10
Африка	2,53	2,30	2,27
Европа	0,19	0,20	0,11
Латинская Америка	1,71	1,15	1,07



После того как таблица вставлена в окне текстового редактора появляется контекстный инструмент Работа с таблицами, содержащий две ленты: *Конструктор* и *Макет*.

Редактирование таблиц



Для вставки и удаления элементов таблицы предназначены инструменты панели *Строки и столбцы* контекстной ленты *Макет*.

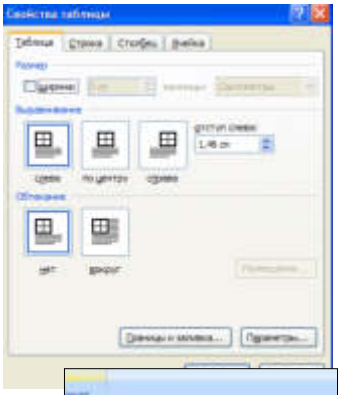
Задание 2.

1. Добавьте после третьей строки новую строку: Азия; 1,63; 1,08; 0,99.

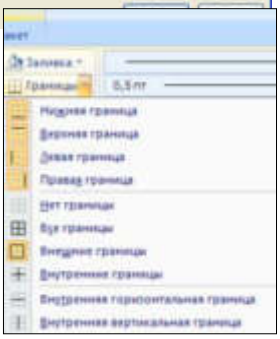
2. Добавьте после первого столбца столбец: 1980-1985; 1,76; 2,8; 1,94; 0,40; 2,1.
3. Добавьте последнюю строку в таблицу: Северная Америка; 0,97; 1,01; 0,91; 0,86.
4. Добавьте последний столбец в таблицу: 2095-2100; 0,06; 0,59; -0,31; 0,01; -0,3; 0,2.

Форматирование таблиц

Тонкие настройки таблицы (параметры строк, столбцов, ячеек) можно произвести в окне Свойства таблицы, которое открывается кнопкой *Свойства* на панели Таблица.



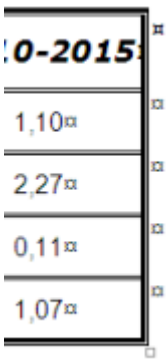
Для настроек границ таблицы служит кнопка Границы. Из ее контекстного меню можно выбрать различные типы границ.



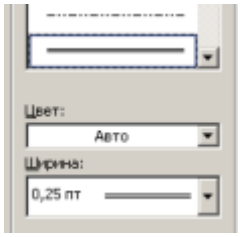
При помощи кнопки Заливка можно изменить цвет заливки ячеек таблицы.

Задание 3.

1. Установите высоту строк, равную 1 см.
2. Установите ширину столбцов, равную 3,5 см.
3. Установите *альбомную* ориентацию страницы (Разметка страницы – *Ориентация*…).
4. Установите для заголовка таблица шрифт *Verdana*, размер шрифта 14 пт, полужирный, курсив.
5. Установите для остальных строк таблицы шрифт *Arial*, размер шрифта 12 пт.
6. Отцентрируйте (выровняйте по центру) текст во всех ячейках таблицы.
7. Установите внешние границы таблицы



8. Установите внутренние границы таблицы



9. Для заголовков таблицы выберите вертикальное напрвление текста (если необходимо, увеличьте высоту строки).
10. Залейте заголовкистолбцов таблицы серым цвтом.
11. Скопируйте таблицу вместе с заголовком.
12. Удалите второй столбец во второй таблице.

13. Используя инструменты панели *Стили таблиц* контекстной ленты Конструктор установите любое автоматическое форматирование для только что скопированной таблицы.

Практическая работа №9 (1 час)

Тема: **Использование формул в расчётных операциях с данными таблиц по заданным условиям**

Цель: Изучение технологии подбора параметра при обратных расчетах.

Оснащение работы: системный блок, монитор, мышь, принтер, сканер, мультимедиа проектор, ОС Windows, драйверы.

План:

1. Изучение теоретического материала
2. Выполнение заданий для практической работы
3. Подготовка отчета о лабораторной работе (домашнее задание)

Теоретическая часть:

Формулы — это уравнения, с помощью которых можно выполнять вычисления, возвращать данные, манипулировать содержимым других ячеек, проверять условия и т. д. Формула всегда начинается со знака равенства (=).

В таблице ниже приведены некоторые примеры формул и их описания.

ФОРМУЛА	ОПИСАНИЕ
=5+2*3	Добавляет 5 к произведению 2 и 3.
=КОРЕНЬ(A1)	Использует функцию КОРЕНЬ для вычисления квадратного корня из значения в ячейке A1.
=СЕГОДНЯ()	Возвращает текущую дату.
=IF(A1>0;"Плюс";"Минус")	Проверяет, содержит ли ячейка A1 значение больше 0. Если результат проверки положителен, в ячейке появляется слово "Плюс", в противном случае — слово "Минус".

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

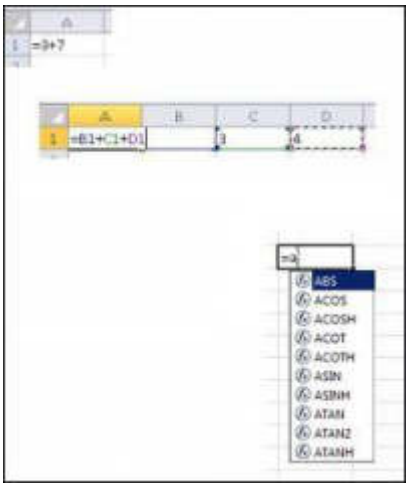
1. Выделение ячейки и ввод текста

Чтобы ввести в ячейку формулу, введите сначала знак равенства (=).

2 Ввод остальной части формулы

Выполните одно из указанных ниже действий.

- Введите комбинацию цифр и операторов, например 3+7.
- С помощью мыши выделите другие ячейки (вставив между ними оператор). Например, выделите ячейку B1 и введите знак "плюс" (+), выделите ячейку C1 и введите +, а затем выделите ячейку D1.
- Введите букву, чтобы выбрать функцию из списка. Например, если ввести "a", будут отображены все функции, имена которых начинаются на букву "a".



3 Завершение ввода формулы

- Чтобы завершить ввод формулы, в которой используется комбинация чисел, ссылок на ячейки и операторов, нажмите клавишу ВВОД.
- Чтобы завершить ввод формулы, в которой используется функция, введите необходимые функции данные и нажмите клавишу ВВОД. Например, функции ABS требуется одно численное значение, которым может быть введенное число или выбранная ячейка с числом.

Полностью введенная формула может выглядеть, как на примерах ниже.

ФОРМУЛА	ОПИСАНИЕ
=3+□	Вычисление суммы двух чисел
=B1□C1+D1	Вычисление суммы значений в трех ячейках
=ABS(-3)	Преобразование числа в его модуль

*Задание 2.*Выполнить упражнения со встроенными функциями, следуя инструкциям видео-урока «Microsoft_Excel_2007_Работа с формулами».

Практическая часть:

Порядок работы.

1. *<u>Задание 1.</u>Составить таблицу по предложенным данным. Вычислить процент и сумму прибыли организации за каждый месяц.*

Организация «ПРОМГАЗ» реализовала газ по цене 25 тенге за килограмм:

- ✓ *В январе – 1100 кг.*
- ✓ *В феврале – 1200 кг.*
- ✓ *В марте -1000 кг.*
- ✓ *В апреле - 800 кг.*
- ✓ *В мае - 960 кг.*
- ✓ *В июне – 1111кг.*
- ✓ *В июле – 1020 кг.*
- ✓ *В августе – 1100 кг.*
- ✓ *В сентябре – 1010 кг.*
- ✓ *В октябре - 1000 кг.*
- ✓ *В ноябре – 800 кг.*
- ✓ *В декабре - 970 кг.*

Месячная реализация газа составляет 900 кг (22500 тенге).

«Расчет рентабельности»

1. *Месяц и количество газа – задать, согласно условия задачи.*
2. *Сумма за реализацию рассчитывается по формуле: количество газа * сумма за 1кг (B3*C3).*
3. *Прибыль, % рассчитывается по формуле: (количество газа *100%)/900 - 100%.*
4. *Прибыль, тенге рассчитывается по формуле: сумма за реализацию - 22500(D3-22500).*

<u>Пример выполнения лабораторной работы:</u>

F3 =D3-22500

	A	B	C	D	E	F
1		Организация "ПРОМГАЗ"				
2	месяц	количество газа	сумма за 1кг	сумма за реализацию	прибыль, %	прибыль, тенге
3	январь	1100	25	27500	22,22%	5000,0
4	февраль	1200	25	30000	33,33%	7500,0
5	март	1000	25	25000	11,11%	2500,0
6	апрель	800	25	20000	-11%	-2500,0
7	май	960	25	24000	7%	1500,0
8	июнь	1111	25	27775	23%	5275,0
9	июль	1020	25	25500	13%	3000,0
10	август	1100	25	27500	22%	5000,0
11	сентябрь	1010	25	25250	12%	2750,0
12	октябрь	1000	25	25000	11%	2500,0
13	ноябрь	800	25	20000	-11%	-2500,0
14	декабрь	970	25	24250	8%	1750,0

Контрольные вопросы:

1. Что такое электронная таблица? (это программа для создания и использования документов с автоматическим расчетом вносимых данных)
2. Какие основные элементы электронной таблицы вам известны? (ячейки)
3. Как задается имя ячейки (строки, столбца) в электронной таблице? (A1, B2)
4. Какие данные можно ввести в ячейку электронной таблицы? (число, текст, формулу) Слайд №3
5. Какие данные размещены в ячейках таблицы (в C1-5, в C2-22, в D1-5, в D2-22)

	A	B	C	D
1	2	1	=A1+3*B1	=A1^2+B1
2	4	6	=A2+3*B2	=A2^2+B2

6. Что такое формула? (Формулой является выражение, задающее указание для математических вычислений.)
7. Какие правила записи формул в электронных таблицах вы знаете? (Выражение начинается со знака равенства «=», что позволяет программе отличить формулу от других данных.)

Отчет о практической работе должен содержать:

1. Тему работы
2. Цель
3. Оснащение
5. Результаты практической части работы.
6. Ответы на контрольные вопросы

Практическая работа № 10 (1час)

***Тема*: Наполнение карточек объектов (товаров, услуг, персоналий) информацией (Создание Базы данных. Обработка в БД)**

Цель: изучение технологии хранения и обработки информации

План:

1. Изучение теоретического материала
2. Выполнение заданий для практической работы
3. Подготовка отчета о лабораторной работе (домашнее задание)

Теоретическая часть

СУБД MS-ACCESS имеет следующую структуру: таблицы – для постоянного хранения данных;

- запросы – для вызова данных из таблиц по определенному шаблону и их
- дальнейшей обработки; формы – для удобного ввода и просмотра данных;
- отчеты – для удобного вывода заданной информации на печать;
- макросы – для задания свойств открываемых объектов и настройки
- порядка их работы; модули – для упрощения вызова, ранее созданных функций.
- Таблицы состоят из полей и записей.

Полями называются столбцы, а строки – записями.

Внести запись в таблицу означает заполнить данными какую- нибудь строку. Чтобы создать таблицу необходимо определить ее поля, типы данных этих полей и, иногда, некоторые дополнительные свойства этих полей.

Не все данные занимают в компьютере одинаковое место. Для их компактного хранения необходимо четко определить: что это текст или число, дата или логический символ. В базах данных под каждый тип данных резервируется некоторое пространство, и если известно, наперед, что оно не будет использовано до конца, его необходимо уменьшить. Как это сделать вы увидите по ходу выполнения данной работы.

Практическая часть:

1. Вызвать программу Access 2007.
2. В окне системы управления базы данных щелкнуть по значку <Новая база данных>. Справа в появившемся окне дать имя новой базе данных «Анкета ГС-31» и щелкнуть по значку папки, находящемуся справа от окна названия . Откроется окно сохранения, найдите свою папку и сохраните в нее новый файл базы данных «Анкета ГС-31». Затем нажмите на кнопку «Создать».
3. Появится окно <Таблица> (Рисунок 1).

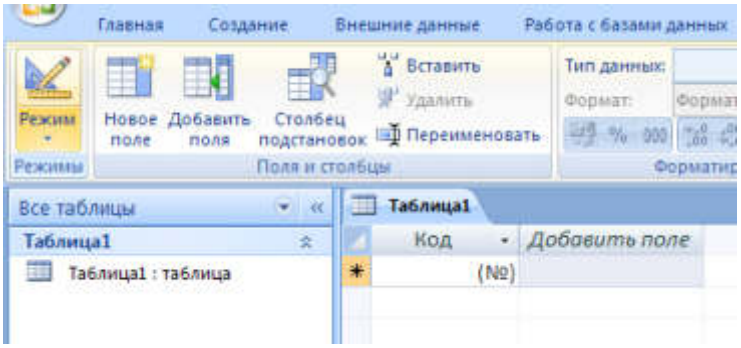


Рисунок 1

4. В появившемся окне откройте меню команды <Режим> и выберите вариант <Конструктор> и сохраните будущую таблицу под названием <Ведомость успеваемости>. Появится окно Конструктора.
5. Заполните поля в Конструкторе данными из *таблицы 1*. Тип данных можно выбрать из меню, появившемся при нажатии на кнопку в ячейке справа.

Обратите внимание*: ключевое поле «Счетчик» внесен в таблицу автоматически. Если*

напротив поля отсутствует значок ключа, то на панели инструментов щелкните по этому значку.

Ведомость успеваемости

Имя поля	Тип данных
Код	Счетчик
Фамилия	Текстовый
Имя	Текстовый
Математика	Числовой
Менеджмент	Числовой
Сервисная деятельность	Числовой
Информационные технологи	Числовой
Стандартизация	Числовой
Гостиничная индустрия	Числовой
Пропуски по неуважительной	Числовой
Пропуски по уважительной п	Числовой

Таблица 1.

6. Перейдите в режим таблицы, щелкнув по кнопке Режим на панели инструментов, Введите данные в этом режиме, заполняя клетки таблицы. Значение поля Код будет меняться автоматически.
7. Заполните базу данных значениями из *таблицы 2*. Напротив каждой фамилии выставьте по всем дисциплинам оценки от 2 до 5

Таблица 2

Код	Фамилия	Имя	Ма-тема-тика	Мене-не-джмент	Сер-вис-ная дея-тель-ность	Ин-фор-маци-онные тех-ноло-гии	Стан-дар-тиза-ция	Гости тинич нич-ная инду-стрия	Про-пуски по неува-уважи жи-тель-ной при-чине	Про-пуски по уважи жи-тель-ной при-чине
1	Иванни-кова	Анна								
2	Баранова	Ирина								
3	Корнило-ва	Ольга								
4	Воробьев	Алексей								
5	Воробьев	Олег								
6	Скоркин	Алек-сандр								
7	Володина	Нина								
8	Новоселов	Алексей								
9	Петрова	Елена								

8. Выполните редактирование ячеек:
   – Замените фамилию Иванникова на Иванова.
9. Отсортируйте:

   а) *фамилии* – по алфавиту (поставьте маркер на любую фамилию в столбце Фамилия и щелкнете мышкой по кнопке на панели инструментов или произведите сортировку с помощью контекстного меню)

   б) *имя* – по алфавиту
10. Сохраните текущую таблицу, щелкнув по кнопке «крестик» в правом верхнем углу окна таблицы.
11. Откройте снова свою базу данных.
12. Выполните поиск записей по образцу: *найти студентку по фамилии Володина*. Для этого установите курсор в поле фамилия, щелкните на кнопке <Бинокль> на панели инстру-

ментов меню Главная и в появившемся диалоговом окне введите в поле <Образец> фамилию *Володина* и щелкните по кнопке <Найти>.

Примечание: Если требуется найти следующую подобную запись, то щелкните мышкой по кнопке <Найти далее>. По окончании работы щелкните по кнопке <Отмена>.

13. Переименуйте поле «Математика» на «Информатика» с помощью контекстного меню. (Верните все как было назад).
14. Скройте столбец Пр н/пр., потом отобразите его назад.
15. Войдите в режим *Конструктора* и назначьте полю Пр н/пр и Пр ув/пр. *Маску ввода* 00 «часов». Заполните эти поля данными от 0 до 99.
16. Завершите работу с Access.

Контрольные вопросы:

1. Опишите возможности СУБД MS Access.
2. Какие объекты входят в состав файла базы данных MS Access?
3. Какие ограничения на имена полей, элементов управления и объектов действуют в MS Access?
4. Чем отличаются режимы работы с объектами базы данных в MS Access: оперативный режим, режим конструктора?

Отчет о практической работе должен содержать:

1. Тему работы
2. Цель
3. Оснащение
5. Результаты практической части работы.
6. Ответы на контрольные вопросы

Практическая работа № 11 (1 час)
Формирование запросов для получения недостающей информации

Цель работы:

- уметь создавать различные запросы

Краткие теоретические сведения:

Для создания запроса необходимо перейти на вкладку Запросы. Щёлкнуть по кнопке Создать, выбрать Конструктор. Добавить таблицу. Дважды щёлкнуть по нужному полю – оно переместится в таблицу внизу окна конструктора запросов. Задать условия отбора данных в соответствующем столбце в строке Условие отбора. Сохранить запрос.

Задание: Самостоятельное определение ключевого поля таблицы. Установка связей. Подчиненная форма. Применение фильтра к сортировке. Попробуем вести учет сдачи предметных зачетов студентами техникума. Для этого создадим две таблицы: в одной будет храниться информация, касающаяся анкетных данных студентов, а другая содержать информацию о зачетах.

1. Запустите MS Access.
2. Создайте новую базу данных под именем Техникум.
3. Для начала разработаем структуру обеих таблиц и установим связи между ними.

Создание таблиц

1. Таблица – Анкета. Создайте таблицу, содержащую следующие поля (в скобках указан тип данных):

- ФИО (текстовый)
- Группа (текстовый)
- Год рождения (числовой)
- Адрес (текстовый)
- Телефон (текстовый, создайте маску ввода 000-00-00)

Определите ключевым полем ФИО. Для этого, выделив в режиме конструктора полеФИО, вызовите контекстное меню и выберите команду Ключевое поле или выполните команду Правка – Ключевое поле. Сохраните таблицу под именем Анкета. Введите не менее 5записей.

2. Таблица – Ведомость Создайте таблицу, содержащую следующие поля (в скобках указан тип данных):

- ФИО (текстовый)
- Зачет № 1 (числовой)
- Зачет № 2 (числовой)
- Зачет № 3 (числовой)

Определите ключевым полем ФИО. Сохраните таблицу под именем Информатика Зачеты. Введите не менее 5записей.

Установка связей между таблицами

1. Находясь в окне Базы данных, выполните команду Сервис – Схема данных.
2. В схему, прежде чем устанавливать связи, нужно добавить те таблицы, между которым эти связи собираемся устанавливать.

3. В окне Добавление таблицы поочередно выберите каждую из таблиц и нажмите для каждой из них, кнопку Добавить. Затем нажмите кнопку Закрыть.
4. Для установки связей переместите мышью поле ФИО из схемы таблицы Анкет на место поля ФИО таблицы Информатика Зачеты.
5. Нажмите кнопку Создать. В окне диалога Схема данных установленные связи отображаются графически. Для редактирования связей достаточно двойного щелчка мыши по линии, изображающей связь.
6. Закройте окно диалога Схема данных, ответив Да на вопрос о сохранении макета Схемы данных.
7. Откройте таблицу Анкета и заполните данными на нескольких учащихся. Примерный вид таблицы на рисунке 3.
8. Сохраните таблицу и закройте ее.

Создание связанной формы

1. В окне базы данных выберите вкладку Формы.
2. В раскрытом окне выберите режим Создание формы с помощью мастера.
3. Выберите в открывшемся окне из Таблица: Анкета следующие поля: ФИО, Класс, а из таблицы: Информатика Зачеты: Зачет №1, Зачет №2, Зачет №3.Нажмите кнопку Далее.
4. Определитесь с дизайном формы и нажмите кнопку Далее.
5. Введите название формы Зачетная ведомость по информатике и нажмите кнопку Готово.

6. Откройте, не закрывая базу данных, таблицу Анкета и таблицу Информатика Зачеты. Скопируйте содержимое поля ФИО из одной таблицы в другую.
7. Откройте Форму Зачетная ведомость по информатикеи, пролистав карточки, проставьте некоторым учащимся оценки (не всем).
8. Закройте форму, предварительно сохранив ее.
9. Откройте таблицу Информатика Зачеты. Обратите внимание, что в этой таблице появились только те записи в которых есть результаты.

Создание запросов

I. *Простой запрос на выборку данных из таблицы* -это запрос, который позволяет выбрать некоторые поля из таблицы. Создадим запрос, позволяющий выбрать из таблицы Анкета поля Фамилия, Группа, Год рождения.

1. Перейти на вкладку Запросы.
2. Щёлкнуть по кнопке Создать, выбрать Конструктор.
3. Добавить таблицу Анкета.

4. Дважды щёлкнуть по полю Фамилия – оно переместится в таблицу внизу окна конструктора запросов. Таким же образом поместить в таблицу поля Группа и Год рождения.
5. Сохранить запрос под именем – Годы рождения.
6. Закрыть окно конструктора запросов.
7. Открыть запрос Годы рождения и убедиться, что он отображает только выбранные поля.

II. Создадим запрос, позволяющий выбрать из таблицы Анкета поля Фамилия, Группа, а из таблицы

Информатика Зачеты оценки за все зачёты - Зачет №1, Зачет №2, Зачет №3. Запрос создаётся аналогично первому запросу. Сохранить запрос под именем Зачёты.

III. *Запрос с условием*– запрос, позволяющий выбрать записи, удовлетворяющие определённому условию.

Создадим запрос, позволяющий выбрать студентов, имеющих 2 за Зачёт № 1.

1. Перейти на вкладку Запросы.
2. Щёлкнуть по кнопке Создать, выбрать Конструктор.
3. Добавить таблицу Информатика Зачеты.
4. Поместить в запрос поля Фамилия, Зачет №1.
5. В строке Условие отбора в столбце Зачёт № 1 поставить – 2.
6. Сохранить запрос под именем – Зачёт №1.
7. Закрыть окно конструктора запросов.
8. Открыть запрос, просмотреть результат.

IV. Создать запрос, который выводить фамилии студентов, имеющих 2 по всем зачётам. Сохранить запрос под именем Двоечники.

V. Создать запрос, который выводить фамилии студентов, имеющих 4или 5 по всем зачётам. Сохранить запрос под именем Хорошисты.

VI.*Запрос с параметром*– это запрос, который запрашивает у пользователя значение какого-либо параметра и выводит записи, удовлетворяющие условию отбора.

Создадим запрос, который по введённой фамилии позволяет выбрать оценки за зачёты указанного студента.

1. Перейти на вкладку Запросы.
2. Щёлкнуть по кнопке Создать, выбрать Конструктор.
3. Добавить таблицу Информатика Зачеты.
4. Поместить в запрос поля Фамилия, Зачет №1, Зачет №2, Зачет №3
5. В строке Условие отбора в столбце Фамилия ввести – [Введите фамилию].
6. Сохранить запрос под именем – Выбор по фамилии.
7. Закрыть окно конструктора запросов.
8. Открыть запрос, просмотреть результат.

Практическая работа №12 (1 час)

Тема: Обработка звука

(Работать в программе обработки звука)

Цель: овладеть практическими навыками установки программы обработки аудиоинформации, обработки аудио информации, сформировать практические умения и навыки по созданию звуковых файлов, познакомиться с технологией записи и монтажа аудиоинформации, обработки звуковой информации в программе Audacity.

План:

1. Изучение теоретического материала
2. Выполнение заданий для практической работы
3. Подготовка отчета о лабораторной работе (домашнее задание)

методические рекомендации по выполнению практической работы;

презентации: «Кодирование звуковой информации», «Цифровой звук», «Оцифровка звука», «Кодирование и обработка звуковой информации»; контрольные вопросы по теме: «Технологии обработки аудиоинформации»; задания для самостоятельного выполнения; алгоритмы-памятки для решения задач; музыкальный файл; тематические разработки для самостоятельного изучения: «Запись звука на компьютере. Программа Звукозапись; «Средства и технологии обработки звуковой информации».

4. Задачи:
5. а) отработать навыки применения установки новой программы по обработке аудиоинформации;
6. б) научиться обрабатывать аудиоинформацию;
7. в) отработать практические умения и навыки по созданию и обработке аудиоинформации;
8. г) научиться проводить запись и монтаж аудиоинформации;
9. д) отработать профессиональные навыки и умения по работе с программой обработки аудиоинформации Audacity;
10. е) открыть и обработать звуковой файл. Сохранить результаты редактирования. Определить объем файла.

Теоретическая часть

Кодирование и обработка звуковой информации

Неотъемлемой составляющей мультимедиа-технологии является обработка звуковой информации, которая получила бурное развитие с начала 90-х годов XX века. Современные

мультимедийные программные средства создания презентаций обязательно оснащены инструментами для работы со звуком. Это даёт возможность пользователю создавать более привлекательный программный продукт, расширяет возможности его применения. Для того, чтобы грамотно использовать такую возможность, необходимо представлять себе принципы обработки звука в современном компьютере.

Звук – это явление аналогового (непрерывного) характера. Колебания воздуха вызывают звуковые волны, которые могут быть представлены графиком в виде синусоиды. Но для представления звука в ПК исходный непрерывный звуковой сигнал надо разбить на части, чтобы закодировать - оцифровать. Чем мельче частицы (больше разбиений), тем точнее передача звука, меньше потери его качества (хотя полностью этого избежать нельзя).

Качество кодирования определяется частотой дискретизации и уровнем кодирования.

Частота дискретизации – это количество измерений уровня сигнала в единицу времени (секунду). Она может находиться в пределах от 8000 до 48000, то есть от 8 до 48 Кгц. При частоте 8 Кгц качество цифрового звука сравнимо со звуком радиотрансляции, а при 44,1 КГц (48 Кгц) – со звуком аудио CD.

Одновременно с временной дискретизацией (частотой) выполняется и амплитудная, то есть измерение значений амплитуды и их представление в виде чисел с определенной точностью (квантование). Эта величина называется уровнем кодирования звука и обычно составляет 16 бит, то есть каждому значению амплитуды звуковой волны соответствует двоичное число в 16 разрядов.

При оцифровке звука возможны моно- и стереорежимы. Получаемый поток двоичных чисел, описывающий звуковой сигнал, называют импульсно-кодовой модуляцией или ИКМ (PCM - Pulse Code Modulation). Такое преобразование звука происходит в аудиоадаптерах (звуковых картах), специальных устройствах расширения, которыми нужно оснастить аппаратную составляющую мультимедиа. Будучи преобразован в цифровую форму, звуковой сигнал "застывает" - в этом виде он уже не подвержен изменениям при хранении и копировании, как обычная аналоговая запись. Если с цифровым звуковым сигналом обращаться аккуратно - его можно хранить вечно и копировать любое число раз без какой-либо потери качества. Сам цифровой звук и всё, связанное с ним, принято обозначать общим термином Digital Audio.

Для управления обработкой звука мультимедиа-технология оснащается и соответствующим ПО. Стандартным звуковым программным набором системной среды Windows являются приложения:

1. Программа «Звукозапись» - это цифровой магнитофон, который позволяет записывать звук в виде файла с расширением wav (англ. wave – волна). Программа может редактировать, микшировать (накладывать) и воспроизводить звук. Можно выбирать режим двоичного кодирования – моно/стерео, частоту дискретизации, уровень.

2. Программа «Универсальный проигрыватель» обеспечивает воспроизведение звука, видео, анимационных файлов.

3. Программа «Лазерный проигрыватель» служит для воспроизведения аудио CD.

4. Программа «Регулятор громкости» служит для настройки параметров воспроизведения звука.

Кроме того, существует множество прикладных программ различного предназначения для обработки звука. Это, прежде всего, всевозможные музыкальные редакторы, которые предоставляют широкие возможности по созданию и редактированию звуковых файлов.

Звуковые файлы, как и файлы изображений, имеют свои форматы. Среди них наиболее распространены следующие:

1. wav - стандартный формат звуковых файлов в компьютерах IBM PC;
2. mid – MIDI-файл;
3. raw - формат так называемой "чистой оцифровки";
4. voc и cmf - форматы представления звука от фирмы Creative (voc – формат оцифрованного звука, cmf -формат нотных партитур и параметров инструментов для синтезаторов OPL3);
5. mod - распространенный формат, содержит оцифровки инструментов и партитуру для них;
6. stm (s3m) - формат Scream Tracker, сопоставимый с форматом mod;
7. aiff - формат звуковых файлов в компьютерах Apple Macintosh.

Для преобразования звукового файла из одного формата в другой используются специальные программы-конверторы (MP3 Compressor, XING MPEG Encoder, и другие). Такое преобразование – частая операция при работе с цифровым звуком. При этом возможно ухудшение его качества из-за уменьшения разрядности уровня кодирования, передискретизации или сжатия звука.

Используя цифровой звук в своей деятельности, пользователь должен знать, что соответствующие файлы имеют очень большие размеры: 3-х минутная стереозапись «весит» около 30 Mb. Это привело к разработке различных методов сжатия цифрового звука. Все они используют приём, при котором из звукового сигнала удаляется информация, малозаметная для слуха. В результате, несмотря на изменения, слуховое восприятие получаемого звукового сигнала практически не меняется, а степень сжатия оправдывает незначительную потерю качества. В настоящее время широко распространена группа методов сжатия звука Audio MPEG, созданная Экспертной Группой по обработке движущихся изображений (MPEG - Moving Pictures Experts Group). Методы Audio MPEG существуют в виде нескольких типов, среди которых наиболее распространён MPEG-1 Layer 3, файлы которого имеют расширение mp3. Audio MPEG разработан для воспроизведения сжатого звука в реальном времени при помощи аппаратного или программного проигрывателя (WinPlay, WinAmp, MacAmp и другие). Audio MPEG используется в аудио CD, DVD, компьютерных звуковых системах, цифровом радио/телевидении и других системах массовой передачи звука.

Существуют уже и программы распознавания речи, позволяющие управлять ПК с помощью голоса.

Вся информация в ПК представлена в дискретной, двоичной форме. Звук не является исключением. Возможность обработки звука давно привлекала разработчиков компьютеров и программного обеспечения.

Вначале звук при помощи микрофона преобразуется в электрический сигнал. Чтобы перевести непрерывный звуковой сигнал (звукозапись) в числовую форму, применяют специальное устройство, входящее в состав звуковой платы- его называют АЦП (аналого - цифровой преобразователь). Именно это устройство, через очень маленькие, равные промежутки времени измеряет электрический сигнал, результат измерения преобразует в положительное или отрицательное двоичное число и передает в оперативную память ПК.

Частота, с которой производится измерение сигнала, называется частотой дискретизации.

Для высокого качества записи (качества записи компакт диска) эта частота должна быть 44 100 Гц (раз в сек), т.е. в два раза выше, чем частота самого высокого звука, который может слышать человек.

Для кодирования каждого измеренного значения применяют четырех-, восьми-, или шестнадцатиразрядный код, в зависимости от желаемого качества записи. (При 16 битном кодировании наиболее точно описывается значение амплитуды звукового сигнала, значит и его качество выше)

Звуковая информация. Звук представляет собой распространяющуюся в воздухе, воде или другой среде волну с непрерывно меняющейся интенсивностью и частотой.

Человек воспринимает звуковые волны (колебания воздуха) с помощью слуха в форме звука различных громкости и тона. Чем больше интенсивность звуковой волны, тем громче звук, чем больше частота волны, тем выше тон звука (рис. 1).

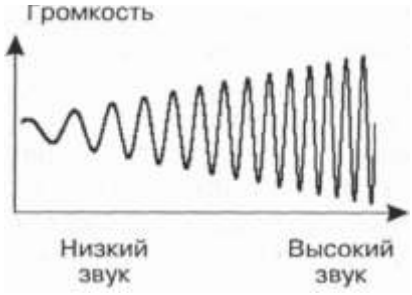


Рис.1. Зависимость громкости и высоты тона звука от интенсивности и частоты звуковой волны

Человеческое ухо воспринимает звук с частотой от 20 колебаний в секунду (низкий звук) до 20 000 колебаний в секунду (высокий звук).

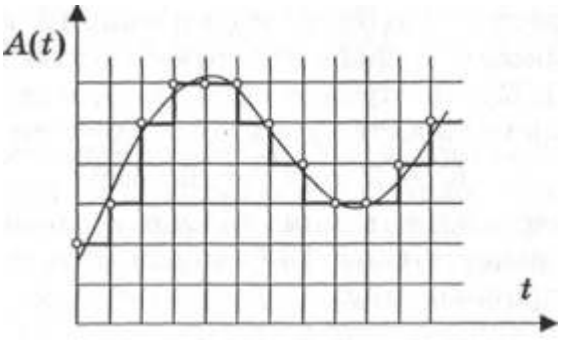
Человек может воспринимать звук в огромном диапазоне интенсивностей, в котором максимальная интенсивность больше минимальной в 10^{14} раз (в сто тысяч миллиардов раз). Для измерения громкости звука применяется специальная единица "децибел"(дбл) (табл.1). Уменьшение или увеличение громкости звука на 10 дбл соответствует уменьшению или увеличению интенсивности звука в 10 раз.

Таблица 1. *Громкость звука*

Звук	Громкость в децибелах
Нижний предел чувствительности человеческого уха	0
Шорох лис□ьев	10
Разговор	60
Гудок автомобиля	90
Реактивный двигатель	120
Болевой порог	140

Временная дискретизация звука. Для того чтобы компьютер мог обрабатывать звук, непрерывный звуковой сигнал должен быть преобразован в цифровую дискретную форму с помощью временной дискретизации. Непрерывная звуковая волна разбивается на отдельные ма-

ленькие временные участки, для каждого такого участка устанавливается определенная величина интенсивности звука.



Таким образом, непрерывная зависимость громкости звука от времени A(t) заменяется на дискретную последовательность уровней громкости. На графике это выглядит как замена гладкой кривой на последовательность "ступенек" (рис. 2).

Глубина кодирования звука. Каждой "ступеньке" присваивается определенное значение уровня громкости звука. Уровни громкости звука можно рассматривать как набор возможных состояний N, для кодирования которых необходимо определенное количество информации I, которое называется глубиной кодирования звука.

Глубина кодирования звука - это количество информации, которое необходимо для кодирования дискретных уровней громкости цифрового звука.

Если известна глубина кодирования, то количество уровней громкости цифрового звука можно рассчитать по формуле $N = 2^I$. Пусть глубина кодирования звука составляет 16 битов, тогда количество уровней громкости звука равно:

$N = 2^I = 2^{16} = 65\ 536$.

В процессе кодирования каждому уровню громкости звука присваивается свой 16-битовый двоичный код, наименьшему уровню звука будет соответствовать код 0000000000000000, а наибольшему - 1111111111111111.

Качество оцифрованного звука. Чем больше частота и глубина дискретизации звука, тем более качественным будет звучание оцифрованного звука. Самое низкое качество оцифрованного звука, соответствующее качеству телефонной связи, получается при частоте дискретизации 8000 раз в секунду, глубине дискретизации 8 битов и записи одной звуковой дорожки (режим "моно"). Самое высокое качество оцифрованного звука, соответствующее качеству аудио-CD, достигается при частоте дискретизации 48 000 раз в секунду, глубине дискретизации 16 битов и записи двух звуковых дорожек (режим "стерео").

Необходимо помнить, что чем выше качество цифрового звука, тем больше информационный объем звукового файла. Можно оценить информационный объем цифрового стереозвукового файла длительностью звучания 1 секунда при среднем качестве звука (16 битов, 24 000

измерений в секунду). Для этого глубину кодирования необходимо умножить на количество измерений в 1 секунду и умножить на 2 (стереозвук):

16 бит * 24 000 *2 = 768 000 бит = 96 000 байт = 93,75 Кбайт.

Звуковые редакторы. Звуковые редакторы позволяют не только записывать и воспроизводить звук, но и редактировать его. Оцифрованный звук представляется в звуковых редакторах в наглядной форме, поэтому операции копирования, перемещения и удаления частей звуковой дорожки можно легко осуществлять с помощью мыши. Кроме того, можно накладывать звуковые дорожки друг на друга (микшировать звуки) и применять различные акустические эффекты (эхо, воспроизведение в обратном направлении и др.).

Звуковые редакторы позволяют изменять качество цифрового звука и объем звукового файла путем изменения частоты дискретизации и глубины кодирования. Оцифрованный звук можно сохранять без сжатия в звуковых файлах в универсальном формателWAV или в формате со сжатием MP3.

При сохранении звука в форматах со сжатием отбрасываются "избыточные" для человеческого восприятия звуковые частоты с малой интенсивностью, совпадающие по времени со звуковыми частотами с большой интенсивностью. Применение такого формата позволяет сжимать звуковые файлы в десятки раз, однако приводит к необратимой потере

Описание программы обработки звуковых файлов Audacity

Audacity – многофункциональный звуковой аудиоредактор на русском языке, способный совмещать работу на нескольких звуковых дорожках одновременно. Выпущен данный софт на условиях GNU General Public License, а работает на следующих операционных системах: Microsoft Windows, Mac OS X, Linux и других.

К основным возможностям утилиты относятся:

1. Функция импорта и экспорта различных звуковых файлов, таких как WAV, Vorbis, MP3 (с кодировщиком LAME MP3), FLAC и многих других форматов;

2. Возможность записи звука с микрофона, линейного входа и прочих источников;

3. Функция, позволяющая совмещать запись и прослушивание звуковых дорожек;

4. Возможность записывать одновременно до 16 каналов. Данная функция доступна только пользователям с многоканальной звуковой картой;

5. Опция, содержащая эффекты и расширения в комплекте поставки, а также эффекты и расширения, которые устанавливаются отдельно (LADSPA); индикаторы уровня записи и воспроизведения;

6. Возможность менять темп звука, при этом, сохраняя высоту тона;

7. Обратная функция, которая позволяет изменять высоту тона и при этом сохранять темп;

8. Функция удаления шума по конкретному образцу - Звуковая чистоте;

9. Встроенная функция, позволяющая проводить спектральный анализ с использованием преобразования Фурье.

Главным преимуществом *Audacity на русском* среди подобного рода софта является возможность воспроизведения нескольких дорожек одновременно. Функция выполняется без поддержки многоканального звука. Используются всего два канала, при помощи которых микшируются все дорожки. Скачать и установить звуковой редактор Audacity – это значит, позаботится о качестве любимого звука, предоставить возможность создавать качественные аудиофайлы. При работе с программным обеспечением не может быть проблемы с форматами, так как результаты работы могут сохраняться в любые необходимые форматы.

Как пользоваться программой Audacity

Меню содержит следующие команды:

"Файл"

"Правка"

"Вид"

Это стандартные команды, значение которых объяснять не стоит.

Также есть функция "Проект", при помощи которой выполняется захват (импорт) файла в программу.

Функция "Эффекты" позволяет усовершенствовать аудиофайл, использовав такие эффекты: хорус, ревер, смена темпа, эхо, эквалайзер и смена высоты звука.

Главная панель содержит функции для выполнения проигрывания (игра, остановка, переход), а также для редактирования файла (изменить громкость, переместить и многое другое).

Группа круглых клавиш, которая появится перед пользователем, предназначена для управления аудиофайлом проекта в целом. Набор квадратных кнопок, предназначен для работы с отдельным треком (выделение, изменение огибающей, изменение сэмплов, масштабирование, сдвиг дорожки и полиинструментальный режим).

Самые важные – это кнопки "Редактирование" (вырезать, скопировать, вставить, обрезать, заполнить тишиной и многое другое). Название кнопок раскрывает суть предназначения.

Последняя операция – сохранение. Для этого необходимо просто указать нужный формат.

Практическая часть:

1. Установить программу обработки аудиоинформации *Audacity.* Убедиться, что установка прошла нормально и на Рабочем столе компьютера высветился ярлык установленной программы.

2. Подключить к компьютеру наушники и микрофон.

Подключаем микрофон в розовый вход на задней панели компьютера

После чего запускаем программу для обработки звуковой информации.

3. Познакомьтесь с описанием программы Audacity.

1) Самостоятельно создайте в своей рабочей папке папку *Звукозапись.*

2) Запустите программу Audacity.

3) Познакомиться с интерфейсом программы. Рассмотреть кнопки и пиктограммы. Записать назначение каждой кнопки.

4) Зафиксировать вид экрана программы.

5) Приступить к работе с программой. Выполнить следующие упражнения:

а) записать фразу, делая небольшие паузы после каждого слова, для этого:

- щёлкните по кнопке Записать

б) прочитайте в микрофон следующий текст:

«Один, два, три, шесть, семь, восемь, четыре, пять»

- для прекращения записи щелкните по кнопке «Остановить»

- прослушайте полученный звуковой файл, используя кнопку «Воспроизвести»

- вырежьте фрагмент «шесть, семь, восемь» (клавиши: CTRL + X или кнопки на панели);

- вставьте фрагмент «шесть, семь, восемь» после слова «пять» (клавиши: CTRL + V)

Проверьте: получилась фраза «один, два, три, четыре, пять, шесть, семь, восемь»

- из полученной фразы вырезать слова «один, два» и «семь, восемь»

Проверьте: получилась фраза «три, четыре, пять, шесть»

- сохраните полученный звуковой файл, в созданной папке *Звукозапись* для этого выполните команды: Файл \ Сохранить проект как…

- Имя файла – *Счёт*, расширение *.aup* будет присвоено автоматически

- сохраните этот же звуковой файл в папке *Звукозапись* в формате .wav, для этого выполните следующие команды: Файл \ Экспортировать.

- Имя файла *Счёт2*, расширение .wav будет присвоено автоматически

Проверьте: в папке *Звукозапись* должны быть сохранёны файлы Счёт.aup , Счёт2.wav

6) Примените к звуковому файлу Счёт.aup эффект «*Плавное затухание*», для этого:

- выделите на графике последнюю часть звуковой волны (около 2 секунд)

- выполните команду *Эффекты* \ *Плавное затухание*

- прослушайте полученные изменения

- сохраните полученный файл в папке *Звукозапись* под именем *Счёт3.aup*

7) Запишите фразу:

«Количество информации, передаваемое за единицу времени, называется скоростью передачи, или скоростью информационного потока»

- скопируйте слово «информации» (клавиши: CTRL + C) и вставьте его после слова «передачи» (клавиши: CTRL + V)

Проверьте: получена фраза «Количество информации, передаваемое за единицу времени, называется скоростью передачи *информации*, или скоростью информационного потока»

- сохраните полученную фразу в папке *Звукозапись*, имя файла *Информация.wav*

8) Запишите фразу:

«Программное обеспечение компьютера можно разделить на три части: системное, прикладное и системы программирования или среда программирования»

- скопируйте слово «*программное обеспечение*» и вставьте его после слов «*системное*» и «*прикладное*»

- вырежьте слова «*системы программирования или*» (CTRL + X)

Проверьте: получена фраза «Программное обеспечение компьютера можно разделить на три части: системное программное обеспечение, прикладное программное обеспечение и среда программирования»)

- сохраните полученную фразу в папке *Звукозапись*, имя файла: *ПО.wav*

8) Используя музыкальный файл Детские песни «Облака», сохраните его в папке *Звукозапись* под именем *Облака.mp3*

- оставьте только припев песни (*остальное вырезать*):

Облака - белогривые лошадки!
Облака - что вы мчитесь без оглядки?
Не смотрите вы, пожалуйста, свысока,
А по небу прокатите нас, облака!

- примените эффект Эквалайзер для второй и третьей части четверостишья (параметры подберите самостоятельно)

- сохраните в папке *Звукозапись*, имя файла *Облака2. wav*

- удалите файл *Облака.mp3*

В результате работы в папке Звукозапись будет 2 проекта и 4 звуковых файла.

Контрольные вопросы

1. Как частота дискретизации и глубина кодирования влияют на качество цифрового звука?

2. С помощью каких устройств можно ввести звук в компьютер?

3. С помощью каких устройств можно ввести звук в компьютере?

4. Что называется периодом измерений?

5. Запишите формулу для расчета размера цифрового аудиофайла.

Задания для самостоятельного выполнения

1. Задание с выборочным ответом. Звуковая плата производит двоичное кодирование аналогового звукового сигнала. Какое количество информации необходимо для кодирования каждого из 65 536 возможных уровней интенсивности сигнала?

1) 16 битов; 2) 256 битов; 3) 1 бит; 4) 8 битов.

2. Задание с развернутым ответом. Оценить информационный объем цифровых звуковых файлов длительностью 10 секунд при глубине кодирования и частоте дискретизации звукового сигнала, обеспечивающих минимальное и максимальное качество звука:

а) моно, 8 битов, 8000 измерений в секунду;

б) стерео, 16 битов, 48 000 измерений в секунду.

3. Задание с развернутым ответом. Определить длительность звукового файла, который уместится на дискете 3,5" (учтите, что для хранения данных на такой дискете выделяется 2847 секторов объемом 512 байтов каждый):

а) при низком качестве звука: моно, 8 битов, 8000 измерений в секунду;

б) при высоком качестве звука: стерео, 16 битов, 48 000 измерений в секунду.

4. Задание с развернутым ответом. В распоряжении пользователя имеется память объемом 2,6 Мб. Необходимо записать цифровой аудиофайл с длительностью звучания 1 минута. Какой должна быть частота дискретизации и разрядность?

Отчет о практической работе должен содержать:

1. Тему работы

2. Цель

3. Оснащение

5. Результаты практической части работы.

6. Ответы на контрольные вопросы

Практическая работа №13 (1 час)

Тема: **"Обработка растровых графических изображений"**

(Работы в программе обработки растровых графических изображений)

Цель: Изучить и освоить основные методы и приемы обработки растровых графических изображений с помощью программы - графического редактора *Adobe Photoshop*.

План:

1. Изучение теоретического материала
2. Выполнение заданий для практической работы
3. Подготовка отчета о лабораторной работе (домашнее задание)

Теоретическая часть

В настоящее время программ, работающих с графикой, очень много: простейшие программы для просмотра изображений (See , View), программы для создания и редактирования простейших рисунков (Paint , Paintbrush), программы для профессионального редактирования рисунков (PhotoEditor , PhotoShop) и программные пакеты, объединяющие все эти функции и дополняющие их (Corel Draw).

Для хранения графических файлов используются разные форматы – способы кодирования информации. На формат указывает расширение в имени файла. Некоторые графические программы могут поддерживать небольшое количество форматов, а, например,Corel Draw или PhotoEditor – почти все существующие графические форматы. В редакторах, поддерживающих несколько форматов, при сохранении файла можно выбрать и формат для их хранения.

Самые простые и самые громоздкие форматы - BMP иPCX . Они поддерживаются и читаются всеми графическими редакторами, но объем файлов-картинок таких форматов слишком велик.

В сети Internet чаще всего используются два формата графики – JPG и GIF . Самый экономичный из них – формат JPG .

К графическим файлам относятся и файлы с трехмерной графикой, часто содержащей и движение. Одно из расширений (формат) таких файлов – AVI . Картинки-мультфильмы создаются программами-аниматорами. К ним можно отнести GIF -аниматор, 3 DStudio, DMAX .

Одной из программ, позволяющих создавать и редактировать графику, является графический редактор Paint , который автоматически устанавливается вместе с ОС Windows .

Программа Paint является простейшим редактором графических изображений.

В современных графических редакторах используется два вида графики: растровая и векторная . Отличаются они принципом кодирования графики (форматом) и вследствие этого, отражением рисунка на экране после преобразования.

В случае растровой графики рисунок хранится в виде матрицы рисунка (растра), состоящего из конкретного набора цветных точек. При изменении (увеличении) масштаба каждая точка преобразуется в квадратик большего размера, то есть из маленькой аккуратной картинки после увеличения можно получить набросок в стиле кубизма.

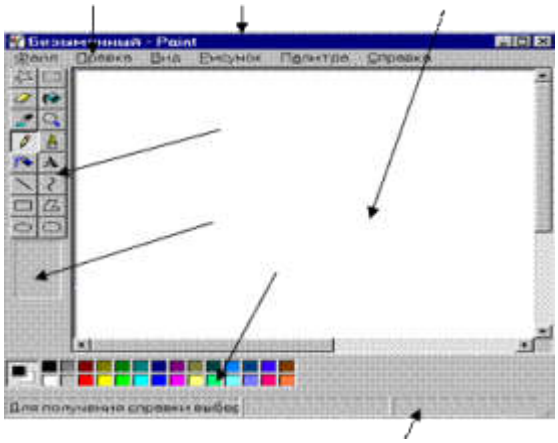
В случае векторной графики, изображение сохраняется в виде набора математических , описывающих линии, из которых состоит рисунок. Чем выше скорость процессора, тем быстрее пересчитывается изображение. При увеличении размера рисунка, заданного в векторной графике, изображение пересчитывается и выглядит после этого также гладко, как и при меньшем размере.

Векторное и растровое изображение используется и для отображения текста. Рядом слева с названием шрифта, описанного в векторном формате, находятся две буквы TT, это начальные буквы слов True Type. Названия шрифтов, описанных в растровом формате, такого комментария не содержат.

Графический редактор Paint запускается командой: Пуск => Программы => Стандартные => Paint

Как и все окна приложений ОС Windows , окно редактораPaint имеет строку заголовка, строку меню, панель инструментов, рабочее поле, горизонтальную и вертикальную полосу прокрутки, окно атрибутов, палитру, строку служебной информации.

строка меню строка заголовка рабочее поле



1. панель инструментов
2. окно атрибутов
3. палитра

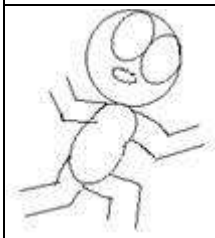
4. строка служебной информации

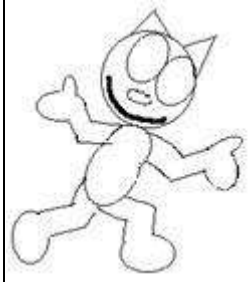
При создании изображения с помощью графического редактора, удобно основные части рисунка изображать в отдельном месте (как заготовки для аппликации).

По мере необходимости следует изменять пропорции заготовок (сжимать\растягивать, отображать симметричные элементы,стирать часть элемента, изменять угол наклона и т.п.). А затем сдвигая заготовки, собирать в стороне из частей готовый рисунок, используя Выделение блоков, Копирование, Перемещение, Удаление.

Для детальной дорисовки изображения следует использовать команду Изменение масштаба и в укрупненном масштабе дорисовать все неточности.

В качестве примера приведем технологию рисования кота Леопольда:

	1. Нарисуйте четыре, не пересекающих друг друга эллипса. 1 – голова 2 – глаз 3 - верхняя часть туловища 4 - нижняя часть туловища Детали 2,3,4 выделите и выполните наклон деталей по вертикали на 15-20 градусов.
	2. Переместите глаза на положенное место, используя прозрачный фон, операцию копирования и резинку для стирания лишних частей.
	3. Соедините два эллипса туловища и совместите его с головой, используя операции предыдущего шага.
	4. Используя Карандаш и Прямую нарисуйте руки и ноги кота. Нарисуйте отдельно эллипс для носа, наклоните его и перенесите на положенное место, используяпрозрачный фон.

	5. Используя Прямую нарисуйте уши кота. 6. Нарисуйте отдельно эллипс для лапок. Затем используя различные наклоны для эллипса, скопируйте его на положенные места, используя при этом прозрачный фон. 7. Лишние линии сотрите резинкой.
  	8. Используя толстую Кривую нарисуйте рот кота. 9. Нарисуйте отдельно эллипс для пальцев. Затем используя различные наклоны для эллипса, скопируйте его на положенные места, используя при этом прозрачный фон. 10. Лишние линии сотрите резинкой, а если переусердствуйте, то дорисуйте Карандашом, используякрупный масштаб рисунка. 11. Используя Кривую и наклоненный эллипс, нарисуйте хвост кота. 12. Нарисуйте Карандашом волосы кота, используя крупный масштаб рисунка. 13. Используя наклоненные эллипсы, нарисуйте зрачки кота.
	14. Нарисуйте Карандашом бантик и усы кота, используя крупный масштаб рисунка. 15. Проверьте все ли линии, частей, предназначенных для закраски замкнуты. Разрывы дорисуйтеКарандашом . 16. Используя заливку закрасьте кота.

Практическая часть:

1. Освоить основные приемы работы с графическим редактором Paint .
2. Изучить приемы изображения в растровой и векторной графике.
3. Запустите графический редактор Paint ;
4. Создайте 2 надписи своего имени в растровой и векторной графике, используя размер шрифта 76;
5. Создайте новый файл;
6. Определитесь с размерами рисунка;
7. Сохраните новый файл в своей папке;
8. Нарисуйте кота Леопольда или Снеговика, используя следующие указания:

9. 9.Рисуйте основные части рисунка отдельно, а затем собирайте в целое;
10. 10При перемещении частей фон надо сделать прозрачным;
11. Если какая-нибудь из частей не пропорциональна, то с помощью выделения и растяжения или сжатия, поправьте изъян;
12. Если вы используете заливку , то контур должен быть замкнутым;
13. Если при заливке краска заполняет большую область, чем вам требуется, то отмените последнее действие (Правка\Отменить), увеличьте масштаб просмотра и дорисуйте разорванные линии. Затем можно уменьшить масштаб и повторить заливку ;
14. Не забудьте, что резинка стирает цветом фона;
15. Не забывайте сохранять изменения;
16. Контуры частей удобно рисовать в обычном масштабе ;
17. Используйте крупный масштаб для удаления или дорисовки контура или дополнительных штрихов к рисунку;
18. Берите за основу, если это возможно, стандартные фигуры (эллипс, прямоугольник, кривую);
19. Можете сделать какую-нибудь надпись, и затем сохраните эту открытку в своей сетевой папке. (В шестой лабораторной работе вы сможете послать ее по локальной сети своему сокурснику.).

Контрольные вопросы:

1. Какие программы называются графическими редакторами?
2. Приведите примеры известных вам программ для создания графических изображений (статика и анимация), прокомментировав их назначение?
3. Что называется форматом файла?
4. Приведите примеры известных вам форматов графических файлов. Сравните их между собой;
5. Перечислите возможности редактора Paint;
6. Назовите основные элементы окнаPaint. В чем их назначение?
7. Расскажите о назначении инструментов на панелиинструментов;
8. Чем отличается растровый и векторный форматы записи графических изображений?
9. Какой размер листа в вашей версии Paintустанавливается по умолчанию?
10. Для чего предназначены команды пункта меню Файл?

11. Для чего предназначены команды меню:
12. Правка\Очистить выделение;
13. Правка\Повторить;
14. Правка\Копировать в файл?
15. К чему приведет заливка незамкнутой области?
16. Как используется инструмент Выделение произвольной области? В чем заключается разница между двумя инструментами для выделения области?
17. Объясните, какие возможности дает опция Ориентация при подготовке к распечатке файла?
18. Как повернуть часть рисунка на 180°?
19. Как отразить весь рисунок слева направо?

Отчет о практической работе должен содержать:

1. Тему работы
2. Цель
3. Оснащение
5. Результаты практической части работы.
6. Ответы на контрольные вопросы

Практическая работа № 14 (1 час)

Тема : **Создание мультимедийной презентации**

Цель: Создание мультимедийной презентации в Microsoft PowerPoint

План:

1. Изучение теоретического материала
2. Выполнение заданий для практической работы
3. Подготовка отчета о лабораторной работе (домашнее задание)

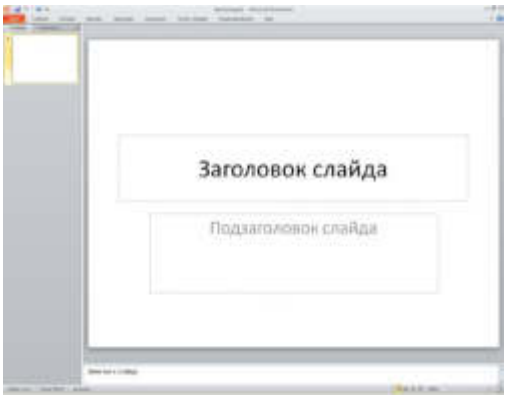
Теоретическая часть:

Под *презентацией* подразумевается наглядное сообщение (доклад), передающий для аудитории новых идеи, Например, представление нового учебного курса, планов фирмы или разработку новых товаров. *Компьютерная презентация* - это мультимедиа информация (файл) для публичного выступления. Компьютерная презентация состоит из последовательности кадров, называемых *слайдами*. Эти слайды можно не только выводить на экран компьютера (или специального проектора), но и напечатать на бумаге (или прозрачной пленке). Популярнейшей программой для создания презентаций является PowerPoint из пакета Microsoft Office. Она позволяет создавать файлы презентаций, включающие демонстрационные слайды, структуру, заметки докладчика и раздаточные материалы (или выдачи) для слушателей.

Практическая часть:

Запуск Microsoft Office PowerPoint 2010

Для начала работы над примером следует выполнить команду Пуск-Все программы-



Microsoft Office PowerPoint 2010 (рис. 1).

Рис. 1. Стартовое окно программы Microsoft Office PowerPoint 2010

Окно и инструментарий PowerPoint во многом напоминают как Microsoft Word, так и другие приложения MS Office 2010.

2. Создаем титульный слайд и шаблон презентации

Учебную презентацию будем создавать по фотоделу и в качестве заглавного слайда будем использовать рисунок, связанный с фотографией.

Выполните команду Главная-Создать слайд и в темах Office найдите вариант Пустой слайд (рис. 2).

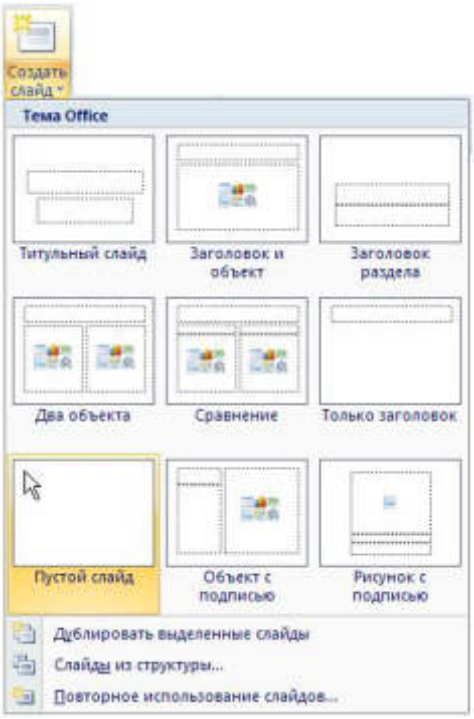
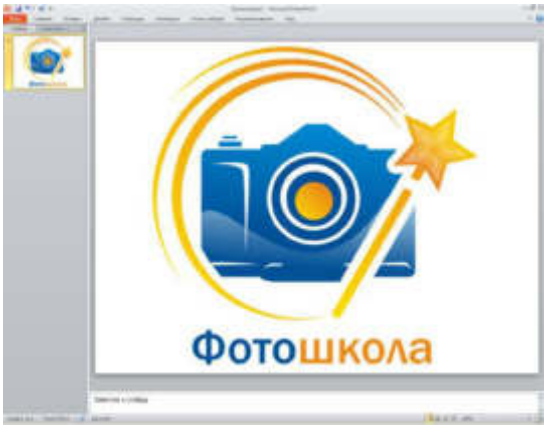


Рис. 2. Выбор темы Office (Выбор макета слайда)

Теперь выполните команду Вставка-Рисунок-Вставить рисунок из файла и укажите нужную



картинку для титульного слайда (рис. 3).

Рис. 3. Вставка рисунка в первый слайд презентации

Теперь установим шаблон для следующих листов презентации и зададим нумерацию листов.

Существует два видов шаблонов PowerPoint: *шаблоны содержания* (тема Office) и *шаблоны дизайна* (Темы). Первые расположены на вкладке Главная и представляют собой заготовки с объектами презентаций (текстом и мультимедиа). Вторые содержат только графику, без начинки слайда объектами. Недостающие элементы оформления можно будет добавить позже при редактировании.

Темой Office мы уже воспользовались, выбрав вариант Пустой слайд. Для выбора дизайнерской темы перейдем на вкладку Дизайн и в поле Темы выберем вариант темы с названием Главная (рис. 4).

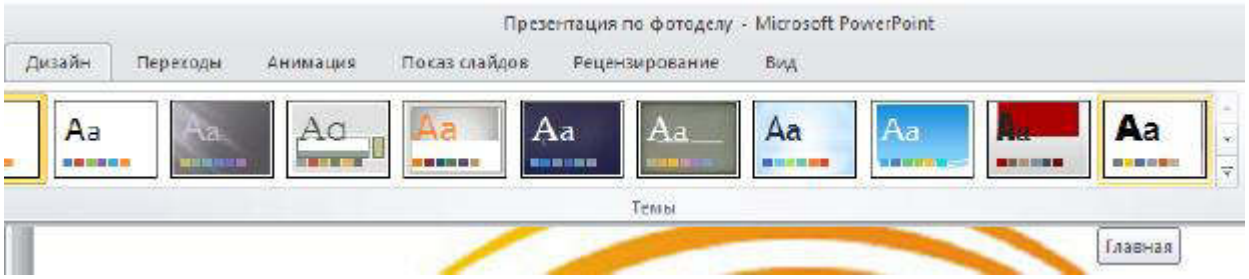


Рис. 9.4. Применение к титульному слайду темы Главная

Чтобы присвоить слайду его порядковый номер следует перейти на вкладку Вставка и нажать на значок Колонтитулы, установить флажок Номер слайда и нажать на кнопку Применить ко всем (рис. 5).

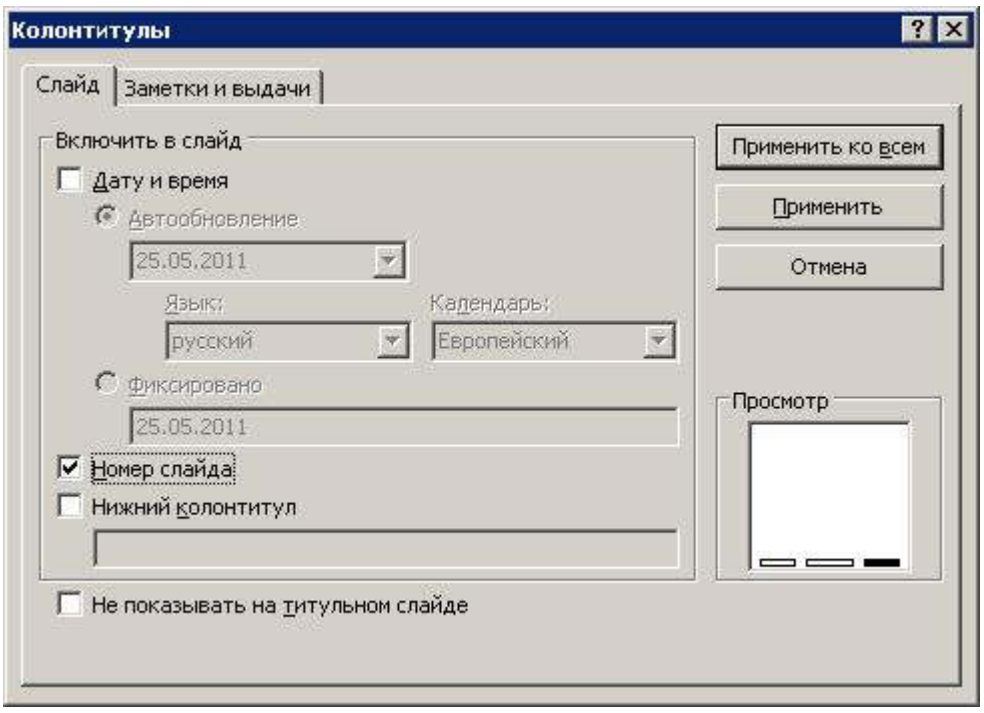


Рис. 5. Флажок Номер слайда установлен

3. Вставка и редактирование слайдов презентации

Как правило, перед созданием слайдов презентации нужно сформировать план ее содержания. Это содержание послужат в дальнейшем заголовками для слайдов. В нашем случае исходным содержанием нашей презентации будут слайды из нескольких других, черновых презентаций.

Создать (после титульного) последующие слайды презентации можно различными способами:

Способ 1

Перейдите на вкладку Структура. Если вы хотите создать пустой слайд такой же структуры, как предыдущий, то поместите курсор в конец предыдущего слайда-абзаца и нажмите на клавишу Enter.

В нашем случае получится пустой второй слайд. Для копирования информации на него представьте, что слайд - это абзац текста, и вставляйте его, как будто работаете в MS Word. Иначе говоря, если прототип слайда уже где-то существует (в другой презентации или в текстовом документе), то выполните команду Копировать, (рис. 6). Затем поместите курсор в нужную точку презентации и выполните команду Вставить. При вставке следует сохранить исходное форматирование документа.

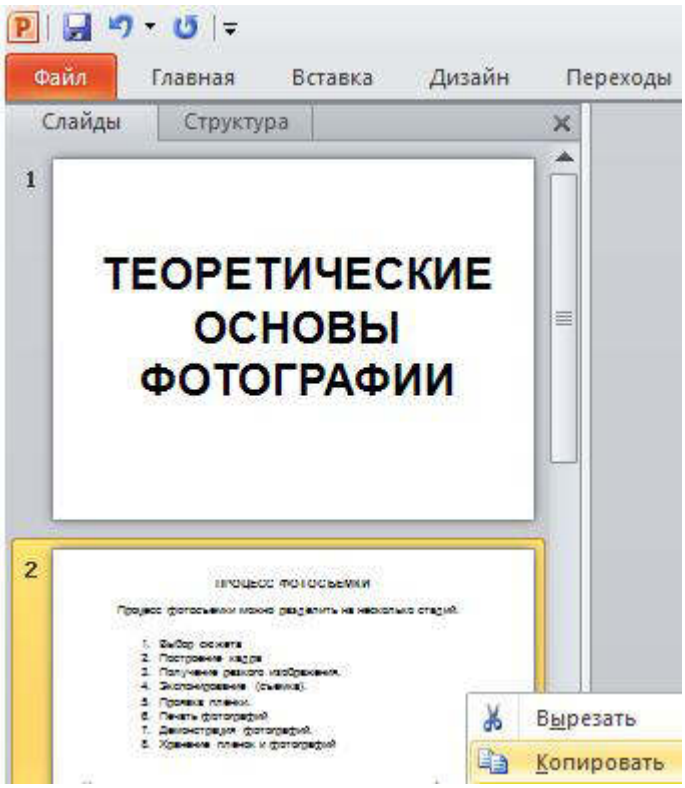


Рис. 6. Показана команда Копировать, для копирование слайда из одной презентации в другую

Способ 2

На вкладке Слайды выделите уже имеющийся слайд презентации, вызовите правой кнопкой мыши контекстное меню и при помощи команды Дублировать слайд получите слайд-близнец, расположенный сразу после его оригинала (рис. 7).

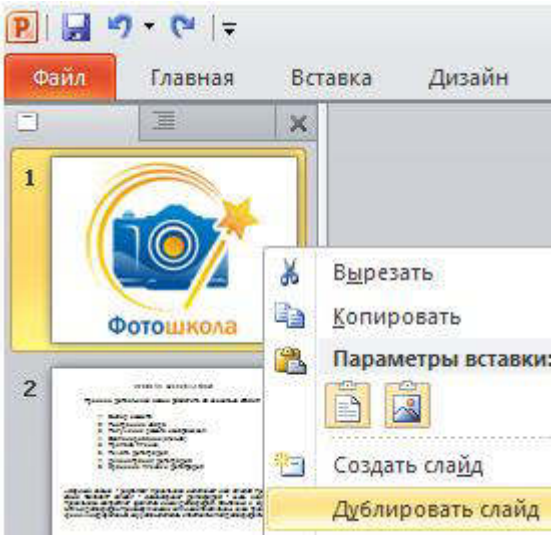


Рис. 7. Получение дубликата слайда

Способ 3

В режиме Сортировщик слайдов нужно выделить мышкой копируемые слайды в черновой презентации и проделать комбинацию клавиш CTRL+C (рис. 8).

Рис. 9.8. Слайды для копирования выделены

Далее перейдите в нашу (чистовую) презентацию и, на вкладке Слайды, выполните комбинацию клавиш CTRL+V.

Так, описанными выше способами мы добавили к титульному слайду еще несколько слайдов презентации (рис. 9).

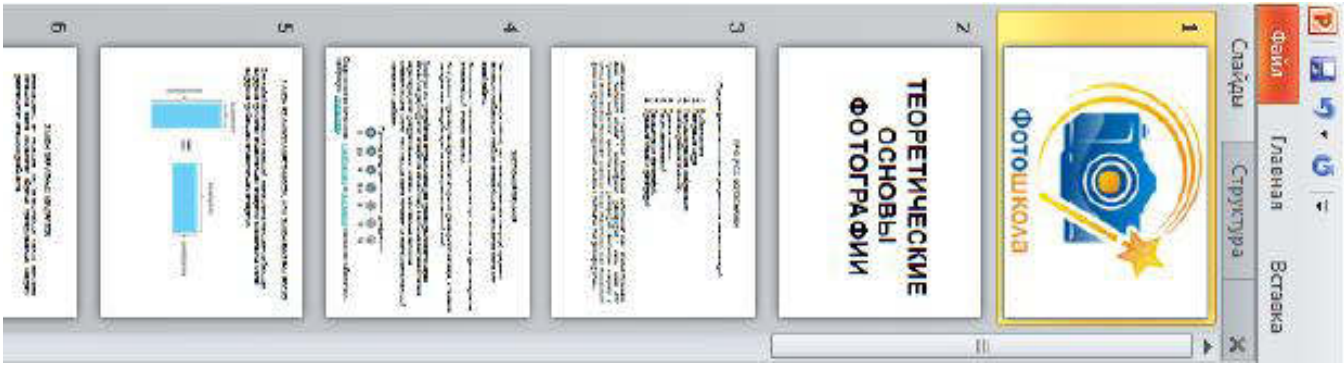


Рис. 9. Добавленные слайды

4. Работа над текстом

В презентации, как правило, меньше текста, чем в обычных текстовых документах. Вместо больших массивов текста правильнее вынести на слайды только то, чего не объяснишь на словах: формулы, рисунки, таблицы и т. п., вплоть до аудио и видеофрагментов. Однако минимальное текстовое наполнение слайда тоже необходимо. Оно служит не только пояснением для картинок и таблиц, но и скелетом, на который опирается тело доклада. Созданные черновые надписи презентации впоследствии редактируют, передвигают с места на место, а также, по мере необходимости, создают новые. Если у выступления есть красная нить - пускай она проходит через все слайды в виде ключевой фразы (девиза или эмблемы).

В предыдущих шагах мы уже выбрали шаблон презентации (тему Главная) и создали основные слайды презентации. Теперь посредством инструментов для работы с текстом улучшим текст слайдов.

В режиме послайдного редактирования повернуть и переместить текст в PowerPoint можно двумя способами. Если большая точность не требуется, воспользуйтесь мышью. Такой метод перемещения текста в PowerPoint почти не отличается от перемещения большинства объектов Windows: текст просто перетаскивают (буксируют) мышью. Единственная тонкость заключается в том, что при перемещении ухватить текст следует за текстовую рамку, которая появляется вокруг него при выделении (рис. 10).

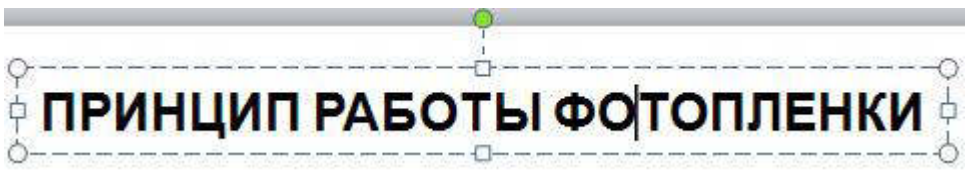


Рис. 10. Маркеры для перемещения и поворота текстового объекта

Чтобы перемещать текст строго по горизонтали или по вертикали, удерживайте нажатой клавишу Shift.

Если же перетащить один из боковых или угловых манипуляторов, расположенных на рамке, то вы измените область ввода текста.

Чтобы повернуть текст, щелкните на зеленом круглом маркере над рамкой текста - по углам выделенного объекта появятся жирные точки, а указатель мыши примет такой вид, как на рис. 11. Если навести его на одну из этих жирных точек и, удерживая нажатой левую кнопку, перетащить указатель в сторону, то выделенный текст станет поворачиваться в соответствующем направлении.

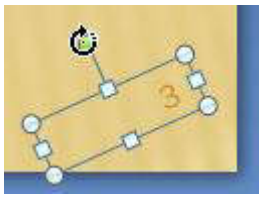


Рис. 11. Пример вращения текста влево

Для изменения размера шрифта текст выделяется, после чего на панели Шрифт вкладки Главная выбираются необходимые параметры шрифта. Для выделенного текста параметры его форматирования выбирают на панели Абзац (рис. 12).

Рис. 12. Панели форматирования шрифта и текста

Один из возможных вариантов создания красивых заголовков для слайдов - использование коллекции WordArt. Выделите любой из заголовков слайда и перейдите на вкладку Вставка. Щелкните на кнопке WordArt и выберите из появившегося диалогового окна, понравившийся вам шрифтовой стиль - рис. 13.

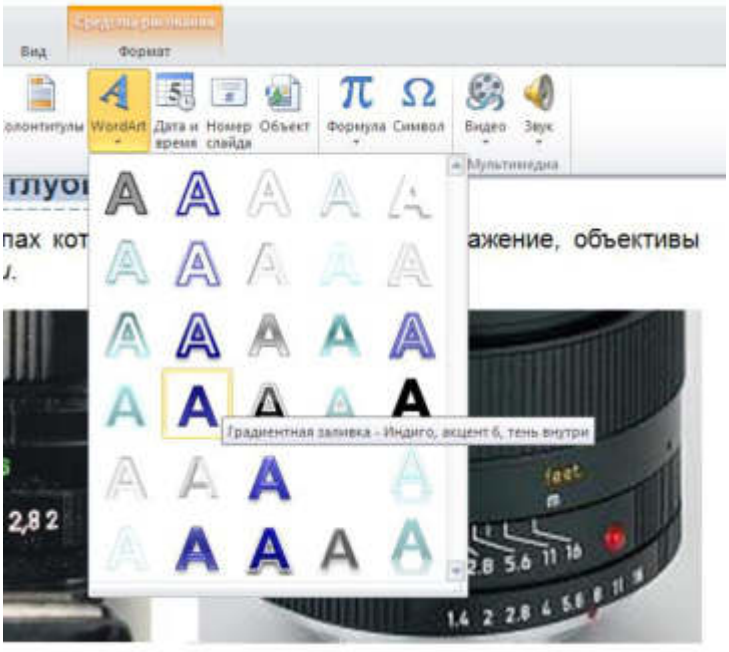


Рис. 9.13. Коллекция стилей WordArt

Однако, излишне вычурное оформление может отвлечь внимание аудитории от главного - основной идеи доклада. Поэтому не пытайтесь применить в одной презентации все возможности PowerPoint.

5. Выравнивание, распределение и группировка объектов

Вкладка Вставка позволяет добавить картинку, рисунок, автофигуру и т.д. У каждого такого объекта PowerPoint есть набор параметров, которые можно менять в окне свойств. Как и в большинстве приложений, имеющих дело с произвольно располагаемыми объектами, в PowerPoint есть функции для упорядочения этих объектов относительно друг друга и слайда - выравнивание и распределение. Эти функции находятся на вкладке Главная в поле Рисование и меню Упорядочить (рис. 14).

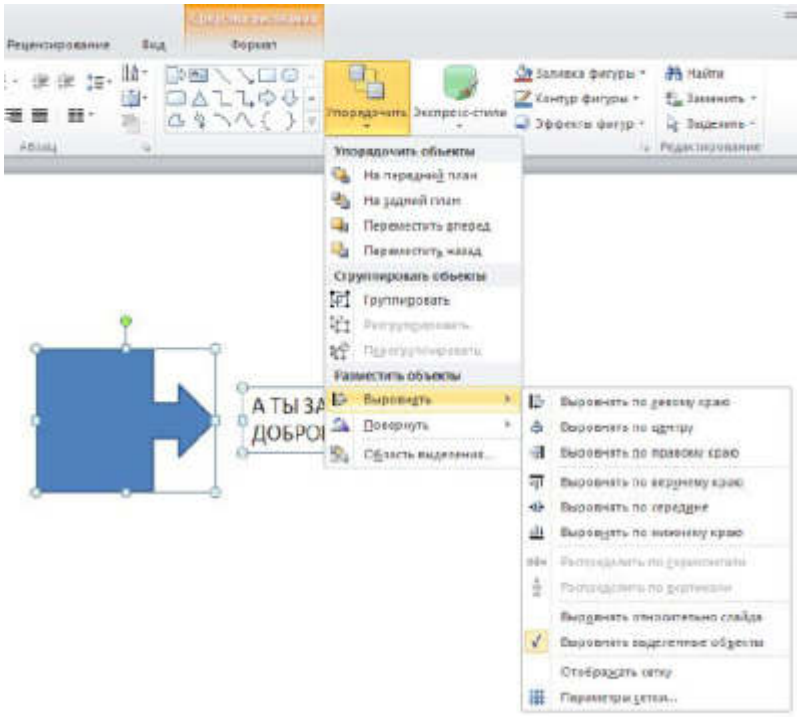


Рис. 9.14. Команды упорядочения объектов в PowerPoint

Прежде чем воспользоваться командами, не забудьте выделить объекты, которые нужно упорядочить. Кроме того, объекты можно объединить в группу, которая будет рассматриваться приложением как один объект. Для этого используете команду Группировать. Если понадобится снова разъединить группу, то воспользуйтесь командой Разгруппировать.

Стандартные графические объекты PowerPoint можно сделать трехмерными. К сожалению, сделать это с графическими объектами, заимствованными из других приложений, невозможно. Но для родных объектов PowerPoint - пожалуйста. Чтобы в этом убедиться, достаточно выделить

один из них и на вкладке Главная в поле Рисование воспользоваться кнопкой Эффекты для фигур (рис. 15).

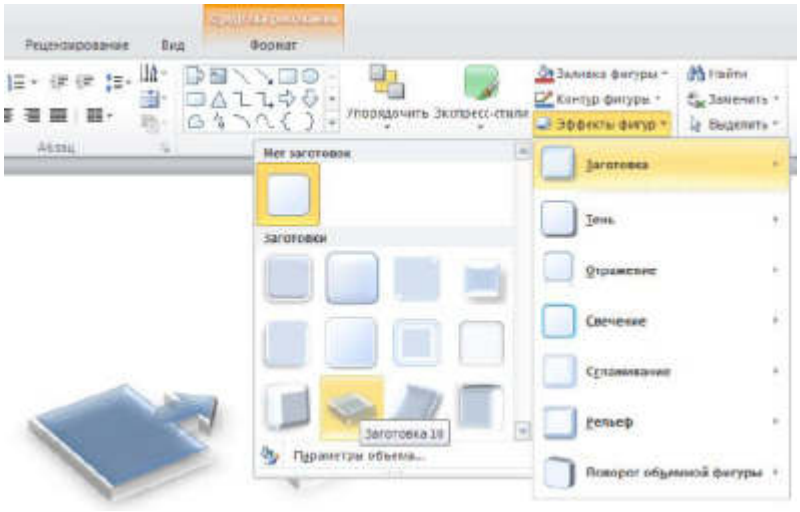


Рис. 15. Инструментарий 3D графики в PowerPoint 2010

6. Анимация объектов на слайдах

Объекты на слайдах могут появляться и исчезать по ходу изложения материала. Для создания подобных эффектов выделите объект анимации и на вкладке Анимация выберите команду Добавить анимацию (рис. 16). Настраивая анимацию можно сделать так, чтобы каждый объект на слайде появлялся определенным образом в определенный момент.

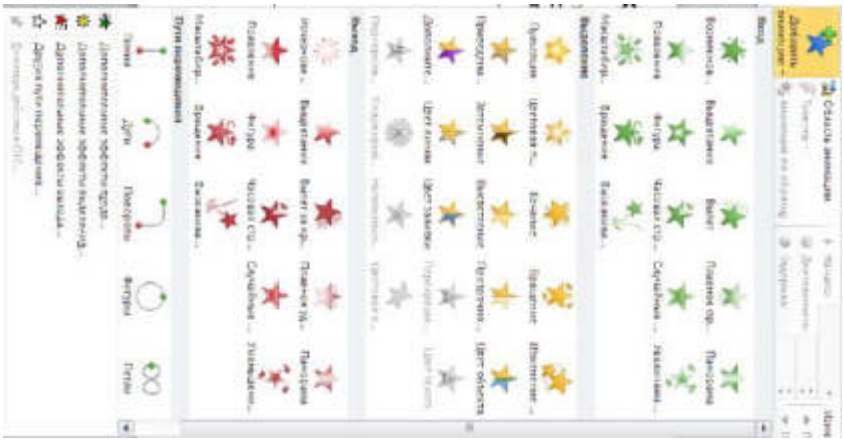


Рис. 9.16. Настройка анимации для фотоснимка

Не следует злоупотреблять анимацией! При создании большого количества разнообразных эффектов с целью приукрасить ваше выступление можно перестараться и отвлечь внимание зрителей от содержательной части вашего доклада.

7. Вставка гиперссылок

Гиперссылки дают возможность электронной презентации быстро менять последовательность слайдов, например, в зависимости от уровня подготовки аудитории или вопросов, возникших в

ходе выступления. Гиперссылка может работать не только в пределах данной презентации, но может также и уводить слушателей в другую презентацию или вообще в другой документ или на сайт Интернет. Гиперссылкой может служить любой объект слайда. Для того чтобы объект стал гиперссылкой, выделите его и выберите из контекстного меню строку Гиперссылка (рис. 17).

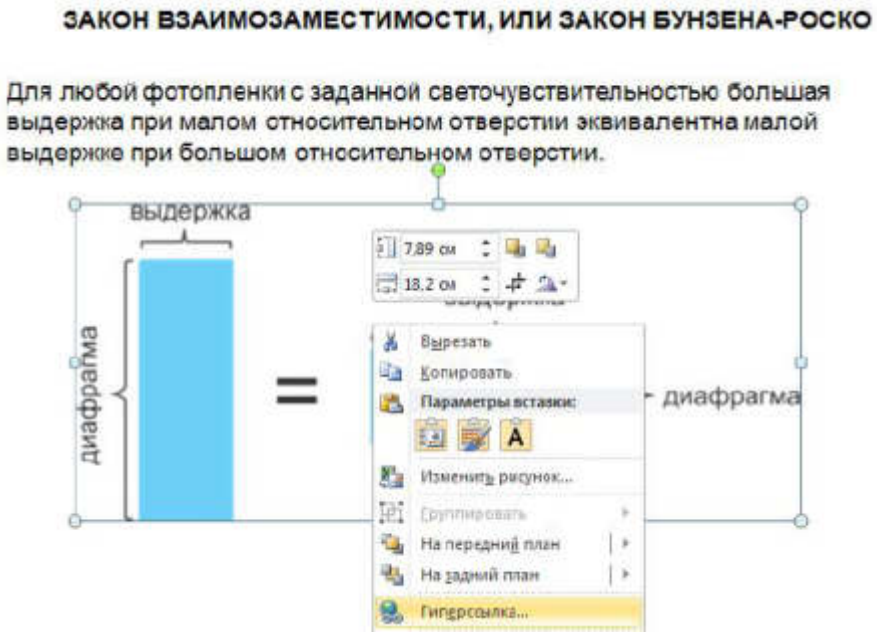


Рис. 17. Вставка гиперссылки

В окне Вставка гиперссылки выберите объект, переход на который хотите запланировать. Лучше всего (если возможно) сделать такой переход двусторонним, то есть на ссылаемом объекте расположить кнопку возврата к исходному документу.

8. Вставка таблиц и диаграмм

Размещение на слайде таких элементов презентации, как таблицы и диаграммы, мало чем отличается от вставки любых других объектов. Однако, пользователь должен понимать, что за каждый из объектов отвечает то приложение, в котором их создают. Если вы умеете создавать таблицы и диаграммы в MS Excel, то вы без проблем создадите их и в PowerPoint.

В PowerPoint на вкладке Конструктор вы можете использовать макеты и/или стили диаграммы (рис. 18). На вкладке Анимация диаграммы PowerPoint можно оживить.

Рис. 18. Инструменты для работы с диаграммами в PowerPoint

Основные инструменты для работы с таблицами расположены в PowerPoint на вкладке Макет (рис. 19).

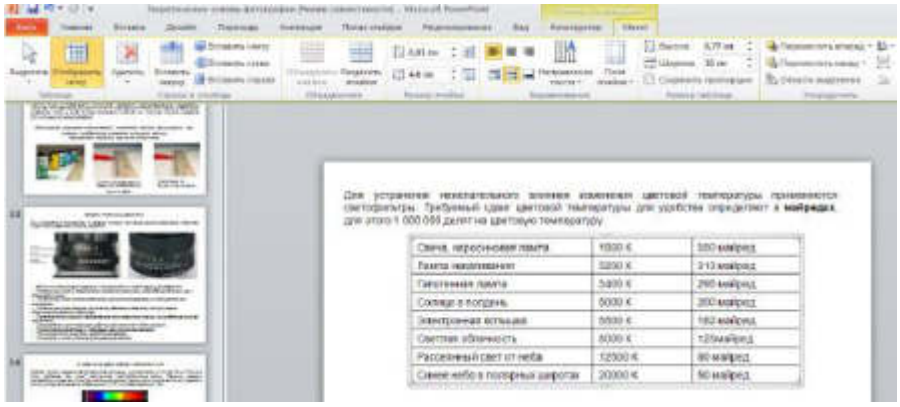


Рис. 19. Инструменты для работы с таблицами в PowerPoint

Однако, часть инструментов для таблиц разбросана по другим вкладкам. Так, на вкладке Вставка вы можете вставить таблицу, а на вкладке Конструктор - выбрать стиль таблицы из образца и/или применить к таблице эффекты оформления (тень, отражение и другие).

9. Исправление ошибок с помощью средства проверки правописания и грамматики

Итак, презентация готова и требуется проверить ошибки и опечатки в тексте презентации. По умолчанию в программе включена автоматическая проверка правописания и программа помечает неправильно написанные слова, так что их можно легко увидеть в ходе работы. Вы можете также воспользоваться системой ручной проверки орфографии PowerPoint, для чего на вкладке Рецензирование выберите команду Орфография (рис. 20). В отобразившемся меню можно найти подходящее исправление или нажать на кнопку Пропустить для игнорирования ошибки.

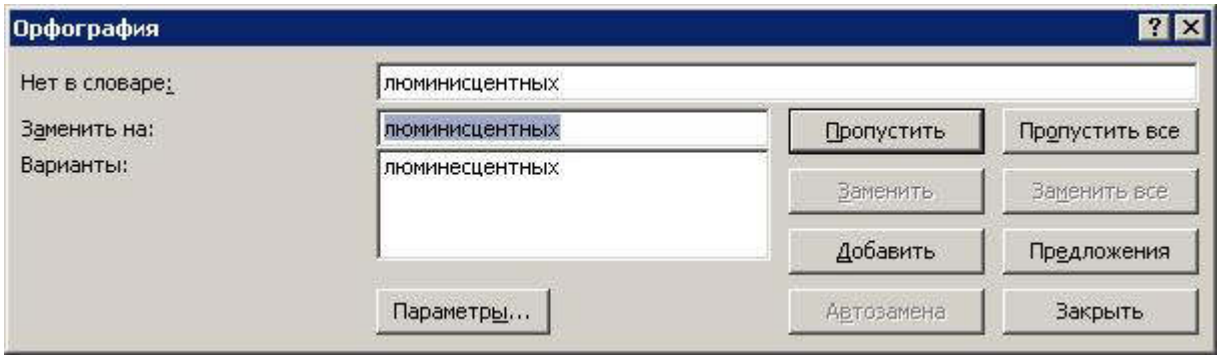


Рис. 20. Проверка правописания и грамматики

10. Сохранение документа

Для сохранения вашей работы выполните команду Файл-Сохранить Как (рис. 21)

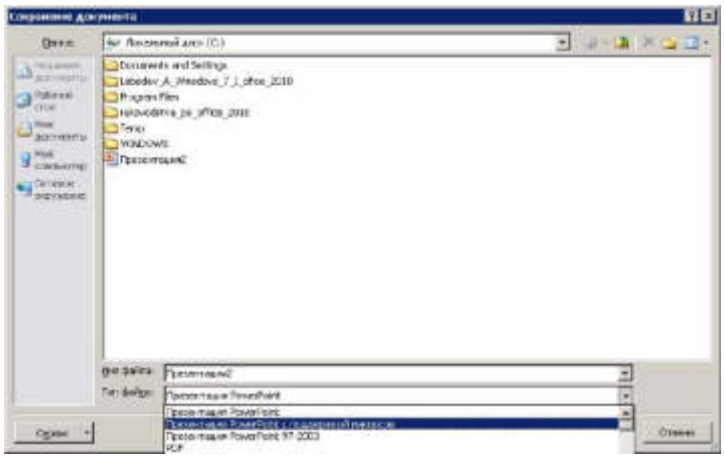


Рис. 9.21. Окно с различными вариантами типов файла

Поговорим о некоторых способах сохранения презентации:

- Презентация PowerPoint - по умолчанию, расширение *.pptx.
- Презентация PowerPoint 97-2003. Расширение *.ppt. Презентации в этом формате можно открывать с помощью PowerPoint 97 и более поздних версий.
- Демонстрация PowerPoint - расширение *.ppsx Презентации, которая всегда будет открываться в режиме Показ слайдов, а не в обычном режиме.
- Рисунок в формате GIF, расширение *.gif. Подобный формат ограничен в цвете (до 256 цветов), зато мало "весит". GIF поддерживает анимацию.
- Рисунок в формате JPEG, расширение *.jpg. Такой слайд в виде графического объекта предназначен для использования на веб-страницах. Формат JPEG поддерживает 16 миллионов цветов и лучше всего подходит для фотографий и сложных изображений.
- Рисунок в формате Portable Network Graphics, расширение *.png. Такой слайд в виде графического объекта также предназначен для использования на веб-страницах. Можно сохранять, восстанавливать и повторно сохранять изображение в формате PNG без потери качества. PNG, в отличие от GIF, не поддерживает анимацию.
- Рисунок в формате TIFF (Tag Image File Format), расширение *.tif. Это полиграфический формат. Рисунки в формате TIFF могут иметь любое разрешение, могут быть цветными, черно-белыми или представленными в оттенках серого.
- Структура, RTF - расширение *.rtf. Такая презентация сохраняется в виде документа, содержащего только текст. Текст из области заметок не включается.
- PDF - Portable Document Format (PDF) - кроссплатформенный формат электронных документов, созданный фирмой Adobe Systems с использованием ряда возможностей языка PostScript. В первую очередь предназначен для представления в электронном виде полиграфической продукции, - значительное количество современного профессионального печатного оборудования может обрабатывать PDF непосредственно. Для просмотра

можно использовать официальную бесплатную программу Adobe Reader, а также программы сторонних разработчиков. Дополнительная полезная информация (Приложение 1)

Таблица 9.1. Горячие клавиши MICROSOFT POWERPOINT	
Сочетание клавиш	Функция
Действия с документом	
Ctrl + O	Открыть презентацию
Ctrl + N	Новая презентация
Ctrl + P	Печать
Ctrl + S	Сохранить
F1	Справка
F12	Сохранить как
Действия с презентациями	
F5	Начать показ слайдов
Enter или стрелка вниз	Следующий слайд
Backspace или стрелка вверх	Предыдущий слайд
Ctrl + P	Активировать перо для примечаний
Ctrl + A	Отключить перо для примечаний
ESC	Завершить показ слайдов
Форматирование	
Ctrl + A	Выделить все
F7 Проверка правописания	
Ctrl + C	Копировать
Ctrl + X	Вырезать
Ctrl + V	Вставить
Ctrl + Z	Отменить
Ctrl + Y	Повторить

Ctrl + B	Жирный
Ctrl + I	Курсив
Ctrl + U	Подчеркивание
Ctrl + L	Выровнять по левой стороне
Ctrl + E	Выровнять по центру
Ctrl + R	Выровнять по правой стороне
Редактирование	
Ctrl + F	Поиск
Ctrl + H	Замена
Ctrl + K	Вставить гиперссылку
Ctrl + M	Новый слайд

Контрольные вопросы.

1. В каких случаях создаются презентации?
2. Какие существует видов шаблонов?
3. Способы создания (после титульного) последующих слайдов презентации?
4. Как повернуть текст на слайде?
5. Как создать красивые заголовки для слайдов?
6. Как объединять и разгруппировать объекты?
7. Для чего применяются эффекты для автофигур?
8. Понятие анимации на слайдах
9. Понятие гиперссылок

Отчет о практической работе должен содержать:

1. Тему работы
2. Цель
3. Оснащение
5. Результаты практической части работы.
6. Ответы на контрольные вопросы

Практическая работа №15 (1час)

Тема: **Работа в программе обработки видео файлов**

Цель: Научиться находить документ, пользоваться гиперссылками в документе.

Оснащение работы: системный блок, монитор, мышь, принтер, сканер, мультимедиа проектор, ОС Windows, драйверы.

План:

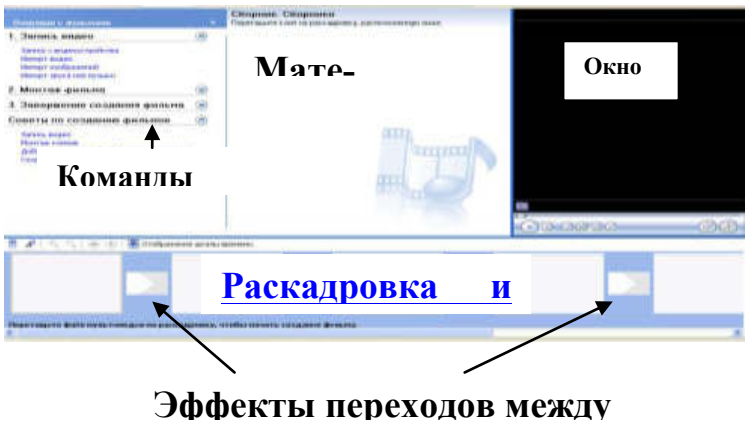
1. Изучение теоретического материала
2. Выполнение заданий для практической работы
3. Подготовка отчета о лабораторной работе (домашнее задание)

Теоретическая часть

Мультимедиа – устройства, позволяющие представлять информацию в аудио и видеовиде.

Мультимедийные программы – программные средства, позволяющие обрабатывать аудио и видеоинформацию

Окно программы Windows Movie Maker



Область, в которой создаются и монтируются проекты, отображается в двух видах: на раскадровке и на шкале времени. В процессе создания фильма можно переключаться между этими двумя видами.

Раскадровка

Раскадровка является видом по умолчанию в программе Windows Movie Maker. Раскадровку можно использовать для просмотра и изменения последовательности клипов проекта. Кроме того, в этом виде можно просмотреть все добавленные видеоэффекты и видеопереходы.

Шкала времени

Шкала времени позволяет просматривать и изменять временные параметры клипов проекта. С помощью кнопок на шкале времени можно выполнять такие операции, как изменение вида проекта, увеличение или уменьшение деталей проекта, запись комментария или настройка уровня звука. Чтобы вырезать нежелательные части клипа, используйте маркеры монтажа, которые отображаются при выборе клипа. Проект определяют все клипы, отображаемые на шкале времени.

Видео

Видеодорожка позволяет узнать, какие видеоклипы, изображения или названия были добавлены в проект. Можно развернуть видеодорожку, чтобы отобразить соответствующее звуковое сопровождение видео, а также все добавленные видеопереходы. Если добавить видеоэффекты в изображение, видео или название, на клипах появится маленький значок, указывающий на то, что в этот клип добавлен видеоэффект.

Аудио

Звуковая дорожка позволяет просмотреть звук, который включен во все видеоклипы, добавленные в проект. Как и дорожка перехода, звуковая дорожка отображается только в том случае, если развернута видеодорожка.

Практическая часть:

СОЗДАНИЕ ВИДЕОКЛИПА.

1. В левом меню выбрать команду «Импорт Видео» или «Импорт изображений», или «Импорт звука». Указать полное имя файла. Полученные фрагменты отобразятся в окне «Материалы для клипа».
2. «Перетащить» фрагменты на шкалу Раскадровки, каждый фрагмент в отдельный кадр.
3. Для корректировки видеоизображения Перейти в режим «Шкалы времени». Обрезать ненужные фрагменты.
4. Перейти в режим Раскадровки, добавить эффекты из левого меню «Просмотр Видеоэффектов», «Просмотр Видеопереходов»
5. При необходимости создать название и титры.
6. Сохранить фильм:
   a. Файл — Сохранить файл фильма
   b. Указать куда будет сохраняться фильм.
   c. Указать имя файла и место сохранения.

ХОД РАБОТЫ

1. Ознакомиться с теорией
2. Выполнить задания
3. Ответить на контрольные вопросы

ЗАДАНИЯ.

1. Создать видеоклип по предложенной преподавателем теме.
2. Использовать в клипе видеоматериал, фотографии, музыку, голосовое сопровождение.
3. Использовать в клипе различные эффекты
4. Сохранить фильм в своей личной папке.

Контрольные вопросы:

1. Понятие мультимедиа.
2. Виды мультимедийных устройств.
3. Понятие мультимедийных программ.
4. Виды мультимедийных программ.
5. Возможности программы Windows Movie Maker.
6. Технология создания видеоклипа.
7. Назначение Шкалы времени.
8. Назначение Шкалы раскадровки.
9. Дорожки, входящие в состав Шкалы времени.
10. Способы сохранения фильма.

Отчет о практической работе должен содержать:

1. Тему работы
2. Цель
3. Оснащение
5. Результаты практической части работы.
6. Ответы на контрольные вопросы

Практическая работа № 16 (1час)

Тема: **Поиск заданной информации в Интернете** ***Сохранение найденной информации по заданным условиям***

Цель: Изучение способов поиска информации в тексте, в файловых структурах, в базах данных, в сети Интернет

Оснащение работы: системный блок, монитор, мышь, принтер, сканер, мультимедиа проектор, ОС Windows, драйверы.

План:

1. Изучение теоретического материала
2. Выполнение заданий для практической работы
3. Подготовка отчета о лабораторной работе (домашнее задание)
4. Изучить основные понятия поиска информации
5. Выполнить поиск информационного объекта в тексте
6. Выполнить поиск информационного объекта в файловых структурах
7. Выполнить поиск информационного объекта в сети Интернет
8. Ответить на контрольные вопросы

Теоретическая часть:

Поиск информации - задача, которую человечество решает уже многие столетия. По мере роста объема информационных ресурсов, потенциально доступных одному человеку, были выработаны все более изощренные и совершенные поисковые средства и приемы, позволяющие найти необходимый документ. Обширные возможности для работы с большими массивами информации дают поисковые сервисы Interner.

При наличии первичных сведений по теме поиска, документы можно разыскивать поисковых системах. При этом следует различать приемы *простого, расширенного, контекстного и специального поиска*.

Под *простым поиском* понимается поиск Web-ресурсов по одному или нескольким ключевым словам. Недостаток простого поиска заключается в том, что обычно он выдает слишком много документов, среди которых трудно выбрать наиболее подходящие.

При использовании *расширенного поиска* ключевые слова связывают между собой операторами логических отношений. Расширенный поиск применяют в тех случаях, когда приемы простого поиска дают слишком много результатов. С помощью логических отношений поисковое задание формируют так, чтобы более точно детализировать задание и ограничить область отбора, например по дате публикации или типу данных.

Контекстный поиск – это поиск по точной фразе. Он удобен для реферативного поиска информации, но доступен далеко не во всех поисковых системах. Прежде всего, чтобы обеспечивать такую возможность, система должна работать не только индексированными файлами, но и с полноценными образами Web-страниц. Эта операция достаточно медленная, и ее выполняют не все поисковые системы.

Специальный поиск применяют при розыске Web-страниц, содержащих ссылки на заданные адреса URL, содержащих заданные данные в служебных полях, например в поле заголовка и т.п.

Расширенный поиск. Кроме средства простого поиска обычно поисковые службы предоставляют средства расширенного поиска. Эти средства позволяют более точно формулировать поисковое задание, но требуют определенного опыта и работают заметно медленнее. В большинстве поисковых систем команды расширенного поиска формируются с помощью логических команд. Удобство использования логических команд в частности связано с тем, что команды простого поиска у многих поисковых систем реализованы по-разному. Каждая система стремится сделать средства простого поиска наиболее удобными, а средства расширенного поиска – наиболее стандартными. Тем не менее, для обозначения логических операторов в различных поисковых системах используются разные обозначения. Поэтому желательно перед осуществлением расширенного поиска желательно изучить синтаксис поисковых запросов выбранной поисковой системы.

Рассмотрим подробнее операторы логических отношений (логические команды).

Логическая оператор OR (ИЛИ) служит для формирования поискового запроса, если искомый текст должен содержать хотя бы один из терминов, соединенных данным оператором. Этот оператор в различных поисковых системах может обозначаться одним из следующих способов: | ; OR; ИЛИ.

Например, результат запроса «Чёрное OR море» - будет представлен списком ссылок на документы, в которых есть слово «Чёрное», или слово «море», или оба этих слова вместе.

В некоторых поисковых системах, как отмечалось выше, по умолчанию ключевые слова в запросе связаны именно этим логическим отношением.

С помощью логического оператора AND (И) осуществляется поиск документов, содержащих все термины, соединенные данным оператором. Этот оператор может обозначаться одним из следующих способов: +; AND; &; И.

Например, по запросу – «Чёрное AND море» - будут найдены документы, в которых содержатся слова «черное» и «море».

Логической оператор NOT (НЕ) позволяет производить поиск документов, в тексте которых отсутствуют термины, следующие за данным оператором. Этот оператор может обозначаться одним из следующих способов: not; !; ~; НЕ.

Например, по запросу – «Чёрное NOT море», результат - документы, в которых есть слово «Чёрное» и нет слова «море».

С помощью логических операций можно создавать достаточно сложные запросы. Запрос из нескольких слов, перемежающихся операторами, будет истолкован в соответствии с их приоритетом. Операторы AND и NOT традиционно имеют более высокий приоритет, поэтому запрос из нескольких слов при обработке сначала группируется по операторам AND и NOT, и лишь потом по операторам OR. Например, по запросу «Чёрное AND море OR Крым» будут найдены документы, либо содержащие обязательно слова: «Чёрное» и «море», либо слово «Крым», либо все три слова.

Изменить порядок группировки можно использованием скобок. Оператор, стоящий в скобках, будет выполняться в первую очередь Использование скобок позволяет строить вложенные запросы и передавать их операторам в качестве аргументов. Так по запросу «Чёрное AND (море OR Крым)» будут найдены документы, в которых обязательно содержится слово «Чёрное» и одно из двух слов «море» или «Крым».

С помощью вложенных запросов можно значительно ограничивать область отбора, освобождая результирующий список от ненужных ссылок. Так, например, если нас интересует информация об отдыхе на юге на море, но исключительно на российском побережье, то можно попробовать использовать примерно такой запрос – «отдых AND ((Азовское OR Чёрное) AND море) NOT (Крым OR Турция OR Болгария)».

Использование круглых скобок для управления порядком исполнения задания на поиск разрешается большинством крупнейших поисковых систем.

РАСШИРЕННЫЙ ПОИСК YANDEX.RU

Оператор	Описание
"	Обнаруживает точные слова в кавычках или фразы
|	Найти любое из слов. Достаточно поставить между словами символ |, и вы получите страницы, где содержится хоть одно из слов запроса.
~	Исключает страницы, содержащие слово или фразу.
()	Вы можете строить сколь угодно сложные конструкции, подставляя в каждом из операторов вместо отдельного слова целые выражения. Чтобы Яндекс при этом правильно понимал вас, заключайте выражения в круглые скобки.
&	Ограничить поиск страницами, где слова запроса находятся в пределах предложения
&&	Если вам нужны документы, где присутствуют заданные слова — неважно, на каком расстоянии и в каком порядке — соедините их оператором

!	Слова с большой и маленькой буквы считаются разными формами одного слова, поэтому все равно, какой регистр использовать в запросе. Исключением является оператор точной формы. Это полезно, если искомое имя собственное совпадает с распространенным словосочетанием, например, группа !Черный кофе. Все слова, которые вы приводите в запросе, по умолчанию ищутся с учетом морфологии. Чтобы отключить ее, используйте оператор ! перед словом (без пробела).
/	Вы можете указать максимально допустимое расстояние между двумя любыми словами запроса, поставив после первого слова символ /, сразу за которым идет число, означающее расстояние.
*	Замена части слова. Журналист*
?	Замена любого символа. Журналистик?

РАСШИРЕННЫЙ ПОИСК RAMBLER.RU

Оператор	Описание
+	Используется для включения общих слов
"	Слова запроса, заключенного в двойные кавычки, ищутся в документах именно в том порядке и в тех формах, в которых они встретились в запросе. Таким образом, двойные кавычки можно использовать и просто для поиска слова в заданной форме (по умолчанию слова находятся во всех формах).
NOT	Оператор NOT позволяет сформировать запрос, которому отвечают документы, удовлетворяющие левой части запроса и не удовлетворяющие правой. Так, результатом поиска по запросу собака NOT кошка будут все документы, в которых есть слово "собака" и нет слова "кошка". Это особенно полезно в случаях, если искомая словоформа является одновременно формой другого слова: Женя - имя собственное, а также деепричастие. Если мы ищем именно человека по имени Женя, в запросе можно написать Женя NOT женить.
&&	Два запроса, соединенные оператором &&, образуют сложный запрос, которому удовлетворяют только те документы, которые одновременно удовлетворяют обоим этим запросам. Иными словами, по запросу собака && кошка найдутся только те документы, которые содержат и слово "собака", и слово "кошка".
||	Сложному запросу, состоящему из двух запросов, соединенных оператором ||, удовлетворяют все документы, удовлетворяющие хотя бы одному из этих двух запросов. По запросу *собака* || *кошка* найдутся документы, в которых есть хотя

	бы одно из двух слов - слово "собака" или слово "кошка" (или оба эти слова вместе).
()	Использование скобок позволяет строить вложенные запросы и передавать их операторам в качестве аргументов, а также перекрывать приоритеты операторов, принятые по умолчанию.

Организация специального поиска. С помощью средств специального поиска можно осуществлять поиск документов: содержащих искомые данные в различных полях (заголовке, ключевых словах, описании); расположенных на каком-либо сайте; содержащих те или иные ссылки и т.п. Рассмотрим подробнее некоторые возможности специального поиска.

Многие поисковые системы позволяют разыскивать Web-документы по тексту, содержащемуся в заголовках. Поиск по заголовкам существенно уменьшает количество найденных ссылок, но очень точно выводит на нужные материалы. Ведь каждая Web-страница может иметь заголовок, если её автор не поленился его создать. И заголовок Web-страницы обычно точно характеризует тему материала, который содержится на ней.

Например, если вам нужна информация о дистанционном обучении, то целесообразно искать страницы, в которых это сочетание присутствует в заголовке. Таким образом, в отличие от простого запроса мы отсекаем те документы, где эти слова не являются значимыми, т.е. не определяют тему статьи.

Оператором или командой такого поиска является title, Этот оператор может обозначаться одним из следующих способов: title:, t:, title =, $title и т.п. После оператора следую ключевые слова. В некоторых поисковых системах ключевые слова следует заключать в скобки, в других они пишутся без скобок. Например, в Яndex команда поиска в заголовке записывается так: $title (очное обучение).

Поиск по сайту. С помощью поисковых систем можно осуществлять поиск информации не во всем Web пространстве, а на каком то конкретном сайте (если конечно последний проиндексирован поисковой системой). Соответствующий оператор может обозначаться следующим одним из следующих способов: url=, url:, u:, #url=". Далее следует адрес Web-узла, некоторые системы требуют заключения адреса в кавычки.

Если в запросе просто записать данный оператор с адресом какого-либо Web-узла, то будет получен список документов, проиндексированных поисковой системой на данном сайте. Но этот оператор можно комбинировать с другими, тем самым, осуществляя поиск информации по всем правилам построения запросов на данном сайте. Например, если мы хотим найти информацию об очном обучении на сайте БелГУ, то запрос в поисковой системе Апорт будет выглядеть следующим способом: URL=http://www.bsu.edu.ru + “очное обучение”.

Кроме того, поисковые системы могут предлагать другие возможности специального поиска: поиск по тексту ссылок, поиск в описании документа, поиск в списке ключевых слов Web-страниц, поиск по подписям к рисункам и т.п. Следует знать, что синтаксис поисковых запросов, да и состав доступных операторов, отличается в различных поисковых указателях. Поэтому перед осуществлением поиска в той или иной поисковой системе, следует изучить страницу справки по поиску в данной системе.

СПЕЦИАЛЬНЫЙ ПОИСК APORT.RU

Оператор	Описание
title=	Указанное после равенства слово или конструкция в круглых скобках должны искаться в заголовках документов (перед круглыми скобками знак равенства можно опускать). По запросу title=(папа или мама) будут найдены документы, содержащие в заголовке слово папа или слово мама, или оба слова одновременно.
anchor=	Указанное после равенства слово или конструкция в круглых скобках должны искаться в тексте ссылок. По запросу anchor=(ненавижу Интернет) будут найдены документы, в тексте ссылок на которые, встречаются оба слова: ненавидеть и Интернет.
text=	Указанное после равенства слово или конструкция в круглых скобках должны искаться только в обычном тексте. По умолчанию слова запроса ищутся как в тексте, так и во всех указанных выше полях. Чтобы искать только по тексту, выдаваемому в основное окно браузера, следует использовать данный оператор. По запросу text=(оглавление или содержание) будут найдены документы, в которых любое из указанных слов встречается в пределах основного текста документа.

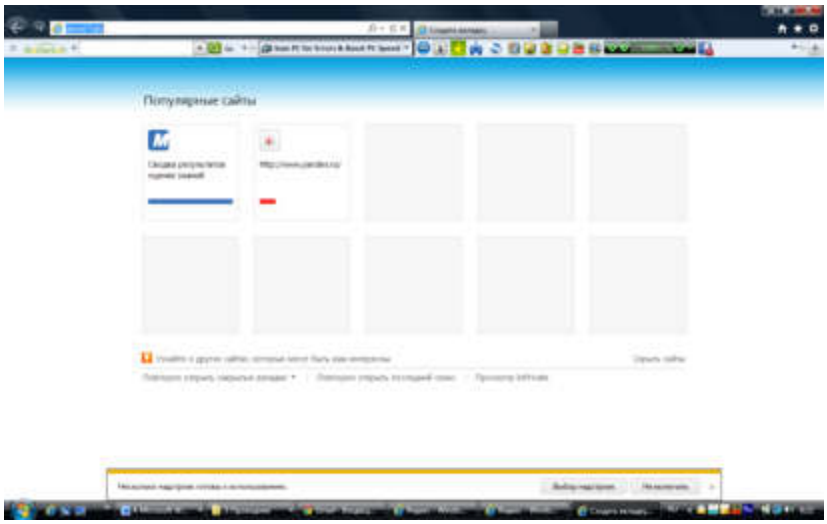
Практическая часть:

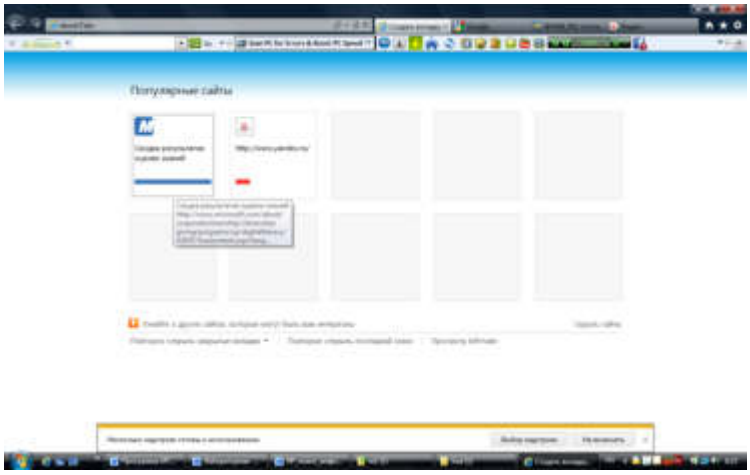
Задание:

Найти, как называется самое большое пресноводное озеро в мире.

Порядок выполнения.

- Запустил обозреватель MS Internet Explorer.
- В адресной строке набрал адрес поискового WWW-сервера.
- Открыл новое окно браузера, выполнив последовательность команд в главном меню Файл - Создать - Окно или использовал сочетание клавиш Ctrl+N.
- Повторил п.п. 2, 3 не менее четырех раз. В разные окна браузера загрузил главные страницы поисковых машин.
- Сравнил интерфейсы поисковых WWW-серверов.





- С помощью справочных систем познакомился с основными средствами простого и расширенного поиска.
- Организовал поиск, заполнил таблицу и прокомментировал результаты поиска:

Ключевая фраза	Результаты поиска			
	Yandex	Google	Rambler	Апорт
информационные технологии в образовании	Нашлось 84 млн ответов	примерно 1 390 000	Начало формы По запросу найдено 297 тыс. сайтов, 28 млн. документов	92 736 документов

"информационные технологии в образовании"	Нашлась 261 тыс. ответов	примерно 834 000	По запросу найдено 12 тыс. сайтов, 267 тыс. документов	3 665 (534 сайта) документов
педагогические технологии личностно-ориентированного обучения	Нашлась 431 тыс. ответов	примерно 232 000	найдено 19 тыс. сайтов, 200 тыс. документов	842 (235 сайтов) документа

- Дополнил таблицу самостоятельно построенными запросами.
- Познакомился с избранными документами и оцените их релевантность (смысловое соответствие результатов поиска указанному запросу). Организовал поиск интересующей меня информации и внес результаты в таблицу.

Контрольные вопросы

1. Как выполнить поиск информационного объекта в текстовом процессоре MS Word?
2. Как выполнить поиск информационного объекта в файловых структурах Windows?
3. Перечислите известные вам поисковые машины.
4. Для чего в некоторых поисковых системах используется расширенный поиск?
5. Каким логическим оператором связаны ключевые слова в простом запросе в рассмотренных поисковых машинах?
6. Как в поисковой машине Яndex осуществить поиск точной формы слов?
7. Какие области поиска можно определить в Яndex?
8. Как в Яndex указать расстояние между ключевыми словами?
9. Что обозначает двойное & (&&) в запросе Яndex?
10. Как правильно в запросе Яndex поставить знаки «+» и «–» ?
11. Как в поисковых машинах осуществить поиск по точной фразе?
12. Для чего используются скобки в запросах?
13. Что обозначает двойное | (||) в запросе Rambler?

Отчет о практической работе должен содержать:

1. Тему работы
2. Цель
3. Оснащение

5. Результаты практической части работы.

6. Ответы на контрольные вопросы

Практическая работа №17 (1 час)

Тема Хранение и скачивание файлов из сети Интернет (FTP).

Цель: Научиться рационально использовать интернет-сервисы размещения и хранения файлов.

Ход выполнения работы.

Пользователь, не раз сталкивается с ситуацией, когда нужного файла нет под рукой, по причине того, что ты забыл флешку дома и т.п. Или ещё хуже, вышедший из строя жесткий диск, унес с собой на свалку очень важную информацию. Что бы избежать этих моментов применяется хранение файлов в интернете.

Сначала нужно подобрать службу хранения файлов с удовлетворяющими вас условиями. Такие сервисы различаются по времени хранения, условиям скачивания файлов и доступностью из разных точек планеты - есть и другие различия, но эти наиболее существенны.

Время хранения может быть от двух недель (например, iFolder) до бесконечности (Rapidshare). Но надо обязательно ознакомиться с условиями конкретного сервиса - тот же iFolder, ограничивая срок хранения двумя неделями по умолчанию, дает возможность продлить его. А Rapidshare, не ограничивая срока, тем не менее удалит файл, если в течение длительного времени к нему никто не будет обращаться.

Главное различие в условиях скачивания в том, что часть этих сервисов может требовать оплаты за скачивание вашего файла. Причем некоторые из них могут делиться частью оплаты за каждое скачивание с вами. Другие сервисы бесплатны, но требуют перед скачиванием просмотра рекламы спонсоров или просто ожидания очереди в течение некоторого времени. Кроме того, бесплатное скачивание в большинстве случаев имеет ограниченную скорость и ограниченное число файлов в сутки.

Один из таких сервисов является – Яндекс.Диск.

Сервис этот привлекает прежде всего тем, что вы без особого труда сможете увеличить диск для хранения ваших данных до 10 Гб. Бесплатно иметь в сети интернет хранилище такого объема, очень даже неплохо и проблема с резервным копированием нужных документов решена.

1. Задание 1

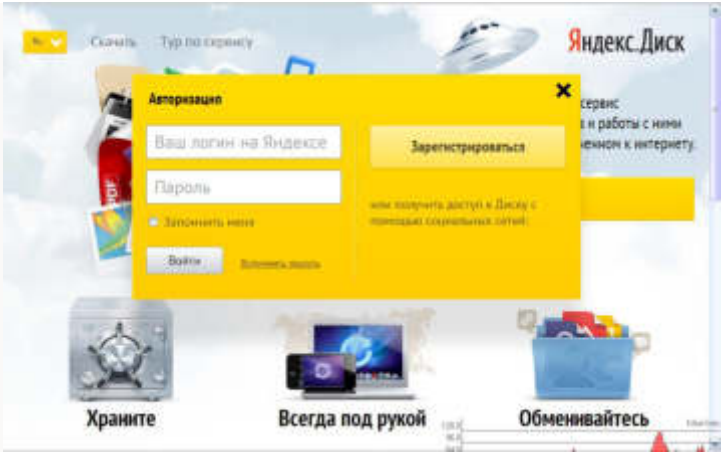
Размещение файлов на сервисе Яндекс-Диск

1.1. В браузере вводим URL адрес http://disk.yandex.ru/

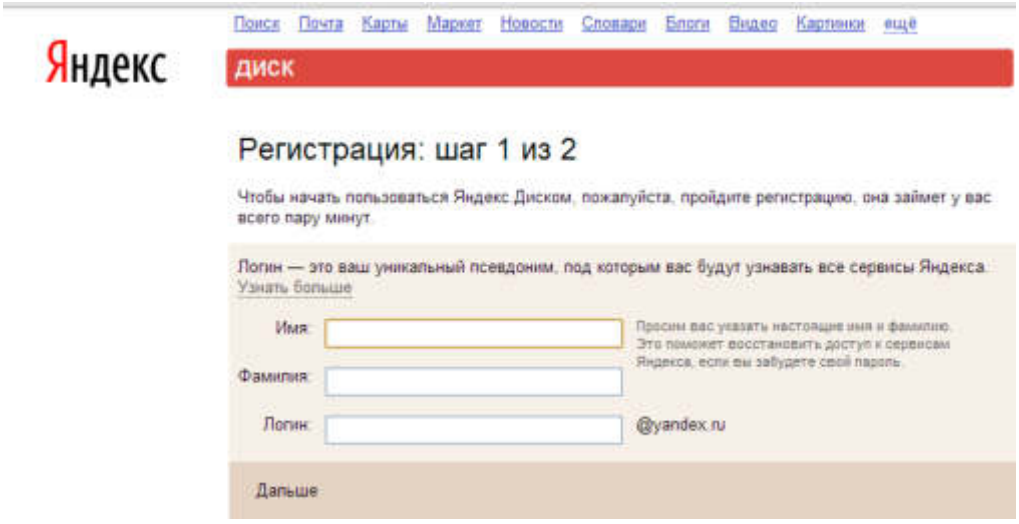
1.2. Выбираем в открывшейся web-странице пункт ВОЙ-

ТИ

1.3. Выбираем пункт Регистрация.



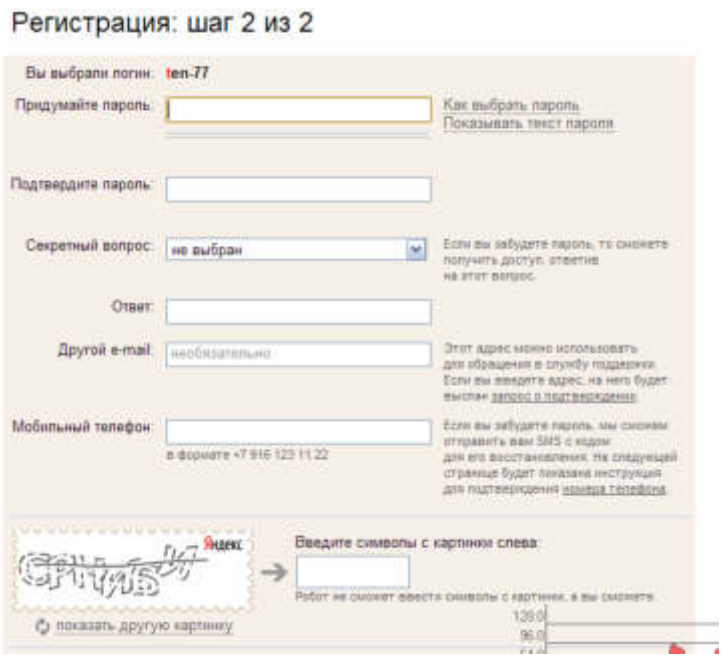
1.4. В окне регистрации введите свои данные



- Имя
- Фамилию
- Придумайте логин – Вводить на латинском языке!!

1.5. Нажмите Далее

1.6. В окне регистрации введите Данные:



- Пароль
- Повторите пароль
- Выберите секретный вопрос
- Введите ответ на секретный вопрос
- Введите номер мобильного телефона
- Введите секретный код с картинки

1.7. После регистрации нажимаете ссылку Начать пользоваться Яндекс.Диском

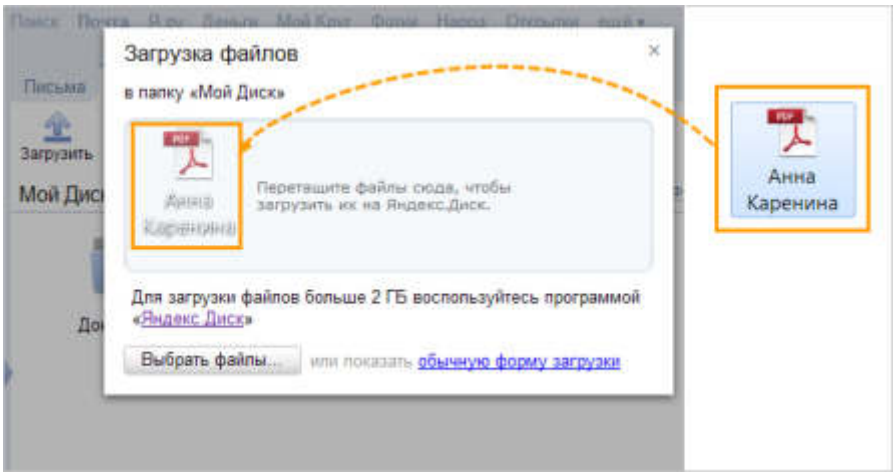
1.8. Появиться окно Вашего почтового ящика.

1.9. Выберите пункт верхнего меню ЕЩЕ

1.10. В раскрывшемся списке выберите пункт ДИСК

1.11. Нажмите кнопку Загрузить.

1.12. Перетащите файлы на страницу или выберите их, нажав кнопку Выбрать файлы.



1.13. После добавления файлов не забудьте выйти из сервиса!

Задание 2 Самостоятельно разместите файлы на сервисе Ifolder.ru

Использование сервиса Ifolder

После выбора сервиса хранения останется чисто техническая часть операции из нескольких шагов. Для начала перейдите на сайт сервиса, например - http://ifolder.ru

Этот сервис не требует регистрации для загрузки вашего файла, можете прямо на главной странице щелкнуть кнопку (или поле ввода) под надписью «Загрузить файл». Откроется диалоговое окно, в котором вам надо найти подготовленный к загрузке файл и нажать кнопку «Открыть». В поле ввода появится имя вашего файла и красный крестик рядом с ним - если вы передумаете, можете удалить этот файл из очереди на загрузку, щелкнув этот самый крестик. Кроме того, ниже этого поля появится еще одно такое же - это на случай, если вам надо загрузить больше одного файла.

Закончив с составлением очереди файлов, щелкните надпись «Загрузить». Файл (или файлы) загрузится на сервер и вам будет представлен отчет о том какие файлы и какого размера были загружены. Здесь же у вас будет возможность ввести текстовое описание этого файла и прикрепить к нему картинку-обложку. Кроме того здесь вы сможете установить пароль - в этом случае скачать файл смогут лишь те, кто введет этот пароль на странице скачивания на сайте сервиса. На этой же странице вам надо ввести свой email-адрес, и капча-код, а затем нажать надпись «подтвердить».

На следующей странице вы получите подтверждение завершения операции, ссылку для управления загруженным файлом и ссылку для его скачивания, которую можете разместить где-либо в сети или разослать тем, кому считаете нужным.

Практическая работа № 18 (1 час)

Тема: **Работа с языками разметки HTML**

Цель: использовать теги для разметки текста выполнять форматирование текста,создавать списки.

Оснащение работы: системный блок, монитор, мышь, принтер, сканер, мультимедиа проектор, ОС Windows, драйверы.

План:

1. Изучение теоретического материала
2. Выполнение заданий для практической работы
3. Подготовка отчета о лабораторной работе (домашнее задание)
4. Ответить на контрольные вопросы

Теоретическая часть:

HTML (от англ. HyperText Markup Language — «язык разметки гипертекста») – стандартный язык разметки документов во Всемирной паутине. Большинство веб-страниц создаются при помощи языка HTML (или XHTML). Язык HTML интерпретируется браузерами и отображается в виде документа в удобной для человека форме.

HTML является приложением («частным случаем») SGML (стандартного обобщённого языка разметки) и соответствует международному стандарту ISO 8879. XHTML же является приложением XML.

Язык HTML был разработан британским учёным Тимом Бернерсом-Ли приблизительно в 1986-1991 годах в стенах Европейского Центра ядерных исследований в Женеве (Швейцария). HTML создавался как язык для обмена научной и технической документацией, пригодный для использования людьми, не являющимися специалистами в области вёрстки. HTML успешно справлялся с проблемой сложности SGML путём определения небольшого набора структурных и семантических элементов – дескрипторов. Дескрипторы также часто называют «тегами». С помощью HTML можно легко создать относительно простой, но красиво оформленный документ. Помимо упрощения структуры документа, в HTML внесена поддержка гипертекста. Мультимедийные возможности были добавлены позже.

Изначально язык HTML был задуман и создан как средство структурирования и форматирования документов без их привязки к средствам воспроизведения (отображения). В идеале, текст с разметкой HTML должен был без стилистических и структурных искажений воспроизводиться на оборудовании с различной технической оснащённостью (цветной экран современного компьютера, монохромный экран органайзера, ограниченный по размерам экран мобильного телефона или устройства и программы голосового воспроизведения текстов). Однако современное применение HTML очень далеко от его изначальной задачи. Например, тег

<TABLE> предназначен для создания в документах таблиц, но, часто используется и для оформления размещения элементов на странице. С течением времени основная идея платформа независимости языка HTML была принесена в жертву современным потребностям в мультимедийном и графическом оформлении.

Практическая часть:

Задание № 1. Задание простейшего HTML-документа

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

< HTML>

<HEAD>

<TITLE>Мой первый HTML-документ</TITLE>

</HEAD>

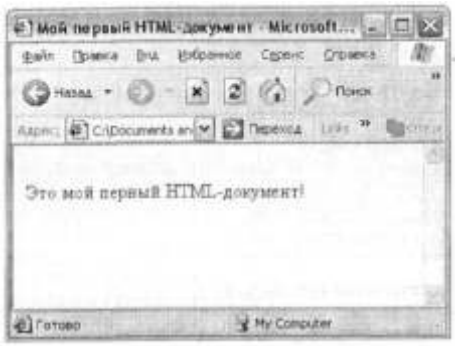
<BODY BGCOLOR="#FFFFFF" TOPMARGIN="30" TEXT="black" LINK="#00FF00" ALINK="#00FF00" VLINK="blue">

Это мой первый HTML-документ!

</BODY>

</HTML>

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере. Результатом должно послужить следующее:



Задание № 2. Использование тега <ACRONYM>

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

HTML>

<HEAD>

<TITLE>Расшифровка аббревиатур с помощью тега ACRONYM </TITLE>

</HEAD>

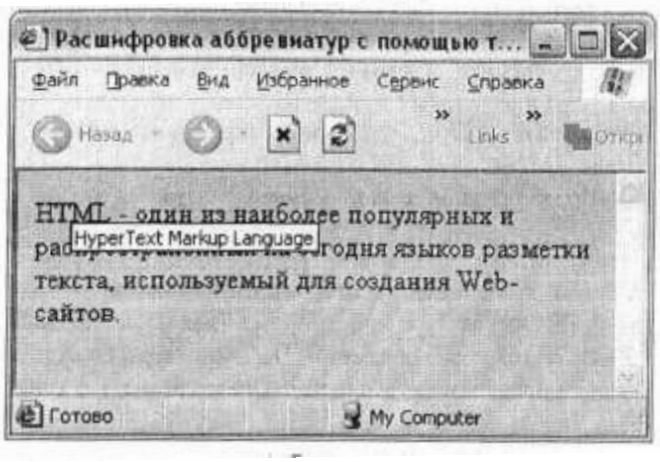
<BODY BGCOLOR="FFFFFF" TEXT="black" LINK="#OOFFOO" ALINK="#OOFFOO" VLINK="blue">

<ACRONYM TITLE="HvperText Markup Lanquage"> HTML </ACRONYM> – один из наиболее популярных и распространенных на сегодня языков разметки текста, используемый для создания Web-сайтов.

</BODY>

<HTML>

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере. Результатом должно послужить следующее:



Задание № 3. Выделение текста курсивом

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

<HTML>

<HEAD>

<TITLE>Выделение текста курсивом </TITLE>

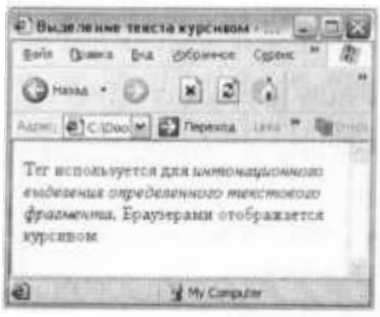
</HEAD>

<BODY> Тег используется для <EM> интонационного выделения определённого текстового фрагмента </EM>. Браузерами отображается курсивом.

</BODY>

<HTML>

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере. Результатом должно послужить следующее:



Задание № 4. Использование тегов <H1>–<H6>

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

<HTML>

<HEAD>

<TITLE>Заголовки H1–H6 </TITLE>

</HEAD>

<BODY>

<H1> Заголовок 1 </H1>

<H2> Заголовок 2 </H2>

<H3> Заголовок 3 </H3>

<H4> Заголовок 4 </H4>

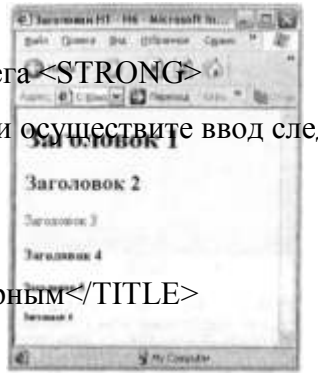
<H5> Заголовок 5 </H5>

<H6> Заголовок 6 </H6>

</BODY>

<HTML>

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере. Результатом должно послужить следующее:



Задание № 5. Использование тега <STRONG>

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

<HTML>

<HEAD>

<TITLE>Выделение текста жирным</TITLE>

</HEAD>

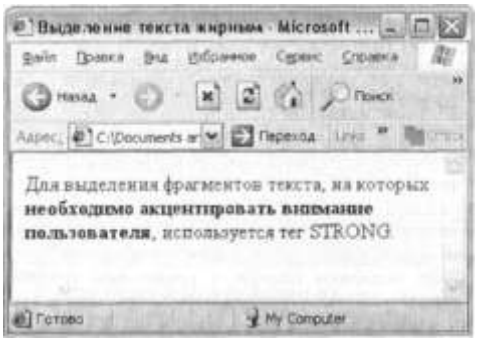
<BODY>

Для выделения фрагментов текста, на которых <STRONG> необходимо акцентировать внимание пользователя</STRONG>, используется тег STRONG.

</BODY>

</HTML>

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере. Результатом должно послужить следующее:



Задание № 6. Использование параметра FACE тега <FONT>

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

<HTML>

<HEAD>

<TITLE> Использование параметра FACE</TITLE>

</HEAD>

<BODY>

<H3>

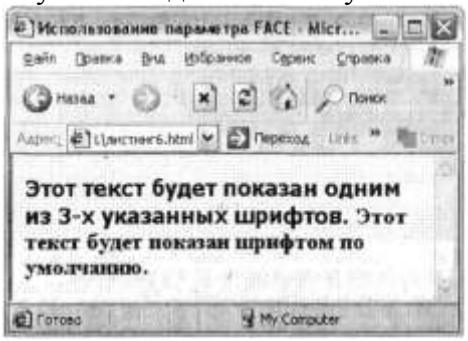
<FONT FACE="Tahoma", "Arial", "Verdana"> Этот текст будет показан шрифтом по умолчанию. </FONT>

</H3>

</BODY>

</HTML>

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере. Результатом должно послужить следующее:



Задание № 7. Использование всех параметров тега <FONT>

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

<HTML>

<HEAD>

<TITLE> Использование параметра FONT </TITLE>

</HEAD>

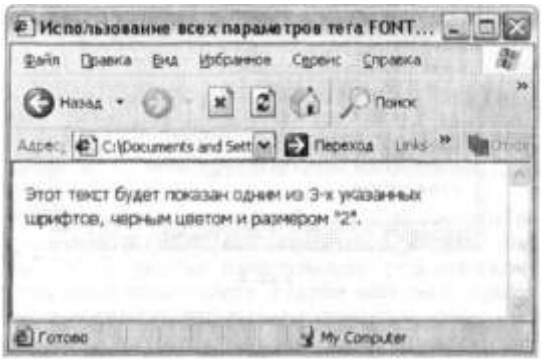
<BODY>

<FONT FACE="Tahoma", "Helvetica" COLOR="#OOOOOO" SIZE="2"> Этот текст будет показан ОДНИМ из 3-х указанных шрифтов, черным цветом и размером "2"</FONT>

</BODY>

</HTML>

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере. Результатом должно послужить следующее:



Задание № 8. Одновременное использование нескольких тегов физического форматирования

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

<HTML>

<HEAD>

<TITLE>

Одновременное использование нескольких тегов физического форматирования

</TITLE>

</HEAD>

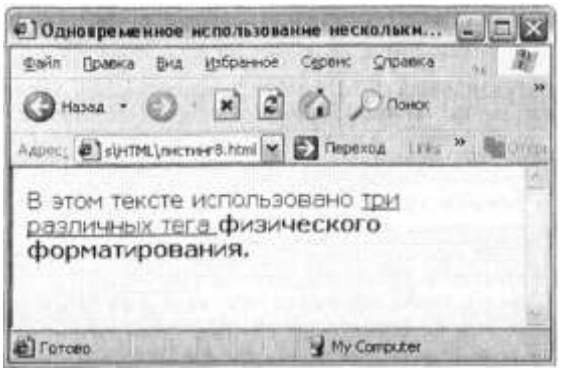
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#OOFFOO" ALINK="#OOFFOO" VLINK="blue">

<FONT FACE="Verdana", "Arial" COLOR="black" SIZE="3"> В этом тексте использовано <U> три различных тега </U> <B> физического форматирования. </B> </FONT>

</BODY>

</HTML>

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере. Результатом должно послужить следующее:



Задание № 9. Использование различных типов выравнивания

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

<HTML>

<HEAD>

<TITLE>Использование различных типов выравнивания </TITLE>

</HEAD>

<BODY>

<P>При создании HTML-документа для обозначения абзаца используется специальный тег <CODE>P</CODE>, который разделяет фрагменты текста вертикальным отступом. </P>

<P ALIGN="CENTER"> Такой тип выравнивания подходит для заголовков документов. </P>

<P ALIGN="RIGHT"> Двойной перевод строки в данном случае не поможет: браузер, интерпретируя код, не воспримет двойной отступ как команду создания абзаца.</P>

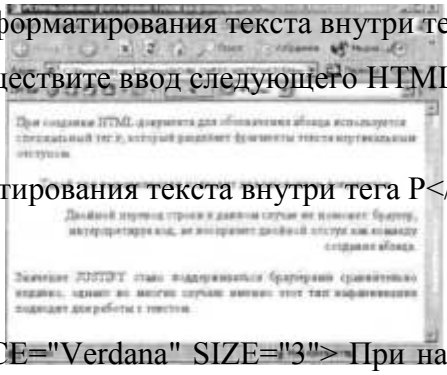
<P ALIGN="JUSTIFY"> Значение JUSTIFY стало поддерживаться браузерами сравнительно недавно, однако во многих случаях именно этот тип выравнивания подходит для работы с текстом.</P>

</BODY>

</HTML>

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере. Результатом должно послужить следующее:

Задание № 10. Использование тегов форматирования текста внутри тега <P>



Запустите программу Блокнот и осуществите ввод следующего HTML кода:<HTML>

<HEAD>

<TITLE>Использование тегов форматирования текста внутри тега P</TITLE>

</HEAD>

<BODY>

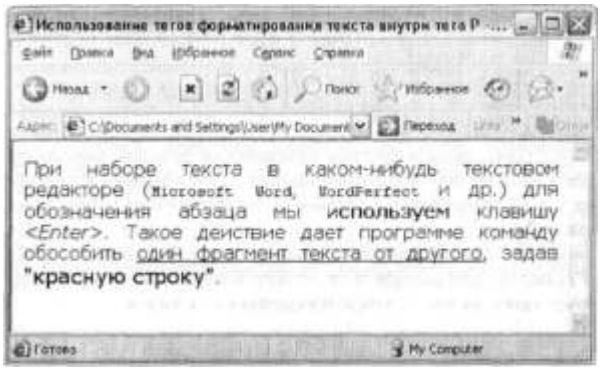
<P ALIGN="JUSTIFY"> <FONT FACE="Verdana" SIZE="3"> При наборе текста в каком-нибудь текстовом редакторе (<TT>Microsoft Word </TT>, <TT>WordPerfect</TT> и др.) для обозначения абзаца мы <STRONG>используем</STRON> клавишу <EM><Enter></EM>. Такое действие даёт программе команду обособить <U>один фрагмент текста от другого, задав <FONT COLOR="red"> <BIG> "красную строку" </BIG></FONT>

</FONT></P>

</BODY>

</HTML>

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере. Результатом должно послужить следующее:



Задание № 11. Создание отступов

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

<HTML>

<HEAD>

<TITLE>Создание отступов</TITLE>

</HEAD>

<BODY >

<P ALIGN="JUSTIFY"> <FONT FACE="Verdana" SIZE="3">

     При наборе текста в какого-нибудь текстовом редакторе (Microsoft Word, WordPerfect др.) для обозначения абзаца мы используем клавишу <Enter>

</FONT> </P>

<P ALIGN="JUSTIFY"> <FONT FACE="Verdana" SIZE="3">

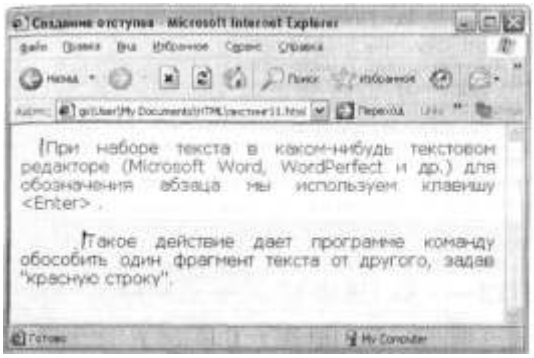
      Такое действие дает программе команду обособить один фрагмент текста от другого, задав "красную строку".

</FONT> </P>

</BODY>

</HTML>

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере. Результатом должно послужить следующее:

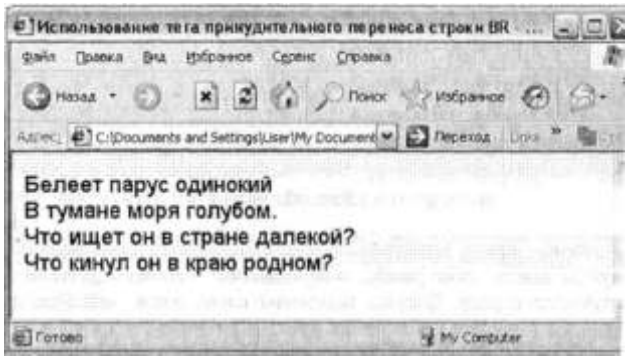


Задание № 12. Использование тега принудительного переноса строки

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

```
<HTML>
<HEAD>
<TITLE> Использование тега принудительного переноса строки <BR></TITLE>
</HEAD>
<BODY >
<P> <FONT FACE="Arial" SIZE="4">
Белеет парус одинокий <BR>
В тумане моря голубом <BR>
Что ищет он в стране далёкой? <BR>
Что кинул он в краю родном?
</FONT> </P>
</BODY>
</HTML>
```

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере. Результатом должно послужить следующее:



Контрольные вопросы.

1. Что представляет собой документ, написанный на языке HTML?
2. Что такое теги?
3. Какие бывают теги?
4. Что такое исходный код?
5. Как обозначают начало и конец тега?
6. Что такое заголовок документа? Какими тегами он выделяется?
7. Что такое тело документа? Какими тегами оно выделяется?
8. Какими буквами следует писать теги, прописными или строчными?
9. Какой тег позволяет перейти на новую строку?
10. Какой тег используют для разбиения текста на параграфы?
11. Какие теги используют для выравнивания текста по левому, правому краю и центру?
12. Как программируется в HTML красная строка?

Отчет о практической работе должен содержать:

1. Тему работы
2. Цель
3. Оснащение
5. Результаты практической части работы.
6. Ответы на контрольные вопросы

Практическая работа № 19 (1 час)

Тема: Закачивание файлов в сеть интернет на Yandex – диск, *Тема:* Скачивание файла по сети, Простые и расширенные поисковые запросы.

Цели: Научиться закачивать информацию на Yandex – диск

Оснащение работы: системный блок, монитор, мышь, принтер, сканер, мультимедиа проектор, ОС Windows, драйверы.

План:

1. Изучение теоретического материала
2. Выполнение заданий для практической работы
3. Подготовка отчета о лабораторной работе (домашнее задание)

Теоретическая часть

Сетевой диск - вещь очень удобная. Не буду углубляться в технологии и терминологию, просто несколько цитат с Яндекса, поскольку описываться будет работа именно с этим сервисом:

- Это бесплатный сервис для хранения ваших файлов и работы с нимина любом устройстве, подключенном к интернету.
- Памятные фотографии и важные документы не пропадут вместе с компьютером. А места для них всегда хватит.
- Компьютер ломается всегда не вовремя. Например, когда на нём единственная версия готового диплома. Сохраняйте важные документы на Диске, и ваша информация никуда не пропадёт.
- Видео не помещается на флешку? На Яндекс. Диске для него хватит места. И, в отличие от флешки, Яндекс.Диск невозможно забыть дома.
- Информация на Диске всегда с вами — в любой точке мира и на любом устройстве, подключенном к интернету.
- Ваши файлы на Яндекс.Диске не привязаны к одному компьютеру или телефону. Вы можете начать работать с документами в офисе, а продолжить дома — с того же места, где остановились.
- Под рукой нет шнура, чтобы скинуть фотографии с телефона? Загрузите фото на Диск с помощью мобильного приложения и откройте на компьютере папку «Яндекс. Диск». Ваши файлы уже там.
- Файлами с вашего Диска легко делиться с друзьями и родными — кому они будут доступны, вы определяете сами.

- Носите с собой флешку, чтобы показать друзьям фотографии из отпуска? Забудьте о ней. Теперь можно положить фотографии в Диск и дать к ним доступ друзьям и родным.
- Экономьте время на отправке файлов — зайдите на Яндекс.Диск и просто поделитесь ссылкой на файл с другом или коллегой. Получив ссылку, они скачают файл сами.

Плюсы и минусы.

Плюсы:

Простота использования

- Возможность использования на компьютерах с любой современной ОС (Windows/Linux/MacOS...)
- Возможность использования на мобильных устройствах (телефоны/смартфоны/планшетники)
- Можно работать как с обычным диском на компьютере (запись/чтение/удаление,установка програм...)
- Можно получить доступ к диску из любой точки мира, с любого компьютера через интернет
- Если ваш компьютер случайно попадет под булбдозер, файлы на сетевом диске останутся в сохранности

Минусы:

Объем диска ограничен условиями сервиса, Яндекс в праве менять эти условия

- Ограничено время хранения (см. условия на яндексе)
- Доступ к диску возможен только при наличии интернет соединения
- Скорость работы с диском зависит от скорости интернет соединения
- То, что находится на диске Яндекса, фактически вам не принадлежит
- Доступ к удаленному диску могут получить злоумышленники, соответственно - не рекомендуется хранить конфиденциальную информацию

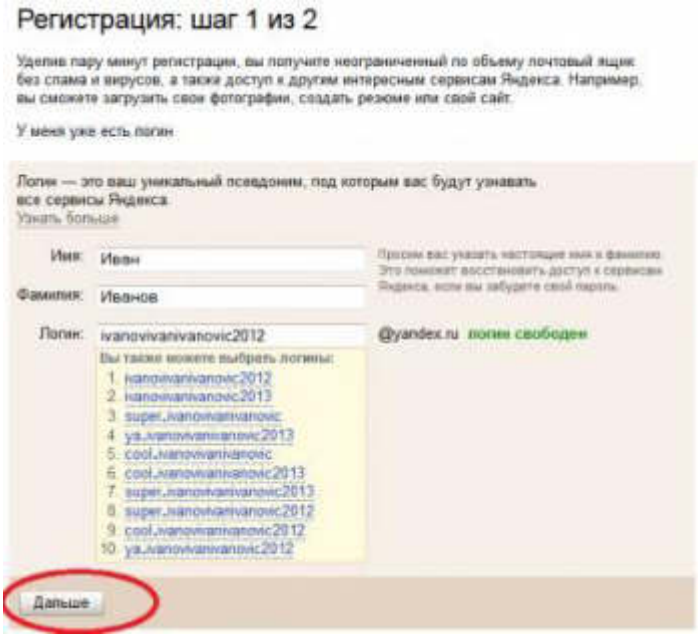
Практическая часть:

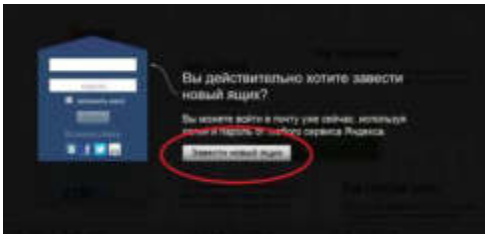
Порядок работы.

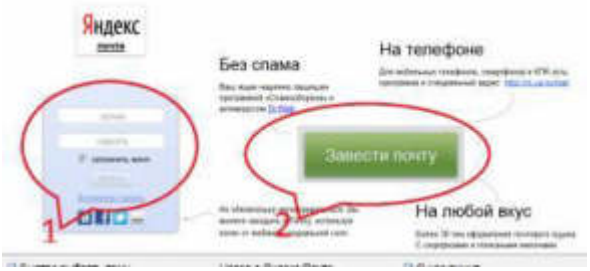
1. Заводим почту на yandex.ru
2. Отправляем заявку на создание яндекс диска
3. Подключаем яндекс диск к компьютеру

1. Заводим почту на yandex.ru

Открываем браузер, в адресную строку вводим http://mail.yandex.ru/ или переходим по этой ссылке. И видим приглашение:

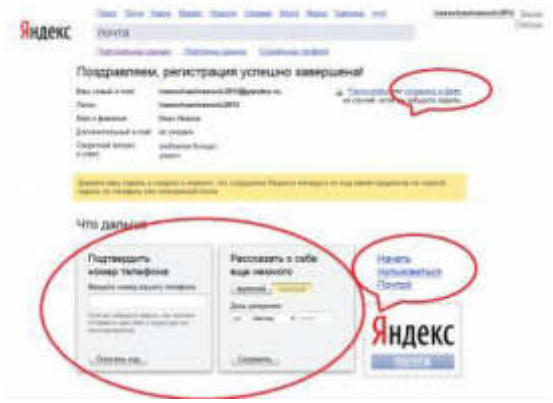




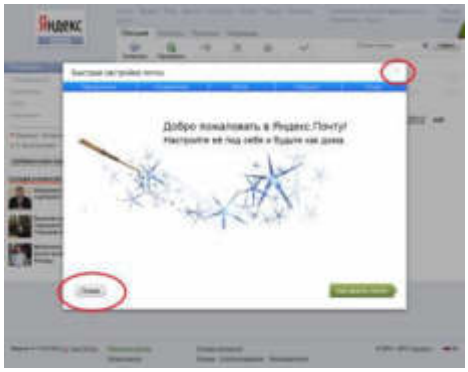


Если у вас уже есть учетная запись почты на яндексе, то вводим логин и пароль в форму входа и входим (1). Если учетной записи у вас еще нет, то нажимаем на кнопку Завести почту (2). В следующем окне еще раз подтверждаем, что хотим завести почту: Затем вводим свои данный (имя, фамилия), придумываем логин. Нажимаем далее: На втором шаге придумываем пароль, секретный вопрос, вводим капчу (символы с картинки), жмем Зарегистрировать:

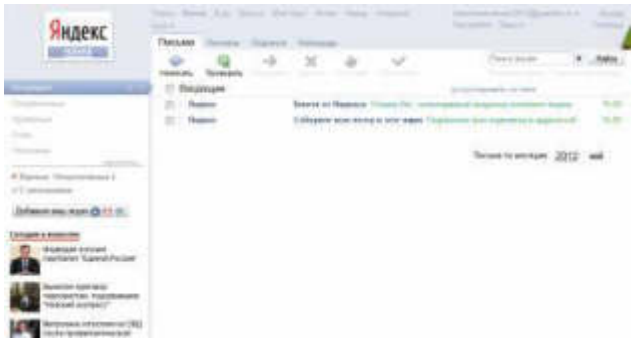
В последнем окне регистрации можем сохранить информацию об учетной записи в файл, указать номер телефона для восстановления доступа, дату рождения, потом переходим по ссылке НАЧАТЬ ПОЛЬЗОВАТЬСЯ ПОЧТОЙ:



Окно настройки почты можно закрыть:



И вот перед нами веб интерфейс нашего почтового ящика:

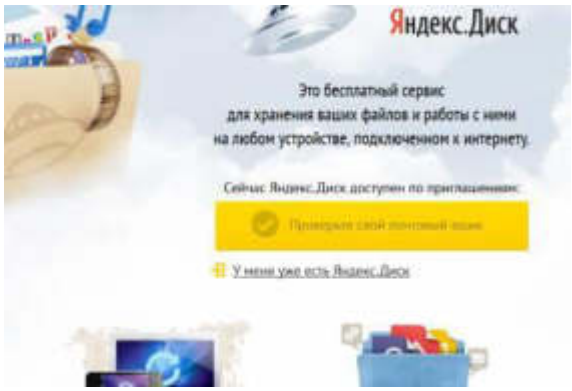


2. Отправляем заявку на создание Яндекс диска.

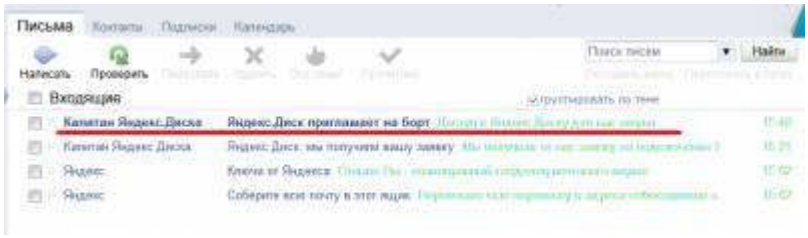
Открываем в браузере новую вкладку и в адресной строке вводим https://disk.yandex.ru/invite/?hash=BHDV1I9C или переходим по этой ссылке. В форму заявки

вводим адрес своего созданного почтового ящика и нажимаем кнопку Оставить заявку:

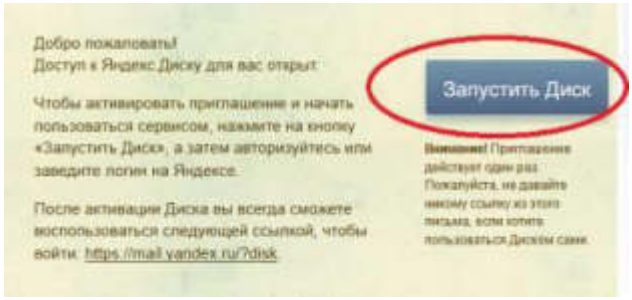
Если заявка принята к рассмотрению, вы увидите всплывающую подсказку, а затем такое окно:



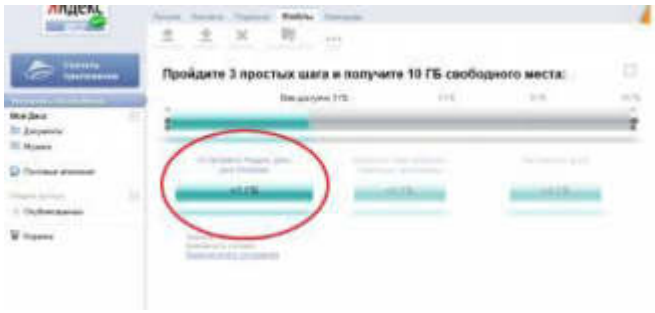
Переходим на вкладку с почтой и ждем несколько минут, нетерпеливым можно периодически нажимать кнопку проверить. И вот пришло долгожданное письмо:



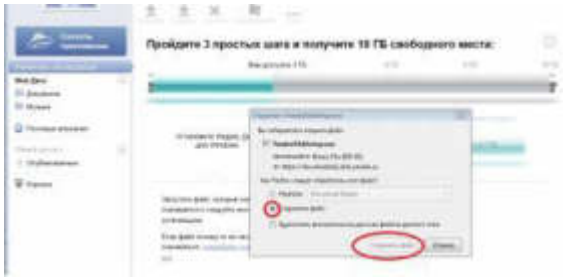
Открываем письмо, нажимаем на кнопку Запустить диск:



По умолчанию предоставляется 3 ГБ дискового пространства, чтобы его увеличить, яндекс предлагает пройти несколько шагов: Во-первых установить специальную программу для работы с Яндекс диском. Программа это бесполезная, но раз просят сделаем:

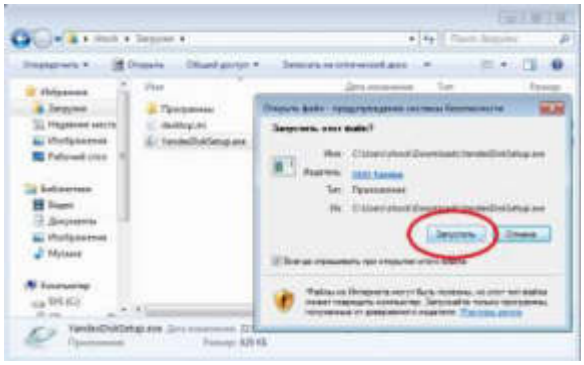


Нажимаем, чтобы сохранить программу:

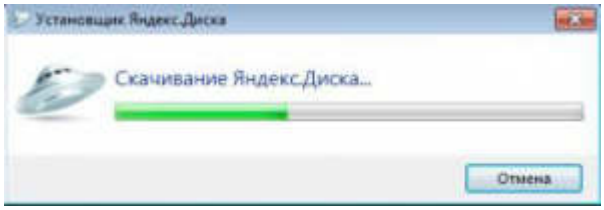


Сохраняем в любое удобное место. У меня это документы---загрузки:

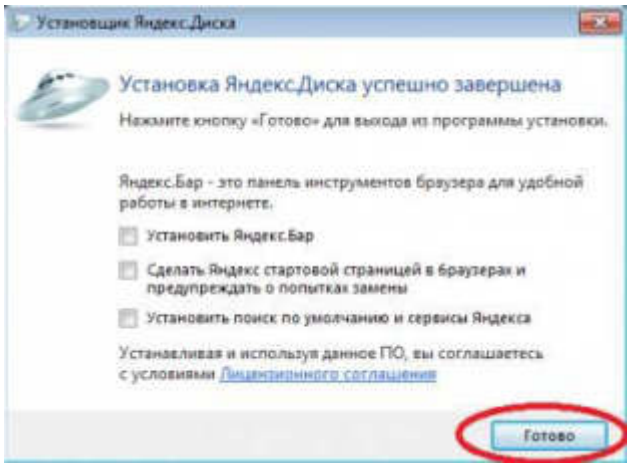
Открываем папку, в которую сохранили программу, и запускаем ее установку:



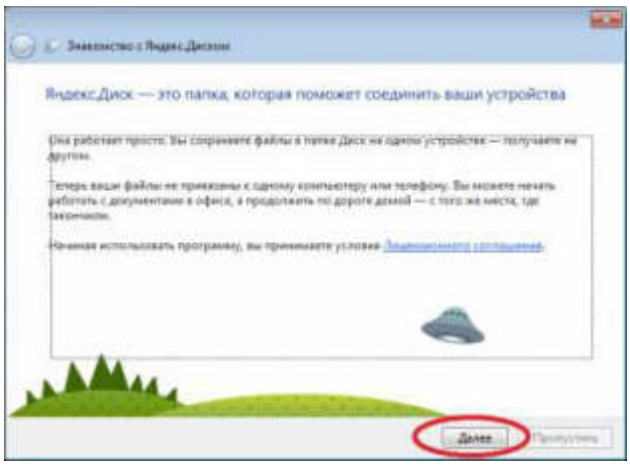
Устанавливаем:

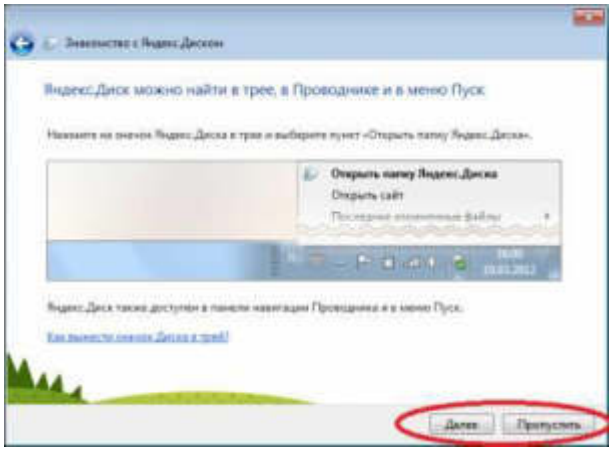


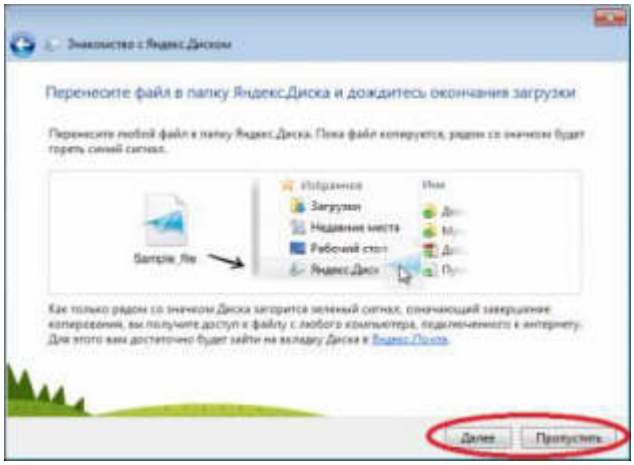
Отмечаем или удаляем нужные/ненужные нам компоненты (я снял все галочки):

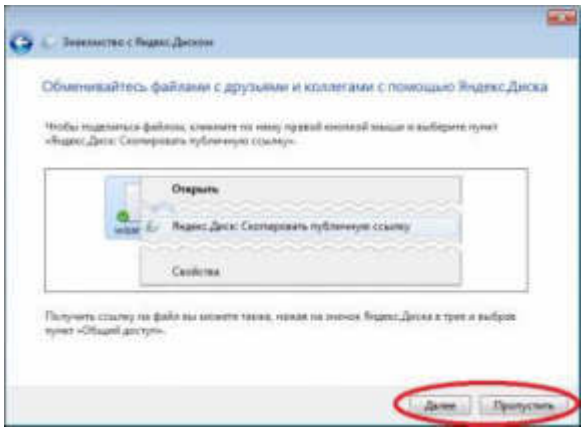


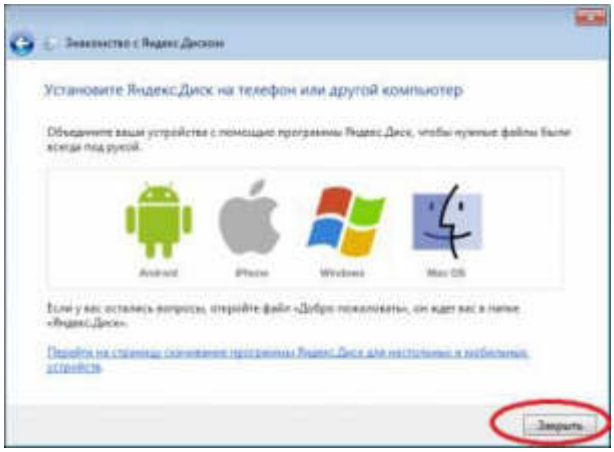
Программа показывает нам несколько полезных советов, которые можно пропустить:



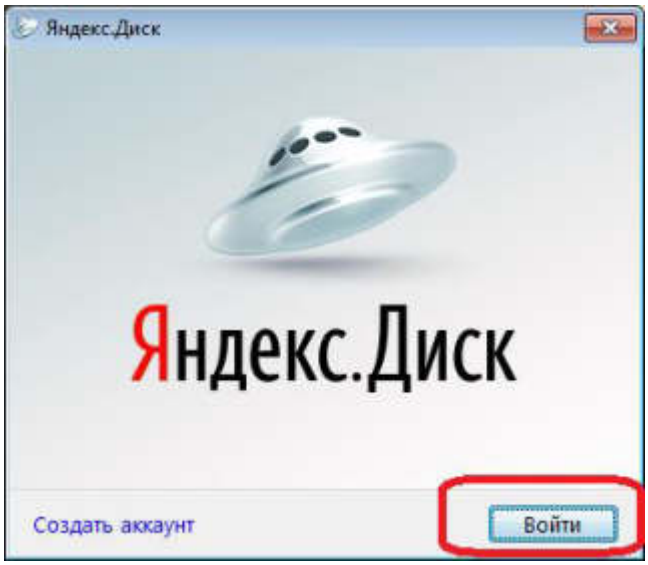




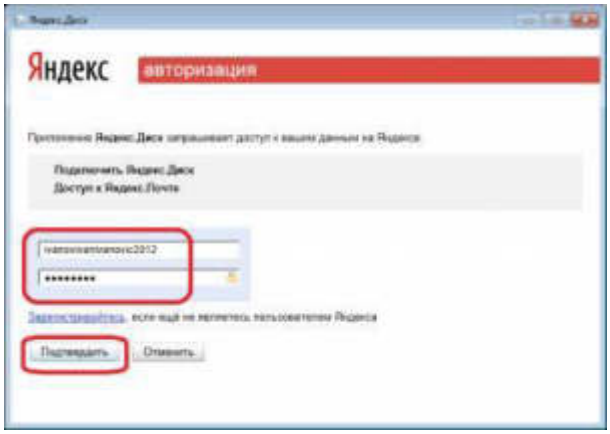




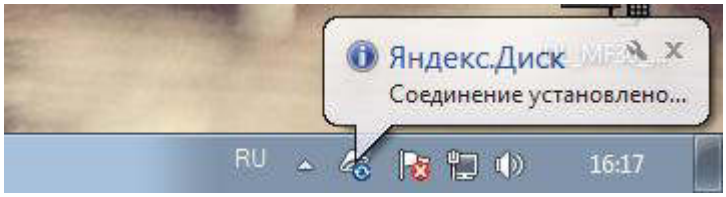
Через несколько секунд после нажатия кнопки Закрыть в установщике, программа запускается:



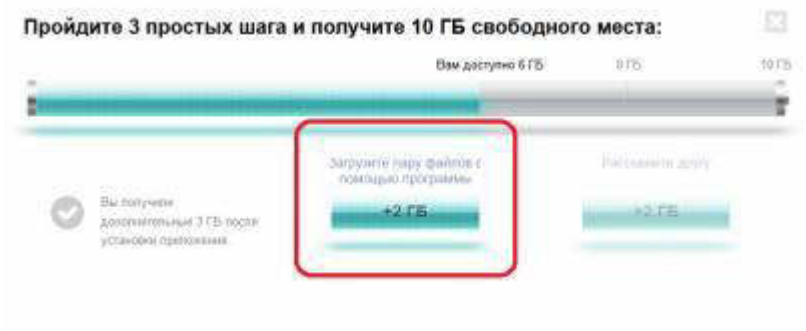
Нажимаем Войти, вводим логин и пароль от почты Яндекс, Подтверждаем:



Теперь программой можно пользоваться:



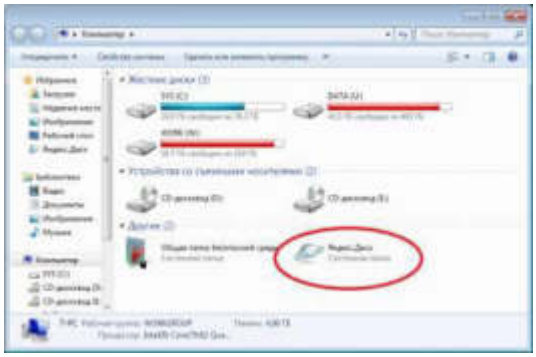
Официальную инструкцию к программе можно посмотреть здесь. Первый шаг на пути увеличения дискового пространства пройден. Посмотрим, что получилось. Набираем в адресной строке постоянный адрес веб интерфейса Яндекс диска: https://mail.yandex.ru/neo2/#disk/landing , и видим:



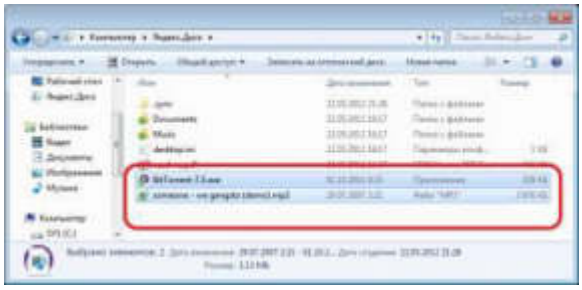
Первый шаг пройден. Доступен второй шаг - Надо загрузить на я.диск пару файлов с помощью

установленной программы. Нажимаем и видим что от нас хотят:

Выполняем - заходим в МОЙ КОМПЬЮТЕР , далее в папку: Яндекс.Диск :



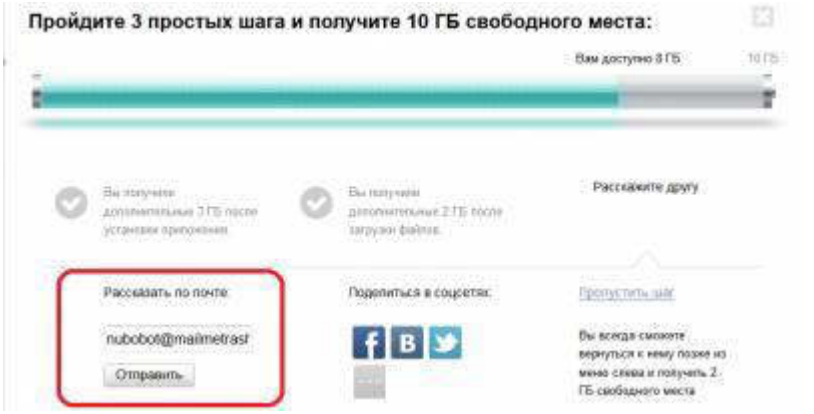
Закидываем в эту папку пару самых маленьких файлов, которые попадуться вам под руку. Чем меньше будет объем файлов, тем быстрее пройдет синхронизация. Когда пройдет синхронизация, файлы будут помечены галочками:



Переходим опять в браузер в веб интерфейса Яндекс диска(https://mail.yandex.ru/neo2/#disk/landing) , и видим:

Два шага пройдены, остался последний - рассказать другу. Нажимаем. И в форме Рассказать по почте вводим любой рабочий адрес электронной почты и

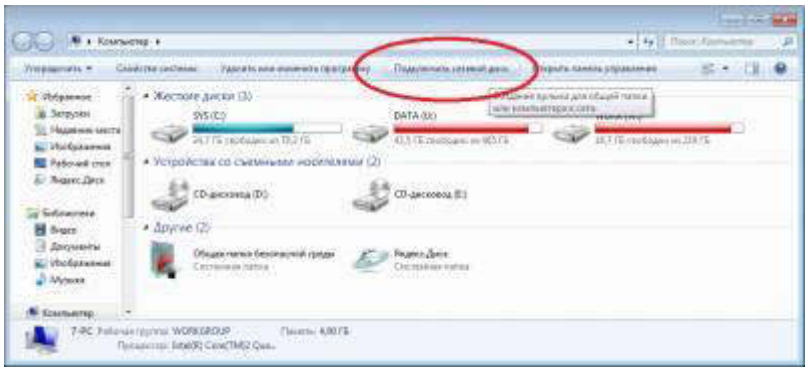
Жмем Отправить:



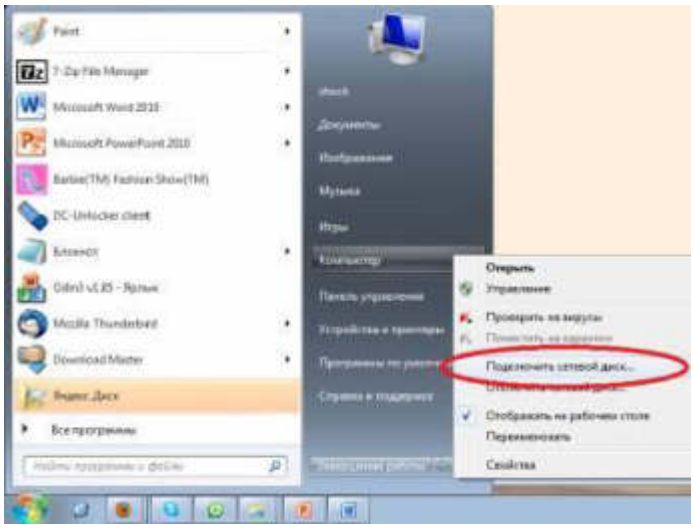
Если вы ввели действительный адрес электронной почты, то через некоторое время увидите, что и третий шаг пройден и у нас есть 10 ГБ дискового пространства:

Подключаем яндекс диск к компьютеру.

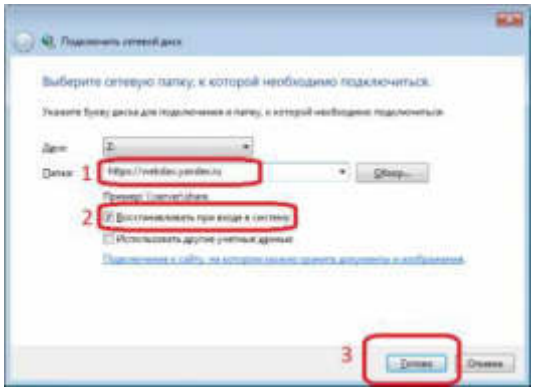
А теперь самое интересное. Подключаем Яндекс.Диск в качестве сетевого диска. Заходим в папку МОЙ КОМПЬЮТЕР и вверху на панели меню нажимаем кнопку Подключить сетевой диск :



То же самое можно сделать другим способом: На панели задач Windows нажимаем кнопку ПУСК, Щелкаем правой кнопкой мыши по ярлыку КОМПЬЮТЕР и в появившемся меню выбираем Подключить сетевой диск:

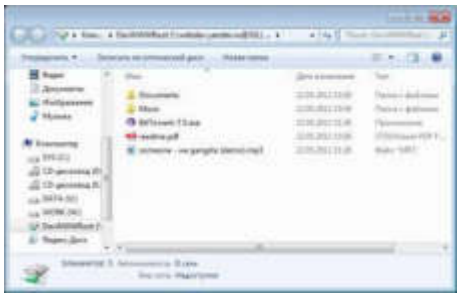


В любом случае результат будет один - откроется такое окно:

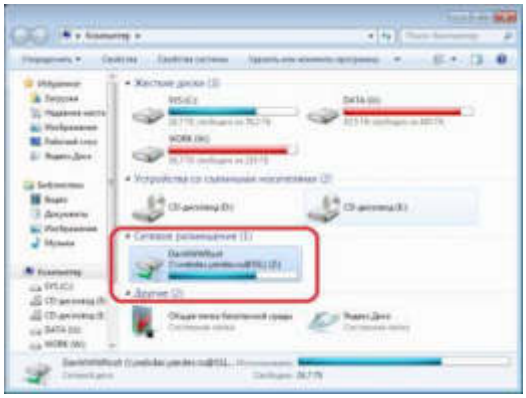


В строке Папка пишем адрес сервера яндекс *https://webdav.yandex.ru* . Отмечаем галочкой пункт Восстанавливать при входе в систему , чтобы не повторять эту процедуру при каждом включении компьютера. И нажимаем кнопку Готово. В следующем окне вводим логин и пароль от яндекс почты и Отмечаем галочкой пункт Запомнить учетные данные. Нажимаем ОК :

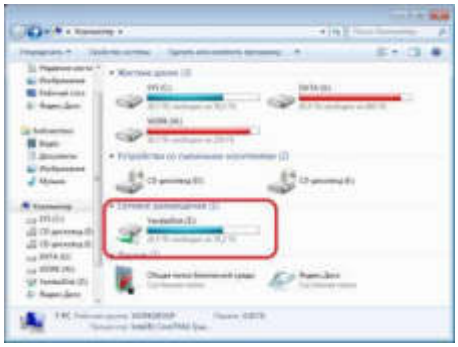
Если все сделано правильно, подключенный диск откроется как любой локальный диск вашего компьютера:



В папке МОЙ КОМПЬЮТЕР он будет выглядеть так :



Если название вашего нового диска вам не нравится, можете его переименовать, например, вот так:



Контрольные вопросы:

4. Перечислите наиболее распространенные виды логических структур веб-сайтов (по материалам п. 2.1).
5. Почему в именах файлов нежелательно использование символов русского алфавита (по материалам п. 2.1)?
6. С какой целью (для использования в какой части сайта) мог быть создан файл *bg_about_me.jpg* (по материалам п. 2.1)?
7. Сформулируйте принцип организации, преимущества, недостатки и область применения решетчатой структуры сайта (по материалам п. 2.1).
8. Укажите достоинства и недостатки пакета Microsoft FrontPage как универсального инструмента для обмена и управления узлами в Web в сравнении с другими средствами редактирования HTML-страниц (см., например, [4]).
9. Параметры (атрибуты) какого элемента определяются заново при изменении в редакторе

цвета фона веб-страницы (см., например, [4])?

10. Какими атрибутами или какими элементами HTML устанавливаются наклон, толщина, подчеркивание и другие особенности шрифта в тексте и его фрагментах (см., например, [4])?

Отчет о практической работе должен содержать:

1. Тему работы
2. Цель
3. Оснащение
5. Результаты практической части работы.
6. Ответы на контрольные вопросы

Тема: Скачивание файла по сети

Цель: Изучить назначение и принцип работы FTP-протокола.

Научиться использовать ftp-клиенты для получения файлов из сети.

Научиться устанавливать и настраивать FTP-сервер (на примере IIS или ProFTPd). Уметь разрабатывать простые приложения, работающие по протоколу FTP.

Оснащение работы: системный блок, монитор, мышь, принтер, сканер, мультимедиа проектор, ОС Windows, драйверы.

План:

1. Изучение теоретического материала
2. Выполнение заданий для практической работы
3. Подготовка отчета о лабораторной работе (домашнее задание)

Теоретическая часть:

ЧТО ТАКОЕ FTP

File Transfer Protocol («протокол передачи файлов») или просто FTP — сетевой протокол, предназначенный для передачи файлов в компьютерных сетях. Протокол FTP позволяет подключаться к серверам FTP, просматривать содержимое каталогов и загружать файлы с сервера или на сервер, кроме того, возможен режим передачи файлов между серверами. Поддержка загрузки файлов на сервер – это основное преимущество FTP перед самым популярным протоколом в Internet – HTTP (его будем изучать на следующей лабораторной работе).

FTP является одним из старейших прикладных протоколов, появившимся задолго до HTTP в 1971 году. До начала 90-х годов на долю FTP приходилось около половины трафика в сети интернет. Данный протокол и сегодня используется для распространения программного обеспечения и доступа к удалённым хостам. Наверняка, каждый из Вас скачивал файлы в

локальных сетях или из сети Internet, набирая в строке браузера или менеджера закачек URL-строку вида «ftp://…». Так вот – вы использовали протокол ftp для получения интересующих Вас данных.

Протокол является текстовым и открытым – передаваемые данные не шифруются. Например, при аутентификации передаёт логин и пароль открытым текстом. Если злоумышленник находится в одном сегменте сети с пользователем FTP, то, используя сниффер (например, Wireshark), он потенциально может «поймать» логин и пароль пользователя, или вообще, при наличии специальной программы, перехватывать передаваемые по FTP файлы без авторизации. Для предотвращения перехвата открытого трафика используют механизмы шифрования трафика, в частности, SSL – Security Sockets Lavel.

Механизм работы с FTP является по своей сути клиент-серверным. В работе по протоколу FTP принимают участие 2 компонента – FTP-сервер и FTP-клиент. Под FTP-протокол зарезервированы два стандартных TCP -порта: 20 и 21.

Первоначальная спецификация протокола описана в RFC 959. В настоящее время разработаны версии FTP для работы с IPv6 (RFC 2428 – FTP Extensions for IPv6 and NATs).

FTP-КЛИЕНТЫ

Для работы с FTP-серверам (см. ниже) необходимо воспользоваться некоторым FTP-клиентом. На данный момент таких клиентом существует очень большое количество: любой современный WEB-браузер, специализированные клиенты (утилита ftp, CuteFTP и т.д), практически все файловые менеджеры (FAR, Total Commander и даже «встроенный» в Windows Explorer и др.).

Общий формат строки подключения, воспринимаемый практически всеми (консольная утилита ftp, о которой пойдет речь далее, является исключением) выглядит следующим образом:

ftp://[user[:password]]@host[:port][/dir],

где *user* – логин пользователя, *password* – пароль пользователя, *host* – доменное имя сервера или его IP-адрес, *port* – номер сетевого порта (в случае, если сервер работает на нестандартном порту), *dir* – каталог, который должен стать текущим после подключения к FTP-серверу. Логин пользователя и его пароль записываются в открытом виде. Например:

- ftp://admin:super_passwd@iipo.tu-bryansk.ru
- ftp://admin@iipo.tu-bryansk.ru:21
- ftp://192.168.128.5:21/pub
- ftp://82.179.88.34

В тоже время можно указывать логин и пароль в отдельных полях (соответствующие настройки или диалоговые окна) при подключении к серверу.

Как видно из упоминания учетной записи пользователя и его логина, FTP-сервера позволяют работать пользователям, предварительно зарегистрированным на данном сервере. Однако в тоже время многие FTP-сервера (определяется не сервером, а его настройками) поддерживают т.н. анонимный режим работы, позволяя любому пользователю подключиться к серверу, не имея на нем учетной записи. Для этого существуют ставшие уже «стандартными» учетные записи:

логин: *anonymous* или *ftp*

пароль: *anonymous* или *ftp* (многие сервера вообще не учитывают в таком случае пароль).

Т.е. на самом деле, если Вы подключаетесь к FTP-серверу без указания параметров Вашей учетной записи, клиент автоматически пытается подключиться под анонимным пользователем.

Рассмотрим с вами работу с кафедральным FTP-сервером на примере получения файла утилиты *netcat* (nc.exe) стандартной утилитой *ftp* (см. рис. 1)

Многие FTP клиенты поз ьные параметры при подключении. На рис. 2 приведено окно подключения программы FAR.

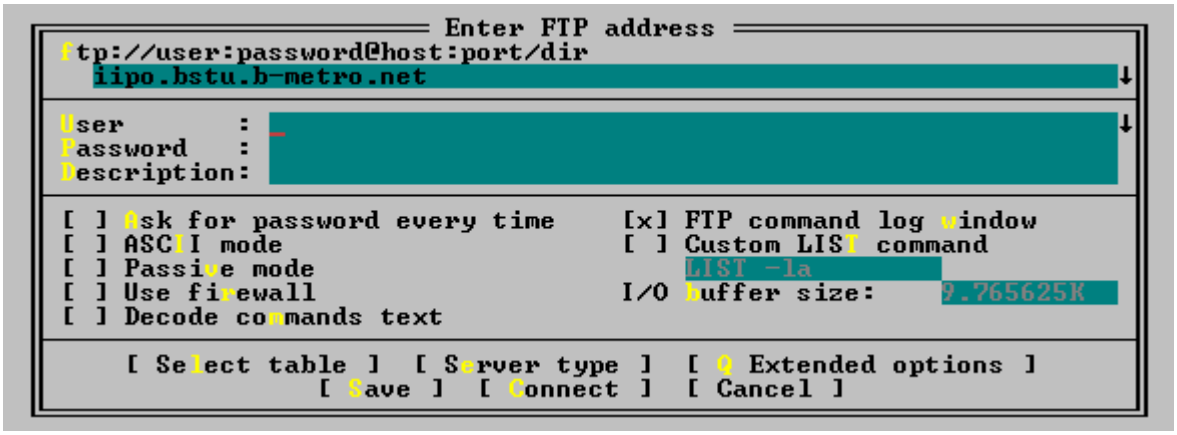


Рис. 2. **Настройки подключения к FTP-серверу в FAR**

FTP–СЕРВЕР

На данный момент существует большое количество реализаций FTP– серверов, как коммерческих, так и свободно распространяемых. В качестве примера рассмотрим настройку FTP-сервера на примере «встроенного» в Windows 2000+ сервера IIS.

Во-первых, данный компонент должен быть установлен из инсталляционного пакета через панель управления и апплет «Добавление и удаление программ». На предлагаемых образах Windows 2000 для Virtual PC сервис уже установлен.

Для получения доступа к настройкам сервера необходимо запустить приложение mmc.exe (например из командной строки). И добавить оснастку «Internet Information Services (IIS) Manager» (см. рис. 3).

После чего мы получаем доступ к настройкам серверов: FTP, HTTP, SMTP (см. рис. 4). Нас будет интересовать только FTP-сервис.

Настройка сервера для работы достаточно проста. Основные шаги настройки приведены на рис. 5-8. Основная цель настройки - организовать возможность скачивания некоторых заданных файлов с настроенного сервера в рамках кафедральной лаборатории.

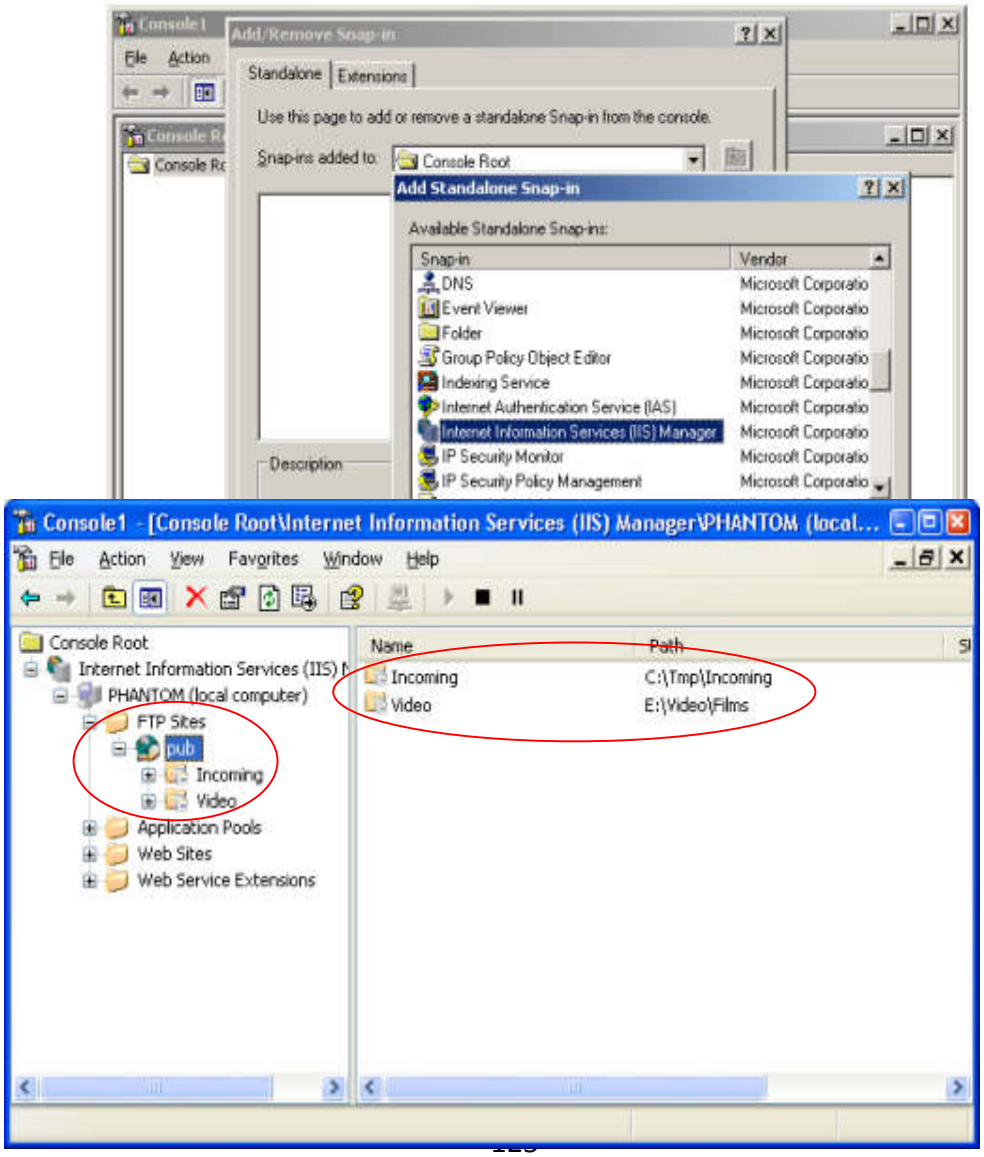
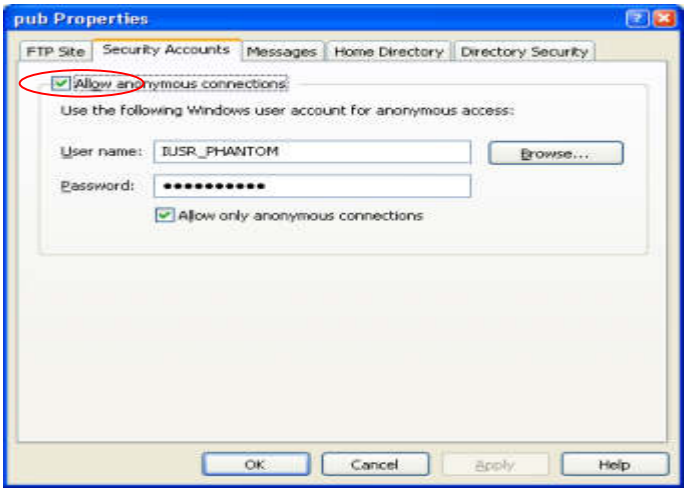


Рис. 5. **Добавление ftp-сайта и виртуальных каталогов**



ПРИНЦИП РАБОТЫ ПРОТОКОЛА FTP

Протокол FTP является протоколом прикладного уровня. Реализуется на базе транспортного протокола TCP. Протокол FTP интересен и необычен тем, что во время работы реализует одновременно два соединения (*дуплексных,* или *двунаправленных*) между клиентом и FTP-сервером, которые несут на себе различную нагрузку. Первое соединение называется управляющим (*control connection*). Оно используется для передачи команд от клиента к серверу (аутентификация клиента, перемещения в дереве каталогов FTP-сервера и т.п.) и получения кодов возврата от сервера клиентом. Второе соединение – это канал данных (*data connection*), предназначенный, собственно, для передачи самих «полезных» данных: списка файлов и содержимого передаваемых файлов.

Канал данных создается каждый раз при получении и ли передачи данных с файловых систем. То, каким образом будет устанавливаться соединение канала данных, определяется режим работы FTP: *активный* или *пассивный*.

В случае активного соединения клиент информирует сервер (посылкой команды «PORT») о том, что он создал прослушивающий сокет на заданном порту (непривилегированном) и ждет подключение на него сервера. После чего сервер подключается к заданному номеру порта клиента, используя со своей стороны порт номер 20 (рис. 9).

Когда клиент желает работать в пассивном режиме он уведомляет об этом сервер (посылкой команды «PASV»), после чего сам сервер создает прослушивающий сокет (также на непривилегированном порту) и уведомляет о его параметрах клиента (откликом «227 Entering Passive Mode…» с параметрами сокета). После чего клиент подключается на указанный сокет (рис. 10).

После установления канала данных сами данные могут передаваться как сервером, так и клиентом в зависимости от контекста выполняемых команд на управляющем соединении.

Все современные реализации FTP-сервером поддерживают как активный, так и пассивный режимы работы.

Активное соединение используется по умолчанию и применимо, если сервер может подключиться напрямую к клиенту. Однако в современных условиях при наличии повсеместно всевозможных прокси серверов это не представляется возможным. Поэтому в большинстве случаев применимым оказывается пассивный режим работы.

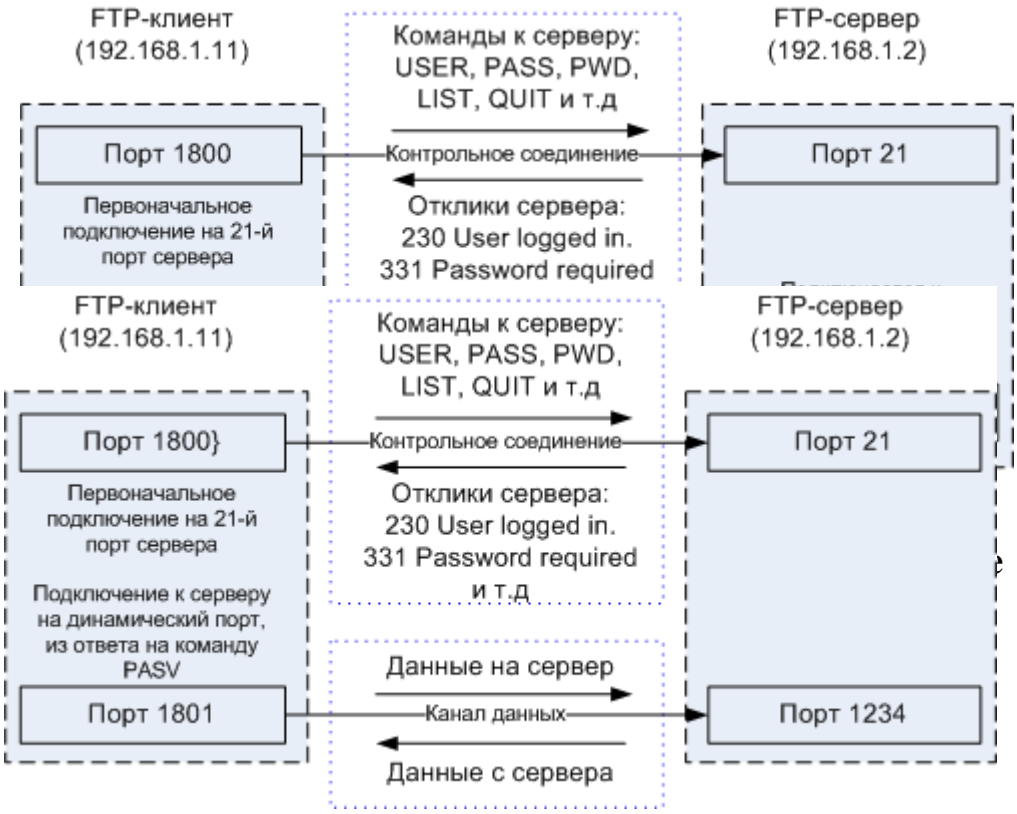


Рис. 10 **Схема работы протокола ftp в пассивном режиме**

ПРИМЕР РАБОТЫ ПО FTP-ПРОТОКОЛУ

Мы уже кратко затронули принцип работы протокола FTP. Сейчас же мы познакомимся с самим протоколом более подробно. В частности, рассмотрим работу протокола на примере получения списка файлов с сервера и последующее получение содержимого некоторого текстового файла.

В данном примере будем использовать кафедральный FTP-сервер (iipo.bstu.b-metro.net он же iipo.tu-bryansk.ru или совсем кратко – linux). С нашей (клиентской) стороны мы будем использовать программу netcat (ftp://iipo.tu-bryansk.ru/pub/Drozdov/Tools/netcat/), которая позволяет не только организовывать TCP-подключение, но также позволяет создавать

прослушивающей TCP-сервер. Это нам понадобится, т.к. в данном примере работу будем вести в активном ftp-режиме. IP-адрес клиентской машины 192.168.128.234.

Работа с ftp начинается становления контрольного соединения (рис. 11)

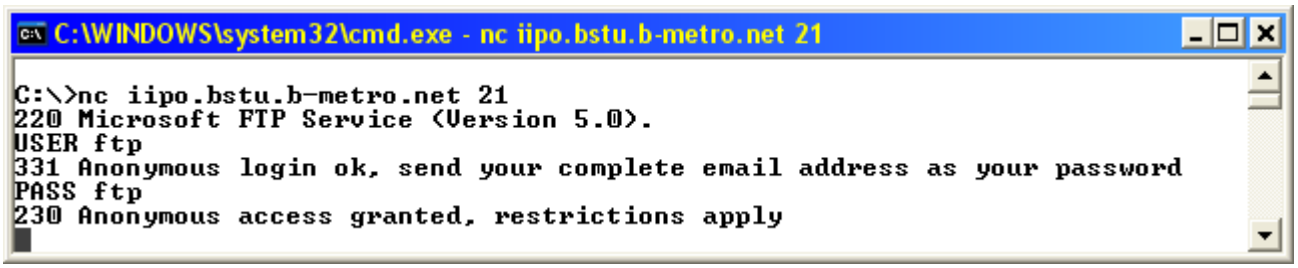


Рис. 11 **Авторизация на FTP-сервере**

Мы установили соединение с севером и авторизовались под анонимным пользователем. В данном случае мы отправили две команды (*USER* и *PASS*) с параметрами (логин и пароль «ftp»). От сервера были получены отклики, начинающиеся (*всегда*) с целого числа.

Теперь нам необходимо получить список файлов. Его можно получить только по каналу данных. Для этого организуем параллельно еще одну сессию работы программы netcat и запустим ее в режиме прослушивания некоторого порта (можно выбрать произвольно, главное чтобы не было коллизий с уже открытыми портами). Возьмем для примера порт 1800 (см. рис. 12).

Рис. 12. **Открытие TCP-порта с номером 1800 на прослушивание**

Теперь уведомим сервер об открытом для подключения порту командой *PORT*. Синтаксис ее простой:

PORT A1, A2, A3, A4, P1, P2

Где A1-A4 разделенные *через запятую* компоненты IP-адреса, P1 и P2 – 2 компоненты порта прослушивания N:

P1 = N / 256 – целочисленное деление;

P2 = N % 256 – остаток от целочисленного деления.

Например, в нашей ситуации команда уведомления сервера будет следующей:

PORT 192,168,128,234,7,8

Теперь пошлем эту команду на сервер и запросим список файлов в текущей директории FTP-командой *NLST*. В результате чего сервер подключается к программе netcat, запущенной в режиме TCP-сервера, шлет в текстовом виде список файлов и закрывает соединение. Далее мы снова создаем канал данных и получим содержимое текстового файла командой *RETR*. Синтаксис ее прост:

RETR «относительны путь к файлу».

Для примера попробуем получить содержимое текстового файла с описанием протокола FTP – rfc959.txt, который имеется на сервере.

Для корректного завершения FTP-сессии клиент должен послать серверу команду *QUIT*.

На рис. 13 представлен итоговый лог работы с контрольным соединением, а на рис. 14 – данные, полученные от сервера по соединениям ftp-данных клиентом – список директорий в корне ftp-сервера и содержимое файла rfc959.txt.

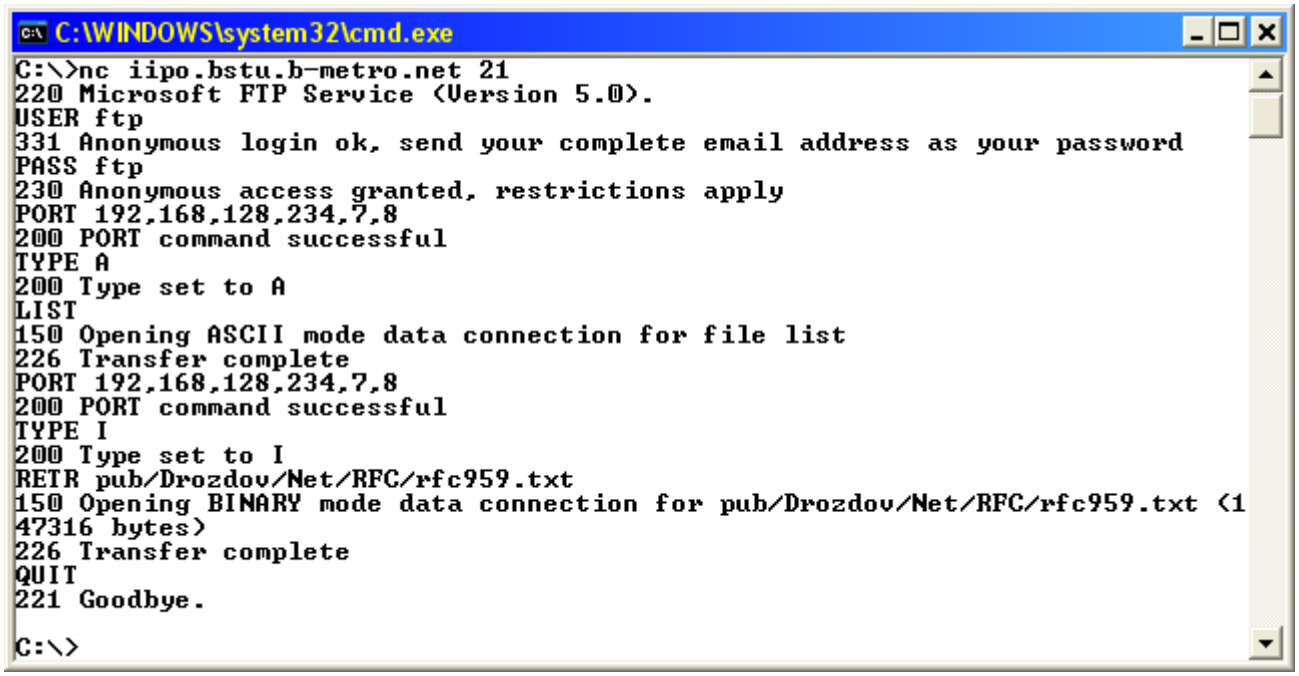


Рис. 13. **Итоговый лог работы с ftp-сервером на контрольном соедине-**

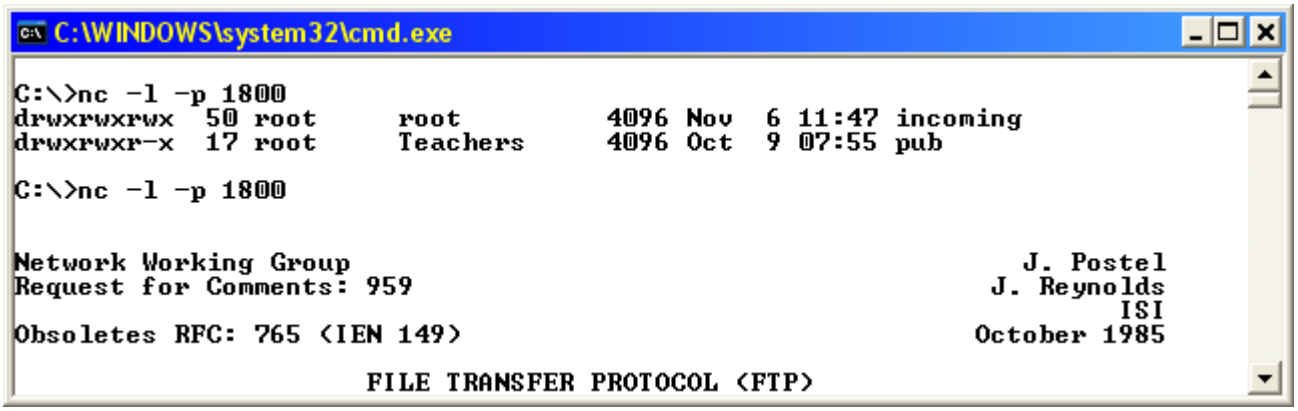


Рис. 14. **Данные, полученные с FTP-сервера**

В примере приведена типичная ситуация получения данных с ftp-сервера клиентскими программами, за исключением, что они при скачивании файлов сохраняют получаемый поток данных на диск.

Все подробности по работе протокола можно прочитать в соответствующем RFC [1]. Также списки базовых команд протокола FTP и кодов откликов с их кратким описанием приводятся в приложении.

Практическая часть:

1. Настроить ftp сервер на базе IIS и протестировать работу с ним, используя любой ftp-клиент.
2. Используя сниффер Wireshark, изучить работу некоторого FTP-клиента с сервером. Рекомендуется использовать кафедральный ftp-сервер. Сопоставить команды и коды откликов с таблицами из приложения.
3. Выполнить пример получения данных с сервера с использованием утилиты netcat в *активном режиме*.
4. Организовать сессию работы с FTP-сервером, используя утилиту netcat и получить содержимое некоторого каталога и произвольного файла в *пассивном режиме*.
5. Написать консольную программу – простейший ftp-клиент для работы с FTP-сервером в пассивном режиме. Программа должна уметь загружать с сервера файл, переданный ей в качестве параметра командной строки и сохранять его на локальном диске.
6. Представить отчет в печатном или электронном виде.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. 1. Что такое FTP?
2. 2. Где применяет протокол FTP?
3. 3. Какие два режимы работы FTP-сервера и клиента существуют? В чем их принципиальное отличие?
4. 4. Какие данные можно передавать по протоколу FTP?
5. Отчет о практической работе должен содержать:
6. 1. Тему работы
7. 2. Цель
8. 3. Оснащение
9. 5. Результаты практической части работы.

6. Ответы на контрольные вопросы

Тема: Простые и расширенные поисковые запросы

Цель :рассмотреть наиболее известные поисковые системы.Овладеть основными навыками простого и расширенного поиска информации. Научиться работать в англоязычной поисковой системе

Оснащение работы: системный блок, монитор, мышь, принтер, сканер, мультимедиа проектор, ОС Windows, драйверы.

План:

1. Изучение теоретического материала
2. Выполнение заданий для практической работы

3. Подготовка отчета о лабораторной работе (домашнее задание)

Теоретическая часть

Виды поисковых систем

Для поиска информации в Интернете нужно использовать поисковые системы, сделанные в виде WWW-серверов. Необходимо соединиться с одним из таких серверов и далее действовать в зависимости от способа поиска, реализованного на данном сервере. Помимо универсальных поисковых систем существуют и специальные, предназначенные для нахождения таких видов информации, как данные о конкретных людях или компаниях(адресные и телефонные справочники), или какой-либо специализированной информации(по финансам, генетике и пр.). Перечень поисковых систем, а также их характеристики представлены в табл. 1.3

Зарубежные и отечественные поисковые системы Зарубежные поисковые службы

www.yahoo.com Yahoo Одна из старейших и наиболее популярных поисковых служб. Если поиск по собственному каталогу не дает результата, привлекаются ресурсы другого поискового указателя. Информация представлена в виде древовидной структуры, в которой на самом верхнем уровне представлены наиболее обобщенные разделы. При выборе одного из разделов происходит переход на другой уровень с более подробным описанием

www.altavista.digital.com AltaVista Одна из крупнейших (но не самая крупная) поисковых систем мира. Обладает огромным охватом Web-пространства и мощным набором поисковых команд. Для начинающих система имеет упрощенный интерфейсwww.lycos.com Lycos Первоначально служба была запущена как поисковая система, основанная на программе-роботе, собирающей информацию из WWW. Однако совсем недавно система перешла на другую модель, основанную на каталогизации Web-узлов

www.infoseek.com InfoSeek Содержит поисковую систему по людям в Интернете

www..google.com Google При генерации списка ссылок в первую очередь выдает ссылки на те страницы, к которым из других документов ведет наибольшее количество ссылок. При проведении поиска по тривиальным словам дает хорошие результаты, потому что руководствуется мнением других пользователей, ранее осуществлявших подобный поиск

www.dejanews.com DejaNews Поиск информации в телеконференциях

www.aport.ru Aport Одна из первых поисковых машин российского Интернета. Система сочетает «машинный» и «человеческий» факторы. Когда по результатам поиска выводится адрес страницы, он сопровождается кратким описанием Web-узла. Система уделяет особое внимание рейтингованию результатов поиска

www.rambler.ru Rambler Рейтинговая система, обладающая всеми основными функциями поисковых указателейwww.yandex.ru Yandex Обладает механизмом морфологического разбора слов, что особенно важно для русского языка. Предоставляет инструменты для расширенного поиска Организация поиска информации

Выбор поискового сервера. Поиск информации начинается с выбора поискового сервера.Для этого необходимо в адресной строке ввести URL сервера. Программа Internet Explorer установит связь с выбранным сервером, и через некоторое время на экране появится его Web-страница. Выбор способа поиска. Каждый из поисковых инструментов предлагает два способа поиска: простой (simple search) и расширенный (advanced search), с использованием специальной формы запроса и без нее.

Правила оформления запроса

1. В запрос информации можно включать специальный символ «*», расширяющий диапазон поиска. Символ «*» позволяет заменить любой другой символ или набор символов до конца слова.

2. Роль прописных букв. Прописные буквы поисковыми системами воспринимаются однозначно, т.е. ищутся тексты, в которых слова записаны точно такими же буквами. Строчные буквы поисковыми системами воспринимаются и как строчные, и как прописные.

3. Если в запросе использованы кавычки, поисковая система разыскивает документы, в которых абсолютно точно есть тот текст, который в нем заключен.

Простой и расширенный поиск информации в Интернете

Каждый из поисковых инструментов предлагает простой и расширенный варианты поиска.

Простой запрос дает значительное количество ссылок на документы, так как в список попадают документы, содержащие одно из слов или простое словосочетание, введенное при запросе,

Правила оформления простого запроса:

1. Сложение наоборот. Если нам надо, чтобы поисковая система нашла страницы, на которых одновременно присутствуют все использованные ключевые слова, то перед каждым из них надо поставить знак «+». Используя знак «+», мы сужаем круг поиска и уменьшаем количество возможных ссылок. Например, необходимо найти информацию о протоколах Интернета. Для этого нужно в строке поиска указать следующее:

+Интернет+протокол

Поисковая система выдаст список страниц, на которых встречаются оба эти слова, хотя, конечно, не исключено, что между ними нет прямой связи.

Система Апорт выполняет такой поиск по умолчанию.

2. Вычитание. Конкретизировать круг поиска информации можно не только знаком «+» но и знаком «-». Например, необходимо найти информацию о протоколах Интернета, но без учета тех страниц, на которых протокол рассматривается как нормативный документ. Для этого нужно в строке поиска указать следующее:

+интернет+протокол~документ

3. Применение джокера. В запрос информации можно включать специальный символ «*», расширяющий диапазон поиска. Символ «*» позволяет заменить любой другой символ или набор символов до конца слова. Например, необходимо найти информацию о протоколах Интернета. Для этого в строке поиска запишем:

+интернет*+протокол*

Поисковая машина осуществит поиск всех документов*в которых встречаются словосочетания, состоящие из слов «Интернет» и «протокол» в различных падежах.

4. Роль прописных букв. Прописные буквы поисковыми системами воспринимаются однозначным образом, т.е. разыскиваются тексты, в которых слова записаны точно так же прописными буквами. Строчные буквы поисковыми системами воспринимаются и как строчные, и как прописные.

Например, поиск по словам:

+глобальная+сеть

вернет результаты, в которых будут встречаться и «глобальная сеть», и «Глобальная Сеть». С другой стороны, поиск по словам: Глобальная Сеть даст гораздо более узкий результат: «Глобальная Сеть».

5. Контекстный поиск. В случае использования кавычек поисковая система разыскивает документы, в которых абсолютно точно есть тот текст, который заключен в кавычки. Поиск с помощью кавычек называется контекстным поиском.

Например:

«глобальная сеть»

Поисковая система найдет все документы, в которых есть подобный текст.

6. Поиск по заголовкам. Каждая страница содержит заголовок. При его отсутствии как заголовок выделяется предложение из первого абзаца. В результате поиска выдается ссылка, в которой присутствует заголовок. Например:

■ Title: глобальная сеть

■ Заг.: глобальная сеть

Поисковая система Апорт имеет несколько альтернативных команд:

■ t=глобальная сеть

■ з=глобальная сеть

Средства расширенного поиска позволяют более точно формулировать поисковое задание.

В большинстве поисковых систем команды расширенного поиска формируются с помощью логических команд.

Преимущество использования логических команд связано с тем, что команды простого поиска у многих поисковых систем реализованы по-разному. Каждая поисковая система стре-

мится сделать средства простого поиска наиболее удобными, а средства расширенного — наиболее стандартными.

Правила оформления расширенного запроса

1. Команда OR. OR (или) служит для формирования поискового задания, состоящего из нескольких ключевых слов, если надо, чтобы разыскиваемый документ содержал любые из этих слов в любой комбинации. Поисковая служба Апорт позволяет использовать как английское написание команды, так и русское. Например: протокол OR TCP/IP.

2. Команда AND. AND (и) служит для формирования поискового задания, когда надо, чтобы разыскиваемый документ содержал одновременно все слова, введенные пользователем. Аналогичные функции выполняет команда «+» простого поиска. Апорт позволяет использовать как английское ключевое слово AND, так и русское слово И, а также знаки «&» и «и».

Например: протокол AND TCP/IP.

3. Команда NOT. NOT (не) устанавливает исключения из результатов поиска. Наряду с командой NOT в системе Апорт разрешается использовать команду «НЕ». Например: протокол NOT TCP/IP.

4. Команда NEAR. NEAR (рядом) — одна из самых удобных команд расширенного поиска. Она позволяет совместить достоинства поиска по ключевым словам и контекстного поиска. При глубоком поиске возможно получение наилучших результатов. В поисковой системе Апорт расстояние между словами задается командой СЛ#(...,...,...), где вместо знака «#» следует поставить число, обозначающее предельное расстояние между указанными словами, а вместо многоточий — список ключевых слов.

Например: СЛ5(протокол, TCP/IP).

5. Вложение команд. Вложение команд позволяет создавать весьма сложные запросы. Оно выполняется с помощью круглых скобок ( ). Команда, стоящая в скобках, выполняется в первую очередь. Например: сеть AND (глобальная OR интернет)

Практическая часть:

Задания

Отчет по выполнению заданий оформить в файле Wordu сохранить в своей папке.

1. Найти и обработать информацию по запросу «Путешествия». (Указать поисковик и выписать количество ссылок и скопировать первые три в отчет)

2. Найти сайты по разделу «Авто», используя тематический каталог на поисковых

системах Yandex или Rambler. (Указать поисковик и выписать количество ссылок и скопировать первые три в отчет)

3. Произвести простой поиск информации с применением различных методов

(сложение наоборот, вычитание, применение джокера, контекстного поиска и др.).

4. Произвести расширенный поиск информации по запросу «Компьютерные системы в

юриспруденции». (Описать различия по сравнению с простым запросом)

5. Найти и оценить сайты по производству промышленной продукции (велосипеды). (Дать аннотацию первых трех)

6. Используя полностью в запросе ключевую фразу «Применение интернет-

телефонии», произвести поиск информации. (Указать поисковик и выписать количество ссылок и скопировать первые три в отчет)

7. Найти и обработать информацию по запросу «Интеллектуальные поисковые

системы». (Скопировать понятие и краткое описание – отформатировать)

8. Произвести поиск информации по предыдущим запросам (на английском языке) на

зарубежных поисковых системах Yahoo, AltaVista и др.

(Воспльзуйтесь программами преводчиками)

9. В адресной строке введите http://www.altavista.telia.com/

10. Введите ключевое слово «финансы» и выберите зону поиска в окне Any language,

например England (Англию).

11. В полученном списке гиперссылок выберите Official site of Investment Banking

Group NIKoil. Откроется сайт на английском языке, но со встроенным переводчиком.

После нажатия на кнопку Русский информация автоматически переведется на русский

язык. (Выписать количество ссылок и скопировать первые три в отчет)

12. Вернитесь с помощью кнопки Назад, расположенной на панели инструментов, на

главную страницу сервера AltaVista и выберите раздел, который вас наиболее

интересует (семья, политика...).

13. В полученном списке гиперссылок выберите, на ваш взгляд, наиболее интересную.

Откроется страница на английском языке. Для того чтобы прочесть полученную информацию на русском языке, воспользуемся переводчиком (Скопировать скрин-шот страницы в отчет)

Контрольные вопросы:
Что такое средство просмотра?
2. Что такое поисковые инструменты?
3. Охарактеризуйте задачи поисковых инструментов.
4. Из чего состоит пользовательский интерфейс поискового инструмента?
5. Что такое запрос?
6. Что такое URL-адрес?
7. Какие существуют виды поисковых систем?
8. Перечислите международные поисковые системы.
9. Перечислите отечественные поисковые системы.
10. Как начать поиск информации?
11. Какие существуют способы поиска информации?
12. Что такое простой и расширенный поиск информации?
13. Простой .поиск информации. Сложение наоборот.
14. Простой поиск информации. Вычитание.
15. Простой поиск информации. Применение джокера.
16. Простой поиск информации. Роль прописных букв.
17. Простой поиск информации. Контекстный поиск.
18. Простой поиск информации. Поиск по заголовкам.
19. Расширенный поиск. Команда OR.
20. Расширенный поиск. Команда AND.
21. Расширенный поиск. Команда NOT.
22. Расширенный поиск. Команда NEAR.
23. Расширенный поиск. Вложение команд.
24. Какие существуют способы перевода сайтов в Интернете?
25. Как можно улучшить качество перевода?
26. Можно ли перевести текст на русском языке на другой язык? Если можно, то как это сделать?
27. На каком этапе наиболее целесообразно использование программы-переводчика

Отчет о практической работе должен содержать:

1. Тему работы

2. Цель

3. Оснащение

5. Результаты практической части работы.

6. Ответы на контрольные вопросы

Практическая работа№ 20 (1 час)
Тема: Форматирование (визуальное - внесение необходимой и удаление лишней информации) и настройка отображения веб-страниц

Цель:

- ✓ научиться форматировать Web-страницы

План:

- ✓ Теоретические сведения
- ✓ Ход работы
- ✓ Контрольные вопросы

Теоретические сведения

<B> … </B> – жирный шрифт;

<I> … </I> – курсив;

<U>… </U> – подчеркнутый текст;

<STRIKE>… </STRIKE>,

<S> … </S> – перечеркнутый шрифт;

<BIG> … </BIG> – шрифт большего размера;

<SMALL> … </SMALL> – шрифт меньшего размера;

_… – подстрочный индекс;

[…] – надстрочный индекс;

<FONT> … </FONT> – изменение типа, цвета и размера шрифта.

В теге <FONT> допускаются атрибуты:

SIZE – размер букв, возможные значения: 1, 2, 3, …, 7. По умолчанию 3;

COLOR – цвет шрифта. Можно указывать шестнадцатеричными кодами или английскими словами. По умолчанию BLACK;

FACE – изменение типа шрифта, возможные значения: ARIAL, COURIER и т. д. По умолчанию VERDANA.

Теги могут комбинироваться, но при этом надо следить, чтобы контейнеры имели вложенную структуру (контейнер, открытый позже, должен быть закрыт до того, как будет закрыт внешний). Например, для комбинации жирного и наклонного шрифтов можно использовать такой код:

<B> <I> пример </I> </B>

В окне браузера мы получим: *пример*

Помимо перечисленных выше тегов существуют шесть контейнерных тегов, удобных

для оформления заголовков: <H1>...</H1>, <H2>...</H2>, …, <H6>…</H6>. Заголовки отличаются от текста и между собой толщиной и высотой букв (в заголовке <H1> самые крупные символы, в <H6> – самые мелкие), а также увеличением интервала по вертикали между заголовком и прочей информацией. Атрибут ALIGN управляет выравниванием заголовков по горизонтали. По умолчанию он принимает значение LEFT. Другие возможные значения: CENTER, RIGHT.

Пример:

<H4 ALIGN=CENTER> 6. Теги выделения смысловых частей текста </H4>

Теги выделения смысловых частей текста

Данный атрибут применим также к графике и таблицам.

Таблица 1 – Значения атрибута *ALIGN*

Значение	Описание применения
Left	Выравнивание по левому краю
Right	Выравнивание по правому краю
Justify	Выравнивание по левому и правому краям
Center	Центрирование

Абзацы в тексте выделяются контейнером <P>…</P>. Атрибут ALIGN задает выравнивание по горизонтали. По умолчанию он принимает значение LEFT. Другие возможные значения: CENTER, RIGHT. При просмотре браузером абзацы отделяются друг от друга пустой строкой. Закрывающий тег </P> необязателен. Его можно использовать только тогда, когда было заказано нестандартное выравнивание текста.

Примеры:

<P> Этот абзац расположен слева </P>

<P ALIGN=CENTER> Этот абзац по центру </P>

<P ALIGN=RIGHT> А вот этот абзац справа </P>

Приведенный выше код при просмотре выглядит так:

Этот абзац расположен слева

Этот абзац по центру

А вот этот абзац справа

Чтобы перейти на новую строку, не вставляя пустую строку в текст, применяется тег <BR>. Закрывающего тега у него нет.

Иногда полезно разделять смысловые части документа горизонтальной чертой. Это де-

лается с помощью одиночного тега <HR>.

Его атрибуты:

SIZE – толщина линии в пикселах. По умолчанию 1 пиксел.

WIDTH – длина линии в пикселах или процентах от ширины экрана. По умолчанию 100%.

ALIGN – выравнивание линии, возможные значения: LEFT, CENTER, RIGHT. По умолчанию LEFT.

COLOR – цвет линии. По умолчанию GRAY.

NOSHADE – линия рисуется без тени. По умолчанию тень есть.

Пример. Код, в котором использованы теги
 и <HR>:

<B> Жизненная позиция </B>

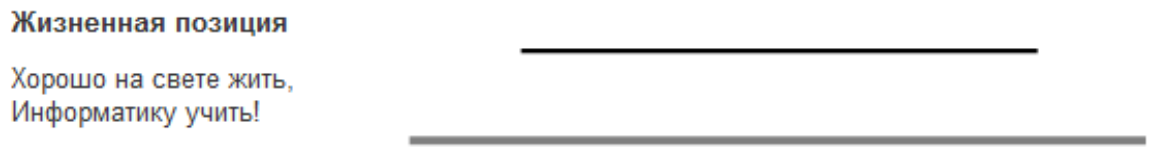
<HR NOSHADE WIDTH=250>

*Хорошо на свете жить,
*

*Информатику учить!
*

<HR SIZE=4 WIDTH=50% ALIGN=CENTER>

В окне браузера этот код изображается так:



Оформление списков

Списки облегчают нахождение нужной информации при перечислении сведений, объединенных общим смыслом: пункты меню, списки товаров, перечень сотрудников и т. п. В HTML предусмотрено несколько вариантов оформления списков, которые чаще всего используются в документации. Здесь будут рассмотрены только маркированные и нумерованные списки.

Для того чтобы оформить фрагмент документа в виде маркированного списка, его помещают в контейнер <UL>… </UL> (Unordered List). Для нумерованного – в контейнер <OL> … </OL> (Ordered List). Затем текст в контейнере разбивается на отдельные пункты списка тегами <LI> (List Item – элемент списка). Закрывающий тег </LI> в конце каждого пункта ставить необязательно. В маркированном списке все пункты автоматически помечаются одинаковыми маркерами, в нумерованном – элементы нумерует сам браузер.

Примеры кода:

<UL>Список дисциплин:	*<OL>Список дисциплин:*
<LI>Информатика	*<LI>Информатика*
<LI>Физика	*<LI>Физика*
<LI>История	*<LI>История*
</UL>	*</OL>*

Это дает на экране следующие списки:

Список дисциплин:	Список дисциплин:
• Информатика • Физика • История	1. Информатика 2. Физика 3. История

Можно организовать многоуровневые списки (разумно ограничиться тремя уровнями вложения). Для этого после каждого <LI> следует разместить контейнеры <UL>… </UL> или <OL> … </OL> с разбивкой информации этого пункта на дополнительные подпункты.

В теге <OL> могут использоваться следующие атрибуты:

TYPE – определяет стиль оформления меток для элементов списка, подробнее ниже;

START – (только для нумерованного списка <OL>) определяет значение первого элемента списка, задается всегда числом (по умолчанию 1). В зависимости от значения атрибута TYPE этому числу сопоставляются разные символы, подробнее ниже;

VALUE – (только для метки <LI> списка <OL>) устанавливает номер текущего элемента списка. Как и в предыдущем случае, метка пункта может быть нечисловой, хотя значение атрибута – всегда число.

Для нумерованного списка в зависимости от стиля нумерации значением атрибута TYPE могут быть:

1 – арабские цифры (1,2,3,...)

a – строчные буквы (a, b, c,…)

A – прописные буквы (A,B,C,...)

i – римские цифры строчные (i,ii,iii,...)

I – римские цифры прописные (I,II,III,...)

По умолчанию значения TYPE устанавливаются в зависимости от степени вложенности списка.

Число, указанное в атрибуте START, показывает порядковый номер элемента в ряду знаков, заданных в TYPE. С этого элемента следует начинать нумерацию пунктов списка.

Ход работы

Задание

Создайте главную страницу фирмы по продаже бытовой электроники

Текст на Web-странице следующий:

Компания *(заголовок центрировать, используя теги H1…, FONT…)*

Открытое акционерное общество Компания основанная в 2006 году,

является одним из ведущих продавцов бытовой электроники в России *(с новой строки, выравнивание по левому краю).*

Основными направлениями деятельности Компании являются: *(выравнивание по центру, новый абзац)*

реализация бытовой электроники ведущих фирм мира через сеть магазинов;

создание сервисных центров по обслуживанию бытовой электроники (*оформлено с помощью списка*);

(вставить картинку)

Компания ищет агентов по сбыту бытовой электроники *(с новой строки, выравнивание по левому краю).*

Контактная информация: *(с новой строки, выравнивание по правому краю*)

Telephone (123) 123-34-56

FAX (123) 123-34-56

Почтовый адрес 123456 г. Город, ул Лесная, 106

Электронная почта

Общая информация: abc@abc.su *(создать гиперссылку на abc@abc.abc.su)*

Продажи: abc@abc.abc.su

Copyright © 2006 Компания

Дополнительное задание

Составить свое резюме по следующему плану (в общей сложности 4-5 строчек, вместо многоточий и пояснений в скобках вставить свои данные):

Я, … (ФИО), родился … (дата и год рождения) в городе … (место, где родились).

В 20… году окончил … школу (лицей, гимназию) с углубленным изучением …, средний балл аттестата …

В 20… году поступил на специальность …. Средний балл по первой сессии (аттестациям) …

В свободное от учебы время увлекаюсь … (перечислить не менее трех увлечений помимо учебы).

Требования к оформлению резюме:

1 После раздела «ФИО, где и когда родился», поставить тег абзаца <P>;

2 После раздела «какую школу кончил, средний балл аттестата» – тег разрыва строки <BR>;

3 Перед разделом «увлечения помимо учебы» вставить в код тег горизонтальной линии <HR>. Оформите линию нестандартным способом. Для этого следует внести в тег <HR> атрибуты длины, толщины, цвета и выравнивания линии. Задайте длину 20 % – 50 % от ширины окна, выравнивание по центру, толщину 4 – 10 пикселей, цвет – любой.

4 Измените вид шрифтов в разных частях резюме. Для этого следует контейнером <H1> выделить свою фамилию, контейнером <I> – город, где Вы родились, контейнером <H6> – информацию о школе, контейнером <FONT> с атрибутами SIZE=7 COLOR=RED – информацию о факультете, контейнером <FONT> с атрибутами SIZE=1 COLOR=GREEN – информацию о приятелях, контейнером <B> выделить какое-нибудь увлечение.

5 Оформите часть текста в виде списка. Для этого следует контейнером <OL> выделить блок об увлечениях, разбить его на пункты тегами <LI>. Аналогично контейнером <UL> и тегами <LI> оформить в виде маркированного списка перечень Ваших приятелей.

6 Измените нумерацию в списке. Для этого с помощью атрибутов TYPE и START поменяйте нумерацию в списке увлечений.

7

Контрольные вопросы

1 Что такое форматирование?

2 Какие тэги используются при форматировании текста?

3 Основные атрибуты тэгов форматирования?

4 Что такое URL в HTML?

5 Как создается гиперссылка в HTML?

КОМПЛЕКТ ПРАКТИЧЕСКИХ РАБОТ
ПО ДИСЦИПЛИНЕ
ПМ.03 «Обработка информации в базах данных»

(наименование учебной дисциплины)

г. Балашиха

ВВЕДЕНИЕ

Настоящий сборник практических работ предназначен в качестве методического пособия при проведении практических работ по программе профессионального модуля «Обработка информации в базах данных» при подготовке квалифицированных рабочих и специалистов среднего звена.

Требования к знаниям и умениям при выполнении практических работ

В результате выполнения практических работ, предусмотренных программой по данному профессиональному модулю, проводится текущий контроль индивидуальных образовательных достижений.

Результаты обучения

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся в ходе освоения профессионального модуля должен:

уметь:

- применять современную методологию для исследования и синтеза информационных моделей предметных областей АСОИУ;
- иметь навык выполнения работ на пред проектные стадии;
- применять современную методологию на стадии технического проектирования - обследование, выбор и системное обоснование проектных решений по структуре информационных моделей и базам данных, по архитектуре банка данных и его компонентам;
- применять методы проектирования баз данных и составления программ взаимодействия с базой данных;

- применять методы организации работы в коллективах разработчиков баз данных;
- иметь представление: о тенденциях и перспективах развития современных систем управления базами данных.

знать:

- принципы организации и архитектуры банков данных;
- модели баз данных;
- современные методы и средства разработки и синтеза структур информационных моделей предметных областей автоматизированных систем обработки информации и управления;
- последовательность и содержание этапов проектирования баз данных;
- современные методики синтеза и оптимизации структур баз данных;
- основные конструкции языков манипулирования данными;
- методики оптимизации процессов обработки запросов;
- современные методы обеспечения целостности данных;
- методы организации баз данных на носителях информации;

ПРАКТИЧЕСКАЯ РАБОТА №1

Тема: «Типы взаимосвязей в модели баз данных»

Цель: Познакомиться на практике с типами взаимосвязей в модели баз данных. Научиться выбирать наиболее эффективную модель для конкретного случая. Иметь понимание преимуществ той или иной модели баз данных.

Время выполнения: 2 урока

Ход работы:

Иерархическая модель (древовидная)

Представляют собой упорядоченную совокупность экземпляров данных типа «дерево», содержащих экземпляры типа «запись». Корневым называется тип, который имеет подчиненные типы, а сам подтипом не является. Подчиненный тип (подтип) является потомком по отношению к типу, который для него является предком.

Плюсы:

- Эффективное использование памяти ЭВМ;
- Малое время выполнения операций над данными.

Минусы:

- Громоздкость;
- Сложные логические связи;
- Сложное понимание.

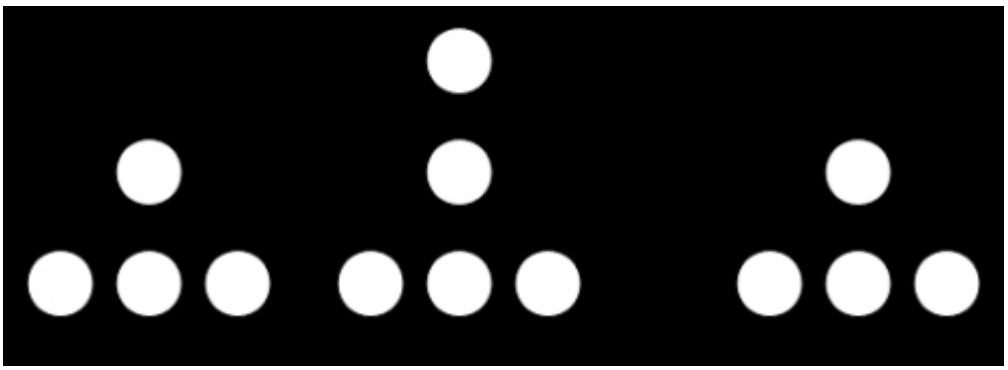


Рисунок 1 – Иерархическая модель

Сетевая модель

Позволяет отображать разнообразные взаимосвязи элементов данных в виде произвольного графа.

Для описания сетевой БД используются две группы типов:

- Запись;
- Связь;

Тип связи определяется для двух типов «запись»- предка и потомка. Запись потомок может иметь произвольное число записей предков.

Плюсы:

- Эффективная реализация по показателям затрат памяти и оперативности;
- Большие возможности для образования произвольных связей.

Минусы:

- Высокая сложность и жесткость БД
- Сложность понимания и выполнения обработки информации в БД
- Ослаблен контроль целостности связей.

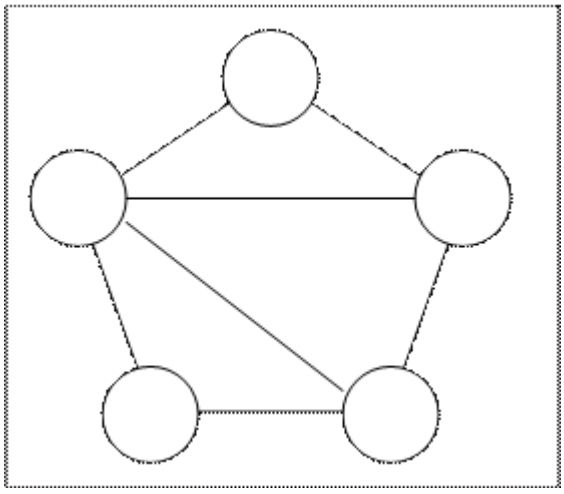


Рисунок 2 – Сетевая модель

Иерархическая модель – таблица

Таблица имеет строки (записи) и столбцы (колонки). Любая строка имеет одинаковую структуру и состоит из полей.

Плюсы:

- Простота, понятность, удобство физической реализации на ЭВМ.

Минусы:

- Отсутствие стандартных средств идентификации отдельных записей и сложность описания иерархических и сетевых связей.

Постреляционная модель

Расширенная реляционная модель, снимающая ограничение неделимости данных, хранящихся в записях таблиц. Постреляционная модель допускает многозначительные поля – поля значения, которых состоят из подзначений. Набор значений многозначных полей считается самостоятельной таблицей, встроенной в основную.

Преимущества:

- Возможность предоставления связанных реляционных таблиц одной постреляционной.
- Высокая наглядность информации и повышенная эффективность ее обработки.

Недостатки:

- Сложное решение проблемы обеспечения целостности и непротиворечивости данных.

Многомерная модель

Узкоспециализируемые СУБД, предназначенные для интерактивной аналитической обработки информации.

Основные понятия, используемые в многомерной модели

Агрегируемость – рассмотрение (данных) информации на различных уровнях ее обобщения.

Историчность данных предполагает обеспечение высокого уровня статичности данных и их взаимосвязей, а также обязательность привязки данных по времени.

Прогнозируемость подразумевает задание функции прогнозирования и применение их к различным временным интервалам.

Преимущества:

- Удобство и эффективность аналитической обработки больших объемов информации, связанных со временем.

Недостатки:

- Громоздкость для построения задач обычной оперативной обработки информации.

Объектно-ориентированная модель

В объектно-ориентированной модели при представлении данных имеется возможность идентифицировать отдельные записи БД. Между записями БД и формами их обработки устанавливаются взаимосвязи с помощью механизмов, подобных соответствующим средствам в объектно-ориентированных языках программирования. Строка объектно-ориентированной БД графически представима в виде дерева, узлами которого являются объекты. Свойства объектов описывается некоторыми стандартным типом (П: string) или типом (используемым) конструируемым пользователем (class). Каждый объект – экземпляр класса считается потомком объекта, в котором он определен как свойство. Объект – экземпляр класса принадлежит своему классу и имеет одного родителя. Родовые отношения в БД образуют связную иерархию объектов. Логическая структура объектно-ориентированной БД внешне похожа на структуру иерархической БД. Основное отличие между ними – в методах манипулирования данными.

Преимущества:

- Возможность отображения информации о сложных взаимосвязях объектов;

- объектно-ориентированная модель БД позволяет идентифицировать отдельную запись БД и определить функции их обработки.

Недостатки:

- высокая понятийная сложность;
- неудобство обработки;
- низкая скорость выполнения запросов.

Задание: Используя средства векторной графики MS Word построить следующие модели данных и дать им краткое описание и характеристику:

- Иерархическая модель (древовидная)
- Сетевая модель
- Постреляционная модель
- Объектно-ориентированная модель

За основу взять тематику «Библиотека» и следующие экземпляры данных:

- книга;
- автор;
- издательство;
- читатель;
- жанр;
- библиотека.

Контрольные вопросы:

1. Какие модели баз данных вы знаете?
2. Перечислите преимущества и приведите пример применения конкретной модели данных.
3. Какие типы взаимосвязей формируются в приведенных вами моделях баз данных?

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №2

Тема: «Обеспечение непротиворечивости и целостности баз данных»

Цель: Научиться на практике соблюдать требования непротиворечивости и целостности баз данных при их проектировании и разработке.

Время выполнения: 2 урока

Ход работы:

На практике часто используются связи, устанавливающие различные виды соответствия между объектами «связанных» типов, — это один к одному (1:1), один ко многим (1: М), многие ко многим (М: М).

Связь один к одному означает, что каждому экземпляру первого объекта (А) соответствует только один экземпляр второго объекта (В) и, наоборот, каждому экземпляру второго объекта (В) соответствует только один экземпляр первого объекта (А).

Связь один ко многим означает, что каждому экземпляру одного объекта (А) может соответствовать несколько экземпляров другого объекта (В), а каждому экземпляру второго объекта (В) может соответствовать только один экземпляр первого объекта (А).

Связь многие ко многим означает, что каждому экземпляру одного объекта (А) могут соответствовать несколько экземпляров второго объекта (В) и, наоборот, каждому экземпляру второго объекта (В) могут соответствовать тоже несколько экземпляров первого объекта (А).

Пример 1.1. Рассмотрим совокупность следующих информационных объектов:

СТУДЕНТ (Номер студента, ФИО, Дата рождения, Номер группы);

СТИПЕНДИЯ (Номер студента, Размер стипендии);

ГРУППА (Номер группы, Специальность);

ПРЕПОДАВАТЕЛЬ (Код преподавателя, ФИО, Должность).

Здесь информационные объекты СТУДЕНТ и СТИПЕНДИЯ связаны отношением один к одному, так как каждый студент может иметь только одну стипендию и каждая стипендия может быть назначена только одному студенту.

Информационные объекты ГРУППА и СТУДЕНТ связаны отношением один ко многим, так как одна группа может включать в себя много студентов, в то время как каждый студент может обучаться только в одной группе.

Информационные объекты СТУДЕНТ и ПРЕПОДАВАТЕЛЬ связаны отношением многие ко многим, так как один студент может обучаться у многих преподавателей и один преподаватель может обучать многих студентов.

Для пользователей АИС важно, чтобы база данных отображала предметную область однозначно и непротиворечиво, т.е. чтобы она удовлетворяла условию целостности.

Выделяют два основных типа ограничений по условию целостности данных в базе.

1. Каждая строка таблицы должна отличаться от остальных ее строк значением хотя бы одного столбца.

Пример: Сотрудники одного отдела могут оказаться полными тезками, иметь одинаковые должность и телефон.

2. Внешний ключ не может быть указателем на несуществующую строку той таблицы, на которую он ссылается. Это ограничение называется ограничением целостности данных в базе по ссылкам.

Пример: В столбце Название отдела таблицы «СОТРУДНИК» хранятся сведения о принадлежности сотрудников к отделу. Здесь Название отдела является внешним ключом для ссылки на таблицу «ОТДЕЛ». Для обеспечения ограни-чения целостности по ссылкам каждое Название отдела из таблицы «СОТРУДНИК» должно принадлежать конкретному отделу из таблицы «ОТДЕЛ».

Задание: Перечислите и поясните все условия, которые должны соблюдаться, для следующих схем данных, для обеспечения целостности данных в базе данных:

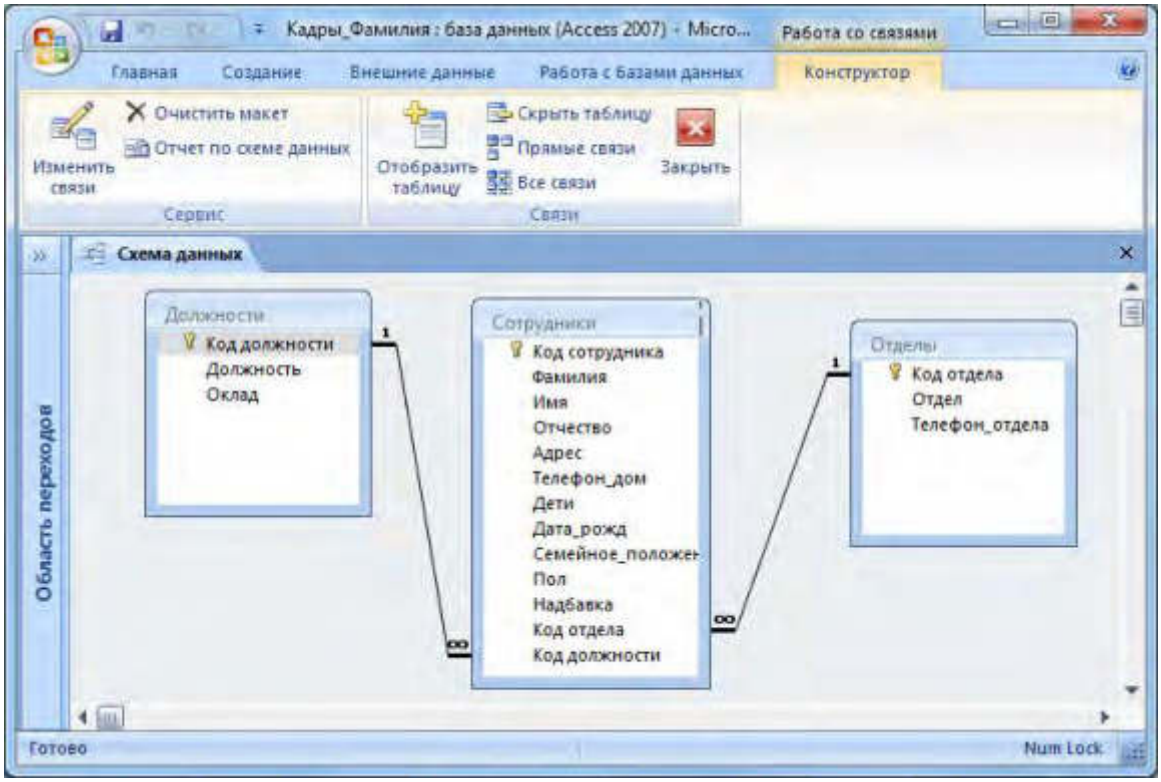


Рисунок 1 – Схема данных базы данных «Кадры»

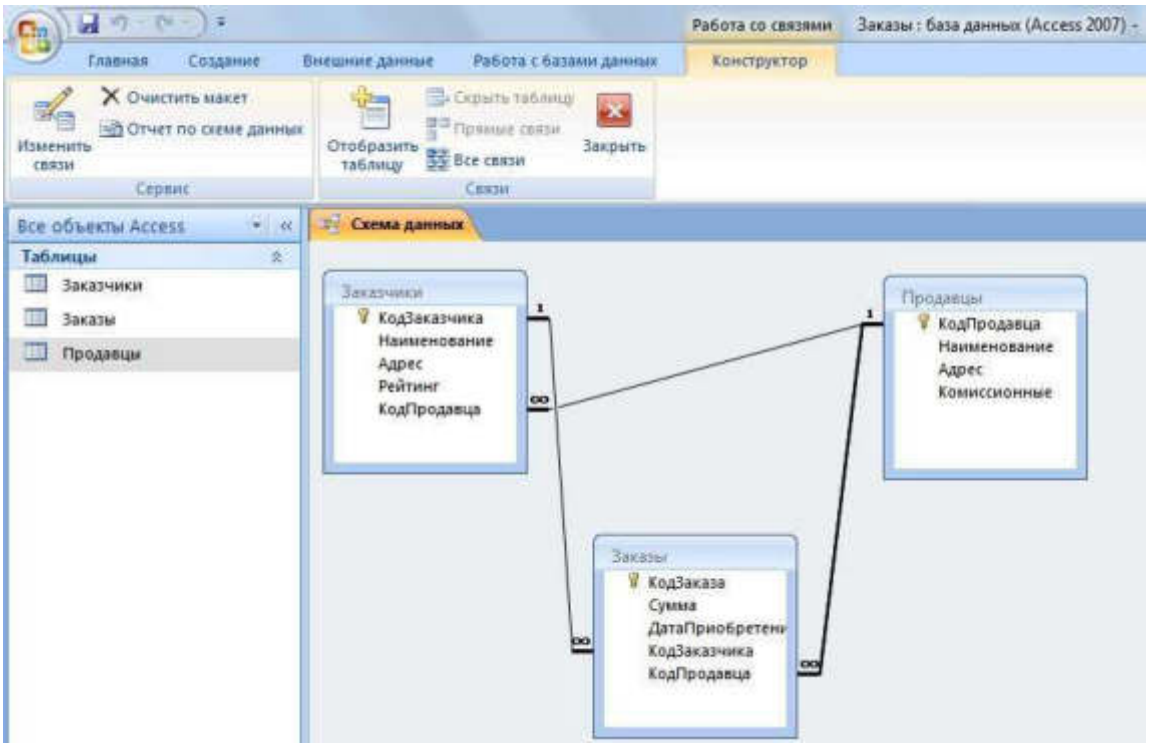


Рисунок 2 – Схема данных базы данных «Заказы»

Контрольные вопросы:

1. Какие типы связей вы знаете?
2. Дайте описание и приведите пример связи один к одному.
3. Дайте описание и приведите пример связи один ко многим.
4. Дайте описание и приведите пример связи многие ко многим.
5. Назовите и опишите два основных типа ограничений по условию целостности данных в базе.

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №3

Тема: «Основы реляционной алгебры»

Цель: Познакомиться с основами реляционной алгебры. Научиться применять ее на практике при проектировании БД.

Время выполнения: 2 урока

Ход работы:

Реляционная алгебра — замкнутая система операций над отношениями в реляционной модели данных. Операции реляционной алгебры также называют реляционными операциями.

Первоначальный набор из 8 операций был предложен Э. Коддом в 1970-е годы и включал как операции, которые до сих пор используются (проекция, соединение и т.д.), так и операции, которые не вошли в употребление (например, деление отношений).

В процессе развития реляционной теории и практики было предложено несколько новых реляционных операций, например, полусоединение (SEMI-JOIN) и полуразность, или анти-полусоединение (ANTI-SEMI-JOIN), CROSS APPLY и OUTER APPLY, транзитивное замыкание (TCLOSE) и др.

Обзор реляционной алгебры

Третья часть реляционной модели, манипуляционная часть, утверждает, что доступ к реляционным данным осуществляется при помощи реляционной алгебры или эквивалентного ему реляционного исчисления.

В реализациях конкретных реляционных СУБД сейчас не используется в чистом виде ни реляционная алгебра, ни реляционное исчисление. Фактическим стандартом доступа к реляционным данным стал язык SQL (Structured Query Language). Язык SQL представляет собой смесь операторов реляционной алгебры и выражений реляционного исчисления, использующий синтаксис, близкий к фразам английского языка и расширенный дополнительными возможностями, отсутствующими в реляционной алгебре и реляционном исчислении. Вообще, язык доступа к данным называется ***реляционно полным***,

если он по выразительной силе не уступает реляционной алгебре (или, что то же самое, реляционному исчислению), т.е. любой оператор реляционной алгебры может быть выражен средствами этого языка. Именно таким и является язык SQL.

Замкнутость реляционной алгебры

Реляционная алгебра представляет собой набор операторов, использующих отношения в качестве аргументов, и возвращающие отношения в качестве результата. Таким образом, реляционный оператор f выглядит как функция с отношениями в качестве аргументов:

$$R = f(R_1, R_2, \ldots, R_n)$$

Реляционная алгебра является замкнутой, т.к. в качестве аргументов в реляционные операторы можно подставлять другие реляционные операторы, подходящие по типу:

$$R = f(f_1(R_{11}, R_{12}, \ldots) f_2(R_{21}, R_{22}, \ldots), \ldots)$$

Таким образом, в реляционных выражениях можно использовать вложенные выражения сколь угодно сложной структуры.

Каждое отношение обязано иметь уникальное имя в пределах базы данных. Имя отношения, полученного в результате выполнения реляционной операции, определяется в левой части равенства. Однако можно не требовать наличия имен от отношений, полученных в результате реляционных выражений, если эти отношения подставляются в качестве аргументов в другие реляционные выражения. Такие отношения будем называть ***неименованными отношениями***. Неименованные отношения реально не существуют в базе данных, а только вычисляются в момент вычисления значения реляционного оператора.

Традиционно, вслед за Коддом, определяют восемь реляционных операторов, объединенных в две группы.

Теоретико-множественные операторы:

- Объединение
- Пересечение
- Вычитание

- Декартово произведение

Специальные реляционные операторы:

- Выборка
- Проекция
- Соединение
- Деление

Не все они являются независимыми, т.е. некоторые из этих операторов могут быть выражены через другие реляционные операторы.

Отношения, совместимые по типу

Некоторые реляционные операторы (например, объединение) требуют, чтобы отношения имели одинаковые заголовки. Действительно, отношения состоят из заголовка и тела. Операция объединения двух отношений есть просто объединение двух множеств кортежей, взятых из тел соответствующих отношений. Но будет ли результат отношением? Во-первых, если исходные отношения имеют разное количество атрибутов, то, очевидно, что множество, являющееся объединением таких разнотипных кортежей нельзя представить в виде отношения. Во-вторых, пусть даже отношения имеют одинаковое количество атрибутов, но атрибуты имеют различные наименования. Как тогда определить заголовок отношения, полученного в результате объединения множеств кортежей? В-третьих, пусть отношения имеют одинаковое количество атрибутов, атрибуты имеют одинаковые наименования, но определенны на различных доменах. Тогда снова объединение кортежей не будет образовывать отношение.

Определение 1. Будем называть отношения ***совместимыми по типу***, если они имеют идентичные заголовки, а именно,

Отношения имеют *одно и то же множество имен атрибутов*, т.е. для любого атрибута в одном отношении найдется атрибут с таким же наименованием в другом отношении,

Атрибуты с одинаковыми именами *определены на одних и тех же доменах*.

Некоторые отношения не являются совместимыми по типу, но становятся таковыми после некоторого переименования атрибутов. Для того чтобы такие отношения можно было использовать в реляционных операторах, вводится вспомогательный ***оператор переименования атрибутов***.

Оператор переименования атрибутов

Оператор переименования атрибутов имеет следующий синтаксис:

R RENAME $Atr_1, Atr_2, \ldots$ AS $NewAtr_1, NewAtr_2, \ldots$

где

R - отношение,

$Atr_1, Atr_2, \ldots$ - исходные имена атрибутов,

$NewAtr_1, NewAtr_2, \ldots$ - новые имена атрибутов.

В результате применения оператора переименования атрибутов получаем новое отношение, с измененными именами атрибутов.

Пример 1.

Следующий оператор возвращает неименованное отношение, в котором атрибут $City_Num$ переименован в $CityId$:

$City$ RENAME $City_Num$ AS $CityId$

Теоретико-множественные операторы

Объединение

Определение 2. ***Объединением*** двух совместимых по типу отношений A и B называется отношение с тем же заголовком, что и у отношений A и B, и телом, состоящим из кортежей, принадлежащих или A, или B, или обоим отношениям.

Синтаксис операции объединения:

A UNION B

Замечание. Объединение, как и любое отношение, не может содержать одинаковых кортежей. Поэтому, если некоторый кортеж входит и в отношение A, и отношение B, то в объединение он входит один раз.

Пример 2. Пусть даны два отношения A и B с информацией о сотрудниках:

Табельный номер	**Фамилия**	**Зарплата**
1	Иванов	1000
2	Петров	2000
3	Сидоров	3000

Таблица 1 Отношение A

Табельный номер	**Фамилия**	**Зарплата**
1	Иванов	1000
2	Пушников	2500
4	Сидоров	3000

Таблица 2 Отношение B

Объединение отношений A и B будет иметь вид:

Табельный номер	**Фамилия**	**Зарплата**
1	Иванов	1000
2	Петров	2000
3	Сидоров	3000
2	Пушников	2500
4	Сидоров	3000

Таблица 3 Отношение A UNION B

Замечание. Как видно из приведенного примера, потенциальные ключи, которые были в отношениях A и B *не наследуются* объединением этих отношений. Поэтому, в объединении отношений A и B атрибут "Табельный номер" может содержать дубликаты значений. Если бы это было не так, и ключи наследовались бы, то это противоречило бы понятию объединения как "объединение множеств". Конечно, объединение отношений A и B имеет, как и любое отношение, потенциальный ключ, например, состоящий из всех атрибутов.

Пересечение

Определение 3. ***Пересечением*** двух совместимых по типу отношений A и B называется отношение с тем же заголовком, что и у отношений A и B, и телом, состоящим из кортежей, принадлежащих одновременно обоим отношениям A и B.

Синтаксис операции пересечения:

A INTERSECT B

Пример 3. Для тех же отношений A и B, что и в предыдущем примере пересечение имеет вид:

Табельный номер	**Фамилия**	**Зарплата**
1	Иванов	1000

Таблица 4 Отношение A INTERSECT B

Замечание. Казалось бы, что в отличие от операции объединения, потенциальные ключи могли бы наследоваться пересечением отношений. Однако это не так. Вообще, *никакие реляционные операторы не передают результатирующему отношению никаких данных о потенциальных ключах*. В качестве причины этого можно было бы привести тривиальное соображение, что так получается более просто и симметрично - все операторы устроены одинаково. На самом деле причина более глубока, и заключается в том, что потенциальный ключ - семантическое понятие, отражающее различимость объектов предметной области. Наличие потенциальных ключей *не выводится* из структуры отношения, а явно задается для каждого отношения, исходя из его смысла. Реляционные же операторы являются *формальными* операциями над отношениями и выполняются одинаково, независимо от смысла данных, содержащихся в отношениях. Поэтому, реляционные операторы ничего не могут "знать" о смысле данных. Трактовка результата реляционных операций - дело пользователя.

Вычитание

Определение 4. ***Вычитанием*** двух совместимых по типу отношений A и B называется отношение с тем же заголовком, что и у отношений A и B, и

телом, состоящим из кортежей, принадлежащих отношению A и не принадлежащих отношению B.

Синтаксис операции вычитания:

A MINUS B

Пример 4. Для тех же отношений A и B, что и в предыдущем примере вычитание имеет вид:

Табельный номер	**Фамилия**	**Зарплата**
2	Петров	2000
3	Сидоров	3000

Таблица 5 Отношение A MINUS B

Декартово произведение

Определение 5. ***Декартовым произведением*** двух отношений $A(A_1, A_2, \ldots, A_n)$ и $B(B_1, B_2, \ldots, B_m)$ называется отношение, заголовок которого является ***сцеплением заголовков*** отношений A и B:

$(A_1, A_2, \ldots, A_n, B_1, B_2, \ldots, B_m)$,

а тело состоит из кортежей, являющихся ***сцеплением кортежей*** отношений A и B:

$(a_1, a_2, \ldots a_n, b_1, b_2, \ldots, b_m)$,

таких, что $(a_1, a_2, \ldots, a_n) \in A$, $(b_1, b_2, \ldots b_m) \in B$.

Синтаксис операции декартового произведения:

A TIMES B

Замечание. Мощность произведения A TIMES B равна произведению мощностей отношений A и B, т.к. каждый кортеж отношения A соединяется с каждым кортежем отношения B.

Замечание. Если в отношения A и B имеются атрибуты с одинаковыми наименованиями, то перед выполнением операции декартового произведения такие атрибуты необходимо переименовать.

Замечание. Перемножать можно любые два отношения, совместимость по типу при этом не требуется.

Пример 5. Пусть даны два отношения A и B с информацией о поставщиках и деталях:

Номер поставщика	**Наименование поставщика**
1	Иванов
2	Петров
3	Сидоров

Таблица 6 Отношение A (Поставщики)

Номер детали	**Наименование детали**
1	Болт
2	Гайка
3	Винт

Таблица 7 Отношение B (Детали)

Декартово произведение отношений A и B будет иметь вид:

Номер поставщика	**Наименование поставщика**	**Номер детали**	**Наименование детали**
1	Иванов	1	Болт
1	Иванов	2	Гайка
1	Иванов	3	Винт
2	Петров	1	Болт
2	Петров	2	Гайка
2	Петров	3	Винт
3	Сидоров	1	Болт
3	Сидоров	2	Гайка
3	Сидоров	3	Винт

Таблица 8 Отношение A TIMES B

Замечание. Сама по себе операция декартового произведения не очень важна, т.к. она не дает никакой новой информации, по сравнению с исходными отношениями. Для реальных запросов эта операция почти никогда не используется. Однако операция декартового произведения важна для выполнения специальных реляционных операций, о которых речь пойдет ниже.

Специальные реляционные операторы

Выборка (ограничение, селекция)

Определение 6. ***Выборкой (ограничением, селекцией)*** на отношении A с условием c называется отношение с тем же заголовком, что и у отношения A, и телом, состоящем из кортежей, значения атрибутов которых при подстановке в условие c дают значение ИСТИНА. c представляет собой логическое выражение, в которое могут входить атрибуты отношения A и (или) скалярные выражения.

В простейшем случае условие c имеет вид $X \theta Y$, где θ - один из операторов сравнения ($=, \neq, <, \leq, >, \geq$ и т.д.), а X и Y - атрибуты отношения A или скалярные значения. Такие выборки называются θ-***выборки*** (***тэта-выборки***) или θ-***ограничения***, θ-***селекции***.

Синтаксис операции выборки:

A WHERE c,

или

A WHERE $X \theta Y$

Пример 6. Пусть дано отношение A с информацией о сотрудниках:

Табельный номер	**Фамилия**	**Зарплата**
1	Иванов	1000
2	Петров	2000
3	Сидоров	3000

Таблица 9 Отношение A

Результат выборки A WHERE Зарплата < 3000 будет иметь вид:

Табельный номер	**Фамилия**	**Зарплата**

1	Иванов	1000
2	Петров	2000

Таблица 10 Отношение A WHERE Зарплата <3000

Смысл операции выборки очевиден - выбрать кортежи отношения, удовлетворяющие некоторому условию. Таким образом, операция выборки дает "*горизонтальный срез*" отношения по некоторому условию.

Проекция

Определение 7. ***Проекцией*** отношения A по атрибутам $X,Y,\ldots,Z$, где каждый из атрибутов принадлежит отношению A, называется отношение с заголовком $(X,Y,\ldots Z)$ и телом, содержащим множество кортежей вида $(x,y,\ldots z)$, таких, для которых в отношении A найдутся кортежи со значением атрибута X равным x, значением атрибута Y равным y, …, значением атрибута Z равным z.

Синтаксис операции проекции:

$A[X,Y,\ldots,Z]$

Замечание. Операция проекции дает "*вертикальный срез*" отношения, в котором удалены все возникшие при таком срезе дубликаты кортежей.

Пример 7. Пусть дано отношение A с информацией о поставщиках, включающих наименование и месторасположение:

Номер поставщика	**Наименование поставщика**	**Город поставщика**
1	Иванов	Уфа
2	Петров	Москва
3	Сидоров	Москва
4	Сидоров	Челябинск

Таблица 11 Отношение A (Поставщики)

Проекция $A[\text{Город поставщика}]$ будет иметь вид:

Город поставщика

Уфа
Москва
Челябинск

Таблица 12 Отношение A[Город поставщика]

Соединение

Операция соединения отношений, наряду с операциями выборки и проекции, является одной из наиболее важных реляционных операций.

Обычно рассматривается несколько разновидностей операции соединения:

Общая операция соединения

Θ -соединение (тэта-соединение)

Экви-соединение

Естественное соединение

Наиболее важным из этих частных случаев является операция естественного соединения. Все разновидности соединения являются частными случаями общей операции соединения.

Общая операция соединения

Определение 8. ***Соединением*** отношений A и B по условию c называется отношение

$(A \text{ TIMES } B) \text{ WHERE } c$

c представляет собой логическое выражение, в которое могут входить атрибуты отношений A и B и (или) скалярные выражения.

Таким образом, операция соединения есть результат последовательного применения операций декартового произведения и выборки. Если в отношениях A и B имеются атрибуты с одинаковыми наименованиями, то перед выполнением соединения такие атрибуты необходимо переименовать.

Тэта-соединение

Определение 9. Пусть отношение A содержит атрибут X, отношение B содержит атрибут Y, а Θ - один из операторов сравнения ($=, \neq, <, \leq, >, \geq$ и т.д.).

Тогда Θ ***-соединением*** отношения A по атрибуту X с отношением B по атрибуту Y называют отношение

$(A \text{ TIMES } B) \text{ WHERE } X \Theta Y$

Это частный случай операции общего соединения.

Иногда, для операции Θ-соединения применяют следующий, более короткий синтаксис:

$A[X \Theta Y]B$

Пример 8. Рассмотрим некоторую компанию, в которой хранятся данные о поставщиках и поставляемых деталях. Пусть поставщикам и деталям присвоен некий статус. Пусть бизнес компании организован таким образом, что поставщики имеют право поставлять только те детали, статус которых не выше статуса поставщика (смысл этого может быть в том, что хороший поставщик с высоким статусом может поставлять больше разновидностей деталей, а плохой поставщик с низким статусом может поставлять только ограниченный список деталей, важность которых (статус детали) не очень высока).

Номер поставщика	Наименование поставщика	**X (Статус поставщика)**
1	Иванов	4
2	Петров	1
3	Сидоров	2

Таблица 13 Отношение A (Поставщики)

Номер детали	**Наименование детали**	**Y (Статус детали)**
1	Болт	3
2	Гайка	2
3	Винт	1

Таблица 14 Отношение B (Детали)

Ответ на вопрос "какие поставщики имеют право поставлять какие детали?" дает Θ -соединение $A[X \geq Y]B$:

Номер поставщика	Наименование поставщика	X (Статус поставщика)	Номер детали	Наименование детали	Y (Статус детали)
1	Иванов	4	1	Болт	3
1	Иванов	4	2	Гайка	2
1	Иванов	4	3	Винт	1
2	Петров	1	3	Винт	1
3	Сидоров	2	2	Гайка	2
3	Сидоров	2	3	Винт	1

Таблица 15 Отношение "Какие поставщики поставляют какие детали"

Экви-соединение

Наиболее важным частным случаем Θ -соединения является случай, когда Θ есть просто равенство.

Синтаксис ***экви-соединения***:

$A[X = Y]B$

Пример 9. Пусть имеются отношения P, D и PD, хранящие информацию о поставщиках, деталях и поставках соответственно (для удобства введем краткие наименования атрибутов):

Номер поставщика PNUM	**Наименование поставщика PNAME**
1	Иванов
2	Петров
3	Сидоров

Таблица 16 Отношение P (Поставщики)

Номер детали DNUM	**Наименование детали DNAME**
1	Болт
2	Гайка
3	Винт

Таблица 17 Отношение D (Детали)

Номер поставщика PNUM	*Номер детали DNUM*	**Поставляемое количество VOLUME**
1	*1*	100
1	*2*	200
1	*3*	300
2	*1*	150
2	*2*	250
3	*1*	1000

Таблица 18 Отношение PD (Поставки)

Ответ на вопрос, какие детали поставляются поставщиками, дает экви-соединение $P[PNUM = PNUM]PD$. На самом деле, т.к. в отношениях имеются одинаковые атрибуты, то требуется сначала переименовать атрибуты, а потом выполнить экви-соединение. Запись становится более громоздкой:

$$(P \text{ RENAME } PNUM \text{ AS } PNUM1)[PNUM1 = PNUM2](PD \text{ RENAME } PNUM \text{ AS } PNUM2)$$

Обычно, такой сложной формой записи не пользуются. Но как бы то ни было, в результате имеем отношение:

Номер поставщика PNUM1	**Наименование поставщика PNAME**	**Номер поставщика PNUM2**	**Номер детали DNUM**	**Поставляемое количество VOLUME**

1	Иванов	1	1	100
1	Иванов	1	2	200
1	Иванов	1	3	300
2	Петров	2	1	150
2	Петров	2	2	250
3	Сидоров	3	1	1000

Таблица 19 Отношение "Какие детали поставляются какими поставщиками"

Недостатком экви-соединения является то, что если соединение происходит по атрибутам с одинаковыми наименованиями (а так чаще всего и происходит!), то в результирующем отношении появляется два атрибута с одинаковыми значениями. В нашем примере атрибуты PNUM1 и PNUM2 содержат дублирующие данные. Избавиться от этого недостатка можно, взяв проекцию по всем атрибутам, кроме одного из дублирующих. Именно так действует естественное соединение.

Естественное соединение

Определение 10. Пусть даны отношения $A(A_1,A_2,\ldots,A_n,X_1,X_2,\ldots,X_p)$ и $B(X_1,X_2,\ldots,X_p,B_1,B_2,\ldots,B_m)$, имеющие одинаковые атрибуты $X_1,X_2,\ldots,X_p$ (т.е. атрибуты с одинаковыми именами и определенные на одинаковых доменах).

Тогда ***естественным соединением*** отношений A и B называется отношение с заголовком $(A_1,A_2,\ldots,A_n,X_1,X_2,\ldots,X_p,B_1,B_2,\ldots,B_m)$ и телом, содержащим множество кортежей $(a_1,a_2,\ldots a_n,x_1,x_2,\ldots,x_p,b_1,b_2,\ldots,b_m)$, таких, что $(a_1,a_2,\ldots a_n,x_1,x_2,\ldots,x_p)\in A$ и $(x_1,x_2,\ldots,x_p,b_1,b_2,\ldots b_m)\in B$.

Естественное соединение настолько важно, что для него используют специальный синтаксис:

A JOIN B

Замечание. В синтаксисе естественного соединения не указываются, по каким атрибутам производится соединение. Естественное соединение производится *по всем* одинаковым атрибутам.

Замечание. Естественное соединение эквивалентно следующей последовательности реляционных операций:

Переименовать одинаковые атрибуты в отношениях

Выполнить декартово произведение отношений

Выполнить выборку по совпадающим значениям атрибутов, имевших одинаковые имена

Выполнить проекцию, удалив повторяющиеся атрибуты

Переименовать атрибуты, вернув им первоначальные имена

Замечание. Можно выполнять последовательное естественное соединение нескольких отношений. Нетрудно проверить, что естественное соединение (как, впрочем, и соединение общего вида) обладает свойством ***ассоциативности***, т.е.

$$(A \text{ JOIN } B) \text{ JOIN } C = A \text{ JOIN } (B \text{ JOIN } C)$$

поэтому такие соединения можно записывать, опуская скобки:

$$A \text{ JOIN } B \text{ JOIN } C$$

Пример 10. В предыдущем примере ответ на вопрос "какие детали поставляются поставщиками", более просто записывается в виде естественного соединения трех отношений $P \text{ JOIN } PD \text{ JOIN } D$ (для удобства просмотра порядок атрибутов изменен, это является допустимым по свойствам отношений):

Номер поставщика PNUM	**Наименование поставщика PNAME**	**Номер детали DNUM**	**Наименование детали DNAME**	**Поставляемое количество VOLUME**
1	Иванов	1	Болт	100
1	Иванов	2	Гайка	200
1	Иванов	3	Винт	300
2	Петров	1	Болт	150

2	Петров	2	Гайка	250
3	Сидоров	1	Болт	1000

Таблица 20 Отношение P JOIN PD JOIN D

Деление

Определение 11. Пусть даны отношения $A(X_1,X_2,\ldots,X_n,Y_1,Y_2,\ldots,Y_m)$ и $B(Y_1,Y_2,\ldots,Y_m)$, причем атрибуты $Y_1,Y_2,\ldots Y_m$ - общие для двух отношений. ***Делением отношений*** A на B называется отношение с заголовком $(X_1,X_2,\ldots,X_n)$ и телом, содержащим множество кортежей $(x_1,x_2,\ldots,x_n)$, таких, что для *всех* кортежей $(y_1,y_2,\ldots y_m)\in B$ в отношении A найдется кортеж $(x_1,x_2,\ldots,x_n,y_1,y_2,\ldots y_m)$.

Отношение A выступает в роли ***делимого***, отношение B выступает в роли ***делителя***. Деление отношений аналогично делению чисел с остатком.

Синтаксис операции деления:

A DEVIDBY B

Замечание. Типичные запросы, реализуемые с помощью операции деления, обычно в своей формулировке имеют слово "все" - "какие поставщики поставляют *все* детали?".

Пример 11. В примере с поставщиками, деталями и поставками ответим на вопрос, "какие поставщики поставляют *все* детали?".

В качестве делимого возьмем проекцию $X = PD[PNUM, DNUM]$, содержащую номера поставщиков и номера поставляемых ими деталей:

Номер поставщика PNUM	**Номер детали DNUM**
1	1
1	2
1	3
2	1

2	2
3	1

Таблица 21 Проекция X=PD[PNUM,DNUM]

В качестве делителя возьмем проекцию $Y = D[DNUM]$, содержащую список номеров *всех* деталей (не обязательно поставляемых кем-либо):

Номер детали DNUM
1
2
3

Таблица 22 Проекция Y=D[DNUM]

Деление X DEVIDEBY Y дает список номеров поставщиков, поставляющих *все* детали:

Номер поставщика PNUM
1

Таблица 23 Отношение X DEVIDEBY Y

Оказалось, что только поставщик с номером 1 поставляет все детали.

Выполнить ниже перечисленные задания для следующей таблицы:

Номер поставщика PNUM	**Наименование поставщика PNAME**	**Номер детали DNUM**	**Наименование детали DNAME**	**Поставляемое количество VOLUME**
1	Иванов	1	Болт	100
1	Иванов	2	Гайка	200
1	Иванов	3	Винт	300

2	Петров	1	Болт	150
2	Петров	2	Гайка	250
3	Сидоров	1	Болт	1000

Задание 1: Получить имена поставщиков, поставляющих деталь номер 2.

Задание 2: Получить имена поставщиков, поставляющих по крайней мере одну гайку.

Задание 3: Получить имена поставщиков, поставляющих все детали.

Задание 4: Получить имена поставщиков, не поставляющих деталь номер 2.

Контрольные вопросы:

1. Что такое реляционная алгебра?
2. Причислите и опишите теоретико-множественные операторы.
3. Причислите и опишите специальные реляционные операторы.
4. Почему реляционная алгебра является замкнутой?

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №4

Тема: «Этапы разработки баз данных»

Цель: Познакомиться с этапами разработки БД. Научиться создавать дорожные карты и сопутствующую документацию на этапе проектирования.

Время выполнения: 2 урока

Ход работы:

При разработке БД можно выделить следующие этапы работы:

I этап. Постановка задачи.

На этом этапе формируется задание по созданию БД. В нем подробно описывается состав базы, назначение и цели ее создания, а также перечисляется, какие виды работ предполагается осуществлять в этой базе данных (отбор, дополнение, изменение данных, печать или вывод отчета и т. д).

II этап. Анализ объекта.

На этом этапе рассматривается, из каких объектов может состоять БД, каковы свойства этих объектов. После разбиения БД на отдельные объекты необходимо рассмотреть свойства каждого из этих объектов, или, другими словами, установить, какими параметрами описывается каждый объект. Все эти сведения можно располагать в виде отдельных записей и таблиц. Далее необходимо рассмотреть тип данных каждой отдельной единицы записи. Сведения о типах данных также следует занести в составляемую таблицу.

III этап. Синтез модели.

На этом этапе по проведенному выше анализу необходимо выбрать определенную модель БД. Далее рассматриваются достоинства и недостатки каждой модели и сопоставляются с требованиями и задачами создаваемой БД. После такого анализа выбирают ту модель, которая сможет максимально

обеспечить реализацию поставленной задачи. После выбора модели необходимо нарисовать ее схему с указанием связей между таблицами или узлами.

IV этап. Выбор способов представления информации и программного инструментария.

После создания модели необходимо, в зависимости от выбранного программного продукта, определить форму представления информации.

В большинстве СУБД данные можно хранить в двух видах:

- с использованием форм;
- без использования форм.

Форма – это созданный пользователем графический интерфейс для ввода данных в базу.

V этап. Синтез компьютерной модели объекта.

В процессе создания компьютерной модели можно выделить некоторые стадии, типичные для любой СУБД.

Стадия 1. Запуск СУБД, создание нового файла базы данных или открытие созданной ранее базы.

Стадия 2. Создание исходной таблицы или таблиц.

Создавая исходную таблицу, необходимо указать имя и тип каждого поля. Имена полей не должны повторяться внутри одной таблицы. В процессе работы с БД можно дополнять таблицу новыми полями. Созданную таблицу необходимо сохранить, дав ей имя, уникальное в пределах создаваемой базы.

При проектировании таблиц, рекомендуется руководствоваться следующими основными принципами:

1. Информация в таблице не должна дублироваться. Не должно быть повторений и между таблицами. Когда определенная информация хранится только в одной таблице, то и изменять ее придется только в одном месте. Это

делает работу более эффективной, а также исключает возможность несовпадения информации в разных таблицах. Например, в одной таблице должны содержаться адреса и телефоны клиентов.

2. Каждая таблица должна содержать информацию только на одну тему. Сведения на каждую тему обрабатываются намного легче, если они содержатся в независимых друг от друга таблицах. Например, адреса и заказы клиентов лучше хранить в разных таблицах, с тем, чтобы при удалении заказа информация о клиенте осталась в базе данных.

3. Каждая таблица должна содержать необходимые поля. Каждое поле в таблице должно содержать отдельные сведения по теме таблицы. Например, в таблице с данными о клиенте могут содержаться поля с названием компании, адресом, городом, страной и номером телефона. При разработке полей для каждой таблицы необходимо помнить, что каждое поле должно быть связано с темой таблицы. Не рекомендуется включать в таблицу данные, которые являются результатом выражения. В таблице должна присутствовать вся необходимая информация. Информацию следует разбивать на наименьшие логические единицы (Например, поля "Имя" и "Фамилия", а не общее поле "Имя").

4. База данных должна иметь первичный ключ. Это необходимо для того, чтобы СУБД могла связать данные из разных таблиц, например, данные о клиенте и его заказы.

Стадия 3. Создание экранных форм.

Первоначально необходимо указать таблицу, на базе которой будет создаваться форма. Ее можно создавать при помощи мастера форм, указав, какой вид она должна иметь, или самостоятельно. При создании формы можно указывать не все поля, которые содержит таблица, а только некоторые из них. Имя формы может совпадать с именем таблицы, на базе которой она создана. На основе одной таблицы можно создать несколько форм, которые могут отличаться видом или количеством используемых из данной таблицы

полей. После создания форму необходимо сохранить. Созданную форму можно редактировать, изменяя местоположение, размеры и формат полей.

Стадия 4. Заполнение БД.

Процесс заполнения БД может проводиться в двух видах: в виде таблицы и в виде формы. Числовые и текстовые поля можно заполнять в виде таблицы, а поля типа MEMO и OLE – в виде формы.

VI этап. Работа с созданной базой данных.

Работа с БД включает в себя следующие действия:

- поиск необходимых сведений;
- сортировка данных;
- отбор данных;
- вывод на печать;
- изменение и дополнение данных.

Задание: Создать проект базы данных «Учебная часть» согласно следующим исходным данным:

В БД «Деканат» должны храниться данные о студентах, группах студентов, об оценках студентов по различным дисциплинам, о преподавателях, о стипендиях и т.д. Ограничимся данными о студентах, группах студентов и об оценках студентов по различным дисциплинам. Определим сущности, атрибуты сущностей и основные требования к функциям БД с ограниченными данными.

Основными предметно-значимыми сущностями БД «Учебная часть» являются: Студенты, Группы студентов, Дисциплины, Успеваемость.

1. Определить основные предметно-значимые атрибуты сущностей.
2. Определить основные требования к функциям БД.
3. Определить логические связи между сущностями.

Контрольные вопросы:

1. Перечислите основные этапы разработки БД.
2. Дайте характеристику этапу анализа модели.
3. Дайте характеристику этапу синтеза модели.

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №5

Тема: «Проектирование на основе модели типа объект – отношение»

Цель: Познакомиться с правилами и алгоритмом проектирования БД на основе модели типа объект-отношение.

Время выполнения: 2 урока

Ход работы:

Имеется целый ряд методик создания информационно-логических моделей. Одна из наиболее популярных в настоящее время методик при разработке моделей использует ERD (Entity-Relationship Diagrams). В русскоязычной литературе эти диаграммы называют «объект — отношение» либо «сущность — связь». Модель ERD была предложена Питером Пин Шен Ченом в 1976 г. К настоящему времени разработано несколько ее разновидностей, но все они базируются на графических диаграммах, предложенных Ченом. Диаграммы конструируются из небольшого числа компонентов. Благодаря наглядности представления они широко используются в CASE-средствах (Computer Aided Software Engineering).

Рассмотрим используемую терминологию и обозначения.

Сущность (Entity)— реальный либо воображаемый объект, имеющий существенное значение для рассматриваемой предметной области, информация о котором подлежит хранению.

Каждая сущность должна обладать уникальным идентификатором. Каждый экземпляр сущности должен однозначно идентифицироваться и отличаться от всех других экземпляров данного типа (сущности). Каждая сущность должна обладать некоторыми свойствами:

• иметь уникальное имя; причем к этому имени должна всегда применяться одна и та же интерпретация (определение сущности). И наоборот: одна и та же интерпретация не может применяться к различным именам, если только они не являются псевдонимами;

• обладать одним или несколькими атрибутами, которые либо принадлежат сущности, либо наследуются ею через связь;

• обладать одним или несколькими атрибутами, которые однозначно идентифицируют каждый экземпляр сущности.

Сущность может быть независимой либо зависимой. Признаком зависимой сущности служит наличие у нее наследуемых через связь атрибутов (рис. 1.).

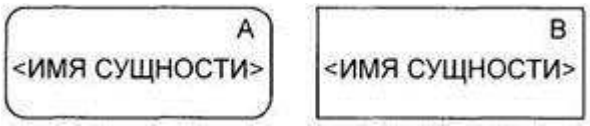


Рис. 1. Графическое обозначение сущностей: A — зависимая, B — независимая

Каждая сущность может обладать любым количеством связей с другими сущностями модели.

Связь (Relationship) — поименованная ассоциация между двумя сущностями, значимая для рассматриваемой предметной области. Одна из участвующих в связи сущностей — независимая, называется родительской сущностью, другая — зависимая, называется дочерней или сущностью-потомком. Как правило, каждый экземпляр родительской сущности ассоциирован с произвольным (в том числе нулевым) количеством экземпляров дочерней сущности. Каждый экземпляр сущности-потомка ассоциирован в точности с одним экземпляром сущности-родителя. Таким образом, экземпляр сущности-потомка может существовать только при существовании сущности-родителя.

Связи дается имя, выражаемое грамматическим оборотом глагола и помещаемое возле линии связи. Имя каждой связи между двумя данными сущностями должно быть уникальным, но имена связей в модели не обязаны быть уникальными. Каждая связь имеет определение. Определение связи образуют соединением имени сущности-родителя, имени связи, выражения степени связи и имени сущности-потомка.

НОТАЦИЯ	ОБОЗНАЧЕНИЕ СВЯЗИ
IDEF1	Много Один
IE	Много Один
Любая	Необязательная Обязательная

Например, связь продавца с контрактом может быть определена следующим образом:

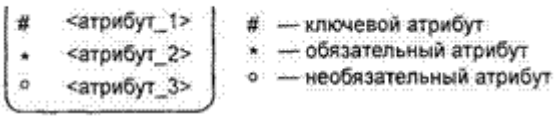
• продавец может получить вознаграждение за один или более Контрактов;

• контракт должен быть инициирован ровно одним Продавцом.

На диаграмме связь изображается отрезком (ломаной). Концы отрезка с помощью специальных обозначений (рис. 2) указывают степень связи. Кроме того, характер линии — штриховая или сплошная, указывает обязательность связи.

Атрибут — любая характеристика сущности, значимая для рассматриваемой предметной области. Он предназначен для квалификации, идентификации, классификации, количественной характеристики или выражения состояния сущности. Атрибут представляет тип характеристик (свойств), ассоциированных с множеством реальных или абстрактных объектов (людей, мест, событий, состояний, идей, пар предметов и т. д.) (рис. 3). Экземпляр атрибута — это определенная характеристика конкретного экземпляра сущности. Экземпляр атрибута определяется типом характеристики (например, «Цвет») и ее значением (например, «лиловый»), называемым значением атрибута. В ER-модели атрибуты ассоциируются с конкретными сущностями. Каждый экземпляр сущности должен обладать одним конкретным значением для каждого своего атрибута.

Рис. 3. Графическое отображение характеристики



Атрибут может быть либо *обязательным*, либо *необязательным*. Обязательность означает, что атрибут не может принимать неопределенных значений (null values). Атрибут может быть либо описательным (т. е. обычным дескриптором сущности), либо входить в состав уникального идентификатора (первичного ключа).

Уникальный идентификатор — это атрибут или совокупность атрибутов и/или связей, однозначно характеризующая каждый экземпляр данного типа сущности. В случае полной идентификации экземпляр данного типа сущности полностью идентифицируется своими собственными ключевыми атрибутами, в противном случае в идентификации участвуют также атрибуты другой сущности — родителя.

Характер идентификации отображается в диаграмме на линии связи (рис. 4).

Рис. 4. Графическое отображение характера идентификации

Каждый атрибут идентифицируется уникальным именем, выражаемым грамматическим оборотом существительного, описывающим представляемую атрибутом характеристику. Атрибуты изображаются в виде списка имен внутри блока ассоциированной сущности, причем каждый атрибут занимает отдельную строку. Атрибуты, определяющие первичный ключ, размещаются наверху списка и выделяются знаком «#».

Каждая сущность должна обладать хотя бы одним возможным ключом. Возможный ключ сущности — это один или несколько атрибутов, чьи значения однозначно определяют каждый экземпляр сущности. При существовании нескольких возможных ключей один из них обозначается в качестве первичного ключа, а остальные — как альтернативные ключи.

В настоящее время на основе подхода Чена создана методология IDEF1X, которая разработана с учетом таких требований, как простота

изучения и возможность автоматизации. IDEFlX-диаграммы используются рядом распространенных CASE-средств (в частности, ERwin, Design/IDEF).

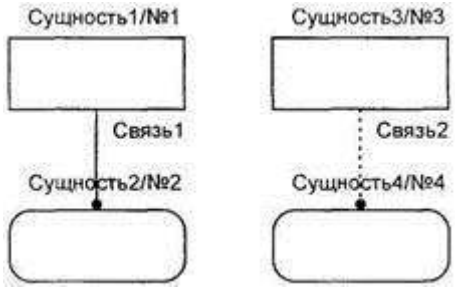


Рис. 5. Графическое изображение сущностей и связей

Сущность в методологии IDEF1X называется независимой от идентификаторов или просто независимой, если каждый экземпляр сущности может быть однозначно идентифицирован без определения его отношений с другими сущностями. Сущность называется зависимой от идентификаторов или просто зависимой, если однозначная идентификация экземпляра сущности зависит от его отношения к другой сущности (рис. 5).

Каждой сущности присваивается уникальное имя и номер, разделяемые косой чертой «/» и помещаемые над блоком.

Если экземпляр сущности-потомка однозначно определяется своей связью с сущностью-родителем, то связь называется идентифицирующей, в противном случае — неидентифицируюшей.

Идентифицирующая связь между сущностью-родителем и сущностью-потомком изображается сплошной линией. На рис. 5: №2 — зависимая сущность, Связь 1 — идентифицирующая связь. Сущность-потомок в идентифицирующей связи является зависимой от идентификатора сущностью. Сущность-родитель в идентифицирующей связи может быть, как независимой, так и зависимой от идентификатора сущностью (это определяется ее связями с другими сущностями).

Штриховая линия изображает неидентифицирующую связь. На рис. 5: №4 — независимая сущность, Связь 2 — неидентифицирующая связь. Сущность-потомок в неидентифицируюшей связи будет независимой от

идентификатора, если она не является также сущностью-потомком в какой-либо идентифицирующей связи.

Связь может дополнительно определяться с помощью указания степени или мощности (количества экземпляров сущности-потомка, которое может существовать для каждого экземпляра сущности-родителя). В IDEF1X могут быть выражены следующие мощности связей:

• каждый экземпляр сущности-родителя может иметь ноль, один или более связанных с ним экземпляров сущности-потомка;

• каждый экземпляр сущности-родителя должен иметь не менее одного связанного с ним экземпляра сущности-потомка;

• каждый экземпляр сущности-родителя должен иметь не более одного связанного с ним экземпляра сущности-потомка;

• каждый экземпляр сущности-родителя связан с некоторым фиксированным числом экземпляров сущности-потомка.

Мощность связи обозначается, как показано на рис. 6 (мощность по умолчанию — *N).*

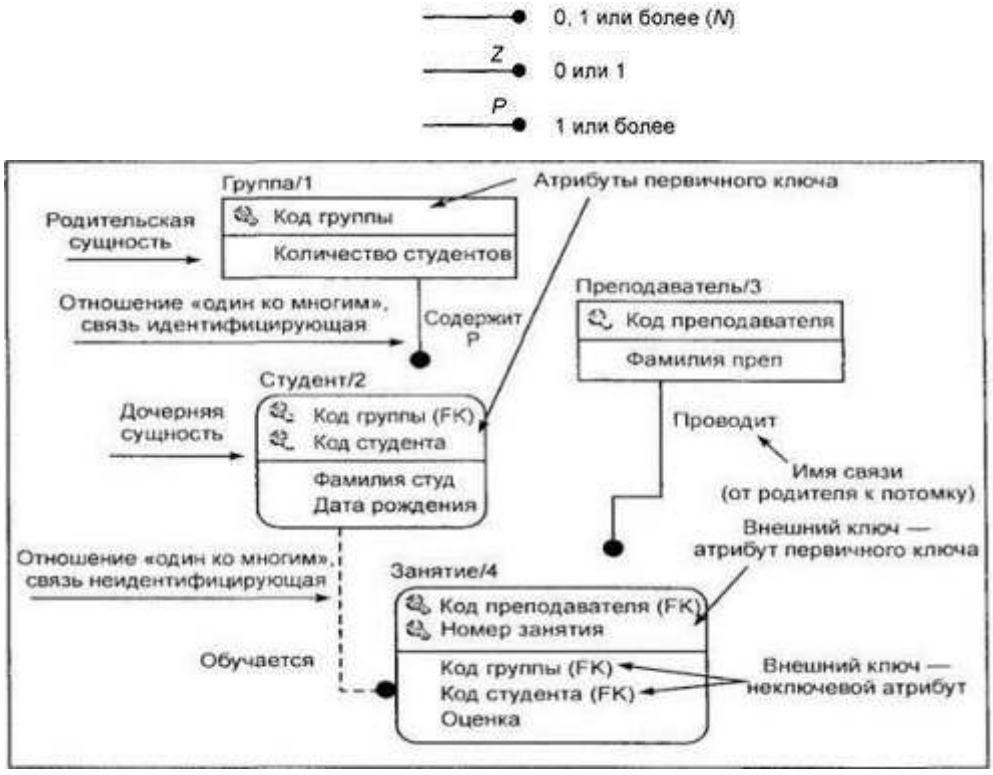


Рис. 7. Атрибуты и первичные

Атрибуты изображаются в виде списка имен внутри блока сущности. Атрибуты, определяющие первичный ключ, размещаются наверху списка и отделяются от других атрибутов горизонтальной чертой (рис. 7).

Сущности могут иметь также внешние ключи (Foreign Key). При идентифицирующей связи они используются в качестве части или целого первичного ключа, при неидентифицирующей — служат неключевыми атрибутами. В списке атрибутов внешний ключ отмечается буквами FK в скобках.

В результате получается информационно-логическая модель, которая используется рядом распространенных CASE-средств, таких, как ERwin, Design/IDEF. В свою очередь, CASE-технологии имеют высокие потенциальные возможности при разработке баз данных и информационных систем, а именно, увеличение производительности труда, улучшение качества программных продуктов, поддержка унифицированного и согласованного стиля работы.

Задание: На основе вышеизложенного построить ER диаграмму для БД «Учебная часть»:

Контрольные вопросы:

1. Перечислите основные этапы построения ERD.
2. Перечислите и опишите составляющие элементы ERD.

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №6

Тема: «Нормализация баз данных»

Цель: Научиться на практике поэтапно проводить нормализацию БД. Организовывать хранение данных в БД согласно требованиям первой, второй и третьей нормальной формы.

Время выполнения: 2 урока

Ход работы:

Требования нормализации.

Реквизиты каждого информационного объекта канонической модели данных должны отвечать требованиям, соответствующим третьей нормальной форме реляционной модели данных:

- Информационный объект должен содержать уникальный идентификатор-ключ;
- Все описательные реквизиты должны быть взаимонезависимы, т.е. между ними не должно быть зависимостей;
- Все реквизиты, входящие в составной ключ, также должны быть взаимонезависимы;
- Каждый описательный реквизит должен функционально полно зависеть от ключа, т.е. каждому значению ключа должно соответствовать только одно значение описательного реквизита, а при составном ключе описательные реквизиты должны зависеть от всей совокупности реквизитов, образующих ключ;
- Каждый описательный реквизит должен зависеть от ключа не транзитивно, т.е. не должен зависеть через другой промежуточный реквизит.

Выполнение требований нормализации обеспечивает построение канонической модели данных и создание ее на основе реляционной базы данных без дублирования описательных данных, а также возможность автоматического поддержания связной целостности данных средствами

СУБД при обновлении базы данных (добавлении и удалении записей, изменении значений в ключевых полях).

Описание нормализации

Нормализация — это процесс организации данных в базе данных, включающий создание таблиц и установление отношений между ними в соответствии с правилами, которые обеспечивают защиту данных и делают базу данных более гибкой, устраняя избыточность и несогласованные зависимости.

Избыточность данных приводит к непродуктивному расходованию свободного места на диске и затрудняет обслуживание баз данных. Например, если данные, хранящиеся в нескольких местах, потребуется изменить, в них придется внести одни и те же изменения во всех этих местах. Изменение адреса клиента гораздо легче реализовать, если в базе данных эти сведения хранятся только в таблице Customers и нигде больше.

Что такое «несогласованные зависимости»? Пользователь, которому нужно узнать, например, адрес определенного клиента, вполне обоснованно будет искать его в таблице Customers (клиенты), но искать в ней сведения о зарплате сотрудника, который работает с этим клиентом, не имеет смысла. Зарплата сотрудника связана с сотрудником (зависит от него), поэтому эти сведения следует хранить в таблице Employees (сотрудники). Несогласованные зависимости могут затруднять доступ к данным, так как путь к данным при этом может отсутствовать или быть неправильным.

Существует несколько правил нормализации баз данных. Каждое правило называется «нормальной формой». Если выполняется первое правило, говорят, что база данных представлена в «первой нормальной форме». Если выполняются три первых правила, считается, что база данных представлена в «третьей нормальной форме». Есть и другие уровни нормализации, однако для большинства приложений достаточно нормализовать базы данных до третьей нормальной формы.

Как и в случае со многими другими формальными правилами и спецификациями, обеспечить полное соответствие реальным ситуациям не всегда возможно. Как правило, для выполнения нормализации приходится создавать дополнительные таблицы, и некоторые клиенты считают это нежелательным. Собираясь нарушить одно из первых трех правил нормализации, убедитесь в том, что в приложении учтены все связанные с этим проблемы, такие как избыточность данных и несогласованные зависимости.

Первая нормальная форма

Устраните повторяющиеся группы в отдельных таблицах.

Создайте отдельную таблицу для каждого набора связанных данных.

Идентифицируйте каждый набор связанных данных с помощью первичного ключа.

Не используйте несколько полей в одной таблице для хранения похожих данных. Например, для слежения за товаром, который закупается у двух разных поставщиков, можно создать запись с полями, определяющими код первого поставщика и код второго поставщика.

Что произойдет при добавлении третьего поставщика? Добавление третьего поля нежелательно, так как для этого нужно изменять программу и таблицу, поэтому данный способ плохо адаптируется к динамическому изменению числа поставщиков. Вместо этого можно поместить все сведения о поставщиках в отдельную таблицу Vendors (поставщики) и связать товары с поставщиками с помощью кодов товаров или поставщиков с товарами с помощью кодов поставщиков.

Вторая нормальная форма

Создайте отдельные таблицы для наборов значений, относящихся к нескольким записям.

Свяжите эти таблицы с помощью внешнего ключа.

Записи могут зависеть только от первичного ключа таблицы (составного ключа, если необходимо). Возьмем для примера адрес клиента в системе бухгалтерского учета. Этот адрес необходим не только таблице Customers, но и таблицам Orders, Shipping, Invoices, Accounts Receivable и Collections. Вместо того чтобы хранить адрес клиента как отдельный элемент в каждой из этих таблиц, храните его в одном месте: или в таблице Customers, или в отдельной таблице Addresses.

Третья нормальная форма

Устраните поля, не зависящие от ключа.

Значения, входящие в запись и не являющиеся частью ключа этой записи, не принадлежат таблице. Если содержимое группы полей может относиться более чем к одной записи в таблице, подумайте о том, не поместить ли эти поля в отдельную таблицу.

Например, в таблицу Employee Recruitment (наем сотрудников) можно включить адрес кандидата и название университета, в котором он получил образование. Однако для организации групповой почтовой рассылки необходим полный список университетов. Если сведения об университетах будут храниться в таблице Candidates, составить список университетов при отсутствии кандидатов не получится. Таким образом, создайте вместо этого отдельную таблицу Universities и свяжите ее с таблицей Candidates при помощи ключа — кода университета.

Исключение. Выполнять нормализацию баз данных до третьей нормальной формы теоретически желательно, но не всегда практично. Например, для устранения всех возможных зависимостей между полями таблицы Customers придется создать отдельные таблицы для хранения сведений о городах, почтовых индексах, торговых представителях, категориях клиентов и любых других сведений, которые могут дублироваться в нескольких записях. С теоретической точки зрения нормализация желательна. Однако значительное увеличение числа

маленьких таблиц может привести к снижению производительности СУБД или исчерпанию памяти и числа дескрипторов открытых файлов.

Выполнять нормализацию до третьей нормальной формы может быть целесообразно только для часто изменяемых данных. Если при этом сохранятся зависимые поля, спроектируйте приложение так, чтобы при изменении одного из этих полей пользователь должен был проверить все связанные поля.

Другие формы нормализации

Кроме описанных нормальных форм, существует четвертая нормальная форма, которую также называют нормальной формой Бойса-Кодда (BCNF), и пятая нормальная форма, но на практике они применяются редко. Несоблюдение этих правил может привести к ухудшению архитектуры базы данных, но на функциональности это сказаться не должно.

Задание: Выполните нормализацию таблицы с вымышленными данными о студентах.

Таблица до нормализации:

Студент	Преподаватель	Кафедра	Кабинет1	Кабинет2	Кабинет3
1022	Петров	412	101-07	143-01	159-02
4123	Иванов	216	201-01	211-02	214-01

Контрольные вопросы:

1. Перечислите и опишите формы нормализации какие вы знаете.
2. Дайте характеристику первой нормальной формы.
3. Дайте характеристику второй нормальной формы.
4. Дайте характеристику третьей нормальной формы.

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №7

Тема: «Разработка таблиц в Microsoft Access»

Цель: Получить практический опыт создания таблиц БД с помощью конструктора таблиц MS Access.

Время выполнения: 4 урока

Ход работы:

Рассмотрим этапы создания БД «Учебная часть» с помощью СУБД **Access** . Сначала составляем ERD для **базы данных** «Учебная часть». Этапы проектирования ERD изложены в Практической работе №5.

После создания модели запускаем приложение Access . Открывается окно приложение Access на странице Приступая к работе с Microsoft Access . В разделе Новая пустая база данных щелкаем на пиктограмме Новая база данных. В правой части окна появится информация об имени файла и указана директория для его хранения. По умолчанию имя файла **- База данных1**.accdb.

Изменить имя файла и путь к директории для хранения файла БД можно в окне "Файл новой базы данных" щелкнув на пиктограмме "Поиск расположения для размещения базы данных". Установив имя файла – Учебная_часть_.accdb и требуемое имя директории в окне "Файл новой базы данных", надо щелкнуть на кнопке ОК, окно закроется.

Далее необходимо щелкнуть на кнопке Создать, чтобы создать пустую базу данных. При создании новой пустой базы данных окно приложения Access открывается на контекстной вкладке "Режим таблицы". В окне отображается новая пустая таблица с именем Таблица 1 в режиме таблица, представленная на Рис. 1.

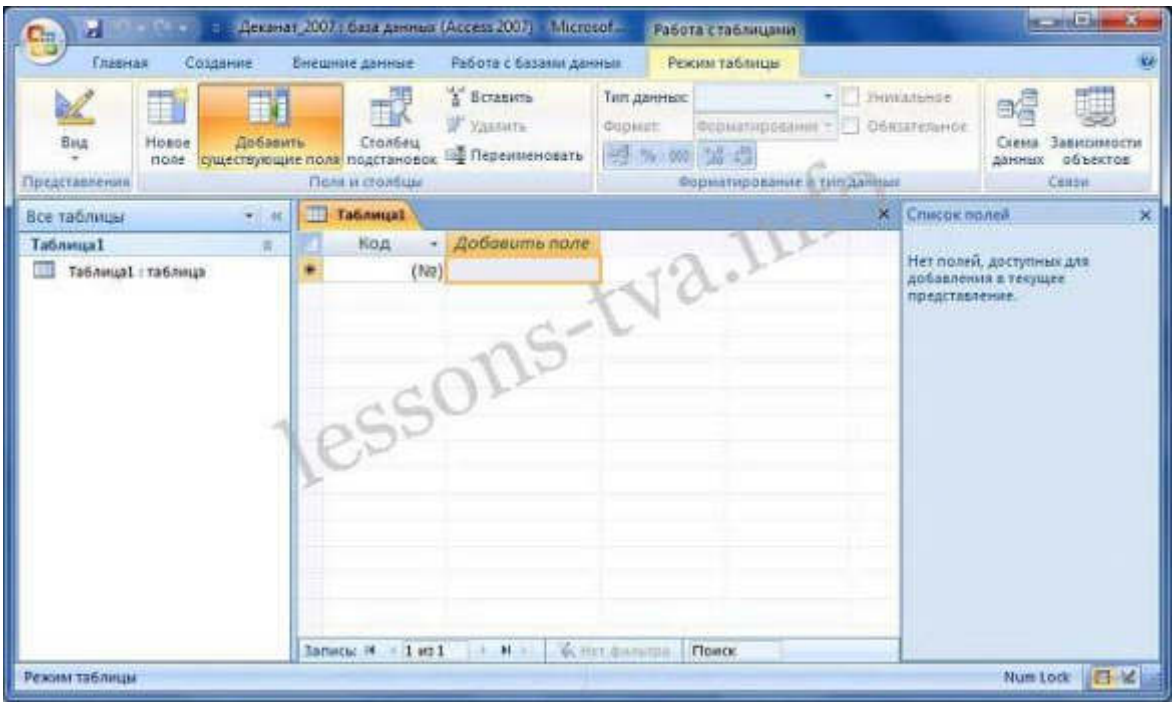


Рис. 1.

Далее необходимо перейти в режим Конструктор и создать структуру первой таблицы базы данных. Для этого необходимо щелкнуть на пиктограмме Вид и выбрать режим Конструктор.

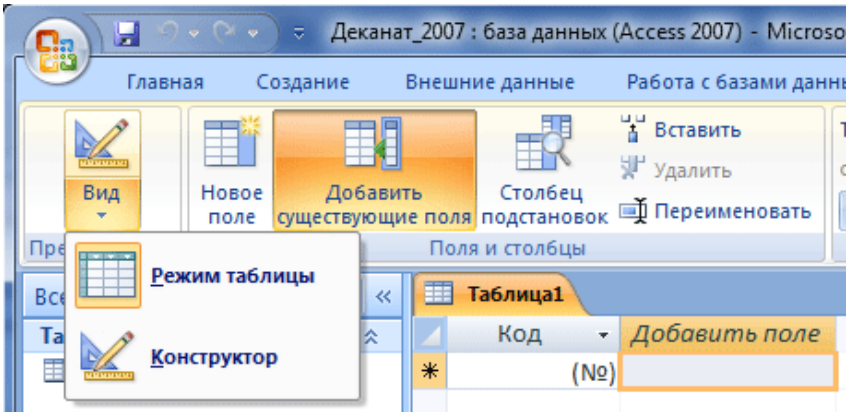


Рис. 2.

Откроется окно Сохранение, в котором надо указать имя Группы студентов и нажать кнопку ОК.

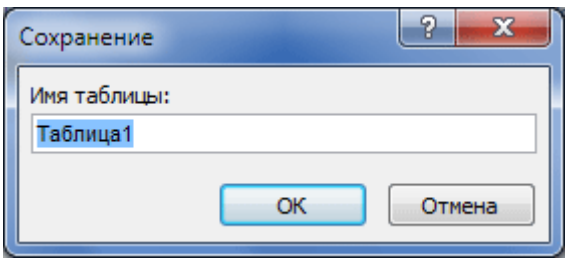


Рис. 3.

Откроется таблица Группы студентов в режиме Конструктор

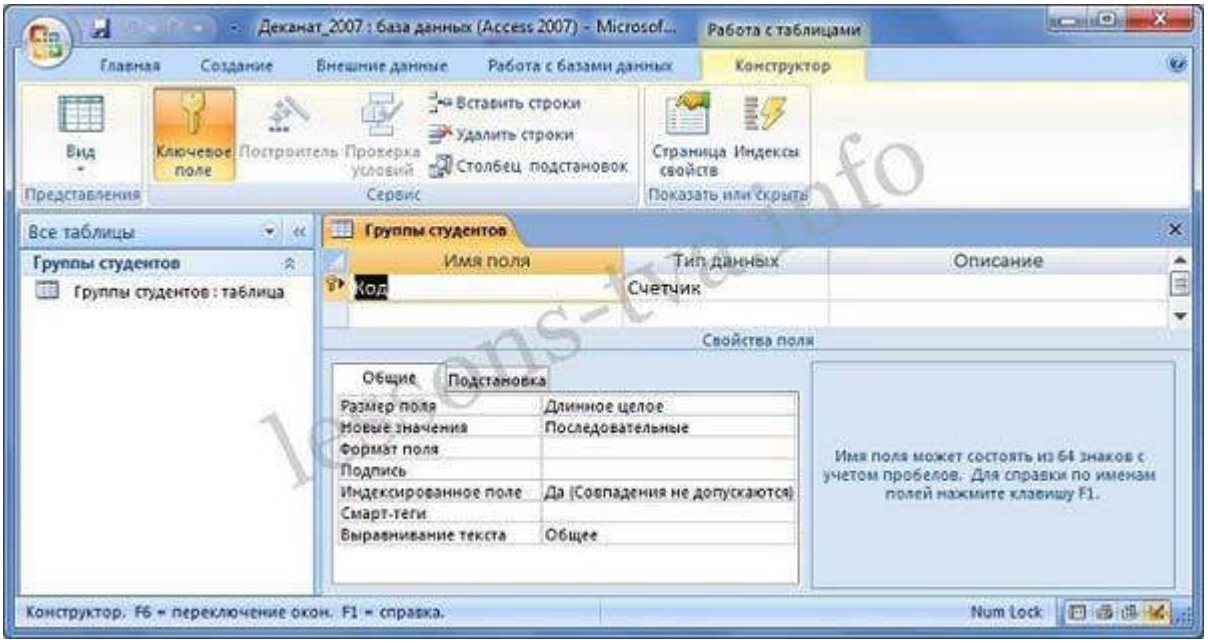


Рис. 4.

Создаем структуру таблицы Группы студентов. В первую строку колонки "Имя поля" вводим код группы студентов (КодГруппы) и нажимаем клавишу Enter. Курсор переместится в колонку Тип данных. Access по умолчанию назначает тип данных - Счетчик. Нажимаем клавишу Enter, при этом курсор переместится в колонку Описание, при необходимости вводим описание данных.

Первой строке таблицы (поле КодГруппы) Access по умолчанию назначает поле первичного ключа. Для первичного ключа в свойствах поля устанавливается значение Индексированного поля: **Да (Совпадения не допускаются)**. Далее заполняем вторую строку (второе поле таблицы), Имя поля - Название, Тип данных - текстовый. Третья строка: Имя поля - Курс, Тип данных - числовой и четвертая строка Имя поля - Семестр, Тип данных - числовой. При этом для имени поля "Название" в разделе свойства поля необходимо установить размер поля - 6.

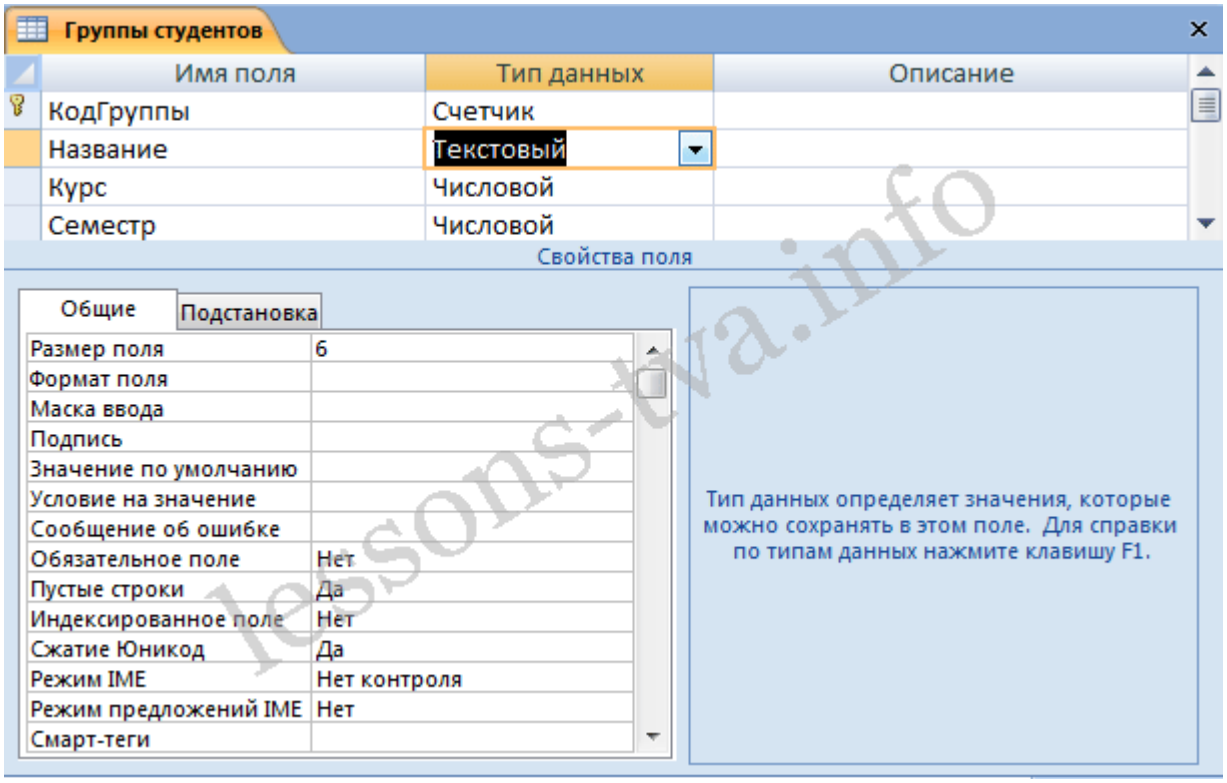


Рис. 5.

Затем создаем структуры остальных трех таблиц в соответствии с характеристиками таблиц-объектов Студенты, Дисциплины, Успеваемость. Обязательно соблюдайте указанную последовательность создания структуры таблиц.

Необходимо отметить, что в структуре таблицы "Студенты" для поля КодГруппы (вторичный ключ) установите значение Индексированного поля: **Да (Совпадения допускаются)** и тип данных - **мастер подстановок**. В структуре таблицы "Успеваемость" для поля КодСтуденты (вторичный ключ) и поля КодДисциплины (вторичный ключ) установите значение Индексированного поля: **Да (Совпадения допускаются)** и тип данных - **мастер подстановок**.

Структуры остальных таблиц: Студенты, Дисциплины, Успеваемость:

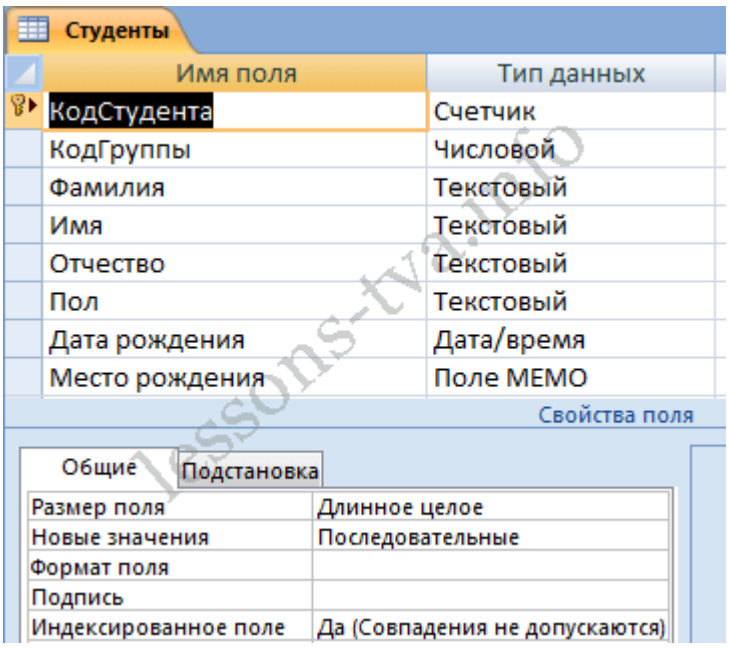


Рис. 6

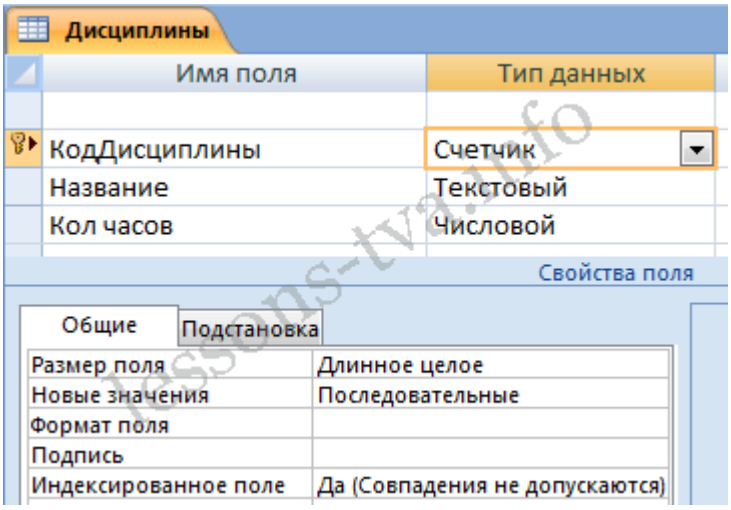


Рис. 7

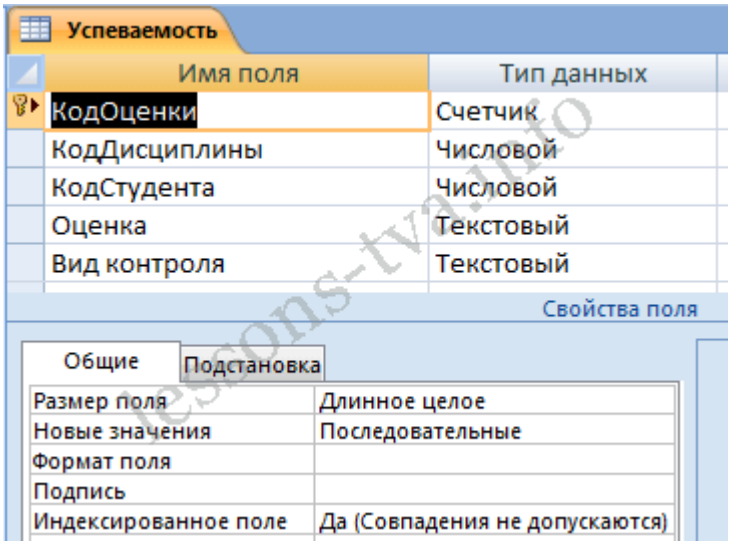


Рис. 8

Задание: Создать в MS Access базу данных «Справочник кадров», а в ней следующие таблицы:

Таблица: Справочник кадров

Имя поля	Тип	Размер
ТабНом (ключевое поле)	Текстовый	5
Фамилия	Текстовый	25
Имя	Текстовый	15
Отчество	Текстовый	15
ДатаРожд	Дата/время	Краткий формат даты
КодДок	Текстовый	2
Серия	Текстовый	7
Номер	Текстовый	6
ДатаВыд	Дата/время	Краткий формат даты
КемВыд	Текстовый	25
ИНН	Текстовый	12
Страх	Текстовый	20
КодСтраны	Текстовый	3
Регион	Текстовый	2
Индекс	Текстовый	6
Район	Текстовый	25
Город\ НасПункт	Текстовый	25
Улица	Текстовый	25
Дом	Текстовый	3
Корпус	Текстовый	2
Квартира	Текстовый	3
Пол	Текстовый	1
КодОтд	Текстовый	3
Катег	Текстовый	40
ГрупПерс	Текстовый	3
ДатаПриема	Дата/время	Краткий формат даты
Резидент	Числовой	Длинное целое
Совмест	Числовой	Длинное целое
РежимРаб	Числовой	Длинное целое
Ставка	Числовой	Двойное с плавающей
Коэф	Числовой	Двойное с плавающей
ЗаСтаж	Числовой	Двойное с плавающей
ЗаРасш	Числовой	Двойное с плавающей
ПерсДопл	Числовой	Двойное с плавающей
ЗаКласс	Числовой	Двойное с плавающей
ЗаНенорм	Числовой	Двойное с плавающей
ЗаСовм	Числовой	Длинное целое
Проф	Числовой	Длинное целое
Дети	Числовой	Длинное целое
Вычет	Числовой	Двойное с плавающей
ВычетДети	Числовой	Длинное целое
ЛьготаНал	Числовой	Длинное целое
Компенс	Числовой	Двойное с плавающей
Пособие	Числовой	Двойное с плавающей

%Алим	Числовой	Длинное целое
%Штраф	Числовой	Целое
СберБанк	Числовой	Длинное целое
Увольнение	Текстовый	1

Таблица: Аванс

Имя поля	**Тип**	**Размер**
ТабНом (Поле со	Текстовый	5
Аванс	Числовой	Двойное с плавающей
РазВыпл	Числовой	Двойное с плавающей

Таблица: Больничный

Имя поля	**Тип**	**Размер**
ТабНом (поле со	Текстовый	5
ТипБол	Текстовый	1
ДатаНач	Дата/время	Краткий формат даты
ДатаКон	Дата/время	Краткий формат даты
Дни1	Числовой	Длинное целое
Дни2	Числовой	Длинное целое
Процент	Числовой	Двойное с плавающей

Таблица: Доначисления

Имя поля	**Тип**	**Размер**
ТабНом (поле со	Текстовый	5
Оклад	Числовой	Двойное с плавающей
Премия	Числовой	Двойное с плавающей
Доплата	Числовой	Двойное с плавающей
Совмещения	Числовой	Двойное с плавающей
ЗаСтаж	Числовой	Двойное с плавающей
Бол	Числовой	Двойное с плавающей
Отпуск	Числовой	Двойное с плавающей
КомпУв	Числовой	Двойное с плавающей
Налог	Числовой	Двойное с плавающей

Таблица: Календарь

Имя поля	**Тип**	**Размер**
Месяц	Числовой	Двойное с плавающей
Год	Числовой	Двойное с плавающей
РабДни	Числовой	Двойное с плавающей
Суб	Числовой	Двойное с плавающей
КалендДни	Числовой	Двойное с плавающей
ПраздДни	Числовой	Двойное с плавающей
Часы	Числовой	Двойное с плавающей
НазвМес	Текстовый	8

Таблица: Отпускные

Имя поля	**Тип**	**Размер**
ТабНом (поле со	Текстовый	5
№Приказа	Текстовый	6
ДатаПрик	Дата/время	Краткий формат даты
НачОтп	Дата/время	Краткий формат даты
КонОтп	Дата/время	Краткий формат даты
ДниКомп	Числовой	Длинное целое

Таблица: Помощь

Имя поля	**Тип**	**Размер**
ТабНом (поле со	Текстовый	5
Помощь	Числовой	Двойное с плавающей
Обряды	Числовой	Двойное с плавающей

Таблица: Премия

Имя поля	**Тип**	**Размер**
ТабНом	Текстовый	5
ФИО	Текстовый	25
%Мес	Числовой	Двойное с плавающей
Разовая	Числовой	Длинное целое
Увольнение	Текстовый	1

Таблица: Табель

Имя поля	**Тип**	**Размер**
ТабНом	Текстовый	5
ФИО	Текстовый	25
Катег	Текстовый	25
Месяц	Числовой	Длинное целое
Год	Числовой	Длинное целое
ОтрабДни	Числовой	Длинное целое
Часы	Числовой	Длинное целое
Ночные	Числовой	Длинное целое
Праздничные	Числовой	Длинное целое
Увольнение	Текстовый	1

Контрольные вопросы:

1. С помощью какого инструмента создаются таблицы в MS Access?
2. Какие типы данных существую в MS Access?

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №8

Тема: «Разработка запросов в Microsoft Access»

Цель: Научиться создавать запросы любой сложности с помощью инструментов MS Access.

Время выполнения: 4 урока

Ход работы:

Знакомство с запросами

Запросы упрощают просмотр, добавление, удаление или изменение данных в базе данных Access. Среди других целей использования запросов можно отметить:

- быстрый поиск определенных данных путем фильтрации с применением определенных критериев (условий);
- вычисление или сведение данных;
- автоматизированное управление данными, например, регулярный просмотр актуальных данных.

Запросы как средство поиска данных и работы с ними

В хорошо структурированной базе данных сведения, которые требуется представить с использованием формы или отчета, зачастую хранятся в разных таблицах. Запрос может извлечь информацию из разных таблиц и собрать ее для отображения в виде формы или отчета. Запрос может представлять собой обращение к данным для получения информации из базы данных или выполнения действий с данными. Запрос можно использовать для получения ответа на простой вопрос, выполнения расчетов, объединения данных из разных таблиц, а также для добавления, изменения или удаления данных в таблице. Это очень гибкий инструмент: существует много типов запросов, и каждый тип создается с учетом задачи.

Основные типы запросов	Назначение

Запрос на выборку	Получение данных из таблицы и выполнение вычислений.
Запрос на изменение	Добавление, изменение или удаление данных. Для каждой задачи существует специальный тип запроса на изменение. Запросы на изменение недоступны в веб-приложениях Access.

Создание запроса на выборку

Запрос на выборку позволяет просматривать данные только из определенных полей таблицы либо из нескольких таблиц одновременно или же находить данные, которые соответствуют определенным условиям. Дополнительные сведения вы найдете в статье Создание простого запроса на выборку.

Просмотр данных из выбранных полей

Например, если база данных содержит таблицу с различной информацией о товарах, а необходимо просмотреть список товаров и их цены, запрос на выборку создается таким образом, чтобы возвратить только названия товаров и соответствующие цены.

Откройте базу данных и на вкладке Создание нажмите кнопку Конструктор запросов.

В диалоговом окне Добавление таблицы на вкладке Таблицы дважды щелкните таблицу Товары, затем закройте диалоговое окно.

Допустим, в таблице "Товары" содержатся поля "Наименование товара" и "Цена по прейскуранту". Дважды щелкните элементы Наименование товара и Цена по прейскуранту, чтобы добавить эти поля в бланк запроса.

На вкладке Конструктор нажмите кнопку Выполнить. Запрос будет выполнен, и отобразится список товаров и цен на них.

Одновременный просмотр данных из нескольких связанных таблиц

Рассмотрим пример базы данных магазина пищевых продуктов, в которой требуется просмотреть заказы, полученные от клиентов из определенного города. Допустим, данные о заказах и данные о клиентах хранятся в двух таблицах под названием "Заказчики" и "Заказы" соответственно. Если в каждой таблице имеется поле "Код заказчика", которое является основой отношение "один-ко-многим" между этими двумя таблицами, вы можете создать запрос, который возвратит сведения о заказах клиентов, живущих в определенном городе, например, в Тюмени, используя следующую процедуру.

1. Откройте базу данных. На вкладке Создание в группе Запросы нажмите кнопку Конструктор запросов.

2. В диалоговом окне Добавление таблицы на вкладке Таблицы дважды щелкните элементы Заказчики и Заказы.

3. Закройте диалоговое окно Добавление таблицы. Обратите внимание на линию (называемую соединением), которая соединяет поле "Код" в таблице "Заказчики" с полем "Код заказчика" в таблице "Заказы". Эта линия отображает связь между двумя таблицами.

4. В таблице "Клиенты" дважды щелкните элементы Организация и Город, чтобы добавить эти поля в бланк запроса.

5. В бланке запроса в столбце Город снимите флажок в строке Показать.

6. В строке Условие отбора столбца Город введите Тюмень.

7. Если снять флажок Показать, в результатах запроса не будет отображаться город, а слово Тюмень в строке Условие отбора означает, что требуется просмотреть только те записи, для которых в поле "Город" указано значение "Тюмень". В этом случае запрос возвращает данные только о тех клиентах, которые находятся в Тюмени. Для использования поля в условии отбора показывать его на экране не обязательно.

8. В таблице "Заказы" дважды щелкните элементы Код заказа и Дата размещения, чтобы добавить эти поля в два следующих столбца в бланке запроса.

9. На вкладке Конструктор в группе Результаты нажмите кнопку Выполнить. Происходит выполнение запроса и отображается список заказов клиентов из Тюмени.

10. Нажмите клавиши CTRL+S, чтобы сохранить запрос.

Создание запроса с параметрами

Если часто требуется выполнять варианты определенного запроса, можно использовать запрос с параметрами. При выполнении запроса с параметрами у пользователя запрашиваются значения полей, которые затем используются для создания условий для запроса.

В продолжение предыдущего примера, где было показано, как создавать запрос на выборку, который возвращает информацию о заказах для клиентов из Тюмени, можно изменить этот запрос таким образом, чтобы при каждом его запуске выводилось приглашение на ввод названия города. Откройте базу данных, созданную в предыдущем примере.

1. В области навигации щелкните правой кнопкой мыши запрос Заказы по городу (созданный в предыдущем разделе) и выберите в контекстном меню пункт Конструктор.

2. В бланке запроса в строке Условие отбора столбца "Город" удалите слово Тюмень и введите [Для какого города?].

Строка [Для какого города?] является предложением ввести параметр. Квадратные скобки показывают, что при выполнении запроса должно появиться предложение ввести данные, а текст (в данном случае Для какого города?) представляет собой вопрос, отображаемый в предложении.

1. Установите флажок в строке Показать столбца "Город", чтобы в результатах запроса отображался город.

2. На вкладке Конструктор в группе Результаты нажмите кнопку Выполнить. Запрос предложит ввести значение в строке "Город".

3. Введите слово Москва и нажмите клавишу ВВОД, чтобы увидеть заказы для клиентов в Москве. Но что делать, если значения, которые можно указать, неизвестны? В приглашении на ввод можно использовать подстановочные знаки.

4. На вкладке Главная в группе Представления нажмите кнопку Представление и выберите пункт Конструктор.

5. В бланке запроса в строке Условие отбора столбца Город введите Like [Для какого города?]&"*". В этом предложении ввести параметр ключевое слово Like, амперсанд (&) и звездочка (*), заключенная в кавычки, позволяют ввести сочетание знаков, включая подстановочные знаки, для получения разных результатов. Например, если пользователь вводит *, запрос возвращает все города; если пользователь вводит М, запрос возвращает все города, начинающиеся на букву "М"; если пользователь вводит *с*, запрос возвращает все города, в названиях которых имеется буква "с".

6. На вкладке Конструктор в группе Результаты нажмите кнопку Выполнить, в строке приглашения запроса введите Создать и нажмите клавишу ВВОД.

В результате выполнения запроса будет отображен список заказов от клиентов из Москвы.

Указание типов данных для параметра

Можно также указать, данные какого типа разрешается вводить в качестве значения параметра. Тип данных можно настроить для любого параметра, но особенно важно сделать это для числовых и денежных данных, а также данных о дате и времени. Когда для параметра указан тип данных, пользователи получают более понятные сообщения об ошибках в случае

ввода данных неправильного типа, например, ввода текста, когда ожидаются денежные данные.

Если параметр настроен таким образом, чтобы принимать текстовые данные, любое введенное значение интерпретируется как текст и сообщение об ошибке не отображается.

Чтобы указать тип данных для параметра в запросе, выполните процедуру, описанную ниже.

1. Когда запрос открыт в конструкторе, на вкладке Конструктор в группе Показать или скрыть нажмите кнопку Параметры.

2. В диалоговом окне Параметры запроса в столбце Параметр введите текст запроса на ввод значения для каждого параметра, для которого требуется указать тип данных. Убедитесь, что каждый из параметров соответствует запросу, который используется в строке Условие отбора в бланке запроса.

3. В столбце Тип данных выберите тип данных для каждого параметра.

Создание итогового запроса

Строка "Итог" в таблице очень удобна, но для ответа на более сложные вопросы используется запрос итоговых значений. Такой запрос представляет собой запрос на выборку, позволяющий группировать данные и составлять сводку данных, например, когда требуется просмотреть итоги продаж каждого товара. В запросе итоговых значений можно использовать статистическую функцию Sum для просмотра итогов продаж каждого товара.

Чтобы получить итоговые значения промежуточных сумм для товаров, можно следующим образом изменить запрос "Промежуточные суммы для товаров", созданный в предыдущем примере.

На вкладке Главная нажмите кнопку Режим и выберите Конструктор.

Запрос "Промежуточные суммы для товаров" будет открыт в конструкторе.

На вкладке Конструктор в группе Показать или скрыть нажмите кнопку Итоги.

В бланке запроса отобразится строка Итоги. Примечание: Несмотря на схожие названия, строка Итоги в бланке и строка Итог в таблице — не одно и то же.

С помощью строки Итоги в бланке можно группировать данные по значениям полей.

Строку Итог из таблицы можно добавить в результаты запроса итоговых значений.

При использовании строки Итоги в бланке необходимо выбрать статистическую функцию для каждого поля. Если выполнять вычисления с полем не требуется, можно сгруппировать данные по этому полю.

Во втором столбце бланка в строке Итог выберите в раскрывающемся списке вариант Sum.

На вкладке Конструктор в группе Результаты нажмите кнопку Выполнить. Происходит выполнение запроса, а затем отображается список товаров с промежуточными суммами.

Нажмите клавиши CTRL+S, чтобы сохранить запрос. Оставьте запрос открытым.

Выполнение расчетов на основе данных

Обычно таблицы не используются для хранения вычисляемых значений, например, промежуточных сумм, даже если они основаны на данных из одной базы данных. В некоторых случаях вычисленные значения могут устареть, поскольку данные, на основе которых они были рассчитаны, изменились. Например, не стоит хранить чей-либо возраст в таблице, поскольку придется обновлять это значение каждый год; вместо этого можно хранить дату рождения, а затем использовать запрос для расчета возраста.

Например, существует база данных с информацией о товарах, которые вы хотите продать. Она содержит таблицу под названием "Сведения о заказе", в которой находится информация о товарах, например, цена и количество каждого товара. Можно вычислить промежуточные суммы с помощью запроса, который умножает количество каждого товара на цену за единицу этого товара, количество каждого товара на цену за единицу этого товара и скидку этого товара, а затем вычитает общую скидку из общей цены. Если в предыдущем примере была создана база данных, откройте ее и выполните следующие действия.

1. На вкладке Создание нажмите кнопку Конструктор запросов.
2. В диалоговом окне Добавление таблицы на вкладке Таблицы дважды щелкните Сведения о заказе.
3. Закройте диалоговое окно Добавление таблицы.
4. В таблице "Сведения о заказе" дважды щелкните Код товара, чтобы добавить это поле в первый столбец бланка запроса.
5. Во втором столбце бланка щелкните правой кнопкой мыши строку Поле, а затем выберите в контекстном меню команду Область ввода.
6. В диалоговом окне Область ввода введите или вставьте следующее выражение: Промежуточный итог: ([Количество]*[Цена за единицу])-([Количество]*[Цена за единицу]*[Скидка])
7. Нажмите кнопку ОК.
8. На вкладке Конструктор нажмите кнопку Выполнить. Происходит выполнение запроса, а затем отображается список товаров с промежуточными суммами для каждого заказа.
9. Нажмите клавиши CTRL+S, чтобы сохранить запрос, и назовите его Промежуточные суммы для товаров.

Просмотр сводных данных и статистических показателей

При использовании таблиц для записи операций или хранения постоянно встречающихся числовых данных удобно иметь возможность

просмотреть статистические показатели для этих данных, например, суммарные или средние значения. В Access в таблица можно добавлять итоговую строку. Итоговая строка — это строка внизу таблицы, которая отображает итоговое или другое статистическое значение.

- Выполните запрос "Промежуточные суммы для товаров" и оставьте результаты открытыми в режим таблицы.
- На вкладке Главная щелкните Итоги. В нижней части таблицы появится новая строка со словом Итог в первом столбце.
- Щелкните ячейку в последней строке с именем Итог.
- Щелкните стрелку для просмотра доступных статистических функций. Поскольку в столбце содержатся текстовые данные, предлагается только два варианта: Нет и Количество.
- Выберите Количество. Содержимое ячейки изменится с Итог на число значений в столбце.
- Щелкните соседнюю ячейку (второй столбец). Обратите внимание на стрелку, которая появилась в ячейке.
- Щелкните стрелку и выберите Сумма. В поле будет отображаться сумма значений в столбце.
- Оставьте запрос открытым в режиме таблицы.

Создание перекрестного запроса

Пусть теперь необходимо просмотреть промежуточные суммы для товаров, а также статистические показатели по месяцам, т. е. в каждой строке должны отображаться промежуточные суммы для товара, а в каждом столбце — промежуточные суммы за месяц. Для вывода промежуточных сумм для товара и промежуточных сумм за месяц используйте перекрестный запрос.

Примечание: В веб-приложении Access перекрестный запрос не используется.

Вы можете снова изменить запрос "Промежуточные суммы для товаров", чтобы он возвращал строки промежуточных сумм для товаров и столбцы промежуточных сумм по месяцам.

- На вкладке Главная в группе Представления нажмите кнопку Представление и выберите пункт Конструктор.
- В группе Настройка запроса щелкните элемент Добавление таблицы.
- В диалоговом окне Добавление таблицы дважды щелкните элемент Заказы, а затем нажмите кнопку Закрыть.
- На вкладке Конструктор в группе Тип запроса щелкните элемент Перекрестная таблица. В бланке строка Показать скрыта, а отображается строка Перекрестная таблица.
- В третьем столбце бланка щелкните правой кнопкой мыши строку Поле , а затем выберите в контекстном меню пункт Область ввода. Откроется окно Область ввода.
- В диалоговом окне Область ввода введите или вставьте следующее выражение: Месяц: "Месяц" & DatePart("м", [Дата заказа])
- Нажмите кнопку ОК.
- В строке Перекрестная таблица выберите следующие значения в раскрывающемся списке: Заголовки строк для первого столбца, Значение для второго столбца и Заголовки столбцов для третьего.
- На вкладке Конструктор в группе Результаты нажмите кнопку Выполнить. Происходит выполнение запроса, а затем отображаются промежуточные суммы, собранные по месяцам.
- Нажмите клавиши CTRL+S, чтобы сохранить запрос.

Создание запроса на создание таблицы

Для создания новой таблицы на основе данных, которые хранятся в других таблицах, можно использовать запрос на создание таблицы.

Примечание: В веб-приложениях Access запросы на создание таблицы недоступны.

Например, пусть требуется отправить данные о заказах в Ростове партнеру из Ростова, который использует Access для подготовки отчетов. Вместо отправки всех данных о заказах можно отправить только те данные, которые относятся к заказам в Ростове.

Можно создать запрос на выборку, содержащий данные о заказах в Ростове, а затем использовать этот запрос для создания новой таблицы. Для этого используйте описанную ниже процедуру.

- Откройте базу данных из предыдущего примера.
- Для выполнения запроса на создание таблицы может потребоваться включить содержимое базы данных.
- Примечание: Если под лентой появится сообщение о включении базы данных, щелкните Включить содержимое. Если база данных уже находится в надежном расположении, строка сообщения не появится.
- На вкладке Создание в группе Запросы нажмите кнопку Конструктор запросов.
- В диалоговом окне Добавление таблицы дважды щелкните Сведения о заказе и Заказы и закройте диалоговое окно Добавление таблицы.
- В таблице Заказы дважды щелкните поля Код заказчика и Город получателя, чтобы добавить их в бланк.
- В таблице Сведения о заказе дважды щелкните элементы Код заказа, Код товара, Количество, Цена за единицу и Скидка, чтобы добавить эти поля в бланк.
- В столбце Город получателя бланка снимите флажок в строке Показать. В строке Условие отбора введите 'Ростов' (включая одинарные кавычки). Проверьте результаты выполнения запроса, прежде чем использовать их для создания таблицы.
- На вкладке Конструктор в группе Результаты нажмите кнопку Выполнить.

- Нажмите клавиши CTRL+S, чтобы сохранить запрос.
- В поле Имя запроса введите Запрос по заказам в Ростове и нажмите кнопку ОК.
- На вкладке Главная в группе Представления нажмите кнопку Представление и выберите пункт Конструктор.
- На вкладке Конструктор в группе Тип запроса нажмите кнопку Создание таблицы.
- В диалоговом окне Создание таблицы в поле Имя таблицы введите Заказы в Ростове и нажмите кнопку ОК.
- На вкладке Конструктор в группе Результаты нажмите кнопку Выполнить.
- В диалоговом окне подтверждения нажмите кнопку Да, и в области навигации отобразится новая таблица.

Создание запроса на добавление

Для извлечения данных из одной или нескольких таблиц и добавления их в другую таблицу можно использовать запрос на добавление.

Примечание: Запрос на добавление недоступен в веб-приложениях Access.

Предположим, вы создали таблицу для совместной работы с партнером из Ростова, но вспомнили, что этот партнер работает также с клиентами из Казани. Необходимо добавить в эту таблицу строки с данными по Казани. Используя следующую процедуру, можно добавить эти данные в таблицу "Заказы в Ростове".

- Откройте "Запрос по заказам в Ростове" в конструкторе.
- На вкладке Конструктор в группе Тип запроса выберите команду Добавить. Откроется диалоговое окно Добавление.
- В диалоговом окне Добавление щелкните стрелку в поле Имя таблицы и выберите Заказы в Ростове в раскрывающемся списке, а затем нажмите кнопку ОК.

- В бланке в строке Условие отбора столбца "Город получателя" удалите значение 'Ростов' и введите 'Казань'.
- В строке Добавление записей в таблицу выберите соответствующее поле для каждого столбца.
- В этом примере значения в строке Добавление записей в таблицу должны соответствовать значениям в строке Поле, но это не требуется для нормальной работы запросов на добавление.
- На вкладке Конструктор в группе Результаты нажмите кнопку Выполнить.

Примечание: При выполнении запроса, который возвращает большое количество данных, может появится сообщение об ошибке, в котором будет сказано, что отменить запрос не удастся. Попробуйте увеличить ограничение сегмента памяти до 3 МБ, чтобы выполнить запрос до конца.

Создание запроса на обновление

Для изменения данных в таблицах, а также для ввода условий, указывающих, какие строки следует обновить, можно использовать запрос на обновление. Запрос на обновление позволяет просмотреть обновленные данные перед выполнением обновления.

Важно: Запрос на изменение невозможно отменить. Возможно, перед обновлением следует создать резервные копии всех таблиц, которые будут обновлены запросом на обновление.

Примечание: Запрос на обновление недоступен в веб-приложениях Access.

В предыдущем примере вы добавили строки в таблицу "Заказы в Ростове". В ней в поле "Код товара" отображается числовой код товара. Чтобы данные отчетов были более информативными, замените коды товаров их названиям. Используйте следующую процедуру.

- Откройте таблицу "Заказы в Ростове" в конструкторе.

- В строке "Код товара" измените тип данных Числовой на Текстовый.
- Сохраните и закройте таблицу "Заказы в Ростове".
- На вкладке Создание в группе Запросы нажмите кнопку Конструктор запросов.
- В диалоговом окне Добавление таблицы дважды щелкните Заказы в Ростове и Товары и закройте диалоговое окно Добавление таблицы.
- На вкладке Конструктор в группе Тип запроса нажмите кнопку Обновить.
- В бланке больше не будут отображаться строки Сортировка и Показать и появится строка Обновление.
- В таблице Заказы в Ростове дважды щелкните элемент Код товара, чтобы добавить это поле в бланк.
- В бланке в строке Обновление столбца Код товара введите или вставьте следующую строку: [Товары].[Наименование]
- Совет: Запрос на обновление можно использовать для удаления значений полей; для этого используется пустая строка ("") или значение NULL в строке Обновление.
- В строке Условие отбора введите или вставьте следующую строку: [Код продукта] Like ([Товары].[Код])
- Можно узнать, какие значения будут изменены запросом на обновление, просмотрев запрос в режиме таблицы.
- На вкладке Конструктор выберите Режим >Режим таблицы. Запрос возвращает список кодов товаров, которые будут обновлены.
- На вкладке Конструктор нажмите кнопку Выполнить.
- При открытии таблицы "Заказы в Ростове" можно будет увидеть, что числовые значения в поле "Код товара" заменены наименованиями из таблицы "Товары". Читайте статью Создание запроса на обновление.

Создание запроса на удаление

Для удаления данных из таблиц, а также для ввода условий, указывающих, какие строки следует удалить, можно использовать запрос на удаление. Запрос на удаление позволяет просмотреть удаляемые строки перед выполнением удаления.

Примечание: Запросы на удаление недоступны в веб-приложениях Access.

Предположим, готовясь отправить таблицу "Заказы в Ростове" из предыдущего примера партнеру в Ростов, вы заметили, что некоторые строки содержат пустые поля. Перед отправкой таблицы их необходимо удалить. Можно открыть таблицу и удалить строки вручную, но если их много и есть четкие условия отбора, удобнее использовать запрос на удаление.

Вы можете использовать запрос для удаления из таблицы "Заказы в Ростове" строк, в которых отсутствует значение "Код заказа". Для этого выполните описанную ниже процедуру.

На вкладке Создание нажмите кнопку Конструктор запросов.

В диалоговом окне Добавление таблицы дважды щелкните Заказы в Ростове, затем закройте диалоговое окно Добавление таблицы.

На вкладке Конструктор в группе Тип запроса нажмите кнопку Удаление. В бланке больше не будут отображаться строки Сортировка и Показать и появится строка Удалить.

В таблице Заказы в Ростове дважды щелкните поле Код заказа , чтобы добавить его в бланк.

В бланке в строке Условие отбора в столбце "Код заказа" введите Is Null.

На вкладке Конструктор в группе Результаты нажмите кнопку Выполнить.

Задание: Создать следующие запросы для базы данных «Справочник кадров», созданной в предыдущей практической работе в MS Access:

- запрос на выборку
- запрос на выборку из нескольких связанных таблиц
- запрос с параметрами
- итоговый запрос
- перекрестный запрос
- запрос на создание таблицы
- запрос на добавление записи
- запрос на обновление записи
- запрос на удаление таблицы
- запрос на удаление записи

Контрольные вопросы:

1. С помощью каких инструментов создаются запросы в MS Access?
2. Какие типы запросов существую в MS Access?

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №9

Тема: «Разработка форм в Microsoft Access»

Цель: Получить практический опыт создания форм любой сложности средствами СУБД MS Access.

Время выполнения: 4 урока

Ход работы:

Создание формы в Access

В форме Access, как на витрине магазина, удобно просматривать и открывать нужные элементы. Так как форма — это объект, с помощью которого пользователи могут добавлять, редактировать и отображать данные, хранящиеся в базе данных классического приложения Access, ее внешний вид играет важную роль. Если база данных классического приложения Access используется несколькими пользователями, хорошо продуманные формы — залог точности данных и эффективности работы с ними.

Создать форму на компьютере в базе данных Access можно несколькими способами. В этой статье мы рассмотрим наиболее распространенные из них.

Создание формы из существующей таблицы или запроса в Access

Чтобы создать форму из таблицы или запроса базы данных, в области навигации щелкните таблицу или запрос, содержащие данные для формы, а затем на вкладке Создание нажмите кнопку Форма.

Приложение Access создаст форму и отобразит ее в режиме макета. При необходимости вы можете изменить оформление, например, настроить размер текстовых полей в соответствии с данными.

Создание пустой формы в Access

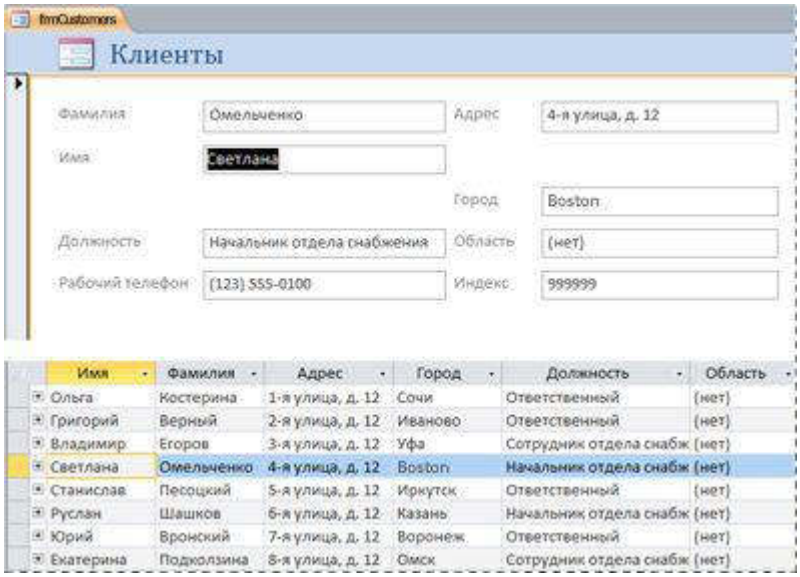
Чтобы создать форму без элементов управления и заранее отформатированных элементов, выполните следующее. На вкладке Создание

нажмите кнопку Пустая форма. Приложение Access откроет пустую форму в режиме разметки страницы и отобразит область Список полей.

- В области Список полей щелкните знак плюс (+) рядом с одной или несколькими таблицами, содержащими поля, которые нужно включить в форму.
- Чтобы добавить поле к форме, дважды щелкните его или перетащите на форму. Чтобы добавить сразу несколько полей, щелкните их по очереди, удерживая нажатой клавишу CTRL. Затем перетащите выбранные поля на форму.
- Примечание: Порядок таблиц в области Список полей может изменяться в зависимости от выбранной части формы. Если не удается добавить поле в форму, выберите другую часть формы и повторите попытку.
- С помощью инструментов в группе Элементы управления на вкладке Работа с макетами форм можно добавить в форму эмблему компании, заголовок, номера страниц, дату и время.
- Если требуется добавить в форму другие виды элементов управления, откройте вкладку Конструктор и выберите нужные элементы в группе Элементы управления.

Создание разделенной формы в Access

Разделенная форма позволяет одновременно отображать данные в двух представлениях: в представлении формы и в режиме таблицы. То есть вы можете использовать возможности обоих этих представлений в одной форме. Например, можно быстро найти запись в режиме таблицы, а затем просмотреть или изменить ее в представлении формы. Эти две части формы связаны с одним и тем же источником данных и всегда синхронизированы друг с другом.



Чтобы создать новую разделенную форму с помощью инструмента "Разделенная форма", в области навигации щелкните таблицу или запрос с нужными данными, а затем на вкладке Создание нажмите кнопку Другие формы и выберите пункт Разделенная форма.

Access создаст форму, которую можно изменять по своему усмотрению. Например, при необходимости можно настроить размер текстовых полей в соответствии с данными.

Создание формы, отображающей несколько записей, в Access

Форма нескольких элементов (также называется ленточной) удобна, если требуется отобразить несколько записей. Кроме того, ее легче настроить, чем таблицу. Такую форму можно создать с помощью инструмента "Несколько элементов".

В области навигации щелкните таблицу или запрос с данными, которые должны отображаться в форме.

На вкладке Создание нажмите кнопку Другие формы и выберите пункт Несколько элементов.

Приложение Access создаст форму и отобразит ее в режиме макета. В этом режиме вы можете вносить изменения в структуру формы и видеть, как изменяется представление данных. Например, можно настроить размер полей в соответствии с данными.

Создание формы, содержащей подчиненную форму, в Access

При работе со связанными данными, хранящимися в отдельных таблицах, часто приходится просматривать их из нескольких таблиц или запросов в одной и той же форме, поэтому подчиненные формы отлично подойдут в такой ситуации. В зависимости от потребностей подчиненные формы можно добавлять несколькими способами.

Создание формы навигации в Access

Форма навигации — это простая форма, содержащая элемент навигации. Формы навигации являются полезным дополнением любой базы данных, но особенно важны для баз данных, публикуемых в Интернете, так как область навигации Access не отображается в браузере.

Откройте базу данных, в которую нужно добавить форму навигации.

На вкладке Создание в группе Формы нажмите кнопку Навигация и выберите нужный стиль формы навигации.

Access создаст форму, добавит на нее элемент навигации и отобразит форму в режиме макета.

Задание: В режиме мастера или автоформы создать формы для ввода и редактирования данных для следующих таблиц, созданных в практической работе №7 в БД «Справочник кадров»: «Аванс», «Больничный», «Доначисления», «Календарь», «Отпускные», «Помощь». Для этого необходимо перейти на вкладку «Создание» - «Другие формы» - «Мастер форм».

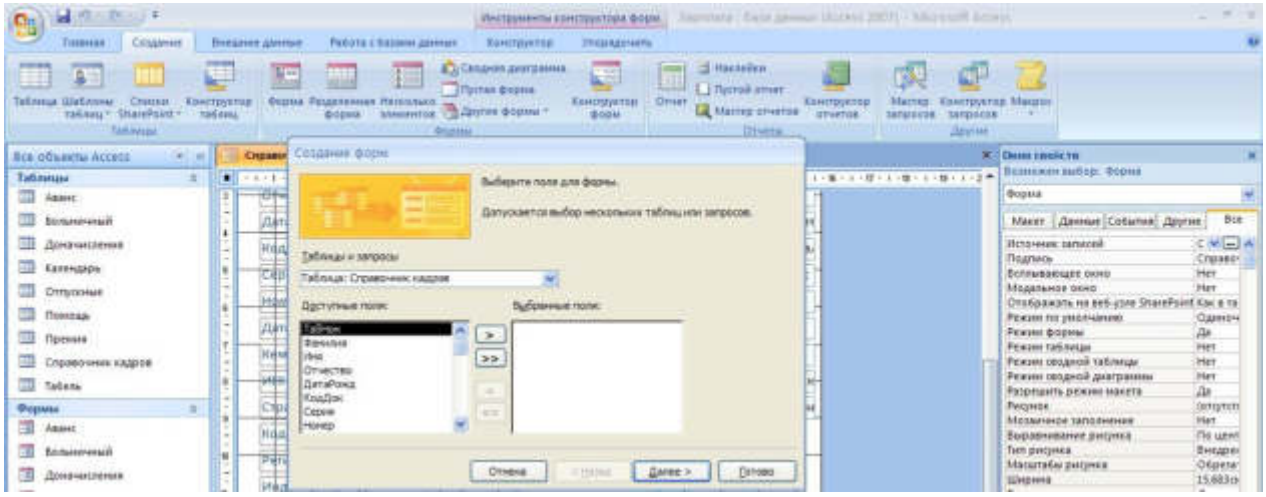


Рисунок 5 – Создание формы для редактирования данных

В форме СПРАВОЧНИК КАДРОВ сформировать вычисляемое поле Тарификация. Для этого открыть форму в режиме конструктор, добавить новое поле «Тарификация» и в свойствах поля на вкладке «Данные» ввести значение в строку данные: [Ставка]*[Коэф] + [Ставка]*[Коэф]*([ЗаРасшир] + [ЗаСтаж] + [ЗаКласс] + [ЗаНенорм] + [ЗаСовм])/100. При выполнении этой процедуры ОБЯЗАТЕЛЬНО использовать построитель выражений.

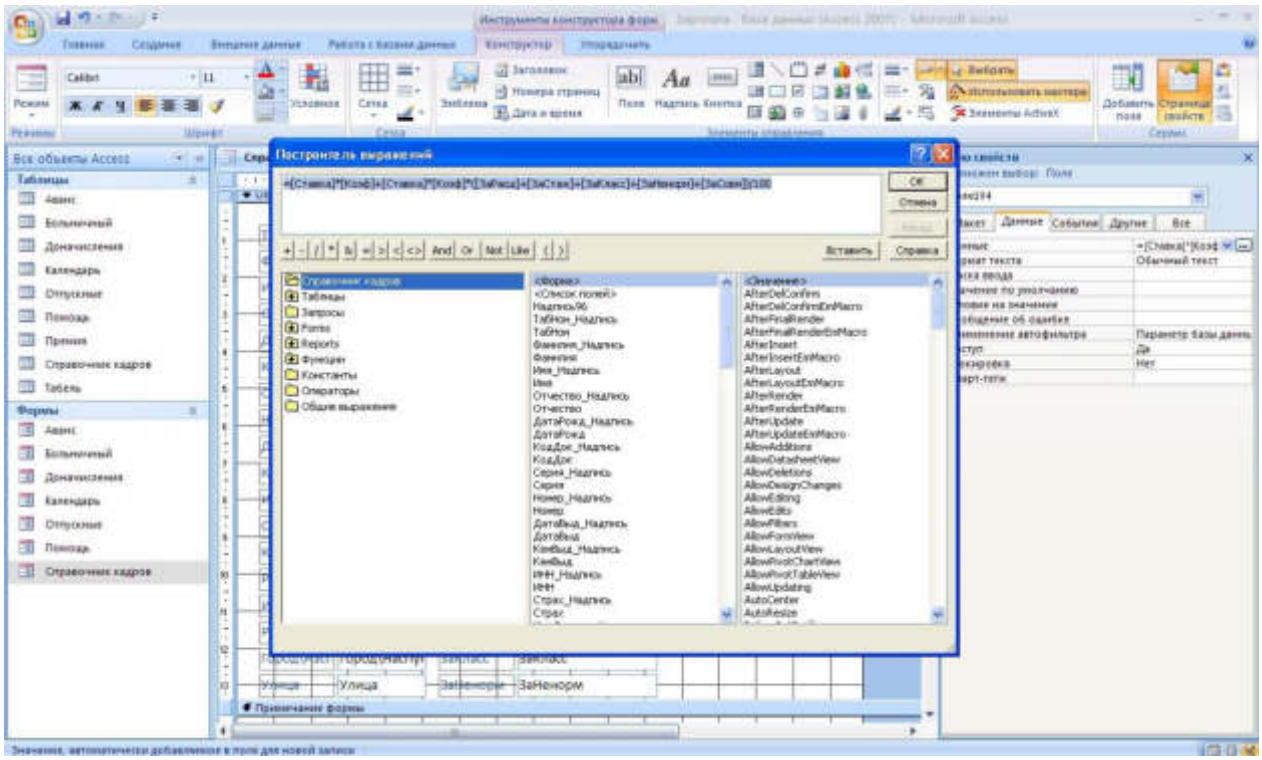


Рисунок 7 – Создание вычисляемого поля «Тарификация»

Это поле является контрольной суммой и является проверкой ввода данных:

- если в каком-то из образующих полей отсутствует значение (Is Null), то в результате тоже будет Is Null;
- если значение поле на порядок больше истинного значения, то неверно введены данные или не учтен процентный формат исходных полей.

Контрольные вопросы:

1. С помощью каких инструментов создаются формы в MS Access?
2. Какие типы форм существую в MS Access?

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №10

Тема: «Ввод и анализ данных с помощью форм»

Цель: Научится автоматизировать процесс ввода и анализа данных в СУБД MS Access с помощью форм.

Время выполнения: 4 урока

Ход работы:

В **Access** можно вводить данные непосредственно в таблицу в режиме таблица. Но обычно для ввода данных в БД Access используют **формы** (forms). Form ускоряет работу с базой данных. Form в БД - это структурированное интерактивное окно с элементами управления, в котором отображаются поля одной или нескольких таблиц, или запросов.

Форму можно использовать для ввода, изменения или отображения данных из таблицы или запроса. В Microsoft Office Access предусмотрены новые средства, помогающие быстро создавать forms, а также новые типы форм и функциональные возможности.

Формы в БД Access можно создавать с помощью различных средств:

- инструмента Form;
- инструмента Разделенная form;
- инструмента Несколько элементов;
- инструмента Пустая form;
- Мастера form;
- Конструктора form.

Все средства создания форм помещены в группу forms на вкладке Создание (рис. 1).

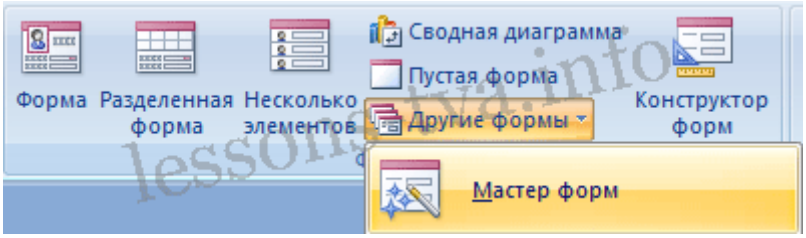


Рис. 1.

Forms, которые используют таблицы целесообразно выполнять с помощью Мастера или указанных инструментов, а дорабатывать их, т.е. вносить необходимые изменения, можно в режиме **макета** или **конструктора**. В Microsoft Access предусмотрено два режима внесения изменений и дополнений в формы: режим макета и режим конструктора. Переход между режимами (открыть, режим макета, конструктор) можно выполнить, щелкнув правой кнопкой мыши форму в области переходов, а затем выбрать нужный режим в контекстном меню.

Режим макета — это более наглядный режим редактирования (изменения) форм, чем режим конструктора. В режиме макета изменения выполняются фактически в реальной форме, поэтому в этом режиме целесообразно выполнять более простые изменения, связанные с ее внешним видом.

В тех случаях, когда в режиме макета невозможно выполнить изменения в форме, целесообразно применять режим конструктора. Режим конструктора предоставляет пользователю более широкие возможности для редактирования (изменения) форм, в этом режиме можно добавлять поля, настраиваемые элементы и составлять программы.

Инструмент "Форма". Для быстрого создания формы, т.е. создания одним щелчком мыши можно воспользоваться инструментом Form. В этом случае надо выделить таблицу в области объектов. Затем перейти на вкладку Создание и щелкнуть на пиктограмме Form. На экране будет отображена form (рис 2).

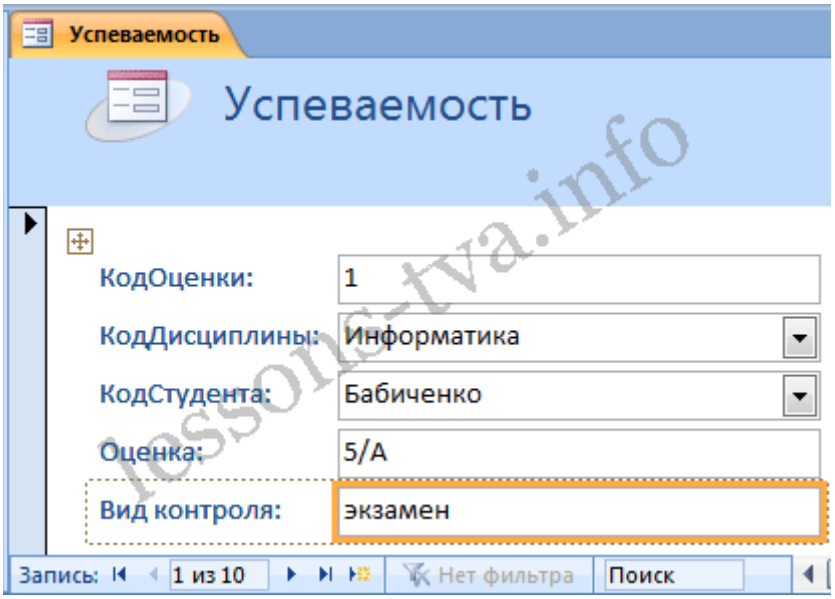


Рис. 2.

Если Access обнаруживает одну таблицу, связанную отношением «один-ко-многим» с таблицей или запросом, который использовался для создания формы, Access добавляет таблицу данных в форму, основанную на связанной таблице или запросе. Если таблица данных в форме не нужна, ее можно удалить.

Средство "Разделенная форма". Разделенная form — новая возможность в Microsoft Access , которая позволяет одновременно отображать данные в режиме формы и в режиме таблицы. В области объектов (переходов) выделить таблицу, например Успеваемость. Далее щелкнуть на пиктограмме "Разделенная forms" на вкладке Создать. На экране будет отображена form (3).

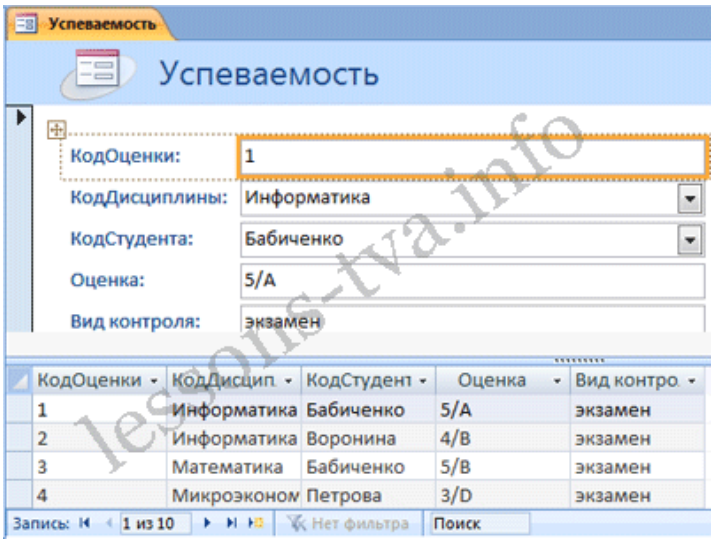


Рис. 3.

Инструмент "Несколько элементов". Форму, в которой отображается не одна, а одновременно несколько записей, можно создать инструментом "Несколько элементов" (рис. 4). Чтобы создать данную форму выделим в области объектов (переходов) одну из таблиц (например, Успеваемость). Затем перейдем на вкладку Создание и щелкнем на пиктограмме "Несколько элементов". На экране будет отображена form (рис. 4) в режиме макета.

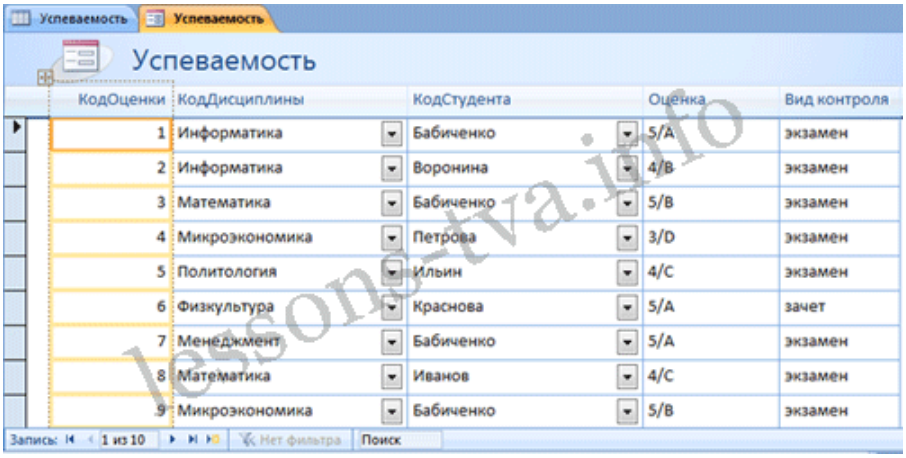


Рис. 4.

Form похожа на таблицу, в ней одновременно отображаются несколько записей. Но эта form предоставляет возможности для настройки, так как она отображается в режиме макета. В режиме макета можно легко осуществлять доработку формы (например, добавлять элементы управления и т.д.).

Средство Пустая форма. Этот инструмент можно использовать в том случае, если необходимо быстро создать форму с несколькими полями. Form открывается в режиме "Работа с макетами форм" и при этом отображается область Список полей (рис. 5).

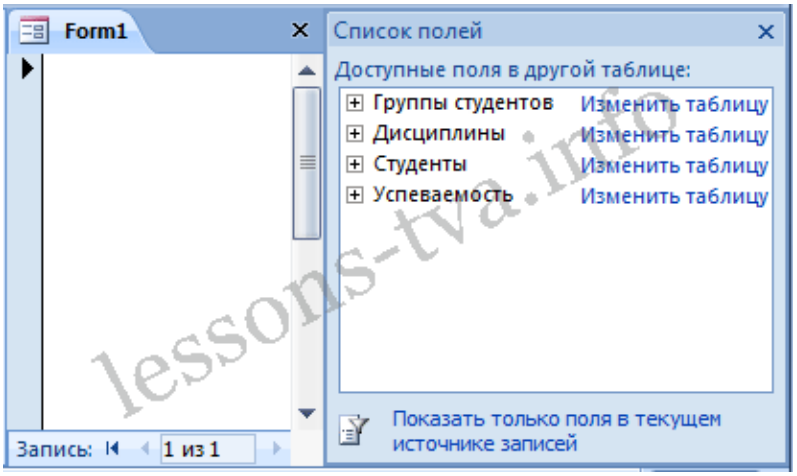


Рис. 5.

Мастер форм. Создание форм при помощи мастера форм осуществляется быстро, и это средство позволяет включить в форму поля из нескольких связанных таблиц или запросов. На вкладке Создание в группе Формы надо нажать кнопку Другие формы, а затем выбрать команду Мастер форм. Откроется окно диалога Создание форм, в котором необходимо отвечать на вопросы каждого текущего экрана Мастера и щелкать на кнопке Далее.

В первом окне необходимо выбрать поля из источника данных (таблиц или запросов). Для этого надо открыть список Таблицы и запросы, щелкнув на кнопку, справа. Например, выберем из списка таблицу Студенты.

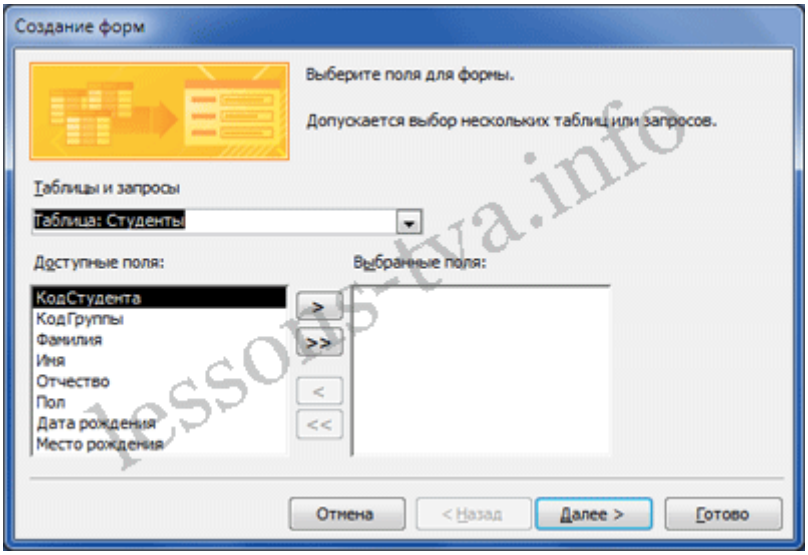


Рис. 6.

Затем все "Доступные поля" переведем в "Выбранные поля", выделив их и щелкнув на кнопку >>. Необходимо отметить, что, если form создается на основе нескольких таблиц, необходимо повторить действия для каждой таблицы – источника. Затем необходимо щелкнуть на кнопке Далее. В следующем окне надо выбрать внешний вид, например в один столбец и щелкнуть Далее. В следующем окне выберем требуемый стиль - официальный

После выбора стиля, требуется перейти в последнее окно, щелкнув на кнопке Далее. В последнем окне Мастера требуется ввести имя (например, Студенты мастер_форм) и указать дальнейшие действия: Открыть форму для просмотра и ввода данных; Изменить макет формы.

После ввода имени формы (например, Студенты), выбора режима: «Открыть форму для просмотра и ввода данных» и щелчка на кнопке Готово,

получим следующую форму для ввода и просмотра записей в таблицу Студенты.

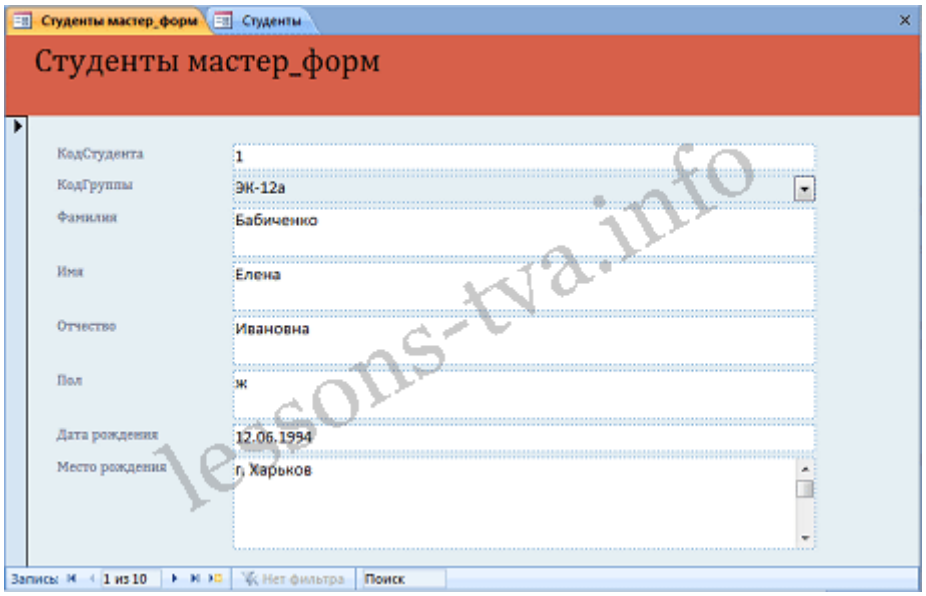


Рис. 7.

Конструктор форм. Для создания новой пустой формы Студенты необходимо выполнить следующее:

В окне приложения Access выбрать вкладку Создание. Выполнить щелчок на пиктограмме "Конструктор форм". В окне редактирования появится окно Form1 с пустой областью данных.

Для отображения списка полей требуемой таблицы выполнить щелчок на пиктограмме "Добавить существующие поля", появится список таблиц. Щелкнув на знак "+" таблицы (например, Студенты), откроется список необходимых полей (рис.7).

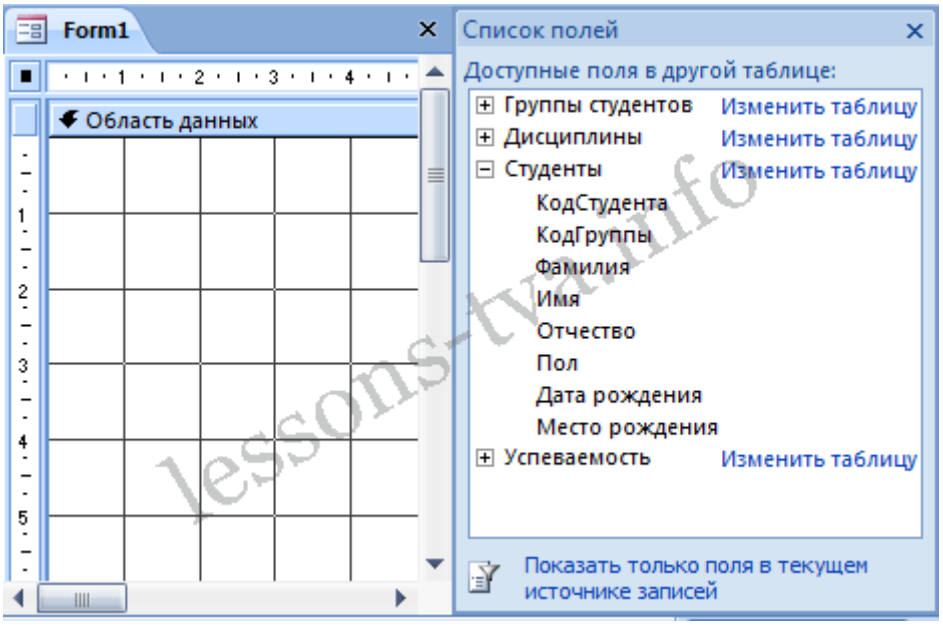


Рис. 8.

3. Поля из списка переместить на форму. Добавление полей осуществляется при нажатой левой кнопки мыши.

4. Поместить поля на форму (рис. 9).

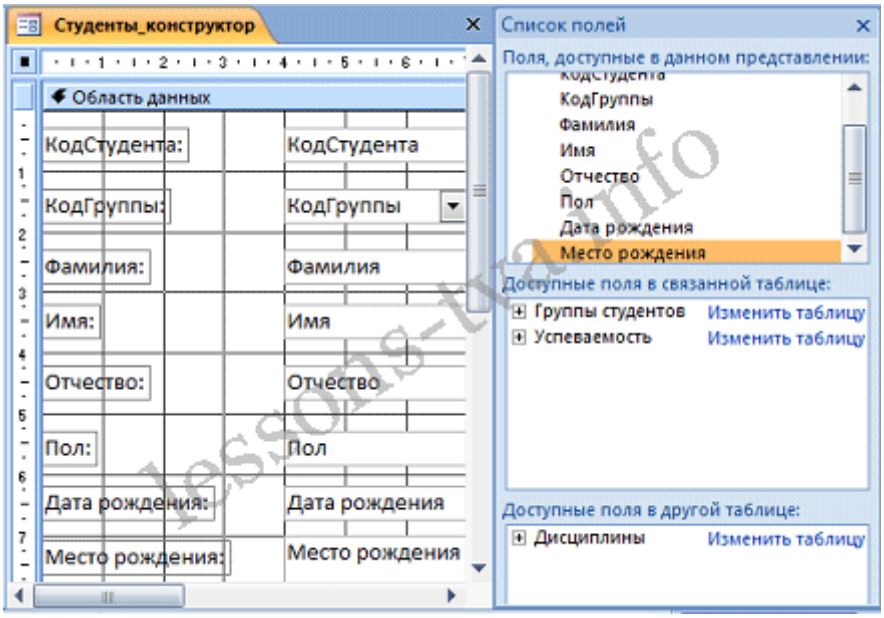


Рис. 9.

5. Перемещение полей и их имен по форме производиться следующим образом:

Выделить поле с именем щелчком мыши. Вокруг него появятся маркеры перемещения и изменения размеров. Перемещать поле можно вместе с привязанным к нему именем или отдельно от него.

Для перемещения поместить указатель мыши на квадратик, находящийся в левом верхнем углу элемента. Указатель мыши в виде четырех направленной стрелки позволяет перемещать объект.

Нажать кнопку мыши и, удерживая ее, буксировать поле или его имя в нужное место в форме. Затем отпустить кнопку мыши.

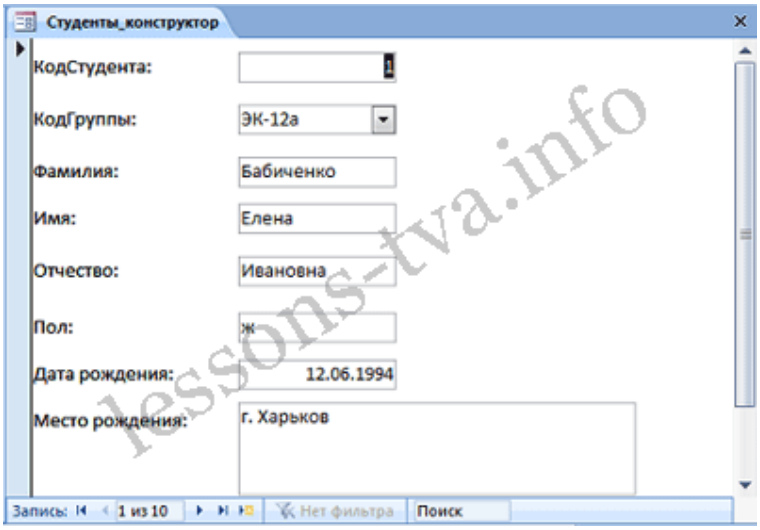
Для изменения надписи, связанной с полем необходимо выполнить на ней двойной щелчок мышью и выполнить необходимые изменения. Затем закрыть окно.

Для изменения размеров поместить курсор на размерные маркеры, при этом курсор примет вид двунаправленной стрелки. Нажать кнопку мыши, буксировать в нужном направлении, затем отпустить кнопку мыши.

Для удаления поля выделить его, нажать клавишу Delete или другим способом.

6. Сохранить форму.

7. Просмотреть форму Студенты_конструктор, выполнив на ней двойной щелчок в области переходов.



lessons-tva.info

Рис. 10.

Если вид формы не удовлетворяет, ее можно открыть в режиме Конструктор и внести необходимые изменения, затем сохранить.

Задание: Используя приведенные ниже данные, заполнить таблицы в БД «Справочник кадров» с помощью созданных форм.

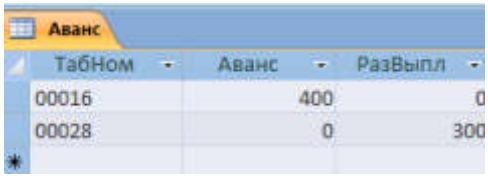
СПРАВОЧНИК КАДРОВ

ТабНо	Ф	И	Отчество	ДатаРо	КодД	Сер	Номер	ДатаВы	КемВыд	ИНН
00006	Баланчина	СВЕТЛАНА	СЕМЕНОВНА	08.06.19	10	07	748379	15.07.20	ОВД Пром. р. г.	2609013419
00007	Мануйлова	ЕЛЕНА	АЛЕКСЕЕВНА	17.07.19	10	10	579558	15.07.20	ОВД Пром. р. г.	2609013422
00008	Суслов	Александр	Александрович	21.03.19	10	09	618250	15.07.20	ОВД Кисловодс. РИК	2635058204
00009	Боженов	Юрий	Петрович	23.01.19	10	07	693731	15.07.20	ОВД Пром. р. г.	2607025648
00010	Гуляева	Любовь	Григорьевна	27.06.19	10	07	509424	15.07.20	ОВД Пром. р. г.	2635039017
00014	Гонтарь	ВАЛЕРИЙ	НИКОЛАЕВИЧ	03.04.19	10	08	727101	15.07.20	ОВД Пром. р. г.	2636020542
00015	Володина	Ирина	Алексеевна	17.12.19	10	06	590940	15.07.20	ОВД Пром. р. г.	2628006782
00016	Гаранин	Олег	Михайлович	23.11.19	10	07	553998	15.07.20	ОВД Пром. р. г.	2635026266
00017	Жеймо	ВИКТОР	МИХАЙЛОВИЧ	16.01.19	10	07	601939	15.07.20	ОВД Пром. р. г.	2609013420
00018	Гаркуша	ГРИГОРИЙ	ИВАНОВИЧ	02.07.19	10	07	637783	15.07.20	ОВД Пром. р. г.	2635026266
00019	Иваницкая	ЕВГЕНИЯ	ВЛАДИМИРОВНА	08.09.19	10	07	390840	15.07.20	ОВД Пром. р. г.	2634031706
00020	Мелехов	НИКОЛАЙ	АФАНАСЬЕВИЧ	05.05.19	10	07	650707	15.07.20	ОВД Пром. р. г.	2624021143
00027	Ивов	ВЛАДИМИР	ЮРЬЕВИЧ	26.04.19	10	07	686308	15.07.20	ОВД Пром. р. г.	2635104626
00028	Лисянский	ВИКТОР	МИХАЙЛОВИЧ	09.11.19	10	07	011266	15.07.20	ОВД Пром. р. г.	2631008067
00029	Королева	ГАЛИНА	ИВАНОВНА	01.01.19	10	07	631156	15.07.20	ОВД Пром. р. г.	2628033301
00030	Миртов	НИКОЛАЙ	МИХАЙЛОВИЧ	12.01.19	10	11	589256	15.07.20	ОВД Пром. р. г.	2609012876
00031	Корнилова	ЮЛИЯ	АЛЕКСАНДРОВНА	19.10.19	10	07	011266	15.07.20	ОВД Пром. р. г.	2624046991
00032	Кудряшов	ЮРИЙ	НИКОЛАЕВИЧ	15.08.19	10	10	550119	15.07.20	ОВД Пром. р. г.	2617009642
00034	Крамова	КЛАВДИЯ	АЛЕКСАНДРОВНА	04.09.19	10	07	390845	05.06.20	ОВД г.Буденновска	2636014094
00038	Лиманская	ВИКТОРИЯ	ВИКТОРОВНА	11.04.19	10	07	390845	05.06.20	ОВД г.Буденновска	2624046991
00061	Семеренко.	НИКОЛАЙ	ИВАНОВИЧ	29.09.19	10	06	579819	05.06.20	ОВД г.Буденновска	2630037602
00062	Самарина	ГАЛИНА	ИВАНОВНА	22.08.19	10	07	728148	05.06.20	ОВД г.Буденновска	2635058689
00063	Сорин	Анатолий	Викторович	02.01.19	10	07	608298	05.06.20	ОВД г.Буденновска	2608032058
00075	Грибова	ИРИНА	СЕРГЕЕВНА	23.03.19	10	07	074384	05.06.20	ОВД г.Буденновска	2631007430
00100	Иванов	АНДРЕЙ	НИКОЛАЕВИЧ	11.03.19	10	07	612355	05.06.20	ОВД г.Буденновска	2624010361
00079	Череваненк	Максим	Алексеевич	14.10.19	10	07	556197	05.06.20	ОВД г.Буденновска	2631007430
00082	Ряднов	ЮРИЙ	АЛЕКСАНДРОВИЧ	23.11.19	10	06	585713	05.06.20	ОВД г.Буденновска	2628013747
00083	Котов	АНАТОЛИЙ	ГРИГОРЬЕВИЧ	12.01.19	10	07	718915	05.06.20	ОВД г.Буденновска	2628014447

ТабНо	Стр	КодСтра	Реги	ИНДЕК	Рай	Город\ нас. пункт	Ули	Д	Корп	Кварти	П	Кодо
00006	005-170-941 11	643	26	357830	КИРОВСКИЙ Р-Н	НОВОПАВЛОВСК Г	МИРА УЛ	97				м
00007	005-170-943 13	643	26	357830	КИРОВСКИЙ Р-Н	НОВОПАВЛОВСК Г	КИРОВА УЛ	37	10	34		м
00008	005-170-949 19	643	26	355037		СТАВРОПОЛЬ Г	ФРОЛЕНКО УЛ	20		76		м
00009	005-170-996 26	643	26	356100	ИЗОБИЛЬН. Р-Н	ИЗОБИЛЬНЫЙ Г	ЮЖНАЯ УЛ	18				м
00010	039-181-898 86	643	26	355003		СТАВРОПОЛЬ Г	ЮЖНАЯ	23		14		ф
00014	005-170-950 12	643	26	355001		СТАВРОПОЛЬ Г	ОКТЯБРЬСКАЯ	18	2	28		м
00015	051-254-753 33	643	26	357700		КИСЛОВОДСК Г	ПОБЕДЫ ПР-КТ	15		191		ф
00016	005-170-952 14	643	26	355003		СТАВРОПОЛЬ Г	МОРОЗОВА УЛ	10		18		м
00017	011-814-665 18	643	26	357830	КИРОВСКИЙ Р-Н	НОВОПАВЛОВСК Г	ОКТЯБРЬСКАЯ	10				ф
00018	005-170-953 15	643	26	355029		СТАВРОПОЛЬ Г	МИРА УЛ	45		57		м
00019	007-036-738 22	643	26	355000		СТАВРОПОЛЬ Г	БАУМАНА УЛ	19				ф
00020	012-950-218 16	643	26	357920		БУДЕННОВСК Г	ЧЕХОВА ПР-КТ	16	А	1		ф
00027	070-815-316 45	643	26	355003		СТАВРОПОЛЬ	ПИРОГОВА	26	2	166		ф
00028	005-170-991 21	643	26	357030		НЕВИННОМЫССК Г	ГАГАРИНА УЛ	68		53		м
00029	107-089-741 63	643	26	356500	ПЕТРОВСК. Р-Н	СВЕТЛОГРАД Г	МАТРОСОВА УЛ	18	2	11		ф
00030	016-638-240 46	643	26	355000		СТАВРОПОЛЬ Г	ДОВАТОРЦЕВ	31		2		ф
00031	072-435-462 54	643	26	357830	КИРОВСКИЙ Р-Н	НОВОПАВЛОВСК Г	ЮЖНАЯ	58				ф
00032	005-170-984 22	643	26	357830	КИРОВСКИЙ Р-Н	НОВОПАВЛОВСК Г	50 ЛЕТ ВЛКСМ	51	3	116		м
00034	005-170-964 18	643	26	355037		СТАВРОПОЛЬ Г	РУСТАВЕЛИ УЛ	34	А	55		м
00038	067-441-072 60	643	26	356100	ИЗОБИЛЬН. Р-Н	ИЗОБИЛЬНЫЙ Г	50 ЛЕТ ВЛКСМ	71	1	60		ф
00061	023-407-747 24	643	26	355003		СТАВРОПОЛЬ Г	БЕЛИНСКОГО	31				ф
00062	003-981-807 49	643	26	355001		СТАВРОПОЛЬ Г	ФОМЕНКО	10		40		ф
00063	005-170-986 24	643	26	357700		КИСЛОВОДСК Г	ПАРКОВАЯ	3		2		м
00075	093-136-002 39	643	26	355003		СТАВРОПОЛЬ Г	ЮБИЛЕЙНАЯ	13				ф
00100	005-170-981 19	643	26	357830	КИРОВСКИЙ Р-Н	НОВОПАВЛОВСК Г	ШИРОКАЯ	28		32		м
00079	083-799-002 08	643	26	355029		СТАВРОПОЛЬ Г	ОКТЯБРЬСКАЯ	14				ф
00082	003-242-142 74	643	26	355000		СТАВРОПОЛЬ Г	ПОЛЕВАЯ УЛ	32				ф
00083	021-327-595 16	643	26	357920		БУДЕННОВСК Г	КОРОЛЕНКО УЛ	20		11		м

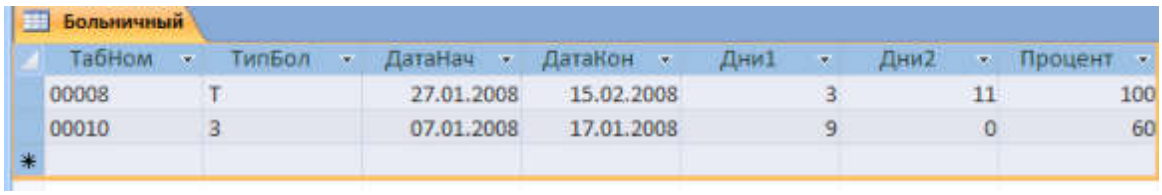
ТабНо	Кат	ГрупПе	ЗаСтаж	ДатаПрие	Резеде	Совмес	Режим	Ставк	Ко	ЗаРасш	ПерсДо	ЗаКла	ЗаНено
00006	первый зам. Управляющего	ШТ	20	17.10.199	1	0	5	8000	1	0	0	0	0
00007	главный экономист	АГК	15	16.05.199	1	0	5	3500	1	20	1500	0	25
00008	Специалист 1 категории	АГК	20	08.09.199	1	0	5	3000	1	10	375	0	0
00009	Ведущий специалист	шт	10	17.10.199	1	0	5	3100	1	10	450	0	0
00010	зам. Управляющего	ФБ	20	01.02.199	1	0	5	7500	1	20	600	0	15
00014	начальник сектора БИЗИ	АГК	10	23.06.199	1	0	5	3200	1	10	450	0	0
00015	гл.бухгалтер	ФБ	20	22.05.199	1	0	5	4000	1	10	900	0	30
00016	Сторож	МОП	0	14.05.199	1	0	5	3500	1	0	0	0	0
00017	ст.экономист по валюте	ФБ	20	15.06.199	1	0	5	3800	1	20	600	0	0
00018	Сторож	МОП	0	17.10.199	1	0	5	3000	1	0	0	0	0
00019	Специалист 1 категории	ФБ	15	14.02.199	1	0	5	3100	1	10	375	0	0
00020	нач. сектора фин.- хоз. деят-ти	ФБ	15	14.02.199	1	0	5	3700	1	20	600	0	35
00027	Специалист 1 категории	ФБ	0	25.10.199	1	0	5	3500	1	20	0	0	0
00028	Водитель	МОП	0	05.01.199	2	0	6	3150	1	0	0	15	0
00029	машинистка	ФБ	20	23.08.199	1	0	5	4000	1	20	0	0	0
00030	Начальник отдела	ФБ	15	15.03.199	1	0	5	5000	1	15	675	0	0
00031	Ведущий специалист	ФБ	10	01.02.199	1	0	5	3860	1	10	450	0	0
00032	инженер	шт	0	23.10.199	1	0	5	3650	1	20	0	0	0
00034	гл. экономист	шт	15	23.05.199	1	0	5	8300	1	20	600	0	0
00038	старший инспектор	ФБ	15	05.10.199	1	0	6	5875	1	20	600	0	15
00061	Специалист 1 категории	ФБ	15	10.02.199	1	0	0	3800	1	0	0	0	0
00062	Главный специалист	ФБ	15	18.06.199	1	0	5	4800	1	15	487,5	0	0
00063	Специалист 1 категории	шт	30	21.06.199	1	0	5	3900	1	0	0	0	0
00075	экономист по цен. Бумагам	шт	15	08.04.199	1	0	5	4500	1	10	375	0	0
00100	Сторож	МОП	0	24.09.199	1	0	5	3500	1	0	0	0	0
00079	Специалист 1 категории	ФБ	15	14.06.199	1	0	5	3750	1	10	375	0	25
00082	Вед. инспектор по работе с перс.	ФБ	15	27.02.199	1	0	5	5300	1	20	487,5	0	0
00083	Главный специалист	ФБ	10	18.10.199	1	0	5	6000	1	10	250	0	25

ТабН	ЗаСовм	Проф	Дети	Выч	Вычет_дет	ЛьготаНал	Компе	Пособи	%Али	%Штра	СберБа	увольнен
0000	0	1	0	1	1	0	0	0	0	0	0	
0000	0	1	2	1	1	0	0	0	0	0	0	
0000	0	1	2	1	1	0	0	0	0	0	0	
0000	0	1	0	1	1	0	0	0	0	0	0	
0001	0	1	1	1	1	0	0	0	0	0	0	
0001	0	1	2	1	1	0	0	0	0	0	0	
0001	0	1	1	1	1	0	0	0	0	0	0	
0001	0	1	0	1	1	0	0	0	0	0	0	у
0001	30	1	0	1	1	0	0	0	0	0	0	
0001	0	0	0	7,5	1	0	0	0	0	0	0	
0001	25	1	0	1	1	0	0	0	0	0	0	
0002	0	1	1	1	1	0	0	0	0	0	0	
0002	20	1	0	1	1	0	0	0	0	0	1	
0002	0	1	0	1	1	0	0	0	0	0	0	
0002	0	1	0	1	1	0	0	0	0	0	0	
0003	0	1	1	1	1	0	0	0	0	0	0	
0003	20	1	2	1	1	0	0	0	0	0	0	
0003	0	1	0	1	1	0	0	0	0	0	0	
0003	0	1	2	1	2	0	0	0	0	10	0	
0003	0	1	2	1	1	0	0	0	0	0	0	
0006	0	1	1	1	1	0	0	166,98	0	0	1	
0006	0	1	0	1	1	1	0	0	0	24	0	
0006	30	1	2	1	1	0	0	0	0	0	0	
0007	0	1	0	1	1	0	0	0	0	0	0	
0010	0	1	0	1	1	0	0	0	0	0	0	
0007	0	1	1	1	1	0	0	0	0	0	0	
0008	35	1	0	1	1	0	0	0	0	0	1	
0008	0	1	0	1	1	0	0	0	0	0	1	



Аванс

ТабНом	Аванс	РазВыпл
00016	400	0
00028	0	300



Больничный

ТабНом	ТипБол	ДатаНач	ДатаКон	Дни1	Дни2	Процент
00008	Т	27.01.2008	15.02.2008	3	11	100
00010	З	07.01.2008	17.01.2008	9	0	60

При заполнении поля ТипБол воспользуемся следующими условными обозначениями

З – заболевание. 60% (до 3 лет стаж), 80%(3-5 лет), 100%(Свыше 5 лет).

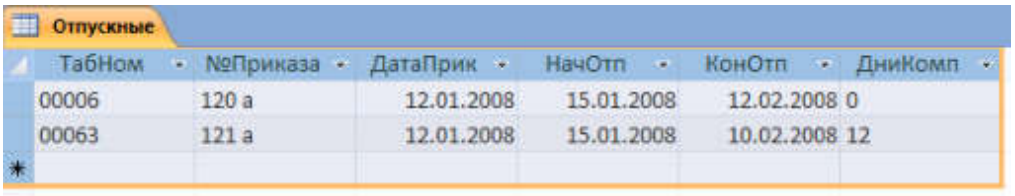
Т – производственная травма. Оплата 100% из фонда заработной платы (все страховые начисления).

Д – декретный отпуск. 100%

Б – бытовая травма. Первые 5 календарных дней не оплачиваются, потом как З.

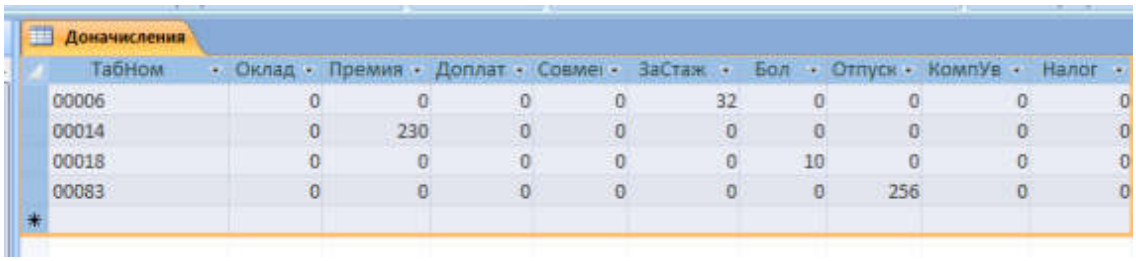
У – уход за ребенком. Первые 7 календарных дней по установленному % оплаты, а далее - 50%.

О – уход за ребенком для матери-одиночки. Первые 10 календарных дней оплачиваются по установленному % оплаты, а затем по 50%.



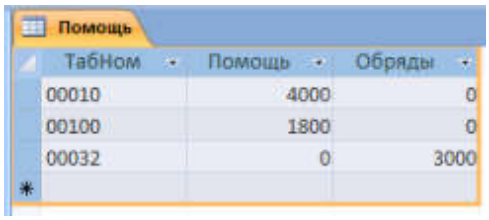
Отпускные

ТабНом	№Приказа	ДатаПрик	НачОтп	КонОтп	ДниКомп
00006	120 а	12.01.2008	15.01.2008	12.02.2008	0
00063	121 а	12.01.2008	15.01.2008	10.02.2008	12



Доначисления

ТабНом	Оклад	Премия	Доплат	Совмес	ЗаСтаж	Бол	Отпуск	КомпУв	Налог
00006	0	0	0	0	32	0	0	0	0
00014	0	230	0	0	0	0	0	0	0
00018	0	0	0	0	0	10	0	0	0
00083	0	0	0	0	0	0	256	0	0



Помощь

ТабНом	Помощь	Обряды
00010	4000	0
00100	1800	0
00032	0	3000

Календарь

Месяц	Год	РабДни	Суб	КаленДни	ПраздДни	Часы	ИмяМес
1	2008	19	4	31	5	151	Январь
2	2008	19	4	28	1	151	Февраль
3	2008	22	5	31	1	175	Март
4	2008	21	4	30	0	168	Апрель
5	2008	21	5	31	2	167	Май
6	2008	22	4	30	1	174	Июнь
7	2008	21	4	31	0	168	Июль
8	2008	23	4	31	0	184	Август
9	2008	22	4	30	0	176	Сентябрь
10	2008	21	5	31	0	168	Октябрь
11	2008	21	4	30	1	167	Ноябрь
12	2008	22	5	31	0	176	Декабрь

Контрольные вопросы:

1. С помощью каких инструментов наполняются таблицы в MS Access?

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №11

Тема: «Создание отчетов в Microsoft Access»

Цель: Получить практический опыт создания отчетов любой сложности средствами СУБД MS Access.

Время выполнения: 4 урока

Ход работы:

Основные сведения об отчетах в Access

С помощью отчетов можно просматривать, форматировать и группировать информацию в базе данных Microsoft Access. Например, можно создать простой отчет в виде списка номеров телефонов всех контактов или сводный отчет по итогам продаж компании в разных регионах за разные периоды.

В этой статье представлен обзор отчетов в Access, сведения об основах создания отчета и использования функций сортировки, группирования и обобщения данных, а также о том, как предварительно просмотреть и напечатать отчет.

Обзор отчетов в Access

Возможности использования отчета

Отчет — это объект базы данных, удобный для представления ее информации с одной из следующих целей:

- отображение или распространение сводки данных;
- архивирование моментальных снимков данных;
- предоставление сведений об отдельных записях;
- создание надписей.

Составные части отчета

Можно создавать «несвязанные» отчеты, в которых данные не отображаются, но в этой статье мы рассмотрим отчет, связанный с источником данных (таблицей или запросом). Макет отчета разбит на

разделы, которые можно просмотреть в режиме конструктора. Понимание принципов работы каждого раздела поможет создать лучшие отчеты. Например, от выбора раздела для размещения вычисляемого элемента управления зависит способ вычисления результата. Ниже перечислены типы разделов и указано назначение каждого из них:

Раздел	Отображение раздела при печати	Использование раздела
Заголовок отчета.	В начале отчета.	В заголовок включается информация, обычно помещаемая на обложке, например, эмблема компании, название отчета или дата. Если в заголовке отчета помещен вычисляемый элемент управления, использующий статистическую функцию Sum, сумма рассчитывается для всего отчета. Заголовок отчета печатается перед верхним колонтитулом.
Верхний колонтитул.	Вверху каждой страницы.	Верхний колонтитул используется в тех случаях, когда нужно, чтобы название отчета повторялось на каждой странице.
Заголовок группы.	В начале каждой новой группы записей.	Используется для печати названия группы. Например, если отчет сгруппирован по изделиям, в заголовках групп можно указать их названия. Если поместить в заголовок группы вычисляемый

		элемент управления, использующий статистическую функцию Sum, сумма будет рассчитываться для текущей группы. Заголовок группы может состоять из нескольких разделов в зависимости от добавленных уровней группирования. Дополнительные сведения о создании верхних и нижних колонтитулов группы см. раздел «Добавление группирования, сортировки или итоговых данных».
Область данных.	Отображается один раз для каждой строки в источнике записей.	В нем размещаются элементы управления, составляющие основное содержание отчета.
Примечание группы.	В конце каждой группы записей.	Примечание группы можно использовать для печати сводной информации по группе. Нижний колонтитул группы может состоять из нескольких разделов в зависимости от добавленных уровней группирования.
Нижний колонтитул.	Внизу каждой страницы.	Используется для нумерации страниц и для печати постраничной информации.
Примечание	В конце отчета.	Примечание отчета можно

отчета.	Примечание: В режиме конструктора примечание отчета отображается под нижним колонтитулом. Однако во всех остальных режимах (например, в режиме макета, при печати или предварительном просмотре) этот раздел помещается *над*нижним колонтитулом, сразу после примечания последней группы или строки области данных на последней странице.	использовать для печати итогов и другой сводной информации по всему отчету.

Создать значимый отчет гораздо проще, если таблицы в базе данных имеют хорошо разработанные структуру и связи.

Создание отчета в Access

Чтобы создать отчет для базы данных Access на компьютере, выполните следующие действия.

Действие 1. Выбор источника записей

Источником записей для отчета может быть таблица, именованный или внедренный запрос. Источник записей должен содержать все строки и столбцы данных, которые требуется отобразить в отчете.

Если нужные данные содержатся в существующей таблице или запросе, выделите эту таблицу или запрос в области навигации и перейдите к действию 2.

Если источник записей еще не создан, выполните одно из перечисленных ниже действий.

Перейдите к действию 2 и воспользуйтесь инструментом Пустой отчет;

ИЛИ

создайте таблицы или запрос, которые будут содержать нужные данные, выберите их в области навигации и перейдите к действию 2.

Действие 2. Выбор инструмента отчета

Инструменты отчета расположены на вкладке Создать в группе Отчеты. В следующей таблице описаны такие параметры:

Средство	Описание
Отчет	Позволяет создать простой табличный отчет, содержащий все поля из источника записей, который выбран в области навигации.
Конструктор отчетов	Открывает в режиме конструктора пустой отчет, в который можно добавить необходимые поля и элементы управления.
Пустой отчет	Позволяет открыть пустой отчет в режиме макета и отобразить область задач «Список полей», из которой можно добавить поля в отчет.
Мастер отчетов	Служит для вызова пошагового мастера, с помощью которого можно задать поля, уровни группировки и сортировки и параметры макета.
Наклейки	Вызывает мастер, в котором можно выбрать стандартный

	или настраиваемый размер подписей, набор отображаемых полей и порядок их сортировки.

Действие 3. Создание отчета

Для выбора требуемого инструмента нажмите соответствующую кнопку панели инструментов. После появления мастера следуйте всем его командам и на последней странице нажмите кнопку Готово. Access отображает отчет в режиме макета.

Отформатируйте отчет, чтобы добиться желаемого внешнего вида.

Измените размер полей и подписей, выделяя их и перетаскивая края.

Расположите поля в нужном порядке, выделяя их (и соответствующие подписи, если они есть) и перетаскивая в нужное место.

Также можно, щелкая поля правой кнопкой мыши, с помощью команд контекстного меню объединять или разбивать ячейки, удалять и выделять поля и выполнять другие задачи форматирования.

Кроме того, отчет можно сделать более привлекательным и удобным с помощью функций, описанных в разделах ниже.

Добавление группировки, сортировки и итогов

Чтобы быстро добавить в отчет группировку, сортировку или итоги, щелкните правой кнопкой мыши поле, к которому необходимо применить соответствующую функцию, и выберите нужную команду в контекстном меню.

Кроме того, когда отчет открыт в режиме макета или конструктора, можно добавить эти функции с помощью области "Группировка, сортировка и итоги".

Если область "Группировка, сортировка и итоги" не открыта, на вкладке Конструктор в группе Группировка и итоги выберите команду Группировка.

Нажмите кнопку Добавить группировку или Добавить сортировку и выберите поле, по которому требуется сгруппировать или отсортировать данные.

Чтобы задать дополнительные параметры или добавить итоги, в строке группировки или сортировки выберите команду Больше.

Выделение данных с помощью условного форматирования

Доступ включает средства для выделения данных в отчете. Вы можете добавить правила условного форматирования для каждого элемента управления или группу элементов управления и отчетах клиента, вы также можете добавить гистограммы для сравнения данных.

Добавление условного форматирования к элементам управления:

Щелкните отчет в области навигации правой кнопкой мыши и выберите пункт Режим макета.

Выберите необходимые элементы управления и на вкладку Формат, в группе Форматирование элемента управления, нажмите кнопку Условное форматирование.

Совет: Чтобы выбрать несколько элементов управления, щелкайте их, удерживая клавишу CTRL.

В диалоговом окне Диспетчер правил условного форматирования выберите команду Создать правило.

В диалоговом окне Новое правило форматирования выберите значение в поле Выберите тип правила.

Чтобы создать правило, которое будет проверяться отдельно для каждой записи, выберите вариант Проверьте значения в текущей записи или используйте выражение.

Чтобы создать правило для сравнения записей друг с другом с помощью гистограмм, выберите вариант Сравнить с другими записями.

В поле Изменение описания правила укажите правило, которое будет применяться при форматировании, и тип форматирования, а затем нажмите кнопку ОК.

Если нужно создать еще одно правило для этого элемента управления или группы элементов, повторите процедуру, начиная с действия 4.

Настройка цвета и шрифтов

Для настройки цвета и шрифтов попробуйте параметр Тема приложения.

Откройте отчет в режиме макета, щелкнув правой кнопкой мыши в области навигации и выбрав Макет Vросмотр.

В пункте Работа с макетами отчетов на вкладке Конструктор выберите параметр Темы и наводите курсор на различные темы в галерее, чтобы предварительно просмотреть эффекты. Чтобы выбрать тему, щелкните ее и сохраните отчет.

С помощью коллекций Цвета и Шрифты можно задать цвета и шрифты независимо друг от друга.

Добавление логотипа или фонового рисунка

В отчет также можно добавить логотип или фоновый рисунок. При обновлении рисунка он автоматически обновляется в любом месте базы данных.

Добавление или удаление рисунка

В области навигации щелкните отчет правой кнопкой мыши и выберите команду Макет.

В отчете щелкните место, куда необходимо добавить рисунок, и на вкладке Конструктор в группе Колонтитулы выберите пункт Логотип.

Перейдите к рисунку и нажмите кнопку Открыть. Access добавит рисунок в отчет.

Чтобы удалить рисунок, щелкните его правой кнопкой мыши и выберите в контекстном меню пункт «Удалить».

Добавление фонового изображения

В области навигации щелкните отчет правой кнопкой мыши и выберите команду Макет.

На вкладке Формат в группе Фон нажмите кнопку Фоновый рисунок.

Выберите рисунок из списка Коллекция изображений или нажмите кнопку Обзор, выберите рисунок и нажмите кнопку ОК.

Предварительный просмотр и печать отчета

Предварительный просмотр отчета

Щелкните правой кнопкой мыши имя отчета в области навигации и выберите команду Предварительный просмотр. Команды на вкладке Предварительный просмотр можно использовать, чтобы:

- напечатать отчет;
- изменить размер или макет страницы;
- изменить масштаб или просмотреть сразу несколько страниц;
- обновить данные в отчете;
- экспортировать отчет в другой формат файла.
- Нажмите кнопку Закрыть режим предварительного просмотра

Печать отчета

Печать отчета без предварительного просмотра

Щелкните правой кнопкой мыши имя отчета в области навигации и выберите команду Печать. Отчет будет отправлен на принтер, используемый по умолчанию.

Примечание: При выборе отчета в области навигации и выборе команды Печать на вкладке Файл можно установить дополнительные параметры печати (количество страниц и копий) и указать принтер.

Чтобы открыть диалоговое окно, в котором можно выбрать принтер, указать число копий и задать другие параметры, выберите команду Печать.

Задание: Создание простых отчетов.

Создадим простой отчет для таблицы «Справочник кадров». Для этого выберем таблицу «Справочник кадров». На ленте нажмем на вкладку «Создать» - «Простой отчет»

.

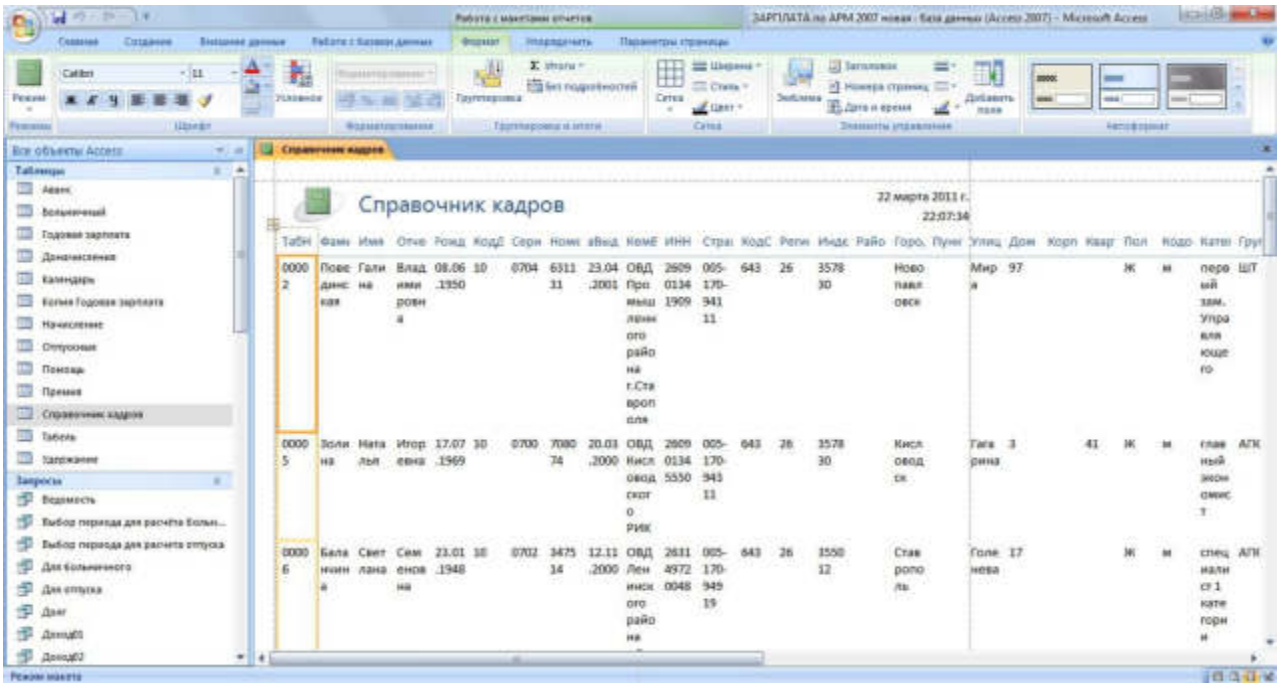


Рисунок 1 – Простой отчет таблицы «Справочник кадров» в режиме «Макет» Просмотрим этот отчет в режиме «Конструктор».

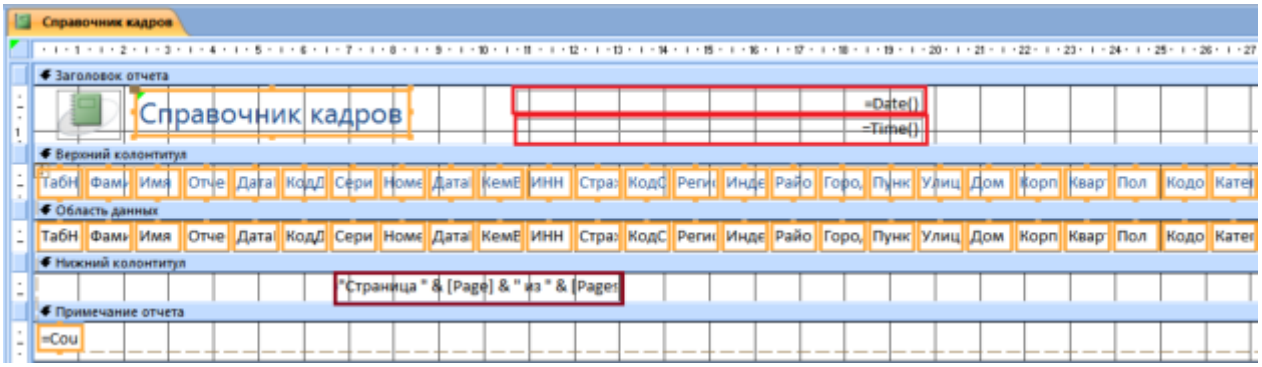


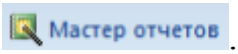
Рисунок 2 - Простой отчет таблицы «Справочник кадров» в режиме «Конструктор»

Выделенные поля являются формулами:

Date()	Текущая дата
Time()	Текущее время
Page	Текущий номер страницы
Pages	Общее количество страниц

Создайте простые отчеты для всех таблиц.

Задание 2. Создание отчетов с помощью «Мастер отчетов».

Создадим отчет при помощи мастера отчетов к таблице «Табель». Для этого выберем таблицу «Табель», перейдем на ленте на вкладку «Создание» и нажать кнопку «Мастер отчетов» 

. Далее следуем инструкциям на рисунках.

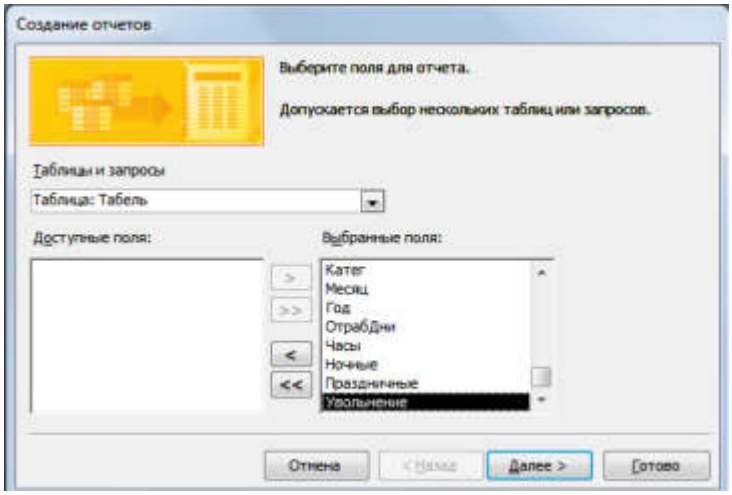


Рисунок 7 – Выбор всех доступных полей для создания запроса

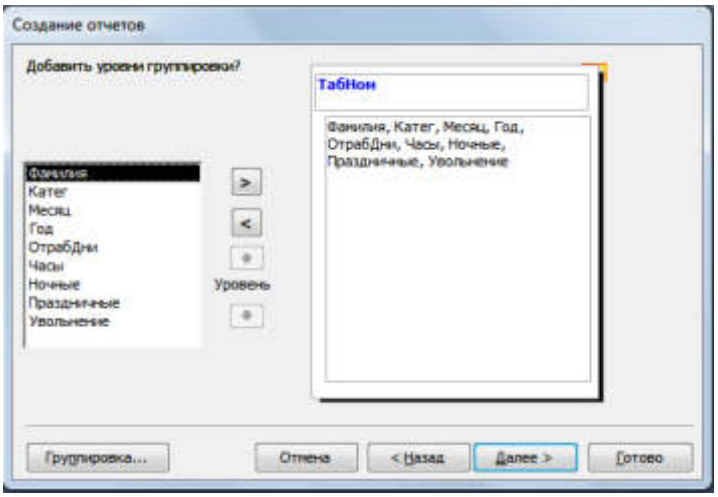


Рисунок 8 – Выбор для группировки поля «ТабНом»

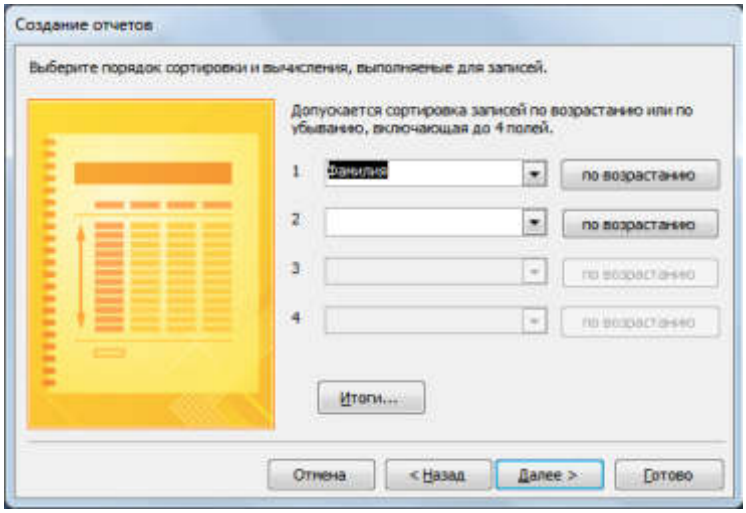


Рисунок 9 – Выбор для сортировки поле «Фамилия»

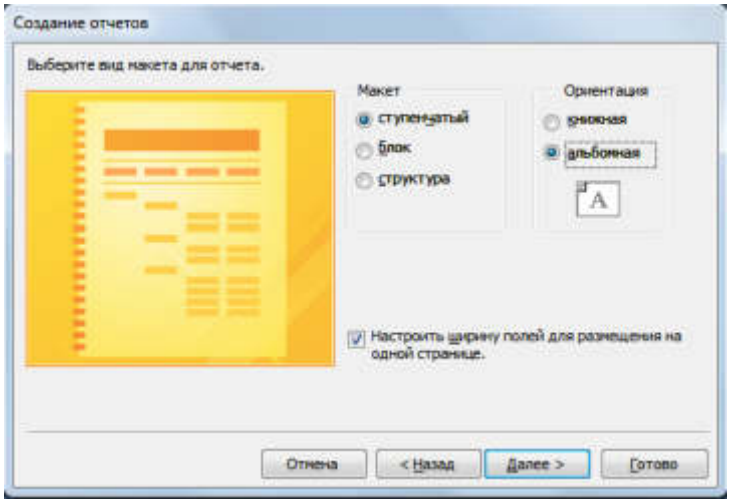


Рисунок 10 – Выбор вида макета для отчета и его ориентации на странице

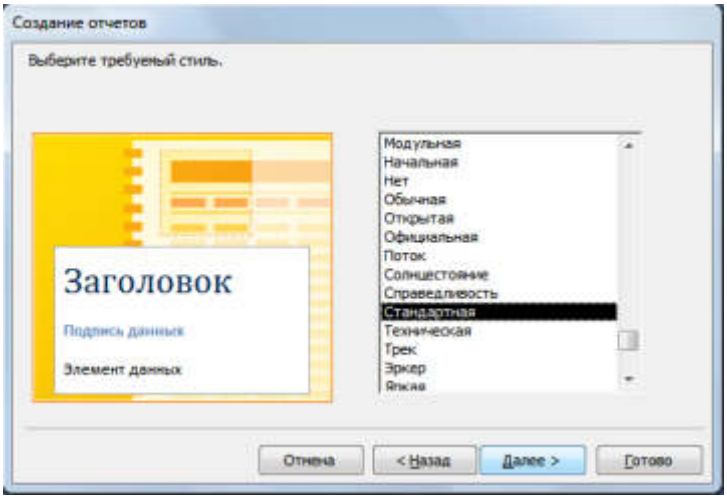


Рисунок 11 – Выбор стиля отчета

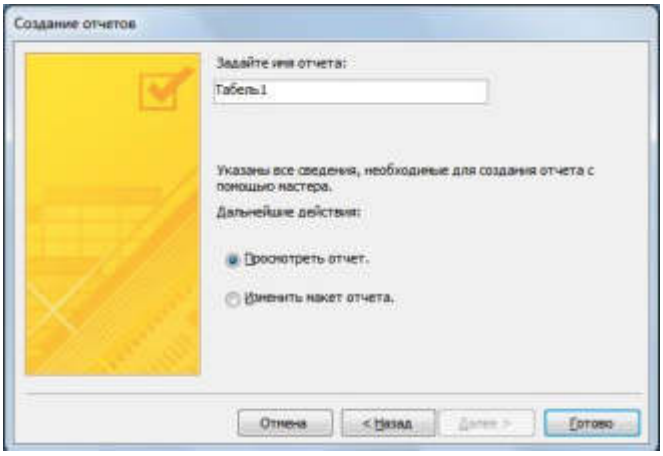


Рисунок 12 – Задание имя отчета и выбор дальнейших действий

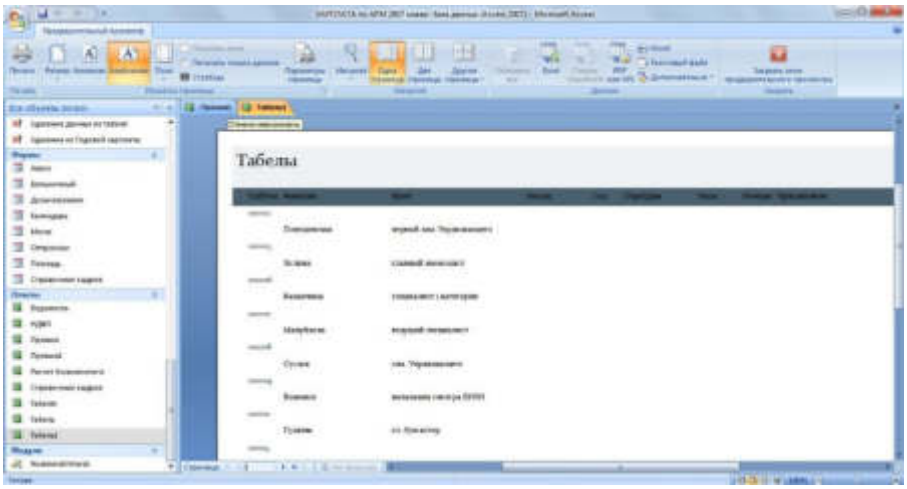


Рисунок 13 – Отчет «Табель1» в режиме «Предварительный просмотр»

Создайте аналогичный отчет «Премия1» таблицы «Премия» при помощи мастера отчетов (рисунок 13).

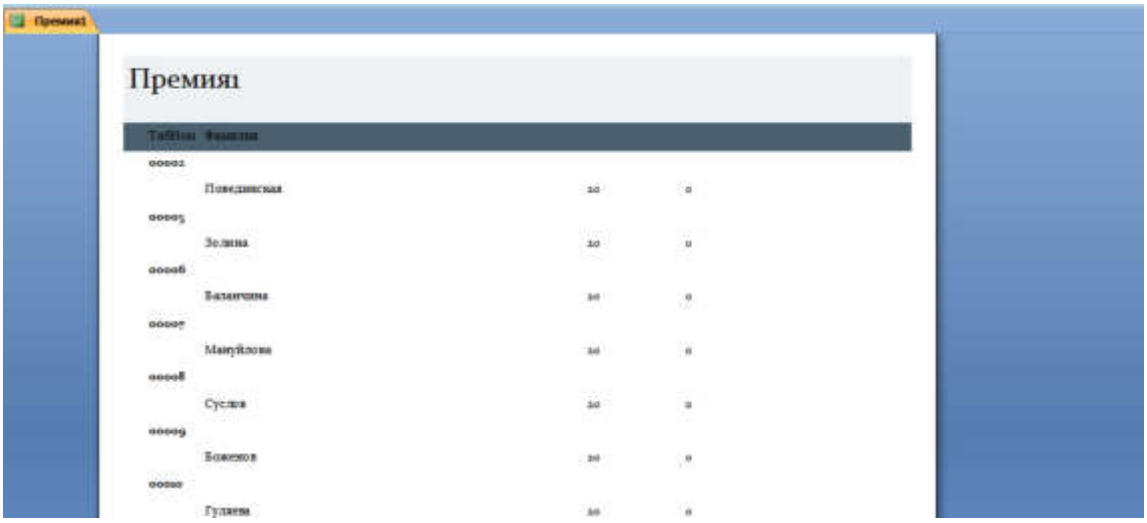


Рисунок 14 – Отчет «Премия1» в режиме «Предварительный просмотр»

Контрольные вопросы:

1. С помощью каких инструментов создаются отчеты в MS Access?
2. Как подготовить отчет к печати?
3. Какие типы отчетов существуют в MS Access?

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №12

Тема: «Вывод результатов запросов в виде отчетов»

Цель: Научиться выводить результаты запросов в виде отчетов с помощью инструментов СУБД MS Access.

Время выполнения: 4 урока

Ход работы:

Запрос является мощным и удобным средством выборки взаимосвязанных данных. Поэтому с помощью запроса можно подготовить данные для сложного отчета.

Рассмотрим технологию создания сложного отчета с использованием запроса на примере формирования бланка "Экзаменационная ведомость" для группы студентов по заданному предмету. Бланк должен иметь форму документа, используемого для ввода данных после внесения оценок преподавателем в этот бланк.

Проект макета отчета. Бланк " Экзаменационная ведомость", которая должна выводиться из базы данных в режиме просмотра и печати отчета.

государственный комитет российской федерации
по высшему образованию

Санкт-Петербургский университет экономики и финансов

Экзаменационная ведомость

№ п/п	Фамилия И.О.	Отметка о сдаче	Подпись преподавателя

Итого:

Отлично_____

Хорошо_____

Удовлетворительно_____

Неудовлетворительно_____

Явка

Подпись преподавателя

Рис. 1 Проект макета отчета Экзаменационная ведомость.

Рассмотрим подготовку запроса, обеспечивающего выборку информации, необходимой для формирования бланка экзаменационной ведомости.

Создание запроса для подготовки данных в отчет

Определение схемы данных запроса

Для вывода в отчет реквизитов, указанных в проекте макета нужно определить таблицы - источники и их взаимосвязи.

Таблицы источники данных. Для вывода в шапку бланка экзаменационной ведомости значений реквизитов Вид сдачи, Группа необходимы данные из полей ВИДЗ, НГ таблицы ИЗУЧЕНИЕ. Для вывода значений реквизитов с общей надписью Преподаватель (фамилия, ученое звание) необходимы данные из полей ФИО, ЗВ таблицы ПРЕПОДАВАТЕЛЬ. Для вывода реквизита Название предмета необходимы данные из поля НП таблицы ПРЕДМЕТ.

Для вывода в табличную часть бланка экзаменационной ведомости значений реквизитов N п/п, Фамилия И.О. необходимы данные из полей НС, ФИО таблицы СТУДЕНТ.

Таким образом, запрос для выборки этих взаимосвязанных данных должен быть построен на основе таблицы ИЗУЧЕНИЕ, ПРЕПОДАВАТЕЛЬ, ПРЕДМЕТ, СТУДЕНТ.

Связи между таблицами запроса. При создании запроса связи между таблицами установятся автоматически. Связи таблиц ИЗУЧЕНИЕ, ПРЕДМЕТ, ПРЕПОДАВАТЕЛЬ определяются в соответствии со схемой данных БД.

При создании запроса между таблицами СТУДЕНТ и ИЗУЧЕНИЕ автоматически установится также связь по одноименному полю НГ (номер группы). Эта связь является связью - объединением, которой нет в схеме данных базы. Заметим, что эти таблицы находятся в отношениях многие-ко-многим, поскольку один студент изучает много предметов, и один предмет изучается многими студентами. Связь, установленная между таблицами

СТУДЕНТ и ИЗУЧЕНИЕ, определяет операцию симметричного объединения. При этом записи из этих таблиц объединяются и добавляются в результат только в том случае, если связанные поля содержат одинаковые значения.

Конструирование запроса для подготовки макета отчета.

Процесс включения в запрос необходимых таблиц и полей из них является достаточно простым и выполняется по технологии, подробно рассмотренной в главе 5. В строке бланка запроса Условие отбора (Criteria) определим параметры запроса |Номер группы| и |Наименование предмета| для аналогового ввода их значений при выполнении запроса. Это позволяет получить данные для конкретной ведомости.

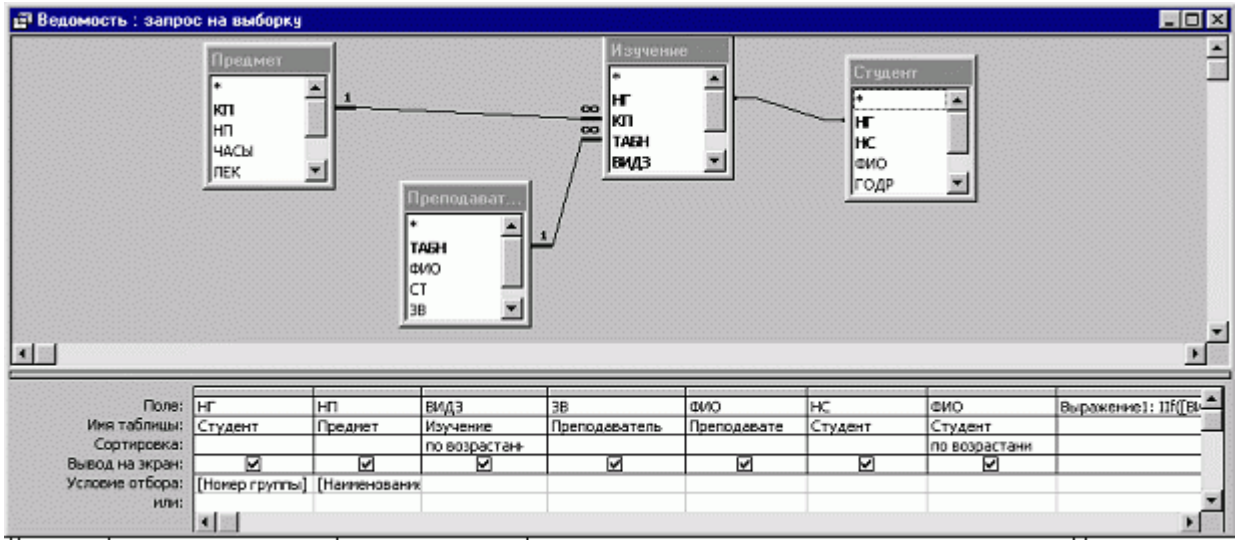


Рис. 2 Запрос на выборку в режиме конструктора для формирования отчета-бланка экзаменационной ведомости

В отчете необходимо выводить значения реквизита Вид сдачи: Экзамен, Зачет, которых нет непосредственно в таблицах БД, но они могут быть получены на основе значений поля ВИДЗ таблицы ИЗУЧЕНИЕ. Два значения реквизита Вид сдачи: "Экзамен" и "Зачет" соответствуют двум возможным значениям поля ВИДЗ: "лек" и "пр". Фактически нужно вместо значения "лек" формировать слово "Экзамен", а вместо "пр" - слово "Зачет". Для этого надо в запрос ввести новое поле, которое формируется как вычисляемое. Это поле должно содержать встроенную функцию управления.

Выражение 1: if ([ВИДЗ] = "лек"; "Экзамен"; "Зачет")

Именно этого вычисляемого поля по умолчанию является "Выражение 1:"

Функция Iif ("immediate if" - мгновенное условие) аналогична инструкции if… Then… Else и имеет следующий формат:

if (условие; если Истина; если Ложь)

В соответствии с форматом этой функции, если выполнится условие (ВИД)= "лек" (т.е. в поле ВИДЗ находится значение "лек"), ТО РЕЗУЛЬТАТОМ ФУНКЦИИ БУДЕТ "Экзамен". В противном случае, т.е. если (ВИДЗ) = "пр" результатом функции будет слово "Зачет".

В предыдущем рисунке приведен в окончательном виде запрос для подготовки данных отчета-бланка экзаменационной ведомости. Этот запрос сохранен под именем "Ведомость".

Формирование записей результатов в запросе.

В процессе формирования записей результата из таблицы СТУДЕНТ последовательно выбираются записи с заданным номером группы (НГ).

С каждой записью таблицы СТУДЕНТ объединяются связанные по номеру группы НГ записи из таблицы ИЗУЧЕНИЕ. Из таблицы ИЗУЧЕНИЕ в этом объединении участвуют только записи с заданным наименованием предмета.

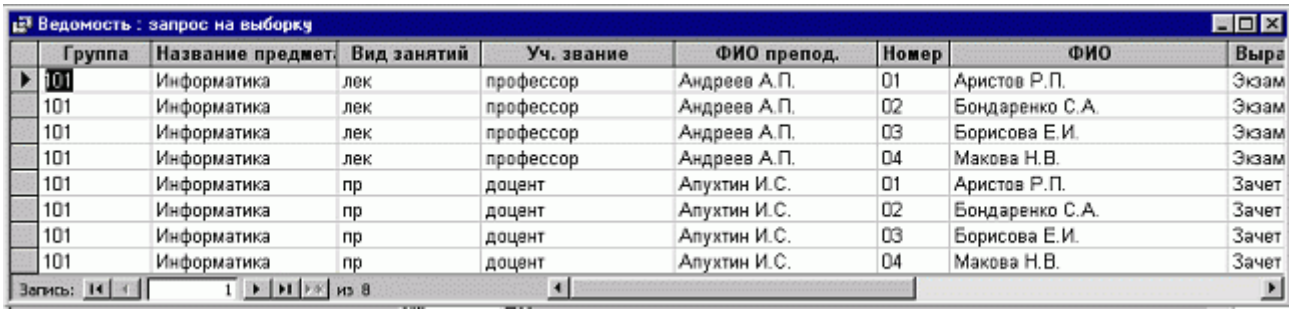
Дополнение формируемых запросом записей полями из таблицы ПРЕДМЕТ и ПРЕПОДАВАТЕЛЬ не добавляет новых записей. Эти таблицы представляют сторону один в отношении один-ко-многим и служат лишь для расшифровки кодов КП и ТАБН из таблицы ИЗУЧЕНИЕ.

Если по заданному предмету проводится не одно, а два вида занятий: лекции и практика, то число формируемых записей удваивается. В этом случае должны быть сформированы две ведомости - одна для сдачи экзамена, другая для сдачи зачета.

Заметим, что таблица результатов запроса содержит ненормализованные данные, что проявляется в повторяемости значений в

столбцах ГРУППА, НАИМЕНОВАНИЕ ПРЕДМЕТА, ФИО ПРЕПОД., УЧ. ЗВАНИЕ, ВИД ЗАНЯТИЙ и ВЫРАЖЕНИЕ1. Поэтому данная таблица не соответствует проекту макета отчета Экзамена ведомость не может непосредственно использоваться как выходной документ.

Использование средств Access для подготовки отчета обеспечивает преобразование получаемой таблицы результатов запроса в выходной документ нужной формы в соответствии с проектом макета отчета.



Ведомость : запрос на выборку

Группа	Название предмет	Вид занятий	Уч. звание	ФИО препод.	Номер	ФИО	Выра
101	Информатика	лек	профессор	Андреев А.П.	01	Аристов Р.П.	Экзам
101	Информатика	лек	профессор	Андреев А.П.	02	Бондаренко С.А.	Экзам
101	Информатика	лек	профессор	Андреев А.П.	03	Борисова Е.И.	Экзам
101	Информатика	лек	профессор	Андреев А.П.	04	Макова Н.В.	Экзам
101	Информатика	пр	доцент	Апухтин И.С.	01	Аристов Р.П.	Зачет
101	Информатика	пр	доцент	Апухтин И.С.	02	Бондаренко С.А.	Зачет
101	Информатика	пр	доцент	Апухтин И.С.	03	Борисова Е.И.	Зачет
101	Информатика	пр	доцент	Апухтин И.С.	04	Макова Н.В.	Зачет

Запись: 1 из 8

Рис.3 Результат выполнения запрса с данными для экзаменационной ведомости

Конструирование отчета на основе запроса

Рассмотрим технологию отчета на основе запроса Ведомость подготовленного выше. Для конструирования отчета в окне базы данных выберем объект Отчеты (Reports) и нажмем кнопку Создать (NEW). В окне Новый отчет (New Report) выберем запрос Ведомость, который будет источником данных для отчета. Выберем строку Конструктор (Design) для разработки отчета.

Размещение данных в разделах отчета

Результаты запроса содержат много полей с повторяющимися значениями НГ, НП, ФИО преподавателя, ЗВ, ВИДЗ, Выражение 1: Данные в отчете должны быть размещены, как показано на макете отчета Экзаменационная ведомость. Значение каждого из этих полей должно быть представлено в отчете в заголовке один раз.

Заметим, что результаты для отчета получены по заданной группе и предмету. Поскольку по предмету может быть два вида занятий и для каждого вида занятий предполагается вывод отдельной ведомости,

необходимо выполнить группировку по полю ВИДЗ. Для группировки по виду занятия нажмем кнопку Сортировка и группировка (Sorting and Grouping) на панели инструментов конструктора отчетов и заполним поля открывшегося окна.

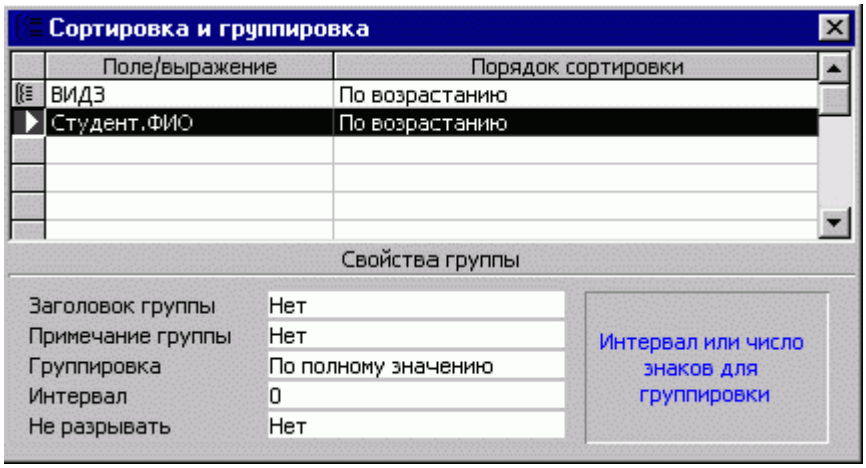


Рис. 4 Окно определения групп записей по полю ВИДЗ и сортировки по полю НС.

После определения группировки в окне конструктора отчета появятся разделы Заголовок группы "ВИДЗ" (ВИДЗ Header) и Примечание группы "ВИДЗ" (ВИДЗ Footer).

Чтобы каждая группировка могла быть оформлена как самостоятельная экзаменационная ведомость, будем формировать шапку макета ведомости в заголовке группы ВИДЗ, а строки о подведении итогов сдачи экзамена (или зачета) в примечании группы ВИДЗ.

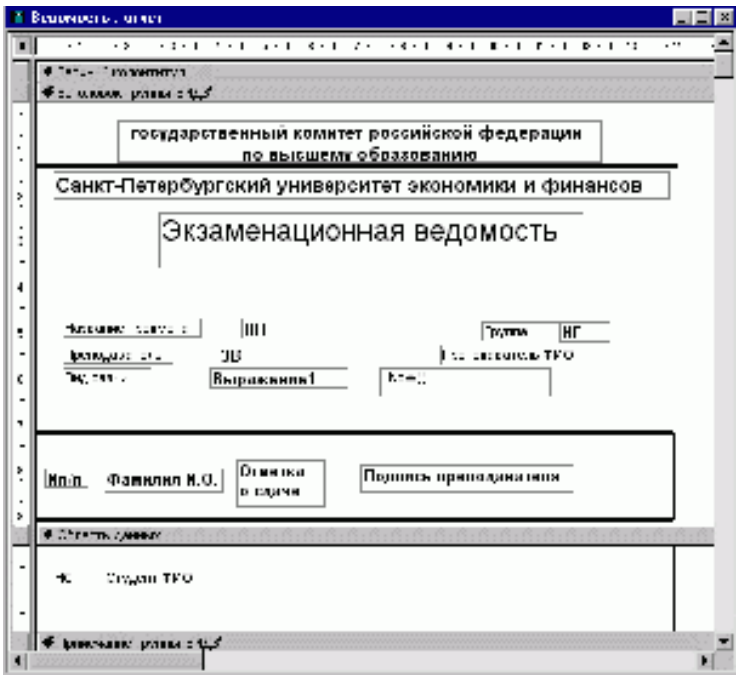


Рис. 5 Оформление шапки и табличной части отчета.

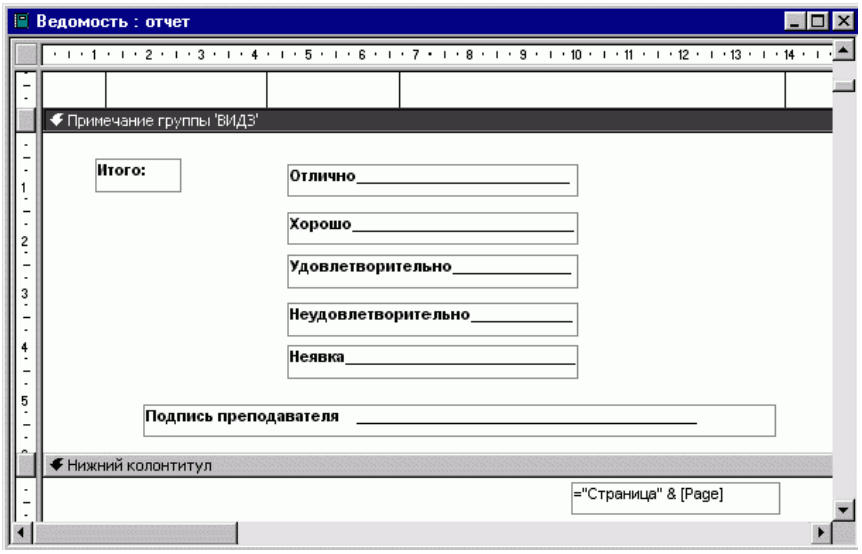


Рис.6 Оформление итоговой части отчета.

Создание текстовых элементов отчета.

Для создания в заголовке и примечании группы ВИДЗ текстовых элементов за исключением названий реквизитов выводимых полей, используем кнопку Надпись (Label). Для оформления текста можно воспользоваться кнопками панели форматирования. Кроме того, можно задавать параметры надписей в окне свойств этих элементов. Окно свойств элемента открывается кнопками панели инструментов Свойства (Properties). Элемент предварительно должен быть выделен.

При размещении элементов может понадобиться расширить раздел отчета. Для этого установить курсор мыши на его нижнюю границу так, чтобы появилась двунаправленная стрелка, и переместить эту стрелку вниз.

Оформление табличной части отчета.

Для размещения в отчете полей вызовем окно списка полей запроса Ведомость, нажав на панели конструктора отчетов кнопку. Список панелей (Field List) .

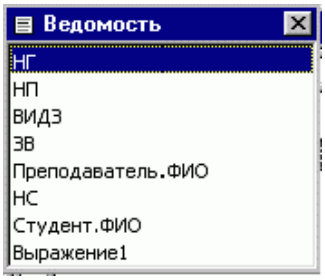


Рис. 7 Список полей запроса Ведомость

В область данных разместим поля НС и ФИО, на основе содержимого которых будут формироваться строки табличной части отчета в соответствии с проектом макета отчета. Для этого из списка полей перетащим поля НС (номер студента) и СТУДЕНТ. ФИО в область данных.

Каждое поле представляется двумя элементами: собственно, полем и его подписью. Например, поле ФИО таблицы СТУДЕНТ представляется элементом СТУДЕНТ. ФИО, отображающим значение поля, и элементом, содержащим подпись этого поля из таблицы базы данных - Фамилия И.О. Установим нужный шрифт в элементах. Используем команду меню Формат |Размер| по размеру данных (Format |Size| Size To Fit) для установки размеров элемента по размеру текста подписи.

Замети, что подпись поля из табличной части должна быть перемещена для формирования заголовка столбцов в раздел Заголовок группы "ВИДЗ" (ВИДЗ Header). Для этого выделим подпись поля. Затем вырежем его. Активизируем раздел Заголовок группы "ВИДЗ" (ВИДЗ Header) и вставим подпись в этот раздел.

Оформим элементы в соответствии с требованиями проекта макета отчета. В частности, заменим подпись поля "Номер студента на N п/п". Для того чтобы в бланке экзаменационной ведомости были пустые столбцы для оценок и подписи (заполняемые преподавателем вручную на экзамене), добавим в строку с заголовками столбцов новые текстовые элементы "Отметка о сдаче" и "Подпись преподавателя".

Размещение полей в заголовке

Значения полей НГ, НП, ФИО преподавателя, ЗВ, Выражение 1 (экзамен, зачет) повторяются во всех записях группировки. Поэтому они должны размещаться в заголовке группы. Перетащим эти поля из списка полей в раздел Заголовок группы "ВИДЗ" (ВИДЗ Header).

Добавление текущей даты и номера страницы

Для добавления в отчет текущей даты воспользуемся встроенной функцией Now (). Для этого создадим несвязанный элемент, нажав кнопку Поле (Text Box) на панели элементов и разместим его в нужном месте. Зададим в окне его свойств на вкладке Данные (Data) в строке Данные (Control Source) выражение = Now (), а на вкладке Макет (Format) в строке Формат поля (Format) выберем значение - Средний формат даты (Medium Date).

Для добавления номера страницы в раздел Нижний колонтитул создадим несвязанный элемент и заполним в его свойствах строку Данные (Control Source) выражением = "Страница" @ [Page].

Создадим горизонтальные и вертикальные линии в соответствии с макетом, воспользовавшись кнопкой Линии (Line). Установим нужную толщину линий в свойствах этого элемента.

Сохраним отчет под именем "Экзаменационная ведомость".

Предварительный просмотр отчета.

При подготовке к просмотру отчета, построенного на базе запроса с параметрами, Acceess предварительно выполняет запрос и выводит диалоговые окна ввода параметров отчета.

Просмотр отчета. Для отображения отчета Экзаменационная ведомость на экране в том виде, в котором он будет напечатан, выйдем из режима конструктора отчетов, нажав кнопку Вид (View). Последовательно появляются диалоговые окна для ввода параметров запроса, которые являются в то же время параметрами отчета.

Введем значение параметра "Номер группы" - "101" и значение параметра "Наименование предмета" - "Информатика". Для данных значений параметров отчет будет состоять из двух страниц, на первой из которых будет представлена экзаменационная ведомость для экзамена по информатике, на второй - для сдачи зачета. Первая и вторая страницы отчета Экзаменационная ведомость приведены на рисунках

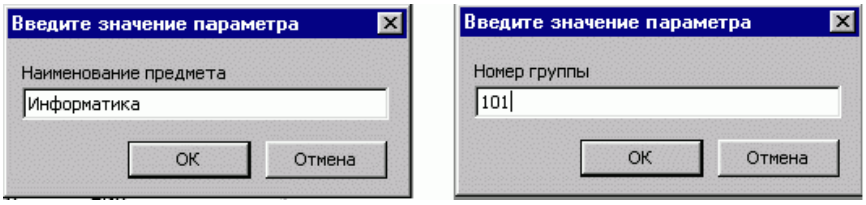


Рис. 8 Диалоговые окна ввода параметров отчета

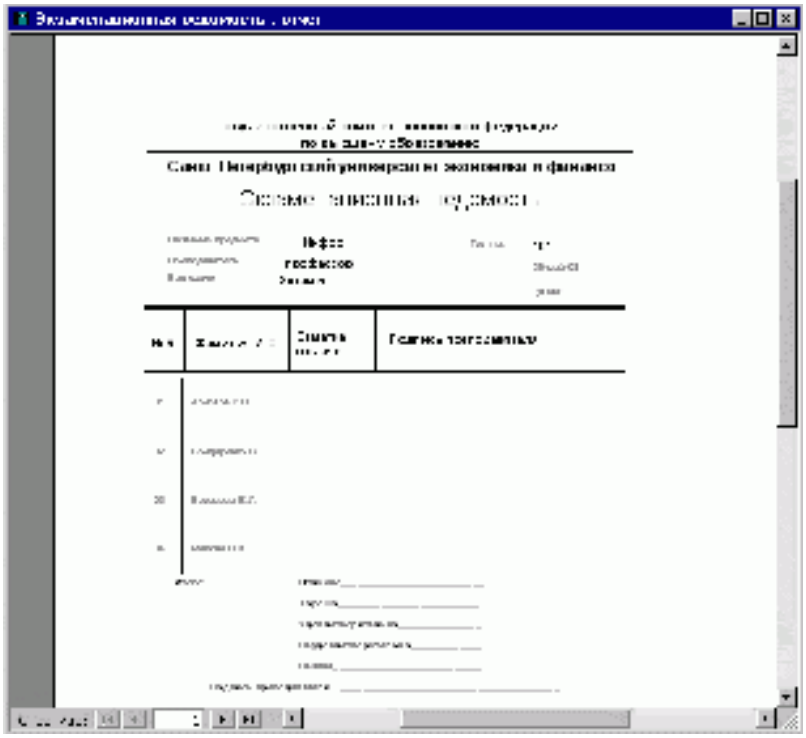


Рис. 9 Вывод бланка экзаменационной ведомости для лекций на первой странице отсчета

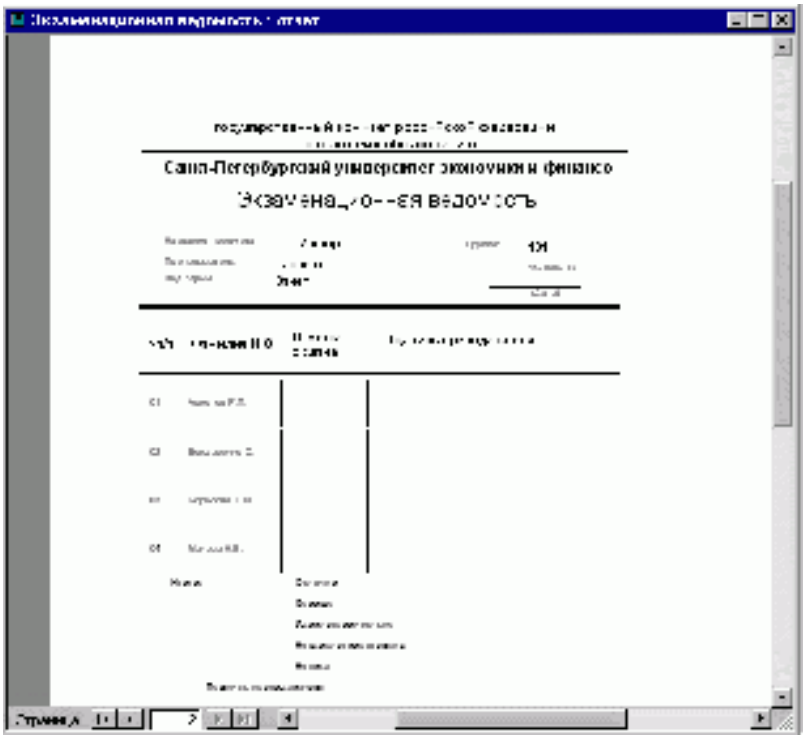


Рис. 10 Вывод бланка экзаменационной ведомости для практики на второй странице отчета

Выводы и оформление отчета

ПРАКТИЧЕСКАЯ РАБОТА №13

Тема: «Управление объектами с помощью макросов»

Цель:

Время выполнения: 4 урока

Ход работы:

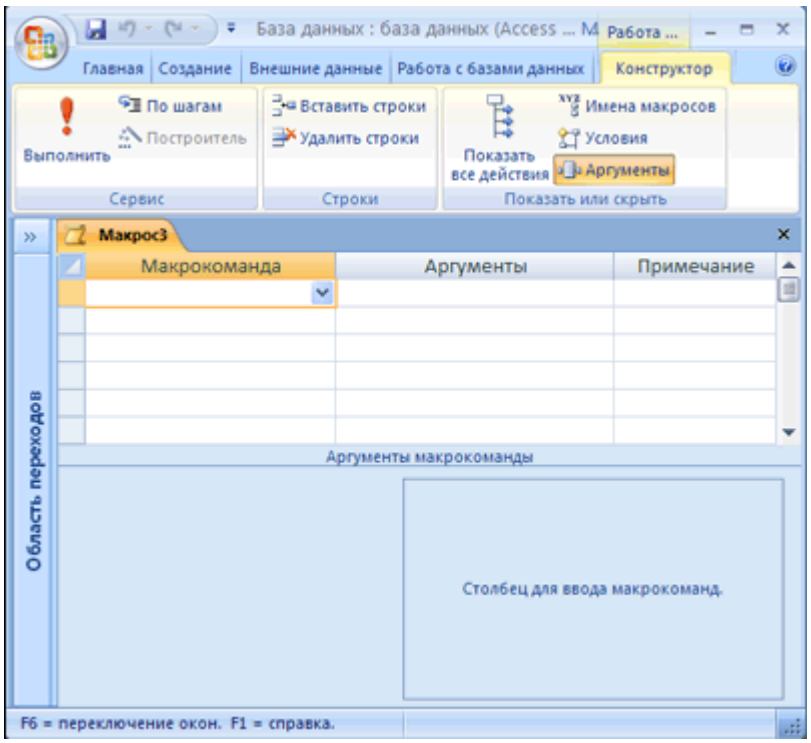
Что такое макрос?

Макрос — это средство, которое позволяет автоматизировать задачи и добавить функциональные возможности в формы, отчетов и элементов управления. Например, если кнопки Добавить в форму, связать событие OnClick кнопки для макроса и макрос содержит команды, что требуется выполнять каждый раз, когда она нажата кнопка.

В приложении Access полезно упрощенный язык программирования, который записи путем создания перечень действий, выполняемых можно считать макросы. Когда вы создаете макроса, выберите каждое из этих действий из раскрывающегося списка и заполните необходимые сведения для каждого действия. Макросы позволяют добавлять функциональные возможности для форм, отчетов и элементы управления без написания кода в модуле Visual Basic для приложений (VBA). Макросы предоставляют набор команд, доступных в VBA и большинство людей проще, можно создать макрос, чем писать код VBA.

Предположим, что вы хотите запустить отчет непосредственно из одной формы для ввода данных. Можно добавить кнопку в форму, а затем создайте макрос, который запускает отчет. Макрос могут быть изолированного макроса (отдельный объект в базе данных), которое затем привязано к событию OnClick кнопки или непосредственно в событие OnClick самой кнопки можно внедрить макрос — это новая функция Office Access. Так или иначе, нажмите кнопку макрос запускается и запуск отчета.

Макрос создается с помощью построителя макросов, показанного на следующем рисунке.



Чтобы отобразить построитель макросов на вкладке Создание в группе другое нажмите кнопку макрос. Если эта команда недоступна, щелкните стрелку под модуль или Модуль класса кнопки и нажмите кнопку макрос

Общие сведения о макросах

Макрос термина часто используется для ссылок на объекты макрос автономную (объекты, которые отображаются в разделе **макросы** в области навигации), но на самом деле, один объект макрос может содержать несколько макросы. В этом случае он называется *Группа макросов*. Группа макросов отображается в области навигации как один объект-макрос, но группу макросов фактически содержит более одного макрос. Конечно же можно создать каждый макрос в отдельный макрос-объект, но часто имеет смысл сгруппировать несколько связанных макросов в один объект-макрос. Имя в столбце **Имя макроса** определяет каждого макроса.

Макрос состоит из отдельных макрокоманд. Для большинства макрокоманд требуется один или несколько аргументов. Каждому макросу в группе может быть присвоено имя и добавлены условия для контроля за

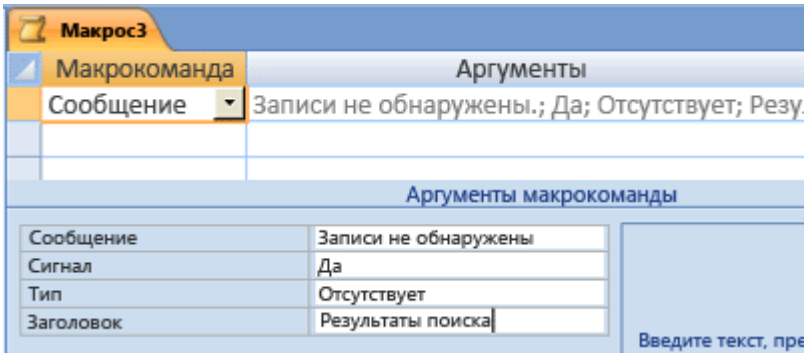
выполнением каждой макрокоманды. В следующем разделе подробно рассмотрена каждая эта особенность.

Имена макросов

Если вашего макроса объект содержит только один макрос имена макросов не нужны. Вы только что может относиться к макросу по имени макроса объект. Тем не менее в случае группу макросов необходимо назначить уникальное имя для каждого макроса. Если в столбце **Имя макроса** не отображается в конструктор макросов, щелкните **Имена макросов** в группе " **Показать или скрыть** " на вкладке " **Конструктор** ". Дополнительные сведения о выполнении макросов в группах макроса появляется далее в этой статье.

Аргументы

Аргумент — это значение, которое обеспечивает необходимую для макрокоманды информацию, например, какая строка должна отображаться в окне сообщения, с каким элементом управления следует выполнять действия и т. п. Некоторые аргументы являются обязательными, другие — нет. Аргументы отображаются в области **Аргументы макрокоманды** в нижней части окна построителя макросов.



Новая функция Office Access построитель макросов является столбец **аргументы**, которое позволяет просматривать (но не изменять) аргументы макрокоманды на той же строке действие. Это немного упрощает прочитать макрос, поскольку не требуется для выбора каждой действие для отображения аргументов. Чтобы отобразить столбец **аргументов**, нажмите

кнопку **аргументы** в группе " **Показать или скрыть** " на вкладке " **Конструктор** ".

Условия

Условие определяет требования, которые должны быть соблюдены, для того чтобы была выполнена макрокоманда. Можно использовать любое выражение, результатом которого являются значения «Истина» или «Ложь» либо «Да» или «Нет». Если выражение вычисляется как «Ложь», «Нет» или 0 (нуль), макрокоманда не будет выполнена. При любом другом значении выражения макрокоманда будет выполнена.

Одно условие может управлять несколькими макрокомандами, если в столбце **Условие** ввести многоточие (...) для каждой следующей по порядку макрокоманды, к которой будет применяться данное условие. Если выражение вычисляется как «Ложь», «Нет» или 0 (нуль), ни одна из макрокоманд не будет выполнена. При любом другом значении выражения будут выполнены все макрокоманды.

Для отображения в построителе макросов столбца **Условия** на вкладке **Конструктор** в группе **Отображение** нажмите кнопку **Условия**

Выражение, используемое как условие	**Условие, при котором выполняется макрокоманда**
[Город]="Париж"	Поле «Город» в форме, из которой запускается макрос, имеет значение «Париж».
DCount("[КодЗаказа]", "Заказы")>35	Количество записей в поле «КодЗаказа» таблицы «Заказы» превышает 35.
DCount("*", "Заказано", "[КодЗаказа]=Forms![КодЗаказа]![КодЗаказа]")>3	В таблице «Заказано» содержится более 3 записей, у которых поле «КодЗаказа» в таблице соответствует полю «КодЗаказа» в форме «КодЗаказа».

[ДатаИсполнения] Between #2-фев-2006# And #2-мар-2006#	Значение поля «ДатаИсполнения» в форме, из которой запускается макрос, попадает в интервал со 2 февраля 2006 по 2 марта 2006 г.
Forms![Товары]![На складе]<5	Значение поля «На складе» в форме «Товары» меньше 5.
IsNull([Имя])	"Имя" в форме, из которого выполняется макрос значение **null** (не имеет значения). Это выражение является эквивалентна [имя] Is Null.
[Страна]="UK" And Forms![СуммыПродаж]![ВсегоЗаказов]>100	Поле «Страна» в форме, из которой запускается макрос, имеет значение «Великобритания», и значение поля «ВсегоЗаказов» в форме «СуммыПродаж» больше 100.
[Страна] In ("Франция", "Италия", "Испания") And Len([Индекс])<>5	Поле «Страна» в форме, из которой запускается макрос, имеет значение «Франция», «Италия» или «Испания», и почтовый индекс содержит не 5 символов.
MsgBox("Подтвердить изменения?",1)=1	Нажата кнопка **ОК** в диалоговом окне, в котором функция **MsgBox** отображает текст «Подтвердить изменения?». Если в диалоговом окне нажата кнопка **Отмена** эта макрокоманда будет пропущена.
[TempVars]![MsgBoxResult]=2	Временная переменная, которая используется для хранения результата окна сообщения, равна 2 (vbCancel=2).

Совет: Чтобы приложение Access временно пропускало макрокоманду, введите в качестве условия значение «Ложь». Временный пропуск макрокоманды может применяться при поиске ошибок в макросе.

Макрокоманды

Макрокоманды — это простейшие элементы, из которых строится макрос. В приложении Access предусмотрен большой выбор макрокоманд, которые позволяют выполнять разнообразные действия. Например, наиболее часто используются макрокоманды для открытия отчета, поиска записи, отображения окна сообщения или применения фильтра к форме или отчету.

Новые возможности макросов в Office Access

В предыдущих версиях Access для выполнения наиболее часто используемых функций требовалось написание кода VBA. В Office Access 2007 включены новые возможности и макрокоманды, которые позволяют избежать этого. Благодаря им процесс добавления функциональных возможностей в базу данных стал проще и надежнее.

Внедренные макросы В новой версии появилась возможность внедрять макросы в любые события, предусмотренные в форме, отчете или элементе управления. Внедренный макрос не отображается в области переходов, он становится компонентом формы, отчета или элемента управления, в которых он был создан. При создании копии формы, отчета или элемента управления, содержащих внедренные макросы, в этой копии также будут содержаться макросы.

Увеличение размера безопасность при Изображение кнопки кнопку Показать все действия в конструктор макросов не выделен, только макрокоманд и Выполнить Команду аргументы, которые доступны для использования, не требующие надежных состояние, чтобы запустить. Макрос, построенные с помощью этих действий будет выполняться, даже когда база данных находится в режиме отключения (когда VBA запрещено). Баз данных, содержащих макрокоманд, которые не входят в списке надежных — или баз данных, которые содержат код VBA — должны быть явно предоставлено надежные состояния.

Обработка ошибок и отладка Office Access предоставляет новый макрос действий, включая OnError (аналогично инструкции "На ошибке" на языке VBA) и Устранитьошибкумакроса, которые позволяют выполнять конкретные действия при возникновении ошибок во время выполнения макроса. Кроме того, новый SingleStep макрокоманды позволяет режим один шаг в любой момент в макросе, так что наблюдать за выполнение макроса одно действие одновременно.

Временные переменные Три новых макрокоманд (SetTempVar Удалитьвремпеременную RemoveAllTempVars) позволяют создавать и использовать временные переменные в макросах. Можно использовать их в условных выражениях управления запуск макросов или и передачи данных из отчетов или форм, или для других целей, для которого требуется места временного хранения для значения. Эти временные переменные доступны на языке VBA, кроме того, можно также использовать их для обмена данными и обратно модули VBA.

Создание макроса

В Office Access макрос или группа макросов могут быть заключены в объект макроса (называемый иногда изолированным макросом), или макрос может быть внедрен в любое свойство события в форме, отчете или элементе управления. Внедренные макросы становятся частью объекта или элемента управления, в которые они внедрены. Изолированные макросы отображаются в области переходов в разделе Макросы, в отличие от внедренных.

Функции построителя макросов

Построитель макросов служит для создания и изменения макросов. Он открывается следующим образом.

На вкладке Создание в группе другое нажмите кнопку макрос. Если эта команда недоступна, щелкните стрелку под модуль или Модуль класса кнопки и нажмите кнопку макрос.

Приложение Access откроет построитель макросов.

Построитель макросов используется для создания списка макрокоманд, которые должны выполняться при запуске макроса. При первом открытии построителя макросов отображаются столбцы Макрокоманда, Аргументы и Примечание.

В области Аргументы макрокоманды при необходимости вводятся или изменяются аргументы для каждой макрокоманды. В поле описания отображается краткое описание каждой макрокоманды и аргумента. Для просмотра описания щелкните макрокоманду или аргумент.

В следующей таблице показаны команды, доступные на вкладке Конструктор в построителе макросов.

Группа	Команда	Описание
Сервис	Выполнить	Выполнение макрокоманд, перечисленных в макросе.
	Пошаговое выполнение	Включение режима пошагового выполнения макроса. При запуске макроса в этом режиме происходит поочередное выполнение каждой макрокоманды. После завершения каждой макрокоманды отображается диалоговое окно Пошаговое исполнение макроса. Для перехода к следующей макрокоманде нажмите в этом диалоговом окне кнопку Шаг. Нажмите кнопку Остановить все макросы, чтобы остановить исполнение этого и всех остальных макросов. Нажмите кнопку Далее, чтобы выйти из пошагового режима и выполнить остальные макрокоманды без остановки.
	Построитель	Эта кнопка активируется при вводе аргумента макрокоманды, содержащего выражение.

		Щелкните кнопку Построитель, чтобы открыть диалоговое окно Построитель выражений, в котором можно создать выражение.
Строки	Вставить строки	Добавление одной или нескольких пустых строк макрокоманд над выбранными строками.
	Удалить строки	Удаление выбранных строк макрокоманд.
Показать или скрыть	Показать все макрокоманды	Отображение большего или меньшего количества макрокоманд в раскрывающемся списке Макрокоманда. Чтобы отобразить полный список макрокоманд, выберите команду Показать все макрокоманды. Когда доступен полный список макрокоманд, кнопка Показать все макрокоманды выделена. При выборе макрокоманды из полного списка перед запуском макрокоманды может потребоваться предоставить базе данных явный статус доверия. Для отображения краткого списка, включающего только макрокоманды, которые могут использоваться в базе данных, не имеющей состояния надежной, кнопка Отобразить все макрокоманды не должна быть выделена. Совет: Если кнопка Отобразить все макрокоманды выделена, снимите выделение, нажав эту кнопку.

	Если кнопка Отобразить все макрокоманды не выделена, доступен краткий список макрокоманд для надежной базы данных.
Имена макросов	Отображение или скрытие столбца Имя макроса. Имена макросов необходимы в группе, чтобы можно было различать отдельные макросы, в противном случае они не обязательны. Дополнительные сведения см. в разделе Создание группы макросов далее в этой статье.
Условия	Отображение или скрытие столбца Условие. Этот столбец служит для ввода выражений, которые определяют условия выполнения макрокоманды.
Аргументы	Отображение или скрытие столбца Аргументы. В этом столбце отображаются аргументы для каждой макрокоманды, что упрощает просмотр макроса. Если столбец Аргументы не отображается, приходится щелкать каждую макрокоманду и просматривать аргументы в разделе Аргументы макрокоманды. Ввод аргументов в столбце Аргументы невозможен.

Создание изолированного макроса

На вкладке Создание в группе Другие выберите Макрос. Если эта команда недоступна, нажмите стрелку под кнопкой Модуль или Модуль класса, а затем нажмите кнопку Макрос.

Откроется построитель макросов.

Добавление макрокоманды в макрос:

В построителе макросов щелкните первую пустую ячейку в столбце Макрокоманда.

Введите нужную макрокоманду или щелкните стрелку, чтобы раскрыть список действий и выбрать из него нужную макрокоманду.

В разделе Аргументы макрокоманды укажите аргументы макрокоманды, при необходимости. Чтобы просмотреть краткое описание каждого аргумента, щелкните поле аргумент и затем прочитать описание аргумента с правой стороны. Tips

Аргументы макрокоманды, значения которых являются имя объекта базы данных можно задать аргумент путем перетаскивания объект из области навигации в поле аргумент Имя объекта действие.

Можно также создать макрокоманду, перетащив объект базы данных из области переходов в пустую строку в построителе макросов. При перетаскивании таблицы, запроса, формы, отчета или модуля в построитель макросов добавляется макрокоманда, открывающая таблицу, запрос, форму или отчет. При перетаскивании макроса в построитель макросов добавляется макрокоманда, запускающая макрос.

При необходимости введите примечания к макрокоманде.

Чтобы добавить в макрос следующую макрокоманду, перейдите в другую строка действия и повторите шаг 2. Приложение Access выполняет макрокоманды в том порядке, в котором они перечислены.

Создание группы макросов

Группа макросов создается для объединения нескольких связанных макросов в один объект макроса.

На вкладке Создание в группе Другие выберите Макрос. Если эта команда недоступна, нажмите стрелку под кнопкой Модуль или Модуль класса, а затем нажмите кнопку Макрос.

Откроется построитель макросов.

На вкладке " Конструктор " в группе Показать или скрыть щелкните Имена макросов Изображение кнопки, если он еще не выбран. Столбец Имя макроса будет отображаться в конструктор макросов.

Примечание: Имена макросов необходимы в группе, чтобы можно было различать отдельные макросы. Имя макроса появляется в одной строке с первой макрокомандой макроса. Столбец имени макроса остается пустым всех для последующих макрокоманд данного макроса. Макрос заканчивается при вводе следующего имени макроса в этом столбце.

В столбце Имя макроса введите имя первого макроса из группы.

Введите макрокоманды, выполняемые в первом макросе:

В столбце Макрокоманда щелкните стрелку, чтобы раскрыть список макрокоманд.

Выберите макрокоманду, которую требуется добавить.

В разделе Аргументы макрокоманды укажите аргументы макрокоманды, при необходимости. Чтобы просмотреть краткое описание каждого аргумента, щелкните поле аргумент и затем прочитать описание аргумента с правой стороны. Tips

Для аргументов макрокоманд, параметры которых служат именами объектов базы данных, можно задать аргумент, перетащив объект из области переходов в поле аргумента Имя объекта.

Можно также создать макрокоманду, перетащив объект базы данных из области переходов в пустую строку в построителе макросов. При перетаскивании таблицы, запроса, формы, отчета или модуля в построитель макросов добавляется макрокоманда, открывающая таблицу, запрос, форму или отчет. При перетаскивании макроса в построитель макросов добавляется макрокоманда, запускающая макрос.

При необходимости введите примечания к макрокоманде.

Перейдите к следующей пустой строке в столбце Имя макроса и введите имя очередного макроса из группы.

Добавьте макрокоманды для выполнения в этом макросе.

Повторите действия 5 и 6 для каждого макроса в группе.

При запуске группы макросов, дважды щелкнув его в области навигации или, нажав кнопку запуска Изображение кнопки в группе " Сервис " на вкладке " Конструктор ", Access запустит первого макроса, только остановка при наступлении второй имя макроса.

При сохранении группы макросов указываемое имя относится ко всей группе макросов. Это имя отображается в разделе Макросы в области переходов. Для ссылки на отдельный макрос внутри группы используется следующий синтаксис:

ИмяГруппыМакросов.ИмяМакроса

Например «кнопки.товары» ссылается на макрос «Товары» в группе макрос кнопок. В списке макрос, например, в списке аргументов имя макроса для ЗапускМакроса откроется продукты макрос как «кнопки.товары».

Создание внедренного макроса

Внедренные макросы отличаются от изолированных тем, что хранятся в свойствах событий в формах, отчетах или элементах управления. Они не отображаются как объекты в разделе **Макросы** в области переходов. Это облегчает управление базой данных, поскольку отпадает необходимость отслеживать отдельные макрообъекты, содержащие макросы для формы или отчета. Используйте следующую процедуру для создания внедренного макроса.

Откройте форму или отчет, который будет содержать макрос в конструкторе или режиме макета. Чтобы открыть форму или отчет, щелкните правой кнопкой мыши в области навигации и нажмите кнопку **Режим конструктора** или **Режим макета** .

Если страница свойств еще не отображена, для ее отображения нажмите клавишу F4.

Щелкните элемент управления или раздел, содержащий свойство события, в которое следует внедрить макрос.

Чтобы выбрать всю форму или отчет, выберите **Отчет** в раскрывающемся списке вверху страницы свойств.

Откройте вкладку **Событие** на странице свойств.

Щелкните свойство события, в которое нужно внедрить макрос, и нажмите кнопку рядом с полем.

В диалоговом окне **Построитель** выделите пункт **Макросы** и затем нажмите кнопку **ОК**.

В построителе макросов щелкните первую строку в столбце **Макрокоманда**.

В раскрывающемся списке **Макрокоманда** выберите нужную макрокоманду.

Укажите в области **Аргументы макрокоманды** все требуемые аргументы и перейдите к следующей строке макрокоманды.

Повторяйте шаги 8 и 9, пока не завершите построение макроса.

Нажмите кнопку **Сохранить** , затем нажмите кнопку **Закрыть** .

Макрос будет выполняться каждый раз при возникновении события.

Примечание: Access позволяет создавать группу макросов в качестве внедренного макроса. Однако только первый макрос в группе выполняется при наступлении события. Последующие макросы, пропускаются, если они называются из внутри внедренного макроса сам (например, по **ПриОшибке)**.

Редактирование макроса

Вставка строки макрокоманды Щелкните строку макроса, выше которой нужно вставить новую макрокоманду, и на вкладке Конструктор в группе Строки нажмите кнопку Вставить строки Изображение кнопки.

Удаление строки макрокоманды Щелкните строку макрокоманды, которую нужно удалить, и на вкладке Конструктор в группе Строки нажмите кнопку Удалить строки Изображение кнопки.

Перемещение строки макрокоманды Выберите макрокоманду, щелкнув область выделения строки слева от имени макрокоманды. Перетащите

область выделения строки, чтобы переместить макрокоманду в другое положение.

Дополнительные сведения о макрокомандах

В построителе макросов можно получить дополнительные сведения о макрокоманде или аргументе, выделив нужный элемент и ознакомившись с описанием в поле, расположенном в правом нижнем углу окна построителя макросов. Кроме того, каждой макрокоманде посвящен отдельная статья справки. Для получения дополнительных сведений о макрокоманде щелкните макрокоманду в списке и затем нажмите клавишу F1.

Запуск макроса

Макросы автономного можно работать в любой из следующих способов: напрямую (например, в области навигации), в группу макрос из другого макроса в модуле VBA или в ответ на событие, которое должно состояться форме, отчете или элементе управления. Внедренный макрос в формы, отчета или элемента управления можно выполнить, щелкнув выполнить Изображение кнопки на вкладке " Конструктор ", пока макрос в режиме конструктора; в противном случае — макрос будет запускаться только при каждом запустившего его связанного события.

Непосредственный запуск макроса

Для непосредственного запуска макроса выберите один из следующих вариантов.

Найдите макрос в области переходов и дважды щелкните его имя.

На вкладке Инструменты базы данных в группе Макрос нажмите кнопку Выполнить макрос Изображение кнопки, выберите макрос в списке Имя макроса и нажмите кнопку ОК.

Если макрос открыт в режиме конструктора, нажмите кнопку запустить Изображение кнопки на вкладке " Конструктор " в группе " Сервис ". Чтобы

открыть макрос в режиме конструктора, щелкните правой кнопкой мыши в области навигации и нажмите кнопку Режим конструктора Изображение кнопки.

Запуск макроса, находящегося в группе макросов

Чтобы запустить макрос, находящийся в группе макросов, выполните одно из следующих действий.

На вкладке Инструменты базы данных в группе Макрос нажмите кнопку Выполнить макрос Изображение кнопки и выберите нужный макрос в списке Имя макроса.

В каждой группы макросов в формате ИмяГруппыМакросов.имяМакросаавтоматически включаются запись для каждого макроса.

Нажмите кнопку ОК.

Укажите макрос в качестве параметра свойство события в форме или отчете или в качестве аргумента имя макроса ЗапускМакроса . Ссылки на макрос с помощью следующего синтаксиса:

ИмяГруппыМакросов.ИмяМакроса

Например, для запуска макроса «Категории» в группе «Кнопки кнопочной формы» используется следующее значение свойства события:

Кнопки кнопочной формы.Категории

Выполнение макроса, который находится в группу макросов в пределах процедуры на языке VBA, используя метод ЗапускМакроса объекта DoCmd и с использованием синтаксиса приведенной выше, для ссылки на макрос.

Запуск макроса из другого макроса или из процедуры VBA

Добавьте ЗапускМакроса макрос или процедуру.

Чтобы добавить ЗапускМакроса макрос, выберите в списке действие в пустой строке макрокоманды ЗапускМакроса и задайте аргумент Имя макроса имя макроса, который необходимо выполнить.

Чтобы добавить ЗапускМакроса процедуры на языке VBA, добавьте в ЗапускМакроса метод объекта DoCmd процедуры, а затем укажите имя макроса, который необходимо выполнить. Например, следующий экземпляр метода ЗапускМакроса запускать макрос, именованный макрос Мои:

DoCmd.RunMacro "My Macro"

Запуск макроса в ответ на событие в форме, отчете или элементе управления

Несмотря на новую возможность внедрять макросы непосредственно в свойства событий в формах, отчетах и элементах управления можно, как и в предыдущих версиях Access, создавать изолированные макросы и затем связывать их с событиями.

После создания изолированного макроса откройте форму или отчет в режиме конструктора или макета.

Откройте окно свойств для формы, отчета, раздела или элемента управления в форме или отчете.

Откройте вкладку Событие.

Выберите свойство события для события, которое вы хотите запустить макрос. Например для запуска макроса на событие изменения, нажмите кнопку На изменение свойства.

Выберите имя изолированного макроса в раскрывающемся списке.

Сохраните форму или отчет.

Задание 1: Внедрение макроса в событие отчета «Отсутствие данных»

Если при запуске отчета в источнике данных отсутствуют записи, отображается пустая страница отчета — то есть страница, на которой отсутствуют данные. Иногда желательно, чтобы вместо пустой страницы отображалось окно сообщения. Идеальным решением в этой ситуации является использование внедренного макроса.

Откройте отчет в режиме конструктора или макета.

Если страница свойств еще не отображена, для ее отображения нажмите клавишу F4.

Откройте вкладку **Событие** на странице свойств.

Выберите событие **Отсутствие данных**.

Нажмите кнопку .

В диалоговом окне **Построитель** выделите пункт **Макросы** и затем нажмите кнопку **ОК**.

Введите макрокоманды и аргументы из следующей таблицы.

Макрокоманда	Аргументы
MsgBox	«Записи не обнаружены.»; «Да»; «Сведения»; «Нет данных»
ОтменитьСобытие	[аргументы отсутствуют]

Обратите внимание, что в предыдущей таблице аргументы представлены в том виде, в каком они отображаются в столбце **Аргументы**. Фактически они вводятся в разделе **Аргументы макрокоманды**, как показано в следующей таблице.

Аргумент макрокоманды	Значение
Сообщение	Записи не обнаружены.
Сигнал	Да
Тип	Сведения
Заголовок	Нет данных

Нажмите кнопку **Закрыть**.

Построитель макросов будет закрыт, а для события **Отсутствие данных** будет выведено сообщение **[Внедренный макрос]**.

Сохраните и закройте отчет.

Если при следующем запуске отчета записи не будут обнаружены, появится окно сообщения. При нажатии в этом окне кнопки **ОК** отчет будет отменен без отображения пустой страницы.

Контрольные вопросы:

4. С помощью каких инструментов создаются макросы в MS Access?
5. Как можно запускать макросы?
6. Какие типы макросов существуют в MS Access?

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №14

Тема: «Разработка меню пользователя»

Цель: Научиться разрабатывать пользовательские интерфейсы используя автоматизированные средства СУБД MS Access

Время выполнения: 4 урока

Ход работы:

Создание формы навигации в Access

Форма навигации — это простая форма, содержащая элемент навигации. Формы навигации являются полезным дополнением любой базы данных, но особенно важны для баз данных, публикуемых в Интернете, так как область навигации Access не отображается в браузере.

Откройте базу данных, в которую нужно добавить форму навигации.

На вкладке Создание в группе Формы нажмите кнопку Навигация и выберите нужный стиль формы навигации.

Access создаст форму, добавит на нее элемент навигации и отобразит форму в режиме макета.

Задание 1: Создать для разработанных форм в БД «Справочник кадров» кнопки «Добавить запись», «Дублировать запись», «Сохранить запись», «Найти запись», «Удалить запись», «Закрыть форму». Для этого необходимо открыть требуемую форму и переключиться в режим конструктора (в случае необходимости раздвинуть поле «Примечание»).

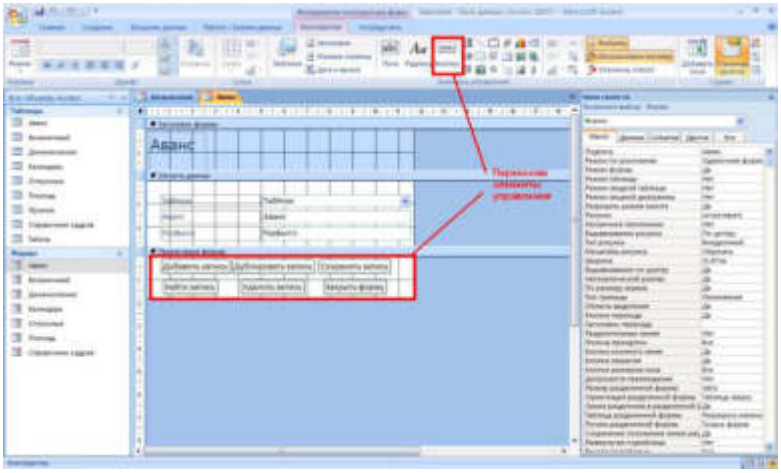


Рисунок 1 – Создание кнопок управления для разработанной формы

Задание 2: Создать главную форму для БД «Справочник кадров».

Контрольные вопросы:

1. С помощью каких инструментов создаются формы навигации в MS Access?

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №15

Тема: «Переменные, константы и типы данных в VBA»

Цель: Научиться объявлять и работать с переменными, константами в VBA, манипулировать и преобразовать типы данных.

Время выполнения: 4 урока

Ход работы:

Язык программирования VBA служит для написания кода программы. VBA имеет свой алфавит и синтаксис. Программный код VBA состоит из группы инструкций, записанных с учетом определенного алфавита и синтаксических правил. В алфавит VBA входят: прописные и строчные буквы латинского алфавита, прописные и строчные буквы кириллицы, цифры 0 - 9, символы подчеркивания "_", специальные символы и составные символы.

В исходном тексте программы можно использовать комментарии. Комментарии - это запись в строке после апострофа или ключевого слова REM. Комментарии - это пояснения в исходном тексте программы.

Программы, написанные на языке программирования VBA, оперируют данными определенного типа. Поэтому основными понятиями в VBA, как и других языках программирования являются типы данных. Тип данных определяет область допустимых значений из множества данных, набор операций, которые можно применять к этим значениям, правила выполнения операций и способ хранения (место хранения и объем данных) в памяти компьютера. Типы данных подразделяются на две категории: базовые (встроенные) и пользовательские (собственные). VBA позволяет создавать пользовательские типы данных.

В VBA для представления значений данных используются переменные и константы. Они используются для хранения временных значений и выполнения вычислений в ходе выполнения программы. С одной стороны (с технической точки зрения) переменная - это отдельный именованный

участок памяти для хранения данных определенного типа. С другой стороны (в тексте программы) переменная - это ее имя. Другими словами, переменная - это пара, включающая ячейку памяти и сопоставленное ей имя. Переменные – это объекты, которые предназначены для хранения данных. Значение переменной может быть изменено в процессе выполнения программы.

Константа - это также именованный участок памяти для хранения постоянных значений. Константы - это объекты, значения которых остаются постоянными и не могут быть изменены во время выполнения программы. Переменные и константы – это именованные хранилища данных.

Основными видами данных VBA являются константы и переменные. Тип данных указывает, в каком виде данные хранятся в памяти. Переменные предоставляют в программе значения или объекты. Прежде чем использовать переменную ее необходимо объявить (декларировать) с помощью оператора DIM (формат: DIM Имя_Переменной AS Тип_Данных).

Например, DIM MyType AS Integer. Для объявления какой-либо переменной необходимо ввести ключевое слово DIM. Это ключевое слово сообщает VBA, что объявляется переменная и резервируется область памяти для ее хранения. Далее вводим имя той переменной, которую надо объявить - MyType. Затем надо записать ключевое слово AS (как), которое сообщает, что определяется тип данных для переменной MyType. Далее объявим переменную MyType как целое число, и вводим тип данных Integer.

При объявлении констант используется ключевое слово Const. Одновременно с объявлением константы присваивается и ее значение.

К основным типам данных VBA относятся: числовые, символьные (строки), логические, дата и объектные.

Числовой тип данных

В VBA используется шесть различных численных типов данных (самый обширный тип данных): Byte, Integer, Long, Single, Double, Currency.

Целые типы:

Byte - целые числа (интервалы: 0 - 255, резервируется байт - 1 байт);

Integer - целые числа (-32768 - 32767, 2 байта);

Long - длинные целые числа (-2.1E9 - 2.1E9, 4 байта).

Плавающие типы:

Single - вещественные числа одинарной точности с плавающей точкой (интервалы: -3,402823E38 - -1,401298E-45 и 1,401298E-45 - 3,402823E38, 4 байта);

Double - вещественные числа двойной точности с плавающей точкой (-1,79769313486232E308 - -4,94065645841247E-324 и 4,94065645841247E-324 - 1,79769313486232E308, 8 байт).

Типы с фиксированной точкой Currency - числа, имеющие до 15 цифр до десятичной точки и четыре цифры после нее (-922337203685477,5808 - 922337203685477,5807).

Строковый тип данных

Строковый тип:

string фиксированной длины - текстовая информация в интервале от 1 до 65400, резервируется память длиной в 1 строку;

string переменной длины - текстовая информация в интервале от 0 до 2 млрд. символов, резервируется память длиной 10 байт + длина строки.

Прочие типы данных

Логический тип Boolean - логические значения (принимает одно из двух логических значений: True - False, 2 байта).

Date - информации о дате и времени (от 1 января 100 года до 31 декабря 9999 года, 8 байт).

Object - ссылка на объект типа формы, отчета или элемента управления (ссылка на объект, 4 байта). Объектные типы данных хранят информацию о каком-либо объекте со всеми свойствами и его возможностями.

Если для переменной, которая используется в процедуре VBA, не объявлен тип данных, то, по умолчанию, будет задан тип данных Variant.

Типы Variant

Variant (числа) - числовое значение (любое числовое значение, 16 байт);

Variant (символы) - строковое значение (любое символьное значение 22 байта + длина строки).

Далее рассмотрим более подробно область переменных в программах VBA.

Задание: Выполните следующие примеры в редакторе VBA, проанализируйте полученные результаты.

Пример 1.

```
Sub VariantDemo ()
MyVar = "123"
MyVar = MyVar / 2
MyVar = "Ответ: " & MyVar
MsgBox MyVar
End Sub
```

Пример 2. Какое сообщение выдаст следующая процедура?

```
Sub VariantDemo 2()
MyVar = "123"
MyVar = MyVar + MyVar
MyVar = "Ответ: " & MyVar
MsgBox MyVar
End Sub
```

Пример 3.

```
Sub VariantDemo2()
MyVar = "123"
MsgBox TypeName(MyVar)
MyVar = MyVar / 2
MsgBox TypeName(MyVar)
MyVar = "Ответ: " & MyVar
```

```
MsgBox TypeName(MyVar)
MsgBox MyVar
End Sub
```

Пример 4.

```
Sub TimeTestU
Dim x As Integer, y As Integer
Dim A As Integer, B As Integer, C As Integer
Dim i As Integer, j As Integer
Dim StartTime As Date, EndTime As Date
' Сохранение времени начала вычислений
StartTime = Timer
' Выполнение вычислений
x = 0
y = 0
For i = 1 To 5000
For j = 1 To 1000
A = x + y + i
B = y - x - i
C = x - y - i
Next j
Next i
' Получение времени окончания вычислений
EndTime = Timer
' Отображение общего времени
MsgBox Format(EndTime – StartTime)
End Sub
```

Пример 5.

Сравните время выполнения программы из Примера 4 и время выполнения этой же программы при превращении операторов Dim в комментарии.

Контрольные вопросы:

1. Перечислите и опишите типы данных VBA.
2. Опишите и приведите пример данных которые могут храниться в переменных типа Variant.

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №16

Тема: «Процедуры в VBA»

Цель: Научиться создавать процедуры в MS Access средствами VBA для решения нестандартных задач.

Время выполнения: 4 урока

Ход работы:

VBA поддерживает два типа подпрограмм: процедуры и функции.

Функция — это подпрограмма, которая возвращает результат. Вызов функции является выражением, и может использоваться в других выражениях или в правой части оператора присваивания.

Процедура — это любая подпрограмма, которая не является функцией. Любой макрос VBA является подпрограммой типа "процедура".

Для объявления процедуры в VBA используется ключевое слово Sub:

```
Sub <имяПроцедуры> [(<списокПараметров>)]
    <операторы>
End Sub
```

где:

<имяПроцедуры> – любой допустимый идентификатор VBA;

<списокПараметров> – список формальных параметров процедуры, если он пуст, то такая процедура является макросом;

<операторы> - любая последовательность операторов VBA.

Листинг №.1 Пример объявления процедуры

```
' Процедура выводит в отладчик максимальное из трех чисел
Sub sMax3(A As Long, B As Long, C As Long)
    If (A > B) And (A > C) Then
        Debug.Print "Max is "; A
    ElseIf (B > A) And (B > C) Then
        Debug.Print "Max is "; B
    Else
```

```
            Debug.Print "Max is "; C
        End If
    End Sub
```

Синтаксис объявления функции несколько сложнее, чем синтаксис процедуры:

```
    Function <имяФункции> [(<списокПараметров>)] [As <типФункции>]
        <операторы>
        . . .
        <имяФункции> = <возвращаемое_значение>
        [<операторы>]
    End Function
```

где:

<имяФункции> - любой допустимый идентификатор;

<списокПараметров> – список формальных параметров процедуры;

<типФункции> - имя любого поддерживаемого VBA типа данных;

<операторы> - любая последовательность операторов VBA.

<возвращаемое_значение> - результат, передаваемый в вызывающую программу.

Листинг №2. Пример объявления функции

```
    ' Функция возвращает максимальное из трех чисел
    Function fMax3(A As Long, B As Long, C As Long) As Long
        If (A > B) And (A > C) Then
            fMax3 = A
        ElseIf (B > A) And (B > C) Then
            fMax3 = B
        Else
            fMax3 = C
        End If
    End Function
```

Передача параметров в подпрограммы

Подпрограммы VBA могут принимать для обработки формальные параметры, указываемые при объявлении. При вызове они заменяются фактическими параметрами, т.е. реально используемыми в вызывающей программе. В VBA список формальных параметров подпрограммы представляет имена переменных, разделенных запятой. При этом желательно указать тип каждой переменной:

Function | Sub <имяПроцедуры> (<параметр1> As <тип>, _ <параметр2> As <тип>, ..., <параметрN> As <тип>)

Если тип данных параметра не указан, то автоматически будет использован тип Variant. Список параметров может быть пустым как для процедуры, так и для функции. В этом случае после имени процедуры ставятся пустые круглые скобки.

При передаче фактических параметров в подпрограмму может использоваться один из двух различных способов:

- передача по значению;
- передача по ссылке.

При передаче по значению, в подпрограмме создается копия полученного фактического параметра. Все действия внутри подпрограммы выполняются над этой копией и при выходе из подпрограммы все изменения теряются. Если переменная передается по ссылке, то процедуре или функции будет передан адрес этой переменной. Тем самым вызываемая процедура может изменить значение фактического параметра: если будет измен формальный параметр процедуры, то переданный при вызове ей фактический параметр тоже изменит свое значение.

Способ передачи указывается при описании параметров в строке объявления подпрограммы. Имени параметра может предшествовать один из явных описателей способа передачи:

ByRef – задает передачу по ссылке;

ByVal – задает передачу по значению.

По умолчанию выполняется передача по ссылке.

Листниг 15. Передача параметров в подпрограмму

```
' sample10 - вызывающая программа (макрос)
' ByValByRefDemo - вызываемая процедура
Sub sample10()
    Dim a As Long, b As Long, c As Long ' фактические параметры
    a = 10
    b = 10
    c = 10
    ByValByRefDemo a, b, c ' передача фактических параметров
    Debug.Print "2: " & "a = " & a & "; b = " & b & "; c = " & c
End Sub
Sub ByValByRefDemo(x As Long, ByVal y As Long, ByRef z As Long)
    ' выполнение действий над формальными параметрами
    x = x * 2 y = y * 3 z = z * 4
    Debug.Print "1: " & "x = " & x & "; y = " & y & "; z = " & z
End Sub
```

Здесь объявлены две процедуры: sample10 и ByValByRefDemo. Процедура sample10 вызывает процедуру ByValByRefDemo и передает ей предварительно инициализированные переменные a, b и c. Процедура ByValByRefDemo получает значения переменных a, b и c в виде формальных параметров x, y и z соответственно, выполняет над ними указанные действия, выводит результат и завершается. После возврата из подпрограммы процедура sample10 выводит значения переменных a, b, c в окно отладчика (рис. 4).

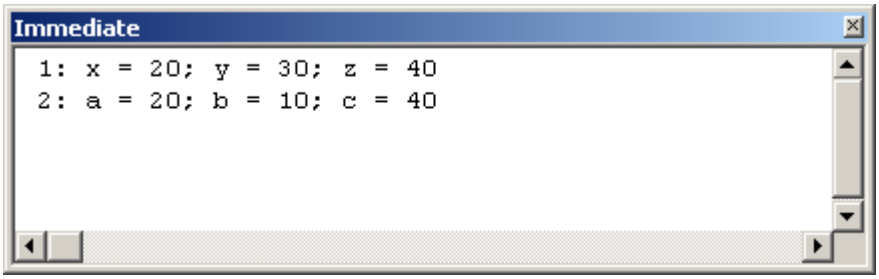


Рис. 1. Передача параметров в подпрограмму

Именованные параметры

При вызове подпрограмм в VBA параметры необходимо передавать в определенном порядке. Отсутствующие необязательные параметры отмечаются запятыми. Нарушение этого правила часто приводит к ошибкам – легко пропустить или поменять местами параметры. Чтобы избежать этих проблем, в VBA можно использовать именованные параметры функций. Для этого в вызове подпрограммы явно указываются имена параметров (как это было задано при объявлении подпрограммы), каждому из которых присваивается требуемое значение с помощью оператора ":=" ("двоеточие_равно"). При использовании именованных параметров можно не обозначать отсутствующие параметры и, кроме того, порядок перечисления параметров может быть произвольным.

Следующий пример показывает два обращения к функции MsgBox, которые имеют один и тот же результат:

```
' обычный вызов
MsgBox "Здравствуй, мир!", , "Окно приветствия"
' вызов, с использованием именованных параметров
MsgBox Prompt:= "Здравствуй, мир!", Title:= "Окно приветствия"
```

Вызов подпрограмм

Подпрограммы из программ на VBA могут быть вызваны различными способами:

Процедуру (Sub)с непустым списком параметров можно вызвать только из другой процедуры или функции так:

<имяПроцедуры> <списокПараметров>

Или так:

Call <имяПроцедуры> (<списокПараметров>) ’использована инструкция Call.

Процедура с пустым списком параметров рассматривается VBA как командный макрос. Ее также можно вызвать двумя способами:

из другой процедуры или функции;

с помощью комбинации клавиш быстрого вызова, команд меню или кнопок панелей инструментов.

Функцию (Function) можно вызывать точно так же, как и процедуру, в виде отдельного оператора. В этом случае возвращаемое функцией значение игнорируется.

Листинг №16. Вызов процедуры

```
Sub sample11() ' вызывающий макрос
    Dim usr As String
    usr = InputBox("Login")
    Hello usr ' вызов процедуры Hellobез Call
    Call Hello(usr) ' использование инструкции Call
End Sub
' процедура принимает один параметр, формирует строку сообщения,
' выводит сообщение в окно отладчика
Sub Hello(usrname As String)
    Debug.Print "Hello, " & usrname & "!"
End Sub
```

Если в проекте используется несколько подпрограмм с одинаковыми названиями (это возможно, если они в разных модулях), то при их вызове перед именем подпрограммы надо указывать (через точку) имя модуля, в котором процедура расположена:

<имяМодуля>.<имяПроцедуры> <списокФактическихПараметров>

Например

```
MyModule.MySub fArg, sArg ' вызов процедуры из модуля MyModule
```

Для вызова общих (Public) подпрограмм из другого проекта дополнительно к именам модуля и подпрограммы указывается имя проекта:

<имяПроекта>.<имяМодуля>.<имяПроцедуры>
<списокФактическихПараметров>

Например

someVal = MyProject.MyModule.MyFunc(fArg, sArg) ' вызов функции из проекта MyProject

Прерывание подпрограммы

В случае необходимости, выполнение процедуры или функции может быть прервано досрочно. Для этого нужно использовать инструкцию прерывания Exit Sub. В этом случае синтаксис объявления примет следующий вид (на примере объявления процедуры):

```
Sub <имяПроцедуры> [(<списокпараметров>)]
    <операторы>
    Exit Sub
    <операторы>
End Sub
```

Задание: Выполнить программы из приведенных выше листингов в редакторе VBA. Проанализировать ход их выполнения и полученный результат.

Контрольные вопросы:

1. Что такое процедура в VBA?
2. Что такое функция в VBA?
3. Чем отличается функция от процедуры?

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №17

Тема: «Функции в VBA»

Цель: Научиться создавать, вызывать и передавать значения функциям в VBA

Время выполнения: 4 урока

Ход работы:

Функция — это подпрограмма, которая возвращает результат. Вызов функции является выражением, и может использоваться в других выражениях или в правой части оператора присваивания.

Синтаксис объявления функции:

Function <имяФункции> [(<списокПараметров>)] [As <типФункции>]

    <операторы>

    . . .

    <имяФункции> = <возвращаемое_значение>

    [<операторы>]

End Function

где:

<имяФункции> - любой допустимый идентификатор;

<списокПараметров> – список формальных параметров процедуры;

<типФункции> - имя любого поддерживаемого VBA типа данных;

<операторы> - любая последовательность операторов VBA.

<возвращаемое_значение> - результат, передаваемый в вызывающую программу.

Листинг №2. Пример объявления функции

```
' Функция возвращает максимальное из трех чисел
Function fMax3(A As Long, B As Long, C As Long) As Long
    If (A > B) And (A > C) Then
        fMax3 = A
    ElseIf (B > A) And (B > C) Then
```

```
            fMax3 = B
        Else
            fMax3 = C
        End If
    End Function
```

Функцию (Function) можно вызывать точно так же, как и процедуру, в виде отдельного оператора. В этом случае возвращаемое функцией значение игнорируется.

Листинг №16. Вызов процедуры

```
Sub sample11() ' вызывающий макрос
    Dim usr As String
    usr = InputBox("Login")
    Hello usr ' вызов процедуры Hellobез Call
    Call Hello(usr) ' использование инструкции Call
End Sub
' процедура принимает один параметр, формирует строку сообщения,
' выводит сообщение в окно отладчика
Sub Hello(usrname As String)
    Debug.Print "Hello, " & usrname & "!"
End Sub
```

Если в проекте используется несколько подпрограмм с одинаковыми названиями (это возможно, если они в разных модулях), то при их вызове перед именем подпрограммы надо указывать (через точку) имя модуля, в котором процедура расположена:

<имяМодуля>.<имяПроцедуры> <списокФактическихПараметров>

Например

```
MyModule.MySub fArg, sArg ' вызов процедуры из модуля MyModule
```

Для вызова общих (Public) подпрограмм из другого проекта дополнительно к именам модуля и подпрограммы указывается имя проекта:

<имяПроекта>.<имяМодуля>.<имяПроцедуры>
<списокФактическихПараметров>

Например

someVal = MyProject.MyModule.MyFunc(fArg, sArg) ' вызов функции из проекта MyProject

Задание: Выполнить программы из приведенных выше листингов в редакторе VBA. Проанализировать ход их выполнения и полученный результат.

Контрольные вопросы:

1. Что такое функции в VBA?
2. Как вызвать функцию в виде отдельного оператора?

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №18

Тема: «Область действия переменных и процедур в VBA»

Цель: Познакомиться с областью действия переменных и процедур в VBA. Усвоить различия между Public и Private/

Время выполнения: 2 урока

Ход работы:

Область действия переменных

Область действия переменной определяет, в каких модулях и процедурах она может использоваться. Существуют следующие типы областей действия переменных.

Область действия	**Способ объявления переменной**
Отдельная процедура	В процедуру включается оператор Dim или static
Отдельный модуль	Перед первой процедурой в модуле вводится оператор Dim или private
Все модули	Перед первой процедурой в модуле вводится оператор Public

Локальные переменные

Локальная переменная — это переменная, объявленная в процедуре.

Локальные переменные могут использоваться только в процедуре, в которой они объявлены.

После выполнения процедуры переменная становится невостребованной, поэтому Excel освобождает соответствующую область памяти.

Если требуется сохранить значение переменной, объявите ее как static.

Чтобы объявить локальную переменную — вставьте оператор Dim между операторами Sub и End Sub .

Dim – сокращение от *Dimension (Размерность).* В старых версиях BASIC этот оператор использовался исключительно для объявления размерности массива.

Другой способ указания типа данных для переменной : язык VBA позволяет присоединить символ к названию, чтобы указать ее тип данных. Пример, можно объявить переменную MyVar как целое число, добавив к ее названию символ %: Dim MyVar %

Символы объявления типов данных представлены для большинства типов данных VBA (отсутствующие в таблице типы данных не имеют собственного символа объявления типа).

Тип данных	**Символ объявления типа**
Integer	%
Long	&
Single	!
Double	#
Currency	@
String	S

• Локальные переменные позволяют экономно использовать память, так как VBA освобождает память, которую они используют, после окончания выполнения процедуры.

Переменные уровня модуля

Иногда необходимо, чтобы переменная была доступна во всех процедурах модуля. В таком случае объявите переменную *перед* первой процедурой модуля (за пределами процедур или функций).

В приведенном ниже примере оператор Dim — первая инструкция в модуле. Обе процедуры MySub и YourSub имеют доступ к переменной CurrentValue .

Dim CurrentValue As Integer

Sub MySub{)

' -[Здесь вводится текст процедуры] -

End SUb

Sub YourSub()

'-[Здесь вводится текст процедуры] -

End Sub

Значение переменной уровня модуля не изменяется при окончании выполнения процедуры.

Переменные Public

Чтобы сделать переменную доступной во всех процедурах всех модулей VBA в проекте, необходимо объявить переменную на уровне модуля с помощью ключевого слова Public перед первой процедурой модуля, например, так:

Public CurrentRate as Long

Код объявления переменных Public должен вводиться в стандартном модуле VB A, а не в коде модуля листа или формы.

Переменные Static

Переменные Static — особый случай. Они объявляются на уровне процедуры и сохраняют свое значение после окончания процедуры.

Sub MySub()

Static Counter As Integer

'-[Здесь вводится текст процедуры] –

End Sub

Работа с константами

Константа – именованное значение или строка, которая не меняется при выполнении программы.

Объявление констант

Константы объявляются с помощью оператора Const .

Примеры :

Const NumQuarters as Integer = 4

Const Rate = .0725, Period = 12

Const ModName as String = "Budget Macros"

Public Const AppName as String = "Budget Application"

Во втором примере тип данных не объявлен. Следовательно, указанные две константы имеют тип Variant .

Константы имеют область действия как и переменные.

Область действия	**Способ объявления константы**
Отдельная процедура	В процедуре или функции
Отдельный модуль	Перед первой процедурой в модуле
Все модули	Перед первой процедурой в модуле с ключевым словом Public

При попытке изменить значение константы в процедуре VBA вы получите ошибку.

Использование предопределенных констант

В Excel и VBA существует целый ряд предопределенных констант, которые можно использовать без объявления.

В следующей процедуре для изменения ориентации страницы активного листа на альбомную применена встроенная константа (xlLandscape):

```
Sub SetToLandscape()
ActiveSheet.PageSetup.Orientation = xlLandscape
End Sub
```

Константу xlLandscape можно обнаружить путем записи макроса.

Описание констант можно найти в справочной системе.

Если включен параметр AutoList Members , то можно получить помощь непосредственно при вводе кода. Во многих случаях VBA автоматически перечисляет все константы, присваиваемые определенному свойству.

Задание: Выполнить программы из приведенных выше примеров в редакторе VBA. Проанализировать ход их выполнения и полученный результат.

Контрольные вопросы:

1. Что такое область видимости переменных?

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №19

Тема: «Ветвления в VBA»

Цель: Познакомиться на практике с командами ветвления в VBA и их различиями.

Время выполнения: 4 урока

Ход работы:

Ветвление - это управляющая операция языка программирования, позволяющая изменить порядок выполнения алгоритма программы.

Для реализации ветвлений в программах на VBA используются следующие операторы:

- условный оператор If;
- оператор выбора Select Case;
- оператор безусловного перехода Goto.

Условный оператор If

Условный оператор – это структура, которая выбирает ту или иную ветвь кода процедуры на основе некоторого предопределенного условия или группы условий. Общий синтакис условного оператора:

```
If <условие1> Then
        <Блок операторов1>
[ElseIf <условие2> Then
        <Блок операторов2>
[ElseIf <условиеN> Then
        <Блок операторовN>]
[Else
        <Блок операторов_Else>]]
End If
```

где:

<условиеN> - проверяемое условное выражение;

<Блок операторовN>; - операторы, выполняемые при истинности условия;

<Блок операторов_Else> - операторы, выполняемые по умолчанию.

Блоки ElseIf и Else - необязательны.

Помимо приведенной полной формы, поддерживается синтаксис краткой однострочной формы условного оператора следующего вида:

If <условие> Then <оператор>

Примеры использования условного оператора приведены в листингах 3 и 4.

Листинг 3. Полная форма условного оператора

```
Sub sample4()
 Dim value As Long
 Const MSG = "Вы ввели число, "
 value = InputBox(prompt:="Введите число", Title:="Пример 4")

 If value = 0 Then
        MsgBox (MSG & "равное 0")
 ElseIf value > 0 Then
        MsgBox (MSG & "большее 0")
 Else
        MsgBox (MSG & "меньшее 0")
 End If
End Sub
```

Листинг 4. Краткая форма условного оператора

```
Sub sample5()
Dim value As Long
Const MSG = "Вы ввели число, "
value = InputBox(prompt:="Введите число", Title:="Пример 4")
If value = 0 Then MsgBox (MSG & "равное 0")
If value > 0 Then MsgBox (MSG & "большее 0")
```

```
If value < 0 Then MsgBox (MSG & "меньшее 0")
End Sub
```

Логические выражения могут быть построены с помощью логических функций Not, And, Or, Xor, Imp, Eqv (листинг 5).

Листинг 5. Составные логические выражения

```
'Программа запрашивает имя пользователя и пароль,
'проверяет введенную информацию и выводит сообщение.
Sub sample5()
    Const UNAME = "User"
    Const PASSWD = "TopSecret"
    login = InputBox(prompt:="Логин")
    pass = InputBox(prompt:="Пароль")

    If (login = UNAME) And (pass = PASSWD) Then
        MsgBox ("Добро пожаловать!")
    Else
        MsgBox ("Вы не авторизованы!")
    End If
End Sub
```

Оператор выбора Select Case

Оператор выбора применяется в том случае, если проверяемое условное выражение может принимать много значений. В такой ситуации запись кода получается более компактной и наглядной, чем при использовании условного оператора.

Формат оператора Select Case:

```
Select Case <проверяемоеВыражение>
    Case <списокЗначений1> <блокОператоров1>
    [Case <списокЗначений2> <блокОператоров2>]
    [. . .]
    [Case Else <блокОператоров_Else>]
```

End Select

Проверяемое выражение вычисляется в начале работы оператора Select Case. *СписокЗначений* – это одно или несколько выражений, разделенных запятой. При выполнении оператора проверяется, соответствует ли хотя бы один из элементов этого списка проверяемому выражению. Эти элементы списка значений могут иметь одну из трех форм:

<выражение> - в этом случае проверяется, совпадает ли значение *проверяемогоВыражения* с этим выражением.

<выражение1> To <выражение2> - проверяется, находится ли значение *проверяемогоВыражения*в указанном диапазоне значений.

Is <логическаяОперация> <выражение> - *проверяемое выражение* сравнивается с указанным значением с помощью заданной логической операции (или операции отношения).

Отыскивается только первый подходящий элемент списков выражений.

Пример использования оператора Select Case приведен в листинге 6.

Листинг 6. Оператор выбора

```
'Процедура выводит на экран окно сообщения,
'содержащее командные кнопки Yes, No и Cancel; затем определяет
'выбранную пользователем кнопку и выводит сообщение,
'поясняющее этот выбор.
Sub sample7()
    Const mTitle = "Демонстрация кнопок MsgBox"
    Dim Resp As Integer
    Resp = MsgBox(prompt:="Выберите кнопку", Title:=mTitle, _
Buttons:=vbYesNoCancel + vbQuestion)
    Select Case Resp
    Case Is = vbYes MsgBox prompt:="Вы выбрали кнопку 'Да'", 
Title:=mTitle, _ Buttons:=vbInformation
    Case Is = vbNo MsgBox prompt:="Вы выбрали кнопку 'Нет'", 
Title:=mTitle, _ Buttons:=vbInformation
```

```
        Case Is = vbCancel MsgBox prompt:="Вы выбрали кнопку 'Отмена'", Title:=mTitle, _ Buttons:=vbCritical
            End Select
    End Sub
```

Оператор безусловного перехода GoTo

Оператор безусловного перехода GoTo всегда изменяет порядок выполнения операторов в процедуре или функции VBA без проверки каких-либо условий. Синтаксис оператора GoTo:

GoTo <метка>

где:

<Метка> – это любая допустимая метка в той же процедуре или функции, которая содержит оператор GoTo.

Листинг 7. Использование GoTo

```
'Программа будет запрашивать ввод значения пока не будет введено четное число
    Sub sample6()
        getValue: ' метка
        value = InputBox(prompt:="Введите четное число")
        If (value Mod 2 <> 0) Then GoTo getValue
    End Sub
```

Для иллюстрации решения задачи из листинга 7 приведем один из альтернативных способов, с помощью оператора цикла (листинг 8).

Листинг 8. Отказ от использования GoTo.

```
    Sub sample7()
        Do
            value = InputBox(prompt:="Введите четное число")
        Loop While value Mod 2 <> 0
    End Sub
```

Задание: Выполнить программы из приведенных выше листингов в редакторе VBA. Проанализировать ход их выполнения и полученный результат.

Контрольные вопросы:

1. Что такое ветвления в VBA?
2. Какие операторы ветвления в VBA вы знаете?

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №20

Тема: «Циклы в VBA»

Цель: Познакомиться на практике с операторами циклов в VBA и их различиями.

Время выполнения: 2 урока

Ход работы:

Операторы цикла используются для повторения выполнения действия или группы действий заданное количество раз. Количество повторений (итераций цикла) может быть предопределено или вычислено.

VBA поддерживает циклические конструкции двух видов:

- Циклы с фиксированным числом повторений (*циклы со счетчиком*).
- Циклы с неопределенным числом повторений (*циклы с условием*).

Для всех видов циклов используется понятие *тело цикла*, определяющее блок операторов, заключенных между начальным и конечным операторами цикла. Каждое повторение выполнения операторов тела цикла называется *итерация*.

Фиксированные циклы

VBA предоставляет две управляющие структуры для организации фиксированного цикла: For … Next (цикл со счетчиком) и For Each … Next (цикл с перечислением).

Оператор For … Next это типовой цикл со счетчиком, выполняющий заданное число итераций. Синтаксис оператора For … Next:

For <счетчик> = <начЗначение> To <конЗначение> [Step <приращение>]

    <блок операторов>

Next [<счетчик>]

Пример использования оператора For ... Next.

Листинг 9. Оператор For … Next

```
' ЗАДАЧА: Составить программу, которая получает два числа от пользователя.
' Складывает все числа в диапазоне, заданном этими двумя числами, а затем
' отображает результирующую сумму.

Sub sample7()
    Dim i As Integer ‘счетчик цикла
    Dim sStart ‘начальное значение счетчика
    Dim sEnd ‘конечное значение счетчика
    Dim sSum As Long ‘результирующая сумма

    sStart = InputBox("Введите первое число:")
    sEnd = InputBox("Введите второе число:")
    sSum = 0

    For i = CInt(sStart) To CInt(sEnd)
        sSum = sSum + i
    Next i

    MsgBox "Сумма чисел от " & sStart & " до " & sEnd & " равна: " & sSum
End Sub
```

Оператор цикла For Each … Next относится к категории операторов объектного типа, т.е. применяется в первую очередь к коллекциям объектов, а также к массивам. Тело цикла выполняется фиксированное число раз, соответствующее числу элементов массива или коллекции. Формат оператора For Each … Next:

For Each <элемент> In <группа> <блок операторов> Next [<элемент>]

Циклы с условием (неопределенные циклы)

Циклы с условием используются в тех случаях, когда повторяющиеся действия нужно выполнять только при определенных условиях. Количество итераций не определено и в общем случае может быть равно нулю (в частности, для циклов с предусловием). VBA предлагает разработчикам несколько управляющих структур для организации циклов с условием:

Четыре вида циклов Do..Loop, которые различаются типом проверяемого условия и временем выполнения этой проверки.

Непрерываемый цикл While … Wend.

Цикл Do While … Loop - типичный *цикл с предусловием*. Условие проверяется до того, как выполняется тело цикла. Цикл продолжает свою работу, пока это <условие> выполняется (т.е. имеет значение True). Так как проверка выполняется в начале, то тело цикла может ни разу не выполниться. Формат цикла Do While … Loop:

```
Do While <условие>
    <блок операторов>
Loop
```

Листинг 10. Цикл Do While … Loop

```
' ЗАДАЧА: Составить программу, которая предусматривает ввод пользователем
' произвольной последовательности чисел. Ввод должен быть прекращен
' только после того, как сумма введенных нечетных чисел превысит 100.
Sub sample8()
    Dim OddSum As Integer ‘сумма нечетных чисел
    Dim OddStr As String ‘строка с нечетными числами

    Dim Num ‘для приема вводимых чисел
    OddStr = "" ‘инициализация выходной строки
    OddSum = 0 ‘инициализация суммы OddSum
```

```
        Do While OddSum < 100 ‘начало цикла
            Num = InputBox("Введите число: ")
            If (Num Mod 2) <> 0 Then ‘проверка на четность
                OddSum = OddSum + Num ‘накопление суммы нечетных чисел
                OddStr = OddStr & Num & " "
            End If
        Loop
        'вывод строки с нечетными числами
        MsgBox prompt:="Нечетные числа: " & OddStr
    End Sub
```

Оператор Do … Loop While предназначен для организации *цикла с постусловием*. Условие проверяется после того, как тело цикла, будет выполнено хотя бы один раз. Цикл продолжает свою работу, пока <условие> остается истинным. Формат цикла Do … Loop While:

Do <блок операторов> Loop While<условие>

Листинг 11. Цикл с постусловием

```
' ЗАДАЧА: Составить программу игры "Угадай число". Программа должна случайным
' образом генерировать число в диапазоне от 1 до 1000, пользователь должен
' угадать это число. Программа на каждое вводимое число выводит подсказку
' "больше" или "меньше".
Sub sample8()
Randomize Timer ' инициализация генератора случайных чисел
    Dim msg As String ' строка сообщения
    Dim SecretNumber As Long, UserNumber As Variant
```

```
Begin:      SecretNumber = Round(Rnd * 1000) ' число, сгенерированное компьютером
            UserNumber = Empty ' число, вводимое пользователем
            Do ' игровой процесс
                Select Case True
                    Case IsEmpty(UserNumber): msg = "Введите число"
                    Case UserNumber > SecretNumber: msg = "Слишком много!"
                    Case UserNumber < SecretNumber: msg = "Слишком мало!"
                End Select
                UserNumber = InputBox(prompt:=msg, Title:="Угадай число")
            Loop While UserNumber <> SecretNumber
            ' проверка
            If MsgBox("Играть еще? ", vbYesNo + vbQuestion, "Вы угадали!") = vbYes Then
                GoTo Begin
            End If
        End Sub
```

Циклы Do Until … Loop и Do … Loop Until являются инверсиями ранее рассмотренных циклов с условием. В общем случае они работают аналогично, за исключением того, что тело цикла выполняется при ложном условии (т.е. <условие>=False). Формат цикла Do Until … Loop:

Do Until <условие> <блок операторов> Loop

Формат цикла Do … Loop Until:

```
Do
    <блок операторов>
Loop Until<условие>
```

Практическое задание: Перепишите программы из листингов 10 и 11 с использованием инвертированных операторов цикла.

Цикл While … Wend также относится к циклам с условием. Данный оператор полностью соответствует структуре Do While … Loop. Формат цикла While … Wend:

While <условие>

    <блок операторов>

Wend

Отличительной особенностью этого оператора является невозможность принудительного завершения (прерывания) тела цикла (оператор Exit Do не работает в цикле While … Wend).

Контрольные вопросы:

1. Что такое циклы в VBA?
2. Какие циклы вы знаете? Приведите примеры использования циклов.

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №21

Тема: «Выход из циклов и процедур в VBA»

Цель: Познакомиться на практике с командами досрочного завершения выполнения циклов и процедур в VBA и их различиями.

Время выполнения: 2 урока

Ход работы:

Прерывание цикла

Для досрочного завершения итерации и выхода из цикла применяется оператор Exit. Этот оператор применим в любой циклической структуре, кроме While ... Wend. Общий синтаксис использования Exit для прерывания цикла таков:

```
<начало_цикла>
    [<блок операторов1>]
    Exit (For | Do)
    [<блок операторов2>]
    [Exit (For | Do)]
    ...
<конец_цикла>
```

При выполнении оператора Exit цикл прерывается, и управление передается оператору, следующему за оператором <конец_цикла>. В теле цикла может присутствовать несколько операторов Exit.

Принудительный выход из цикла

```
Sub sample9()
    For i = 1 To 10000000
        If i = 10 Then Exit For ' выход из цикла, когда счетчик достигнет 10
    Next
End Sub
```

Прерывание подпрограммы

В случае необходимости, выполнение процедуры или функции может быть прервано досрочно. Для этого нужно использовать инструкцию прерывания Exit Sub. В этом случае синтаксис объявления примет следующий вид (на примере объявления процедуры):

```
Sub <имяПроцедуры> [(<списокпараметров>)]
    <операторы>
    Exit Sub
    <операторы>
End Sub
```

Оператор безусловного перехода GoTo

Оператор безусловного перехода GoTo всегда изменяет порядок выполнения операторов в процедуре или функции VBA без проверки каких-либо условий. Синтаксис оператора GoTo:

```
GoTo <метка>
```

где:

<Метка> – это любая допустимая метка в той же процедуре или функции, которая содержит оператор GoTo.

Листинг 7. Использование GoTo

```
'Программа будет запрашивать ввод значения пока не будет введено четное число
Sub sample6()
    getValue: ' метка
    value = InputBox(prompt:="Введите четное число")
    If (value Mod 2 <> 0) Then GoTo getValue
End Sub
```

Для иллюстрации решения задачи из листинга 7 приведем один из альтернативных способов, с помощью оператора цикла (листинг 8).

Листинг 8. Отказ от использования GoTo.

```
Sub sample7()
```

```
    Do
        value = InputBox(prompt:="Введите четное число")
    Loop While value Mod 2 <> 0
End Sub
```

Задание: Выполнить программы из приведенных выше примеров в редакторе VBA. Проанализировать ход их выполнения и полученный результат.

Контрольные вопросы:

1. Когда используется выход из циклов, процедур? Приведите примеры.

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №22

Тема: «Модули в VBA»

Цель: Научиться на практике создавать и редактировать модули в VBA.

Время выполнения: 4 урока

Ход работы:

Любая программа на VBA представлена в виде проекта. Проект - это совокупность программных модулей различных типов. В свою очередь модуль - это основная программная единица уровня проекта, в которой размещаются определения переменных, констант, типов, подпрограмм и т.д.

Visual Basic поддерживает три типа модулей, которые могут быть включены в проект и использованы:

- Стандартные модули - это модули, в которых можно описать доступные во всем проекте процедуры.
- Модули классов содержат описание свойств, методов и событий пользовательских классов.
- Модули форм содержат процедуры обработки событий, генерируемых элементами управления в формах проекта.

Структура модуля VBA включает два неявных (т.е. не требующих специального описания) раздела: общий (General) и объявлений (Declarations). В общем разделе задаются параметры среды (Option Base, Option Explicit), приводятся описания глобальных переменных, констант и типов. Раздел объявлений предназначен для описания процедур и функций.

Области видимости

VBA поддерживает две области видимости для переменных и подпрограмм: локальную и глобальную.

Локальные переменные определены на уровне подпрограммы с помощью ключевых слов Dim или Static. Они доступны только внутри этой

подпрограммы и по выходу из нее уничтожаются. Глобальные переменные объявляются на уровне модуля. Такие переменные доступны:

для всех подпрограмм модуля, в котором они объявлены (при объявлении с ключевым словом Private, либо Dim);

для всего приложения – при объявлении с ключевым словом Public;

Подпрограммы VBA могут быть объявлены на двух уровнях – уровне проекта (Public) и уровне модуля (Private). Например:

Public Sub Query(price, count) ' видимость на уровне проекта

Private Sub Sub Query(price, count) ' видимость на уровне модуля

По умолчанию используется уровень проекта.

При описании локальных переменных можно использовать ключевое слово Static. Такие переменные являются статическими и сохраняют значения между вызовами.

Встроенные функции VBA

В языке программирования VBA предусмотрено несколько десятков *встроенных функций*. Они доступны в любой программе на языке VBA, при этом безразлично, в среде какого программного продукта мы находимся — Excel, Word, Access или, к примеру, AutoCAD. Используются они очень активно, и во многих ситуациях без них не обойтись. Встроенные функции обычно группируют по назначению: математические, строковые, преобразования типов, логические и т.п. В справке по VBA имеется подробная информация о всех встроенных функциях. Здесь же приведем краткое описание только некоторых.

Функции приведения типов

Используются для конвертации типов данных. Вот перечень этих функций: *CBool(), CByte(), CCur(), CDate(), CDbl(), CDec(), CInt(), CLng(), CSng(), CStr(), CVar(), CVDate(), CVErr()*.

Просмотреть, что в итоге получилось, можно при помощи функции *TypeName()*, например:

nVar1 = CInt(InputBox("Введите значение"))

MsgBox *TypeName(nVar1)*

Кроме того, еще несколько полезных для конвертации функций:

Str() — позволяет перевести числовое значение в строковое, при этом вставляет пробел впереди для положительных чисел.

Val() — извлекает из строки числовое значение (до первого нечислового символа, кроме точки). Очень удобно, когда вперемежку с числовыми данными прописываются единицы измерения или валюта.

Чтобы не возникло ошибок при конвертации, можно вначале проверять значения на возможность конвертации при помощи функций *IsNumeric()* и *IsDate()*. Для проверки на соответствие специальным значениям можно использовать функции *IsArray()*, *IsEmpty()*, *IsError()*, *IsMissing()*, *IsNull()* и *IsObject()*. Все эти функции возвращают True или False в зависимости от результатов проверки переданного им значения.

Строковые функции

Left(), *Right()*, *Mid()* — получить фрагмент строки слева, справа или из середины исходной строки соответственно.

Len() — получить число символов в строке.

LCase() и *UCase()* — перевести строку в нижний и верхний регистры соответственно.

LSet() и *RSet()* — заполнить строку символами без изменения длины (соответственно слева и справа). Лишние символы обрезаются, на место недостающих подставляются пробелы.

LTrim(), *RTrim()*, *Trim()* — убрать пробелы соответственно слева, справа или и слева, и справа.

Replace() — заменить в строке одну последовательность символов на другую.

StrComp() — сравнить две строки.

StrReverse() — "перевернуть" строку, разместив ее символы в обратном порядке.

Математические функции

Функций для работы с числовыми значениями в VBA очень много. Приведем только некоторые из них.

ABS() — эта функция возвращает абсолютное значение переданного ей числа.

Int(), *Fix()* и *Round()* позволяют по разному округлять числа: Int возвращает ближайшее меньшее целое, Fix() отбрасывает дробную часть, Round() округляет до указанного количества знаков после запятой.

Rnd() и команда *Randomize* используются для получения случайных значений. Обычный синтаксис при применении Rnd выглядит так:

случайное_число = Int(минимум + (Rnd()* максимум))

*MsgBox(Int(1 + (Rnd() * 100)))*

Перед вызовом функции *Rnd()* следует выполнить команду Randomize для инициализации генератора случайных чисел.

Sgn() — позволяет вернуть информацию о знаке числа. Возвращает 1, если число положительное, -1, если отрицательное и 0, если проверяемое число равно 0.

Функции для работы с датой и временем

Основные функции VBA для работы с датой/временем:

Date() — возвращает текущую системную дату.

Time() возвращает текущее системное время, а *Now()* — дату и время вместе.

DateAdd() — возможность добавить к дате указанное количество лет, кварталов, месяцев и так далее — вплоть до секунд.

DateDiff() — возможность получить разницу между датами.

DatePart() — возвращает указанную часть даты (например, только год, только месяц или только день недели).

DateSerial() — формирует значение даты на основе передаваемых символьных значений. То же самое делает *DateValue()*, отличия — в формате принимаемых значений. Аналогичным образом (для времени) работают *TimeSerial()* и *TimeValue()*.

Day() (а также *Year()*, *Month()*, *Weekday()*, *Hour()*, *Minute()*, *Second()*) — специализированные заменители функции DatePart(), которые возвращают нужную часть даты.

MonthName() — возвращает имя месяца словами по его номеру. Возвращаемое значение зависит от региональных настроек. Если они русские, то вернется русское название месяца.

Timer() — возвращает количество секунд, прошедших с полуночи.

Функции взаимодействия с пользователем

Для организации диалога с пользователем VBA представляет две встроенные функции - MsgBox и InputBox. Окно сообщений MsgBox выводит сообщения для пользователя, а окно ввода InputBox обеспечивает возможность получения информации от пользователя.

Функция *MsgBox()* выводит на экран диалоговое окно, содержащее сообщение, устанавливает режим ожидания нажатия кнопки пользователем, а затем возвращает значение типа Integer, указывающее, какая кнопка была нажата.

Формат функции MsgBox:

MsgBox (Prompt [, Buttons] [, Title] [, HelpFile, Context])

Назначение параметров:

Prompt – строковое выражение, отображаемое как сообщение в диалоговом окне;

Title – строковое выражение, отображаемое в строке заголовка диалогового окна. Если этот параметр опущен, в строку заголовка помещается имя приложения, из которого запускается программа VBA;

HelpFile – строковое выражение, определяющее имя файла справки, содержащего справочные сведения о данном диалоговом окне; обычно это файл, который вы уже должны были создать с помощью Windows Help Compiler. Если этот параметр указан, необходимо также указать параметр Context;

Context – числовое выражение, определяющее номер соответствующего раздела справочной системы. Если этот параметр указан, необходимо указать также параметр HelpFile;

Buttons – числовое выражение, которое задает количество и тип кнопок в диалоговом окне MsgBox. Buttons указывает также кнопку по умолчанию в диалоговом окне и содержит ли это диалоговое окно стандартные значки Windows для предупредительных сообщений и запросов пользователя. Значения Buttons можно получить из справочной системы.

Простой пример использования функции MsgBox:

```
Sub sample3()
    MsgBox "Видите ли вы две кнопки?", vbYesNo + vbInformation, "Сообщение"
End Sub
```

В данном примере MsgBox применяется не в виде функции, а в виде процедуры (т.е. не возвращает никакого значения). Следовательно, код выбранной кнопки нигде не сохраняется и не может быть использован. Чтобы определить, какая кнопка была нажата, MsgBox необходимо вызвать как функцию, т.е. сохранить возвращаемое значение в переменную (Листинг 17.

Листинг 17. Использование MsgBox

```
Sub sample4()
    Dim res ' объявляем переменную
    ' вызываем MsgBox и сохраняем значение в переменной
    res = MsgBox("Видите ли вы две кнопки?", vbYesNo + vbInformation, "Сообщение")
    Debug.Print res ' печатаем полученное значение
End Sub
```

При выполнении этого макроса, когда пользователь выбирает кнопку Yes или No в переменной res сохраняется число, соответствующее выбранной кнопке.

Вместо возвращаемых функцией MsgBox целочисленных значений удобнее пользоваться предопределенными константами VBA. В таб. 11 приведены возвращаемые значения констант функции MsgBox.

Таблица 11. Возвращаемые значения функции MsgBox

Константа	Означает, что пользователь нажал кнопку
vbAbort	Стоп (Abort)
vbCancel	Отмена (Cancel)
vbIgnore	Пропустить (Ignore)
vbNo	Нет (No)
vbOk	Ок
vbRetry	Повтор (Retry)
vbYes	Да (Yes)

Дополним код листинга 17 проверкой возвращенного значения (листинг 18).

Листинг 18. Проверка возращаемого значения MsgBox

```
Sub sample5()
    ' вызываем MsgBox и сохраняем значение в переменной
    res = MsgBox("Видите ли вы две кнопки?", vbYesNo + vbInformation, "Сообщение")

    ' проверяем, какая кнопка нажата
    If res = vbYes Then : MsgBox "Вы нажали Yes", , "Результат выбора"
    Else : MsgBox "Вы нажали No", , "Результат выбора"
    End If
End Sub
```

Функция *InputBox()* выводит на экран диалоговое окно, содержащее сообщение и поле ввода, устанавливает режим ожидания ввода текста, а

затем возвращает значение типа String, содержащее текст, введенный в поле. Формат функции InputBox:

InputBox (Prompt [, Title] [, Default] [, XPos] [, Ypos] [, HelpFile, Context])

Назначение параметров:

Prompt – строковое выражение, отображаемое как сообщение в диалоговом окне;

Title – строковое выражение, отображаемое в строке заголовка диалогового окна. Если этот параметр опущен, в строку заголовка помещается имя приложения;

Default – строковое выражение, отображаемое в поле ввода и используемое по умолчанию, если пользователь не введет другую строку. Если этот параметр опущен, поле ввода изображается пустым;

Xpos и Ypos – числовые выражения, которые указывают местоположение окна ввода и являются координатами верхнего левого угла диалогового окна; Xpos – горизонтальное расстояние от левого края окна; Ypos – вертикальное расстояние от верхнего края окна. Если эти параметры опущены, диалоговое окно выравнивается по центру экрана;

HelpFile – строковое выражение, определяющее имя существующего файла справки, содержащего справочные сведения о данном диалоговом окне. Если этот параметр указан, необходимо также указать параметр Context;

Context – числовое выражение, определяющее номер соответствующего раздела справочной системы. Если этот параметр указан, необходимо указать также параметр HelpFile.

Приведем пример использования функции InputBox для получения имени пользователя.

```
Sub sample6()
    Dim username As String
    username = InputBox("Введите ваше имя ", "Пример 6")
```

```
    MsgBox ("Здравствуйте, "+username)
End Sub
```

В результате выполнения этого макроса на экран последовательно выводятся диалоговые окна ввода и вывода (рис. 1).

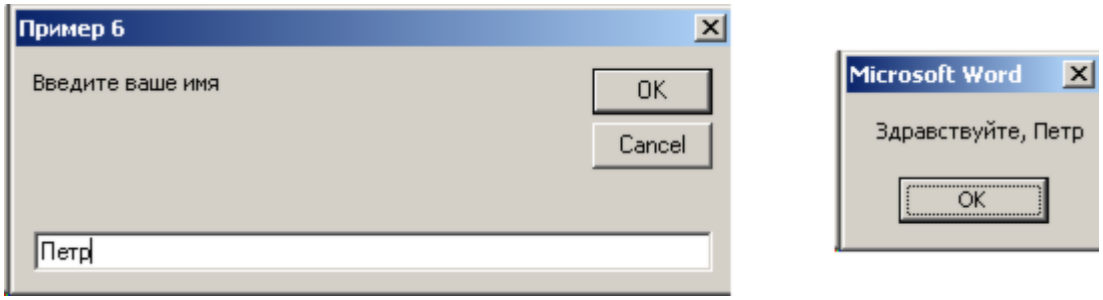


Рис. 1 Интерактивные функции VBA

Задание: Выполнить программы из приведенных выше примеров и листингов в редакторе VBA. Проанализировать ход их выполнения и полученный результат.

Контрольные вопросы:

1. Какие типы модулей VBA вы знаете?
2. Что такое область видимости модуля?

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №23

Тема: «Оператор CREATE»

Цель: Научиться создавать таблицы с помощью оператора CREATE языка SQL в MS Access.

Время выполнения: 2 урока

Ход работы:

Оператор SQL CREATE служит для создания объектов базы данных. Разные СУБД работают с разными объектами, но наиболее общими для большинства СУБД будут команды создания таблицы (оператор SQL CREATE TABLE) и создания представления (оператор SQL CREATE VIEW).

Синтаксис оператора SQL CREATE TABLE выглядит следующим образом:

```
CREATE TABLE  table_name (
column_name1 data_type(size),
column_name2 data_type(size),
column_name3 data_type(size)
...
)
```

Пример оператора SQL CREATE. Используя оператор SQL CREATE создать таблицу Planets содержащую следующие поля: ID (числовой), PlanetName (символьный), Radius (вещественный), SunSeason (вещественный), OpeningYear (числовой), HavingRings (логический), Opener (символьный):

```
CREATE TABLE Planets (
ID int,
PlanetName varchar(10),
Radius float (10),
SunSeason float(10),
OpeningYear int,
```

HavingRings bit,

Opener varchar(30)

)

В данном коде использовались типы СУБД MS SQL ACCESS. В других СУБД названия типов могут отличаться!

В результате выполнения запроса будет создана таблица с перечисленными полями и их типами.

Пояснение: Поле HavingRings имеет тип bit, принимающее 2 значения — 1 или 0 (TRUE, FALSE).

Задание: Создать в MS Access базу данных «Справочник кадров», а в ней следующие таблицы с помощью оператора SQL CREATE:

Таблица: Аванс

Имя поля	**Тип**	**Размер**
ТабНом (Поле со	Текстовый	5
Аванс	Числовой	Двойное с плавающей
РазВыпл	Числовой	Двойное с плавающей

Таблица: Больничный

Имя поля	**Тип**	**Размер**
ТабНом (поле со	Текстовый	5
ТипБол	Текстовый	1
ДатаНач	Дата/время	Краткий формат даты
ДатаКон	Дата/время	Краткий формат даты
Дни1	Числовой	Длинное целое
Дни2	Числовой	Длинное целое
Процент	Числовой	Двойное с плавающей

Таблица: Доначисления

Имя поля	**Тип**	**Размер**
ТабНом (поле со	Текстовый	5
Оклад	Числовой	Двойное с плавающей
Премия	Числовой	Двойное с плавающей
Доплата	Числовой	Двойное с плавающей
Совмещения	Числовой	Двойное с плавающей
ЗаСтаж	Числовой	Двойное с плавающей
Бол	Числовой	Двойное с плавающей
Отпуск	Числовой	Двойное с плавающей
КомпУв	Числовой	Двойное с плавающей
Налог	Числовой	Двойное с плавающей

Таблица: Календарь

Имя поля	**Тип**	**Размер**
Месяц	Числовой	Двойное с плавающей
Год	Числовой	Двойное с плавающей
РабДни	Числовой	Двойное с плавающей
Суб	Числовой	Двойное с плавающей
КалендДни	Числовой	Двойное с плавающей
ПраздДни	Числовой	Двойное с плавающей
Часы	Числовой	Двойное с плавающей
НазвМес	Текстовый	8

Таблица: Табель

Имя поля	**Тип**	**Размер**
ТабНом	Текстовый	5
ФИО	Текстовый	25
Катег	Текстовый	25
Месяц	Числовой	Длинное целое
Год	Числовой	Длинное целое
ОтрабДни	Числовой	Длинное целое
Часы	Числовой	Длинное целое
Ночные	Числовой	Длинное целое
Праздничные	Числовой	Длинное целое
Увольнение	Текстовый	1

Контрольные вопросы:

1. Для чего служит оператор SQL CREATE?
2. Как создать таблицу с помощью оператора SQL CREATE?

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №24

Тема: «Оператор INSERT»

Цель: Научиться добавлять записи в таблицы с помощью оператора INSERT языка SQL в MS Access.

Время выполнения: 2 урока

Ход работы:

Оператор SQL INSERT позволяет вносить изменения в структуру таблиц: добавлять записи (строки) и заполнять их значениями.

Оператор SQL INSERT имеет следующий синтаксис:

INSERT INTO table_name ([column_name, ...]) VALUES (expressions, ...)

Так же, значения могут быть записаны и без указания столбцов:

INSERT INTO table_name VALUES (expressions, ...)

Запись значений может производиться и с помощью оператора SQL SELECT:

INSERT INTO table_name SELECT column_name,... FROM table_name

Используя оператор SQL SELECT можно вставить более одной записи. В случае, если если для каких-либо полей таблицы не указаны значения, они будут заменены значением по умолчанию null.

Примеры оператора SQL INSERT. Имеется следующая таблица Planets:

ID	PlanetName	Radius	SunSeason	OpeningYear	HavingRings	Opener
1	Mars	3396	687	1659	No	Christiaan Huygens
2	Saturn	60268	10759.22	—	Yes	—
3	Neptune	24764	60190	1846	Yes	John Couch Adams

Пример 1. С помощью оператора SQL INSERT добавить запись, присвоив полям таблицы следующие значения — ID: 4; PlanetName: Venus; Radius: 6051; SunSeason: 243; OpeningYear: 1610; HavingRings: No; Opener: Galileo Galilei;

```
INSERT INTO Planets (ID, PlanetName, Radius, SunSeason, OpeningYear, Hav
VALUES (4, "Venus", 6051, 243, 160, "No", "Galileo Galilei")
```

Результат: После выполнения команды вывода всей таблицы:

```
SELECT * FROM Planets
```

Будет выведена таблица Planets с добавленной записью:

ID	PlanetName	Radius	SunSeason	OpeningYear	HavingRings	Opener
1	Mars	3396	687	1659	No	Christiaan Huygens
2	Saturn	60268	10759.22	—	Yes	—
3	Neptune	24764	60190	1846	Yes	John Couch Adams
4	Venus	6051	243	1610	No	Galileo Galilei

Пример 2. Имеется 2 таблицы — PlanetWithoutRings(Планеты без колец) и PlanetsWithRings (Планеты с кольцами). Как понятно из названия, они отличаются полем HavingRings. В первой таблице все значения в нем «No», во второй «Yes».

PlanetWithoutRings:

ID	PlanetName	Radius	SunSeason	OpeningYear	HavingRings	Opener
1	Mars	3396	687	1659	No	Christiaan Huygens
2	Mercury	2439	115.88	1631	No	Nicolaus Copernicus
3	Venus	6051	243	1610	No	Galileo Galilei

PlanetsWithRings:

ID	PlanetName	Radius	SunSeason	OpeningYear	HavingRings	Opener
1	Saturn	60268	10759.22	—	Yes	—
2	Neptune	24764	60190	1846	Yes	John Couch Adams

Задача: С помощью оператора SQL INSERT вставить записи из таблицы PlanetsWithRings в таблицу PlanetsWithoutRings.

Данную задачу можно решить двумя запросами SQL INSERT, как было показано в Примере 1. Однако, мы воспользуемся вставкой строк с помощью оператора SQL SELECT:

```
INSERT INTO PlanetsWithoutRings
SELECT ID, PlanetName, Radius, SunSeason, OpeningYear, HavingRings, Ope
FROM PlanetsWithRings
```

Так как названия столбцов в обоих таблицах совпадает, то указывать их в операторе SQL INSERT необязательно. Вышеописанный запрос возьмет все данные из указанных столбцов из одной таблицы и вставит в другую. Посмотреть результат можно запросом:

```
SELECT * FROM PlanetsWithoutRings
```

Результат:

ID	PlanetName	Radius	SunSeason	OpeningYear	HavingRings	Opener
1	Mars	3396	687	1659	No	Christiaan Huygens
2	Mercury	2439	115.88	1631	No	Nicolaus Copernicus
3	Venus	6051	243	1610	No	Galileo Galilei
1	Saturn	60268	10759.22	—	Yes	—
2	Neptune	24764	60190	1846	Yes	John Couch Adams

Задание: Выполнить запросы из приведенных выше примеров в редакторе запросов SQL MS ACCESS. Проанализировать ход их выполнения и полученный результат.

Контрольные вопросы:

1. Для чего служит оператор SQL INSERT? Приведите примеры использования.

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №25

Тема: «Оператор SELECT»

Цель: Научиться делать выборку из таблиц с помощью оператора SELECT языка SQL в MS Access.

Время выполнения: 2 урока

Ход работы:

Оператор SQL SELECT является одним из основных операторов языка SQL. Именно с его помощью происходит выборка значений, хранящихся в базе данных. В структуру запроса оператора SQL SELECT могут быть включены многие дополнительные операторы: уточняющие условие выборки, производящие группировку, сортировку выходных значений и т.д.

Оператор SQL SELECT имеет следующий синтаксис:

```
SELECT column_list
FROM table_name
[WHERE condition
GROUP BY expression
HAVING condition
ORDER BY expression]
```

Необязательные операторы обрамлены квадратными скобками [].

В параметре column_list указываются названия столбцов таблицы, которые необходимо вывести, либо символ “*”, позволяющий вывести все столбцы таблицы. Ключевым словом FROM задается название таблицы или таблиц, из которых следует брать столбцы. Оператор SQL WHERE задает дополнительные условия выборки. Оператор SQL GROUP BY используют для группирования результата по столбцу или по нескольким столбцам. Оператор SQL HAVING включают в запрос для задания условия агрегатных функций. Наконец, оператор SQL ORDER BY используется для сортировки значений.

Примеры оператора SQL SELECT. Имеется следующая таблица Planets:

ID	PlanetName	Radius	SunSeason	OpeningYear	HavingRings	Opener
1	Mars	3396	687	1659	No	Christiaan Huygens
2	Saturn	60268	10759.22	—	Yes	—
3	Neptune	24764	60190	1846	Yes	John Couch Adams
4	Mercury	2439	115.88	1631	No	Nicolaus Copernicus
5	Venus	6051	243	1610	No	Galileo Galilei

Пример 1. С помощью оператора SQL SELECT вывести названия планет (Name):

```
SELECT PlanetName FROM Planets
```

Результат:

PlanetName
Mars
Saturn
Neptune
Mercury
Venus

Пример 2. С помощью оператора SQL SELECT вывести названия планет, у которых есть кольца (HavingRings):

```
SELECT PlanetName FROM Planet WHERE HavingRings = 'Yes'
```

Результат:

PlanetName
Saturn
Neptune

Пример 3. С помощью оператора SQL SELECT вывести информацию о планете Нептун:

```
SELECT * FROM Planets WHERE PlanetName = 'Neptune'
```

Результат:

ID	PlanetName	Radius	SunSeason	OpeningYear	HavingRings	Opener
3	Neptune	24764	60190	1846	Yes	John Couch Adams

Задание: Выполнить запросы из приведенных выше примеров в редакторе запросов SQL MS ACCESS. Проанализировать ход их выполнения и полученный результат.

Контрольные вопросы:

1. Для чего служит оператор SQL SELECT? Приведите примеры использования.

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №26

Тема: «Операторы DROP и DELETE»

Цель: Научиться удалять таблицы и записи с помощью операторов DROP и DELETE языка SQL в MS Access.

Время выполнения: 2 урока

Ход работы:

Оператор SQL DROP является универсальным оператором удаления объектов базы данных.

Оператор SQL DROP имеет одинаковый синтаксис для удаления различных объектов:

DROP [INDEX | TABLE | DATABASE] object

Где object — название базы, таблицы, столбца в таблице, индекса.

Примеры оператора SQL DROP.

Пример 1. Удалить таблицу Artists используя оператор SQL DROP:

DROP TABLE Artists

Пример 2. Удалить базу данных Libraly используя оператор SQL DROP:

DROP DATABASE Libraly

Оператор DELETE удаляет строки из временных или постоянных базовых таблиц, представлений или курсоров, причем в двух последних случаях действие оператора распространяется на те базовые таблицы, из которых извлекались данные в эти представления или курсоры. Оператор удаления имеет простой синтаксис:

DELETE FROM <имя таблицы >

[WHERE <предикат>];

Если предложение WHERE отсутствует, удаляются все строки из таблицы или представления (представление должно быть обновляемым). Более быстро эту операцию (удаление всех строк из таблицы) можно в Transact-SQL также выполнить с помощью команды

TRUNCATE TABLE <имя таблицы>

Однако есть ряд особенностей в реализации команды TRUNCATE TABLE, которые следует иметь в виду:

- не журнализируется удаление отдельных строк таблицы; в журнал записывается только освобождение страниц, которые были заняты данными таблицы;
- не отрабатывают триггеры, в частности, триггер на удаление;
- команда неприменима, если на данную таблицу имеется ссылка по внешнему ключу, и даже если внешний ключ имеет опцию каскадного удаления.
- значение счетчика (IDENTITY) сбрасывается в начальное значение.

Пример 3

Требуется удалить из таблицы Laptop все портативные компьютеры с размером экрана менее 12 дюймов.

DELETE FROM Laptop

WHERE screen < 12;

Все блокноты можно удалить с помощью оператора

DELETE FROM Laptop;

или

TRUNCATE TABLE Laptop;

Transact-SQL расширяет синтаксис оператора DELETE, вводя дополнительное предложение FROM:

FROM <источник табличного типа>

При помощи источника табличного типа можно конкретизировать данные, удаляемые из таблицы в первом предложении FROM.

При помощи этого предложения можно выполнять соединения таблиц, что логически заменяет использование подзапросов в предложении WHERE для идентификации удаляемых строк. Поясним сказанное на примере.

Пример 4

Пусть требуется удалить те модели ПК из таблицы Product, для которых нет соответствующих строк в таблице PC.

Используя стандартный синтаксис, эту задачу можно решить следующим запросом:

```
DELETE FROM Product
WHERE type = 'pc' AND
 model NOT IN (SELECT model
 FROM PC
 );
```

Заметим, что предикат type = 'pc' необходим здесь, чтобы не были удалены также модели принтеров и портативных компьютеров.

Эту же задачу можно решить с помощью дополнительного предложения FROM следующим образом:

```
DELETE FROM Product
FROM Product pr LEFT JOIN
 PC ON pr.model = PC.model
WHERE type = 'pc' AND
 PC.model IS NULL;
```

Здесь применяется внешнее соединение, в результате чего столбец PC.model для моделей ПК, отсутствующих в таблице PC, будет содержать NULL-значение, что и используется для идентификации подлежащих удалению строк.

Задание: Выполнить запросы из приведенных выше примеров в редакторе запросов SQL MS ACCESS. Проанализировать ход их выполнения и полученный результат.

Контрольные вопросы:

2. Для чего служит оператор SQL DELETE? Приведите примеры использования.

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №27

Тема: «Операторы WHERE и ORDER BY»

Цель: Научиться создавать выборку с параметрами и сортировать результат из таблиц с помощью операторов WHERE и ORDER BY языка SQL в MS Access.

Время выполнения: 2 урока

Ход работы:

Оператор SQL WHERE служит для задания дополнительного условия выборки, операций вставки, редактирования и удаления записей.

Оператор SQL WHERE имеет следующий синтаксис:

WHERE condition

Условие (*condition*) может включать в себя предикаты AND, OR, NOT, LIKE, BETWEEN, IS, IN, ключевое слово NULL, операторы сравнения и равенства (<, >, =).

Примеры оператора SQL WHERE. Имеется следующая таблица Planets:

ID	PlanetName	Radius	SunSeason	OpeningYear	HavingRings	Opener
1	Mars	3396	687	1659	No	Christiaan Huygens
2	Saturn	60268	10759.22	—	Yes	—
3	Neptune	24764	60190	1846	Yes	John Couch Adams
4	Mercury	2439	115.88	1631	No	Nicolaus Copernicus
5	Venus	6051	243	1610	No	Galileo Galilei

Пример 1. Используя оператор SQL WHERE вывести записи, значение радиуса (Radius) которых находится в пределах от 3000 до 9000:

SELECT * FROM Planets WHERE Radius BETWEEN 3000 AND 9000

Результат:

ID	PlanetName	Radius	SunSeason	OpeningYear	HavingRings	Opener
1	Mars	3396	687	1659	No	Christiaan Huygens
5	Venus	6051	243	1610	No	Galileo Galilei

Пример 2. Используя оператор SQL WHERE вывести название планеты (PlanetName), год ее открытия (OpeningYear) и имя

первооткрывателя (Opener), планет, чье название не начинается или не заканчивается на букву «s»:

```
SELECT PlanetName, OpeningYear, Opener
FROM Planets
WHERE PlanetName NOT LIKE '%s'
AND PlanetName NOT LIKE 'S%'
```

Результат:

PlanetName	OpeningYear	Opener
Neptune	1846	John Couch Adams
Mercury	1631	Nicolaus Copernicus

Оператор SQL ORDER BY выполняет сортировку выходных значений. Оператор SQL ORDER BY можно применять как к числовым столбцам, так и к строковым. В последнем случае, сортировка будет происходить по алфавиту.

Оператор SQL ORDER BY имеет следующий синтаксис:

```
ORDER BY column_name [ASC | DESC]
```

Сортировка может производиться как по возрастанию, так и по убыванию значений.

Параметр **ASC** (по умолчанию) устанавливает порядок сортирования по возрастанию, от меньших значений к большим.

Параметр **DECS** устанавливает порядок сортирования по убыванию, от больших значений к меньшим.

Примеры оператора SQL ORDER BY. Имеется следующая таблица Artists:

Singer	Album	Year	Sale
The Prodigy	Invaders Must Die	2008	1200000
Drowning Pool	Sinner	2001	400000
Massive Attack	Mezzanine	1998	2300000
The Prodigy	Fat of the Land	1997	600000
The Prodigy	Music For The Jilted Generation	1994	1500000
Massive Attack	100th Window	2003	1200000
Drowning Pool	Full Circle	2007	800000
Massive Attack	Danny The Dog	2004	1900000
Drowning Pool	Resilience	2013	500000

Пример 3. Вывести все записи таблицы, упорядоченные по названию исполнителя с помощью оператора SQL ORDER BY:

```
SELECT *
FROM Artists
ORDER BY Singer
```

Результат:

Singer	Album	Year	Sale
Drowning Pool	Sinner	2001	400000
Drowning Pool	Full Circle	2007	800000
Drowning Pool	Resilience	2013	500000
Massive Attack	Mezzanine	1998	2300000
Massive Attack	100th Window	2003	1200000
Massive Attack	Danny The Dog	2004	1900000
The Prodigy	Invaders Must Die	2008	1200000
The Prodigy	Fat of the Land	1997	600000
The Prodigy	Music For The Jilted Generation	1994	1500000

Пример 4. С помощью оператора SQL ORDER BY вывести названия исполнителя, альбома, год выпуска тех альбомов, которые выпущены после 2005 года, упорядоченные по убыванию года:

```
SELECT Singer, Album, Year
FROM Artists
WHERE Year > 2005
ORDER BY Year DESC
```

Результат:

Singer	Album	Year
Drowning Pool	Resilience	2013
The Prodigy	Invaders Must Die	2008
Drowning Pool	Full Circle	2007

Задание: Выполнить запросы из приведенных выше примеров в редакторе запросов SQL MS ACCESS. Проанализировать ход их выполнения и полученный результат.

Контрольные вопросы:

1. Для чего служат операторы SQL WHERE и ORDER BY? Приведите примеры использования.

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №28

Тема: «Оператор UPDATE»

Цель: Научиться обновлять записи в таблицах с помощью оператор UPDATE языка SQL в MS Access.

Время выполнения: 2 урока

Ход работы:

Оператор SQL UPDATE используется для изменения значений в записях таблицы.

Оператор SQL UPDATE имеет следующий синтаксис:

UPDATE table_name SET expression [WHERE condition]

Напоминаю, что записи в квадратных скобках [] являются необязательными.

После ключевого слова SET должен идти список столбцов таблицы, которые следует изменить, и новые значения в формате Имя_столбца = значение.

Дополнительное условие, описываемое в операторе SQL WHERE, помогает более гибко совершать манипуляции данными.

Примеры оператора SQL UPDATE. Имеется следующая таблица Planets:

ID	PlanetName	Radius	SunSeason	OpeningYear	HavingRings	Opener
1	Mars	3396	687	1659	No	Christiaan Huygens
2	Saturn	60268	10759.22	—	Yes	—
3	Neptune	24764	60190	1846	Yes	John Couch Adams
4	Mercury	2439	115.88	1631	No	Nicolaus Copernicus
5	Venus	6051	243	1610	No	Galileo Galilei

Пример 1. С помощью оператора SQL UPDATE изменить название планеты Neptune на Pluton:

```
UPDATE Planets
SET PlanetName = 'Pluton'
WHERE ID = 3
```

В этом примере условие оператора SQL WHERE является обязательным, так как без него все поля столбца PlanetName во всей таблице изменились бы на Pluton. В данном случае, нам приходит на помощь столбец ID, ибо он является Первичным ключом, позволяющим однозначно идентифицировать запись.

Выполним запрос оператором SQL SELECT, чтобы посмотреть изменения в записи:

```
SELECT *
FROM Planets
WHERE ID = 3
```

Результат:

ID	PlanetName	Radius	SunSeason	OpeningYear	HavingRings	Opener
3	Pluton	24764	60190	1846	Yes	John Couch Adams

Пример 2. С помощью оператора SQL UPDATE у первых трех записей таблицы изменить значение наличия колец (HavingRings) на «No» и обнулить поле ID

Код примера для **MS SQL Server**:

```
UPDATE TOP(3) Planets
SET HavingRings = 'No', ID = NULL
```

Код примера для **MySQL**:

```
UPDATE Planets
SET HavingRings = 'No', ID = NULL
LIMIT 3
```

Выполним проверку:

```
SELECT TOP(3) *
FROM Planets
```

Результат:

ID	PlanetName	Radius	SunSeason	OpeningYear	HavingRings	Opener
NULL	Mars	3396	687	1659	No	Christiaan Huygens
NULL	Saturn	60268	10759.22	—	No	—
NULL	Neptune	24764	60190	1846	No	John Couch Adams

Задание: Выполнить запросы из приведенных выше примеров в редакторе запросов SQL MS ACCESS. Проанализировать ход их выполнения и полученный результат.

Контрольные вопросы:

2. Для чего служит оператор SQL UPDATE? Приведите примеры использования.

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №29

Тема: «Хранимые процедуры в SQL»

Цель: Научиться создавать хранимые процедуры с помощью языка SQL в MS Access.

Время выполнения: 2 урока

Ход работы:

Как правило, мы в работе с БД используем одни и те же запросы, либо набор последовательных запросов. Хранимые процедуры позволяют объединить последовательность запросов и сохранить их на сервере. Это очень удобный инструмент, и сейчас вы в этом убедитесь. Начнем с синтаксиса:

```
CREATE PROCEDURE имя_процедуры (параметры)
begin
операторы
end
```

Параметры это те данные, которые мы будем передавать процедуре при ее вызове, а операторы - это собственно запросы. Давайте напишем свою первую процедуру и убедимся в ее удобстве. В уроке 10, когда мы добавляли новые записи в БД shop, мы использовали стандартный запрос на добавление вида:

```
INSERT INTO customers (name, email) VALUE ('Иванов Сергей', 'sergo@mail.ru');
```

Т.к. подобный запрос мы будем использовать каждый раз, когда нам необходимо будет добавить нового покупателя, то вполне уместно оформить его в виде процедуры:

```
CREATE PROCEDURE ins_cust(n CHAR(50), e CHAR(50))
begin
insert into customers (name, email) value (n, e);
end
```

Обратите внимание, как задаются параметры: необходимо дать имя параметру и указать его тип, а в теле процедуры мы уже используем имена параметров. Один нюанс. Как вы помните, точка с запятой означает конец запроса и отправляет его на выполнение, что в данном случае неприемлемо. Поэтому, прежде, чем написать процедуру необходимо переопределить разделитель с ; на "//", чтобы запрос не отправлялся раньше времени. Делается это с помощью оператора DELIMITER //:

```
DELIMITER //
```

Таким образом, мы указали СУБД, что выполнять команды теперь следует после //. Следует помнить, что переопределение разделителя осуществляется только на один сеанс работы, т.е. при следующем сеансе работы с MySql разделитель снова станет точкой с запятой и при необходимости его придется снова переопределять. Теперь можно разместить процедуру:

```
CREATE PROCEDURE ins_cust(n CHAR(50), e CHAR(50))
 begin
 insert into customers (name, email) value (n, e);
 end
```

Итак, процедура создана. Теперь, когда нам понадобится ввести нового покупателя нам достаточно ее вызвать, указав необходимые параметры. Для вызова хранимой процедуры используется оператор CALL, после которого указывается имя процедуры и ее параметры. Давайте добавим нового покупателя в нашу таблицу Покупатели (customers):

```
call ins_cust('Сычов Валерий', 'valera@gmail.ru')//
```

Согласитесь, что так гораздо проще, чем писать каждый раз полный запрос. Проверим, работает ли процедура, посмотрев, появился ли новый покупатель в таблице Покупатели (customers):

Появился, процедура работает, и будет работать всегда, пока мы ее не удалим с помощью оператора DROP PROCEDURE название_процедуры.

Как было сказано в начале урока, процедуры позволяют объединить последовательность запросов. Давайте посмотрим, как это делается. Помните в уроке 11 мы хотели узнать, на какую сумму нам привез товар поставщик "Дом печати"? Для этого нам пришлось использовать вложенные запросы, объединения, вычисляемые столбцы и представления. А если мы захотим узнать, на какую сумму нам привез товар другой поставщик? Придется составлять новые запросы, объединения и т.д. Проще один раз написать хранимую процедуру для этого действия.

Казалось бы, проще всего взять уже написанные в уроке 11 представление и запрос к нему, объединить в хранимую процедуру и сделать идентификатор поставщика (id_vendor) входным параметром, вот так:

```
CREATE PROCEDURE sum_vendor(i INT)
 begin
 CREATE VIEW report_vendor AS SELECT magazine_incoming.id_product, magazine_incoming.quantity,
 prices.price, magazine_incoming.quantity*prices.price AS summa FROM magazine_incoming, prices
 WHERE magazine_incoming.id_product= prices.id_product AND id_incoming=
 (SELECT id_incoming FROM incoming WHERE id_vendor=i);
 SELECT SUM(summa) FROM report_vendor;
 end
```

Но так процедура работать не будет. Все дело в том, что в представлениях не могут использоваться параметры. Поэтому нам придется несколько изменить последовательность запросов. Сначала мы создадим представление, которое будет выводить идентификатор поставщика (id_vendor), идентификатор продукта (id_product), количество (quantity), цену (price) и сумму (summa) из трех таблиц Поставки (incoming), Журнал поставок (magazine_incoming), Цены (prices):

```
CREATE VIEW report_vendor AS SELECT incoming.id_vendor,
```

```
magazine_incoming.id_product, magazine_incoming.quantity,
prices.price, magazine_incoming.quantity*prices.price AS summa
FROM incoming, magazine_incoming, prices
WHERE magazine_incoming.id_product= prices.id_product AND
magazine_incoming.id_incoming= incoming.id_incoming;
```

А потом создадим запрос, который просуммирует суммы поставок интересующего нас поставщика, например, с id_vendor=2:

```
SELECT SUM(summa) FROM report_vendor WHERE id_vendor=2;
```

Вот теперь мы можем объединить два этих запроса в хранимую процедуру, где входным параметром будет идентификатор поставщика (id_vendor), который будет подставляться во второй запрос, но не в представление:

```
CREATE PROCEDURE sum_vendor(i INT)
begin
CREATE VIEW report_vendor AS SELECT incoming.id_vendor,
magazine_incoming.id_product, magazine_incoming.quantity,
prices.price, magazine_incoming.quantity*prices.price AS summa
FROM incoming, magazine_incoming, prices
WHERE magazine_incoming.id_product= prices.id_product AND
magazine_incoming.id_incoming= incoming.id_incoming;
SELECT SUM(summa) FROM report_vendor WHERE id_vendor=i;
end
```

Проверим работу процедуры, с разными входными параметрами:

```
call sum_vendor(1)//
call sum_vendor(2)//
call sum_vendor(3)//
```

Как видите, процедура срабатывает один раз, а затем выдает ошибку, говоря нам, что представление report_vendor уже имеется в БД. Так происходит потому, что при обращении к процедуре в первый раз, она создает представление. При обращении во второй раз, она снова пытается

создать представление, но оно уже есть, поэтому и появляется ошибка. Чтобы избежать этого возможно два варианта.

Первый - вынести представление из процедуры. То есть мы один раз создадим представление, а процедура будет лишь к нему обращаться, но не создавать его. Предварительно не забудет удалить уже созданную процедуру и представление:

```
DROP PROCEDURE sum_vendor//
DROP VIEW report_vendor//
CREATE VIEW report_vendor AS SELECT incoming.id_vendor,
 magazine_incoming.id_product, magazine_incoming.quantity,
 prices.price, magazine_incoming.quantity*prices.price AS summa
 FROM incoming, magazine_incoming, prices
 WHERE magazine_incoming.id_product= prices.id_product AND
 magazine_incoming.id_incoming= incoming.id_incoming//
CREATE PROCEDURE sum_vendor(i INT)
 begin
 SELECT SUM(summa) FROM report_vendor WHERE id_vendor=i;
 end
```

Проверяем работу:

```
call sum_vendor(1)//
call sum_vendor(2)//
call sum_vendor(3)//
```

Второй вариант - прямо в процедуре дописать команду, которая будет удалять представление, если оно существует:

```
CREATE PROCEDURE sum_vendor(i INT)
 begin
 DROP VIEW IF EXISTS report_vendor;
 CREATE VIEW report_vendor AS SELECT incoming.id_vendor,
 magazine_incoming.id_product, magazine_incoming.quantity,
 prices.price, magazine_incoming.quantity*prices.price AS summa
```

```
FROM incoming, magazine_incoming, prices
WHERE magazine_incoming.id_product= prices.id_product AND
magazine_incoming.id_incoming= incoming.id_incoming;
SELECT SUM(summa) FROM report_vendor WHERE id_vendor=i;
end
```

Перед использованием этого варианта не забудьте удалить процедуру sum_vendor, а затем проверить работу:

Как видите, сложные запросы или их последовательность действительно проще один раз оформить в хранимую процедуру, а дальше просто обращаться к ней, указывая необходимые параметры. Это значительно сокращает код и делает работу с запросами более логичной.

Задание: Создать хранимые процедуры из приведенных выше примеров в редакторе запросов SQL MS ACCESS. Проанализировать ход их выполнения и полученный результат.

Контрольные вопросы:

1. Для чего служат хранимые процедуры в SQL? Приведите примеры их использования.

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №30

Тема: «Триггеры»

Цель: Научиться создавать триггеры с помощью языка SQL в MS Access.

Время выполнения: 2 урока

Ход работы:

Дается определение триггера, область его использования, место и роль триггера в обеспечении целостности данных. Описываются типы триггеров. Рассматриваются операторы создания, изменения, удаления триггера. Программирование триггера иллюстрируется примерами создания триггеров для реализации ограничений целостности и сбора статистических данных.

Определение триггера в стандарте языка SQL

Триггеры являются одной из разновидностей хранимых процедур. Их исполнение происходит при выполнении для таблицы какого-либо оператора языка манипулирования данными (DML). Триггеры используются для проверки целостности данных, а также для отката транзакций.

Триггер – это откомпилированная SQL-процедура, исполнение которой обусловлено наступлением определенных событий внутри реляционной базы данных. Применение триггеров большей частью весьма удобно для пользователей базы данных. И все же их использование часто связано с дополнительными затратами ресурсов на операции ввода/вывода. В том случае, когда тех же результатов (с гораздо меньшими непроизводительными затратами ресурсов) можно добиться с помощью хранимых процедур или прикладных программ, применение триггеров нецелесообразно.

Триггеры – особый инструмент SQL-сервера, используемый для поддержания целостности данных в базе данных. С помощью ограничений целостности, правил и значений по умолчанию не всегда можно добиться нужного уровня функциональности. Часто требуется реализовать сложные алгоритмы проверки данных, гарантирующие их достоверность и реальность.

Кроме того, иногда необходимо отслеживать изменения значений таблицы, чтобы нужным образом изменить связанные данные. Триггеры можно рассматривать как своего рода фильтры, вступающие в действие после выполнения всех операций в соответствии с правилами, стандартными значениями и т.д.

Триггер представляет собой специальный тип хранимых процедур, запускаемых сервером автоматически при попытке изменения данных в таблицах, с которыми триггеры связаны. Каждый триггер привязывается к конкретной таблице. Все производимые им модификации данных рассматриваются как одна транзакция. В случае обнаружения ошибки или нарушения целостности данных происходит откат этой транзакции. Тем самым внесение изменений запрещается. Отменяются также все изменения, уже сделанные триггером.

Создает триггер только владелец базы данных. Это ограничение позволяет избежать случайного изменения структуры таблиц, способов связи с ними других объектов и т.п.

Триггер представляет собой весьма полезное и в то же время опасное средство. Так, при неправильной логике его работы можно легко уничтожить целую базу данных, поэтому триггеры необходимо очень тщательно отлаживать.

В отличие от обычной подпрограммы, триггер выполняется неявно в каждом случае возникновения триггерного события, к тому же он не имеет аргументов. Приведение его в действие иногда называют запуском триггера. С помощью триггеров достигаются следующие цели:

проверка корректности введенных данных и выполнение сложных ограничений целостности данных, которые трудно, если вообще возможно, поддерживать с помощью ограничений целостности, установленных для таблицы;

выдача предупреждений, напоминающих о необходимости выполнения некоторых действий при обновлении таблицы, реализованном определенным образом;

накопление аудиторской информации посредством фиксации сведений о внесенных изменениях и тех лицах, которые их выполнили;

поддержка репликации.

Основной формат команды CREATE TRIGGER показан ниже:

<Определение_триггера>::=

CREATE TRIGGER имя_триггера

BEFORE | AFTER <триггерное_событие>

ON <имя_таблицы>

[REFERENCING

<список_старых_или_новых_псевдонимов>]

[FOR EACH { ROW | STATEMENT}]

[WHEN(условие_триггера)]

<тело_триггера>

триггерные события состоят из вставки, удаления и обновления строк в таблице. В последнем случае для триггерного события можно указать конкретные имена столбцов таблицы. Время запуска триггера определяется с помощью ключевых слов BEFORE (триггер запускается до выполнения связанных с ним событий) или AFTER (после их выполнения).

Выполняемые триггером действия задаются для каждой строки (FOR EACH ROW), охваченной данным событием, или только один раз для каждого события (FOR EACH STATEMENT).

Обозначение <список_старых_или_новых_псевдонимов> относится к таким компонентам, как старая или новая строка (OLD / NEW) либо старая или новая таблица (OLD TABLE / NEW TABLE). Ясно, что старые значения не применимы для событий вставки, а новые – для событий удаления.

При условии правильного использования триггеры могут стать очень мощным механизмом. Основное их преимущество заключается в том, что

стандартные функции сохраняются внутри базы данных и согласованно активизируются при каждом ее обновлении. Это может существенно упростить приложения. Тем не менее следует упомянуть и о присущих триггеру недостатках:

Сложность: при перемещении некоторых функций в базу данных усложняются задачи ее проектирования, реализации и администрирования;

Скрытая функциональность: перенос части функций в базу данных и сохранение их в виде одного или нескольких триггеров иногда приводит к сокрытию от пользователя некоторых функциональных возможностей. Хотя это в определенной степени упрощает его работу, но, к сожалению, может стать причиной незапланированных, потенциально нежелательных и вредных побочных эффектов, поскольку в этом случае пользователь не в состоянии контролировать все процессы, происходящие в базе данных;

Влияние на производительность: перед выполнением каждой команды по изменению состояния базы данных СУБД должна проверить триггерное условие с целью выяснения необходимости запуска триггера для этой команды. Выполнение подобных вычислений сказывается на общей производительности СУБД, а в моменты пиковой нагрузки ее снижение может стать особенно заметным. Очевидно, что при возрастании количества триггеров увеличиваются и накладные расходы, связанные с такими операциями.

Неправильно написанные триггеры могут привести к серьезным проблемам, таким, например, как появление "мертвых" блокировок. Триггеры способны длительное время блокировать множество ресурсов, поэтому следует обратить особое внимание на сведение к минимуму конфликтов доступа.

Реализация триггеров в среде MS SQL Server

В реализации СУБД MS SQL Server используется следующий оператор создания или изменения триггера:

<Определение_триггера>::=

```
{CREATE | ALTER} TRIGGER имя_триггера
ON {имя_таблицы | имя_просмотра }
[WITH ENCRYPTION ]
{
{ { FOR | AFTER | INSTEAD OF }
{ [ DELETE] [,] [ INSERT] [,] [ UPDATE] }
[ WITH APPEND ]
[ NOT FOR REPLICATION ]
AS
 sql_оператор[...n]
} |
{ {FOR | AFTER | INSTEAD OF } { [INSERT] [,]
 [UPDATE] }
[ WITH APPEND]
[ NOT FOR REPLICATION]
AS
{ IF UPDATE(имя_столбца)
[ {AND | OR} UPDATE(имя_столбца)] [...n]
|
IF (COLUMNS_UPDATES(){оператор_бит_обработки}
 бит_маска_изменения)
{оператор_бит_сравнения }бит_маска [...n]}
sql_оператор [...n]
}
}
```

Триггер может быть создан только в текущей базе данных, но допускается обращение внутри триггера к другим базам данных, в том числе и расположенным на удаленном сервере.

Рассмотрим назначение аргументов из команды CREATE | ALTER TRIGGER.

Имя триггера должно быть уникальным в пределах базы данных. Дополнительно можно указать имя владельца.

При указании аргумента WITH ENCRYPTION сервер выполняет шифрование кода триггера, чтобы никто, включая администратора, не мог получить к нему доступ и прочитать его. Шифрование часто используется для скрытия авторских алгоритмов обработки данных, являющихся интеллектуальной собственностью программиста или коммерческой тайной.

Типы триггеров

В SQL существует два параметра, определяющих поведение триггеров:

AFTER. Триггер выполняется после успешного выполнения вызвавших его команд. Если же команды по какой-либо причине не могут быть успешно завершены, триггер не выполняется. Следует отметить, что изменения данных в результате выполнения запроса пользователя и выполнение триггера осуществляется в теле одной транзакции: если произойдет откат триггера, то будут отклонены и пользовательские изменения. Можно определить несколько AFTER-триггеров для каждой операции (INSERT, UPDATE, DELETE). Если для таблицы предусмотрено выполнение нескольких AFTER-триггеров, то с помощью системной хранимой процедуры sp_settriggerorder можно указать, какой из них будет выполняться первым, а какой последним. По умолчанию в SQL Server все триггеры являются AFTER-триггерами.

INSTEAD OF. Триггер вызывается вместо выполнения команд. В отличие от AFTER-триггера INSTEAD OF-триггер может быть определен как для таблицы, так и для просмотра. Для каждой операции INSERT, UPDATE, DELETE можно определить только один INSTEAD OF-триггер.

Триггеры различают по типу команд, на которые они реагируют.

Существует три типа триггеров:

INSERT TRIGGER – запускаются при попытке вставки данных с помощью команды INSERT.

UPDATE TRIGGER – запускаются при попытке изменения данных с помощью команды UPDATE.

DELETE TRIGGER – запускаются при попытке удаления данных с помощью команды DELETE.

Конструкции [DELETE] [,] [INSERT] [,] [UPDATE] и FOR | AFTER | INSTEAD OF } { [INSERT] [,] [UPDATE] определяют, на какую команду будет реагировать триггер. При его создании должна быть указана хотя бы одна команда. Допускается создание триггера, реагирующего на две или на все три команды.

Аргумент WITH APPEND позволяет создавать несколько триггеров каждого типа.

При создании триггера с аргументом NOT FOR REPLICATION запрещается его запуск во время выполнения модификации таблиц механизмами репликации.

Конструкция AS sql_оператор[...n] определяет набор SQL- операторов и команд, которые будут выполнены при запуске триггера.

Отметим, что внутри триггера не допускается выполнение ряда операций, таких, например, как:

Создание, изменение и удаление базы данных;

Восстановление резервной копии базы данных или журнала транзакций.

Выполнение этих команд не разрешено, так как они не могут быть отменены в случае отката транзакции, в которой выполняется триггер. Это запрещение вряд ли может каким-то образом сказаться на функциональности создаваемых триггеров. Трудно найти такую ситуацию, когда, например, после изменения строки таблицы потребуется выполнить восстановление резервной копии журнала транзакций.

Программирование триггера

При выполнении команд добавления, изменения и удаления записей сервер создает две специальные таблицы: inserted и deleted . В них

содержатся списки строк, которые будут вставлены или удалены по завершении транзакции. Структура таблиц inserted и deleted идентична структуре таблиц, для которой определяется триггер. Для каждого триггера создается свой комплект таблиц inserted и deleted, поэтому никакой другой триггер не сможет получить к ним доступ. В зависимости от типа операции, вызвавшей выполнение триггера, содержимое таблиц inserted и deleted может быть разным:

Команда INSERT – в таблице inserted содержатся все строки, которые пользователь пытается вставить в таблицу; в таблице deleted не будет ни одной строки; после завершения триггера все строки из таблицы inserted переместятся в исходную таблицу;

Команда DELETE – в таблице deleted будут содержаться все строки, которые пользователь попытается удалить; триггер может проверить каждую строку и определить, разрешено ли ее удаление; в таблице inserted не окажется ни одной строки;

Команда UPDATE – при ее выполнении в таблице deleted находятся старые значения строк, которые будут удалены при успешном завершении триггера. Новые значения строк содержатся в таблице inserted. Эти строки добавятся в исходную таблицу после успешного выполнения триггера.

Для получения информации о количестве строк, которое будет изменено при успешном завершении триггера, можно использовать функцию @@ROWCOUNT; она возвращает количество строк, обработанных последней командой. Следует подчеркнуть, что триггер запускается не при попытке изменить конкретную строку, а в момент выполнения команды изменения. Одна такая команда воздействует на множество строк, поэтому триггер должен обрабатывать все эти строки.

Если триггер обнаружил, что из 100 вставляемых, изменяемых или удаляемых строк только одна не удовлетворяет тем или иным условиям, то никакая строка не будет вставлена, изменена или удалена. Такое поведение

обусловлено требованиями транзакции – должны быть выполнены либо все модификации, либо ни одной.

Триггер выполняется как неявно определенная транзакция, поэтому внутри триггера допускается применение команд управления транзакциями. В частности, при обнаружении нарушения ограничений целостности для прерывания выполнения триггера и отмены всех изменений, которые пытался выполнить пользователь, необходимо использовать команду ROLLBACK TRANSACTION.

Для получения списка столбцов, измененных при выполнении команд INSERT или UPDATE, вызвавших выполнение триггера, можно использовать функцию COLUMNS_UPDATED(). Она возвращает двоичное число, каждый бит которого, начиная с младшего, соответствует одному столбцу таблицы (в порядке следования столбцов при создании таблицы). Если бит установлен в значение "1", то соответствующий столбец был изменен. Кроме того, факт изменения столбца определяет и функция UPDATE (имя_столбца).

Для удаления триггера используется команда

DROP TRIGGER {имя_триггера} [,...n]

Приведем примеры использования триггеров.

Пример 1. Использование триггера для реализации ограничений на значение. В добавляемой в таблицу Сделка записи количество проданного товара должно быть не меньше, чем его остаток из таблицы Склад.

Команда вставки записи в таблицу Сделка может быть, например, такой:

```
INSERT INTO Сделка
 VALUES (3,1,-299,'01/08/2002')
```

Создаваемый триггер должен отреагировать на ее выполнение следующим образом: необходимо отменить команду, если в таблице Склад величина остатка товара оказалась меньше продаваемого количества товара с введенным кодом (в примере код товара=3). Во вставляемой записи количество товара указывается со знаком "+", если товар поставляется, и со

знаком "-", если он продается. Представленный триггер настроен на обработку только одной добавляемой записи.

```
CREATE TRIGGER Триггер_ins
ON Сделка FOR INSERT
AS
IF @@ROWCOUNT=1
BEGIN
 IF NOT EXISTS(SELECT *
 FROM inserted
 WHERE -inserted.количество<=ALL(SELECT
 Склад.Остаток
 FROM Склад,Сделка
 WHERE Склад.КодТовара=
 Сделка.КодТовара))
 BEGIN
 ROLLBACK TRAN
 PRINT
 'Отмена поставки: товара на складе нет'
 END
END
```

Пример 1. Использование триггера для реализации ограничений на значение.

Пример 2. Использования триггера для сбора статистических данных.

Создать триггер для обработки операции вставки записи в таблицу Сделка, например, такой команды:

```
INSERT INTO Сделка
 VALUES (3,1,200,'01/08/2002')
```

поставляется товар с кодом 3 от клиента с кодом 1 в количестве 200 единиц.

При продаже или получении товара необходимо соответствующим образом изменить количество его складского запаса. Если товара на складе еще нет, необходимо добавить соответствующую запись в таблицу Склад. Триггер обрабатывает только одну добавляемую строку.

```
 ALTER TRIGGER Триггер_ins
ON Сделка FOR INSERT
AS
DECLARE @x INT, @y INT
IF @@ROWCOUNT=1
--в таблицу Сделка добавляется запись
--о поставке товара
BEGIN
--количество проданного товара должно быть не
--меньше, чем его остаток из таблицы Склад
IF NOT EXISTS(SELECT *
 FROM inserted
 WHERE -inserted.количество<
 =ALL(SELECT Склад.Остаток
 FROM Склад,Сделка
 WHERE Склад.КодТовара=
 Сделка.КодТовара))
 BEGIN
 ROLLBACK TRAN
 PRINT 'откат товара нет '
 END
--если записи о поставленном товаре еще нет,
--добавляется соответствующая запись
--в таблицу Склад
 IF NOT EXISTS ( SELECT *
 FROM Склад C, inserted i
```

```
 WHERE C.КодТовара=i.КодТовара )
 INSERT INTO Склад (КодТовара,Остаток)
 ELSE
--если запись о товаре уже была в таблице
--Склад, то определяется код и количество
--товара издобавленной в таблицу Сделка записи
 BEGIN
 SELECT @y=i.КодТовара, @x=i.Количество
 FROM Сделка C, inserted i
 WHERE C.КодТовара=i.КодТовара
--и производится изменения количества товара в
--таблице Склад
 UPDATE Склад
 SET Остаток=остаток+@x
 WHERE КодТовара=@y
 END
END
```

Пример 2. Использования триггера для сбора статистических данных.

Пример 3. Создать триггер для обработки операции удаления записи из таблицы Сделка, например, такой команды:

```
DELETE FROM Сделка WHERE КодСделки=4
```

Для товара, код которого указан при удалении записи, необходимо откорректировать его остаток на складе. Триггер обрабатывает только одну удаляемую запись.

```
 CREATE TRIGGER Триггер_del
ON Сделка FOR DELETE
AS
IF @@ROWCOUNT=1 -- удалена одна запись
BEGIN
 DECLARE @y INT,@x INT
```

```
--определяется код и количество товара из
--удаленной из таблицы Склад записи
 SELECT @y=КодТовара, @x=Количество
 FROM deleted
--в таблице Склад корректируется количество
--товара
 UPDATE Склад
 SET Остаток=Остаток-@x
 WHERE КодТовара=@y
END
```

Пример 3. Триггер для обработки операции удаления записи из таблицы

Пример 4. Создать триггер для обработки операции изменения записи в таблице Сделка, например, такой командой:

```
UPDATE Сделка SET количество=количество-10
WHERE КодТовара=3
```

во всех сделках с товаром, имеющим код, равный 3, уменьшить количество товара на 10 единиц.

Указанная команда может привести к изменению сразу нескольких записей в таблице Сделка. Поэтому покажем, как создать триггер, обрабатывающий не одну запись. Для каждой измененной записи необходимо для старого (до изменения) кода товара уменьшить остаток товара на складе на величину старого (до изменения) количества товара и для нового (после изменения) кода товара увеличить его остаток на складе на величину нового (после изменения) значения. Чтобы обработать все измененные записи, введем курсоры, в которых сохраним все старые (из таблицы deleted) и все новые значения (из таблицы inserted).

```
 CREATE TRIGGER Триггер_upd
ON Сделка FOR UPDATE
AS
```

```
DECLARE @x INT, @x_old INT, @y INT, @y_old INT
-- курсор с новыми значениями
DECLARE CUR1 CURSOR FOR
 SELECT КодТовара,Количество
 FROM inserted
-- курсор со старыми значениями
DECLARE CUR2 CURSOR FOR
 SELECT КодТовара,Количество
 FROM deleted
OPEN CUR1
OPEN CUR2
-- перемещаемся параллельно по обоим курсорам
 FETCH NEXT FROM CUR1 INTO @x, @y
 FETCH NEXT FROM CUR2 INTO @x_old, @y_old
 WHILE @@FETCH_STATUS=0
 BEGIN
--для старого кода товара уменьшается его
--количество на складе
 UPDATE Склад
 SET Остаток=Остаток-@y_old
 WHERE КодТовара=@x_old
--для нового кода товара, если такого товара
--еще нет на складе, вводится новая запись
 IF NOT EXISTS (SELECT * FROM Склад
 WHERE КодТовара=@x)
 INSERT INTO Склад(КодТовара,Остаток)
 VALUES (@x,@y)
 ELSE
--иначе для нового кода товара увеличивается
--его количество на складе
```

```
 UPDATE Склад
 SET Остаток=Остаток+@y
 WHERE КодТовара=@x
 FETCH NEXT FROM CUR1 INTO @x, @y
 FETCH NEXT FROM CUR2 INTO @x_old, @y_old
END
CLOSE CUR1
CLOSE CUR2
DEALLOCATE CUR1
DEALLOCATE CUR2
```

Пример 4. триггер для обработки операции изменения записи в таблице

В рассмотренном триггере отсутствует сравнение количества товара при изменении записи о сделке с его остатком на складе.

Пример 5. Исправим этот недостаток. Для генерирования сообщения об ошибке используем в теле триггера команду MS SQL Server RAISERROR, аргументами которой являются текст сообщения, уровень серьезности и статус ошибки.

```
 ALTER TRIGGER Триггер_upd
ON Сделка FOR UPDATE
AS
DECLARE @x INT, @x_old INT, @y INT,
 @y_old INT ,@o INT
DECLARE CUR1 CURSOR FOR
 SELECT КодТовара,Количество
 FROM inserted
DECLARE CUR2 CURSOR FOR
 SELECT КодТовара,Количество
 FROM deleted
OPEN CUR1
OPEN CUR2
```

```
FETCH NEXT FROM CUR1 INTO @x, @y
FETCH NEXT FROM CUR2 INTO @x_old, @y_old
WHILE @@FETCH_STATUS=0
BEGIN
SELECT @o=остаток
FROM Склад
WHERE кодтовара=@x
IF @o<-@y
BEGIN
RAISERROR('откат',16,10)
CLOSE CUR1
CLOSE CUR2
DEALLOCATE CUR1
DEALLOCATE CUR22
ROLLBACK TRAN
RETURN
END
UPDATE Склад
SET Остаток=Остаток-@y_old
WHERE КодТовара=@x_old
IF NOT EXISTS (SELECT * FROM Склад
WHERE КодТовара=@x)
INSERT INTO Склад(КодТовара,Остаток)
VALUES (@x,@y)
ELSE
UPDATE Склад
SET Остаток=Остаток+@y
WHERE КодТовара=@x
FETCH NEXT FROM CUR1 INTO @x, @y
FETCH NEXT FROM CUR2 INTO @x_old, @y_old
```

```
END
CLOSE CUR1
CLOSE CUR2
DEALLOCATE CUR1
DEALLOCATE CUR2
```

Пример 5. Исправленный вариант триггера для обработки операции изменения записи в таблице

Пример 6. В примере 5 происходит отмена всех изменений при невозможности реализовать хотя бы одно из них. Создадим триггер, позволяющий отменять изменение только некоторых записей и выполнять изменение остальных.

В этом случае триггер выполняется не после изменения записей, а вместо команды изменения.

```
 ALTER TRIGGER Триггер_upd
ON Сделка INSTEAD OF UPDATE
AS
DECLARE @k INT, @k_old INT
DECLARE @x INT, @x_old INT, @y INT
DECLARE @y_old INT ,@o INT
DECLARE CUR1 CURSOR FOR
 SELECT КодСделки, КодТовара,Количество
 FROM inserted
DECLARE CUR2 CURSOR FOR
 SELECT КодСделки, КодТовара,Количество
 FROM deleted
OPEN CUR1
OPEN CUR2
 FETCH NEXT FROM CUR1 INTO @k,@x, @y
 FETCH NEXT FROM CUR2 INTO @k_old,@x_old,
 @y_old
```

```
WHILE @@FETCH_STATUS=0
BEGIN
SELECT @o=остаток
FROM Склад
WHERE КодТовара=@x
IF @o>=-@y
BEGIN
RAISERROR('изменение',16,10)
UPDATE Сделка SET количество=@y,
КодТовара=@x
WHERE КодСделки=@k

UPDATE Склад
SET Остаток=Остаток-@y_old
WHERE КодТовара=@x_old

IF NOT EXISTS (SELECT * FROM Склад
WHERE КодТовара=@x)
INSERT INTO Склад(КодТовара, Остаток)
VALUES (@x,@y)
ELSE
UPDATE Склад
SET Остаток=Остаток+@y
WHERE КодТовара=@x
END
ELSE
RAISERROR('запись не изменена',16,10)
FETCH NEXT FROM CUR1 INTO @k,@x, @y
FETCH NEXT FROM CUR2 INTO @k_old,@x_old,
@y_old
```

```
END
CLOSE CUR1
CLOSE CUR2
DEALLOCATE CUR1
DEALLOCATE CUR2
```

Пример 6. Триггер, позволяющий отменять изменение только некоторых записей и выполнять изменение остальных.

Задание: Создать триггеры из приведенных выше примеров в редакторе запросов SQL MS ACCESS. Проанализировать ход их выполнения и полученный результат.

Контрольные вопросы:

1. Для чего служат триггеры?
2. Какие группы триггеров вы знаете?

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №31

Тема: «Работа с клиент-серверными моделями»

Цель: Познакомиться с принципом работы клиент-серверных приложений. Научиться проектировать клиент-серверные БД.

Время выполнения: 2 урока

Ход работы:

Как правило компьютеры и программы, входящие в состав информационной системы, не являются равноправными. Некоторые из них владеют ресурсами (файловая система, процессор, принтер, база данных и т.д.), другие имеют возможность обращаться к этим ресурсам. Компьютер (или программу), управляющий ресурсом, называют сервером этого ресурса (файл-сервер, сервер базы данных, вычислительный сервер...). Клиент и сервер какого-либо ресурса могут находится как на одном компьютере, так и на различных компьютерах, связанных сетью.

В рамках многоуровневого представления вычислительных систем можно выделить три группы функций, ориентированных на решение различных подзадач:

- функции ввода и отображения данных (обеспечивают взаимодействие с пользователем);
- прикладные функции, характерные для данной предметной области;
- функции управления ресурсами (файловой системой, базой даных и т.д.)

Рис.1. Компоненты сетевого приложения

Выполнение этих функций в основном обеспечивается программными средствами, которые можно представить в виде взаимосвязанных компонентов (рис. 1), где:

- *компонент представления* отвечает за пользовательский интерфейс;
- *прикладной компонент* реализует алгоритм решения конкретной задачи;
- *компонент управления* ресурсом обеспечивает доступ к необходимым ресурсам.

Автономная система (компьютер, не подключенный к сети) представляет все эти компоненты как на различных уровнях (ОС, служебное ПО и утилиты, прикладное ПО), так и на уровне приложений (не характерно для современных программ). Так же и сеть — она представляет все эти компоненты, но, в общем случае, распределенные между узлами. Задача сводится к обеспечению сетевого взаимодействия между этими компонентами.

Архитектура «клиент-сервер» определяет общие принципы организации взаимодействия в сети, где имеются *серверы*, узлы-поставщики некоторых специфичных функций (сервисов) и *клиенты*, потребители этих функций.

Практические реализации такой архитектуры называются клиент-серверными технологиями. Каждая технология определяет собственные или использует имеющиеся правила взаимодействия между клиентом и сервером, которые называются *протоколом обмена (протоколом взаимодействия)*.

Двухзвенная архитектура

В любой сети (даже одноранговой), построенной на современных сетевых технологиях, присутствуют элементы клиент-серверного взаимодействия, чаще всего на основе **двухзвенной архитектуры**. Двухзвенной (two-tier, 2-tier) она называется из-за необходимости

распределения *трех* базовых *компонентов* между *двумя узлами* (клиентом и сервером).

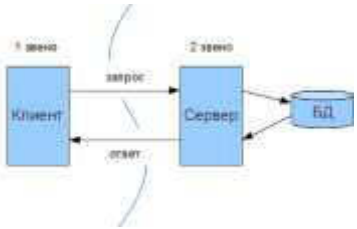


Рис.2. Двухзвенная клиент-серверная архитектура

Двухзвенная архитектура используется в клиент-серверных системах, где сервер отвечает на клиентские запросы напрямую и в полном объеме, при этом используя только собственные ресурсы. Т.е. сервер не вызывает сторонние сетевые приложения и не обращается к сторонним ресурсам для выполнения какой-либо части запроса (рис. 2)

Расположение компонентов на стороне клиента или сервера определяет следующие основные модели их взаимодействия в рамках двухзвенной архитектуры:

сервер терминалов — распределенное представление данных;

файл-сервер — доступ к удаленной базе данных и файловым ресурсам;

сервер БД — удаленное представление данных;

сервер приложений — удаленное приложение.

Перечисленные модели с вариациями представлены на рис. 3.

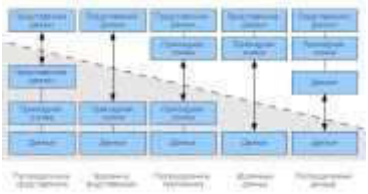


Рис.3. Модели клиент-серверного взаимодействия

Исторически первой появилась модель распределенного представления данных (модель сервер терминалов). Она реализовывалась на универсальной ЭВМ (мэйнфрейме), выступавшей в роли сервера, с подключенными к ней алфавитно-цифровыми терминалами. Пользователи выполняли ввод данных с клавиатуры терминала, которые затем передавались на мэйнфрейм и там

выполнялась их обработка, включая формирование «картинки» с результатами. Эта «картинка» и возвращалась пользователю на экран терминала.

С появлением персональных компьютеров и локальных сетей, была реализована модель файлового сервера, представлявшего доступ файловым ресурсам, в т.ч и к удаленной базе данных. В этом случае выделенный узел сети является файловым сервером, на котором размещены файлы базы данных. На клиентах выполняются приложения, в которых совмещены компонент представления и прикладной компонент (СУБД и прикладная программа), использующие подключенную удаленную базу как локальный файл. Протоколы обмена при этом представляют набор низкоуровневых вызовов операций файловой системы.

Такая модель показала свою неэффективность ввиду того, что при активной работе с таблицами БД возникает большая нагрузка на сеть. Частичным решением является поддержка тиражирования (репликации) таблиц и запросов. В этом случае, например, при изменении данных, обновляется не вся таблица, а только модифицированная ее часть.

С появлением специализированных СУБД появилась возможность реализации другой модели доступа к удаленной базе данных — модели сервера баз данных. В этом случае ядро СУБД функционирует на сервере, прикладная программа на клиенте, а протокол обмена обеспечивается с помощью языка SQL. Такой подход по сравнению с файловым сервером ведет к уменьшению загрузки сети и унификации интерфейса «клиент-сервер». Однако, сетевой трафик остается достаточно высоким, кроме того, по прежнему невозможно удовлетворительное администрирование приложений, поскольку в одной программе совмещаются различные функции.

С разработкой и внедрением на уровне серверов баз данных механизма хранимых процедур появилась концепция *активного сервера БД*. В этом случае часть функций прикладного компонента реализованы в виде

хранимых процедур, выполняемых на стороне сервера. Остальная прикладная логика выполняется на клиентской стороне. Протокол взаимодействия — соответствующий диалект языка SQL.

Преимущества такого подхода очевидны:

возможно централизованное администрирование прикладных функций;

снижение стоимости владения системой (TOC, total cost of ownership) за счет аренды сервера, а не его покупки;

значительное снижение сетевого трафика (т.к. передаются не SQL-запросы, а вызовы хранимых процедур).

Основной недостаток — ограниченность средств разработки хранимых процедур по сравнению с языками высокого уровня.

Реализация прикладного компонента на стороне сервера представляет следующую модель — сервер приложений. Перенос функций прикладного компонента на сервер снижает требования к конфигурации клиентов и упрощает администрирование, но представляет повышенные требования к производительности, безопасности и надежности сервера.

В настоящее время намечается тенденция возврата к тому, с чего начиналась клиент-серверная архитектура — к централизации вычислений на основе модели терминал-сервера. В современной реинкарнации терминалы отличаются от своих алфавитно-цифровых предков тем, что имея минимум программных и аппаратных средств, представляют мультимедийные возможности (в т.ч. *графический пользовательский интерфейс*). Работу терминалов обеспечивает высокопроизводительный сервер, куда вынесено все, вплоть до виртуальных драйверов устройств, включая драйверы видеоподсистемы.

Трехзвенная архитектура

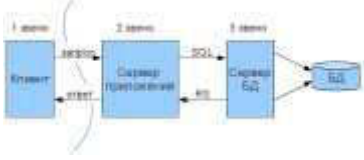


Рис.4. Трехзвенная клиент-серверная архитектура

Еще одна тенденция в клиент-серверных технологиях связана со все большим использованием распределенных вычислений. Они реализуются на основе модели сервера приложений, где сетевое приложение разделено на две и более частей, каждая из которых может выполняться на отдельном компьютере. Выделенные части приложения взаимодействуют друг с другом, обмениваясь сообщениями в заранее согласованном формате. В этом случае двухзвенная клиент-серверная архитектура становится трехзвенной (three-tier, 3-tier).

Как правило, третьим звеном в трехзвенной архитектуре становится сервер приложений, т.е. компоненты распределяются следующим образом (рис. 4):

Представление данных — на стороне клиента.

Прикладной компонент — на выделенном сервере приложений (как вариант, выполняющем функции промежуточного ПО).

Управление ресурсами — на сервере БД, который и представляет запрашиваемые данные.

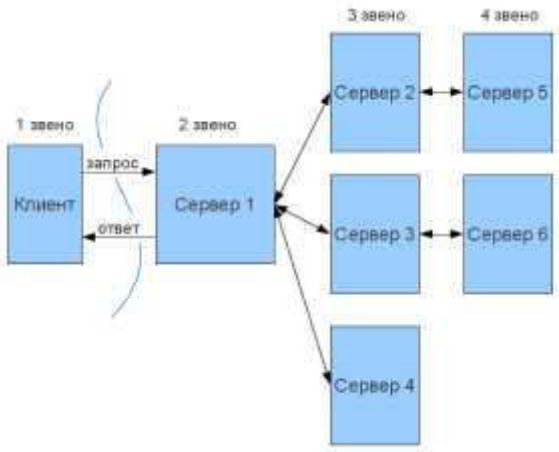


Рис.5. Многозвенная (N-tier) клиент-серверная архитектура

Трехзвенная архитектура может быть расширена до многозвенной (N-tier, Multi-tier) путем выделения дополнительных серверов, каждый из которых будет представлять собственные сервисы и пользоваться услугами прочих серверов разного уровня. Абстрактный пример многозвенной модели приведен на рис. 5.

Сравнение архитектур

Двухзвенная архитектура проще, так как все запросы обслуживаются одним сервером, но именно из-за этого она менее надежна и предъявляет повышенные требования к производительности сервера.

Трехзвенная архитектура сложнее, но благодаря тому, что функции распределены между серверами второго и третьего уровня, эта архитектура представляет:

Высокую степень гибкости и масштабируемости.

Высокую безопасность (т.к. защиту можно определить для каждого сервиса или уровня).

Высокую производительность (т.к. задачи распределены между серверами).

Клиент-серверные технологии

Архитектура клиент-сервер применяется в большом числе сетевых технологий, используемых для доступа к различным сетевым сервисам. Кратко рассмотрим некоторые типы таких сервисов (и серверов).

Web-серверы

Изначально представляли доступ к гипертекстовым документам по протоколу HTTP (Huper Text Transfer Protocol). Сейчас поддерживают расширенные возможности, в частности работу с бинарными файлами (изображения, мультимедиа и т.п.).

Серверы приложений

Предназначены для централизованного решения прикладных задач в некоторой предметной области. Для этого пользователи имеют право *запускать серверные программы* на исполнение. Использование серверов приложений позволяет снизить требования к конфигурации клиентов и упрощает общее управление сетью.

Серверы баз данных

Серверы баз данных используются для обработки пользовательских запросов на языке SQL. При этом СУБД находится на сервере, к которому и подключаются клиентские приложения.

Файл-серверы

Файл-сервер *хранит* информацию в виде файлов и представляет пользователям доступ к ней. Как правило файл-сервер обеспечивает и определенный уровень защиты от несакционированного доступа.

Прокси-сервер

Во-первых, действует как посредник, помогая пользователям получить информацию из Интернета и при этом обеспечивая защиту сети.

Во-вторых, сохраняет часто запрашиваемую информацию в кэш-памяти на локальном диске, быстро доставляя ее пользователям без повторного обращения к Интернету.

Файрволы (брандмауэры)

Межсетевые экраны, анализирующие и фильтрующие проходящий сетевой трафик, с целью обеспечения безопасности сети.

Почтовые серверы

Представляют услуги по отправке и получению электронных почтовых сообщений.

Серверы удаленного доступа (RAS)

Эти системы обеспечивают связь с сетью по коммутируемым линиям. Удаленный сотрудник может использовать ресурсы корпоративной ЛВС, подключившись к ней с помощью обычного модема.

Это лишь несколько типов из всего многообразия клиент-серверных технологий, используемых как в локальных, так и в глобальных сетях.

Для доступа к тем или иным сетевам сервисам используются клиенты, возможности которых характеризуются понятием «толщины». Оно определяет конфигурацию оборудования и программное обеспечение, имеющиеся у клиента. Рассмотрим возможные граничные значения:

«Тонкий» клиент

Этот термин определяет клиента, вычислительных ресурсов которого достаточно лишь для запуска необходимого сетевого приложения через web-интерфейс. Пользовательский интерфейс такого приложения формируется

средствами *статического* HTML (выполнение JavaScript не предусматривается), вся прикладная логика выполняется на сервере. Для работы тонкого клиента достаточно лишь обеспечить возможность запуска web-браузера, в окне которого и осуществляются все действия. По этой причине web-браузер часто называют "универсальным клиентом".

«Толстый» клиент

Таковым является рабочая станция или персональный компьютер, работающие под управлением собственной дисковой операционной системы и имеющие необходимый набор программного обеспечения. К сетевым серверам «толстые» клиенты обращаются в основном за дополнительными услугами (например, доступ к web-серверу или корпоративной базе данных). Так же под «толстым» клиентом подразумевается и клиентское сетевое приложение, запущенное под управлением локальной ОС. Такое приложение совмещает компонент представления данных (графический пользовательский интерфейс ОС) и прикладной компонент (вычислительные мощности клиентского компьютера).

В последнее время все чаще используется еще один термин: «rich»-client. «Rich«-клиент своего рода компромисс между «толстым» и «тонким» клиентом. Как и «тонкий» клиент, «rich»-клиент также представляет графический интерфейс, описываемый уже средствами XML и включающий некоторую функциональность толстых клиентов (например, интерфейс drag-and-drop, вкладки, множественные окна, выпадающие меню и т.п.)

Прикладная логика «rich»-клиента также реализована на сервере. Данные отправляются в стандартном формате обмена, на основе того же XML (протоколы SOAP, XML-RPC) и интерпретируются клиентом.

Некоторые основные протоколы «rich»-клиентов на базе XML приведены ниже:

XAML (eXtensible Application Markup Language) — разработан Microsoft, используется в приложениях на платформе .NET;

XUL (XML User Interface Language) — стандарт, разработанный в рамках проекта Mozilla, используется, например, в почтовом клиенте Mozilla Thunderbird или браузере Mozilla Firefox;

Flex — мультимедийная технология на основе XML, разработанная Macromedia/Adobe.

Заключение

Итак, *основная идея архитектуры «клиент-сервер» состоит в разделении сетевого приложения на несколько компонентов*, каждый из которых реализует специфический набор сервисов. Компоненты такого приложения могут выполняться на разных компьютерах, выполняя серверные и/или клиентские функции. Это позволяет повысить надежность, безопасность и производительность сетевых приложений и сети в целом.

Задание: Согласно выбранной индивидуальной темы определитесь с требуемой клиент-серверной архитектурой и клиент-серверной технологией. Аргументируйте свой выбор. Составьте требуемые схемы согласно выбранной модели.

Контрольные вопросы:

1. В чем заключается основная идея К-С взаимодействия?
2. В чем отличия между понятиями «клиент-серверная архитектура» и «клиент-серверная технология»?
3. Перечислите компоненты К-С взаимодействия.
4. Какие задачи выполняет компонент представления в К-С архитектуре?
5. С какой целью средства доступа к БД представлены в виде отдельного компонента в К-С архитектуре?
6. Для чего бизнес-логика выделена как отдельный компонент в К-С архитектуре?
7. Перечислите модели клиент-серверного взаимодействия.

8. Опишите модель «файл-сервер».
9. Опишите модель «сервер БД».
10. Опишите модель «сервер приложений»
11. Опишите модель «сервер терминалов»
12. Перечислите основные типы серверов.

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №32

Тема: «Обеспечение непротиворечивости и целостности баз данных в SQL Server 2016 Express»

Цель: Познакомиться с принципами и правилами обеспечения непротиворечивости и целостности баз данных в SQL Server 2016 Express. Научиться применять их на практике при проектировании БД.

Время выполнения: 4 урока

Ход работы:

Целостность данных - система правил, используемых для поддержания связей между записями в связанных таблицах, обеспечивающих защиту от случайного удаления или изменения данных

Второстепенные термины

• обеспечение целостности данных;

• ограничения целостности данных.

Структурная схема терминов

Обеспечение непротиворечивости и целостности данных

Целостность данных - система правил, используемых для поддержания связей между записями в связанных таблицах, обеспечивающих защиту от случайного удаления или изменения данных.

Обеспечение целостности БД - система мер, направленных на поддержание правильности данных в БД в любой момент времени.

Целостность данных обеспечивается набором специальных предложений, называющихся ограничениями целостности.

- это набор определенных правил, которые устанавливают допустимость данных и связей между ними. Ограничения целостности могут

относиться к разным объектам БД: атрибутам (полям), записям, отношениям, связям между ними.

Для полей могут быть использованы следующие виды ограничений:

1. Тип и формат поля автоматически допускают ввод только данных определенного типа.

2. Задание диапазона значений, как правило, используется для числовых полей. Диапазон значений может быть ограничен с двух сторон (закрытый диапазон), а может с какой-то одной: верхней или нижней (открытый диапазон).

3. Недопустимость пустого поля. Позволяет избежать появления в БД недописанных записей, в которых пропущены какие-либо обязательные атрибуты.

4. Задание списка значений, позволяет избежать излишнего разнообразия данных, если его можно ограничить.

5. Проверка на уникальность значения какого-либо поля, позволяет избежать появления записей-дубликатов.

Ограничения для связей

Пусть имеются отношение между двумя таблицами. Одна из таблиц будет являться главной (родительской), а вторая подчиненной (дочерней или связанной).

При отношениях "один ко многим" или "многие к одному" таблица с отношением

"один" всегда будет родительской. Поле, по которому происходит связь в дочерней таблице, называется внешним ключом.

1. Невозможно ввести в поле внешнего ключа связанной таблицы значение, не содержащееся в ключевом поле родительской таблицы.

Пример

В таблицу "Заказы" невозможно ввести "Код товара", которого нет в таблице

"Товары" (код несуществующего товара).

2. Не допускается удаление записи из главной таблицы, если существуют связанные с ней записи в подчиненной таблице.

3. Невозможно изменить значение ключевого поля в главной таблице, если существуют записи, связанные с данной.

Задание: Согласно выбранной индивидуальной темы спроектировать структуру БД, соблюдая все требования к целостности данных. Создать ER диаграмму. Привести все таблицы к третьей нормальной форме. Определить связи между таблицами.

Контрольные вопросы:

1. Дайте определение целостности данных.
2. Что понимают под ограничениями целостности?
3. Какие ограничения целостности применяются для полей?
4. Перечислите ограничения целостности для связей.

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №33

Тема: «Этапы разработки баз данных в SQL Server 2016 Express»

Цель: Научиться проектировать БД и создавать дорожную карту разработки и внедрения БД с использованием СУБД SQL Server 2016 Express.

Время выполнения: 8 уроков

Ход работы:

При разработке БД можно выделить следующие этапы работы.

I этап. Постановка задачи.

На этом этапе формируется задание по созданию БД. В нем подробно описывается состав базы, назначение и цели ее создания, а также перечисляется, какие виды работ предполагается осуществлять в этой базе данных (отбор, дополнение, изменение данных, печать или вывод отчета и т. д).

II этап. Анализ объекта.

На этом этапе рассматривается, из каких объектов может состоять БД, каковы свойства этих объектов. После разбиения БД на отдельные объекты необходимо рассмотреть свойства каждого из этих объектов, или, другими словами, установить, какими параметрами описывается каждый объект. Все эти сведения можно располагать в виде отдельных записей и таблиц. Далее необходимо рассмотреть тип данных каждой отдельной единицы записи. Сведения о типах данных также следует занести в составляемую таблицу.

III этап. Синтез модели.

На этом этапе по проведенному выше анализу необходимо выбрать определенную модель БД. Далее рассматриваются достоинства и недостатки каждой модели и сопоставляются с требованиями и задачами создаваемой БД. После такого анализа выбирают ту модель, которая сможет максимально

обеспечить реализацию поставленной задачи. После выбора модели необходимо нарисовать ее схему с указанием связей между таблицами или узлами.

IV этап. Выбор способов представления информации и программного инструментария.

После создания модели необходимо, в зависимости от выбранного программного продукта, определить форму представления информации.

В большинстве СУБД данные можно хранить в двух видах:

с использованием форм;

без использования форм.

Форма – это созданный пользователем графический интерфейс для ввода данных в базу.

V этап. Синтез компьютерной модели объекта.

В процессе создания компьютерной модели можно выделить некоторые стадии, типичные для любой СУБД.

Стадия 1. Запуск СУБД, создание нового файла базы данных или открытие созданной ранее базы.

Стадия 2. Создание исходной таблицы или таблиц.

Создавая исходную таблицу, необходимо указать имя и тип каждого поля. Имена полей не должны повторяться внутри одной таблицы. В процессе работы с БД можно дополнять таблицу новыми полями. Созданную таблицу необходимо сохранить, дав ей имя, уникальное в пределах создаваемой базы.

При проектировании таблиц, рекомендуется руководствоваться следующими основными принципами:

1. Информация в таблице не должна дублироваться. Не должно быть повторений и между таблицами. Когда определенная информация хранится только в одной таблице, то и изменять ее придется только в одном месте. Это

делает работу более эффективной, а также исключает возможность несовпадения информации в разных таблицах. Например, в одной таблице должны содержаться адреса и телефоны клиентов.

2. Каждая таблица должна содержать информацию только на одну тему. Сведения на каждую тему обрабатываются намного легче, если они содержатся в независимых друг от друга таблицах. Например, адреса и заказы клиентов лучше хранить в разных таблицах, с тем, чтобы при удалении заказа информация о клиенте осталась в базе данных.

3. Каждая таблица должна содержать необходимые поля. Каждое поле в таблице должно содержать отдельные сведения по теме таблицы. Например, в таблице с данными о клиенте могут содержаться поля с названием компании, адресом, городом, страной и номером телефона. При разработке полей для каждой таблицы необходимо помнить, что каждое поле должно быть связано с темой таблицы. Не рекомендуется включать в таблицу данные, которые являются результатом выражения. В таблице должна присутствовать вся необходимая информация. Информацию следует разбивать на наименьшие логические единицы (Например, поля "Имя" и "Фамилия", а не общее поле "Имя").

4. База данных должна иметь первичный ключ. Это необходимо для того, чтобы СУБД могла связать данные из разных таблиц, например, данные о клиенте и его заказы.

Стадия 3. Создание экранных форм.

Первоначально необходимо указать таблицу, на базе которой будет создаваться форма. Ее можно создавать при помощи мастера форм, указав, какой вид она должна иметь, или самостоятельно. При создании формы можно указывать не все поля, которые содержит таблица, а только некоторые из них. Имя формы может совпадать с именем таблицы, на базе которой она создана. На основе одной таблицы можно создать несколько форм, которые могут отличаться видом или количеством используемых из данной таблицы

полей. После создания форму необходимо сохранить. Созданную форму можно редактировать, изменяя местоположение, размеры и формат полей.

Стадия 4. Заполнение БД.

Процесс заполнения БД может проводиться в двух видах: в виде таблицы и в виде формы. Числовые и текстовые поля можно заполнять в виде таблицы, а поля типа MEMO и OLE – в виде формы.

VI этап. Работа с созданной базой данных.

Работа с БД включает в себя следующие действия:

- поиск необходимых сведений;
- сортировка данных;
- отбор данных;
- вывод на печать;
- изменение и дополнение данных.

Задание: Согласно предоставленной темы составить дорожную карту проектирования, разработки и внедрения БД. Составить сопутствующие схемы, таблицы и диаграммы для этапов с первого по четвертый. Аргументировать принятые решения.

Контрольные вопросы:

1. Назовите два подхода применяющиеся при построении БД и дайте их характеристику.
2. Назовите уровни архитектуры СУБД.
3. Какие режимы работы СУБД Вы знаете?
4. Этапы проектирования БД.
5. Кратко расскажите, что происходит на каждом этапе проектирования БД.
6. Понятие транзакции.
7. Основные причины низкой эффективности проектируемых БД.
8. Чем отличается проектирование БД от разработки БД.
9. Назовите этапы разработки БД.

10. Кратко расскажите, что происходит на каждом этапе разработки БД.

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №34

Тема: «Проектирование на основе модели типа объект – отношение в SQL Server 2016 Express»

Цель: Научиться проектировать БД в SQL Server 2016 Express на основе модели типа объект – отношение.

Время выполнения: 8 уроков

Ход работы:

Общий подход

Общий подход к проблеме семантического моделирования характеризуется четырьмя основными этапами.

1. Прежде всего следует попытаться задать множество *семантических* концепций (понятий), которые могут быть полезны при неформальном обсуждении реального мира.

• Например, можно согласиться с тем, что мир состоит из объектов. (Хотя невозможно с заданной точностью утверждать, что является объектом, а что — нет. Несмотря на это, концепция объектов весьма полезна для описания реального мира, по крайней мере с интуитивной точки зрения.)

• Развивая эту концепцию, можно допустить, что объекты образуют типы объектов. Например, можно предположить, что все отдельные сотрудники являются экземплярами некоторого типа объекта EMPLOYEE (сотрудник). Преимущество такой группировки заключается в том, что все объекты определенного типа будут обладать некоторыми общими свойствами (например, получение сотрудниками зарплаты), а потому такая группировка может привести к некоторой (и совершенно очевидной) *экономии представления.* В реляционных терминах, например, общность выражается в заголовке отношения.

• Более того, можно согласиться с тем, чтобы каждый объект имел некое особое свойство, которое служит для *идентификации* этого объекта, т.е. с тем, чтобы каждый объект обладал идентичностью.

• Наконец, можно также предположить, что каждый объект может быть связан с другими объектами с помощью отношений.

И т.д. Однако здесь следует заметить, что все эти термины (экземпляр объекта, тип объекта, свойство, отношение и т.д.) определены *не* совсем точно или строго, поскольку некоторые из них являются всего лишь семантическими, а не формальными терминами. Например, первый термин не является формальным, а второй, третий и четвертый термины — являются.

2. Далее нужно определить множество соответствующих *символических* (т.е. формальных) объектов, которые могут быть использованы для представления описанных выше семантических концепций. Например, в расширенной реляционной модели RM/T [12.5] предлагается несколько особых видов отношений, которые называются *E-* и *P-отношениями.* Грубо говоря, E-отношения (от entity-relations) представляют объекты, а P-отношения (от property-relations) — свойства, однако E- и P-отношения, конечно, имеют некоторые формальные определения, тогда как объекты и свойства их не имеют.

3. Кроме того, нужно вывести множество формальных правил целостности для работы с такими формальными объектами. Например, модель типа RM/T содержит правило целостности под названием *целостность свойств,* которое утверждает, что для каждого элемента P-отношения должен существовать соответствующий ему элемент E-отношения (таким образом отражается тот факт, что каждое свойство должно быть свойством некоторого объекта).

4. Наконец, следует задать множество формальных операторов для манипулирования этими формальными объектами. Например, в модели RM/T присутствует оператор *PROPERTY,* который можно использовать с E-

отношением и всеми соответствующими Р-отношениями таким образом, чтобы собрать вместе все свойства данного объекта.

Объекты, правила и операторы, описанные в пунктах 2—4, вместе образуют расширенную модель данных, если эти конструкции действительно являются супермножеством конструкций одной из базовых моделей, таких как, например, базовая реляционная модель. Хотя на самом деле в таком контексте нет четкого различия между расширенной и базовой моделью. *Однако здесь следует обратить пристальное внимание на то, что правила и операторы являются такой же частью расширенной модели, как и объекты* (точно так же, как это организовано в реляционной модели). Тем не менее, с точки зрения проектирования баз данных операторы являются менее важной частью модели по сравнению с объектами и правилами целостности. Поэтому за исключением нескольких комментариев, посвященных операторам, основное внимание в этой главе уделяется не операторам, а объектам и правилам.

В пункте 1 упоминалась попытка определения множества семантических концепций, которые были бы полезны при описании реального мира. Некоторые из них, а именно объекты, свойства, связи и подтипы, показаны на рис. 12.1. Для каждой из этих концепций на рисунке приведено неформальное определение и несколько типичных примеров. Обратите внимание, что все эти специально подобранные примеры иллюстрируют возможность рассматривать один и тот же предмет реального мира одними пользователями как объект, другими — как свойство, а третьими — как отношение. (Этот пример прекрасно демонстрирует, почему невозможно дать строгое определение такого термина, как "объект".) Основная цель семантического моделирования (еще не совсем достигнутая) как раз и заключается в том, чтобы поддерживать такую *гибкость интерпретации.*

Понятие	Неформальное	Примеры

	определение	
ОБЪЕКТ	Различимые объекты	Поставщик, товар, поставка, сотрудник, отдел, человек, произведение, концерт, оркестр, дирижер, заказ, ассортимент заказов

СВОЙСТВО	Информация, описывающая объект	Номер поставщика, количество поставок, отдел сотрудника, рост человека, вид концерта, дата заказа
ОТНОШЕНИЕ	Объект, который служит для организации взаимодействия двух или нескольких других объектов	Поставка (поставщик-товар), назначение (сотрудник-отдел)
ПОДТИП	Тип объекта Y является подтипом типа объекта X тогда и только тогда, когда каждый объект типа Y обязательно является объектом типа X	Тип "сотрудник" является подтипом типа "человек"

Рис. 1. Некоторые полезные семантические концепции

В заключение следует заметить, что в предыдущих главах отношения рассматривались как своего рода отдельные объекты, причем подчеркивалось, что они таким же образом будут рассматриваться во всей

книге. Кроме того, в качестве одного из преимуществ реляционной модели отмечалось единство представления всех объектов, включая зависимости между объектами, на основе отношений. Тем не менее, концепция отношений (так же, как и концепция объектов) действительно *интуитивно* полезна при описании реального мира; более того, представленный ниже в этой главе подход будет в значительной степени опираться на различия между объектами и отношениями. Поэтому в нескольких следующих разделах будет принята терминология на основе отношений, а некоторые другие более подробные сведения на эту тему будут добавлены позже.

Обзор модели объект/отношение

Вернемся к проблеме применения идей семантического моделирования для проектирования баз данных. Как уже упоминалось выше, одним из наиболее известных и широко распространенных подходов является так называемый подход типа объект/отношение— О/О. Этот подход был основан на предложенной в 1976 году Ченом «модели объект/отношение" и с тех пор неоднократно усовершенствовался Ченом и многими другими исследователями (об этом можно прочесть, например, в). Дальнейшее обсуждение в этой главе в основном посвящено именно этому подходу. (Следует подчеркнуть, что эта модель является далеко не единственной "расширенной" моделью; их существует достаточно большое количество, например, в приведены общие вводные сведения по некоторым из них, а в — вводные обзоры по данной тематике.)

Модель объект/отношение содержит аналоги всех семантических объектов, которые перечислены на рис. 2.1 и более подробно обсуждаются далее в этой главе. (Следует сразу же сказать, что, несмотря на некоторые различия в терминологии, большинство обсуждаемых идей имеет много общего с некоторыми чертами реляционной модели.)

Замечание. В 3 представлена не только сама модель объект/отношение, но и соответствующая ей диаграммная техника или "О/О-диаграммы". Более подробно 0/0-диаграммы описаны ниже, а на рис. 2 показан пример такой диаграммы. Этот пример с представлением структуры данных компании описан ниже и будет полезен при изучении последующего материала этой главы.

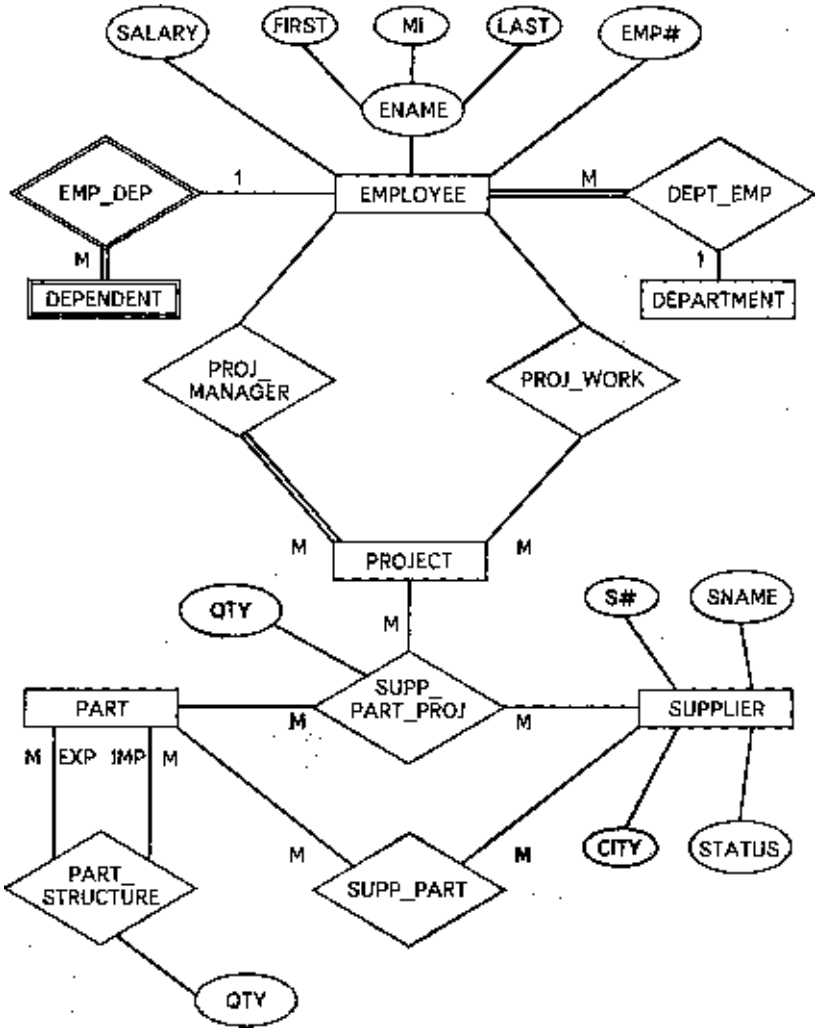


Рис. 2. Диаграмма модели объект/отношение

Объекты

Согласно определению, данному Ченом , объектом называется "предмет, который может быть четко идентифицирован". При этом объекты подразделяются на правильные объекты и слабые объекты. Слабым объектом называется объект, который находится в зависимости от некоторого другого объекта, т.е. он *не может* существовать, если не существует этот другой объект. Например, на рис. 12.2 объекты DEPENDENT (подчиненный

сотрудник) являются слабыми, поскольку они не могут существовать (в контексте данной базы данных), если не существует соответствующий объект EMPLOYEE (сотрудник). В частности, если данный сотрудник (EMPLOYEE) будет удален, то все зависящие от него объекты (DEPENDENT) также будут удалены. Правильным объектом называется объект, который не является слабым: например, правильными объектами могут являться сотрудники. Некоторые авторы вместо термина "правильный объект" используют термин "сильный объект".

Свойства

Объекты (и отношения) обладают некоторыми свойствами (их также называют *атрибутами,* однако здесь этот термин применяться не будет, поскольку он уже имеет особое значение в реляционной модели). Все объекты одного типа обладают некоторыми общими свойствами: например, все сотрудники имеют номер, имя, зарплату и т.д. (В данном примере в качестве свойства сознательно не упоминается параметр "номер отдела", причина разъясняется несколько ниже.) Значения свойств каждого типа извлекаются из соответствующего множества значений, которое в реляционных терминах называется доменом. Ниже перечислены некоторые виды свойств и их особенности.

• Простое или составное свойство (например, составное свойство "имя сотрудника" может складываться из простых свойств "имя", "промежуточные инициалы" и "фамилия").

• Ключевое свойство, уникальное в некотором контексте (например, имя подчиненного для определенного сотрудника уникально только в контексте этого сотрудника).

• Однозначное или многозначное свойство (т.е. в данном случае разрешены повторяющиеся группы, однако эта концепция не представлена на рис. 12.2).

• Отсутствующее свойство (т.е. "неизвестное" или "неприменимое", эта концепция не представлена на рис. 2).

• Базовое или производное свойство (например, общее количество определенного товара может быть выведено на основе суммирования отдельных количеств поставок данного товара, эта концепция не представлена на рис. 2).

Отношения

В 3 отношение определяется как "ассоциация объектов". Например, между отделами и сотрудниками задано отношение (DEPT_EMP), которое означает, что в данном отделе работает заданное множество сотрудников. Так же, как и в случае с объектами, необходимо видеть принципиальную разницу между *типами* и *экземплярами* отношений, однако при неформальном описании такими тонкостями можно пренебречь.

Объекты, включенные в данное отношение, называются участниками этого отношения, а количество участников данного отношения называетсястепенью этого отношения. Следует отметить, что в данном случае значение термина "степень" несколько отличается от его значения в реляционной модели.

Пусть *R* является типом отношения, которое содержит тип объекта *E* в качестве участника. Если *каждый* экземпляр типа объекта *E* находится по крайней мере в одном экземпляре отношения R, то участие *E* в отношении *R* называется полным, в противном случае — частичным. Например, если каждый товар должен поставляться по крайней мере одним поставщиком, то участие товаров в отношении между поставщиками и товарами (SUPP_PART) является полным. Но если допустима такая ситуация, когда некоторый товар не поставляется ни одним из поставщиков, то участие товаров в отношении SUPP_PART является частичным.

Отношения в модели объект/отношение могут иметь тип один-к-одному, один-ко-многим, многие-к-одному или многие-ко-многим (для упрощения изложения далее предполагается, что все отношения бинарные, т.е. степени два, хотя, конечно, изложение этих концепций и терминологию

можно непосредственно расширить и на отношения более высокой степени). Здесь читатель, знакомый с основами реляционной модели, мог бы заметить, что случай отношения многие-ко-многим является единственным истинным отношением, и для этого случая требуется создать отдельное представление (отношения типа один-к-одному и один-ко-многим всегда могут быть представлены с помощью внешнего ключа). Однако в некоторых случаях отношения типа один-к-одному и один-ко-многим следует рассматривать так же, как отношение типа многие-ко-многим. По крайней мере, если существует какая-нибудь возможность расширить их спустя некоторое время до отношения типа многие-ко-многим. И только если такой возможности нет, их следует рассматривать иначе. (Конечно, порой такой возможности может не быть вовсе; например, всегда остается верным утверждение, что окружность обладает только одной точкой, которая является ее центром.)

Подтипы

Замечание. Идеи, представленные в этом разделе, не были включены в оригинальную модель объект/отношение 3, а были добавлены позднее. Подробнее об этом можно прочесть, например, в работе Теори, Янга и Фрая (Teorey, Yang, Fry).

Каждый объект имеет по крайней мере один тип, однако у определенного объекта может быть несколько подтипов. Например, если некоторые сотрудники — программисты (а все программисты являются сотрудниками), то можно сказать, что тип PROGRAMMER (программист) является подтипом супертипа EMPLOYEE (сотрудник). Программисты *автоматически* обладают всеми свойствами сотрудников, однако обратное утверждение не верно (например, для программистов может быть задано свойство "язык программирования, которым программист владеет в наибольшей мере", в общем случае не применимое ко всем сотрудникам). Аналогично, программисты автоматически участвуют во всех отношениях, в которых участвуют сотрудники, однако обратное утверждение не верно (например, программисты могут входить в профессиональное

компьютерное общество, в которое в общем случае сотрудники не входят). О таких свойствах и отношениях супертипа говорят, что они наследуются подтипом. (Наследование будет обсуждаться более подробно в последующих главах этой книги.)

Обратите внимание, что некоторые программисты (PROGRAMMER) могут быть прикладными программистами (APPLICATION_PROGRAMMER), а некоторые — системными (SYSTEM_PROGRAMMER). Таким образом, можно сказать, что типы APPLICATION_PROGRAMMER и SYSTEM_PROGRAMMER являются подтипами супертипа PROGRAMMER (и т.д.). Иначе говоря, подтип объекта все еще является *типом* объекта и может иметь собственные подтипы. Данный тип объекта, его непосредственные подтипы, подтипы подтипов и т.д. вместе образуют иерархию типов для данного типа объекта, пример которой показан на рис. 3.

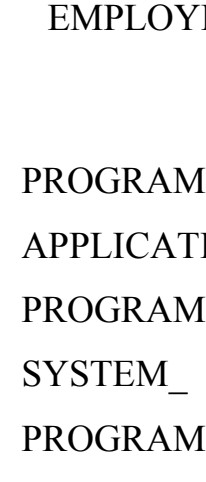


Рис. 3. Пример иерархии типов

Для читателей, знакомых с какой-нибудь системой, в которой поддерживается иерархическая структура данных, необходимо отметить, что иерархии типов не следует путать с иерархиями данных. На рис. 12.3, например, вовсе не подразумевается, что один сотрудник (EMPLOYEE) может иметь несколько соответствующих программистов (PROGRAMMER). Наоборот, для одного экземпляра типа EMPLOYEE существует *максимум* один соответствующий программист (PROGRAMMER),

представляющий того же самого сотрудника (EMPLOYEE) в роли программиста. На этом краткое обсуждение основных структурных особенностей модели объект/отношение завершено и можно перейти к рассмотрению диаграмм типа объект/отношение.

Диаграммы объект/отношение

Как уже объяснялось в предыдущем разделе, в 3 не только введена сама модель объект/отношение, но также и представлена концепция диаграммы объект/отношение (0/0-диаграмма). Такая диаграмма (также называется схемой) является методом представления логической структуры базы данных в графическом виде для более простого и понятного выражения основных компонентов макета базы данных (рисунок порой стоит тысячи слов). Действительно, популярность применения модели объект/отношение для проектирования баз данных скорее всего объясняется именно наличием диаграммной техники, чем какой-либо другой причиной. Далее правила создания 0/0-диаграмм разъясняются на примерах, приведенных на рис. 2 и 3.

Замечание. Так же, как и сама модель объект/отношение, диаграммная техника постоянно совершенствуется, поэтому в этом разделе будет описана ее версия, которая в некоторых важных отношениях отличается от оригинальной методики, предложенной Ченом .

Объекты

Каждый тип объекта показан в виде отдельного прямоугольника с его именем внутри, причем *слабые* типы объектов изображаются в *двойной* рамке. *Примеры* (см. рис. 2).

• Правильные объекты:

DEPARTMENT

EMPLOYEE

SUPPLIER

PART

PROJECT

• Слабые объекты:

DEPENDENT

Свойства

Свойства показаны в виде эллипсов с названием свойства, соединенных сплошной линией с соответствующим объектом (или отношением). Эллипс обведен штриховой линией, если свойство производное, и двойной линией, если свойство многозначное. Если свойство составное, то составляющие его свойства показаны в виде других эллипсов, соединенных с эллипсом составного свойства с помощью дополнительных линий. Ключевые свойства, как правило, подчеркиваются, а множества значений не показываются вовсе.

Примеры (см. рис. 2).

• Для EMPLOYEE:

EMP# (ключевое свойство)

ENAME (составное, состоящее из свойств FIRST, MI и LAST)

SALARY

• Для SUPPLIER:

S# (ключевое свойство)

SNAME

STATUS

CITY

• Для SUPP_PART_PROJ:

QTY

• Для PART_STRUCTURE:

QTY

Для экономии места другие свойства на рис. 2 не показаны.

Отношения

Каждый тип отношения показан в виде ромба с названием отношения внутри. Ромб окружен двойной линией, если отношение задано между слабым типом объекта и типом объекта, от существования которого находится в зависимости слабый тип объекта. Участники каждого отношения присоединены к соответствующему отношению сплошными линиями.

Каждая такая линия содержит надпись "1" или "M" для обозначения типа отношения, один-к-одному, один-ко-многим и т.д. Двойная линия обозначает полное участие.

Примеры (см. рис. 2).

• DEPT_EMP (отношение типа один-ко-многим между объектами DEPARTMENT и EMPLOYEE)

• EMP_DEP (отношение типа один-ко-многим между объектом EMPLOYEE и объектом слабого типа DEPENDENT)

• PROJ_WORK и PROJ_MANAGER (оба отношения между объектами EMPLOYEE и PROJECT, первое имеет тип многие-ко-многим, а второе — один-ко-многим)

• SUPP_PART_PROJ (отношение многие-ко-многим-ко-многим, включающее объекты SUPPLIER, PART и PROJECT)

• SUPP_PART (отношение многие-ко-многим между объектами SUPPLIER и PART)

• PART_STRUCTURE (отношение многие-ко-многим между объектами PART и PART)

Обратите внимание, что в последнем случае две линии от PART к PART_STRUCTURE отличаются своими надписями с указанием различных выполняемых "ролей" (EXP и IMP, которые обозначают соответственно "раскладывание товара по компонентам" и "складывание товара по компонентам"). Отношение PART_STRUCTURE является типичным примером рекурсивного отношения.

Подтипы и супертипы

Пусть У является подтипом *X*, тогда от У к *X* можно провести сплошную линию со стрелкой на конце для представления математического "подмножества" оператора (поскольку множество всех У является подмножеством множества всех X.

Примеры (см. рис. 3, хотя стрелки на нем не показаны).

• Тип PROGRAMMER является подтипом типа EMPLOYEE.

• Типы APPLICATION_PROGRAMMER и SYSTEM_PROGRAMMER являются подтипами типа PROGRAMMER.

Задание: Согласно предоставленной темы составить модель объект-отношение. Составить сопутствующие схемы, таблицы и диаграммы. Аргументировать принятые решения.

1. Выделить необходимый набор сущностей, отражающих предметную область и информационные потребности пользователей.

2. Определить необходимый набор атрибутов каждой сущности, выделив идентифицирующие атрибуты.

3. Классифицировать атрибуты каждой сущности (описательные, указывающие, вспомогательные).

4. Определить сущности вида подтип/супертип, где это необходимо.

5. Определить связи между сущностями.

6. Проанализировав структуру связей, исключить избыточные.

7. Определить множественность и условность связей.

8. Дать формулировку связей с точки зрения каждой участвующей сущности.

9. Формализовать связи вида 1:1, 1:M, M:N.

10. Классифицировать сущности, разделив их на стержневые, ассоциативные, характеристические, обозначающие.

11. Построить ER-диаграмму модели базы данных.

12. Описать модель базы данных на языке инфологического проектирования.

Контрольные вопросы:

1. Каковы задачи, решаемые на этапе инфологического проектирования?

2. В чем состоит отличие понятия типа сущности и элемента сущности?

3. Каковы способы представления сущности?

4. Каковы правила атрибутов?
5. Как классифицируются атрибуты?
6. Что такое безусловная, условная, биусловная, рекурсивная связь?
7. Каковы фундаментальные виды связей?
8. Как формализуется связь 1:1?
9. Как формализуется связь 1:M?
10. Как формализуется связь M:N?
11. Что такое подтип и супертип?
12. Что такое композиция связей?

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №35

Тема: «Нормализация баз данных в SQL Server 2016 Express»

Цель: Получить практический приведения БД к первой, второй и третьей нормальной форме в SQL Server 2016 Express.

Время выполнения: 8 уроков

Ход работы:

Нормализация — это процесс организации данных в базе данных, включающий создание таблиц и установление отношений между ними в соответствии с правилами, которые обеспечивают защиту данных и делают базу данных более гибкой, устраняя избыточность и несогласованные зависимости.

Избыточность данных приводит к непродуктивному расходованию свободного места на диске и затрудняет обслуживание баз данных. Например, если данные, хранящиеся в нескольких местах, потребуется изменить, в них придется внести одни и те же изменения во всех этих местах. Изменение адреса клиента гораздо легче реализовать, если в базе данных эти сведения хранятся только в таблице Customers и нигде больше.

Что такое «несогласованные зависимости»? Пользователь, которому нужно узнать, например, адрес определенного клиента, вполне обоснованно будет искать его в таблице Customers (клиенты), но искать в ней сведения о зарплате сотрудника, который работает с этим клиентом, не имеет смысла. Зарплата сотрудника связана с сотрудником (зависит от него), поэтому эти сведения следует хранить в таблице Employees (сотрудники). Несогласованные зависимости могут затруднять доступ к данным, так как путь к данным при этом может отсутствовать или быть неправильным.

Существует несколько правил нормализации баз данных. Каждое правило называется «нормальной формой». Если выполняется первое правило, говорят, что база данных представлена в «первой нормальной форме». Если выполняются три первых правила, считается, что база данных

представлена в «третьей нормальной форме». Есть и другие уровни нормализации, однако для большинства приложений достаточно нормализовать базы данных до третьей нормальной формы.

Как и в случае со многими другими формальными правилами и спецификациями, обеспечить полное соответствие реальным ситуациям не всегда возможно. Как правило, для выполнения нормализации приходится создавать дополнительные таблицы, и некоторые клиенты считают это нежелательным. Собираясь нарушить одно из первых трех правил нормализации, убедитесь в том, что в приложении учтены все связанные с этим проблемы, такие как избыточность данных и несогласованные зависимости.

В описаниях ниже приведены соответствующие примеры.

Первая нормальная форма

Устраните повторяющиеся группы в отдельных таблицах.

Создайте отдельную таблицу для каждого набора связанных данных.

Идентифицируйте каждый набор связанных данных с помощью первичного ключа.

Не используйте несколько полей в одной таблице для хранения похожих данных. Например, для слежения за товаром, который закупается у двух разных поставщиков, можно создать запись с полями, определяющими код первого поставщика и код второго поставщика.

Что произойдет при добавлении третьего поставщика? Добавление третьего поля нежелательно, так как для этого нужно изменять программу и таблицу, поэтому данный способ плохо адаптируется к динамическому изменению числа поставщиков. Вместо этого можно поместить все сведения о поставщиках в отдельную таблицу Vendors (поставщики) и связать товары с поставщиками с помощью кодов товаров или поставщиков с товарами с помощью кодов поставщиков.

Вторая нормальная форма

Создайте отдельные таблицы для наборов значений, относящихся к нескольким записям.

Свяжите эти таблицы с помощью внешнего ключа.

Записи могут зависеть только от первичного ключа таблицы (составного ключа, если необходимо). Возьмем для примера адрес клиента в системе бухгалтерского учета. Этот адрес необходим не только таблице Customers, но и таблицам Orders, Shipping, Invoices, Accounts Receivable и Collections. Вместо того чтобы хранить адрес клиента как отдельный элемент в каждой из этих таблиц, храните его в одном месте: или в таблице Customers, или в отдельной таблице Addresses.

Третья нормальная форма

Устраните поля, не зависящие от ключа.

Значения, входящие в запись и не являющиеся частью ключа этой записи, не принадлежат таблице. Если содержимое группы полей может относиться более чем к одной записи в таблице, подумайте о том, не поместить ли эти поля в отдельную таблицу.

Например, в таблицу Employee Recruitment (наем сотрудников) можно включить адрес кандидата и название университета, в котором он получил образование. Однако для организации групповой почтовой рассылки необходим полный список университетов. Если сведения об университетах будут храниться в таблице Candidates, составить список университетов при отсутствии кандидатов не получится. Таким образом, создайте вместо этого отдельную таблицу Universities и свяжите ее с таблицей Candidates при помощи ключа — кода университета.

Исключение. Выполнять нормализацию баз данных до третьей нормальной формы теоретически желательно, но не всегда практично. Например, для устранения всех возможных зависимостей между полями таблицы Customers придется создать отдельные таблицы для хранения сведений о городах, почтовых индексах, торговых представителях, категориях клиентов и любых других сведений, которые могут

дублироваться в нескольких записях. С теоретической точки зрения нормализация желательна. Однако значительное увеличение числа маленьких таблиц может привести к снижению производительности СУБД или исчерпанию памяти и числа дескрипторов открытых файлов.

Выполнять нормализацию до третьей нормальной формы может быть целесообразно только для часто изменяемых данных. Если при этом сохранятся зависимые поля, спроектируйте приложение так, чтобы при изменении одного из этих полей пользователь должен был проверить все связанные поля.

Другие формы нормализации

Кроме описанных нормальных форм, существует четвертая нормальная форма, которую также называют нормальной формой Бойса-Кодда (BCNF), и пятая нормальная форма, но на практике они применяются редко. Несоблюдение этих правил может привести к ухудшению архитектуры базы данных, но на функциональности это сказаться не должно.

Задание: Согласно разработанной ранее модели в практической работе №34 спроектировать структуру БД, соблюдая все требования к целостности данных. Привести все таблицы к третьей нормальной форме. Определить связи между таблицами. Составить сопутствующие схемы, таблицы и диаграммы. Аргументировать принятые решения.

Контрольные вопросы:

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №36

Тема: «Управление пользователями в Oracle»

Цель: Научиться создавать пользователь, группы пользователей и определять групповую политику в БД Oracle.

Время выполнения: 2 урока

Ход работы:

Администратор базы данных создает пользователей с помощью инструкции CREATE USER.

```
CREATE USER user
IDENTIFIED BY   password;
```

```
CREATE USER  demo
IDENTIFIED BY demo;
```

Администратор базы данных **создает пользователя** путем выполнения инструкции CREATE USER. На момент создания пользователь не имеет никаких привилегий. Затем администратор базы данных может предоставить этому пользователю определенные привилегии. Этими привилегиями определяется, что именно пользователь может делать на уровне базы данных.

На рисунке показан сокращенный синтаксис создания пользователя.

Элементы синтаксиса:

user имя создаваемого пользователя;

Password указывает, что пользователь должен входить в систему с этим паролем.

Более подробно это изложено в руководстве *Oracle Database 11g SQL Reference (Справочник по SQL для базы данных Oracle 11g)*.

Примечание. Начиная с базы данных Oracle 11g, в паролях учитывается регистр букв. При использовании в пароле прописные и строчные буквы различаются.

УПРАВЛЕНИЕ ПОЛЬЗОВАТЕЛЯМИ И ПРОФИЛЯМИ ORACLE

Просмотр информации о пользователях (Users Oracle):

SET LINESIZE 120

COLUMN USERNAME FORMAT A25

COLUMN ACCOUNT_STATUS FORMAT A20

SELECT USERNAME, PROFILE, ACCOUNT_STATUS, EXPIRY_DATE FROM DBA_USERS;

USERNAME PROFILE ACCOUNT_STATUS EXPIRY_DATE

SCOTT DEFAULT OPEN 12.06.17

ORACLE_OCM DEFAULT EXPIRED & LOCKED 06.07.15

OJVMSYS DEFAULT EXPIRED & LOCKED 06.07.15

SYSKM DEFAULT EXPIRED & LOCKED 06.07.15

Просмотр информации о профилях(Oracle Profiles):

SET LINESIZE 120

COLUMN PROFILE FORMAT A20

COLUMN RESOURCE_NAME FORMAT A25

COLUMN RESOURCE_TYPE FORMAT A20

COLUMN LIMIT FORMAT A20

SELECT PROFILE, RESOURCE_NAME, RESOURCE_TYPE, LIMIT FROM DBA_PROFILES;

PROFILE RESOURCE_NAME RESOURCE_TYPE LIMIT

DEFAULT CONNECT_TIME KERNEL UNLIMITED

DEFAULT PRIVATE_SGA KERNEL UNLIMITED

DEFAULT FAILED_LOGIN_ATTEMPTS PASSWORD 10

DEFAULT PASSWORD_LIFE_TIME PASSWORD 180

DEFAULT PASSWORD_REUSE_TIME PASSWORD UNLIMITED

DEFAULT PASSWORD_REUSE_MAX PASSWORD UNLIMITED

DEFAULT PASSWORD_VERIFY_FUNCTION PASSWORD NULL

Создание профиля и пользователя с этим профилем(Oracle Create Profile & Create User):

CREATE PROFILE password_prof LIMIT

FAILED_LOGIN_ATTEMPTS 6

PASSWORD_LIFE_TIME 60

PASSWORD_REUSE_TIME 60

PASSWORD_REUSE_MAX 5

PASSWORD_LOCK_TIME 1/24

PASSWORD_GRACE_TIME 10

PASSWORD_VERIFY_FUNCTION DEFAULT;

CREATE PROFILE app_user LIMIT

SESSIONS_PER_USER UNLIMITED

CPU_PER_SESSION UNLIMITED

CPU_PER_CALL 3500

CONNECT_TIME 50

LOGICAL_READS_PER_SESSION DEFAULT
LOGICAL_READS_PER_CALL 1200
PRIVATE_SGA 20K
COMPOSITE_LIMIT 7500000;

CREATE USER jfee
IDENTIFIED BY password
DEFAULT TABLESPACE example
TEMPORARY TABLESPACE temp
QUOTA 500K ON example
PROFILE app_user
CONTAINER = CURRENT;

Назначение профиля существующему пользователю (Oracle Assigning a Profile to a User):

ALTER USER psmith PROFILE app_user;

Сброс профиля (Oracle Dropping Profiles):

Профиль DEFAULT ***не может быть сброшен***. При сбросе профиля, который назначен пользователям, необходимо использовать опцию CASCADE.

DROP PROFILE clerk CASCADE;

Изменение значений профиля (Oracle Using the ALTER PROFILE Statement to Set Profile Limits):

ALTER PROFILE prof LIMIT
FAILED_LOGIN_ATTEMPTS 9
PASSWORD_LOCK_TIME 10;

ALTER PROFILE default LIMIT PASSWORD_LIFE_TIME **UNLIMITED**;

Значения параметров пароля в профиле (Oracle Password-Specific Settings in the Default Profile):

Задание: Выполнить запросы из приведенных выше примеров в СУБД Oracle. Проанализировать ход их выполнения и полученный результат.

Контрольные вопросы:

1. Для чего нужны пользователи и группы пользователей в СУБД Oracle?

2. Что такое профиль пользователя в СУБД Oracle?

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №37

Тема: «Принципы работы транзакций в Oracle»

Цель: Научиться создавать, вызывать и редактировать транзакции в БД Oracle.

Время выполнения: 2 урока

Ход работы:

Концепция транзакции лежит в основе реляционной парадигмы. Транзакция состоит из одной или нескольких DML команд и следующей командой или ROLLBACK или COMMIT. Возможно использовать команду SAVEPOINT для определённого управления внутри транзакции. Перед рассмотрением синтаксиса необходимо рассмотреть концепцию транзакций. Связанная с этой темой это тема согласованного чтения; это реализуется автоматически на уровне Oracle сервера, но некоторые программисты могут управлять им с помощью SELECT команд.

Транзакции БД

Механизм Oracle для обеспечения транзакционной целостности основан на сочетании сегментов отмены изменений и файла журнала логов: этот механизм бесспорно лучший из всех созданных на сегодняшний день и полностью удовлетворяет международным стандартам обработки данных. Производители других БД реализуют стандарт своими собственными другими способами. Вкратец, любая реляционная база данных должна удовлетворять тесту ACID: должны быть гарантированы атомарность (A – atomicity), согласованность (C – consistency), изолированность (I – isolation) и долговечность (D – durability).

Атомарность

Принцип атомарности гласит что либо все части транзакции должны быть выполнены успешны либо ни одна из них. Например если бизнес-аналитик утвердил правило что при смене зарплаты сотрудника обязательно изменяется уровень сотрудника то ваша атомарная транзакция будет сосять

из двух частей. БД должна гарантировать что будут применены либо оба изменения, либо ни одного. Если только одно изменения будет успешно то у вас появится сотрудник чья зарплата несовместима с его уровнем: повреждение данных в терминах бизнеса. Если что-нибудь (вообще что-нибудь) пошло не так до подтверждения транзакции, БД должна гарантировать что вся работа совершённая до этого момента от начала транзакции будет отменена: это должно работать автоматически. Несмотря на то что атомарность транзакции звучит как что-то маленькое – транзакции могут быть долгими и очень важными. Рассмотрим другой пример, в бухгалтерской книге не может быть данных на пол-месяца Августа и пол-месяца Сентября: закрытие месяца с точки зрения бизнеса одна атомарная транзакция, которая может обрабатывать миллионы строк и тысячи таблицы и работать несколько часов (или отменяться если что-то пошло не так). Отмена транзакции может быть ручной (выполнив команду ROLLBACK) но она должна быть автоматической и неотменяемой в случае ошибки.

Согласованность

Принцип согласованности данных гласит что результат запроса должен быть согласован с состояним базы данных на момент старта работы запроса. Преставим простой запрос которые считает среднее значение столбца в таблице. Если таблица большая, это займёт достаточно долгое время для прохода по всем строкам таблицы. Если другие пользователи в это время обновляют данные пока запрос выполняется, должен ли запрос брать новые значения или старые? Должен ли результат запроса учитывать строки которые были добавлены или не учитывать строки которые были удалены? Принцип согласованности требует чтобы БД гарантировала что любые изменения после старта запроса не были видны для этого запроса; запрос должен вернуть среднее значение столбца на момент когда запрос был запущен, вне зависимости от того как долго длился запрос и какие изменения были над данными. Оракл гарантирует что если запрос выполнен успешно – результат будет согласованным. Как бы там ни было, если администратор

базы данных не настроил базу данных соотвествующим образом, запрос может не выполнится: возникнет знаменитая ошибка “ORA-1555 snapshot too old”. Раньше было очень сложно решить такие ошибки, но в последних версиях администратор легко может решать эти ситуации.

Изолированность

Принцип изолированности гласит что незаконченная (неподтверждённая транзакция) должна быть невидима для остального мира. Пока транзакция в процессе только сессия которая выполняет эту транзакцию видит зименения. Все остальные сессии должны видеть неизменённые данные. Почему так? Во первых, транзакция может целиком не выполниться до конца (помним про принцип атомарности и согласованности) и поэтому никто не должен видеть изменения которые могут быть отменены. Во вторых во время действия транзакции данные (в терминах бизнеса) бессвязные: для нашего примера обновления зарплаты будет существовать промежуток времени когда зарплата изменена, а уровень ещё нет. Изолированность транзакций требудет чтобы база данных прятала текущие транзакции от других пользователей: они будут видеть данные до изменений пока транзакция выполняется, а затем сразу будут видеть все изменения как согласованный набор данных. Oracle гарантирует изолированность транзакций: нет способа для сессии (отличной от той что делает изменения) увидеть неподтверждённые данные. Чтение неподтверждённых данных (известное как грязное чтение dirty read) не позволяется Oracle (несмотря на то что некоторые другие БД позволяют).

Долговечность

Принцип долговечности указывает на то, что если транзакция успешно завершена, то должно быть невозможно потерять эти данные. Во время работы транзакции принцип изолированности требует чтобы никто кроме сессии выполняющей изменения не видел их. Но как только транзакция успешно завершила работу, изменения должны стать доступны для всех и база данных должна гарантировать что они не будут потеряны. Oracle

выполняет это требование путём записывания всех векторов изменений в файлы логов перед тем как изменения подтверждены. Применив этот логи изменений к резервным копиям, всегда возможно повторить любые изменения которые были выполнены в момент остановки или повреждения базы данных. Конечно данные могут быть потеряны из-за ошибок пользователей, таких как выполнение неверных DML запросов или удаление таблиц. Но с точки зрения Oracle и администратора базы данных, такие события тоже транзакции: согласно принципу долговечности их нельзя отменить.

Выполнение SQL запросов

Весь язык SQL состоит из около дюжины команд. Сейчас нас интересуют команды: SELECT, INSERT, UPDATE и DELETE.

Выполнение команды SELECT

Команда SELECT получает данные. Выполнение команды SELECT это процесс состоящий из нескольких этапов: серверный процесс выполняющий запрос проверит существует ли необходимые блоки данных в памяти, в буфере кэша. Если они там, то выполнение может продолжаться, иначе серверный процесс должен найти данные на диске и скопировать их в буфер кэша.

EXAM TIP

Always remember that server processes read blocks from datafiles into the database buffer cache, DBWn writes blocks from the database buffer cache to the datafiles.

Когда блоки с нужными для выполнения запроса данными находятся в кэш буфере любые дополнительные процессы (такие как сортировка и аггрегирование) продолжаются в PGA сессии. Когда выполнение завершено, результат возвращается пользовательскому процессу.

Как это связано с тестом ACID? Для согласованности, если запрос обнаружит что блок данных изменился с момента старта запроса, серверный процесс найдёт сегмента отката (отмены изменений или сегмент undo)

соответствующий этому изменению, найдёт старую версию данных и (для текущего запроса) отменит изменение. Таким образом изменения которые произошли после начала запроса будут не видны. Похожим образом гарантируется изолированность транзакций, несмотря на то что изолированность основана и на подтверждённых изменениях. Честно говоря, если данные необходимые для отмены изменений не существуют больше в сегменте отката – этот механизм не сработает. Отсюда и следует ошибка “snapshot too old”.

На рисунке 8-4 показан путь обработки запроса SELECT

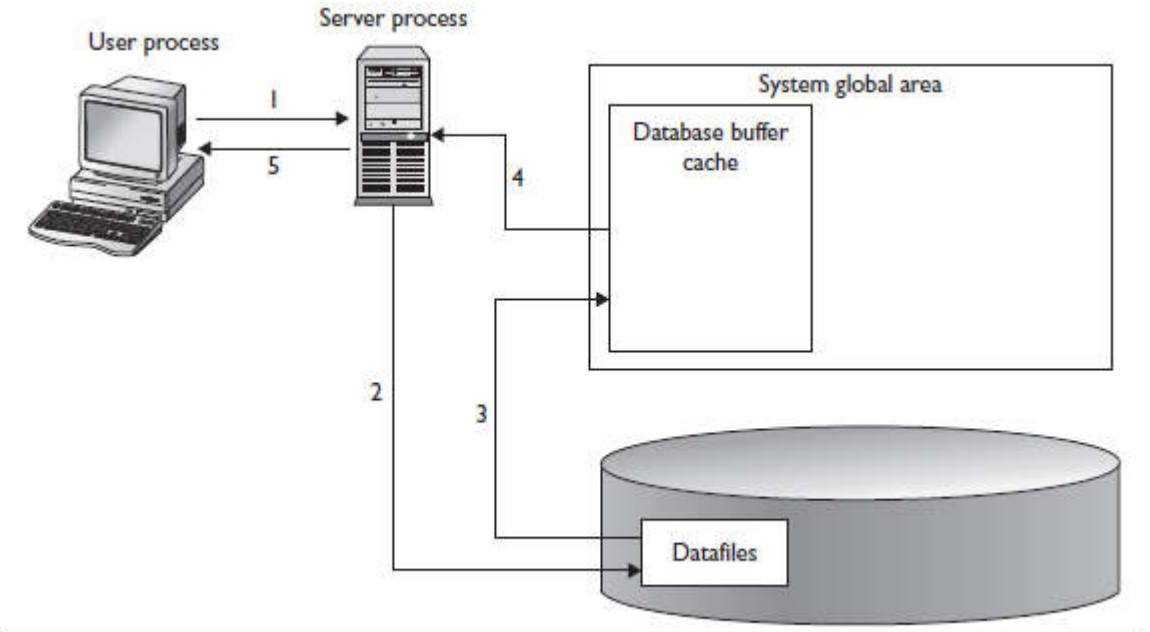


Figure 8-4 The stages of execution of a SELECT

Шаг 1 это передача пользовательского запроса от пользовательского процесса к серверному. Серверный процесс просматривает буфер кэш на наличие нужных блоков и если они в буфере то переходит к шагу4. Если нет то шаг 2 находит блоки в файлах данных и шаг 3 копирует данные в буфер. Шаг 4 передает данные серверному процессу где может быть дополнительная обработка перед тем как шаг 5 вернёт результат запроса пользовательскому процессу.

Выполнение команды UPDATE

Для любой команды DML необходимо работать с блоками данных и блоками отката (undo blocks), а также создавать лог изменений (redo): A,C и I

принципы теста ACIDS требуют создания данных отката; D требует создание данных повтора изменений (redo).

Exam tip

Undo не противоположна redo! Redo защищает все изменения блоков, вне зависимости это изменения блока таблицы, индекса или сегмента отката. Для redo — undo сегмент такой же сегмент как таблцы и все изменения должны быть долговечны (durable)

Первый шаг при выполнении DML команды такой же как и при выполнении команды SELECT: необходимые блоки должны быть найдены в кэф буфере или скопированы с файлов данных в буфер. Единственное отличие это то что дополнительно требуется пустой (или устаревший – expired) блок отката. Затем выполнение становится сложнее чем при команде SELECT.

Вначале блокировки должны быть указаны для все строк и соотвествующих индексов которые будут задействованы в процессе.

Потом создаются данные redo: серверный процесс записывае в логи буфера вектора изменений которые будут применены к данным. Redo данные создаются и для изменений блока данных и для изменений блока отката: если столбец в строке будет обновлен то rowid и новое значение записывается в буфер лога (изменение которое будет применено к блоку таблицы), а также старое значение столбца (изменение для блока отката). Если столбец это часть ключа индекса – то изменения в индексе тоже будут записаны в буфер лога, вместе с изменениями которые будут сделаны в блоке отката для защиты изменений индекса.

После того как все redo данные созданы, обновляются данные в буфер кэше: блок данных обновляется на новую версию с измененным столбцом, а старая версия записывается в блок отката. С этого момента до подтверждения транзакции, все запросы от других сессий обращающиеся к этой строке будут перенаправлены на блок отката. Только сессия которая

делает UPDATE будет видеть актуальную версию строки а блоке таблицы. Такой же принцип применяется для всех связанных индексов.

Выполнение команд INSERT и DELETE

Концептуально INSERT и DELETE управляются в той же манере как и UPDATE. Вначале происходит поиск нужных блоков в буфере и если их нет то они копируются в память.

Redo создается точно так же: все вектора изменений которые будут применены к данным и блокам отката вначале записываются в буфер лога. Для команды INSERT вектор изменений блока таблицы (и возможно блоков индекса) это байты которые составляют новую строку (и возможно новый ключ индекса). Вектор для блока отката это rowid новой строки. Для команды DELETE вектор для блока отката это вся строка.

Ключевым отличием между командами INSERT и UPDATE является количество данных для отката. Когда строка добавляется единственными данными для отката будет запись rowid в блок отката, потому что для отмены команды INSERT единственная информация нужная Oracle это rowid строки и может быть создана команда

delete from table_name where rowid=rowd_id_of_new_row;

Выполнение этой команды отменит изменение.

Для команды DELETE вся строка (которая может быть несколько килобайт) должна быть записана в блок undo, и тогда удаление может быть отменено при необходимости путём генерации запроса который заново добавил полностью строку в таблицу.

Начало и конец транзакции

Сессия начинает транзакция в момент когда она выполняет любую DML команду. Транзакция продолжается сколько угодно следующих DML команд пока сессия не выполнит команду ROLLBACK или COMMIT. Только подтверждённые изменения станут гарантированными и будут доступны для других сессий. Невозможно начать транзакцию внутри транзакции. Стандарт SQL не разрешает пользователям начать транзакцию, а затем начать новую перед завершение первой. Это можно сделать используя PL/SQL (язык Oracle третьего поколеняи), но не стандартным SQL.

Командами управления транзакциями являются команды COMMIT, ROLLBACK и SAVEPOINT. Также могут возникнуть другие обстоятельства помимо явного вызовая команды COMMIT или ROLLBACK которые немедленно прекращают транзакцию

Выполнение DDL или DCL команды

Завершение пользовательского процесса (к примеру пользователь вышел из программы SQL *Plus или SQL Developer)

Клиентская сессия «умерла»

Проблемы в системе

Если пользователь выполняет DDL команду (CREATE, ALTER или DROP) иди DCL команду (GRANT или REVOKE) то активная транзакция (если она сущесвтует) будет подтверждена. Так происходит потому что команды DDL и DCL сами являются транзакциями. Так как в SQL невозможно создать вложенные транзакции, если у пользователя уже выполнялась какая либо транзакция, все команды пользователя будут подтверждены вместе с командой DDL или DCL.

Если вы начали транзакцию выполнив DML запроса, а затем закрыли программу без явного указания COMMIT или ROLLBACK до выхода, транзакция будет прекращена – но прекращена с подтверждением или отменой целиком зависит от программы. У разных программ может быть разное поведение в зависимости от того как вы закончили работу в программе. Например в Windows обычно можно выйти из программы выбрав пункты меню File – Exit или нажав на крестик в правом верхнем углу. Программист мог обработать по разному эти способы завершения и в первом случае указать COMMIT, а во втором ROLLBACK. В любом случае это будет контролируемый выход.

Если клиентская сессия отказывает по какой-либо причине – база данных всегда отменит транзакцию. Такие отказы могут быть по разным причинам: пользовательский процесс мог быть «убит» диспетчером, проблемы с сетью или поломка пользовательской машины. В любом случае

не было явно указана команда COMMIT или ROLLBACK и БД нужно решить что случилось. В таком случае сессия «убивается» и активная транзакция отменяется. И точно так-же база данных ведёт в себя в случае проблем на стороне сервера. Если база данных была закрыта аварийно то при следующем старте все транзакции которые были начаты но явно не завершены будут отменены.

Управление транзакциями: COMMIT, ROLLBACK, SAVEPOINT и SELECT FOR UPDATE

Oracle начинает транзакцию в момент запуска первой DML команды. Транзакция длится до вызова команды ROLLBACK или COMMIT. Команда SAVEPOINT не является частью SQL стандарта и в реальности является легким способом для программиста чтобы отменить изменения частично в обратном порядке.

COMMIT

Выполнение команды COMMIT это тот момент когда многие люди (и даже некоторые администраторы БД) показывают непонимание архитектуры Oracle. Когда вы выполняете COMMIT всё что происходит физически это LGWR записывает буфер логов на диск. DBWn не делает абсолютно ничего. Это одно из самых важных свойств Oracle для достижения высокой производительность БД.

Exam tip

Что делает DBWn в момент выполнения команды COMMIT? Ответ: абсолютно ничего

Чтобы сделать транзакцию долговечной всё что нужно это записать изменения которые были сделаны в процессе транзакции на диск: нет необходимости в актуальных данных на диске. Если изменения записаны, в виде многих копий логов изменений на диске, то даже в случае повреждения базы все транзакции могут быт повторены восстановив резеврную копию данных до ошибки и применив изменения из логов. На данный момент надо понимать тот факт что COMMIT всего лишь очищает буфер лога на диск и

помечает транзакцию как выполненную. Вот почему транзакции в которой были задействованы миллионы обновлений в тысячах файлов в течение нескольких часов могут подветрждаться за долю секунды. Так как LGWR записывает логи практически в режиме реального времени, то виртуально все изменения транзакции уже записаны на диск. Когда вы выполняете COMMIT, LGWR тут же записывает лог на диск: ваша сессия будет ожидать пока запись не закончится. Время задержки будет равно времени которое занимает запись последних данных из буфера логов, что обычно занимает несколько миллисекунд. Потом ваша сессия может продолжать работу и все остальные сессии не будут перенаправлятьяс на данные в сегменте отката при обращении к обновлённым данным, если только принцип согласованности не требует этого. Вектора изменений, записываемые в лог повтора изменений, это все изменения: применяемых и к блокам данных (таблиц и индексов) и к блокам отката.

EXAM TIP

Лог redo включает все изменения: применяемые к сегментам данным и к сегментам undo для потдвержденных и неподтвержденных транзакций

Самое непонятное это то что redo записывается LGWR в файлы будет содержать и подтвержденные и неподтвержденные транзакции. Даже больше, в любой момент DBWn может записать а может и не записать измененные блоки сегментов данных или сегментов отката в файлы данных для подтверждённых и неподтверждённых транзакций. То есть ваша БД на диске противоречива: файлы данных могут хранить данные неподтверждённых транзакций и в них могут отсутствовать подтверждённые изменения. Но в любой момент, в случае проблемы, в файле логов на диске достаточно информации чтобы повторить подтверждённые транзакции которые пропущены в файлах данных (используя изменения для блоков данных) и восстановить сегменты отката (используя изменения блоков отката) нужные для отмены всех неподтверждённых транзакций которые записаны в файлы данных.

Exam tip

Лбая DDL команда, а также GRANT или REVOKE подтвердят текущую транзакцию

ROLLBACK

Пока транзакция в процессе, Oracle хранит образ данных до начала транзакции. Этот образ используется другими сессиями которые обращаются к данным участвующим в транзакции. Также он используется для отменты транзакции автоматически если что-то пойдёт не так или сессия отменит транзакцию.

Синтаксис для отмены транзакции

ROLLBACK [TO SAVEPOINT savepoint];

Состояние данных перед отменой транзакции содержит изменения, но информация нужная для отмены этих изменений доступна. Эта информация используется другими сессиями для выполнения принципа изолированности. ROLLBACK транзакции отменит все изменения восстановив образ данных до начала транзакции: все добавленные строки будут удалены, все удалённые строки восстановлены, все строки в которых менялись значения вернутся к исходному состоянию. Другие сессии даже не будут знать что что-то происходило, они никогда не видели изменений. А сессия которая инициировала транзакцию после отмены будет видеть данные такими какими они были до начала транзакции.

SAVEPOINT

Точка сохранения позволяет программистам устанавливать флаг в транзакции которые затем можно использовать для контроля эффекта отмены транзакции. Вместо отмены всей транзакции и её завершения, становится возможным отменить изменения сделанные после конкретного флага но оставить изменения сделанные до этого флага. Действие транзакции в этот момент продолжается: транзакция не подтверждена, всё ещё можно отменить транзакцию целиком и изменения не видны для других сессий.

Синтаксис команды

SAVEPOINT savepoint

Такая команда создаёт точку в транзакции которая может быть использована в дальнейшем в команде ROLLBACK. На следующей таблице видно количество строк в таблице видимое разным сессиям во время работы транзакции в разные моменты времени. Используемая таблица назвается TAB и у неё один столбец

Command	Rows Visible to the User	Rows Visible to Others
truncate table tab;	0	0
insert into tab values ('one');	1	0
savepoint first;	1	0
insert into tab values ('two');	2	0
savepoint second;	2	0
insert into tab values ('three');	3	0
rollback to savepoint second;	2	0
rollback to savepoint first;	1	0
commit;	1	1
delete from tab;	0	1
rollback;	1	1

В примере с выполнены две транзакции: первая завершена командой COMMIT а вторая ROLLBACK. Видно что использование точек сохранения влияет только внутри транзакции для той сессии которая инициировала транзакцию: вторая сессия не видит ничего что не подтверждено.

SELECT FOR UPDATE

Последняя команда для управления транзакциями это SELECT FOR UPDATE. Oracle, по умолчанию, предоставляет наивысший уровень параллелизма: чтение данных не блокирует запись, запись не блокирует изменение. Другими словами нет проблемы если одна сессия пытается считать данные которые другая сессия изменяет и наоборот. Но иногда вам может понадобиться изменить такое поведение и предотвратить возможность изменения данных которые считаны сессией.

Типичное поведение приложений это выборка данных с помощью команды SELECT, отображение данных пользователю для просмотра и

возможность изменения этих данных. Так как Oracle поддерживает параллельную работу пользователей то ничто не мешает другому пользователю получить те же данные. Если обе сессии попробует сделать какие-либо изменения, то могут возникнуть странные ситуации. Следующий пример показывает такую ситуацию

First User	Second User
`select * from regions;`	`select * from regions;`
	`delete from regions where region_id=5;`
	`commit;`
`update regions set region_name='GB'where region_id=5;`	

Вот что увидит первый пользователь (прелположим что используется SQL *Plus)

```
SQL> select * from regions;
 REGION_ID REGION_NAME
---------- -------------------------
         5 UK
         1 Europe
         2 Americas
         3 Asia
         4 Middle East and Africa
SQL> update regions set region_name='GB' where region_id=5;
0 rows updated.
```

Такой результат немного смущает пользователя. Чтобы решить эту проблему можно заблокировать строки которые вернул запрос

select * from regions for update;

Директива FOR UPDATE приведёт к блокировке таблиц которые возвращает запрос. Другие сессии не смогут изменить данные и таким образом последующие изменения будут успешны: другие сессии не смогут изменить данные. То есть у одной сессии будет согласованное чтение данных, но ценой за это будет то, что другие сессии «зависнут» если они попытаются изменить данные которые заблокированы (другие сессии могут читать эти данные).

Блокировка строк вызванная командой FOR UPDATE будет длиться пока сессия не выполнит команду COMMIT или ROLLBACK. Команду

завершения транзакции необходимо выполнить даже если вы не запускали каких-либо DML команд.

Так называемый «авто-коммит»

Чтоб завершить обзор как обрабатывается управление транзакциями надо рассеять все сомнения о так называемом “auto-commit” или неявном подтверждении (implicit commit). Вы будете часто слышать что Oracle автоматически подтвердит. Первый случай это предыдущий случай когда вы выполнили команду DDL, другая ситуация когда пользователь вышел из программы такой как SQL *Plus.

На самом деле всё очень просто. Не существует такого понятия как авто-коммит. Когда вы выполняете DDL команду, то работает обычный COMMIT которые встроен в команду DDL. Но что происходит когда вы выходите из программы? Если вы используете SQL Plus в Windows и выполняете команду DML а затем команду EXIT (EXIT это команду SQL *Plus а не SQL), ваша транзакция будет подтверждена. Это потому что разработчики SQL *Plus встроили вызов команды COMMIT в команду EXIT. Если же вы нажмёте на красный крест в правом верхнем углу – то произойдёт вызов команды ROLLBACK. Так происходит потому что опять же разработчики SQL *Plus запрограммировали такое поведение программы. В другой операционной системе поведение программы SQL Plus может быть другим, единственный способ узнать это – это протестировать программу (или прочитать исходный код что в случае программы SQL Plus невозможно если вы не работаете в Oracle надо этой программой).

В SQL *Plus есть команда SET AUTOCOMMIT ON. Вызов этой команды указывает SQL *Plus на то как обрабатывать пользовательские запросы: SQL *Plus добавит вызов команды COMMIT после любой DML команды. Таким образом все запросы будут подтверждаться, как только они выполнены. Но опять же всё это происходит полностью на стороне пользовательского процесса; у базы данных нет никакого авто-коммита, и все долго-работающие изменения будут изолированы от других сессий пока

запрос не выполнится успешно. Даже в таком случае если вы запустите долгий запрос на выполнение, потом к примеру, завершите пользовательский процесс через диспетчер задач то PMON обнаружит сессию «призрак» и отменит транзакцию.

Задание: Создать транзакции из приведенных выше примеров в СУБД Oracle. Проанализировать ход их выполнения и полученный результат. Создать транзакции для спроектированной ранее БД на предоставленную тему.

Контрольные вопросы:

1. Что такое транзакция?
2. Чем отличаются транзакции в Oracle от других СУБД?

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №38

Тема: «Обработка функций и триггеров в Oracle»

Цель: Научиться создавать, вызывать и редактировать функции и триггеры в СУБД Oracle

Время выполнения: 4 урока

Ход работы:

Programming Language for SQL (PL/SQL) – решение Oracle, предоставляющее средства, которые позволяют выполнять сложную обработку информации (plsql).

Программные единицы PL/SQL

Процедура — это подпрограмма, которая выполняет специфическое действие (CREATE PROCEDURE).

Функция — это подпрограмма, которая вычисляет значение (CREATE FUNCTION).

PL/SQL пакеты — это объект базы данных, который группирует логически связанные типы, программные объекты и подпрограммы PL/SQL. Пакеты обычно состоят из двух частей — спецификации и тела. **Спецификация пакета** — это интерфейс с вашими приложениями, она объявляет типы, переменные, константы, исключения, курсоры и подпрограммы, доступные для использования в пакете. **Тело пакета** полностью определяет курсоры и подпрограммы, тем самым реализуя спецификацию пакета (CREATE PACKAGE и CREATE PACKAGE BODY).

Динамический SQL

Native Dynamic SQL (NDS)

DBMS_SQL

Триггеры — это хранимая процедура особого типа, которую пользователь не вызывает непосредственно, а исполнение которой обусловлено действием по модификации данных. В Oracle различают следующие виды триггеров: BEFORE INSERT, AFTER INSERT, BEFORE

UPDATE, AFTER UPDATE, BEFORE DELETE, AFTER DELETE (CREATE TRIGGER).

Опции компилятора

Управление зависимостями

Хинты или подсказки (Oracle Hints) — средство, позволяющее явным образом влиять на план запроса. Хинты определяют общие цели и подходы для оптимизации плана выполнения запроса, включая правила и методы доступа к данным — указание порядка соединения таблиц, указание метода соединения таблиц, указание конкретного индекса для доступа к таблице.

Структура блока PL/SQL

Блок PL/SQL может содержать до четырех разделов, однако только один из них является обязательным.

Заголовок. Используется только в именованных блоках, определяет способ вызова именованного блока или программы. Не обязателен.

Раздел объявлений. Содержит описания переменных, курсоров и вложенных блоков, на которые имеются ссылки в исполняемом разделе и разделе исключений. Не обязателен.

Исполняемый раздел. Команды, выполняемые ядром PL/SQL во время работы приложения. Обязателен.

Раздел исключений. Обрабатывает исключения (предупреждения и ошибки). Не обязателен.

Структура блока PL/SQL для процедуры показана на рисунке:

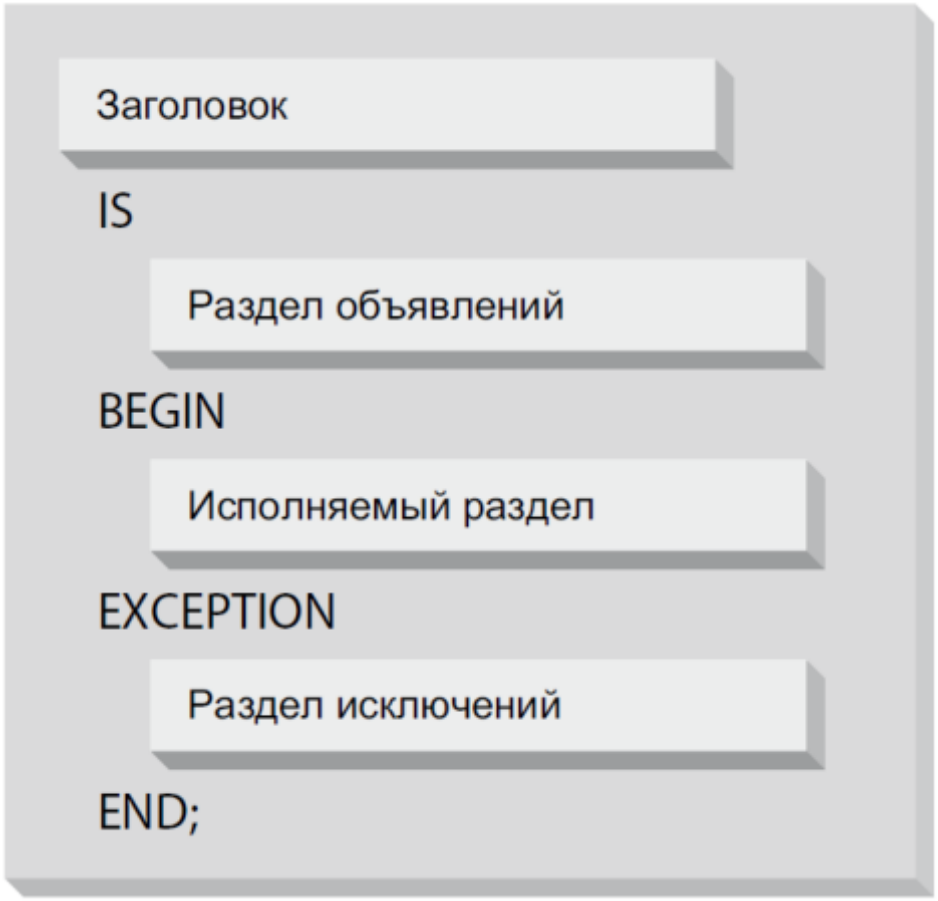


Рисунок «Процедура, содержащая все четыре раздела»:

```
PROCEDURE get_happy (ename_in IN VARCHAR2)
IS
  l_hiredate DATE;
BEGIN
  l_hiredate := SYSDATE - 2;
  INSERT INTO employee
    (emp_name, hiredate)
  VALUES (ename_in, l_hiredate);
EXCEPTION
  WHEN DUP_VAL_IN_INDEX
  THEN
    DBMS_OUTPUT.PUT_LINE
      ('Cannot insert.');
END;
```

Заголовок

Раздел об

Исполняем

Раздел иск

Хранимая процедура

Хранимая процедура — это определенный набор инструкций, написанных на языке PL/SQL. Вызов процедуры приводит к выполнению содержащихся в ней инструкций. Процедура хранится в базе данных, поэтому и называется хранимой. Процедура состоит из двух основных частей:*спецификации и тела*.

Спецификация процедуры (procedure specification) включает в себя имя процедуры и описание ее входных и выходных данных. Эти входные и выходные данные называются формальными параметрами (formal parameters) или формальными аргументами (formal arguments). Если при вызове процедуры указываются параметры командной строки или другие входные данные, эти значения называются фактическими (actual) параметрами или фактическими аргументами.

Пример спецификации

increase_salary_find_tax (increase_percent IN NUMBER := 7, sal IN OUT NUMBER, tax OUT NUMBER)

Здесь мы видим процедуру с тремя формальными параметрами. Слово IN после имени параметра означает, что при вызове процедура может считать из этого параметра входное значение. Слово OUT означает, что процедура может использовать данный параметр для возврата значения в ту программу, из которой она была вызвана. Комбинация IN OUT после имени параметра говорит о том, что параметр может использоваться как для передачи значения процедуре, так и для возврата значения.

Тело процедуры (procedure body) — это блок PL/SQL-кода.

Хранимые функции

Функция PL/SQL похожа на процедуру PL/SQL: она также имеет спецификацию и тело. Главное различие между процедурой и функцией в

том, что функция предназначена для возврата значения, которое может использоваться в более крупном SQL-Операторе.

Триггеры

Триггер — это процедура PL/SQL, которая выполняется автоматически, когда происходит некоторое заданное событие, называемое триггерным событием (triggering event). Например, можно писать триггеры, срабатывающие при выполнении над таблицей операций INSERT, UPDATE или DELETE; при выдаче команд DDL; при входе пользователя в систему или его выходе из системы; при запуске или останове базы данных; при возникновении ошибок. **Между триггерами и процедурами PL/SQL есть три различия:**

Триггеры нельзя вызывать из кода программы. Oracle вызывает их автоматически в ответ на определенное событие.

Триггеры не имеют списка параметров.

Спецификация триггера немного отличается от спецификации процедуры.

Структура блока PL/SQL

Базовый блок PL/SQL состоит из четырех секций:

секции заголовка (header section);

необязательной секции объявлений (declaration section);

выполняемой секции (execution section);

необязательной секции исключений (exception section).

DECLARE — необязательно — переменные, константы, курсоры **BEGIN** — обязательно — команды SQL, управляющие команды PL/SQL **EXCEPTION** — необязательно — действия в случае возникновения ошибок **END;** — обязательно

Анонимный блок (anonumous block) — это блок PL/SQL без секции заголовка, иначе говоря, секции имени, поэтому он и называется анонимным. Анонимные блоки могут выполняться из SQL*Plus и использоваться в функциях, процедурах и триггерах PL/SQL. Вспомните, что сами процедуры,

функции и триггеры также состоят из базовых блоков. Это означает, что базовый блок можно помещать в другой базовый блок.

Секция объявлений

Секция объявлений не является обязательной. В случае использования она начинается после секции заголовка и оканчивается перед ключевым, словом BEGIN. Эта секция содержит объявления переменных, констант, курсоров, исключений, функций и процедур PL/SQL, которые будут использоваться в выполняемой секции и секции исключений. Все объявления переменных и констант должны размещаться до объявлений функций или процедур. О переменных и константах PL/SQL будет подробно рассказано в следующем разделе. Объявление сообщает PL/SQL о том, что нужно создать переменную, константу, курсор, функцию или процедуру согласно приведенной спецификации. Когда выполнение базового блока завершается, все элементы, объявленные в секции объявлений, перестают существовать. Элементы, объявленные в секции объявлений базового блока, могут использоваться только в пределах этого блока. Одним словом, все, что находится в секции объявлений, при надлежит блоку и может использоваться только внутри него, а следовательно, существует только на протяжении его времени жизни. Часть кода, в которой может использоваться переменная, называется областью видимости (scope).

Выполняемая секция

Выполняемая секция начинается с ключевого слова BEGIN и заканчивается либо ключевым словом EXCEPTION, если присутствует секция исключений, либо ключевым словом END, за которым следуют необязательное имя функции или процедуры и точка с запятой. Выполняемая секция содержит один и более PL/SQL-операторов, выполняемых при передаче управления данному блоку. Структура выполняемой секции показана ниже.

BEGIN

один и более PL/SQL-операторов

[секция исключений]

END [имя функции или процедуры];

В выполняемом коде PL/SQL чаще всего встречается оператор присваивания (:=). Он указывает, что нужно вычислить выражение справа и поместить результат в переменную слева.

Секция исключений

В ходе выполнения PL/SQL-оператора может возникнуть ошибка, которая сделает невозможным дальнейшее выполнение программы. Такие исключительные ситуации называются исключениями (exceptions). Пользователь, вызвавший процедуру, должен быть проинформирован о возникновении исключения, а также о причинах, его вызвавших. Вы можете выдать пользователю содержательное сообщение об ошибке, или предпринять некоторые корректирующие действия и повторить операцию, выполнявшуюся до возникновения ошибки. Вы также можете откатить изменения, которые были произведены в базе данных к этому моменту. PL/SQL помогает вам во всех этих случаях, предоставляя средства обработки исключений (exception handling).

EXCEPTION

WHEN имя_исключения THEN

действия, предпринимаемые при возникновении исключения

WHEN имя_исключения THEN

действия, предпринимаемые при возникновении исключения

Секция исключений начинается с ключевого слова EXCEPTION и продолжается до конца блока. Каждому исключению соответствует оператор WHEN имя_исключения, указывающий, что должно быть сделано при возникновении данного исключения. Все операторы, находящиеся между оператором, вызвавшим ошибку, и секцией исключений, игнорируются. Выполнение оператора, указанного в секции исключений, называется обработкой исключения (exception handling). Процесс, включающий в себя обнаружение ошибки, определение, какое исключение описывает ее наилучшим образом, и передачу PL/SQL информации, позволяющей найти соответствующий код в секции исключений, называется возбуждением исключения (raising exception).

Переменные

Переменные — это именованные контейнеры. Они могут содержать информацию (данные) различных видов. В зависимости от того, какую информацию в них можно помещать, они имеют различные типы данных, а чтобы отличать их друг от друга, им присваиваются имена. PL/SQL хранит числа в переменных типа NUMBER, а текст — в переменных типа CHAR или VARCHAR2. Синтаксис объявления переменной в PL/SQL может иметь любую из следующих форм:

имя_переменной тип_данных [[NOTNULL] := выражение_по_умолчанию];

имя_переменной тип_данных [[NOT NULL] DEFAULT выражение_по_умолчанию];

Имя_переменной — это любой правильный идентификатор PL/SQL. Правильный идентификатор PL/SQL должен:

Иметь не более 30 символов в длину и не содержать пробельных символов (собственно пробелов и знаков табуляции).

Состоять только из букв, цифр от 0 до 9, символа подчеркивания (_), знака доллара ($) и знака фунта (#).

Начинаться с буквы.

Не совпадать с зарезервированными словами PL/SQL или SQL, которые имеют специальное значение. Например, именем переменной не может быть слово BEGIN, которое обозначает начало выполняемой секции базового блока PL/SQL.

Тип_данных — это любой допустимый тип данных SQL или PL/SQL. Модификатор NOT NULL требует, чтобы переменная имелазначение. Если он указан, переменной должно быть присвоено значение по умолчанию.

Объявление констант PL/SQL

Синтаксис объявления константы имеет следующий вид:

имя_переменной тип_данных CONSTANT := выражение;

В отличие от переменных **константам** обязательно присваивается значение, которое нельзя изменять на протяжении времени жизни константы. Константы очень полезны для поддержания безопасности и дисциплины при разработке больших и сложных приложений. Например, если вы хотите гарантировать, что процедура PL/SQL не будет модифицировать передаваемые ей данные, можете объявить их константами. Если процедура все же попытается их модифицировать, PL/SQL возбудит исключение.

Оператор IF

Оператор IF имеет следующий синтаксис:

IF условие_1 THEN

действие_1;

[ELSIF условие_2 THEN действие_2;]

…

[ELSE альтернативное_действие;]

END IF;

Действие_1 … альтернативное Действие представляют один или несколько PL/SQL-операторов. Каждая группа операторов выполняется только в том случае, если выполнено соответствующее условие. После того как обнаружено выполнение одного из условий, остальные условия не проверяются.

Циклы

PL/SQL предоставляет три различные конструкции для итеративной обработки. Каждая из них позволяет циклически выполнять набор операторов PL/SQL. Выход из цикла осуществляется в зависимости от некоторого условия.

LOOP

Конструкция LOOP имеет следующий синтаксис:

имя_цикла LOOP

операторы;

EXIT имя_цикла [WHEN условие_выхода];

операторы;

END LOOP;

При наличии конструкции WHEN все операторы в теле цикла повторяются до тех пор, пока выражение условие_выхода не примет положительное значение (т.е. не станет истинным). Условие выхода проверяется на каждом проходе, иначе называемом итерацией. Как только выражение принимает значение «истина», все операторы после EXIT пропускаются, итерации прекращаются и выполнение продолжается с

первого оператора, следующего за END LOOP. Если условие WHEN отсутствует, операторы между LOOP и EXIT выполняются только один раз. Очевидно, что опустив условие WHEN, вы поступите нелогично. В конце концов идея цикла состоит в том, чтобы обеспечить потенциально многократное выполнение кода.

Цикл WHILE

Еще одной разновидностью цикла является цикл WHILE. Он хорошо подходит в ситуациях, когда количество итераций заранее неизвестно, и определяется некоторым внешним фактором. Цикл WHILE имеет следующий синтаксис:

```
WHILE условие_выхода LOOP
операторы;
END LOOP;
```

Условие WHILE проверяется перед каждым входом в цикл. Если оно имеет значение «истина», то выполняется очередная итерация.

Цикл FOR

В цикле FOR для подсчета итераций используется переменная-счетчик, называемая также индексом цикла (loop index). По завершении каждой итерации счетчик увеличивается, начиная с нижнего предела, или уменьшается, начиная с верхнего предела. Как только его значение выйдет за указанный диапазон, цикл завершается. **Синтаксис цикла FOR выглядит следующим бразом:**

```
FOR счетчик IN [REVERSE] нижняя_граница .. верхняя_граница
LOOP
```

операторы;

END LOOP;

Курсоры

Курсор — это исключительно важная конструкция PL/SQL, лежащая в основе взаимодействия PL/SQL и SQL. Название «курсор» означает «текущий набор записей». Курсор представляет собой специальный элемент PL/SQL, с которым связан SQL-оператор SELECT. Используя курсор, можно отдельно обрабатывать каждую строку связанного с ним SQL-оператора. Курсор объявляется в секции объявлений базового блока. Он открывается командой OPEN, а выборка строк осуществляется с помощью команды FETCH. После завершения всей обработки курсор закрывается командой CLOSE. Закрытие курсора освобождает те системные ресурсы, которые использовались, пока он был открыт. Строки, выбранные курсором, можно заблокировать, чтобы предотвратить их модификацию другими пользователями. Закрытие курсора или выполнение явной операции COMMIT или ROLLBACK приведет к разблокированию строк. Для SQL-операторов, используемых в коде PL/SQL, применяются скрытые, или неявные (implicit), курсоры, а также явные (explicit) курсоры, т.е. те, которым присвоено имя.

Объявление курсора и атрибуты курсора

Курсор объявляется в процедуре PL/SQL следующим образом:

CURSOR имя_курсора [([параметр_1 [, параметр_2...])]

[RETURN спецификация возврата] IS

оператор_select [FOR UPDATE [OF таблица_или_столбец_1 [, таблица_или_столбец_2...]]]

Параметры курсора похожи на параметры процедуры, за тем исключением, что они всегда являются входными (IN). Использование

параметров OUT или IN OUT невозможно, поскольку курсор не может их модифицировать. Параметры используются в конструкции WHERE курсорного оператора SELECT. Спецификация возврата показывает, записи какого типа будут выбираться оператором SELECT. Таблица_или_столбец — это имя столбца, который предстоит обновлять, или имя таблицы, в которой предстоит удалять или обновлять строки. Оно должно входить в число имен таблиц и столбцов, указанных в операторе SELECT курсора, и предназначено для документирования, показывая, какие элементы могут быть потенциально модифицированы кодом, использующим данный курсор. Команда FOR UPDATE блокирует строки, выбранные оператором SELECT при открытии курсора. Строки остаются заблокированными до тех пор, пока вы не закроете курсор рассмотренными выше способами. **Атрибуты курсора:**

имя_курсора%ISOPEN — Позволяет проверить, открыт ли курсор. Если курсор имя_курсора уже открыт, возвращается значение TRUE

имя_курсора%ROWCOUNT — Количество строк таблицы, возвращенных оператором SELECT курсора

имя_курсора%FOUND — Позволяет проверить, была ли успешной последняя попытка получения записи из курсора. Если запись была выбрана, то возвращается значение TRUE

имя_курсора%NOTFOUND — Противоположен атрибуту FOUND. Если записей больше не найдено, возвращается значение TRUE

Записи PL/SQL

Запись PL/SQL— это набор данных базовых типов. К ней можно обращаться, как к единому целому. Для доступа к отдельным полям записи применяется нотация имя_записи_имя_поля, которую вы уже использовали для столбцов таблицы. Записи могут иметь один из трех типов, перечисленных ниже; вы можете объявлять переменные, имеющие тип записи.

Основанные на таблице (table-based) Эти записи имеют поля, совпадающие по имени и типу со столбцами таблицы. Если курсор выбирает

всю строку — например, оператором SELECT * FROM некоторая_таблица — то возвращаемые им записи можно непосредственно копировать в переменную, имеющую тип записи, основанной на таблице некоторая_таблица.

Основанные на курсоре (cursor-based) Поля этих записей совпадают по имени, типу и порядку с заключительным списком столбцов в курсорном операторе SELECT.

Определенные программистом (programmer-defined) Это записи, тип которых определяете вы сами.

Использование команд OPEN, FETCH и CLOSE

Команды открытия курсора, выборки из курсора и закрытия курсора имеют следующий синтаксис:

OPEN имя курсора;

FETCH имя_курсора INTO переменная_или_список_переменных;

CLOSE имя_курсора;

После открытия курсор содержит набор записей, если в результате успешного выполнения оператора SELECT из базы данных были выбраны заданные строки. Каждая команда FETCH удаляет запись из открытого курсора и перемещает ее содержимое либо в переменную PL/SQL, тип записи которой совпадает с типом записи курсора, либо в группу переменных PL/SQL, где каждая переменная в списке совпадает по типу с соответствующим полем в записи курсора. Перед тем как пытаться выбрать из курсора очередную запись, следует проверить с помощью атрибутов FOUND и NOTFOUND, есть ли в нем еще записи. Выборки из пустого курсора будут все время давать последнюю запись, не приводя к ошибке. Не забывайте проверять атрибуты FOUND и NOTFOUND при использовании FETCH. Фактическая обработка записей из курсора обычно выполняется

внутри цикла. При написании такого цикла неплохо начать с проверки, была ли найдена запись в курсоре. Если да, можно продолжать необходимую обработку; в противном случае следует выйти из цикла. То же самое можно сделать более коротким путем, использовав курсорный цикл FOR. При этом PL/SQL будет осуществлять открытие, выборку и закрытие без вашего участия.

Курсорный цикл FOR

Синтаксис курсорного цикла FOR имеет следующий вид:

```
FOR запись_курсора IN имя_курсора LOOP
операторы;
END LOOP;
```

Этот цикл выбирает записи из курсора в переменную типа запись_курсора. Поля записи_курсора можно использовать для доступа к данным из операторов PL/SQL, выполняемых в цикле. Когда все записи выбраны, цикл завершается. Для удобства открытие и закрытие курсора производится автоматически. Попытавшись выбрать запись из неоткрытого курсора, вы получите сообщение in valid cursor (недействительный курсор). Если не закрывать курсоры, то в конце концов количество открытых курсоров достигнет максимальной величины, допускаемой системой.

Конструкция WHERE CURRENT OF

Когда курсор открывается для обновления или удаления выбранных записей, можно использовать конструкцию ***WHERE CURRENT OF имя_курсора*** для доступа к таблице и строке, которые соответствуют последней записи, выбранной в конструкции WHERE оператора UPDATE или DELETE.

Обработка ошибок

В языке PL/SQL ошибки всех видов интерпретируются как исключения — ситуации, которые не должны возникать при нормальном выполнении программы.

К числу исключений относятся:

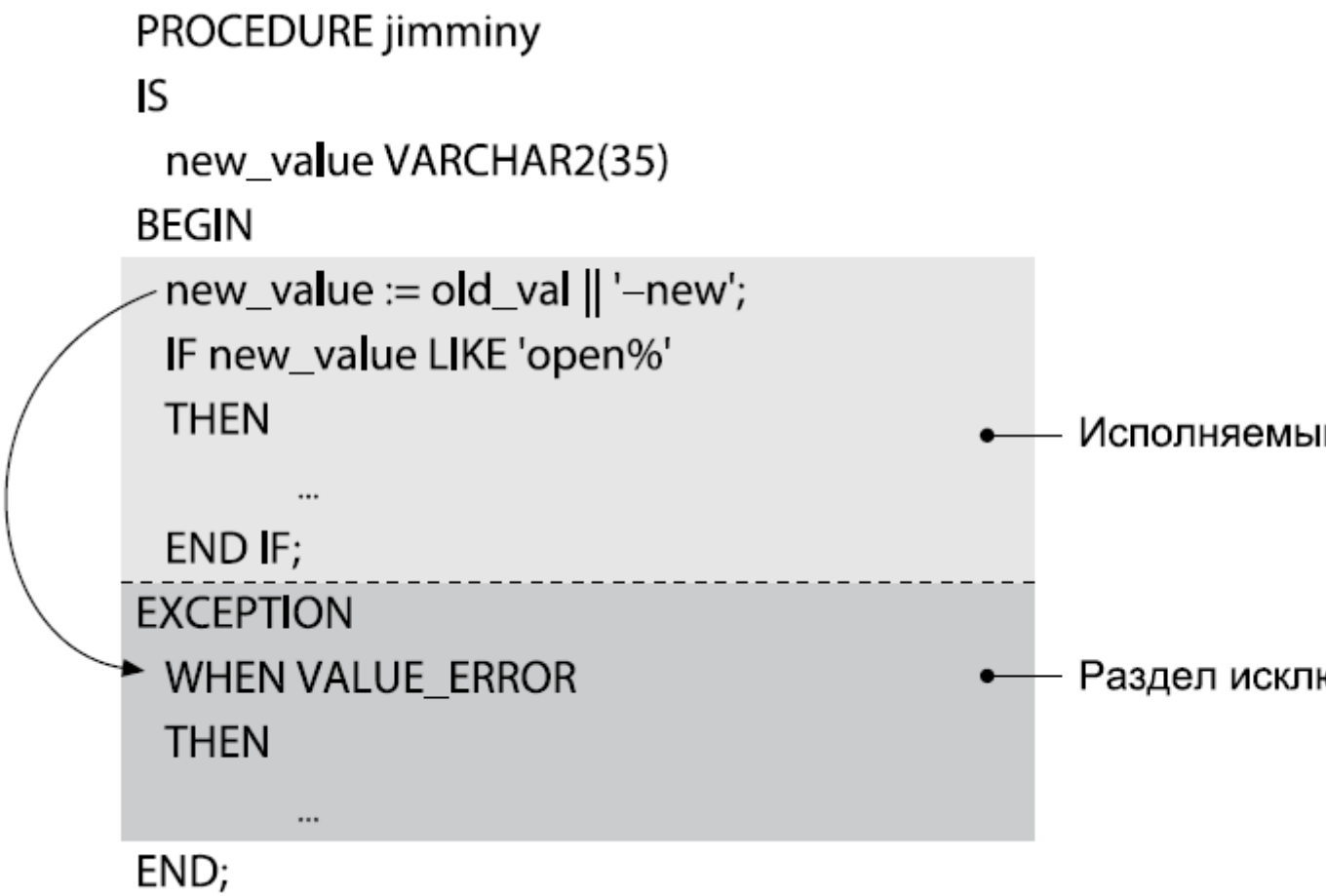
ошибки, генерируемые системой (например, нехватка памяти или повторяющееся значение индекса);

ошибки, вызванные действиями пользователя;

предупреждения, выдаваемые приложением пользователю.

PL/SQL перехватывает ошибки и реагирует на них при помощи так называемых обработчиков исключений. Механизм обработчиков исключений позволяет четко отделить код обработки ошибок от основной логики программы, а также дает возможность реализовать обработку ошибок, управляемую событиями. Независимо от того, как и по какой причине возникло конкретное исключение, оно всегда обрабатывается одним и тем же обработчиком в разделе исключений. **При возникновении ошибки — как системной, так и ошибки в приложении — в PL/SQL инициируется исключение.** В результате выполнение блока прерывается, и управление передается для обработки в раздел исключений текущего блока, если он имеется. После обработки исключения возврат в тот блок, где исключение было инициировано, невозможен, поэтому управление передается во внешний блок.

Схема передачи управления при возникновении исключения:



Существует два типа исключений:

Системное исключение определяется в Oracle и обычно инициируется исполняемым ядром PL/SQL, обнаружившим ошибку. Одним системным исключениям присваиваются имена (например, NO_DATA_FOUND), другие ограничиваются номерами и описаниями.

Исключение, определяемое программистом, актуально только для конкретного приложения. Имя исключения можно связать с конкретной ошибкой Oracle с помощью директивы компилятора EXCEPTION_INIT или же назначить ошибке номер и описание процедурой RAISE_APPLICATION_ERROR.

Исключения

Исключение — это состояние ошибки, которое активизируется — или возбуждается — при возникновении некоторой проблемы. Существует много разных исключений, каждое из которых связано с определенным типом

проблем. При возникновении исключительной ситуации выполнение кода останавливается на операторе, который возбудил исключение, и управление передается той части блока, которая обрабатывает это исключение. Если блок не содержит выполняемой секции, PL/SQL пытается найти выполняемую секцию во включающем базовом блоке (enclosing basic block), т.е. в блоке, который является внешним по отношению к коду, возбудившему исключение. Если в непосредственном включающем блоке отсутствует обработчикданного исключения, то поиск продолжается в блоках следующих уровней, пока не будет найден подходящий обработчик, а если его найти не удается, то выполнение программы прекращается с выдачей сообщения о необрабатываемой ошибке. Часть блока, предназначенная для обработки исключений, — это идеальное место для выдачи информативных сообщений об ошибках и выполнения очистки (cleanup), позволяющей избавиться от всего, что могло бы в дальнейшем вызвать путаницу или проблемы. Если исключение было возбуждено в ходе выполнения процедуры, вставляющей строки в таблицу, то типичная процедура очистки может включать в себя оператор ROLLBACK. После того как управление было передано обработчику исключения, оно уже не возвращается оператору, ставшему причиной этого исключения. Вместо этого управление передается оператору включающего базового блока, который следует сразу за вызовом вложенного блока или процедуры/функции.

Системные исключения

В PL/SQL можно выдавать пользователям информацию об ошибке двумя способами. Первый способ — использовать команду SQLCODE, которая возвращает код ошибки. Этот код представляет собой отрицательное число, обычно равное номеру ошибки ORA, которая выводится при завершении приложения, если исключение осталось необработанным. Второй способ — возвращать текстовое сообщение, описывающее ошибку. Неудивительно, что соответствующая команда называется SQLERRM. В обработчике исключения можно использовать как SQLCODE, так и

SQLERRM. Замечание: не у всех системных исключений есть имена. **Системные исключения:**

CURSOR_ALREADY_OPEN — Попытка открыть уже открытый курсор;

DUP_VAL_ON_INDEX — Попытка вставить повторяющееся значение в столбец, имеющий уникальный индекс, а следовательно, ограничение уникальности;

INVALID_CURSOR — Попытка применить команду FETCH к неоткрытому курсору или попытка закрыть курсор, который не открывался;

NO_DATA_FOUND — Попытка выполнить SELECT INTO, когда SELECT возвращает нулевое количество строк, а также другие причины;

PROGRAM_ERROR — Внутренняя ошибка. Обычно означает, что вам нужно обратиться в службу поддержки Oracle;

STORAGE_ERROR — Программе не хватает системной памяти;

TIME_OUT_ON_RESOURCE — Программа слишком долго ожидала доступности некоторого ресурса;

TOO_MANY_ROWS — SELECT INTO в PL/SQL вернул более одной строки;

VALUE_ERROR — PL/SQL встретил неправильное преобразование или усечение данных, или неправильное ограничение на данные;

ZERO_DMDE — Попытка деления на нуль;

OTHERS — Все прочие исключения и внутренние ошибки, которые не охватываются исключениями, определенными в базовом блоке. Используется в тех случаях, когда вы точно не знаете, какое именованное исключение предстоит обрабатывать, и хотите обрабатывать любые возбуждаемые исключения.

Исключения, определяемые программистом

Одной из удобных возможностей PL/SQL является то, что он позволяет вам определять свои собственные исключения. При возбуждении и обработке они должны именоваться и объявляться аналогично любым другим

элементам PL/SQL. Исключение объявляется в секции объявлений. Аналогично любой другой объявленной там переменной, исключение действительно только для данного блока. Вы можете использовать свои собственные исключения для обработки ошибок, которые система не обнаруживает или не считает за ошибки.

Схема сложного запроса PL/SQL с использованием временных таблиц

Ниже рассмотрена схема получения результирующей таблицы в Oracle Database, используя временные таблицы.

```
PROCEDURE NameYourProcedure IS

--Объявления переменных
VarName type;
--Объявления курсоров, пример:
  CURSOR NameCursor(...) IS
    SELECT ... FROM ...;

--Блок BEGIN ... END
BEGIN
0
        --Здесь можно использовать курсор
1       --Дальше я опишу вариант, который используется вместо
  VIEW, его можно запустить в открытом курсоре или каком-то цикле (FOR
2 LINE ... LOOP)

3       --"Вместо View использовать временные таблицы, а результат
  инсертить в промежуточную таблицу"
```

```
INSERT INTO NameYourTableResult (
    Field1,
    Field2,
    Field3
    ...
    FieldN
)
WITH --в этот блок помещаются временные таблицы, которые и заменяют View
TempTable1 as (
    --Сложный SQL запрос, результат которого будет храниться в TempTable1 на протяжении всего INSERT INTO
)
TempTable2 as (
    --Сложный SQL запрос, в котором мы можем использовать временную таблицу TempTable1 и другие таблицы базы данных
)
TempTable3 as (
    --Сложный SQL запрос, в котором мы можем использовать временные таблицы TempTable1, TempTable2 и другие таблицы базы данных
)
--... и так далее, но главное не увлекаться :)

--Основной блок SELECT, из которого результаты пойдут в команду INSERT INTO (количество полей в SELECT равно количеству полей в INSERT)
--В этом запросе SELECT используются временные таблицы
```

```
    TempTable1, TempTable2, TempTable3 и т.д. и другие источники данных
9               SELECT --Перечень полей
                FROM TempTable3, sourceTable1, sourceTable2, sourceTable3
0               JOIN ...
                WHERE ...;
1       END;

2       END NameYourProcedure;

3

4

5

6

7

8

9

0

1

2
```

Теперь можно использовать обычный SQL для получения предрассчитанных данных в удобное время (например ночью или утром, для оптимизации использования ресурсов сервера и информационных систем).

TOAD

Инструментальное средство TOAD (Tool for Oracle Application Development) является одним из наиболее популярных средств разработки серверных приложений Oracle. TOAD в процессе работы позволяет устанавливать несколько соединений с БД. Доступ к набору инструментальных средств, используемых при разработке и отладке серверных приложений и схем баз данных осуществляется, преимущественно, с помощью пункта главного меню Database.

Database -> Schema Browser – открывает окно браузера, с помощью которого осуществляется доступ к информации об объектах схем, видимых в текущем соединении.

Database -> SQL Editor – используется для написания и выполнения команд и скриптов SQL;

Database -> Procedure Editor – предоставляет пользователю инструментальные средства для разработки блоков PL/SQL;

Database -> SQL Modeller – поддерживает процесс визуального составления SQL-запросов;

Database -> Export -> … – предоставляет пользователю возможность выгрузки данных из БД в виде скриптов SQL;

Database -> Import -> … – импортирует данные в БД;

Database -> Commit – осуществляет фиксацию транзакции в текущем соединении;

Database -> Rollback – осуществляет откат транзакции в текущем соединении.

Термины в исключениях:

Раздел исключений — необязательный раздел блока PL/SQL (анонимного блока, процедуры, функции, триггера или инициализационного раздела пакета), содержащий один или несколько обработчиков исключений. Структура раздела исключений очень похожа на структуру команды CASE.

Инициировать исключение — значит остановить выполнение текущего блока PL/SQL, оповещая исполняемое ядро об ошибке. Исключение может инициировать либо Oracle, либо ваш собственный программный код при помощи команды RAISE или процедуры RAISE_APPLICATION_ERROR.

Обработать исключение — значит перехватить ошибку, передав управление обработчику исключения. Написанный программистом обработчик может содержать код, который в ответ на исключение выполняет определенные действия (например, записывает информацию об ошибке в журнал, выводит сообщение для пользователя или передает исключение во внешний блок).

Область действия — часть кода (конкретный блок или весь раздел), в котором может инициироваться исключение, а также часть кода, инициируемые исключения которого могут перехватываться и обрабатываться соответствующим разделом исключений.

Передача исключения — процесс передачи исключения во внешний блок, если в текущем блоке это исключение не обработано.

Необработанное исключение — исключение, которое передается без обработки из «самого внешнего» блока PL/SQL. После этого управление передается исполнительной среде, которая уже сама определяет, как отреагировать на исключение (выполнить откат транзакции, вывести сообщение об ошибке, проигнорировать ее и т. д.).

Анонимное исключение — исключение, с которым связан код ошибки и описание. Такое исключение не имеет имени, которое можно было бы использовать в команде RAISE или секции WHEN обработчика исключений.

Именованное исключение — исключение, которому имя присвоено либо Oracle (в одном из встроенных пакетов), либо разработчиком. В частности, для этой цели можно использовать директиву компилятора EXCEPTION_INIT (в таком случае имя можно будет применять и для инициирования, и для обработки исключения).

Задание: Создать функции и триггеры из приведенных выше примеров в СУБД Oracle. Проанализировать ход их выполнения и полученный результат. Создать транзакции для спроектированной ранее БД на предоставленную тему.

Контрольные вопросы:

1. Какие бывают триггеры в СУБД Oracle?
2. Как вызвать функцию и передать ей параметры в СУБД Oracle?

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №39

Тема: «Проектирование расширенной реляционной модели»

Цель: Научиться проектировать БД на основе расширенной реляционной модели.

Время выполнения: 4 урока

Ход работы:

Необходимость разработки собственной платформы для проектирования и реализации бизнес-приложений зрела давно и вот в 1995 году компания ЭРСИС сделала первый шаг в этом направлении. Началось все с создания внутрифирменной библиотеки для снижения трудоемкости в разработке программных продуктов. Прошло немного времени, и благодаря появляющемуся опыту и наработкам начал формироваться абсолютно новый и, как показала, практика эффективный подход в разработке все более усложняющихся приложений к базам данных. Так были выработаны дополнительные вспомогательные компоненты, объединенные затем в инструментарий. Процесс становления шел динамически, то есть новый инструментарий появлялся, а старый либо дорабатывался, либо переписывался с нуля.

Разработчики добавили следующие функциональные блоки необходимые для разработки бизнес-приложений:

erSys - машина баз данных управляемая расширенной реляционной моделью;

erAdmin - утилита для управления структурой RDBMS и конфигурациями бизнес-приложений;

erCompute - специальные классы и функции для эффективного выполнения сложных расчетных задач;

erForms - библиотека автоформ для быстрого макетирования бизнес-приложения при разработке;

erAPI - библиотека разработчика;

Очень важным для разработчиков любых приложений является свойство наследования и преемственности, платформа ЭРСИС позволяет не изобретать велосипед, а использовать уже давно наработанные и известные разработки сторонних компаний. Например, генератор отчетов Fast-Report разработанный компанией Fast-Report Inc или компоненты доступа к Системе Управления Базой Данных ORACLE - Direct Oracle Access разработки компании Allround Automations великолепно вольются в код вашего приложения с помощью директив компиляции.

Текущая версия разработки бизнес-приложений RAD-платформы ЭРСИС представляет набор библиотек и инструментов, необходимых для расширения среды программирования Delphi. Все задачи, которые обобщает платформа ЭРСИС можно свести к следующим:

очень быстрое проектирование и создание реляционных моделей данных. Впоследствии легкое проектирование прикладных программных модулей на их основе. Платформа ЭРСИС использует расширенную реляционную модель системы управления данными, под данными понимаются SQL-потоки данных, связи между потоками. SQL-поток это логическое представление сущности Базы Данных и ее атрибутов. ЭРСИС использует расширенное понятие связей, а не только стандартных Foreign-Primary Key.

Достаточно описать поля и связи между потоками и все необходимые SQL-запросы для выборки и корректировки данных будут сгенерированы на уровне платформы ЭРСИС автоматически. Изюминкой является то, что все SQL-запросы создаются динамически из модели на этапе выполнения приложения, независимо от базы данных и разработчика. Будь то ORACLE или MySQL.

При проектировании приложения выполняется привязка логической базы к СУБД, исходя из механизма доступа к ней. Таким образом, платформа ЭРСИС сама автоматически определяет тип синтаксиса SQL, классы источников данных, необходимых при создании SQL-потоков данных.

Соответственно автоматически создаются необходимые компоненты доступа к данным, а в последствии наборы компонентов. Например, если используется BDE, то платформа генерирует TQuery, TUpdateSQL, в последствии при работе с потоком создается минимально необходимое количество TQuery для работы с требуемыми полями данных. Аналогично при реализации той же задачи используя DBExpress, получаем компоненты TSimpleDataSet и TSQLQuery. Стоит отметить что программист в своей работе не касается самих этих компонентов, он работает уровнем выше с дескриптором источника данных. Который и инициализирует сам механизм доступа к данным.

Разработчику не надо ломать голову как назвать компоненты формы, как привязать их к форме, как осуществить связи полученных элементов формы между собой. Все компоненты автоформы автоматически строятся, именуются и используются исходя из описания потока. То есть, предусмотрена автоматическое проектирование и генерация интерфейсных форм для работы с данными потоков.

Среда разработки Borland Delphi являлась основой разработки многих платформ. Но если глядеть в корень, то становится очевидным, что большинство из них копии 1С. То есть существует некоторая система документооборота, к этой системе прилагается конфигуратор и какой-нибудь скриптовый язык. Изюминкой же разработки фирмы ЭРСИС является то, что основным элементом платформы является среда визуального программирования Delphi. Ко всем многочисленным достоинствам Delphi были добавлены те самые свойства, которые делают разработку ЭРСИС такой удобной, быстрой и эффективной в работе, а именно, использование расширенной реляционной модели системы управления базой данных, независимость от типа Базы Данных и автоматическое формирование элементов интерфейса.

Задание: Создать БД в ЭРСИС согласно выбранной теме с помощью инструментов автоматизированного проектирования.

Контрольные вопросы:

1. Что такое расширенная реляционная модель?
2. Где применяются БД с расширенной реляционной моделью?

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №40

Тема: «Проектирование объектно-ориентированной СУБД»

Цель: Научиться проектировать объектно-ориентированные СУБД.

Время выполнения: 4 урока

Ход работы:

Объектно-ориентированные базы данных – базы данных, в которых информация представлена в виде объектов, как в объектно-ориентированных языках программирования. Объектно-ориентированные базы данных обычно рекомендованы для тех случаев, когда требуется высокопроизводительная обработка данных, имеющих сложную структуру[1]. Основные трудности объектно-ориентированного моделирования данных проистекают из того, что такого развитого математичекого аппарата, на который могла бы опираться общая объектно-ориентированная модель данных, не существует. В большой степени поэтому до сих пор нет базовой объектно-ориентированной модели.

Объектно-ориентированный подход базируется на концепциях:

- объекта и идентификатора объекта;
- атрибутов и методов;
- классов;
- иерархии и наследования классов.

Характеристики ООБД

Выбор обязательных характеристик ООБД основан на двух критериях: система ООБД должна быть объектно-ориентированна и должна представлять собой базу данных.

К обязательным характеристикам относят:

Возможность работы со сложными объектами, путем наличия конструктора сложных объектов. Конструктор сложных объектов позволяет применить базовые модели построения объекта на любой вид объекта.

Возможность идентифицировать объект, путем присвоения ему индивидуального идентификатора, не зависящего от значений атрибутов этого объекта

Возможности инкапсуляции, а именно использование права доступа только к спецификации интерфейса методов, а информацию о данных и реализации методов скрыть внутри объектов.

Поддержка типов и классов. Требуется, чтобы в ООБД поддерживалась хотя бы одна концепция различия между типами и классами.

Возможность реализовывать механизм наследования от суперкласса, а именно перенос атрибутов.

Возможность перегрузки в сочетании с полным связыванием. Методы должны применяться к объектам разных типов. Реализация метода должна зависеть от типа объектов, к которым данный метод применяется.

Вычислительная полнота. Язык манипулирования данными должен быть языком программирования общего назначения.

Набор типов данных должен быть расширяемым. Пользователь должен иметь средства создания новых типов данных на основе набора предопределенных системных типов. Более того, между способами использования системных и пользовательских типов данных не должно быть никаких различий.

Общие понятия объектно-ориентированного подхода и их преломление в ООБД

Причиной появления систем объектно-ориентированных баз данных была потребность в более адекватном представлении и моделировании сущностей реального мира, поскольку ООБД обеспечивают гораздо более развитую модель данных, нежели традиционные — реляционные базы данных. Парадигма ООБД основывается на ряде базовых понятий, таких как объект, идентифицируемость, класс, наследование, перегрузка и отложенное связывание. Любой объект при своем создании получает генерируемый системой уникальный идентификатор, который связан с объектом во все

время его существования и не меняется при изменении состояния объекта.Каждый объект имеет состояние и поведение.

Состояние объекта - набор значений его атрибутов.

Значение атрибута объекта - это тоже некоторый объект или множество объектов.

Поведение объекта - набор методов (программный код), оперирующих над состоянием объекта.

Значениями свойства могут являться примитивные значения (такие, как строки или целые числа) и непримитивные объекты. Непримитивный объект, в свою очередь, состоит из набора свойств. Следовательно, объекты можно рекурсивно определять в терминах других объектов. Поведение объекта определяется с помощью методов, которые оперируют над состоянием объекта. Множество объектов с одним и тем же набором атрибутов и методов образует класс объектов. Объект должен принадлежать только одному классу, за исключением ситуации наследования. Допускается порождение нового класса на основе уже существующего класса - наследование. В этом случае новый класс, называемый подклассом существующего класса (суперкласса) наследует все атрибуты и методы суперкласса. В подклассе, кроме того, могут быть определены дополнительные атрибуты и методы.

Достоинства и недостатки ООБД

Достоинства

Объектно-ориентированные базы данных позволяют представлять сложные объекты более непосредственным образом, нежели реляционные системы.

Определение пользовательских абстракций.Объектно-ориентированные базы данных предоставляют возможность определять новые абстракции и управлять реализацией таких абстракций.Современные пакеты ООБД дают пользователю возможность создания нового класса с атрибутами и методами, иметь классы, наследующие атрибуты и методы от

суперклассов, создавать экземпляры класса, каждый из которых обладает уникальным объектным идентификатором, извлекать эти экземпляры по одному или группами, а также загружать и выполнять методы, определять объекты как совокупности других объектов,определять свойства которые тоже могут иметь сложную структуру и определяться с помощью конструктора коллекций

Облегченное проектирование некоторых связей.В объектно-ориентированных базах данных поддерживается средство инверсных связей для выражения взаимных ссылок между двумя объектами (бинарная связь). Такая система обеспечивает ссылочную целостность путем установления соответствующей обратной ссылки сразу же после создания прямой ссылки.

Отсутствие потребности в определяемых пользователями ключах.В модели ООБД имеется понятие идентификаторов объектов, автоматически генерируемых системой и гарантированно уникальных для каждого объекта. Это обстоятельство в сочетании с тем, что в модели ООБД устраняется потребность в определяемых пользователями ключах, дает объектно-ориентированным базам данных и другие преимущества. Во-первых, идентификатор объекта не может быть модифицирован приложением. Во-вторых, понятие идентифицируемости объекта влечет отдельное и согласованное понятие идентичности, не зависимое от того, каким образом производится доступ к объекту или как объект моделируется с помощью описательных данных.

Наличие предикатов сравнения.В РБД сравнение всегда базируется лишь на значениях. В этой модели два кортежа являются одной сущностью, если все их ключевые атрибуты имеют одинаковые значения. Однако в модели ООБД были разработаны и определены иные типы сравнения.

Меньшая потребность в соединениях.Возможность навигации по структурам объектов и проистекающие из этого путевые выражения в терминах атрибутов объектов дают нам возможность по-новому взглянуть на проблему соединений в ООБД. Реляционное соединение — это механизм,

сопоставляющий два отношения на основе значений соответствующих пар атрибутов в этих отношениях. Поскольку в ООБД два класса могут иметь соответствующие пары атрибутов, в этой модели все еще может сохраняться необходимость в реляционном соединении (или явном соединении).

Выигрыш в производительности. В большинстве ООБД при загрузке объекта в память хранимые в этом объекте идентификаторы объектов преобразуются в указатели по памяти.

Объектная алгебра.Объектная алгебра не столь подробно разработана и не является столь же зрелой, как реляционная алгебра. Но как бы то ни было, такая алгебра существует, и в ней определяются пять фундаментальных операций, сохраняющих объекты: union, difference, select, generate и map.

Недостатки

Отсутствие интероперабельности между РБД и ООБД:

минимальную оптимизацию запросов;

отсутствие стандартной алгебры запросов;

отсутствие средств обеспечения запросов;

отсутствие поддержки представлений;

проблемы с безопасностью;

отсутствие поддержки динамических изменений определений классов;

ограниченная поддержка ограничений целостности;

ограниченные возможности настройки производительности;

недостаточная поддержка сложных объектов;

ограниченная интеграция с существующими объектно-ориентированными системами программирования;

ограниченный выигрыш в производительности.

Недостаточность средств для оптимизации запросов .Одной из самых значительных проблем в ООБД является оптимизация декларативных запросов. Оптимизацию запросов к ООБД затрудняет дополнительная сложность самой объектно-ориентированой модели данных.

Отсутствие стандартной алгебры запросов.Это обстоятельство тоже затрудняет оптимизацию запросов.

Отсутствие средств обеспечения запросов.В большинстве ООБД не хватает средств обеспечения запросов. В тех же немногих системах, где имеются достаточные соответствующие средства, язык запросов не совместим с ANSI SQL. Среди этих средств обеспечения запросов нет вложенных подзапросов, запросов с множествами (union, intersection, difference), агрегатных функций и GROUP BY, соединения нескольких классов — возможностей, полностью поддерживаемых в РБД.

Отсутствие поддержки представлений.Разработка объектно-ориентированного механизма представлений осложняется такими свойствами модели, как идентифицируемость объектов.

Проблемы с безопасностью.В РБД поддерживается авторизация, тогда как в большинстве ООБД она отсутствует. РБД предоставляют пользователям передавать и изымать права на чтение или изменение определений и кортежей в отношениях и представлениях. ООБД смогут получить более широкое распространение в области бизнеса лишь в том случае, если эта функция в них будет усовершенствована.

Ограниченная поддержка ограничений целостности.Отсутствуют механизмы объявления ключевых свойств атрибутов (например, атрибут класса не может быть объявлен первичным ключом класса), или ограничений уникальности, явных ограничений целостности, а также пред- и постусловий методов.

Ограниченные возможности настройки производительности.В большинстве ООБД имеются лишь ограниченные средства параметризованной настройки производительности. В РБД инсталляторам предоставляется возможность настраивать производительность системы путем задания большого числа параметров, устанавливаемых системным администратором.

Недостаточная поддержка сложных объектов.Полная функциональность сложных объектов все еще не поддерживается. Можно осуществлять навигацию по ссылкам и кодировать операции с применением этих ссылок, но отсутствуют предопределенные родовые операции, в которых используются различные виды семантики ссылок.

Ограниченная интеграция с объектно-ориентированными системами программирования. Конфликты по именам; необходимость переделывать иерархии классов; склонность ООБД к перегрузке системных операций.

Реляционные БД и ООБД

Не смотря на огромную популярность парадигмы ООП в программировании, в технологии разработки баз данных эта парадигма пока не особо популярна. И тому есть как объективные, так и субъективные причины.

Популярность. Под реляционные базы создано множество замечательных продуктов, которые необходимо поддерживать и развивать. В эти продукты уже вложены большие деньги и заказчики готовы еще вкладывать деньги в их развитие. Напротив, с использованием ООБД разработано сравнительно мало серьезных коммерческих продуктов, существует мало мощных ООСУБД.

Язык запросов и его стандартизация. Еще в далеком 1986 году был принят первый стандарт SQL-86, который определил всю судьбу реляционных БД. После принятия стандарта все разработчики реляционных СУБД обязаны были следовать ему. Для объектно-ориентированных баз данных пока стандарта языка запросов нет. Сейчас среди разработчиков даже нет единого мнения о том, что этот язык запросов должен делать, не говоря уже о том, как он это должен делать.

Математический аппарат. Для реляционных БД в свое время Эдгар Кодд заложил фундамент математического аппарата реляционной алгебры. Этот мат. аппарат объясняет, как должны выполняться основные операции над отношениями в базе данных, доказывает их оптимальность (либо из него

видно, где надо оптимизировать). С другой стороны, для ООБД нет такого аппарата, даже несмотря на то, что работы в этой области ведутся с 80-х годов. Таким образом, в ООБД пока нет строгих терминов, таких как декартово произведение, отношение и т. д.

Проблема хранения данных и методов. В реляционных БД хранятся только голые данные. Что с ними будет делать приложение, зависит уже от приложения. В ООБД же, напротив должны храниться объекты, а объект — это совокупность его свойств (параметры объекта) и методов (интерфейс объекта). Здесь так же нет единого мнения, как ООБД должна осуществлять хранение объектов и как разработчик должен эти объекты разрабатывать и проектировать. Здесь же возникает и проблема хранения иерархии объектов, хранение абстрактных классов и т.п.

Объектно-ориентированные системы управления базами данных (ООСУБД)

ООСУБД позволяет работать с объектами баз данных так же, как с объектами в программировании в ООЯП. ООСУБД расширяет языки программирования, прозрачно вводя долговременные данные, управление параллелизмом, восстановление данных, ассоциированные запросы и другие возможности. Некоторые объектно-ориентированные базы данных разработаны для плотного взаимодействия с такими объектно-ориентированными языками программирования как Python, Java, C#, Visual Basic .NET, C++, Objective-C и Smalltalk; другие имеют свои собственные языки программирования. ООСУБД используют точно такую же модель, что и объектно-ориентированные языки программирования.

СУБД должна обеспечивать:

- Долговременное хранение
- Использование внешней памяти
- Параллелизм
- Восстановление
- Нерегламентированные запросы

Преимущества использования ООСУБД:

- Отсутствует проблема несоответствия модели данных в приложении и БД (impedance mismatch). Все данные сохраняются в БД в том же виде, что и в модели приложения.
- Не требуется отдельно поддерживать модель данных на стороне СУБД.
- Все объекты на уровне источника данных строго типизированы. Больше никаких строковых имен колонок! Рефакторинг объектно-ориентированной базы данных и работающего с ней кода теперь автоматизированный, а не однообразный и скучный процесс.

Задание: Создать проект объектно-ориентированной БД согласно выбранной теме используя ООСУБД.

Контрольные вопросы:

1. Что такое объектно-ориентированая СУБД?
2. В чем особенность объектно-ориентированных СУБД?

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №41

Тема: «Проектирование объектно-ориентированной СУБД CACHE»

Цель: Научиться создавать WEB-приложения с помощью объектно-ориентированной СУБД CACHE

Время выполнения: 6 уроков

Ход работы:

В конце 1997 года компания InterSystems Corp. выпустила постреляционную СУБД Cache'. Компания и раньше занималась системами управления базами данных, в России активно использовались и продолжают использоваться предшественники Cache': MSM, DTM, ISM. За 4 года вышло несколько версий СУБД Cache', в настоящий момент компания предлагает Cache' 4.1.

Cache' 4.1. - высокопроизводительная промышленная СУБД, интегрированная с технологией разработки Web-приложений - Cache' Server Pages.

СУБД Cache' относится к постреляционным СУБД. Термин "постреляционная СУБД" обозначает принадлежность Cache' к СУБД нового поколения. Имеется в виду не столько аспект времени (Cache' появилась после своих основных реляционных конкурентов), сколько ряд технологических преимуществ: единая архитектура данных и полная поддержка Cache' объектно-ориентированных технологий, о которых будет подробно рассказано ниже.

На рисунке 1 изображены основные элементы архитектуры СУБД Cache': платформы, на которых работает Cache', Многомерный сервер данных, три способа доступа к данным, язык описания бизнес-логики Cache' ObjectScript, интерфейсы к средствам проектирования и разработки приложений и Web-технология Cache' Server Pages. Далее мы подробно остановимся на всех основных элементах архитектуры, которые будут рассмотрены подробнее.

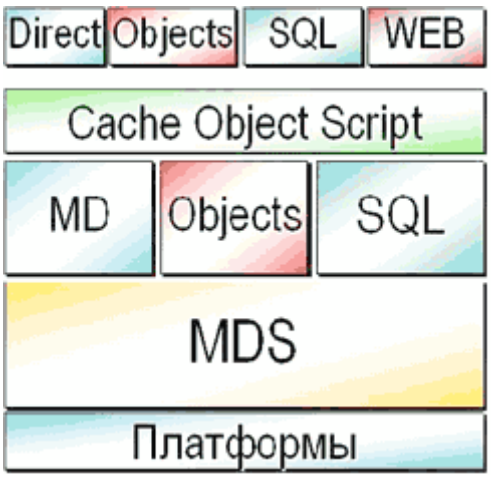


Рисунок 1. Архитектура постреляционной СУБД Cache.

Cache' - кроссплатформенная система. Cache' поддерживает следующие операционные системы: всю линейку Windows, Linux, основные реализации Unix и Open VMS. Планируется поддержка новых реализаций Unix. Большое внимание уделяется новой платформе Itanium.

Данные в Cache' хранятся под управлением Многомерного сервера данных. В основе Cache' лежит транзакционная многомерная модель данных (TMDM), которая позволяет хранить и представлять данные так, как они чаще всего используются. Многомерный сервер данных снимает многие ограничения, накладываемые реляционными СУБД, которые хранят данные в двумерных таблицах, ведь если реляционная модель состоит из большого количества таблиц, что необходимо при работе со сложными структурами данных, это существенно усложняет и замедляет выполнение сложных транзакций и ведет к хранению излишней информации. Cache' хранит данные в виде многомерных разреженных массивов - глобалей. Уникальная транзакционная многомерная модель данных позволяет избежать проблем, присущих реляционным СУБД, оптимизируя данные на уровне хранения.

В отличие от ранних многомерных СУБД, которые были оптимизированы для создания аналитических систем, Cache' ориентирована на системы обработки транзакций (Online Transaction Processing). Многомерный сервер данных Cache' предназначен для обработки транзакций в системах с большими и сверхбольшими БД (сотни гигабайт, терабайты) и

большим количеством одновременно работающих пользователей. Многомерный сервер данных Cache' позволяет разработчикам получить великолепную производительность, отказавшись от хранения избыточных данных и таблиц. Реляционная модель не всегда подходит для описания сложных предметных областей. Транзакционная модель данных Cache' позволяет оптимизировать данные на уровне хранения, поддерживать объектную модель и сложные типы данных. Все эти возможности значительно упрощают создание сложных систем.

В Cache' реализована концепция Единой архитектуры данных. К одним и тем же данным, хранящимся под управлением Многомерного Сервера Данных Cache' есть три способа доступа: прямой, объектный и реляционный:

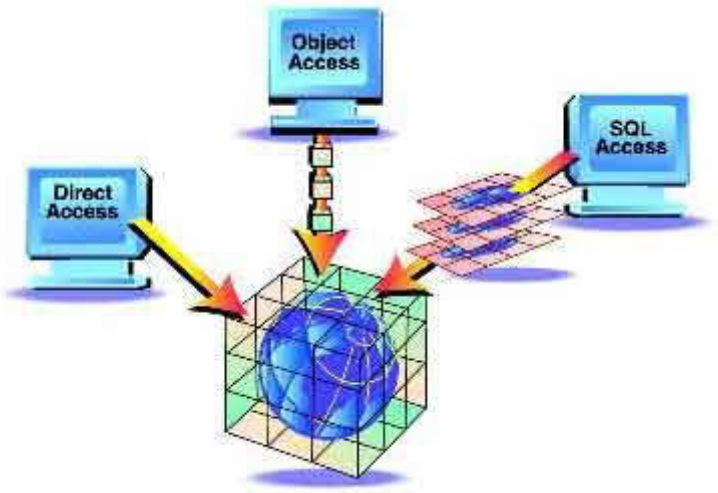


Рисунок 2. Концепция Единой архитектуры данных Cache'.

Cache' Direct Access - прямой доступ к данным, обеспечивает максимальную производительность и полный контроль со стороны программиста. Разработчики приложений получают возможность работать напрямую со структурами хранения. Использование этого типа доступа накладывает определенные требования на квалификацию разработчиков, но понимание структуры хранения данных в Cache' позволяет оптимизировать хранение данных приложения и создавать сверхбыстрые алгоритмы обработки данных.

Cache' SQL - реляционный доступ, обеспечивающий максимальную производительность реляционных приложений с использованием

встроенного SQL. Cache' SQL соответствует стандарту SQL 92. Кроме этого, разработчик может использовать разные типы триггеров и хранимых процедур. Все это позволяет Cache' успешно конкурировать с реляционными СУБД. Даже без использования прямого и объектного доступа приложения на Cache' работают быстрее за счет высокой производительности Многомерного Сервера Данных.

Cache' Objects - объектный доступ, для максимальной продуктивности разработки при использовании Java, Visual C++, VB и других ActiveX-совместимых средств разработки, таких как PowerBuilder и Delphi. В Cache' реализована объектная модель в соответствии с рекомендациями ODMG (Группа управления объектными базами данных - Object Database Management Group). В Cache' полностью поддерживаются наследование (в том числе и множественное), инкапсуляция и полиморфизм. При создании информационной системы разработчик получает возможность использовать объектно-ориентированный подход к разработке, моделируя предметную область в виде совокупности классов объектов, в которых хранятся данные (свойства классов) и поведение классов (методы классов). Cache', поддерживая объектную модель данных, позволяет естественным образом использовать объектно-ориентированный подход как при проектировании (в Rational Rose) предметной области, так и при реализации приложений в ОО-средствах разработки (Java, C++, Delphi, VB). Постреляционная СУБД Cache' конкурирует с объектными СУБД, значительно превосходя их по таким показателям как надежность, производительность и удобство разработки.

Как уже отмечалось, разработчик имеет три способа доступа к одним и тем же данным. Как только определяется класс объектов, Cache' автоматически генерирует реляционное описание этих данных так, что к ним можно обращаться, используя SQL. Подобным же образом, при импорте в словарь данных DDL-описания реляционной базы данных, Cache' автоматически генерирует реляционное и объектное описание данных, открывая тем самым доступ к данным как к объектам. При этом все описания

данных ведутся согласованно, все операции по редактированию проводятся только над одним экземпляром данных. Кроме этого программист может обратиться к тем же данным с помощью прямого доступа.

Cache' позволяет комбинировать три типа доступа, оставляя разработчику свободу выбора. Например, при реализации биллинговой системы объектный доступ может использоваться при описании бизнес-логики приложения и создания пользовательского интерфейса с помощью объектно-ориентированных средств разработки (VB, Delphi, C++), реляционный доступ - для совместимости с другими системами и интеграции с инструментами построения отчетов и аналитической обработки данных (Seagate Info, Cognos, Business Objects). Прямой доступ обеспечивает максимальную производительность и может быть использован для реализации таких операций, в которых применение обычных хранимых процедур, основанных на SQL, не может обеспечить необходимую производительность. В биллинге к таким операциям относятся закрытие периода, массовая загрузка данных (CDR). Использование прямого доступа для реализации подобных операций позволяет увеличить производительность на 1-2 порядка.

Известны случаи перевода в Cache' сложных приложений, которые ранее работали под управлением реляционных СУБД. Например, один из партнеров InterSystems перевел на Cache' биллинговую систему для операторов сотовой связи. Переход осуществлялся следующим образом: сначала существующее приложение с минимальными изменениями переносится под управление Cache'. Приложение на первом этапе работает с Cache', так же как и с реляционной СУБД. Опыт показывает, что даже в этом случае, система начинает работать быстрее. Далее ряд операций был переписан, с использованием прямого способа доступа к БД. На этом этапе удалось увеличить производительность критических операций в десятки и сотни раз.

Кроме этого, можно провести реинжениринг системы. Существует возможность значительно увеличить производительность системы, используя полную поддержку Cache' объектно-ориентированных технологий. При этом можно изменять и структуру базы, и способ работы с Cache' из клиентской части приложения.

Для реализации бизнес-логики БД в СУБД Cache' используется Cache' Object Script. COS - полнофункциональный язык, который имеет все необходимые механизмы для работы с данными с помощью любого способа доступа. С помощью COS разработчик создает методы классов, триггеры, хранимые процедуры, различные служебные программы. В настоящее время ведется работа над созданием еще одного языка описания бизнес-логики - Бейсика. Использование Бейсика позволит облегчить изучение Cache' большому количеству программистов, владеющих этим широко распространенным языком.

СУБД Cache' - открытый продукт, который имеет множество интерфейсов, позволяющих разработчику использовать вместе с Cache' любые современные технологии.

Во-первых, стоит отметить интерфейсы со средствами проектирования и разработки приложений. Специальные компоненты Cache' позволяют проектировать приложения в Rational Rose при объектном подходе, и в ErWin - при реляционном.

Разработчик может реализовывать приложения клиент-сервер, используя практически все средства разработки. При этом он может использовать специальные интерфейсы для прямого и объектного доступа, а стандартные (ODBC, JDBC) - для реляционного. Особый интерес представляет работа с помощью объектного способа доступа в объектно-ориентированных средствах разработки: разработчик работает с классами объектов Cache', обращаясь к их свойствам и выполняя методы классов, так же как и с собственными объектами языка программирования.

Кроме этого, поддерживаются следующие интерфейсы: Native C++, Java, EJB, ActiveX, XML, интерфейсы CallIn и CallOut .

О XML стоит рассказать подробнее. Сейчас уже нет необходимости доказывать преимущества применения XML. Такие задачи как обмен информацией между различными информационными системами, новые протоколы роуминга (например, TAP3), разработка приложений мобильной коммерции решаются с помощью XML. Следовательно, современная СУБД должна предоставлять необходимые механизмы работы с XML.

В Cache' реализована полноценная поддержка XML. Cache' не хранит XML-документы в текстовых файлах, Memo-полях или реляционных таблицах. Полная поддержка Cache' объектной модели позволяет автоматически трансформировать сложные XML-документы в классы объектов Cache'. Из описания классов объектов Cache' можно получить DTD, а сами объекты Cache проецируются в XML-документы. Для создания собственного импорта XML можно воспользоваться SAX-парсером. С помощью Cache' Server Pages, Web-технологии компании InterSystems, можно генерировать не только HTML страницы, но и страницы с XML-содержанием.

Таким образом, появляется возможность использовать XML с Cache' как для обмена информацией между различными информационными системами, так и для реализаций приложений электронной и мобильной коммерции (WAP).

Хотелось бы также подробнее остановится на возможностях разработки Web-приложений в Cache'. Cache' Server Pages - объектная Web-технология Cache', позволяющая использовать производительность и масштабируемость СУБД Cache' для быстрого создания сложных Intranet- и Internet-приложений, взаимодействующих с БД. При разработке Web-приложений, Cache' является не только СУБД, но и Сервером Приложений. В Cache' хранятся классы, соответствующие CSP-страницам, и при обращении к CSP-странице выполняются методы, генерирующие HTML или XML.

Технология Cache' Server Pages - обеспечивает обмен данными между постреляционной СУБД Cache' и Web-сервером, используя стандартные интерфейсы. Такая архитектура позволяет создавать высокопроизводительные, масштабируемые Internet- или Intranet-приложения, так как, во-первых, данные хранятся очень близко к Web-приложению (данные из БД передаются в приложение через высокопроизводительные внутренние интерфейсы, а не через ODBC или JDBC), и, во-вторых, относительно небольшая нагрузка на Web-сервер (Web-сервер только перенаправляет запросы пользователей на Сервер приложений Cache') и высокая производительность СУБД Cache' позволяет обрабатывать запросы большого количества пользователей.

Процесс разработки выглядит следующим образом, дизайнеры занимаются внешним видом Web-приложения, а разработчики с помощью готовых инструментов разработки Web-страниц (например, Macromedia Dreamweaver) или любого текстового редактора дополняют Web-страницы необходимой функциональностью. Для разработки CSP-страниц используются стандартные таги HTML, а также набор дополнительных CSP-тагов и атрибутов для реализации циклов, условий, связывания объектов Cache и формы CSP-страницы, управления данными и т.д. Существует возможность разрабатывать собственные таги приложений (Cache' Application Tags). Cache Server Pages позволяет разработчику использовать для создания методов Cache Object Script, Java и VB script. Механизмы наследования CSP-страниц (CSP - объектно-ориентированная технология, где все CSP-страницы - классы) и собственные таги Cache' обеспечивают возможность повторного использования кода и совместной разработки: например, часть разработчиков будут создавать таги приложений, а другие - использовать созданные таги для конструирования сложных Web-приложений. Такие возможности как поддержка сессии, Гипер-события (изменение содержимого Web-страницы без её перезагрузки), собственные таги позволяют быстро разрабатывать Web-приложения, которые по

функциональности ничем не уступают традиционным приложениям "клиент-сервер".

Партнеры InterSystems используют CSP для написания приложений самой разной сложности. На CSP реализованы различные информационные системы, электронные магазины, системы электронного документооборота, системы для расчета с населением по оплате коммунальных услуг и ряд других интересных приложений.

Разработчики крупных приложений (банковские системы, автоматизированные системы расчетов, системы управления предприятием) предпочитают использовать промышленные СУБД. Cache' обладает всеми характеристиками промышленной системы: высокой производительностью, надежностью, масштабируемостью, открытостью и переносимостью. Ядро Cache' - высокопроизводительный Многомерный сервер данных, ориентированный на обработку транзакций. Для обеспечения надежности в Cache' предусмотрены такие механизмы как журнал до и после записи, теневой сервер, репликация, "горячее" резервное копирование и т.д. Протокол Распределенного Кэша - позволяет строить действительно масштабируемые решения на базе Cache'. Протокол Распределенного Кэша (Cache' Distibuted Cache' Protocol) - уникальная сетевая технологи фирмы InterSystems, которая распределяет базу данных по сети в зависимости от работы приложений, оптимизируя производительность и пропускную способность сети. Cache' - "открытая" система, поддерживается множество интерфейсов к средствам проектирования и разработки приложений. Cache' работает практически на всех популярных платформах с наиболее распространенными Web-серверами. При этом обеспечивается полная переносимость приложений с платформы на платформу.

Еще один важный показатель как для производителя, так и для потребителя системы на базе СУБД - стоимость решения. Этот показатель складывается из стоимости разработанной системы, стоимости аппаратного обеспечения, на котором будет работать система, СУБД, стоимости

внедрения и сопровождения. Решения на базе постреляционной СУБД Cache' выигрывают по стоимости у конкурентов по всем показателям. Разработчик может быстро и качественно создать систему на базе Cache' за счет поддержки объектно-ориентированного подхода и интеграции со средствами проектирования и разработки, СУБД менее требовательна к аппаратному обеспечению: нужной производительности можно добиться на более дешевом сервере, гибкая лицензионная политика позволит снизить стоимость самой СУБД. Решения на базе СУБД легче сопровождать - система очень надежна и не требует сложного администрирования, кроме этого прекрасно работают и мировой, и российский центр технической поддержки.

Один из партнеров InterSystems - разработчик АСР для операторов сотовой связи выиграл тендер у конкурентов, которые предлагали решение на Oracle, после того как был выполнен пилотный проект. В этом проекте в Cache' и Oracle были загружены данные о звонках, и измерялась скорость закрытия периода для загруженных тестовых данных. Хранимая процедура на Cache' работала 40 минут, на Oracle - 2 часа. При этом Cache' работала на сервере стоимостью 5 тысяч $, а Oracle на сервере Sun за 50 тысяч $.

Таким образом, решения на базе Cache' позволяют разработчикам ПО получить дополнительную прибыль от разработки, а потребителям - получить высокопроизводительные решения, сэкономив на приобретении и эксплуатации приложения.

СУБД Cache' становится все более популярной России. На продуктах InterSystems Corp. реализованы банковские системы, автоматизированные системы расчетов для предприятий электросвязи и операторов сотовой связи, системы управления предприятиями, тарификационные системы, Web-порталы и другие интересные приложения. Сегодня есть все предпосылки для роста популярности постреляционной СУБД Cache' в России: промышленная СУБД, поддерживающая объектную модель, предоставляющая разработчику свободу в выборе средств проектирования и разработки, интегрированная с технологией разработки

Web-приложений, позволяет быстро создавать надежные высокопроизводительные решения.

Задание: Создать Web-приложение согласно выбранной теме используя СУБД Cache.

Контрольные вопросы:

1. В чем особенность объектно-ориентированной СУБД CACHE?
2. Где применяется объектно-ориентированная СУБД CACHE?

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №42

Тема: «Проектирование СУБД, основанной на правилах»

Цель: Познакомиться с проектированием СУБД, основанной на правилах.

Время выполнения: 4 урока

Ход работы:

Если внимательно присмотреться к тому, что реально хранится в базе данных, то можно заметить наличие трех различных видов информации. Во-первых, это информация, характеризующая структуры пользовательских данных (описание структурной части схемы базы данных). Такая информация в случае реляционной базы данных сохраняется в системных отношениях-каталогах и содержит главным образом имена базовых отношений и имена и типы данных их атрибутов. Во-вторых, это собственно наборы кортежей пользовательских данных, сохраняемых в определенных пользователями отношениях. Наконец, в-третьих, это правила, определяющие ограничения целостности базы данных, триггеры базы данных и представляемые (виртуальные) отношения. В реляционных системах правила опять же сохраняются в системных таблицах-каталогах, хотя плоские таблицы далеко не идеально подходят для этой цели.

Информация первого и второго вида в совокупности явно описывает объекты (сущности) реального мира, моделируемые в базе данных. Другими словами, это явные факты, предоставленные пользователями для хранения в БД. Эту часть базы данных принято называть экстенсиональной.

Информация третьего вида служит для руководства СУБД при выполнении различного рода операций, задаваемых пользователями. Ограничения целостности могут блокировать выполнение операций обновления базы данных, триггеры вызывают автоматическое выполнение специфицированных действий при возникновении специфицированных условий, определения представлений вызывают явную или косвенную

материализацию представляемых таблиц при их использовании. Эту часть базы данных принято называть интенсиональной; она содержит не непосредственные факты, а информацию, характеризующую семантику предметной области.

Как видно, в реляционных базах данных наиболее важное значение имеет экстенсиональная часть, а интенсиональная часть играет в основном вспомогательную роль. В системах баз данных, основанных на правилах, эти две части как минимум равноправны.

По определению БД называется активной, если СУБД по отношению к ней выполняет не только те действия, которые явно указывает пользователь, но и дополнительные действия в соответствии с правилами, заложенными в саму БД.

Легко видеть, что основа этой идеи содержалась в языке SQL времени System R. На самом деле, что есть определение триггера или условного воздействия, как не введение в БД правила, в соответствии с которым СУБД должна производить дополнительные действия? Плохо лишь то, что на самом деле триггеры не были полностью реализованы ни в одной из известных систем, даже и в System R. И это не случайно, потому что реализация такого аппарата в СУБД очень сложна, накладна и не полностью понятна.

Среди вопросов, ответы на которые до сих пор не получены, следующие. Как эффективно определить набор вспомогательных действий, вызываемых прямым действием пользователя? Каким образом распознавать циклы в цепочке "действие-условие-действие-..." и что делать при возникновении таких циклов? В рамках какой транзакции выполнять дополнительные условные действия и к бюджету какого пользователя относить возникающие накладные расходы?

Масса проблем не решена даже для сравнительно простого случая реализации триггеров SQL, а задача ставится уже гораздо шире. По существу, предлагается иметь в составе СУБД продукционную систему

общего вида, условия и действия которой не ограничиваются содержимым БД или прямыми действиями над ней со стороны пользователя. Например, в условие может входить время суток, а действие может быть внешним, например, вывод информации на экран оператора. Практически все современные работы по активным БД связаны с проблемой эффективной реализации такой продукционной системы.

Вместе с тем, по нашему мнению, гораздо важнее в практических целях реализовать в реляционных СУБД аппарат триггеров. Заметим, что в проекте стандарта SQL3 предусматривается существование языковых средств определения условных воздействий. Их реализация и будет первым практическим шагом к активным БД (уже появились соответствующие коммерческие реализации).

По определению, дедуктивная БД состоит из двух частей: экстенциональной, содержащей факты, и интенциональной, содержащей правила для логического вывода новых фактов на основе экстенциональной части и запроса пользователя.

Легко видеть, что при таком общем определении SQL-ориентированную реляционную СУБД можно отнести к дедуктивным системам. Действительно, что есть определенные в схеме реляционной БД представления, как не интенциональная часть БД. В конце концов не так уж важно, какой конкретный механизм используется для вывода новых фактов на основе существующих. В случае SQL основным элементом определения представления является оператор выборки языка SQL, что вполне естественно, поскольку результатом оператора выборки является порождаемая таблица. Обеспечивается и необходимая расширяемость, поскольку представления могут определяться не только над базовыми таблицами, но и над представлениями.

Основным отличием реальной дедуктивной СУБД от реляционной является то, что и правила интенциональной части БД, и запросы пользователей могут содержать рекурсию. Можно спорить о том, всегда ли

хороша рекурсия. Однако возможность определения рекурсивных правил и запросов дает возможность простого решения в дедуктивных базах данных проблем, которые вызывают большие проблемы в реляционных системах (например, проблемы разборки сложной детали на примитивные составляющие). С другой стороны, именно возможность рекурсии делает реализацию дедуктивной СУБД очень сложной и во многих случаях неразрешимой эффективно проблемой.

Мы не будем здесь более подробно рассматривать конкретные проблемы, применяемые ограничения и используемые методы в дедуктивных системах. Отметим лишь, что обычно языки запросов и определения интенциональной части БД являются логическими (поэтому дедуктивные БД часто называют логическими). Имеется прямая связь дедуктивных БД с базами знаний (интенциональную часть БД можно рассматривать как БЗ). Более того, трудно провести грань между этими двумя сущностями; по крайней мере, общего мнения по этому поводу не существует.

Какова же связь дедуктивных БД с реляционными СУБД, кроме того, что реляционная БД является вырожденным частным случаем дедуктивной? Основным является то, что для реализации дедуктивной СУБД обычно применяется реляционная система. Такая система выступает в роли хранителя фактов и исполнителя запросов, поступающих с уровня дедуктивной СУБД. Между прочим, такое использование реляционных СУБД резко актуализирует задачу глобальной оптимизации запросов.

При обычном применении реляционной СУБД запросы обычно поступают на обработку по одному, поэтому нет повода для их глобальной (межзапросной) оптимизации. Дедуктивная же СУБД при выполнении одного запроса пользователя в общем случае генерирует пакет запросов к реляционной СУБД, которые могут оптимизироваться совместно.

Конечно, в случае, когда набор правил дедуктивной БД становится велик, и их невозможно разместить в оперативной памяти, возникает проблема управления их хранением и доступом к ним во внешней памяти.

Здесь опять же может быть применена реляционная система, но уже не слишком эффективно. Требуются более сложные структуры данных и другие условия выборки. Известны частные попытки решить эту проблему, но общего решения пока нет.

Задание: Спроектировать дедуктивную БД согласно выбранной теме используя СУБД основанную на правилах.

Контрольные вопросы:

В чем особенность СУБД, основанной на правилах?

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №43

Тема: «Системы управления жизненным циклом продукции»

Цель: Познакомиться с системами управления жизненым циклом продукции.

Время выполнения: 4 урока

Ход работы:

Жизненный цикл базы данных (ЖЦ БД) – это процедура создания концептуальной схемы базы данных, определения данных, включаемых в базу данных создание программ обновления и обработки данных.

ЖЦ БД – это процесс проектирования, реализации и поддержания системы базы данных.

ЖЦ БД неразрывно связан с жизненным циклом системы. (ЖЦС). Процедура создания системы, как правило, состоит из таких шагов, как изучение целесообразности, определение требований, проектирование системы, программирование и отладка, установка и запуск, а также оценка работы системы и ее поддержка.

Существует несколько методов ЖЦС. Одни из них основаны на функционально-ориентированном подходе. Это означает, что система рассматривается с точки зрения тех функций, которые она выполняет, а не с точки зрения данных, которыми она оперирует.

Подход, ориентированный на данные, наоборот, основное внимание уделяет анализу данных, нужных для выполнения тех или иных функций. У него есть два преимущества:

Элементы данных являются значительно более стабильной частью системы, чем выполняемые ею функции.

После того, как логическая схема базы данных построена, можно создать любое количество функциональных систем, использующих эту систему.

Функционально-ориентированный подход оправдан лишь для создания временных систем и имеет значительно меньшую ценность в долговременном аспекте.

Жизненный цикл базы данных состоит из шести этапов:

Предварительное планирование

Проверка осуществимости

Определение требований

Концептуальное проектирование

Реализация

Оценка работы и поддержка базы данных.

Предварительное планирование конкретной системы управления базами данных осуществляется в процессе разработки стратегического плана. Ищутся ответы на следующие вопросы:

Сколько используется прикладных программ и какие функции они выполняют?

Какие файлы связаны с каждым из этих приложений?

Какие новые приложения и файлы находятся в процессе создания?

Собранная информация может быть использована для того, чтобы установить связи между текущими приложениями и определить, как используется информация приложений. Она также помогает определить будущие требования к системе и получить от системы базы данных экономическую выгоду. Информация документируется в виде обобщенной концептуальной модели данных.

Проверка осуществимости включает подготовку отчетов по следующим вопросам:

Технологическая осуществимость. Есть ли технология, необходимая для реализации запланированной базы данных?

Операционная осуществимость. Располагает ли компания персоналом, средствами и экспертами, необходимыми для успешного осуществления плана создания базы данных?

Экономическая целесообразность. Можно ли определить выгоды? Окупается ли запланированная система? Можно ли измерить издержки и выгоду?

Определение требований включает выбор целей базы данных, выяснение информационных потребностей различных отделов и руководителей компании и требований к оборудованию и программному обеспечению.

Этап концептуального проектирования включает создание концептуальной схемы базы данных. Спецификации разрабатываются в той степени, которая требуется для перехода к реализации. На этом этапе создаются подробные модели пользовательских представлений данных; затем они интегрируются в концептуальную модель, фиксирующую все элементы корпоративных данных, которые будет содержать база данных.

В процессе реализации базы данных выбирается и приобретается СУБД, затем подробная концептуальная модель превращается в проект реализации базы данных; создается словарь данных, база данных наполняется данными, создаются прикладные программы и обучается пользователь.

Оценка включает опрос пользователей с целью выяснения, какие информационные потребности остались неучтенными. В случае необходимости вносятся изменения. Обеспечивается поддержка системы путем внесения изменений и добавления новых программ и элементов данных по мере расширения и изменения потребностей бизнеса.

Следует отметить коллективную ответственность и функции администрации базы данных за:

Логическое и физическое проектирование базы данных;

Объяснение пользователям структуры базы данных и их обучение процедурам доступа и их обновление;

Защиту базы данных от неверного использования, неправильного доступа и нечаянного повреждения;

Обеспечение приемлемого быстродействия системы.

Задание: Создать дорожную карту и сопутствующую документацию управления жизненным циклом БД, разработанной на ранее.

Контрольные вопросы:

1. Какие вы знаете системы управления жизненным циклом продукции?

Выводы и оформление отчета.

ПРАКТИЧЕСКАЯ РАБОТА №44

Тема: «Базы данных в системах автоматизированного проектирования»

Цель: Научиться разрабатывать БД для САПР

Время выполнения: 4 урока

Ход работы:

В настоящее время практически все программные продукты оперируют большими объемами информации. Это приводит к проблеме ее грамотного хранения, оптимизации и возможности быстрого извлечения для конкретных нужд. Решением этих проблем является организация БД, для хранения, упорядочивания и быстрого поиска, и представления информации.

Работа программного комплекса по проектированию инструментов основана на использовании БД, в которой хранятся все параметры проектируемых инструментов, список ГОСТов с параметрами, ранее созданные проекты инструментов, готовые технологические процессы, операции для технологических процессов, список оборудования для изготовления инструмента, инструменты 2-ого порядка и т.д. БД можно подключить свою, которая имеется на данном объекте, где применяется система. Использование БД позволяет расширять номенклатуру проектируемых изделий, необходимо занести необходимые данные и составить соответствующие таблицы и связи между ними, для того, чтобы программный комплекс мог оперировать этими данными. Поэтому имеется возможность проектирования не только зуборезных инструментов, но и различных осевых инструментов (сверла, развертки, зенкеры) и др.

БД создана в среде MSAccess, но в дальнейшем планируется организация клиент-серверной архитектуры для возможности работать удаленно с БД (БД хранится на сервере, через сервер идет подключение, проверка пользователя и после этого непосредственно работа). Это позволит централизованно управлять содержанием БД и не позволит засорять ее со временем ненужными, мусорными записями

Стоит отметить преимущества использования БД: простота добавления новых записей для обеспечения работы программного комплекса, возможность запомнить результат расчет инструмента и в дальнейшем взять его за основу для расчета такого же типового инструмента, еще более сокращая время работы. Но также имеются и недостатки: необходимость со временем очищать БД от неверных, «мусорных» записей для сохранения быстроты работы программного комплекса, следить за тем, чтобы файл БД не терялся в процессе проектирования, т.к. произойдет потеря всех данных.

В целом применение БД в данном комплексе оправдывает себя и привносит много возможностей в сочетании с небольшими, некритичными минусами.

На рис 1. показана общая структура используемой в программном комплексе БД

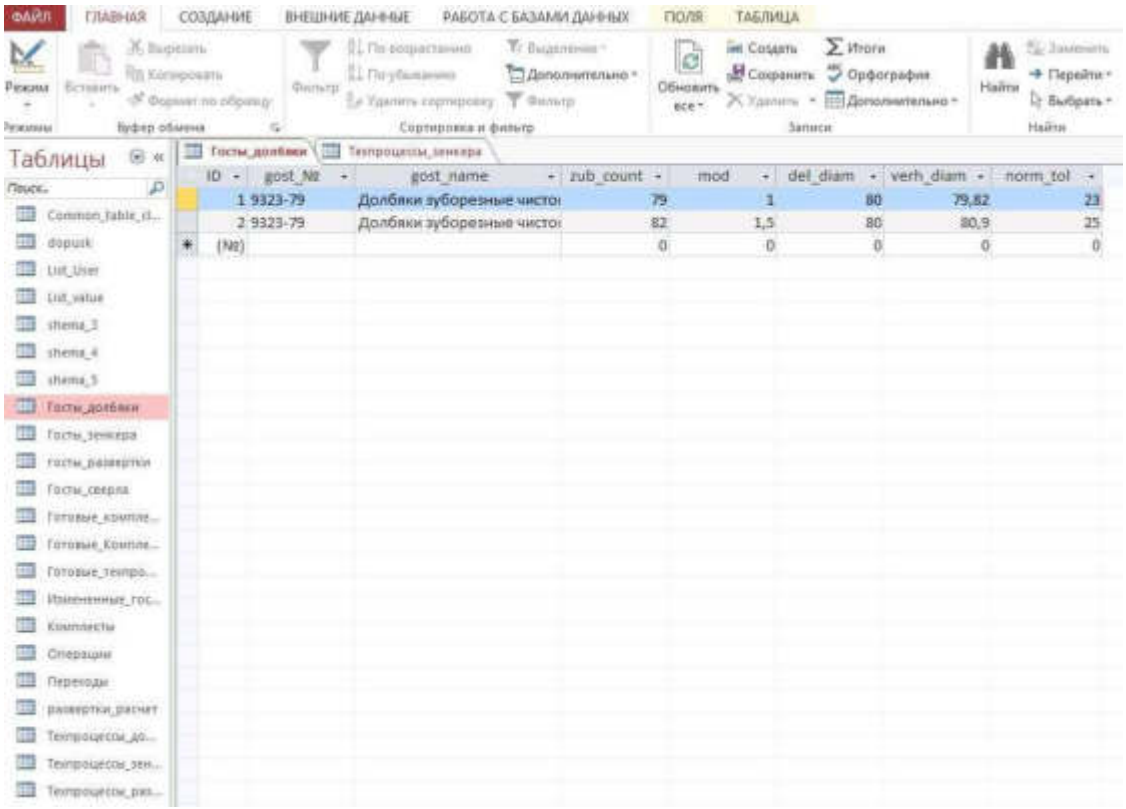


Рис 1. Общая структура используемой в программном комплексе БД

Здесь показаны таблицы, которые содержатся в данной БД (колонка слева). В них хранятся записи ГОСТов на различные изделия. По этим таблицам осуществляется поиск информации. Программный комплекс

передает полученные данные в виде запроса в СУБД (система управления базой данных) MSAccess, где производится поиск нужной информации. Если информация найдена в программу возвращается результат поиска, если нет, то выдается сообщение «не удалось найти информацию».

Как мы видим таблиц достаточно много. Это объясняется тем, что оперировать приходится достаточно большими объемами разнородной информации, которая не связана между собой. Этим обосновано появление такого количества таблиц. Например, таблицы ГОСТов содержат разную информацию для каждого инструмента, отдельно хранятся таблицы необходимые для формирования технологического процесса. Есть также служебные таблицы, которые необходимы для связи главных таблиц, и пользователь не знает об их существовании.

Если есть необходимость изменить что-то в ГОСТе, то по окончании внесения правок эта запись сохраняется в таблицу «измененные_госты_долбяки», которая полностью соответствует по типу информации тому что есть в таблице ГОСТов, но, чтобы не засорять исходную таблицу было принято решение о сохранении в отдельную таблицу. Аналогично происходит и с технологическими процессами: имеется несколько типовых технологических процессов изготовления долбяков, пользователь их видит, если имеется необходимость что-то в нем изменить, он его редактирует и когда происходит сохранение в таблицу «техпроцессы_долбяки», их можно в дальнейшем просмотреть.

Рассмотрим процесс разработки технологических процессов в ручном режиме.

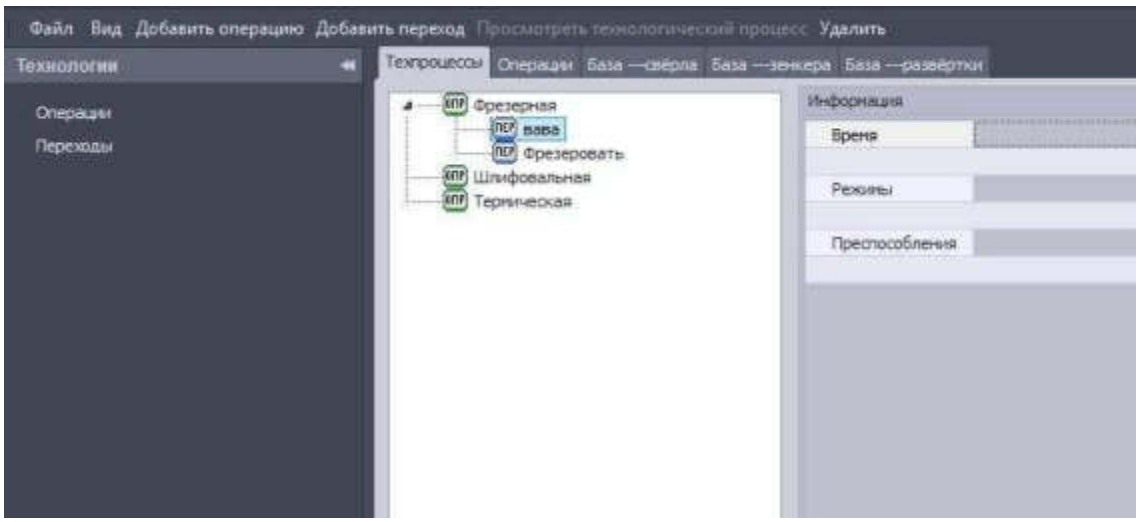


Рис. 2. Последовательность создания технологического процесса

На рис. 2 показан процесс создания технологического процесса. На вкладке технологические процессы видим его представление в виде дерева операции и переходов. Для добавления операции нажимаем пункт «Добавить операцию». Программа делает запрос в БД, в таблицу операции, считывает все записи операции, которые там имеются, и дает пользователю возможность выбрать необходимую операцию. При отсутствии нужного пользователь может перейти в пункт «Операции» и добавить необходимую операцию (рис. 3). Информация сохранится в БД и в дальнейшем будет доступна для использования. Таким образом, формируется технологический как процесс как сказано выше в виде дерева операции.

Добавление перехода осуществляется выбором пункта «Добавить операцию». В окне ввода вводится текст операции, который сохранится в таблице БД «Переходы» и в дальнейшем его можно будет выбрать в соответствующей вкладке.

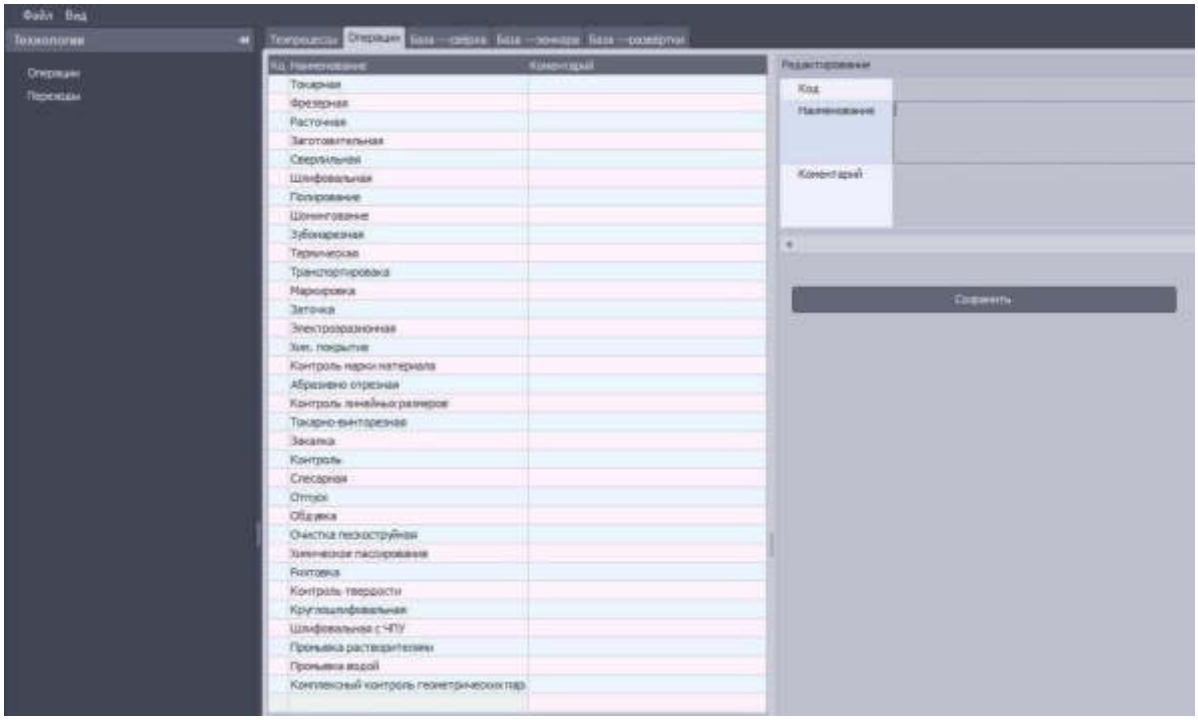


Рис 3. Добавление необходимой операции

После завершения формирования технологического процесса, он сохраняется в таблице «техпроцессы_долбяки» и в дальнейшем может быть использован в таком же проекте, либо проекте аналоге.

По мере использование комплекса БД будет пополнятся и переводится из просто клиентского варианта в клиент-серверный вариант на основе MSSQLServer. База будет храниться на сервере и к ней можно будет подключится удаленно через корпоративную сеть. Это позволит иметь централизованную БД для всего предприятия, организации.

Дальнейшая разработка и совершенствование программного комплекса позволит проектировать более сложные зуборезные инструменты.

Задание: Создать БД базовых простых 3D-моделей для САПР AutoCAD. Проанализировать ход их выполнения и полученный результат.

Контрольные вопросы:

1. Назовите САПР которые используют БД MS?

Выводы и оформление отчета.

Методические рекомендации по выполнению

Лабораторных работ

для обучающихся

по междисциплинарному курсу МДК 01.02 «Хранение, передача и публикация цифровой мультимедийной информации»

профессионального модуля ПМ 02

«Хранение, передача и публикация цифровой мультимедийной информации»

по программам подготовки квалифицированных рабочих, служащих по профессии

09.01.03 «Мастер по обработке цифровой информации

г. Балашиха

2020г.

Пояснительная записка

Цель практических работ - помочь обучающимся систематизировать, закрепить и углубить знания теоретического характера; научить студентов приемам решения лабораторных задач; научить их работать с книгой, нормативно-правовой и распорядительной документацией, пользоваться справочной и научной литературой; формировать умение учиться самостоятельно, т. е. овладевать методами, способами и приемами самообучения, саморазвития и самоконтроля.

Практические работы направлены на овладение студентами умений решения стандартных задач и приобретение навыков практических действий.

Основные требования к содержанию лабораторных работ:

-соответствие содержания практических заданий изученному теоретическому материалу учебной дисциплины; максимальное приближение содержания практических заданий к реальной действительности; поэтапное формирование умения, т.е. движение от знания к умению, от простого умения к сложному и т.д.; использование типовых документов, инструкций, бланков и т.п.

-использование последних версий программного обеспечения**.**

Методические рекомендации по подготовке к практическим работам

Выполнению лабораторной работы предшествует проверка знаний обучающихся - их теоретической готовности к выполнению задания.

Практическая работа предполагает проявление активности и самостоятельности студента при выполнении заданий.

- Прослушайте внимательно инструктаж преподавателя

- Продумайте этапы выполнения задания, распределите время, отберите необходимые обеспечивающие средства.

Подготовка к проведению практических занятий предполагает владение студентами умениями и навыками умственного труда и информационной культуры:

▪ Систематизировать и оформлять записи в виде аннотаций, тезисов, цитат, конспектов.

Подготавливая технические средства и оборудования для выполнения практических работ, следует строго следовать указаниям преподавателя, а также следить за последовательностью деятельности, прописанной в практической работе.

Для обеспечения безопасности работы с персональным компьютером необходимо:

перед началом работы убедиться в исправности электропроводки, выключателей, штепсельных розеток, при помощи которых оборудование включается в сеть, наличии заземления компьютера, его работоспособности. Оценка индивидуальных образовательных достижений по результатам текущего контроля производится в соответствии с универсальной шкалой (таблица).

Процент результативности (правильных ответов)	**Качественная оценка индивидуальных образовательных достижений**	
	балл (отметка)	**вербальный аналог**
90 ÷ 100	5	отлично
80 ÷ 89	4	хорошо
70 ÷ 79	3	удовлетворительно

менее 70	2	не удовлетворительно

Критерии оценки практической работы

При проверке усвоения материала выявляется полнота, прочность усвоения студентом теории и

умение применять ее на практике в знакомых и незнакомых ситуациях.

Отметка зависит также от наличия и характера погрешностей, допущенных студентами.

- *грубая ошибка* – полностью искажено смысловое значение понятия, определения;
- *погрешность* отражает неточные формулировки, свидетельствующие о нечетком представлении рассматриваемого объекта;
- *недочет* – неправильное представление об объекте, не влияющего кардинально на знания, определенные программой обучения;
- *мелкие погрешности* – неточности в устной и письменной речи, не искажающие смысла ответа или решения, случайные описки и т.п.

Исходя из норм (пятибалльной системы), заложенных во всех предметных областях выставляете отметка:

«5» ставится при выполнении всех заданий полностью или при наличии 1-2 мелких погрешностей; «4» ставится при наличии 1-2 недочетов или одной ошибки:

«3» ставится при выполнении 2/3 от объема предложенных заданий;

«2» ставится, если допущены существенные ошибки, показавшие, что студент не владеет обязательными умениями по данной теме в полной мере (незнание основного программного материала):

«1» – отказ от выполнения учебных обязанностей.

Комплект материалов для оценки сформированности общих и профессиональных компетенций

Практическое занятие № 1

«Подключение периферийных устройств к ПК»

Цель работы: изучить организацию, работу и подключение последовательных и параллельных интерфейсов.

Студент должен

уметь:- осуществлять установку и конфигурирование персональных компьютеров и подключение периферийных устройств;

-подготавливать компьютерную систему к работе;

знать: способы подключения стандартных и нестандартных периферийных устройств (ПУ).

Методические указания к работе

Последовательность действий:

1) Установить в корпус блок питания

Рисунок 1 Корпус ПК Рисунок 2 Блок питания

2) Установить в корпус НЖМД, НГМД, CD-ROM

Рисунок.3 НЖМД Рисунок 4 CD-ROM Рисунок 5 НГМД

3) Установить в материнскую плату ЦП с системой охлаждения

Рисунок 6 Мат.плата Рисунок 7 ЦП Рисунок 8 Куллер

4) Установить Мат. Плату с ЦП в корпус ПК

Рисунок 9 Корпус с МП

5) Подключить провода питания к МП и другим устройствам ПК, соединить интерфейсными кабелями установленное оборудование

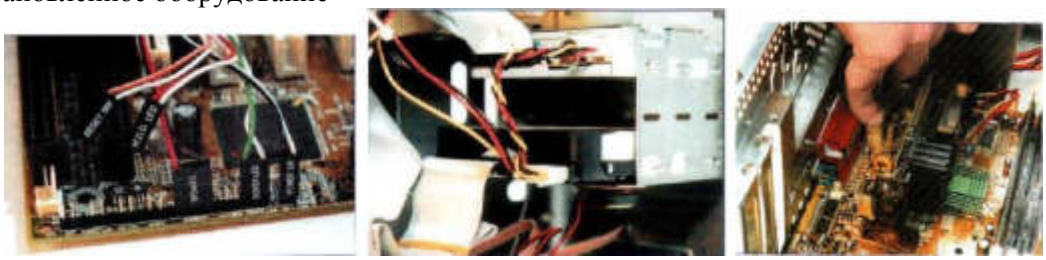


Рисунок 10 Подключение кнопок и светодиодов передней панели корпуса, соединение установленного оборудования, подключение питающих кабелей.

6) Установить ОЗУ и ГК

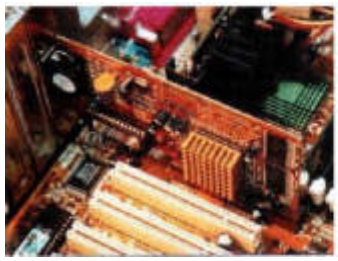


Рисунок 11 установка ГК

Рисунок 12 Установка ОЗУ

7) Проверить еще раз всю проделанную работу, подключить клавиатуру, мышь и монитор и предоставить результат преподавателю для дальнейшей проверки и запуска ПК.

Контрольные вопросы:

1) Для чего нужны перемычки в НЖМД и CD-ROM?
2) Какие интерфейсы используются под ГК?
3) Как сбросить настройки БИОС?
4) Как посчитать необходимую мощность БП?
5) Какие интерфейсы используются при подключении оборудования?
6) Как определить положительный вывод разъемов светодиодов передней панели?

Практическая работа №2 -2часа

Установка операционной системы Windows

1. Цель работы: изучить этапы и способы установки Windows 7

2. Оборудование, приборы, аппаратура, материалы: персональный компьютер, видеоматериал, рабочая тетрадь.

3. Краткие теоретические сведения.

Обзор Windows 7

Windows 7 – наиболее новая клиентская ОС фирмы Microsoft, выпущенная в октябре 2009 года. От предыдущих версий Windows она отличается значительно более удобным пользовательским интерфейсом, с которым Вы и познакомитпесь в данной работе, и более высоким быстродействием. Система все больше завоевывает популярность пользователей.

Запуск системы

- Включите компьютер с инсталлированной Windows 7.
- Менее чем через минуту система загружается и готова к работе.
- Выводится начальное меню с именами пользователей.

Вход в систему и аутентификация пользователя

Выберите Ваше имя пользователя и кликните мышкой по картинке рядом с именем. Как правило, в систему уже введено стандартное имя User. Если для пользователя установлен пароль, введите его.

После входа в систему на экране визуализируется рабочий стол.

Структура рабочего стола, мой компьютер, панель управления

Рабочий стол состоит из иконок приложений (например, Internet Explorer) и панели задач (taskbar) – обычно серого цвета, в нижней части. В левом нижнем углу расположена кнопка Start, при нажатии на которую пользователь может выбрать начальное действие – запуск какого-либо приложения, создание документа и др.

Вид и фон рабочего стола при разных настройках могут отличаться. Для изменения фона рабочего стола необходимо на фоновом рисунке нажать правую кнопку мыши и в контекстном меню выбрать Properties / Desktop, после чего выбрать нужный рисунок фона в выпадающем списке.

Основные пункты стартового меню, визуализируемого в результате нажатия кнопки Start:

- Computer – информация о компьютере, его ресурсах, устройствах, имени, установеленной на нем ОС
- Documents – стандартная папка для создаваемых документов (Вы можете помещать документы и в любую другую более удобную Вам папку)
- Control Panel – панель управления (рис. 36.3)
- Search programs and files – поиск и запуск программ и открытие файлов
- (в нижней части) Shut down – выход из Вашего пользовательского сеанса, выключение компьютера или перезапуск системы.

Рассмотрите более подробно панель управления. Она позволяет управлять ресурсами компьютера. Например, пункт Programs and Features позволяет инсталлировать новые программы, деинсталлировать или инсталлировать вновь ("ремонтировать") уже установленные.

Выберите в стартовом меню пункт Мой компьютер. При этом в специальном окне визуализируется информация о состоянии компьютера.

В окне Мой компьютер визуализируется информация о дисках и некоторых наиболее важных папках и предлагается набор возможных действий и набор других информационных узлов для перехода к ним (например, Network).

Для визуализации основных свойств компьютера (системной информации) выберите в стартовом меню: Мой компьютер / (Правая кнопка мыши) / Свойства. Возникает окно с системной информацией.

Вы видите информацию об ОС, объеме памяти, типе процессора и ряд ссылок, например, Device Manager, кликнув на которую, получите подробную информацию о составе оборудования компьютера и установленных драйверах. Интерфейс оформлен в виде веб-страницы.

Работа с файлами и папками

Работа с файлами и папками (folders) – хранилищами ссылок на файлы и другие папки – осуществляется с помощью программы Windows Explorer.

Выбор файла или папки в директории выполняется одним кликом мышки, вход в директорию или открытие файла – двойным кликом мышки на имени директории или файла. При этом для файла выполняется действие его открытия, зависящее от его типа, - для текстовых файлов – вызов соответствующего редактора (notepad, WordPad, MS Word и др.), для файлов .pdf – вызов Adobe Acrobat, для исполняемых кодов или командных файлов – запуск соответствующей программы или скрипта и т.д. Поэкспериментируйте на своем компьютере с навигацией по файлам и папкам и открытием файлов с документами.

Есть несколько способов запустить программу:

- из Windows Explorer – дважды кликнуть на имени ее файла;
- из меню Start – выбрать пункт Search programs and files. Это – одно из самых удобных нововведений в пользовательском интерфейсе системы Windows 7. При поиске, по мере набора имени программы, выводятся списки программ и файлов с таким именем (префиксом имени), что позволяет очень комфортно выбирать программу для запуска. Выбрав имя редактора notepad. получите окно для его запуска кликом мышки на имени программы.
- Запуск программы из командной строки (Command Prompt): выберите Start / Search Programs and Files / cmd. После запуска командного процессора визуализируется его окно. В окне командной строки наберите имя программы (например, notepad) и нажмите Enter.

Сетевые установки

Для подсоединения компьютера к локальной TCP/IP - сети необходимо выполнить для него сетевые установки – задать IP-адрес и сетевую маску.

Физическое подсоединение к сети сделайте (проверьте) путем подсоединения к сетевому разъему (RJ45) сетевого кабеля вида twisted pair (витая пара), который соединяет Ваш компьютер с сетевым концентратором (hub) или переключателем (switch). Наличие физического соединения индицируется зеленым световым индикатором (проверьте).

Для соединения в сеть служит сетевая карта (сетевой адаптер). Ваша задача – правильно задать IP-адрес компьютера.

Работа на удаленных компьютерах

При работе в локальной сети очень полезная возможность Windows 7 – удаленный вход на другой компьютер Вашей локальной сети. В Windows такая функция системы называется Remote Desktop Connection (удаленный рабочий стол). Для соединения Вы должны знать имя другого компьютера, например, **aphrodite**.

Настройка оборудования и звука в Windows 7

В этом меню в панели управления теперь собраны все настройки, касающиеся внешнего оборудования Вашего компьютера.

1.**Устройства и принтеры**. Здесь собрано все оборудование, которое так или иначе можно отключать и подключать к компьютеру через порт или через сетевое соединение. Так же, здесь отображается сам компьютер.

1. Для каждого устройства доступны как общие настройки, такие как обновление драйвера или удаление его, так и индивидуальные. К примеру, если кликнуть правой кнопкой мыши по значку мыши мы увидим, что возможно прямо отсюда изменить параметры мыши и указателей.

2. Если устройство подключено к компьютеру, но не отображается в списке, то Вам следует воспользоваться мастером добавления устройства, благодаря которому новое устройство будет распознано и установлено. Для этого нажмите кнопку *"Добавление устройства"* в верхнем левом углу.

2. **Автозапуск**. Выбрав пункт автозапуск, Вы сможете настроить автоматический запуск определенных типов устройств или носителей.

3. **Звук**. В **настройках звука** есть выбор устройства для воспроизведения и для записи по умолчанию. Здесь, как и в меню персонализации, вы можете настроить и выбрать звуковую схему оформления. Так же, здесь появилась функция, позволяющая автоматически понижать громкость всех системных звуков при поступлении звонка по Voip-каналу. Автоматически громкость может быть понижена на 80 или 50%, а так же Вы можете отключить все системные звуки при поступлении звонка или оставить их без изменения.

Выход из системы

Для выхода из Вашего сеанса пользователя выберите Start / Shut down.

4. Задания

Задание 1. Просмотрите предложенный видеоматериал. Установка и настройка операционной системы Windows 7.

Задание 2. Напишите понятия и классификацию:

- Операционная система (ОС)
- Операционная система как виртуальная машина
- Операционная система как менеджер ресурсов
- Операционная система как защитник пользователей и программ
- Операционная система как постоянно функционирующее ядро
- Классификация ОС

Задание 3. Опишите этапы подготовки к установке Windows 7

Задание 4. Заполните таблицу. Установка и настройка Windows 7

№	Задание	**Этапы установки Windows 7.** Опишите действия на каждом из этапов.					
1.	Установка Windows 7	Подготовка жесткого диска.	Загрузка с установочного диска.	Определение параметров установки.	Следование инструкциям мастера установки.	Установка драйверов.	Активация ОС

<table>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td rowspan="3">2.</td><td rowspan="3">Первичные настройки Windows 7</td><td colspan="6">Основные этапы настройки Windows 7. Опишите действия на каждом из этапов.</td></tr>
<tr><td colspan="2">Персонализация</td><td colspan="2">Свойства компьютера</td><td>Учетные записи</td><td>Антивирусное ПО</td></tr>
<tr><td colspan="2"></td><td colspan="2"></td><td></td><td></td></tr>
<tr><td rowspan="2">3.</td><td rowspan="2">Контекстное меню Windows 7</td><td colspan="6">Опишите работу с контекстным меню Windows 7</td></tr>
<tr><td colspan="6"></td></tr>
<tr><td rowspan="3">4.</td><td rowspan="3">Оборудование и звук Windows 7</td><td colspan="6">Этапы настройки оборудования и звука Windows 7. Опишите действия на каждом из этапов.</td></tr>
<tr><td colspan="2">Устройства и принтеры</td><td colspan="2">Автозапуск</td><td colspan="2">Звук</td></tr>
<tr><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td rowspan="3">5.</td><td rowspan="3">Сетевые настройки Windows 7</td><td colspan="6">Этапы настройки домашней сети Windows 7. Опишите действия на каждом из этапов.</td></tr>
<tr><td colspan="6">Настройка сетевого подключения Windows 7.
В ОС Windows 7 настройка сетевого подключения сводится к следующим этапам:
• Подсоедините компьютер к свитчу, роутеру или концентратору. Для этого подключите коннектор RJ-45 к порту Ethernet. Чтобы быть уверенным в успешном коннекте к сети, щелкните вкладку «Сеть». Откроется окно со значками, свидетельствующими о добавлении устройств в сеть.
• В Windows 7 настройки сетевого подключения в основном вращаются вокруг апплета «Сеть». Эта папка является удобным инструментом доступа к узлам сети. Итак, если не обнаруживается значок, то возможно было отключено обнаружение или общий доступ.</td></tr>
<tr><td>1. Разработка схемы сети</td><td>2. Выбор необходимого оборудования и поставщика услуг Интернета (дополнительно)</td><td>3. Настройка маршрутизатора</td><td>4. Подключение маршрутизатора к Интернету (дополнительно)</td><td>5. Подключение компьютеров и устройств к сети</td><td>6. Создание домашней группы или включение общего доступа к файлам и принтерам</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

5. Содержание отчета

Отчет должен содержать:

1. Название работы.
2. Цель работы.
3. Задание и его решение.
4. Вывод по работе.

6. Контрольные вопросы

1. Что такое операционная система?
2. Как сделать загрузочный диск (флэшнакопитель)?

3. Какие способы запуска программ в Windows 7 применяются?

Практическая работа №3
Установка операционной системы Ubuntu (Linux)"-2 часа

Цель работы: приобрести опыт установки операционной системы Linux.

План проведения занятия:

1. Закрепить знания о работе с программой VirtualBox.
2. Создать виртуальную машину исходя из предоставленной информации о минимальные аппаратные требования, предлагаемой к установке и изучению операционной системы (ОС).
3. Установить ОС на виртуальный компьютер. Разобрать процесс установки ОС на этапы.
4. Познакомиться с основными группами программ входящих в состав ОС.
5. Подготовить отчет и сдать преподавателю в соответствии с графиком.

Оборудование:

Аппаратная часть: персональный компьютер, сетевой или локальный принтер . *Программная часть:* программа VirtualBox, установочный диск либо образ диска с ОС Linux Ubuntu 9, текстовый процессор Microsoft Word.

Краткие теоретические сведения:

Linux (полное название GNU/Linux, произносится «гну слэш линукс́») — общее название UNIX-подобных операционных систем на основе одноимённого ядра и собранных для него библиотек и системных программ, разработанных в рамках проекта GNU.

GNU/Linux работает на PC-совместимых системах семейства Intel x86, а также на IA-64, AMD64, PowerPC, ARM и многих других.

К операционной системе GNU/Linux также часто относят программы, дополняющие эту операционную систему, и прикладные программы, делающие её полноценной многофункциональной операционной средой.

В отличие от большинства других операционных систем, GNU/Linux не имеет единой «официальной» комплектации. Вместо этого GNU/Linux поставляется в большом количестве так называемых дистрибутивов, в которых программы GNU соединяются с ядром Linux и другими программами.

Дистрибутив — это не просто набор программ, а ряд решений для разных задач пользователей, объединённых едиными системами установки, управления и обновления пакетов, настройки и поддержки.

Самые распространённые в мире дистрибутивы:

- Ubuntu — быстро завоевавший популярность дистрибутив, ориентированный на лёгкость в освоении и использовании.
- openSUSE — бесплатно распространяемая версия дистрибутива SuSE, принадлежащая компании Novell. Отличается удобством в настройке и обслуживании благодаря использованию утилиты YaST.
- Fedora — поддерживается сообществом и корпорацией RedHat, предшествует выпускам коммерческой версии RHEL.
- Debian GNU/Linux — международный дистрибутив, разрабатываемый обширным сообществом разработчиков в некоммерческих целях. Послужил основой для создания множества других дистрибутивов. Отличается строгим подходом к включению несвободного ПО.
- Mandriva — французско-бразильский дистрибутив, объединение бывших Mandrake и Conectiva.
- Archlinux — ориентированный на применение самых последних версий программ и постоянно обновляемый, поддерживающий одинаково как бинарную, так и установку из исходных кодов и построенный на философии простоты KISS, этот дистрибутив ориентирован на компетентных пользователей, которые хотят иметь всю силу и модифицируемость Linux, но не в жертву времени обслуживания.

Помимо перечисленных, существует множество других дистрибутивов, как базирующихся на перечисленных, так и созданных с нуля и зачастую предназначенных для выполнения ограниченного количества задач.

В отличие от Microsoft Windows (Windows NT), Mac OS (Mac OS X) и коммерческих

UNIX-подобных систем, GNU/Linux не имеет географического центра разработки. Нет и организации, которая владела бы этой системой; нет даже единого координационного центра. Программы для Linux — результат работы тысяч проектов. Некоторые из этих проектов централизованы, некоторые сосредоточены в фирмах. Многие проекты объединяют хакеров со всего света, которые знакомы только по переписке. Создать свой проект или присоединиться к уже существующему может любой и, в случае успеха, результаты работы станут, известны миллионам пользователей. Пользователи принимают участие в тестировании свободных программ, общаются с разработчиками напрямую, что позволяет быстро находить и исправлять ошибки и реализовывать новые возможности.

История развития UNIX-систем. GNU/Linux является UNIX-совместимой, однако основывается на собственном исходном коде

Именно такая гибкая и динамичная система разработки, невозможная для проектов с закрытым кодом, определяет исключительную экономическую эффективность GNU/Linux. Низкая стоимость свободных разработок, отлаженные механизмы тестирования и распространения, привлечение людей из разных стран, обладающих разным видением проблем, защита кода лицензией GPL — всё это стало причиной успеха свободных программ.

Конечно, такая высокая эффективность разработки не могла не заинтересовать крупные фирмы, которые стали открывать свои проекты. Так появились Mozilla, OpenOffice.org, свободный клон Interbase (Borland).

Ход работы:

Системные требования ОС Linux Ubuntu 9:

- Процессор: 1 GHz, 32-разрядный;
- Оперативная память: 256 Mb (32-bit)
- Свободное дисковое пространство: 2 Гбайт HDD + 256 Мбайт для swap.
- Видеоадаптер: 64 MB памяти;
- Устройство чтения DVD-дисков.

В процессе установки не забываем регистрировать основные этапы установки в тетради.

1. Создадим виртуальную машину руководствуясь инструкциями лабораторной работы 1, учитываем тип операционной системы, а также минимальные системные требования.
2. Загружаем предлагаемый образ для установки Linux Ubuntu и выбираем язык установки (рис. 24):

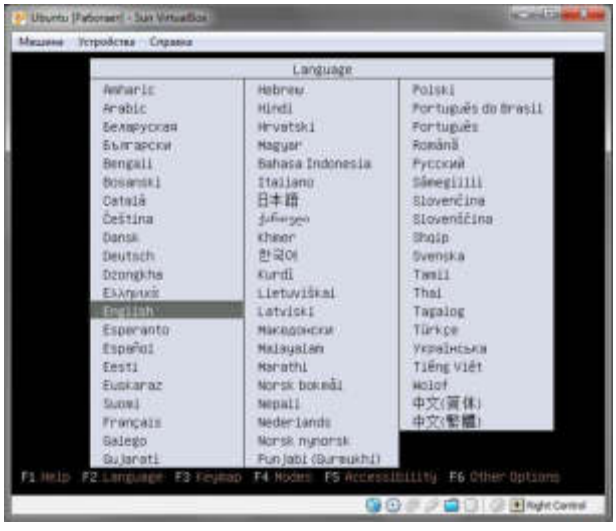


Рис. 24. Выбор языка установки

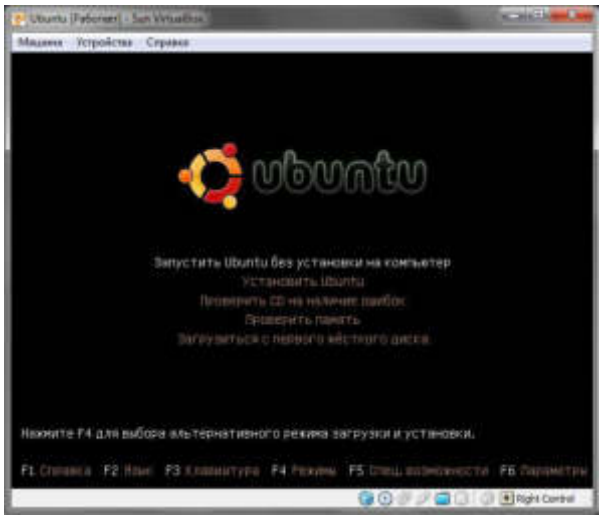


Рис. 25. Меню установки:

3. Выбираем пункт «Установить Ubuntu» (рис. 25). И после непродолжительного ожидания увидим приветствие с предложением начать установку:

ис. 26. Подготовка к установке

4. Нажимаем кнопку «Forward» («Далее»), и увидим меню, в котором можно выбрать часовой пояс (рис. 27):

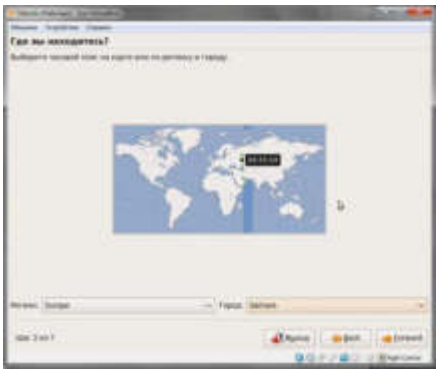


Рис. 27. Настройка часового пояса

5. Нажимаем кнопку «Forward» («Далее»), и увидим меню выбора раскладки клавиатуры (рис. 28):

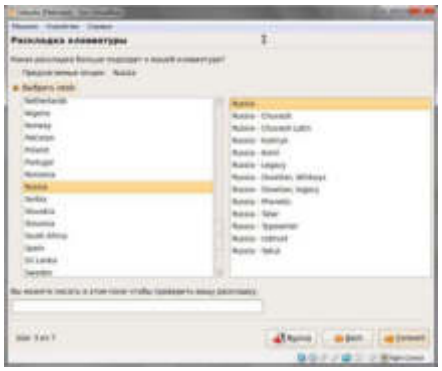


Рис. 28. Настройка раскладки клавиатуры

6. Нажимаем кнопку «Forward» («Далее»), и увидим меню подготовки разделов жесткого диска к установке Ubuntu (рис. 29).

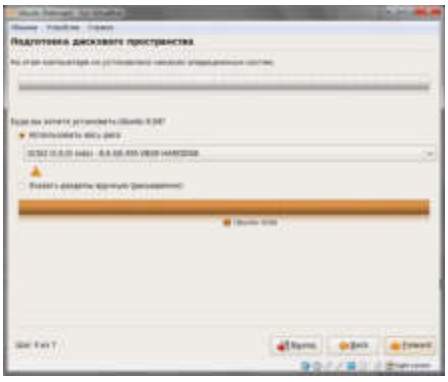


Рис. 29. Подготовка дискового пространсва

7. Выбираем пункт «Использовать весь диск». Пункт «Указать разделы вручную (расширенно)» выполняется только по указанию преподавателя, иначе нажимать не следует.
8. Нажимаем «Forward» («Далее») и видим меню для идентификации будущего пользователя ОС. Необходимо задать имя (Коля, Вася, Маша и т.п.). А также используемое при входе. Во избежание проблем с утратой логина и пароля всем студента задать:

- имя используемое для входа в систему (login) – user
- пароль чтобы держать запись в сохранности (password) – 123456

На вопрос о том, что пароль слишком простой и надо его сменить ответить отрицательно.

- имя компьютера в сети уточнить у преподавателя.

Рис. 30. Меню идентификации будущего пользователя

9. После заполнения всех пунктов меню идентификации, нажимаем «Forward» («Далее»), и видим окно со сводной информацией о будущей ОС.

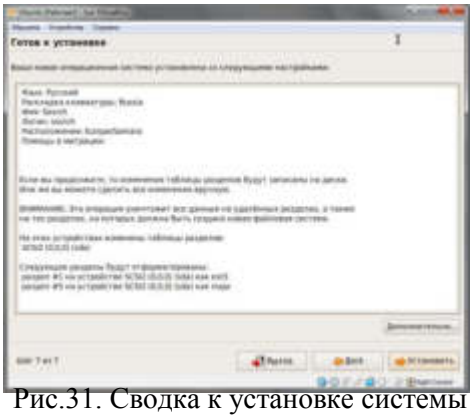


Рис.31. Сводка к установке системы

10. Окно со сводкой о будущей системе представлено - последняя возможность отменить установку или внести какие-то изменения (рис. 31). Нажимаем кнопку «Установить». И в следующем возникшем окне нажимаем «Перезагрузить».
11. Инструкцией предполагается извлечение физического диск, но так как мы работаем с образом, нам необходимо его извлечь, для этого, во время перезагрузки, в верхнем меню окна программы VirtualBox находим пункт «Устройства», и в появившемся меню пункт «Отключить CD/DVD-ROM» (рис. 32). Затем нажмите сочетание клавиш Ctrl+R – перезагрузка виртуальной машины.

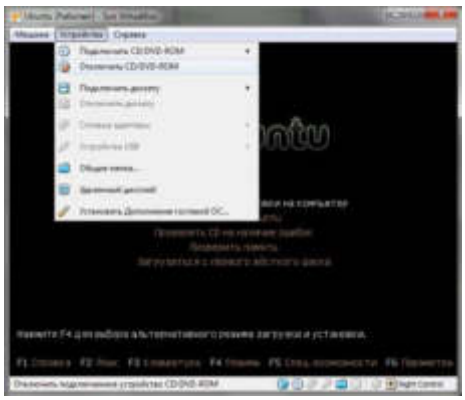


Рис. 32. Меню отключения CD/DVD-ROM

12. И после небольшого диалога с вводом имени пользователя и пароля (имя пишется без заглавных букв) мы увидим ОС Linux Ubuntu с подключенным рабочим столом GNOME (рис. 33).

Рис. 33. ОС Linux Ubuntu с подключенным рабочим столом GNOME.

13. Установите дополнения к гостевой ОС (см. лаб. 1). После монтирования нового носителя (смотри на рабочем столе), найдите на нем файл avtoran.sh, запустите его двойным кликом и в появившемся диалоговом окне нажмите запустить через терминал, далее процесс будет автоматизирован.

14. Подготовить отчет о выполнении лабораторной работы, описать основные этапы установки ОС. Сдать отчет преподавателю в соответствии с графиком.

Контрольные вопросы:

1. Что такое Linux?
2. Что такое дистрибутив?
3. Перечислите основные дистрибутивы Linux. Объясните в чем их отличие.
4. Какую файловую систему использует для работы установленный Вами дистрибутив?
5. Перечислите основные этапы установки операционной системы.

5. Содержание отчета

Отчет должен содержать:

1. Название работы.
2. Цель работы.
3. Задание и его решение.
4. Вывод по работе

Практическая работа №5

Ввод, вывод цифровой мультимедийной информации"

1. **Цель работы:** изучить способы представления текстовой, графической, звуковой информации и видеоинформации, научиться записывать числа в различных системах счисления.

5. **Порядок выполнения работы:**

Повторить требования по соблюдению техники безопасности.

Включение ПК должно производиться в следующей последовательности:

- включить принтер (если он нужен);
- включить монитор;
- включить системный блок.

Перед выключением компьютера завершите все работающие программы и подождите 1-2 сек. (это необходимо, если на вашем ПК предусмотрено кэширование дисков). Далее необходимо:

- выключить системный блок;
- выключить принтер (если он был включен);
- выключить монитор.

6.1 . Ознакомиться с пунктами практической работы;

6.2. Оформите свой отчет согласно седьмому пункту данной практической работы;

6.3. Выполните задание в соответствии со своим вариантом;

6.4 Сделайте вывод о проделанной работе.

7. Содержание отчета:

7.1 Название, цель работы, задание данной практической работы.
7.2 Номер варианта, условие задачи своего варианта и ее решение.
7.3 Перечень контрольных вопросов.
7.4 Вывод о проделанной работе.

8. Теоретические сведения и методические указания

Дискретное представление информации: кодирование цветного изображения в компьютере (растровый подход). Представление и обработка звука и видеоизображения.

Вся информация, которую обрабатывает компьютер должна быть представлена двоичным кодом с помощью двух цифр 0 и 1. Эти два символа принято называть двоичными цифрами или битами. С помощью двух цифр 0 и 1 можно закодировать любое сообщение. Это явилось причиной того, что в компьютере обязательно должно быть организованно два важных процесса: кодирование и декодирование.

Кодирование– преобразование входной информации в форму, воспринимаемую компьютером, то есть двоичный код.

Декодирование– преобразование данных из двоичного кода в форму, понятную человеку.

С точки зрения технической реализации использование двоичной системы счисления для кодирования информации оказалось намного более простым, чем применение других способов. Действительно, удобно кодировать информацию в виде последовательности нулей и единиц, если представить эти значения как два возможных устойчивых состояния электронного элемента:

0 – отсутствие электрического сигнала;

1 – наличие электрического сигнала.

Эти состояния легко различать. Недостаток двоичного кодирования – длинные коды. Но в технике легче иметь дело с большим количеством простых элементов, чем с небольшим числом сложных.

Способы кодирования и декодирования информации в компьютере, в первую очередь, зависит от вида информации, а именно, что должно кодироваться: числа, текст, графические изображения или звук.

Аналоговый и дискретный способ кодирования

Человек способен воспринимать и хранить информацию в форме образов (зрительных, звуковых, осязательных, вкусовых и обонятельных). Зрительные образы могут быть сохранены в виде изображений (рисунков, фотографий и так далее), а звуковые — зафиксированы на пластинках, магнитных лентах, лазерных дисках и так далее.

Информация, в том числе графическая и звуковая, может быть представлена в аналоговой или дискретной форме. При аналоговом представлении физическая величина принимает бесконечное множество значений, причем ее значения изменяются непрерывно. При дискретном представлении физическая величина принимает конечное множество значений, причем ее величина изменяется скачкообразно.

Примером аналогового представления графической информации может служить, например, живописное полотно, цвет которого изменяется непрерывно, а дискретного– изображение, напечатанное с помощью струйного принтера и состоящее из отдельных точек разного цвета. Примером аналогового хранения звуковой информации является виниловая пластинка (звуковая дорожка изменяет свою форму непрерывно), а дискретного– аудиокомпакт-диск (звуковая дорожка которого содержит участки с различной отражающей способностью).

Преобразование графической и звуковой информации из аналоговой формы в дискретную производится путем дискретизации, то есть разбиения непрерывного графического изображения и непрерывного (аналогового) звукового сигнала на отдельные элементы. В процессе дискретизации производится кодирование, то есть присвоение каждому элементу конкретного значения в форме кода.

Дискретизация– это преобразование непрерывных изображений и звука в набор дискретных значений в форме кодов.

Кодирование изображений

Создавать и хранить графические объекты в компьютере можно двумя способами – как *растровое* или как *векторное* изображение. Для каждого типа изображений используется свой способ кодирования.

Кодирование растровых изображений

Растровое изображение представляет собой совокупность точек (пикселей) разных цветов. Пиксель– минимальный участок изображения, цвет которого можно задать независимым образом.

В процессе кодирования изображения производится его пространственная дискретизация. Пространственную дискретизацию изображения можно сравнить с построением изображения из мозаики (большого количества маленьких разноцветных стекол). Изображение разбивается на

отдельные маленькие фрагменты (точки), причем каждому фрагменту присваивается значение его цвета, то есть код цвета (красный, зеленый, синий и так далее).

Для черно-белого изображения информационный объем одной точки равен одному биту (либо черная, либо белая – либо 1, либо 0).

Для четырех цветного – 2 бита.

Для 8 цветов необходимо – 3 бита.

Для 16 цветов – 4 бита.

Для 256 цветов – 8 бит (1 байт).

Качество изображения зависит от количества точек (чем меньше размер точки и, соответственно, больше их количество, тем лучше качество) и количества используемых цветов (чем больше цветов, тем качественнее кодируется изображение).

Для представления цвета в виде числового кода используются две обратных друг другу цветовые модели: **RGB** или **CMYK**. Модель RGB используется в телевизорах, мониторах, проекторах, сканерах, цифровых фотоаппаратах… Основные цвета в этой модели: красный (Red), зеленый (Green), синий (Blue). Цветовая модель CMYK используется в полиграфии при формировании изображений, предназначенных для печати на бумаге.

Цветные изображения могут иметь различную глубину цвета, которая задается количеством битов, используемых для кодирования цвета точки.

Если кодировать цвет одной точки изображения тремя битами (по одному биту на каждый цвет RGB), то мы получим все восемь различных цветов.

R	G	B	Цвет
1	1	1	Белый
1	1	0	Желтый
1	0	1	Пурпурный
1	0	0	Красный
0	1	1	Голубой
0	1	0	Зеленый
0	0	1	Синий
0	0	0	Черный

На практике же, для сохранения информации о цвете каждой точки цветного изображения в модели RGB обычно отводится 3 байта (то есть 24 бита) - по 1 байту (то есть по 8 бит) под значение цвета каждой составляющей. Таким образом, каждая RGB-составляющая может принимать значение в диапазоне от 0 до 255 (всего 2^8=256 значений), а каждая точка изображения, при такой системе кодирования может быть окрашена в один из 16 777 216 цветов. Такой набор цветов принято называть True Color (правдивые цвета), потому что человеческий глаз все равно не в состоянии различить большего разнообразия.

Для того чтобы на экране монитора формировалось изображение, информация о каждой точке (код цвета точки) должна храниться в видеопамяти компьютера. Рассчитаем необходимый объем видеопамяти для одного из графических режимов. В современных компьютерах разрешение экрана обычно составляет 1280x1024 точек. Т.е. всего 1280 * 1024 = 1310720 точек. При глубине цвета 32 бита на точку необходимый объем видеопамяти: 32 * 1310720 = 41943040 бит = 5242880 байт = 5120 Кб = 5 Мб.

Растровые изображения очень чувствительны к масштабированию (увеличению или уменьшению). При уменьшении растрового изображения несколько соседних точек преобразуются в одну, поэтому теряется различимость мелких деталей изображения. При увеличении изображения увеличивается размер каждой точки и появляется ступенчатый эффект, который можно увидеть невооруженным глазом.

Графические форматы файлов

Форматы графических файлов определяют способ хранения информации в файле (растровый или векторный), а также форму хранения информации (используемый алгоритм сжатия).

Наиболее популярные растровые форматы:

BMP

GIF

JPEG

TIFF

PNG

Двоичное кодирование звука

Использование компьютера для обработки звука началось позднее, нежели чисел, текстов и графики.

Звук– волна с непрерывно изменяющейся амплитудой и частотой. Чем больше амплитуда, тем он громче для человека, чем больше частота, тем выше тон.

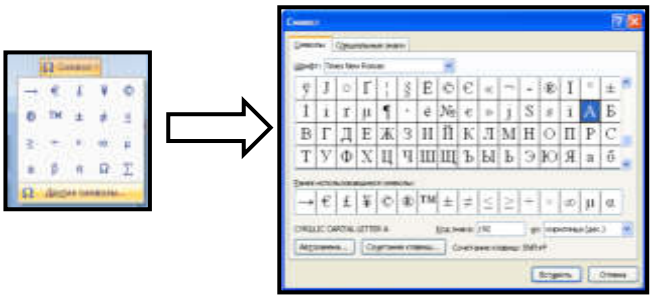
Частота дискретизации– количество измерений уровня сигнала в единицу времени.

Количество уровней громкости определяет глубину кодирования. Современные звуковые карты обеспечивают 16-битную глубину кодирования звука. При этом количество уровней громкости равно $N = 2^{16} = 65536$.

9. Содержание работы:

Вариант №__

Задание №1. Используя таблицу символов, записать последовательность десятичных числовых кодов в кодировке Windows для своих ФИО, названия улицы, по которой проживаете. Таблица символов отображается в редакторе MS Word с помощью команды: вкладка ***Вставка→Символ→Другие символы***



В поле ***Шрифт*** выбираете Times New Roman, в поле ***из*** выбираете кириллица. Например, для буквы «А» (русской заглавной) код знака– 192.

Пример:

И	**В**	**А**	**Н**	**О**	**В**		**А**	**Р**	**Т**	**Е**	**М**
200	**194**	**192**	**205**	**206**	**194**		**192**	**208**	**210**	**197**	**204**

П	**Е**	**Т**	**Р**	**О**	**В**	**И**	**Ч**
207	**197**	**210**	**208**	**206**	**194**	**200**	**215**

Выполнение задания №1

Задание №2. Используя стандартную программу ***БЛОКНОТ***, определить, какая фраза в кодировке Windows задана последовательностью числовых кодов и продолжить код. Запустить ***БЛОКНОТ***. С помощью дополнительной цифровой клавиатуры при нажатой клавише **ALT** ввести код, отпустить клавишу **ALT.** В документе появиться соответствующий символ.

Выполнение задания №2

0255		**0243**	**0247**	**0243**	**0241**	**0252**		**0226**		**0225**	**0232**	**0234**		**0239**	**0238**

0241	**0239**	**0229**	**0246**	**0232**	**0235**	**0224**	**0252**	**0237**	**0238**	**0241**	**0242**	**0232**

заполнить верхнюю строку названием специальности

Задание №3. Заполнить пропуски числами:

	Кбайт	=	байт	=	бит

1.

	Мбайт	=	Кбайт	=	байт

2.

	Гбайт	=	Мбайт	=	Кбайт	=	байт

Решения:

Задание №4. Перевести десятичное число в двоичную систему счисления и сделать проверку:

Задание №5. Записать в развернутой форме двоичное и восьмеричное число и, произведя вычисления, выразить в десятичной системе счисления:

Задание №10. Ответить на контрольные вопросы:

Что такое информация?	
Перечислите свойства информации.	
Какие виды информации Вы знаете?	
Приведите примеры аналогового представления графической информации.	
Что такое пиксель?	
Что такое система счисления?	
Напишите правило перевода десятичных чисел в двоичный код.	
Перечислите единицы измерения информации.	

5. Содержание отчета

Отчет должен содержать:

1. Название работы.
2. Цель работы.
3. Задание и его решение.
4. Вывод по работе

Практическая работа №6

Запись и хранение информации на различных носителях"

2 часа

Тема занятия: Запись ***информации на различные носители. Антивирусные средства защиты.***

Цель выполнения задания: научиться *записывать информацию на различные носители. Уметь использовать антивирусные средства защиты.*

Необходимо знать: *основные приёмы записи данных на носители. Знать популярные антивирусные средства защиты.*

Необходимо уметь: *совершать стандартные действия с документами, антивирусными средствами защиты.*

Оборудование (приборы, материалы, дидактическое обеспечение): *Инструкционные карты, мультимедийные презентации, персональные компьютеры.*

ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

Вредоносная программа — компьютерная программа или переносной код, предназначенный для реализации угроз информации, хранящейся в компьютерной системе, либо для скрытого нецелевого использования ресурсов системы, либо иного воздействия, препятствующего нормальному функционированию компьютерной системы. К вредоносному программному обеспечению относятся сетевые черви, классические файловые вирусы, троянские программы, хакерские утилиты и прочие программы, наносящие вред компьютеру, на котором они запускаются на выполнение, или другим компьютерам в сети.

Формы хранения и носители информации

Форма информации	Носитель информации	Метод записи информации
Механическая	пластинка	аналоговый
Оптическая	бумага	знаково-символьный
	кинофотоплёнка	аналоговый
	лазерный аудио дискCD-A	аналоговый
	лазерный диск CD-ROM, DVD	цифровой
Магнитная	аудиовидео плёнка	аналоговый
	гибкие диски	цифровой
	жёсткие диски	цифровой

Запись информации производится путем изменения параметров носителя.

Запись информации на носители в виде полей и электрического тока осуществляется путем изменения их параметров. Непрерывное изменение параметров сигналов в соответствии со значениями первичного сигнала называется **модуляцией,** дискретное — **манипуляцией.**

ХОД ВЫПОЛНЕНИЯ ЗАДАНИЯ, МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Задание № 1. ***В операционной системе Windows проверить выбранные объекты на наличие вредоносных объектов, выполнить лечение или удаление зараженных объектов***

1) Запустить на выполнение антивирусную программу.
2) Запустить обновление из контекстного меню.
3) Выполнить проверку съемного носителя.
4) Выполнить проверку локального диска.
5) Отчет о работе антивирусной содержит информацию о результатах проверки.

Задание № 2. ***Наберите предложенный текст:***

Компьютерный вирус — это целенаправленно созданная программа, автоматически приписывающая себя к другим программным продуктам, изменяющая или уничтожающая их.

Вредоносные программы можно разделить на три класса: *черви, вирусы и троянские программы*.

Черви — это класс вредоносных программ, использующих для распространения сетевые ресурсы. Используют сети, электронную почту и другие информационные каналы для заражения компьютеров.

Вирусы — это программы, которые заражают другие программы — добавляют в них свой код, чтобы получить управление при запуске зараженных файлов.

Троянские программы — программы, которые выполняют на поражаемых компьютерах несанкционированные пользователем действия, т.е. в зависимости от каких-либо условий уничтожают информацию на дисках, приводят систему к зависанию, воруют конфиденциальную информацию и т.д.

В зависимости ***от среды обитания*** вирусы можно разделить на *сетевые, файловые, загрузочные и файлово-загрузочные.*

Сетевые вирусы распространяются по различным компьютерным сетям.

Файловые вирусы внедряются главным образом в исполняемые модули, т.е. в файлы, имеющие расширения COM и EXE.

Загрузочные вирусы внедряются в загрузочный сектор диска или сектор, содержащий программу загрузки системного диска.

Файлово-загрузочные вирусы заражают файлы и загрузочные сектора дисков.

По способу заражения вирусы разделяются *на резидентные и нерезидентные*.

Резидентный вирус при заражении компьютера оставляет в оперативной памяти свою резидентную часть, которая потом перехватывает обращение операционной системы к объектам заражения (файлам, загрузочным секторам дисков и т.д.) и внедряется в них.

Нерезидентные вирусы не заражают память компьютера и являются активными ограниченное время.

По степени воздействия выделяют **неопасные** вирусы, которые не мешают работе компьютера, **опасные**, которые могут привести к различным нарушениям в работе компьютера, и**очень опасные**, воздействие которых может привести к потере программ, уничтожению данных, стиранию информации в системных областях диска.

Для обнаружения, удаления и защиты от компьютерных вирусов разработано несколько видов специальных программ, которые позволяют обнаруживать и уничтожать вирусы. Такие программы называются **антивирусными**.

РАЗЛИЧАЮТ СЛЕДУЮЩИЕ ВИДЫ АНТИВИРУСНЫХ ПРОГРАММ:

Программы-детекторы осуществляют поиск характерной для конкретного вируса сигнатуры в оперативной памяти и файлах и при обнаружении выдают соответствующие сообщение. Недостатком таких антивирусных программ является то, что они могут находить только те вирусы, которые известны разработчикам таких программ.

Программы-доктора или **флаги** не только находят зараженные вирусами файлы, но и возвращают файлы в исходное состояние. В начале своей работы флаги ищут вирусы в оперативной памяти, уничтожая их, и только затем переходят к «лечению» файлов.

Программы-ревизоры запоминают исходное состояние программ, каталогов и системных областей диска тогда, когда компьютер не заражен вирусом, а затем периодически или по желанию пользователя сравнивают текущее состояние с исходным. Обнаружение изменения выводится на экран монитора.

Программы-фильтры или **сторожа**, представляют собой небольшие резидентные программы, предназначенные для обнаружения подозрительных действий при работе компьютера, характерных для вирусов:

- попытка коррекции файлов с расширениями COM и EXE;
- изменение атрибутов файла;
- прямая запись на диск по абсолютному адресу;
- запись в загрузочные сектора диска;
- загрузка резидентной программы.

При попытке вирусной атаки сторож посылает сообщение и предлагает запретить или разрешить соответствующие действия.

Программы - вакцины или **иммунизаторы** — это резидентные программы, предотвращающие заражение файлов.

Признаки заражения компьютера вирусом. Существует ряд признаков, свидетельствующих о заражении компьютера:

- ✓ вывод на экран непредусмотренных сообщений или изображений;
- ✓ подача непредусмотренных звуковых сигналов;
- ✓ неожиданное открытие и закрытие лотка CD-ROM-устройства;
- ✓ произвольный, без вашего участия, запуск на компьютере каких-либо программ;
- ✓ вывод на экран предупреждения о попытке какой-либо из программ вашего компьютера выйти в Интернет, хотя вы никак не инициировали такое ее поведение (при наличии установленной на вашем компьютере соответствующей антивирусной программы).

Сохраните изменения в документе. Для этого выберите команду меню ***Файл—> Сохранить.***

Контрольные вопросы:

1. Что такое компьютерный вирус? Какими свойствами обладают компьютерные вирусы?
2. Какие угрозы информации способны нанести вредоносные программы?
3. Для чего предназначены антивирусные программы?
4. Каковы функции брандмауэра?
5. В чем разница между антивирусными сканерами и мониторами?
6. Какие существуют признаки заражения компьютерным вирусом?
7. Что необходимо сделать в первую очередь в случае заражения компьютерным вирусом?

5. Содержание отчета

Отчет должен содержать:

1. Название работы.
2. Цель работы.
3. Задание и его решение.
4. Вывод по работе

Практическая работа №7 (2 часа)

"Работа с файловыми системами"

Тема занятия: Работа ***с файлами и папками. Размещение, сохранение, поиск информации в ПК.***

Цель выполнения задания: 1. Научиться использовать окно проводника. 2. Научиться копировать, перемещать Файлы с помощью программы "Проводник". 3. Научиться создавать папки с помощью программы "Проводник" и работать с ними.

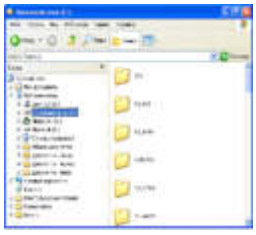
Научиться использовать программу «Проводник» для поиска файлов и папок по имени, дате создания и по содержимому.

Необходимо знать: основные приёмы создания, удаление, переименования папок и файлов используя основное и контекстное меню.

Необходимо уметь: совершать стандартные действия над Папками и Файлами с помощью команд Основного меню; пользоваться кнопками Панели инструментов.

Оборудование (приборы, материалы, дидактическое обеспечение) Инструкционные карты, мультимедийные презентации, персональные компьютеры.

ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

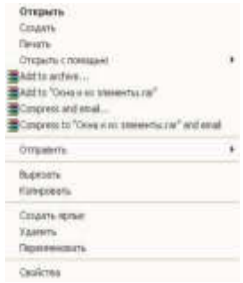


В состав операционной системы Windows входит программа Проводник. Проводник предназначен для управления файловой системой Windows и обеспечивает доступ к локальным и сетевым устройствам, в том числе к периферийным устройствам. Проводник отображает содержимое папок, позволяет открывать, копировать, перемещать, удалять, переименовывать папки и файлы, запускать программы, выводит на экран изображение дерева папок. Окна папок, с которыми вы работаете на рабочем столе, являются частью полного окна Проводника.

По умолчанию окно Проводника включает две панели. Левая панель показывает все ресурсы, представленные в виде иерархического дерева. Эта панель включает все объекты, в том числе папки **Мой компьютер** и **Сетевое окружение**. Если на вашем компьютере не установлено сетевое программное обеспечение, то папка **Сетевое** окружение не появляется.

В правой области (панели содержимого) на экран выводится содержимое выбранной папки.

Для просмотра содержимого папки щелкните на ее значке или на описании в левой панели. В правой панели появится ее содержимое. То, в каком виде объект появляется на панели содержимого, зависит от выбранного способа просмотра. Можно выводить объекты на экран в виде крупных значков, мелких значков, списка или таблицы, выбрав одну из четырех опций меню ***Вид*** или щелкнув на соответствующей кнопке на панели инструментов. Если выбранная папка содержит в себе другие папки, то их значки появляются в панели содержимого среди других объектов, даже если в дереве папка не развернута.



Изучите справочную информацию Windows о поиске файлов и папок, для чего откройте меню **Пуск**, выберите в нем команду **Справка и поддержка**. В окне ***Центр справки и поддержки*** щелкните кнопку **Указатель** в панели инструментов, затем введите образ поиска «поиск файлов и папок». Выберите в списке найденных разделов «**Использование команды Найти в меню Пуск**» и щелкните кнопку **Показать**. Изучите справочную информацию в правой части окна. Закройте окно ***Центр справки и поддержки***.

Для поиска файлов откройте меню **Пуск**, выберите в нем команду **Поиск**, а затем выберите команду **Файлы и папки**. На панели «Что вы хотите найти» выберите ссылку **Файлы и папки**. Введите часть имени или полное имя файла или папки или введите слово или фразу, содержащиеся в этом файле. В поле ***Поиск в***ыберите диск, папку или сетевой ресурс, в котором требуется выполнить поиск, например, диск D: как показано на рис. 3. Щелкнув кнопку **Найти**, запустите процедуру поиска.

ХОД ВЫПОЛНЕНИЯ ЗАДАНИЯ, МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Задание № 1. Определите тип, размер свободного пространства накопителей, установленных на компьютере. Просмотрите, какие папки и Файлы хранятся на дисках.

Методика выполнения работы

1. *Определите, какие накопители установлены на вашем компьютере. Для этого:*

• откройте **Мой компьютер**, дважды щелкнув мышью на значке: по значку и букве, сопровождающей значок, можно судить о типе накопителя;

2. *Определите величину свободного пространства на диске:*

• запустите программу **Проводник**, выбрав в **Главном меню** пункт **Программы** и в подменю пункт **Проводник**;

• вызовите контекстное меню, щелкнув правой кнопкой мыши на значке накопителя;

• выберите пункт меню **Свойства**;

• просмотрите содержание вкладки **Общие** окна **Свойства**;

• закройте окно.

Задание № 2. ***Откройте Мой компьютер и создайте на диске E: с помощью Основного меню папку Новая, в папке Новая — папку Моя и папку Общая***

Методика выполнения работы

1. Создайте на диске **E** в своей папке - папку **Новая**:

• откройте **Мой компьютер** и просмотрите содержимое диска D:

• откройте меню **Файл**, выберите параметр. **Создать**;

• в раскрывшемся подменю выберите пункт **Папка**, таким образом, вы выполните команду **Файл – Создать - Папка**;

• введите в поле ввода имени рядом со значком созданной папки «**Новая**» (вновь созданная папка появляется с именем **Новая** папка);

• нажмите клавишу **Enter**.

2. Создайте папки **Моя** и **Общая** в папке **Новая**:

- откройте папку **Новая**, дважды щелкнув на значке папки;

- создайте папку **Моя** и папку **Общая**, используя технологию работы, описанную в п.1 этого задания.

Задание № 3. Выполните стандартный набор действий над папками и Файлом, используя команды меню Правка программы Проводник: вырезать, копировать, Вставить.

Методика выполнения работы

1. Запустите программу **Проводник** и перейдите в папку **Новая**.

2. Переместите папку **Моя** в папку **Общая**:

- откройте папку **Новая**; выделите папку **Моя**;

- выполните команду **Правка - Вырезать**, папка **Моя** будет перемещена в буфер обмена; откройте папку **Общая**;

- выполните команду **Правка - Вставить**; папка **Моя** будет вставлена из буфера обмена в папку **Общая**.

3. Скопируйте папку **Моя** из папки **Общая** в папку **Новая**:

- откройте папку **Общая**; выделите папку **Моя**;

- выполните команду **Правка - Копировать** — папка **Моя** будет перемещена в буфер обмена; откройте папку **Новая**;

- выполните команду **Правка - Вставить** — папка **Моя** будет вставлена из буфера обмена в папку **Новая**.

4. Удалите **Файл** из папки **Новая**, папку **Моя** из папки **Общая**:

- выделите **Файл**;

- выполните команду **Файл - Удалить**;

Задание № 4. Восстановите удаленные папки.

Методика выполнения работы

Восстановите удаленные папки **Моя** и **Общая**:

- откройте **Корзину**, дважды щелкнув на ее значке;

- в окне Корзины выделите папку **Моя**;

- выполните команду **Файл - Восстановить**;

Очистите корзину:

- откройте **Корзину**, дважды щелкнув на ее значке;

- выполните команду **Файл** - Очистить **Корзину**.

НАБЕРИТЕ ПРЕДЛОЖЕННЫЙ ТЕКСТ

NORTHERN IRELAND

Northern Ireland occupies the north - eastern part of Ireland, which is separated from the island of Great Britain by the North Channel. In the south-west Northern Ireland borders on the Irish Republic (Eire).

Almost all the area of Northern Ireland is a plain of Volcanic origin, deepening in the centre to form the largest lake of the British Isles, Lough Neagh.

The greatly indented coastline of Northen Ireland is abundant in rocks and cliffs.

Northern Ireland has a typical oceanic climate with mild damp winters and cool rainy summers. Forests are rather scarce, moors and meadows prevail.

Northern Ireland is mostly an agrarian district. On small farms they grow crops, especially oats, Vegetables and potatoes. Large areas are taken up by meadows, where cattle graze. On the river banks and on the coasts the population is engaged in fishing.

Сохраните документ. Для этого выберите команду меню ***Файл—> Сохранить.***

Задание № 5. Запустите информационно-поисковую систему.

Методика выполнения работы

1. Первый способ: в главном меню откройте **Файл,** выберите **Найти.** Появится окошко **Обычный поиск.** Ниже увидите **Расширенный поиск**, **найти в документе**.

2. Второй способ: нажмите кнопку **Пуск→Найти→Файлы и Папки.** Откроется окошко **Результаты поиска.**

Контрольные вопросы*:*

8. Как можно запустить программу "Проводник"?
9. Что следует сделать, чтобы развернуть (свернуть) структуру папки?
10. Как можно перейти в папку более высокого уровня?
11. Как создать текстовый файл?
12. Как создать графический файл?
13. Как сохранить файл, закрыть файл, открыть файл?
14. Как найти файл, не зная его полного имени?
15. Как переименовать папку с помощью мыши?
16. Как переместить файл с помощью панели инструментов?
17. Как скопировать папку с помощью контекстного меню?

5. Содержание отчета

Отчет должен содержать:

1. Название работы.
2. Цель работы.
3. Задание и его решение.
4. Вывод по работе

Практическая работа №8

Монтаж видео в Adobe Premiere CC"(4 часа)

Цель работы: знакомство с основами видеомонтажа в программе Adobe Premiere Pro, с принципами создания специальных эффектов, а также освоение интерфейса программы.

Задание:

1. Создайте новый проект. Нажмите кнопку New Project в диалоговом окне с приглашением. Введите в текстовое поле Name имя файла проекта. Затем задайте в поле Location путь к папке, в которую будет сохранен проект. В завершение следует нажать в диалоговом окне New Project кнопку OK.

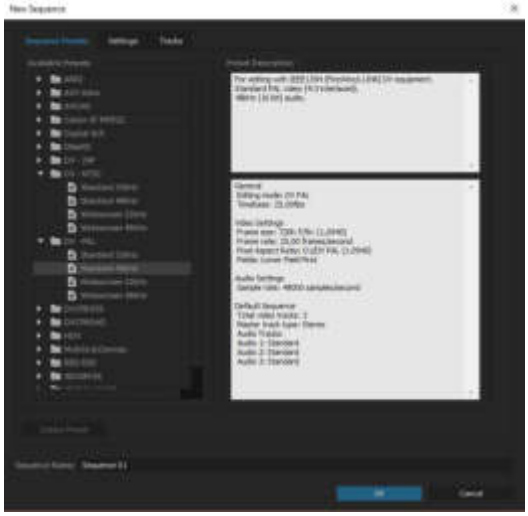
2. Определите установки проекта в диалоговом окне New Sequence.

Установки *— это основные параметры, определяющие настройки панелей, в которых будет происходить редактирование проекта (размер изображения фильма, частота кадров, способ нумерации кадров и т. п.).*

Будем редактировать проект в формате DV (Digital Vimeo — цифровое видео). Для этого достаточно выбрать в левом списке Available Presets (Доступные предустановки) набор установок, составленный разработчиками Premiere.

Например, пункт DV - PAL -Standart 48 kHz (Цифровое видео - стандарт PAL - Стандартный звук 48 кГц) будет означать выбор и качестве среды редактирования фильма формат PAL (европейский стандарт телевидения).

После того как вы выберете одну из предустановок, параметры, составляющие ее набор, отобразятся на панелях Description (Описание) справа.



Отметим остальные форматы:

и *AVCHD — установки для видео, снятого камерой в формате высокой четкости (HD) и использующей кодек MPEG-4 AVC/H.264. Формат AVCHD предназначается для записи видео высокой четкости в режимах 720p, 1080i 1080p;*

• *DV-NTSC — установки для телевещания (NTSC — североаме риканский стандарт телевещания), размер кадра (в пикселях) 720×480;*

DV-PAL — установки для телевещания (PAL — европейский стандарт телевещания), размер кадра 720×576;

HDV (видео высокого разрешения) — более современные форматы, отличающиеся наилучшим качеством изображения, размеры кадра 1280×720

и *больше (например, 1920×1080).*

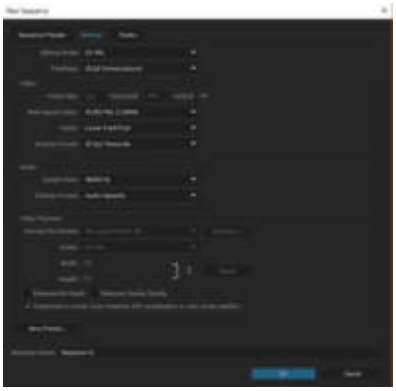
Выбрав предустановку, можно затем изменить некоторые ее параметры вручную на вкладке Settings. С помощью пункта Timebase можно установить определенное значение количества кадров в секунду.

В *блоке Video на данной вкладке можно изменить настройки видео: Frame Size (размер кадра), Pixel Aspect Ratio (соотношение сторон пикселя), Fields (расположение полей), Display Format (формат отображения).*

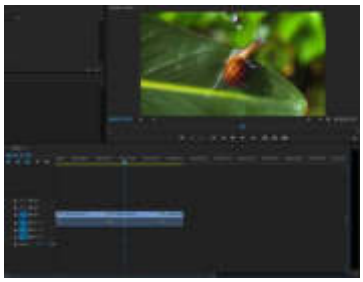
В *блоке Audio задаются параметры Sample Rate (частота дискретизации) и Display Format (формат отображения).*

В *следующем блоке Video Previews можно задать установки предварительного просмотра.*

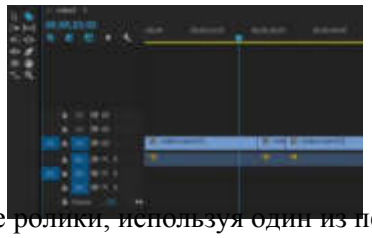
На вкладке Tracks определяется начальное число треков разного типа в проекте.



3. Импортируйте в проект медиафайлы (File-Import). Разместите их на Временной ленте Timeline.



При необходимости разрезания клипа используйте Razor Tool



4. Соедините ролики, используя один из переходов Video transitions (Effects.- Video transitions).

Работа с эффектами в программе осуществляется в двух специальных окнах — Effects и Effect Controls.

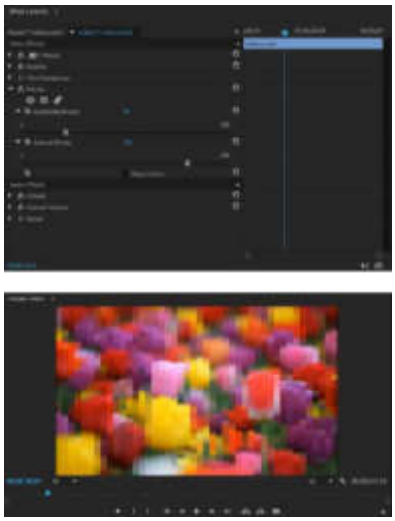
Списки всех эффектов распределены по папкам в зависимости от того, к какой группе они относятся: Presets (предустановки), Audio Effects (звуковые

эффекты), Audio Transitions (звуковые переходы), Video Effects (видеоэффекты), Video Transitions (видеопереходы).

В *окне Effect Controls отображаются дополнительные параметры эффекта.*
Суть эффекта перехода состоит в замещении особым образом видеоизображения или звука одного клипа изображением или звуком другого клипа.

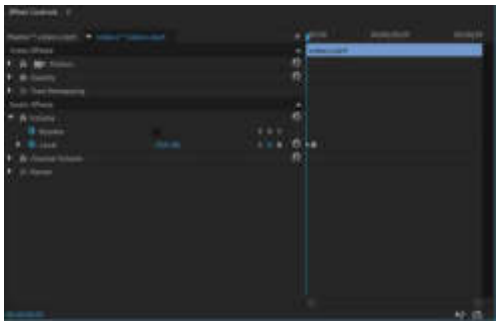
5. Добавьте один из эффектов Video Effects, выставив дополнительные параметры эффекта в окне Effect Controls.

Например, при добавлении эффекта мозаики (Mosaic) в настройках эффекта для изменения размера блоков мозаики по горизонтали передвигайте ползунок по верхней оси (Horizontal Blocks) до получения нужного размера. Аналогичным образом можно изменить размер блоков мозаики по вертикали, передвигая ползунок на следующей оси в разделе Vertical Blocks.

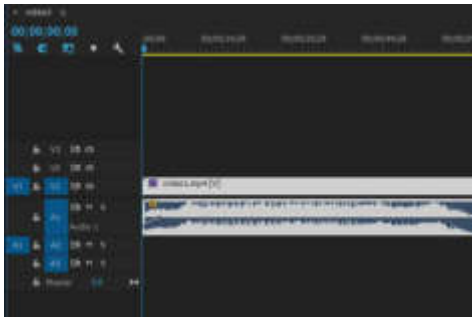


Сделайте плавный переход звука от тихого к нормальному в начале, и от нормального к тихому в конце (Audio Effects – Volume).

Для изменения громкости звука необходимо выставить нужный уровень громкости в dB в окне Effect Controls, на необходимых промежутках времени.



Для удобства можно развернуть дорожку аудио и менять уровни уже на ней.



7. Отключите и включите примененный эффект Video Effects, используя кнопку Toggle the Effect On or Off слева от названия данного эффекта.

При включенном эффекте кнопка находится в нажатом состоянии. Временное отключение эффекта не влияет на параметры настройки этого эффекта.

Чтобы удалить абсолютно все эффекты сразу, кликните правой кнопкой мыши по названию окна Effect Controls и выберите команду Remove Effects. В открывшемся окне поставьте галочки напротив всех удаляемых эффектов

и *нажмите на кнопку ОК.*

8. Экспортируйте видео в формате .avi (File—Export).

Контрольные вопросы

В чем отличие стандартов PAL и NTSC?
Какие существуют форматы видеофайлов?
Как осуществить подрезку краев клипа?
В чем суть эффекта перехода?
Как в Adobe Premiere PRO CC создаются переходы между клипами?
Где можно установить размер кадра?
Как удалить все эффекты сразу?
Как в Adobe Premiere PRO CC полностью убрать звук видео?

5. Содержание отчета

Отчет должен содержать:

1. Название работы.
2. Цель работы.
3. Задание и его решение.

Вывод по работе

Практическая работа №9

Захват видео с рабочего стола ПК с помощью iSpring Free Cam 8» (2 часа)

Цель: повышение профессиональной компетентности педагогических работников в области дистанционных образовательных технологий.

Знакомимся с минимальной по функционалу программой для записи видео с экрана – iSpring Free Cam, с помощью которой можно сделать или отредактировать видеозаписи. Итак, рассмотрим, как сделать видео с экрана с помощью программы iSpring Free Cam. Прежде чем приступим к теме, рассмотрим что предлагает данная программа.

Достоинства программы:

- Бесплатная для скачивания
- Встроенный редактор
- Нет ограничений по длительности записи
- Прекрасно подходит для записи обучающих видеороликов, создания презентаций и видеоинструкций.

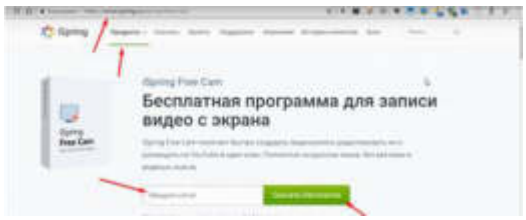
Недостатки:

- Не самое лучшее качество записи
- Скромные возможности встроенного редактора
- Экспорт в единственном формате видео - WMV

iSpring Free Cam – это программа, возможности которой позволяют сделать записи всей области экрана или определенного фрагмента. У данной программы есть функция простого редактирования аудио – извлечение шумовых отрезков и применение эффекта затухания. Интерфейс программы на русском языке и имеет полный набор функций для записи, редактирования с экрана компьютера, ноутбука и публикации видео.

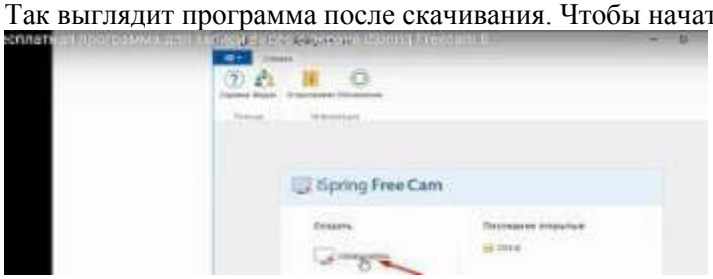
Скачать данную программу можно на сайте www.ispring.ru.

Заходите на главной странице сайта в раздел «Продукты». Далее, вводите email и скачиваете программу.



Установщик появляется на рабочем столе – это ярлык, нажав на который запускаете программу.

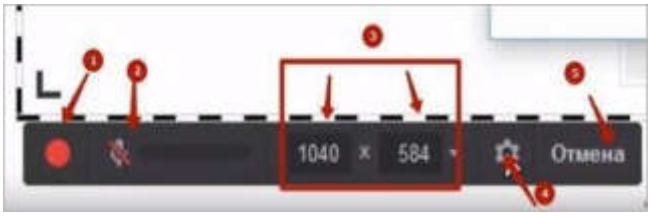
Так выглядит программа после скачивания. Чтобы начать новую запись, нажимаем кнопку «Новая запись».



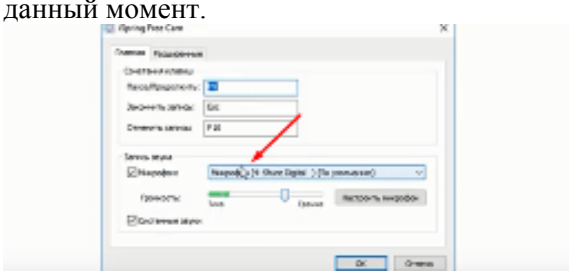
Открывается окно. Зажав кнопку мыши, выбираем необходимый размер области – экран для записи (размер окна), который выделен пунктиром на экране.

Внизу находится «Шкала микрофона». Если микрофон включен, то шкала будет постоянно двигаться. Очень часто в системе есть несколько микрофонов, и Вы будете говорить в один, а активным может оказаться совершенно другой микрофон. Внизу на панели расположена:

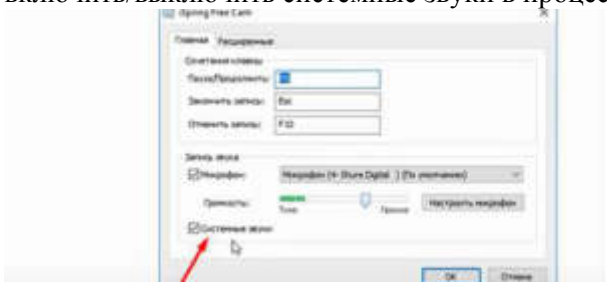
- красная кнопка – это для начала записи (№1)
- кнопка включения/выключения микрофона (№2)
- можно с помощью данной опции отрегулировать размер окна, который выделен пунктиром (№3)
- и раздел «Настройки» – иконка в виде “Шестеренки”, с помощью которой выполняются предварительные настройки (№4).



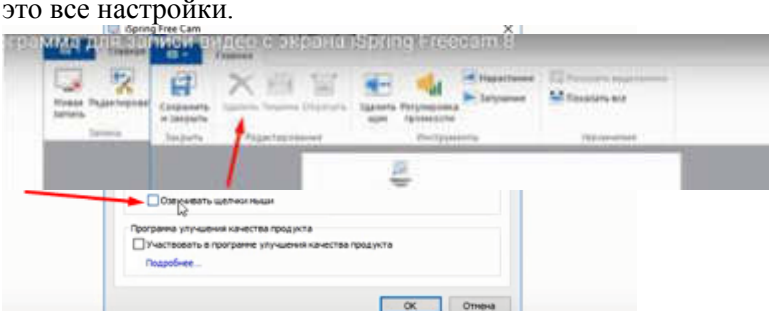
Чтобы проверить, тот ли микрофон включен, заходим в настройки и проверяем, какой микрофон работает в данный момент.



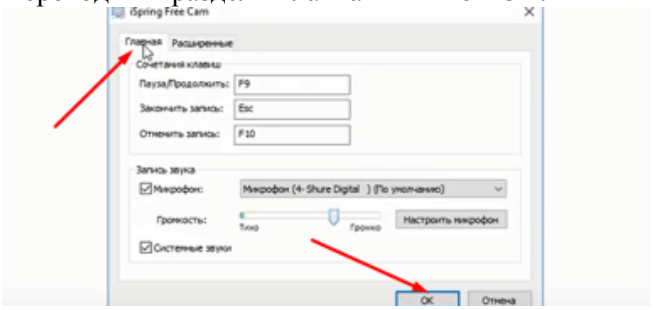
Настраиваем запись экрана программы Перед записью необходимо проверить все настройки. Например, можно включить/выключить системные звуки в процессе записи видео.



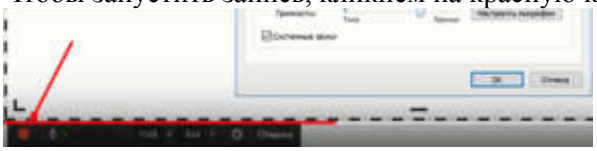
В разделе «Расширенные» настройки запись Курсора можно отключить. выделить или озвучивать его щелчки . Все, это все настройки.



Переходим в раздел «Главная» и жмем ОК.



Чтобы запустить запись, кликнем на красную кнопку.



Через три секунды начнется запись. После окончания видеозаписи, нажимаем кнопку «Готово».

Откроется интерфейс, где можно прослушать то, что записано.

Здесь же можно отредактировать запись, нажав на файл «Редактировать».

Откроется редактор записи, где вносятся все изменения.

что-то

ажимаете

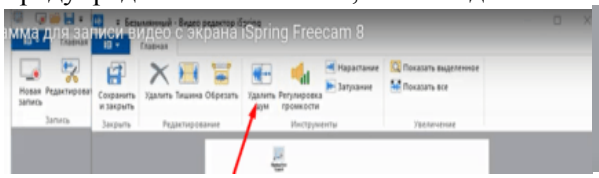
«Удалить».

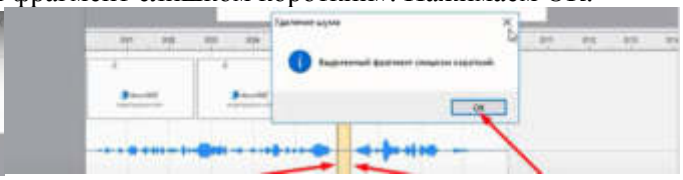
Чтобы обрезать запись, выделяете тот фрагмент, который хотите оставить. Нажимаете на редактор «Обрезать» и все лишнее удаляется, а остается только то, что выделено.

Можно еще настроить нарастание и затухание звука. Допустим, мы хотим, чтобы звук плавно нарастал. Выделяем фрагмент

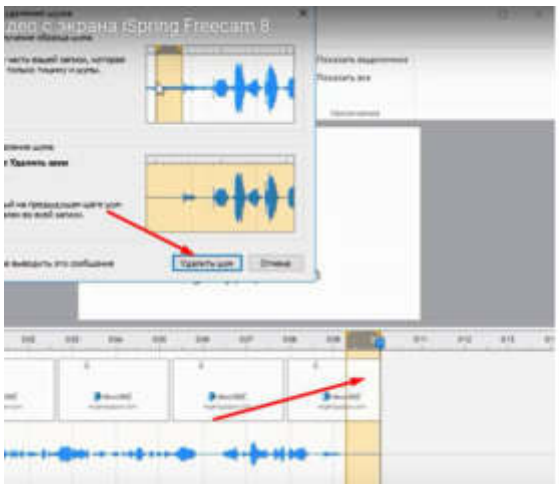
и кликнем по пиктограмме «Нарастание». Звук постепенно будет нарастать. Точно также можно сделать и с затуханием звука. Выделяем фрагмент записи и нажимаем «Затухание». Звук постепенно затухает.

Еще можно удалить шум. Выделяем фрагмент, где только шум. Нажимаем «Удалить шум». Откроется предупредительное окошко, что «Выделенный фрагмент слишком короткий». Нажимаем ОК.

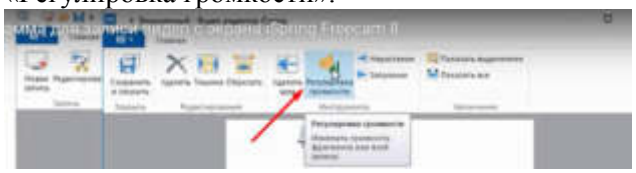




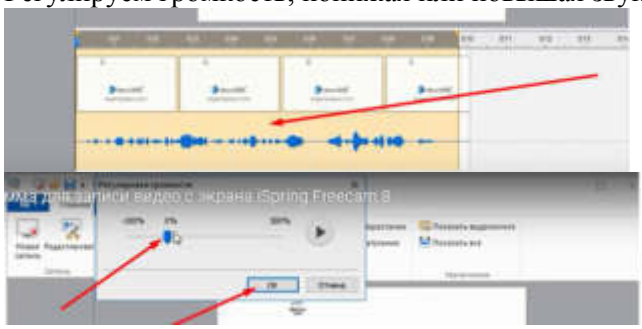
Выделяем необходимый фрагмент для удаления уже побольше и нажимаем «Удалить шум».



Можно отрегулировать громкость. Чтобы отрегулировать громкость, выделяем всю область, высвечивается окошко «Регулировка громкости».

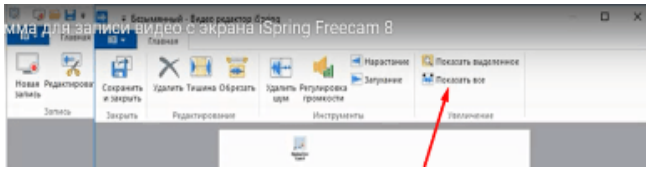


Регулируем громкость, понижая или повышая звук записи.

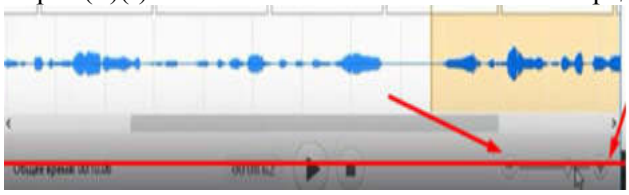


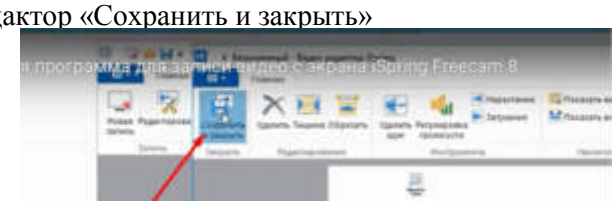
Можно воспользоваться еще такой опцией. Выделили необходимый фрагмент и жмем «Показать выделенное». Программа увеличит данный фрагмент до максимально возможного. Тогда работать будет проще.

Вернуться в обычный режим – это опция «Показать все», нажав на которую будет видно всю шкалу.



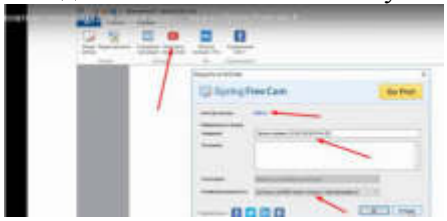
Через (+)(-) меняете масштаб. И нажимаете на редактор «Сохранить и закрыть»





После окончания записи можно сохранить видео или сразу же загрузить на YouTube канал. Для сохранения видеозаписи нажимаем «Сохранить как видео».

Заходите в свой YouTube аккаунт. Выбираете нужную категорию и загружаете.



Как видите программа простая. Функциональные инструменты данной программы позволят записать видео с экрана без проблем! Быстро его отредактировать и загрузить на YouTube канал. Мы разобрали, как сделать видео с экрана, с помощью iSpringFreeCam, научились скачивать и запускать данную программу на своем компьютере.

5. Содержание отчета

Отчет должен содержать:
1.Название работы.
2.Цель работы.
3.Задание и его решение.
4.Вывод по работе

Практическая работа №10

Обработка звуковой информации в Adobe Audition CC". (2 часа)

Цель: дать практические навыки создания и редактирования цифрового звука с помощью компьютера и специальных программ – аудио редакторов.

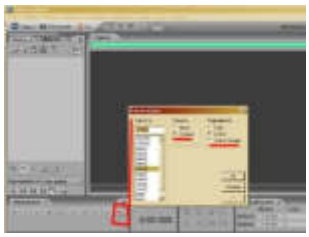
Задание №1 Записать в программу Adobe Audition несколько слов с помощью микрофона, подключенного к компьютеру (или с помощью встроенного микрофона в аудиосистему компьютера)

Ход выполнения

Подключаем микрофон в заднюю панель компьютера в розовый вход.

После чего запускаем программу для обработки звуковой информации **Adobe Audition.**

В отрытом окне программы находим кнопку **«Запись»** и нажимаем. Появится небольшое окошко, где нужно будет определить параметры записываемого звука. Пуска настройки останутся по умолчанию – так как они есть.



Нажав ОК в окне правки звука «побежит» бегунок, оставляя после себя вот такие колебания – так будет выглядеть записанный голос в цифровом виде:

Если нужно прекратить запись, то жмем на кнопку стопа (черный квадратик).

После окончания записи, сохраняем записанную звуковую дорожку **Файл-Сохранить как…**

Задание №2

- Редактирование аудиозаписи: вырезаем определенную часть звуковой дорожки.

Ход выполнения

Итак, для того чтобы удалить или вырезать ненужную часть из всего аудиофайла нужно придерживаться нескольких советов:

- для прослушивания звукового файла используйте кнопкой» **Воспроизвести»**
- максимально часто пользуйтесь инструментом **Масштаб** – именно он позволяет увеличить дорожку звука до мельчайших подробностей для точного удаления ненужной «волны».

Следующим шагом будет выделение ненужного фрагмента.

Наводим курсор мышки на то место, где обнаружен ненужный отрезок, нажимаем левую кнопку мыши и проводим курсором к тому месту, где он заканчивается.

Для **удаления фрагмента** есть три способа:

- через **«Правка-Вырезать»**
- с помощью сочетания клавиш **«Ctrl+X»**
- клавиша **«Delete»**

В первых двух случаях отрезок скопируется в буфер обмена (его потом можно будит вставить в этот или другой звуковой файл).

В последнем случае отрезок полностью удаляется.

5. Содержание отчета

Отчет должен содержать:
1.Название работы.
2.Цель работы.
3.Задание и его решение.
4. Вывод по работе

Практическая работа №11

"Создание музыки в FL Studio" (2 часа)

Цель работы: рассмотреть возможности создания музыки в программе FL Studio и обучить пользователей созданию музыки в данной программе на подробном пошаговом и иллюстрированном примере

Актуальность работы: я считаю, что данная работа актуальна, потому что практически в каждом доме есть компьютер, и практически каждый человек любит музыку. Так же есть люди, которые хотят научиться её создавать, но не имеют соответствующих инструментов и студии для записи.

FL Studio (ранее Fruity Loops) — популярный редактор-секвенсор для написания музыки, созданный программистом Дидье Дэмбреном (также известным под псевдонимом «gol»), который разрабатывал эту программу восемь лет, и выпускаемый компанией Image-Line Software. Музыка создаётся путём записи и сведения аудио- или MIDI-материала. Готовая композиция может быть записана в формате WAV, MP3 или OGG.

Step Sequencer — позволяет быстро создавать и редактировать петли, добавлять новые генераторы (каналы) и удалять ненужные;

Piano Roll — представляет собой двумерную сетку, по вертикальной оси которой отложен уровень высоты звука, по горизонтальной — время, имеет больше возможностей, чем step sequenser;

Playlist — позволяет размещать созданные в Step Sequenser или в Piano Roll петли, или располагать звуковые файлы;

Mixer — здесь размещаются плагины эффектов;

Sample Browser — лёгкий доступ к аудиофайлам, плагинам и настройкам.

Принцип работы программы

FL Studio является дорожечным (паттерн) секвенсором, где создание музыки происходит в *Piano Roll, Step Sequencer* и затем осуществляется сборка в окне *Playlist*. Имеется большой набор уже готовых инструментов и множество эффектов, которые могут быть задействованы в режиме реального времени.

Главная составляющая проекта композиции — генератор (канал).

Генератор синтезирует или воспроизводит звук. Генераторов в проекте композиции может быть неограниченное количество. Каждый генератор обладает своими настройками, уникальным звуком, имитирующим любой инструмент. Для генераторов программируются нотные партитуры, записываемые в *Piano Roll*. Партитуры в FL Studio имеют бесконечную длину. Кусочки партитур (паттерны) складываются в последовательности (располагаются в нужном порядке в списке воспроизведения) в окне *Playlist*. Звук каждого генератора может быть обработан при помощи множества подключаемых эффектов.

В качестве генератора можно подключить любой VST или DXi плагин.

Наиболее известные музыканты, работающие в FL Studio

- Soulja boy
- Sector-D
- 45C.K.
- Criptic Windsome
- Jermz
- Tech N9ne
- Afrojack
- MC Basskill
- Ronald Jenkees

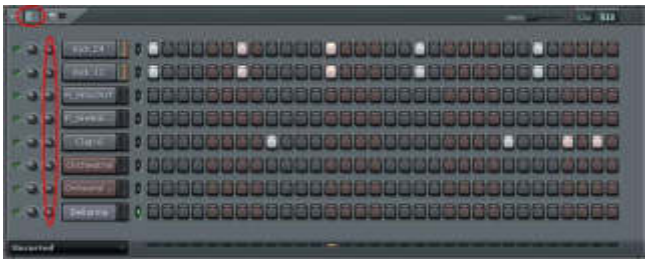
В качестве примера я решил выбрать создание минусовки (аранжировки) в стиле «West Coast Rap». Не сказать, что это отдельный стиль в музыке жанра Rap, но можно с легкостью заявить, что «West Coast Rap» имеет отличие от того же, скажем, Crunk. В качестве представителя данного "стиля" я взял выпустившего недавно новый альбом исполнителя The Game.

Темп/Ритм

Для данного жанра нужно выбрать подходящий темп, я взял «128», так как музыка данного стиля не медленна и не быстра, нечто среднее. А ритм примерно в 1,5 раза медленнее темпа.

Первое, с чего начинается моя минусовка - это бит (ударные). Чтобы создать его, нужно нажать на View step sequencer (или нажать F6), перетащить туда звуки из стандартного набора и расположить в нужном порядке, чтобы создать мелодию.

Звуки я взял такие: kick(удар) и clap(хлопок) и расположил их в таком порядке:



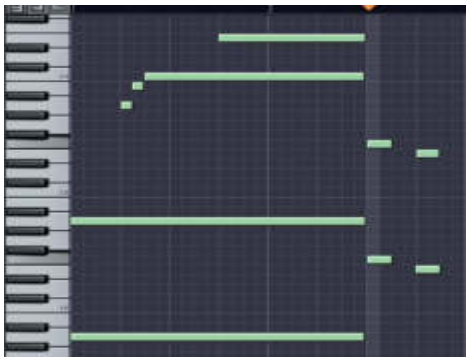
(Получившийся звук в приложении)

Также я взял хэты, чтобы разнообразить свой бит, и расположил их таким образом:

(получившийся звук в приложении)

Написание главной мелодии

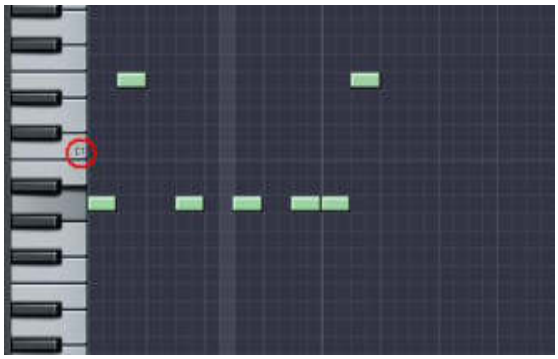
Главным инструментом я выбрал "Concert Piano" из плагина Edirol Orchestral. Чтобы добавить данный плагин нужно нажать: Channels - Add one – Orchestral. Если данного плагина не имеется, то вполне можно использовать плагин "FL KEYS" (для того, чтобы добавить плагин, нажмите Channels - Add One - FL KEYS). Открываем «Piano roll» (нотный редактор) для того, чтобы написать мелодию. Я выбрал низкую октаву.



(Получившаяся мелодия в приложении)

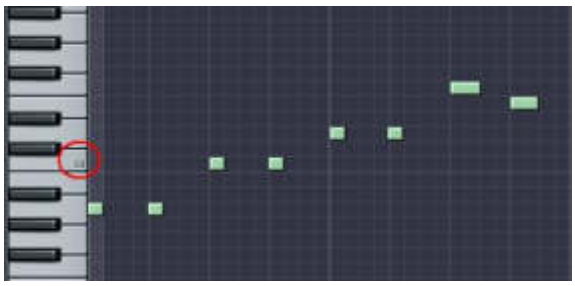
Получившаяся мелодия, по моему мнению, очень подходит для стиля **West Coast**.**Написание второстепенных мелодий**

1) Для первого второстепенного звука я выбрал тот же инструмент (Concert piano) и написал такую мелодию, в средней по звучанию октаве:



(полученная мелодия в приложении)

2) Разбавить сплошное пианино я решил звуком колокольчиков, а именно "Bellarina" из встроенного в саму программу плагина «Sytrus». Для добавления плагина нажмите Channels – add one – Sytrus. Открываем Piano Roll и пишем мелодию, очень похожую на самую первую:



(полученная мелодия в приложении)

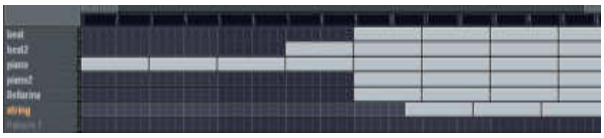
3) Теперь подумаем о припеве. Очень часто применяемый ход - это подчеркивание окончания фразы мелодии. Подчеркнем нашу фразу скрипками «Slow Strings» из плагина «Edirol Orchestral» (напоминаю, что все мелодии пишутся в Piano Roll):

(полученная мелодия в приложении)

Получился очень красивый протяжный звук, очень подходит для припева.

Завершающая стадия

Всё, написание мелодии завершено, остаётся лишь свести все наши звуки в единое целое. Чтобы сделать это, нужно нажать *View Playlist*, где и расположены наши звуки.



(Готовая аранжировка в приложении) Заключение

3. Данная работа – это следующий этап в исследовании мною программ по компьютерным технологиям. Работа по программе Adobe Photoshop была удостоена внимания в 2009-2010 учебном году.

5. Содержание отчета

Отчет должен содержать:

1.Название работы.

2.Цель работы.

3.Задание и его решение.

4. Вывод по работе

Практическая работа №12

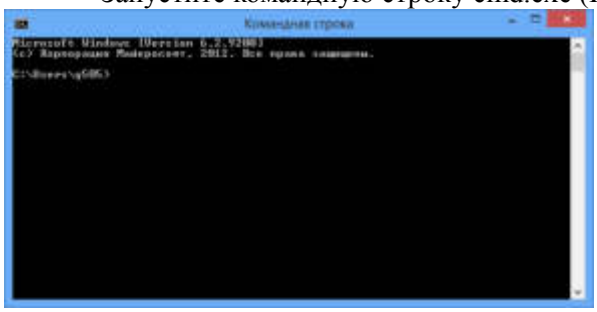
"Выполнение операций с помощью командной строки" (2 часа)

ЗАДАНИЕ. Изучить встроенные утилиты операционной системы Microsoft Windows для работы с файловой системой и диагностики сетевых подключений.

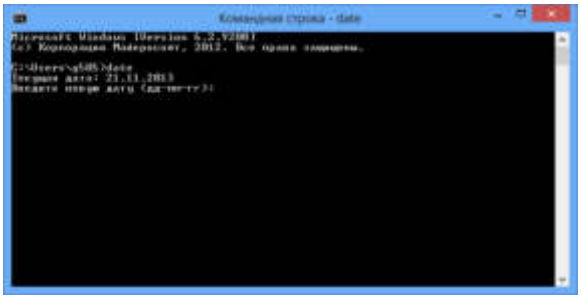
Важно! Все этапы работы необходимо занести в отчет, используя сочетание клавиш Alt+PrtScr для создания снимка активного окна с экрана и последующей его вставки в файл.

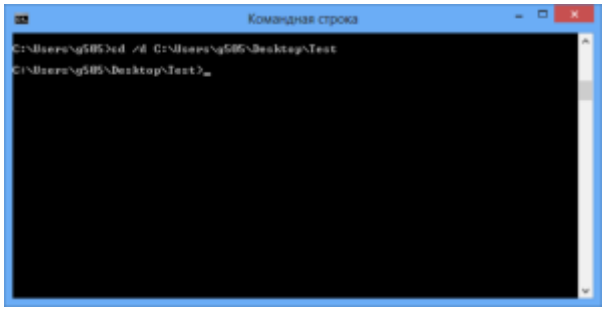
1. Информация о командной строке Windows.
Командная среда — это программный продукт Microsoft, который обеспечивает связь между пользователем компьютера и операционной системой. Командная оболочка Windows использует интерпретатор команд cmd.exe и присутствует во всех версиях операционных систем Windows. Многие возможности и функции управления операционной системой недоступны из графического интерфейса и поэтому cmd является единственным средством доступа к этим инструментам.
Отличием работы из командной строки является полное отсутствие больших и громоздких графических утилит. Пользовательский интерфейс текстовой строки предоставляет среду, в которой выполняются приложения и служебные программы. Среда, эмулирующая DOS имеет множество названий, таких как – командная строка, окно, среда и.т.д.
2. Файлы, каталоги, файловая система - дерево каталогов. Практически вся информация на компьютерах представлена в виде файлов. Файл является основной единицей хранения данных и программ, обрабатывающих эти данные. Файл — это именованная (т.е. снабженная именем) область внешней памяти. Операционная система и прикладные программы (приложения) получают доступ к файлу по его имени. Максимальная длина имени файла или каталога в Windows 256 символов, включая расширение, имя и расширение разделяются точкой. Расширение указывает на вид информации или на приложение, которым может быть открыт этот файл, например myfile.txt - текстовый файл, myfile.doc - документ MS Word и т.д.
2.1. Дерево каталогов
Файлы хранятся в системе вложенных каталогов (директорий) и организуются в файловую систему. Таким образом, файловой системой называется совокупность файлов и каталогов, организованных в древовидную структуру. Ее можно представить как перевернутое вверх корнем дерево. Узлами, из которых расходятся "ветви", являются каталоги, восходящие, в конечном счете, к корневому каталогу. Узлам, из которых не происходит дальнейшего ветвления, как правило, соответствуют файлы, хотя это могут быть и пустые каталоги. Обычно мы говорим: "Каталог (директория) содержит файлы" или "Файл находится в каталоге". Но при этом понимаем, что каталог не является областью памяти, собственно вмещающей сами файлы. Каталог лишь содержит список файлов, с указанием их имен и других атрибутов. По сути, каталог — это специализированный файл, назначением которого является хранение списка отнесенных к нему файлов (в том числе и подкаталогов, которые, как и все каталоги - тоже файлы). "Пустая" файловая система состоит только из корневого каталога.
2.2. Рабочий каталог. Текущий каталог, абсолютный путь и относительный путь.
Когда мы входим в систему, то попадаем в свой рабочий каталог, он задан для нас системным администратором и сами мы его поменять не можем. В момент входа он является так же текущим каталогом. Текущим мы можем сделать любой существующий каталог, к которому у нас есть доступ. Зачем нужен текущий каталог? Дело в том, что операционная система осуществляет доступ к файлу или каталогу через его путевое имя или, попросту путь. Существуют два вида путей, абсолютные и относительные. Абсолютный путь - это последовательность имен каталогов, которая начинается от корневого каталога и, следуя по дереву файловой системы, заканчивается именем каталога или файла, с которым мы хотим работать. Относительный путь может быть указан от текущего или рабочего каталога, что может оказаться значительно короче и удобней, чем использование абсолютного пути. Когда мы находимся в рабочем каталоге, нам не нужно указывать пути к каталагам и файлам находящимся в нем. К другим каталогам придется указать путь, но не абсолютный, а гораздо более короткий. Но об этом немного позднее, а пока приступим к работе.
3. Полезная функция. Попробуйте понажимать клавиши "ВВЕРХ" и "ВНИЗ" на клавиатуре, и Вы увидите, что командная строка запоминает команды, и этими самыми стрелочками их можно перелистывать и, при необходимости редактировать.
ЗАДАНИЕ

- Создайте на рабочем столе папку Test.
- Запустите командную строку cmd.exe (Пуск — ввод с клавиатуры «cmd» без кавычек).



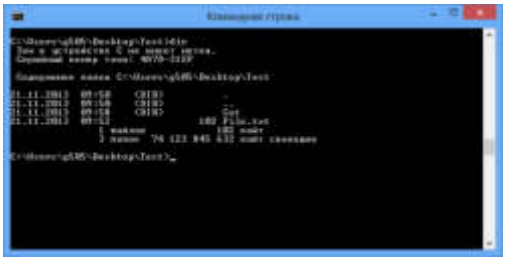
- Проверьте системные дату и время с помощью команд date и time. Для этого в командной строке наберите нужную команду и нажмите Enter.





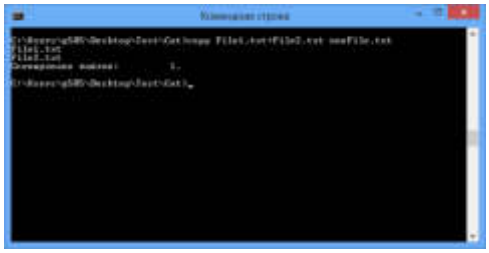
- С помощью утилиты cd измените текущий каталог на каталог Test, созданный Вами ранее.

- С помощью команды md создайте каталог с именем Cat.
- Используя команду copy con, создайте файл с именем File.txt. (Команда copy con означает копирование с консоли, т. е. с клавиатуры). После данной команды введите следующий текст: Ваши Ф.И.О., группа и название лабораторной работы. Закройте файл сочетанием клавиш Ctrl+Z.

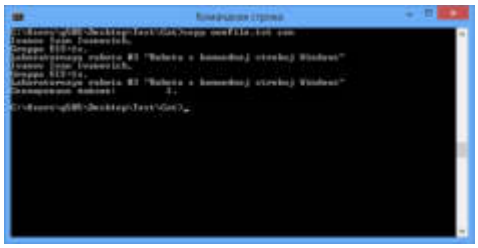


- С помощью команды dir просмотрите список созданных объектов в папке Test. Команда в общей сложности фиксирует 3 каталога (папки), т. к. первая метка указывает на текущий каталог, обозначенный точкой, вторая — на предыдущий каталог (две точки).
- В каталоге Cat с помощью команды copy создайте две копии файла File.txt – File1.txt и File2.txt.

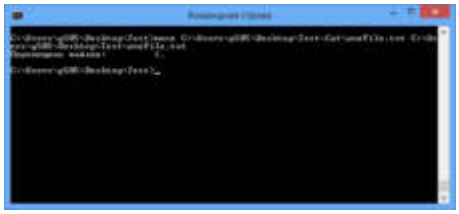
- Объедините файлы File1.txt и File2.txt в файл oneFile.txt с помощью команды copy.



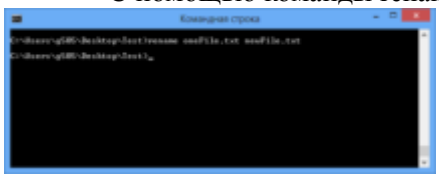
- Просмотрите полученный файл oneFile с помощью утилиты copy.



- С помощью команды move переместите файл oneFile.txt в папку Test.



- С помощью команды rename переименуйте файл oneFile.txt в newFile.txt.

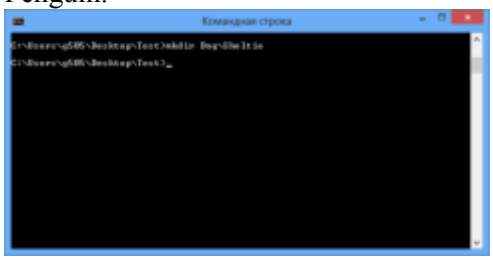


- С помощью той же команды смените расширения у всех файлов в каталоге Cat.

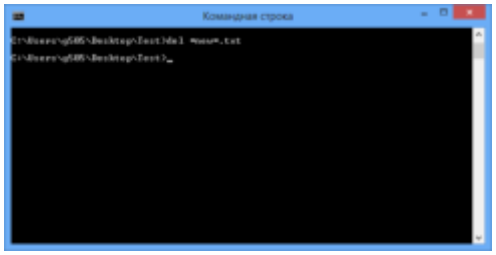
- Измените расширение файлов на исходное.

- Создайте в папке Test две цепочки вложенных каталогов с помощью команды mkdir:

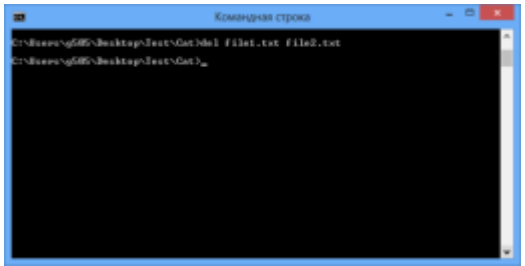
а) Каталог Dog, содержащий каталоги Bulldog и Sheltie.
б) Каталог Bird, содержащий каталоги Flying и Non-flying. Каталог Non-flying содержит два подкаталога: Ostrich и Penguin.

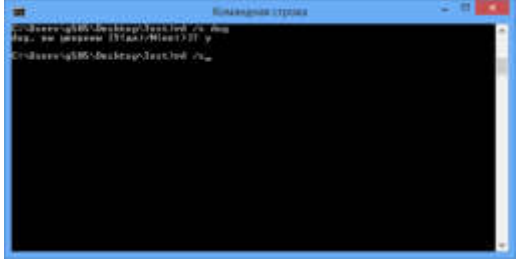


- Используя команду tree. постройте дерево каталогов папки Test.

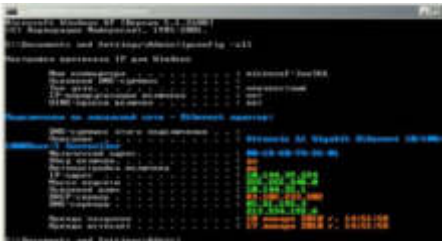


- Удалите все файлы, содержащие в названии слово new, из каталога Test посредством утилиты del.





- Удалите все файлы из каталога Cat.
- Удалите каталоги Bird и Dog с помощью команды rd /s.



1. Утилита ipconfig

Утилита ipconfig служит для отображения параметров текущих сетевых подключений, а также для управления клиентскими сервисами DHCP и DNS.

Синтаксис ввода: **ipconfig [/all] [/renew [адаптер]] [/release [адаптер]]**

При вводе команды могут использоваться ключи (дополнительные параметры). Список наиболее часто употребляемых параметров представлен в таблице 1. При вводе команды ipconfig без параметров выводится только IP-адрес, маска подсети и основной шлюз для каждого сетевого адаптера.

Таблица 1. Список наиболее часто употребляемых ключей при вызове утилиты ipconfig

Ключ	Описание
/all	Отображение полной информации по всем сетевым подключениям.
/release	Сброс текущих параметров сетевого подключения и освобождение сетевого адреса, предоставленного DHCP сервером.
/renew	Обновление параметров сетевого подключения.
/?	Отображает справку в командной строке.

На рисунке 1 показаны результаты ввода команды ipconfig с ключом /all для компьютера с одним активным сетевым подключением.

Рисунок 1. Результаты ввода команды ipconfig с ключом /all.

Результаты вывода можно условно разделить на три группы:

- К первой группе (значения, выделенные синим цветом) относится общая информация о сетевом подключении:
  - “Подключение по локальной сети“ - Имя сетевого

подключения

- “Ethernet адаптер“- Тип адаптера
- “Attansic L1 Gigabit Ethernet 10/100/1000Base-T Controller“- Описание адаптера
- “00-1D-60-74-26-01“- Физический адрес (MAC-адрес) адаптера

- Ко второй группе (значения, выделенные зеленым цветом) относится информация о сетевых настройках подключения:

- “10.144.39.224”- Сетевой адрес (IP-адрес) подключения
- “255.255.248.0”- Маска подсети
- “10.144.32.1”- Адрес шлюза
- “85.21.192.3”, “213.234.192.8”- Адреса серверов DNS

- К третьей группе (значения, выделенные оранжевым цветом) относится информация о деталях аренды адреса у DHCP сервера:

- “Dhcp включен: да”- Функция получения параметров у DHCP сервера включена
- “Автонастройка включена: да”- Функция авто-настройки подключения включена
- “83.102.233.202”- Адрес DHCP сервера, у которого получены параметры
- “10 января 2010 г. 14:51:58”- Дата получения параметров
- “17 января 2010 г. 14:51:58”- Дата истечения аренды сетевого адрес

2. Утилита ping

Утилита ping предназначена для проверки работоспособности соединения между двумя устройствами на уровне протокола IP (сетевом уровне). Утилита выполняет проверку, посылая на указанный сетевой адрес эхо-запросы (ICMP Echo-Request) протокола ICMP и фиксирует получение эхо-ответов (ICMP Echo-Reply).

Синтаксис ввода: **ping [-t] [-a] [-n счетчик] [имя_конечного_устройства]**

Ключ	Описание
/t	Задает для команды ping отправку эхо-запросов к точке назначения до тех пор, пока команда не будет прервана. Для прерывания команды и вывода статистики нажмите

	комбинацию CTRL+BREAK. Для прерывания команды ping и выхода из нее нажмите клавиши CTRL+C.
/a	Задает разрешение DNS имени по IP-адресу назначения, либо IP-адрес по DNS имени назначения. В случае успешного выполнения выводится имя и IP-адрес удалённого устройства.
/n	Задает число отправляемых эхо-запросов.
/?	Отображает справку в командной строке.

При вводе команды могут использоваться ключи (дополнительные параметры). Список наиболее часто употребляемых параметров представлен в таблице 2.

Таблица 2. Список наиболее часто употребляемых ключей при вызове утилиты ping

На рисунке 2 показаны результаты успешного выполнения команды ping:

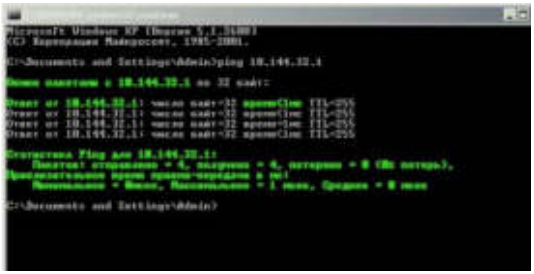


Рисунок 2. Успешное выполнение команды ping.

Полученные результаты можно интерпретировать следующим образом:

- “Обмен пакетами с 10.144.32.1” – локальное устройство начало посылать удалённому устройству эхо запросы.
- “Ответ от 10.144.32.1” – от удалённого устройства пришёл эхо-ответ.
- “время<1мс” – RTT (Round Trip Time) – время, затраченное на отправку эхо-запроса и получение эхо-ответа.
- “Статистика Ping для 10.144.32.1: Пакетов: отправлено = 4, получено = 4, потеряно = 0 (0% потерь), Приблизительное время приема-передачи в мс: Минимальное = 0мсек,

Максимальное = 1 мсек, Среднее = 0 мсек” - Статистика результата выполнения команды ping.

Если сообщение с эхо-ответом не получено в пределах заданного интервала, то выдается сообщение об ошибке "Превышен интервал ожидания для запроса". Интервал по умолчанию равен 4 секунды.

Причины отсутствия эхо-ответа могут быть следующими:

- Удалённое устройство не обрабатывает эхо-запросы или не отправляет эхо-ответы из соображений безопасности.
- Осуществить взаимодействие между устройствами невозможно из-за проблем в каналах связи.
- Одно из устройств работает некорректно.
- Качество линий связи настолько низкое, что эхо-ответы не приходят в течение интервала ожидания.
- Удалённый хост не существует.

На рисунке 3 показаны результаты выполнения команды ping для несуществующего хоста:

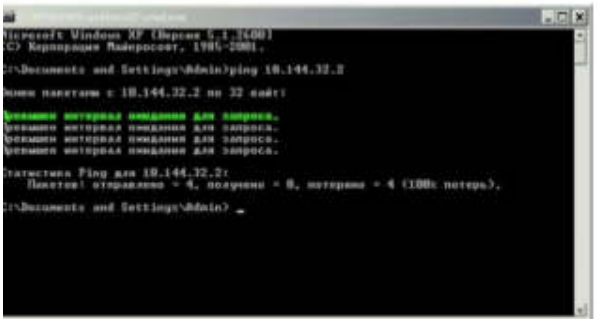


Рисунок 3. Неудачное выполнение команды ping.

3. Утилита traceroute

Утилита traceroute определяет путь, по которому проходят пакеты между локальным и удалённым устройствами. Полученный путь представляет собой это список ближайших интерфейсов устройств, работающих на сетевом уровне, находящихся на пути между устройствами. Утилита используется для локализации проблем, или для сбора информации о наличии устройств в сети. Работа утилиты, так же как и работа утилиты ping основана на отправке эхо-запросов.

Синтаксис ввода: **tracert [-d] [-h число] [имя_конечного_устройства]**

При вводе команды могут использоваться ключи (дополнительные параметры). Список наиболее часто употребляемых параметров

представлен в таблице 2.

Таблица 3. Список наиболее часто употребляемых ключей при вызове

Ключ	Описание
/d	Предотвращает попытки команды tracert разрешения IP-адресов промежуточных устройств в имена. Увеличивает скорость вывода результатов команды tracert
/h	Задает максимальное количество переходов на пути при поиске удалённого устройства.
/?	Отображает справку в командной строке.

утилиты traceroute.

На рисунке 4 показан результат выполнения команды tracert до информационного ресурса www.ya.ru.

Рисунок 4. Результаты трассировки до информационного ресурса www.ya.ru.

Полученные результаты можно интерпретировать следующим образом:

“Трассировка маршрута к ya.ru [77.88.21.8] с максимальным числом прыжков 30:” - Локальное устройство начало посылать удалённому устройству эхо-запросы. Максимальное количество промежуточных устройств, которые могут встретиться на пути, равно 30. Если количество промежуточных устройств превысит 30 – трассировка окажется неудачной. Далее следует вывод ответов, полученных от промежуточных устройств. Формат вывода выглядит следующим образом:

X Y Y Y Имя [IP-адрес], где:

- X – номер промежуточного устройства.
- Y – Время через которое получен эхо-ответ от промежуточного устройства (3 эхо-запроса отправляются последовательно). В

случае, если эхо-ответ не получен по истечении интервала ожидания, вместо времени выводится * (астерик).

- Имя – DNS имя промежуточного устройства
- IP-адрес – сетевой адрес промежуточного устройства. В случае, если DNS имя промежуточного устройства невозможно получить, выводится только сетевой адрес.

Ниже приведена интерпретация некоторых записей из примера:

“1 <1 мс <1 мс <1 мс 85.21.0.239“

- Получены все три эхо-ответа от первого промежуточного устройства. Время отклика во всех 3х случаях не превышает 1 миллисекунду.
- Сетевой адрес устройства 85.21.0.239.
- Имя устройства получить не удалось

“2 * * * Превышен интервал ожидания для запроса.“

- Ни один эхо-ответ от второго промежуточного устройства не получен. Если все последующие записи выглядят таким же образом – удалённое устройство недостижимо. Если эхо-ответы от следующих устройств получены, значит текущее устройство не отправляет эхо-ответы по какой-либо причине.

“4 1 ms 1 ms 1 ms yandex.msk.corbina.net [195.14.32.50]“

- Получены все три эхо-ответа от первого промежуточного устройства. Время отклика во всех 3х случаях не превышает 1 миллисекунду.
- Сетевой адрес устройства 195.14.32.50. Имя устройства yandex.msk.corbina.net.

“7 1 ms 1 ms 1 ms ya.ru [77.88.21.8]”

- Получены все три эхо-ответа от удалённого устройства – информационного ресурса www.ya.ru. Время отклика во всех 3х случаях не превышает 1 миллисекунду. Трассировка успешно завершена.

4. Команда arp

Служит для вывода и изменения записей кэша протокола ARP, который содержит одну или несколько таблиц, использующихся для хранения IP-адресов и соответствующих им физических адресов Ethernet. Для каждого сетевого адаптера Ethernet, установленного в компьютере, используется отдельная таблица.

Синтаксис ввода: **arp [-a [IP_адрес] [-N MAC_адрес]]**

При вводе команды могут использоваться ключи (дополнительные параметры). Список наиболее часто употребляемых параметров

представлен в таблице 4.

Таблица 4. Список наиболее часто употребляемых ключей при вызове утилиты arp.

Ключ	Описание
/a	Вывод содержимого ARP всех адаптеров.
/N	Вывод содержимого ARP кэша определённого адаптера.
/?	Отображает справку в командной строке.

На рисунке 7 показан результат выполнения команды arp.

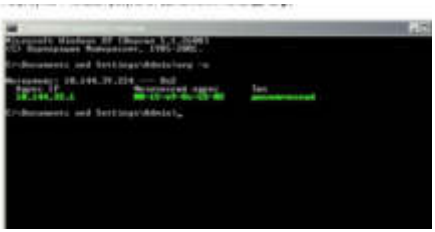


Рисунок 7. Результат выполнения команды arp.

Полученные результаты можно интерпретировать следующим образом:

– “ Адрес IP Физический адрес
10.144.32.1 00-15-e9-8c-22-02 “

Устройство с сетевым адресом 10.144.32.1 и физическим адресом 00-15-e9-8c-22-02.

– “Тип Динамический “ - Данная запись была создана автоматически, когда локальному устройству понадобилось обратиться к устройству с адресом 10.144.32.1. Если в течении 5 минут после создания записи устройства не будут взаимодействовать – запись будет удалена.

ЗАДАНИЕ

1. С помощью утилиты ipconfig определите и запишите в отчет следующую информацию:
   - Название сетевого подключения;
   - Тип используемого адаптера;

- MAC-адрес адаптера;
- IP-адрес сетевого подключения;
- Сетевую маску;
- Основной шлюз;
- IP-адрес DNS-сервера;
- IP-адрес DHCP-сервера.

2. С помощью утилиты ping проверьте доступность следующих устройств:

- Сервер DHSP
- Сервер DNS
- Информационный ресурс www.ystu.ru.

Используя дополнительные ключи, сделайте так, чтобы количество посылаемых эхо-запросов равнялось номеру компьютера (последние 2 цифры в имени компьютера) + 5.

Для каждого устройства и информационного ресурса запишите в отчёт следующую информацию:

- Процент потерь.
- Среднее время приёма передачи.

3. С помощью утилиты tracert проверьте доступность следующих устройств:

- Информационный ресурс www.ystu.ru.
- Информационный ресурс www.ya.ru.

Используя дополнительные ключи, сделать так, чтобы утилита не определяла DNS имена промежуточных устройств.

Запишите в отчёт следующую информацию:

- Количество промежуточных устройств.
- IP-адрес всех промежуточных устройств.

4. С помощью команды arp определите и запишите в отчет MAC-адреса следующих устройств:
   • Основной шлюз;
   • 3 любых компьютера.

5. Содержание отчета

Отчет должен содержать:

1.Название работы.

2.Цель работы.

3.Задание и его решение.

4. Вывод по работе

Практическая работа №13
"Пакетная загрузка информации с помощью Python" (4 часа)

Цель работы: познакомиться со средой разработки Python.

Изучить основные типы данных, команды ввода и вывода данных.

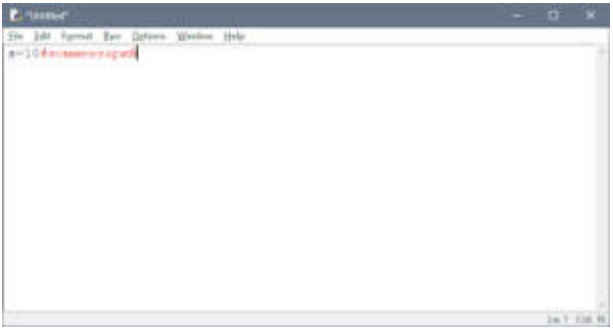
Краткая теория

Python– это объектно-ориентированный, интерпретируемый, переносимый язык
сверхвысокого уровня. Программирование на Python позволяет получать быстро
и качественно необходимые программные модули.В комплекте вместе с
интерпретатором Python идет IDLE (интегрированная среда разработки). По своей сути она подобна интерпретатору, запущенному в
интерактивном режиме с расширенным набором возможностей (подсветка синтаксиса, просмотр объектов, отладка и т.п.).
Для запуска IDLE в Windows необходимо перейти в папку Python в меню
“Пуск и найти там ярлык с именем “IDLE (Python 3.X XX-bit)”.
Для запуска редактора программы (кода) следует выполнить команду File->New File или сочетание клавиш Ctrl+N.
Любая Python-программа состоит из последовательности допустимых символов, записанных в определенном порядке и по определенным правилам.
Программа включает в себя:

- комментарии;
- команды;
- знаки пунктуации;
- идентификаторы;
- ключевые слова.

Комментарии в Python обозначаются предваряющим их символом #

и продолжаются до конца строки (т.е. в Python все комментарии являются однострочными), при этом не допускается использование перед символом # кавычек:



Знаки пунктуации
В алфавит Python входит достаточное количество знаков пунктуации, которые
используются для различных целей. Например, знаки "+" или " *» могут
использоваться для сложения и умножения, а знак запятой ",» - для разделения
параметров функций.
Идентификаторы
Идентификаторы в Python это имена, используемые для обозначения переменной,
функции, класса, модуля или другого объекта.
Ключевые слова
Некоторые слова имеют в Python специальное назначение и представляют собой управляющие конструкции языка.
Ключевые слова в Python:
['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif',
'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal',
'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']

Типы данных

1. None (неопределенное значение переменной)
2. Логические переменные (Boolean Type)
3. Числа (Numeric Type)
    1. int – целое число
    2. float – число с плавающей точкой
    3. complex – комплексное число
4. Списки (Sequence Type)
    1. list – список
    2. tuple – кортеж
    3. range – диапазон
5. Строки (Text Sequence *Type*)
    1. str

Ввод и вывод данных

Ввод данных осуществляется при помощи команды input(список ввода):

```
a = input()
print(a)
```

В скобках функции можно указать сообщение - комментарий к вводимым данным:

```
a = input ("Введите количество: ")
```

Команда input() по умолчанию воспринимает входные данные как строку символов. Поэтому, чтобы ввести целочисленное значение, следует указать тип данных int():

```
a = int (input())
```

Для ввода вещественных чисел применяется команда

```
a=float(input())
```

Вывод данных осуществляется при помощи команды print(список вывода):

```
a = 1
b = 2
print(a)
print(a + b)
print('сумма = ', a + b)
```

Существует возможность записи команд в одну строку, разделяя их

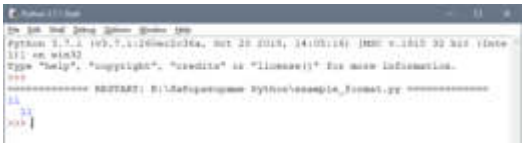
через ;. Однако не следует часто использовать такой способ, это снижает удобочитаемость:
a = 1; b = 2; print(a)
print (a + b)
print ('сумма = ', a + b)
Для команды print может задаваться так называемый сепаратор — разделитель между элементами вывода:
x=2
y=5
print (x, "+", y, "=", x+y, sep = " ")
Результат отобразится с пробелами между элементами: 2 + 5 = 7
Простые арифметические операции над числами

x + y	Сложение
x - y	Вычитание
x * y	Умножение
x / y	Деление

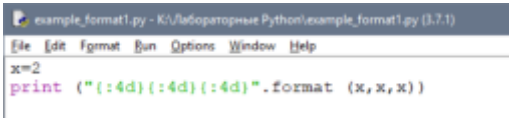
Пример программы на Python

переменных. Результат выполнения программы с

применением простых арифметических операций
Для форматированного вывода используется **format**:
Строковый метод format() возвращает отформатированную версию строки,
заменяя идентификаторы в фигурных скобках {}. Идентификаторы могут
быть позиционными, числовыми индексами, ключами словарей, именами
Синтаксис команды **format**:
поле замены := "{" [имя поля] ["!" преобразование] [":" спецификация] "}"
имя поля := arg_name ("." имя атрибута | "[" индекс "]")*
преобразование := "r" (внутреннее представление) | "s" (человеческое представление)
спецификация := см. ниже
Аргументов в format() может быть больше, чем идентификаторов в строке.
В таком случае оставшиеся игнорируются.
Идентификаторы могут быть либо индексами аргументов, либо ключами:



В результате выведется число 11, а перед ним два пробела, так как указано
использовать для вывода четыре знакоместа.
Или с несколькими аргументами:



В итоге каждое из значений выводится из расчета 4 знакоместа.

Спецификация формата:

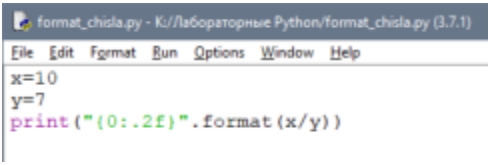
спецификация	:= [[fill]align][sign][#][0][width][,][.precision][type]
заполнитель	:= символ кроме '{' или '}'
выравнивание	:= "<" | ">" | "=" | "^"
знак	:= "+" | "-" | " "
ширина	:= integer
точность	:= integer
тип	:= "b" | "c" | "d" | "e" | "E" | "f" | "F" | "g" | "G" | "n" | "o" | "s" | "x" | "X" | "%"

Тип	**Значение**
'd', 'i', 'u'	Десятичное число.
'o'	Число в восьмеричной системе счисления.
'x'	Число в шестнадцатеричной системе счисления (буквы в нижнем регистре).
'X'	Число в шестнадцатеричной системе счисления (буквы в верхнем регистре).
'e'	Число с плавающей точкой с экспонентой (экспонента в нижнем регистре).
'E'	Число с плавающей точкой с экспонентой (экспонента в верхнем регистре).
'f', 'F'	Число с плавающей точкой (обычный формат).
'g'	Число с плавающей точкой. с экспонентой (экспонента в нижнем регистре), если она меньше,

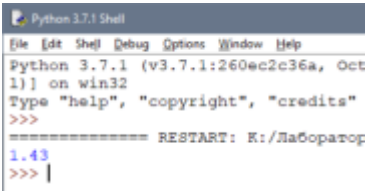
	чем -4 или точности, иначе обычный формат.
'G'	Число с плавающей точкой. с экспонентой (экспонента в верхнем регистре), если она меньше, чем -4 или точности, иначе обычный формат.
'c'	Символ (строка из одного символа или число - код символа).
's'	Строка.
'%'	Число умножается на 100, отображается число с плавающей точкой, а за ним знак %.

Для форматирования вещественных чисел с плавающей точкой используется следующая команда:

print('{0:.2f}'.format(вещественное число))



В результате выведется число с двумя знаками после запятой.



Пример
Напишите программу, которая запрашивала бы у пользователя:
Вариант 0
- ФИО ("Ваши фамилия, имя, отчество?")
- возраст ("Сколько Вам лет?")

- место жительства ("Где вы живете?")
После этого выводила бы три строки:
"Ваше имя"
"Ваш возраст"
"Вы живете в"

Решение

```
a=input('Введите ваши фамилию, имя, отчество ')
b=input('Сколько вам лет? ')
c=input('Где вы живете? ')
print('Ваше имя ',a)
print('Ваш возраст ',b)
print('Вы живете в ',c)
```

Задание «Ввод и вывод данных на языке Python»

1. Напишите программу (файл proba.py), которая запрашивала бы у пользователя два числа и выводила на экран их сумму, разность, произведение и частное.

Наша программа должна иметь дружественный интерфейс – т.е. вести диалог с пользователем. Организуем диалог с помощью функции input и инструкции print:

```
a=int(input("Введите первое число "))
b=int(input("Введите второе число "))
print ("Сумма чисел равна ", a+b)
```

int преобразует введе
клавиатуры в целое число
a,b – переменные, в которы

2. Напишите программу (файл user.py), которая запрашивала бы у пользователя:

- его имя (например, "What is your name?")
- возраст ("How old are you?")
- место жительства ("Where are you live?") После этого выводила бы три строки:

"This is *имя*"

```
print("This is", имя перем
```

"It is *возраст*"

"(S)he live in *место_жительства*"

Вместо *имя, возраст, место_жительства* должны быть данные, введенные пользователем. Примечание: можно писать фразы на русском языке, но, если вы планируете стать профессиональным программистом, привыкайте к английскому.

3. Напишите программу (файл arithmetic.py), которая предлагала бы пользователю решить пример 4 * 100 - 54. Потом выводила бы на экран правильный ответ и ответ пользователя. Подумайте, нужно ли здесь преобразовывать строку в число.

4. Запросите у пользователя четыре числа. Отдельно сложите первые два и отдельно вторые два. Разделите первую сумму на вторую. Выведите результат на экран так, чтобы ответ содержал только целое число. *

Задания для самостоятельной работы (по вариантам)

Напишите программу, которая запрашивала бы у пользователя:

Вариант 1
Имя, Фамилия, Возраст, Место жительства
- фамилия, имя ("Ваши фамилия, имя?")
- возраст ("Сколько Вам лет?")
- место жительства ("Где вы живете?")
После этого выводила бы три строки:
"Ваши фамилия, имя"
"Ваш возраст"
"Вы живете в"

Вариант 2
Имя, , Дата рождения, Образование
- имя ("Ваше, имя?")
- дата рождения ("Ваша дата рождения?")
- образование ("Где Вы учитесь?")
После этого выводила бы три строки:
"Ваше имя"
"Дата рождения"
"Вы учитесь в "

5. Содержание отчета

Отчет должен содержать:

1.Название работы.
2.Цель работы.
3.Задание и его решение.
4. Вывод по работе

Практическая работа №14
"Пакетная обработка файлов с помощью Python" (4часа)

Цель: Рассмотрение способов работы с файлами Python"

1 Работа с файлами

В языке Python широко используются файлы для выполнения операций с данными, которые предполагается хранить длительное время. Для работы с файлами используется специальный объект файла, который содержит методы для чтения данных из файла и записи данных в файл.

1.1 Открытие файла

Открыть файл можно с помощью встроенной функции *open()*:

open(file, mode='r', buffering=-1, encoding=None, errors=None, newline=None, closefd=True, opener=None)

Функция открывает файл и возвращает соответствующий файловый объект для последующего оперирования с ним. Если файл не может быть открыт – вызывается исключение OSError:

- *file – единственный обязательный параметр, задает имя как текстового, так и двоичного файла (и путь к файлу, если файл не находится в текущем каталоге). Также этот параметр может указывать дескриптор файла в виде целого числа. Если задан дескрипртор файла, то он закрывается, когда закрывается файл (в том случае, если аргумент closefd не равен False);*
  - *mode – задает режимы открытия файла:*
    - *'w' – открывает файл только для записи, если такой файл уже существует – его содержимое будет перезаписано, т.е. размер текущего файлоа будет уменьшен до нуля (truncating), иначе – создается новый файл для записи;*

- *'r' – открыть файл только для чтения (значение по умолчанию, синоним 'rt'), указатель устанавливается в начало файла, если файл не найден – возникает исключение FileNotFoundError;*
- *'a' – открыть файл для добавления, если файл существует – предыдущее содержимое файла сохраняется и указатель устанавливается в конец файла (независимо от текущего положения указателя файла), иначе – файл создается для записи;*
- *'b' – режим двоичных данных;*
- *'t' – режим текстовых данных (значение по умолчанию);*
- *'+' – открыть файл для чтения или записи, указатель устанавливается в начало файла. Различие между режимами 'w+' ('w+b') и 'r+' ('r+b') следующее – если указанный файл существует, то в первом случае его содержимое очищается (размер становится равным нулю), а во втором случае – нет;*
- *'x' – при попытке открыть для записи существующий файл возникает исключение FileExistsError (параметр введен начиная с версии Python 3.3);*

(значения 'w', 'r', 'a', 'b', 't', '+' и 'x' при совместном использовании могут указываться в произвольном порядке). Python поддерживает два вида файлов – двоичные, при открытии которых для аргумента mode указано значение 'b', и текстовые, при открытии которых аргумент mode не имеет значения 'b'. Текстовые файлы возвращают содержимое в виде строк, а двоичные файлы – в виде объектов типа bytes (см. подраздел 3.1 лаб. раб. №4);

- *buffering – устанавливает размер буфера при работе с файлом:*
  - *если размер равен 0 (только для двоичных файлов), то режим буферизации не используется и операции чтения и записи выполняются напрямую с диска на диск;*
  - *если размер равен 1 (только для текстовых файлов), то используется построковая буферизация (значение по умолчанию);*

- *если размер больше 1, то для двоичных файлов устанавливается размер буфера, т.е. размер блоков, на которые делится файл. Размер буфера выбирается исходя из параметров устройств обмена с целью оптимизации выполнения файловых операций. По умолчанию его размер равен значению io.DEFAULT_BUFFER_SIZE:*

```
>>> import io
io.DEFAULT_BUFFER_SIZE
8192
```

- *encoding – указывает тип кодировки (используется только для текстовых файлов). Значение по умолчанию можно получить, используя метод locale.getpreferredencoding():*

```
>>> import locale
locale.getpreferredencoding()
'cp1251'
```

- *errors – указывает, как будут обрабатываться ошибки кодирования и декодирования (только для текстового режима):*
  - *если параметр не указан или указано значение 'strict' – при ошибке вырабатывается исключение ValueError;*
  - *'ignore' – игнорирует ошибку;*
  - *'replace' – вызывает установку маркера (например, "?") в том месте, где произошла ошибка;*
  - *'xmlcharrefreplace' – заменяет соответствующим XML-символом;*
  - *backslashreplace' – заменяет ESC-последовательностью с обратным слэшем;*
- *newline – определяет, как будут обрабатываться символы конца строки:*
  - *если параметр не указан или указано значение None – осуществляется следующая трансляция символов конца строки: при записи в файл – символ \n преобразуется в форму, специфичную для платформы*

(\n для Unix и \r\n для Windows), при чтении из файла: из формы, специфичной для платформы в символ \n;

- *если указан параметр '', трансляция символов конца строки не выполняется;*

- *closefd – определяет, будет ли закрыт дескриптор файла после закрытия файла:*
  - *если closefd = True (значение по умолчанию) – будет;*
  - *если closefd = False – нет;*
- *opener – указывает объект ABC-класса callable (например, функцию), который может быть использован для открытия файла.*

Тип файла устанавливается при его открытии в соответствии с указанными значениями аргументов *mode* и *buffering* функции *open()*:

- *текстовый файл (класс _io.TextIOWrapper) – все режимы, в которых аргумент mode не содержит признак 'b' (binary);*
- *двоичный файл с буферизацией для записи (класс _io.BufferedWriter) – mode='wb' или mode='ab' и buffering<>0;*
- *двоичный файл с буферизацией для чтения (класс _io.BufferedReader) – mode='rb' и buffering<>0;*
- *двоичный файл с буферизацией для записи и чтения (класс _io.BufferedRandom) – все режимы, в которых параметр mode содержит признаки 'b+' и buffering<>0;*
- *двоичный файл без буферизации (класс _io.FileIO) – все режимы, в которых аргумент mode содержит признак 'b' и buffering=0.*

Текстовые файлы отличаются тем, что содержат строки, осуществляют кодирование и декодирование символов и обработку символов новой строки.
Пример открытия текстового файла для записи:
```
>>> f= open ( 'text.txt' , 'w' )
```
Указав ссылку на файл, можно определить его параметры:
```
>>> f
<_io.TextIOWrapper name='text.txt' mode='w' encoding='cp1251'>
```
, т.е. это:*текстовый файл класса io.TextIOWrapper;*

- *режим работы – запись;*

- *кодировка символов – 'cp1251'.*

1.2 Методы работы с файлами

После создания текстового файла для работы с ним могут быть использованы следующие методы:

- *write(строка) пишет в файл, открытый для записи, указанную строку, возвращая число записанных символов:*

```
>>> f.write( 'First line\n' )
11
>>> f.write( 'Second line\n' )
12 .
```

Записывать в текстовый файл можно только данные класса строки. Попытка записать в этот файл данные другого типа, например, последовательности байтов (см. описание класса bytes в подразделе 3.1 лаб. раб. №4) приведет к исключению:

```
>>> b=b'0123456789abcdef' # тип bytes
>>> f.write(b)
TypeError: must be str, not bytesread([число_байт]) – читает в одну строку из файла, открытого для чтения, число байтов, указанное в качестве необязательного аргумента. Если метод вызывается без аргументов или значением аргумента является -1, то читается весь файл. Если файл уже был прочитан до конца (встретился символ EOF), то метод read() возвращает пустую строку:
>>> f= open ( 'text.txt' )
>>> f.read()
'First line\nSecond line\n'
>>> f= open ( 'text.txt' )
>>> f.read(15)
'First line\nSeco' ;
```

- *readline() – читает из файла, открытого для чтения, строку:*

```
>>> f= open ( 'text.txt' )
>>> f.readline()
'First                                                    line\n'
>>> f.readline()
'Second                                                  line\n'
>>> f.readline()
'' ;
```

- *readlines([число_байт]) – читает из файла, открытого для чтения, все строки файла, возвращая список строк. Если указан необязательный параметр, то читается заданное число байт, затем дополнительно читаются байты до конца текущей строки (т.е. строки не обрываются):*

```
>>> f= open ( 'text.txt' )
>>> f.readlines()
['First line\n', 'Second line\n']
>>> f= open ( 'text.txt' )
>>> f.readlines(5)
['First                                                   line\n']
```

Текстовый файл можно прочитать в цикле построчно:

```
>>> f= open ( 'text.txt' )
>>> for line in f:
print (line)
First                                                    line
Second line ;
```

- *writelines(список) – записывает в файл строки, заданные в виде списка:*

```
>>> dig=[ '1' , '2' , '3' ]
>>> f= open ( 'text.txt' , 'a' )
>>> f.writelines(dig)
>>> f= open ('text.txt')
>>> for line in f:
print (line)
```

```
First                                                    line
Second                                                   line
123 ;
```

- *close() – закрывает файл, перенося все сделанные в нем изменения на диск, возвращая ресурсы операционной системе. Необходимо закрывать все файлы, которые больше не используются, т.к. не закрытый файл может привести к потере данных. После закрытия обращение к файлу автоматически вызывает ошибку:*

```
>>> f.close()
>>> f.writelines(dig)
ValueError: I/O operation on closed file
>>> f= open ( 'text.txt' )
>>> f.readlines()
['First line\n', 'Second line\n', '123']
```

- *readable() – возвращает значение True, если файл открыт для чтения, и False в противном случае:*

```
>>> f= open ( 'text.txt' )
>>> f.readable()
True ;
```

- *writable() – возвращает значение True, если файл открыт для записи, и False в противном случае:*

```
>>> f.writable()
False ;
```

- *seek(на_сколько_байт[, откуда]) – перемещает указатель текущего положения файла на заданное количество байт от позиции, указанной вторым аргументом:*
  - *0 – начало файла (значение по умолчанию);*
  - *1 – текущая позиция в файле;*
  - *2 – конец файла;*

При этом, число байт для перемещения может быть как положительным (вперед), так и отрицательным (назад). Для проверки этого метода создаем двоичный файл для записи и чтения:

```
>>> f= open ( 'text2.txt' , 'w+b' )
```

Записываем в него последовательность байтов из переменной b (b'0123456789abcdef'):

```
>>> f.write(b)
16,перемещаем указатель файла на 5 байтов вперед (относительно начала файла):

>>> f.seek(5)
5и читаем один байт из файла:

>>> f.read(1)
b'5'tell() – возвращает значение текущей позиции в файле:

>>> f.tell()
6Перемещаем указатель файла на 3 позиции относительного текущего положения:

>>> f.seek(3,1)
9и читаем 4 байта:

>>> f.read(4)
b'9abc'
```

И наконец, перемещаем указатель файла на 9 позиций назад относительно конца файла:

```
>>> f.seek(-9,2)
7и читаем байты из этой позиции до конца файла:
>>> f.read()
b'789abcdef'
```

- *seekable – возвращает значение True, если объект поддерживает метод seek(), и False в противном случае:*

```
>>> f.seekable()
True
```

- *truncate([размер]) – урезает файл до указанного размера, если аргумент не задан – до указателя текущего*

положения. Метод не работает для файла, открытого в режиме "только чтение":

```
>>> f.seek(8)
8>>> f.truncate()
8>>> f.read()
b'01234567'
>>> f.truncate(5)
5>>> f.read()
b'01234'
```

Передача содержимого скрипта клиенту

Работа с текстовыми файлами на языке Python показана в небольшом клиент-серверном приложении, в котором сервер предоставляет пользователю тексты запрашиваемых скриптов. Клиентская часть реализована в виде примера №1, в котором HTML-форма содержит поле ввода имени файла ("filename") и кнопки "Сброс" и "Получить файл":

```
<!DOCTYPE html>
<html>
<head>
<title>Передача файла клиенту</title>
<link rel="stylesheet" href="file:///D:/!PYTHON/lab.css">
</head>
<body>
<h1>Передача файла клиенту</h1>
<form method="get" action="http://zykov/PYTHON/get_file_form.py">
Имя файла:
<input type="text" class="form" name="filename"><br><br>
<input type="reset" value="Сброс" class="kn">  
<input type="submit" value="Получить файл" class="kn">
</form>
</body>
</html>
```

После ввода пользователем имени требуемого файла и нажатия кнопки "Получить файл" браузер передает введенное имя файла серверному приложению *get_file_form.py*:

```
#!c: . . . /Python35-32/python.exe
print ( "Content-Type: text/html\n" )
import cgi, cgitb, my_cgi
file=cgi.parse()[ "filename" ][0]
try :
f= open (file, encoding= "utf-8" )
print (my_cgi.H, "CGI: передача содержимого скрипта " ,file)
for el in f: my_cgi.D,el),
except FileNotFoundError : print (my_cgi.D, "Файл " , file, " не найден" ),
```

которое:

- *с помощью метода cgi.parse() получает имя файла, содержащего требуемый скрипт;*
- *пробует с использованием оператора try/except (см. подраздел 2.2) открыть этот файл для чтения;*
- *если файл с таким именем не найден – возникает исключение "FileNotFoundError", которое обрабатывается секцией except – клиенту сообщается, что такой файл не найден. На этом выполнение скрипта завершается;*
- *если файл найден, открывает его и передает клиенту сообщение "CGI: передача файла" с указанием имени файла;*
- *построчно считывает и передает клиенту содержимое файла.*

Если указать имя скрипта, который осуществляет приведенные выше действия, т.е. серверного приложения *get_file_form.py*, то клиент получит следующую Web-страницу, содержащую текст требуемого скрипта:

CGI: передача содержимого скрипта get_file_form.py

```
#!c:/ . . . Python35-32/python.exe

print("Content-Type: text/html\n")
```

```
import cgi, cgitb, my_cgi

file=cgi.parse()["filename"][0]

try :

f= open (file, encoding= "utf-8" )

print (my_cgi.H, "CGI: передача содержимого скрипта " ,file)

for el in f: my_cgi.D,el)

except FileNotFoundError : print (my_cgi.D, "Файл " , file, " не найден" ).
```

Типы исключений

Python имеет следующие встроенные типы исключений:

- *BaseException – базовый класс для всех встроенных исключений;*
- *Exception – все встроенные исключения, а также все пользовательские исключения должны наследоваться из этого класса;*
- *SystemExit – исключение, порождаемое функцией sys.exit при выходе из программы;*
- *SystemExit – исключение, порождаемое функцией sys.exit при выходе из программы.*
- *KeyboardInterrupt – возникает при прерывании программы пользователем (обычно комбинацией клавиш Ctrl+C;*
- *GeneratorExit – возникает при вызове метода close объекта generator;*
- *StopIteration – возникает встроенной функцией next, если в итераторе больше нет элементов;*
- *ArithmeticError – арифметическая ошибка;*
- *FloatingPointError – возникает при неудачном выполнении операции с плавающей запятой;*
- *OverflowError – возникает, когда результат арифметической операции слишком велик для представления.*

Не появляется при обычной работе с числами (так как Python поддерживает длинные числа), но может возникать в некоторых других случаях;

- *ZeroDivisionError – деление на ноль;*
- *AssertionError – выражение в функции assert ложно;*
- *AttributeError – объект не имеет данного атрибута (значения или метода)*
- *BufferError – операция, связанная с буфером, не может быть выполнена;*
- *EOFError – функция input() наткнулась на конец файла и не смогла при этом ничего прочитать;*
- *ImportError – не удалось импортирование модуля или его атрибута;*
- *LookupError – некорректный индекс или ключ;*
- *IndexError – индекс не входит в диапазон элементов;*
- *KeyError – несуществующий ключ (в словаре, множестве или другом объекте);*
- *MemoryError – недостаточно памяти;*
- *NameError – не найдено переменной с таким именем;*
- *UnboundLocalError – сделана ссылка на локальную переменную в функции, но переменная не определена ранее;*
- *OSError – ошибка, связанная с системой;*
- *io.BlockingIOError – возникает, когда используется блоковая операция ввода/вывода при неблокированном потоковом вводе/выводе;*
- *ChildProcessError – неудача при операции с дочерним процессом;*
- *ConnectionError – базовый класс для исключений, связанных с подключениями;*
- *BrokenPipeError – возникает, когда запись в контейнер невозможна;*
- *ConnectionAbortedError – возникает, когда подключение прекращино устройством;*
- *ConnectionRefusedError – возникает, когда попытка подключиться отвергнута устройством;*
- *ConnectionResetError – возникает, когда попытка подключиться сброшена устройством;*
- *FileExistsError – попытка создания файла или директории, которые уже существуют;*

- *InterruptedError – системный вызов прерван входящим сигналом;*
- *IsADirectoryError – ожидался файл, а это директория;*
- *NotADirectoryError – ожидалась директория, а это файл;*
- *PermissionError – не достаточно прав доступа;*
- *ProcessLookupError – указанного процесса не существует;*
- *TimeoutError – возникает, когда заканчивается интервал времени работы системной функции;*
- *ReferenceError – попытка доступа к атрибуту со слабой ссылкой;*
- *RuntimeError – возникает, когда исключение не попадает ни под одну из других категорий;*
- *SyntaxError – синтаксическая ошибка;*
- *– неправильные отступы;*
- *TabError – смешивание в отступах табуляции и пробелов;*
- *SystemError – внутренняя ошибка;*
- *ValueError – функция получает аргумент правильного типа, но некорректного значения;*
- *UnicodeError – ошибка, связанная с кодированием / раскодированием Unicode в строках;*
- *UnicodeEncodeError – исключение, связанное с кодированием Unicode;*
- *UnicodeDecodeError – исключение, связанное с декодированием Unicode;*
- *UnicodeTranslateError – исключение, связанное с переводом Unicode;*
- *Warning – предупреждение.*

Индивидуальные задания

Разработать приложение на базе технологии "клиент-сервер". В задачи клиентской части входит: разработать HTML-документ, содержащий форму, обязательными элементами которой являются:

- *три текстовых поля для задания данных, выбранных согласно колонке "Данные" табл. 1:*
  - *1 – Фамилия, инициалы;*

    - *2 – Наименование университета;*
    - *3 – Наименование факультета;*
    - *4 – Наименование кафедры;*
    - *5 – Наименование специальности;*
    - *6 – Курс;*
    - *7 – Наименование группы;*
    - *8 – Телефон;*
    - *9 – Мобильный телефон;*
    - *10 – Электронная почта;*
- *кнопка для передачи данных на сервер.*

В задачи серверной части входит:

- *принять данные из формы;*
- *создать каталог "CGI" и сделать его текущим;*
- *записать имена и значения полученных данных соответственно в файлы "names.txt" и "values.txt" с помощью методов, заданных колонкой "Методы"/"Запись" табл. 1 (первая цифра – для файла "names.txt", вторая – для файла "values.txt"):*
    - *1 – write();*
    - *2 – writelines();*
- *определить длину файлов "names.txt" и "values.txt";*
- *прочитать и передать клиенту содержимое файла "values.txt" с помощью метода, заданного колонкой "Методы"/"Чтение" табл. 1:*
    - *1 – read();*
    - *2 – readline();*
    - *3 – readlines();*
- *Создать с помощью встроенных функций bytes() range() (см. разделы 3 и 2) лаб. раб. №4) последовательность чисел длиной, заданной колонкой "Длина" табл. 1 и записать ее в двоичный файл "binary_data.dat";*
- *Прочитать значение байта файла "binary_data.dat", номер которого задан колонкой "Номер" табл. 1;*
- *Переместить указатель файла соответственно значениям колонок "Смещение" и "Откуда" табл. 1 и прочитать значения трех байтов;*
- *Проверить с использованием оператора try и встроенной функции assert(), выполняется ли условие, заданное колонкой*

"Условие" табл. 1, для целого числа, введенного функцией input():

Таблица 1 – Параметры индивидуальных заданий

Номер п/п	**Данные**	**Методы**		**Длина**	**Номер**	**Смещение**	**Откуда**	**Условие**
		записи	**чтения**					
1	1, 2, 6	1	3	20	11	17	0	1
2	1, 2, 7	2	2	16	5	7	1	2
3	1, 2, 8	1	1	25	20	-4	2	3
4	1, 2, 9	2	3	10	4	6	0	4
5	1, 2, 10	1	2	12	6	-1	1	5
6	1, 3, 6	2	1	18	14	-10	2	1
7	1, 3, 7	1	3	15	13	6	0	2
8	1, 3, 8	2	2	24	12	8	1	3
9	1, 3, 9	1	1	8	2	-2	2	4
10	1, 3, 10	2	3	14	8	10	0	5
11	1, 4, 6	1	2	26	20	-12	1	1
12	1, 4, 7	2	1	9	8	-5	2	2
13	1, 4, 8	1	3	19	17	14	0	3
14	1, 4, 9	2	2	13	9	3	1	4
15	1, 4, 10	1	1	30	15	-20	2	5
16	1, 5, 6	2	3	11	1	10	0	1
17	1, 5, 7	1	2	17	5	-9	1	2
18	1, 5, 8	2	1	21	20	-20	2	3
19	1, 5, 9	1	3	23	10	19	0	4
20	1, 5, 10	2	2	27	7	12	1	5

5. Содержание отчета

Отчет должен содержать:

1.Название работы.

2.Цель работы.

3.Задание и его решение.

4. Вывод по работе

Практическая работа №15
"Написание скриптов автоматизации с помощью Python»(4 часа)
Цель: Написание скриптов автоматизации с помощью Python

Набор Python-скриптов для автоматизации рутинных задач SEO-специалиста

Python — простой и лаконичный язык, позволяющий автоматизировать значительную часть рутинных задач и анализировать данные, что особенно актуально при работе с крупными проектами.

Одно из преимуществ языка — наличие большого количества написанных библиотек. Так как многие сервисы отдают свои данные по API, у специалистов есть возможность без глубоких знаний программирования писать скрипты для решения рабочих задач.

Что потребуется:

- Netpeak Spider.
- Базовые знания применения XPath.
- Установленные Python-библиотеки.

Плюсы подхода:

- Быстрое внедрение. Не требуется помощь программиста.
- Не нужно подключение к базе, где хранится контент. Весь контент и его разметку берём прямо со страниц (одновременно минус).
- Используем стандартные SEO-инструменты.

Минусы:

- Необновляемый XML-файл. После изменения контента требуется пересобрать контент и формировать новый XML.

- Новые страницы также не будут попадать в файл. Для них будет необходимо заново парсить контент и формировать XML.
- Создаём нагрузку на свой сайт при сборе контента.

Ниже описана последовательность работ.

Подготавливаем данные

С помощью Screaming frog seo spider или Netpeak Spider парсим контент страниц, для которых будем подключать турбо-страницы.

На этом этапе подготавливаем данные для обязательных элементов, необходимых при формировании XML-файла.

Обязательные поля:

- Link — URL страницы.
- H1 — заголовок страницы.
- Turbo:content — содержимое страницы.

Экспортируем полученные данные в CSV. В результате в CSV-файле должно быть три столбца:

- Link.
- H1.
- Turbo:content.

Скрипт генерации файла

Подключаем нужные библиотеки.

```
import csv

import pandas as pd

import os
```

```
import math
```

Считываем файл с подготовленными данными.

```
data = pd.read_csv('internal_all - internal_all.csv') #дописать ", header=1", если проблема при считывании заголовка

data = data[['Address', 'H1-1', 'текст 1']]

## Если в таблицу попали лишние страницы, их можно легко отфильтровать. Ниже примеры.

# data = data[data['Status Code']==200] # Фильтруем страницы с 200 ответом

# data = data[~data['Address'].str.contains('page')] # Фильтруем страницы не содержащие ""

# data = data.drop(index=0)
```

Выводим информацию о количестве строк в файле и итоговом количестве RSS-файлов, которые будут сгенерированы.

```
rows_in_rss = 1000 # количество строк в одном rss-канале

total_rows = len(data) - 1

total_xml_file = math.ceil((total_rows-1)/rows_in_rss)

print('Всего в файле строк:', total_rows)

print('Будет сгенерировано xml-файлов:', total_xml_file)
```

Формируем структуру RSS-канала. Создаём функцию create_xml, отвечающую за создание начала файла.

```
def create_xml(item_count_next):

    rss_file = open('rss{:.0f}.xml'.format(item_count_next/rows_in_rss), 'w', encoding="utf-8")

    rss_file.write(
```

```
    """<?xml version="1.0" encoding="UTF-8"?>
        <rss xmlns:yandex="http://news.yandex.ru"
            xmlns:media="http://search.yahoo.com/mrss/"
            xmlns:turbo="http://turbo.yandex.ru"
            version="2.0">
            <channel>""")
    rss_file.close()
```

Функция close_xml будет закрывать файл.

```
def close_xml(item_count_next):
    rss_file = open('rss{:.0f}.xml'.format(item_count_next/rows_in_rss), 'a',
encoding="utf-8")
    rss_file.write(
                '    </channel>' + '\n'+
                '</rss>'
                )
    rss_file.close()
```

В функцию data_for_rss передаём номер первой и последней строки. Для этого промежутка будем формировать RSS.

Построчно считываем строки в датафрейме и формируем <item>, записывая получившиеся данные в XML-файл. Каждая строка в датафрейме — новая страница.

```
def data_for_rss(item_count_prev, item_count_next):
        data_rss = data[item_count_prev:item_count_next]
        if len(data_rss) != 0:
```

```
            with open('rss{:.0f}.xml'.format(item_count_next/rows_in_rss),
'a', encoding="utf-8") as rss_file:

                for index, row in data_rss.iterrows():

                        url = str(row[0])

                         h1 = str(row[1])

            text = str(row[2])



            rss_file.write("""<item turbo="true">

                                <link>"""+ url + """</link>

                                <turbo:content>

                                <![CDATA[

                                   <header> <h1>"""+ h1 +"""</h1>

                                   </header>"""

                                   + text+

                                   """<div data-block="share" data-
network="vkontakte,odnoklassniki,facebook,twitter"></div>

                                   ]]>

                                   </turbo:content>

                            </item>""")
```

Делаем проверку размера получившихся фидов. Размер XML-файла не должен превышать 15 МБ. Если размер получился больше, изменяем количество строк в одном файле, изменяя значение переменной rows_in_rss.

```
def size_file(item_count_next):
```

```
        size_final_file_MB =
os.path.getsize('rss{:.0f}.xml'.format(item_count_next/rows_in_rss))/1024/
1024

        if size_final_file_MB < 15:

        print('Файл создан')

    else:       print('Нужно уменьшить шаг')
```

Финальный шаг — генерация фидов.

```
item_count_prev = 0

item_count_next = 1000 # должен быть равен rows_in_rss

count_rss = 0

print(total_xml_file)

while count_rss < total_xml_file:

        create_xml(item_count_next)

    data_list = data_for_rss(item_count_prev,item_count_next)

        close_xml(item_count_next)

        size_file(item_count_next)

        item_count_prev += rows_in_rss

        item_count_next += rows_in_rss

        count_rss += 1
```

5. Содержание отчета

Отчет должен содержать:
1.Название работы.
2.Цель работы.
3.Задание и его решение.
4. Вывод по работе

Практическая работа №16
"Систематизация информации. Формирование требований к безопасности" (2 часа)

Цель работы: изучить законодательный уровень информационной безопасности.

1. Теоретическое введение

Что такое законодательный уровень информационной безопасности?

В деле обеспечения информационной безопасности успех может принести только комплексный подход. Мы уже указывали, что для защиты интересов субъектов информационных отношений необходимо сочетать меры следующих уровней:

- законодательного;
- административного (приказы и другие действия руководства организаций, связанных с защищаемыми информационными системами);
- процедурного (меры безопасности, ориентированные на людей);
- программно-технического.

Законодательный уровень является важнейшим для обеспечения информационной безопасности. Большинство людей не совершают противоправных действий не потому, что это технически невозможно, а потому, что это осуждается и/или наказывается обществом, потому что так поступать не принято.

Мы будем различать на законодательном уровне две группы мер:

- меры, направленные на создание и поддержание в обществе негативного (в том числе с применением наказаний) отношения к нарушениям и нарушителям информационной безопасности (назовем их мерами ограничительной направленности);
- направляющие и координирующие меры, способствующие повышению образованности общества в области информационной безопасности, помогающие в разработке и распространении средств обеспечения информационной безопасности (меры созидательной направленности).

На практике обе группы мер важны в равной степени, но нам хотелось бы выделить аспект осознанного соблюдения норм и правил ИБ. Это важно для всех субъектов информационных отношений, поскольку рассчитывать только на защиту силами правоохранительных органов было бы наивно. Необходимо это и тем, в чьи обязанности входит наказывать нарушителей, поскольку обеспечить доказательность при расследовании и судебном разбирательстве компьютерных преступлений без специальной подготовки невозможно.

Самое важное (и, вероятно, самое трудное) на законодательном уровне - создать механизм, позволяющий согласовать процесс разработки законов с реалиями и прогрессом информационных технологий. Законы не могут опережать жизнь, но важно, чтобы отставание не было слишком большим, так как на практике, помимо прочих отрицательных моментов, это ведет к снижению информационной безопасности.

1.2. Обзор российского законодательства в области информационной безопасности

Основным законом Российской Федерации является Конституция, принятая 12 декабря 1993 года.

В соответствии со статьей 24 Конституции, органы государственной власти и органы местного самоуправления, их должностные лица обязаны обеспечить каждому возможность ознакомления с документами и материалами, непосредственно затрагивающими его права и свободы, если иное не предусмотрено законом.

Статья 41 гарантирует право на знание фактов и обстоятельств, создающих угрозу для жизни и здоровья людей, статья 42 - право на знание достоверной информации о состоянии окружающей среды.

В принципе, право на информацию может реализовываться средствами бумажных технологий, но в современных условиях наиболее практичным и удобным для граждан является создание соответствующими законодательными, исполнительными и судебными органами информационных серверов и поддержание

доступности и целостности представленных на них сведений, то есть обеспечение их (серверов) информационной безопасности.

Статья 23 Конституции гарантирует право на личную и семейную тайну, на тайну переписки, телефонных переговоров, почтовых, телеграфных и иных сообщений, статья 29 - право свободно искать, получать, передавать, производить и распространять информацию любым законным способом. Современная интерпретация этих положений включает обеспечение конфиденциальности данных, в том числе в процессе их передачи по компьютерным сетям, а также доступ к средствам защиты информации.

В Гражданском кодексе Российской Федерации (в своем изложении мы опираемся на редакцию от 15 мая 2001 года) фигурируют такие понятия, как банковская, коммерческая и служебная тайна. Согласно статье 139, информация составляет служебную или коммерческую тайну в случае, когда информация имеет действительную или потенциальную коммерческую ценность в силу неизвестности ее третьим лицам, к ней нет свободного доступа на законном основании, и обладатель информации принимает меры к охране ее конфиденциальности. Это подразумевает, как минимум, компетентность в вопросах ИБ и наличие доступных (и законных) средств обеспечения конфиденциальности.

Весьма продвинутым в плане информационной безопасности является Уголовный кодекс Российской Федерации (редакция от 14 марта 2002 года). Глава 28 - "Преступления в сфере компьютерной информации" - содержит три статьи:

- статья 272. Неправомерный доступ к компьютерной информации;

- статья 273. Создание, использование и распространение вредоносных программ для ЭВМ;

- статья 274. Нарушение правил эксплуатации ЭВМ, системы ЭВМ или их сети.

Первая имеет дело с посягательствами на конфиденциальность, вторая - с вредоносным ПО, третья - с нарушениями доступности и целостности, повлекшими за собой уничтожение, блокирование или модификацию охраняемой законом информации ЭВМ. Включение в сферу действия УК РФ вопросов доступности информационных сервисов представляется нам очень своевременным.

Статья 138 УК РФ, защищая конфиденциальность персональных данных, предусматривает наказание за нарушение тайны переписки, телефонных переговоров, почтовых, телеграфных или иных сообщений. Аналогичную роль для банковской и коммерческой тайны играет статья 183 УК РФ.

Интересы государства в плане обеспечения конфиденциальности информации нашли наиболее полное выражение в Законе "О государственной тайне" (с изменениями и дополнениями от 6 октября 1997 года). В нем гостайна определена как защищаемые государством сведения в области его военной, внешнеполитической, экономической, разведывательной, контрразведывательной и оперативно-розыскной деятельности, распространение которых может нанести ущерб безопасности Российской Федерации. Там же дается определение средств защиты информации. Согласно данному Закону, это технические, криптографические, программные и другие средства, предназначенные для защиты сведений, составляющих государственную тайну.

Первая имеет дело с посягательствами на конфиденциальность, вторая - с вредоносным ПО, третья - с нарушениями доступности и целостности, повлекшими за собой уничтожение, блокирование или модификацию охраняемой законом информации ЭВМ. Включение в сферу действия УК РФ вопросов доступности информационных сервисов представляется нам очень своевременным.

Статья 138 УК РФ, защищая конфиденциальность персональных данных, предусматривает наказание за нарушение тайны переписки, телефонных переговоров, почтовых, телеграфных или иных сообщений. Аналогичную роль для банковской и коммерческой тайны играет статья 183 УК РФ.

Интересы государства в плане обеспечения конфиденциальности информации нашли наиболее полное выражение в Законе "О государственной тайне" (с изменениями и дополнениями от 6 октября 1997 года). В нем гостайна определена как защищаемые государством сведения в области его военной, внешнеполитической, экономической, разведывательной, контрразведывательной и оперативно-розыскной деятельности, распространение которых может нанести ущерб безопасности Российской Федерации. Там же дается определение средств защиты информации. Согласно данному Закону, это технические, криптографические, программные и другие средства, предназначенные для защиты сведений, составляющих государственную тайну; средства, в которых они реализованы, а также средства контроля эффективности защиты информации. Подчеркнем важность последней части определения.

Закон "Об информации, информатизации и защите информации"

Основополагающим среди российских законов, посвященных вопросам информационной безопасности, следует считать закон "Об информации, информатизации и защите информации" от 20 февраля 1995 года номер 24-ФЗ (принят Государственной Думой 25 января 1995 года). В нем даются основные определения и намечаются направления развития законодательства в данной области.

Процитируем некоторые из этих определений:

- информация - сведения о лицах, предметах, фактах, событиях, явлениях и процессах независимо от формы их представления;

- документированная информация (документ) - зафиксированная на материальном носителе информация с реквизитами, позволяющими ее идентифицировать;

- информационные процессы - процессы сбора, обработки, накопления, хранения, поиска и распространения информации;

- информационная система - организационно упорядоченная совокупность документов (массивов документов) и информационных технологий, в том числе с использованием средств вычислительной техники и связи, реализующих информационные процессы;

- информационные ресурсы - отдельные документы и отдельные массивы документов, документы и массивы документов в информационных системах (библиотеках, архивах, фондах, банках данных, других информационных системах);

- информация о гражданах (персональные данные) - сведения о фактах, событиях и обстоятельствах жизни гражданина, позволяющие идентифицировать его личность;

- конфиденциальная информация - документированная информация, доступ к которой ограничивается в соответствии с законодательством Российской Федерации;

- пользователь (потребитель) информации - субъект, обращающийся к информационной системе или посреднику за получением необходимой ему информации и пользующийся ею.

Мы, разумеется, не будем обсуждать качество данных в Законе определений. Обратим лишь внимание на гибкость определения конфиденциальной информации, которая не сводится к сведениям, составляющим государственную тайну, а также на понятие персональных данных, закладывающее основу защиты последних.

Закон выделяет следующие цели защиты информации:

- предотвращение утечки, хищения, утраты, искажения, подделки информации;

- предотвращение угроз безопасности личности, общества, государства;

- предотвращение несанкционированных действий по уничтожению, модификации, искажению, копированию, блокированию информации;

- предотвращение других форм незаконного вмешательства в информационные ресурсы и информационные системы, обеспечение правового режима документированной информации как объекта собственности;

- защита конституционных прав граждан на сохранение личной тайны и конфиденциальности персональных данных, имеющихся в информационных системах;

- сохранение государственной тайны, конфиденциальности документированной информации в соответствии с законодательством;

- обеспечение прав субъектов в информационных процессах и при разработке, производстве и применении информационных систем, технологий и средств их обеспечения.

Отметим, что Закон на первое место ставит сохранение конфиденциальности информации. Целостность представлена также достаточно полно, хотя и на втором месте. О доступности ("предотвращение несанкционированных действий по ... блокированию информации") сказано довольно мало.

Продолжим цитирование:

"Защите подлежит любая документированная информация, неправомерное обращение с которой может нанести ущерб ее собственнику, владельцу, пользователю и иному лицу".

По сути, это положение констатирует, что защита информации направлена на обеспечение интересов субъектов информационных отношений.

Далее. "Режим защиты информации устанавливается:

- в отношении сведений, отнесенных к государственной тайне, - уполномоченными органами на основании Закона Российской Федерации "О государственной тайне";

- в отношении конфиденциальной документированной информации - собственником информационных ресурсов или уполномоченным лицом на основании настоящего Федерального закона;

- в отношении персональных данных - федеральным законом."

Здесь явно выделены три вида защищаемой информации, ко второму из которых принадлежит, в частности, коммерческая информация. Поскольку защите подлежит только документированная информация, необходимым условием является фиксация

коммерческой информации на материальном носителе и снабжение ее реквизитами. Отметим, что в данном месте Закона речь идет только о конфиденциальности; остальные аспекты ИБ забыты.

Обратим внимание, что защиту государственной тайны и персональных данных берет на себя государство; за другую конфиденциальную информацию отвечают ее собственники.

Как же защищать информацию? В качестве основного закон предлагает для этой цели мощные универсальные средства: лицензирование и сертификацию. Процитируем статью 19.

Информационные системы, базы и банки данных, предназначенные для информационного обслуживания граждан и организаций, подлежат сертификации в порядке, установленном Законом Российской Федерации "О сертификации продукции и услуг".

Информационные системы органов государственной власти Российской Федерации и органов государственной власти субъектов Российской Федерации, других государственных органов, организаций, которые обрабатывают документированную информацию с ограниченным доступом, а также средства защиты этих систем подлежат обязательной сертификации. Порядок сертификации определяется законодательством Российской Федерации.

Организации, выполняющие работы в области проектирования, производства средств защиты информации и обработки персональных данных, получают лицензии на этот вид деятельности. Порядок лицензирования определяется законодательством Российской Федерации.

Интересы потребителя информации при использовании импортной продукции в информационных системах защищаются таможенными органами Российской Федерации на основе международной системы сертификации.

Здесь трудно удержаться от риторического вопроса: а есть ли в России информационные системы без импортной продукции? Получается, что на защите интересов потребителей стоит в данном случае только таможня...

И еще несколько пунктов, теперь из статьи 22:

Владелец документов, массива документов, информационных систем обеспечивает уровень защиты информации в соответствии с законодательством Российской Федерации.

Риск, связанный с использованием несертифицированных информационных систем и средств их обеспечения, лежит на собственнике (владельце) этих систем и средств. Риск, связанный с использованием информации, полученной из несертифицированной системы, лежит на потребителе информации.

Собственник документов, массива документов, информационных систем может обращаться в организации, осуществляющие сертификацию средств защиты информационных систем и информационных ресурсов, для проведения анализа достаточности мер защиты его ресурсов и систем и получения консультаций.

Владелец документов, массива документов, информационных систем обязан оповещать собственника информационных ресурсов и (или) информационных систем о всех фактах нарушения режима защиты информации.

Из пункта 5 следует, что должны обнаруживаться все (успешные) атаки на ИС. Вспомним в этой связи один из результатов опроса (см. лекцию 1): около трети респондентов-американцев не знали, были ли взломаны их ИС за последние 12 месяцев. По нашему законодательству их можно было бы привлечь к ответственности...

Далее, статья 23 "Защита прав субъектов в сфере информационных процессов и информатизации" содержит следующий пункт:

Защита прав субъектов в указанной сфере осуществляется судом, арбитражным судом, третейским судом с учетом специфики правонарушений и нанесенного ущерба. Очень важными являются пункты статьи 5, касающиеся юридической силы электронного документа и электронной цифровой подписи:

Юридическая сила документа, хранимого, обрабатываемого и передаваемого с помощью автоматизированных информационных и телекоммуникационных систем, может подтверждаться электронной цифровой подписью.

Юридическая сила электронной цифровой подписи признается при наличии в автоматизированной информационной системе программно-технических средств, обеспечивающих идентификацию подписи, и соблюдении установленного режима их использования.

Право удостоверять идентичность электронной цифровой подписи осуществляется на основании лицензии. Порядок выдачи лицензий определяется законодательством Российской Федерации.

Таким образом, Закон предлагает действенное средство контроля целостности и решения проблемы "неотказуемости" (невозможности отказаться от собственной подписи).

Таковы важнейшие, на наш взгляд, положения Закона "Об информации, информатизации и защите информации". На следующей странице будут рассмотрены другие законы РФ в области информационной безопасности.

Другие законы и нормативные акты

Следуя логике Закона "Об информации, информатизации и защите информации", мы продолжим наш обзор Законом "О лицензировании отдельных видов деятельности" от 8 августа 2001 года номер 128-ФЗ (Принят Государственной Думой 13 июля 2001 года). Начнем с основных определений.

- "**Лицензия** - специальное разрешение на осуществление конкретного вида деятельности при обязательном соблюдении лицензионных требований и условий, выданное лицензирующим органом юридическому лицу или индивидуальному предпринимателю.

- **Лицензируемый вид деятельности** - вид деятельности, на осуществление которого на территории Российской Федерации требуется получение лицензии в соответствии с настоящим Федеральным законом.

- **Лицензирование** - мероприятия, связанные с предоставлением лицензий, переоформлением документов, подтверждающих наличие лицензий, приостановлением и возобновлением действия лицензий, аннулированием лицензий и контролем лицензирующих органов за

соблюдением лицензиатами при осуществлении лицензируемых видов деятельности соответствующих лицензионных требований и условий.

- **Лицензирующие органы** - федеральные органы исполнительной власти, органы исполнительной власти субъектов Российской Федерации, осуществляющие лицензирование в соответствии с настоящим Федеральным законом.

- **Лицензиат** - юридическое лицо или индивидуальный предприниматель, имеющие лицензию на осуществление конкретного вида деятельности."

Статья 17 Закона устанавливает перечень видов деятельности, на осуществление которых требуются лицензии. Нас будут интересовать следующие виды:

- распространение шифровальных (криптографических) средств;

- техническое обслуживание шифровальных (криптографических) средств;

- предоставление услуг в области шифрования информации;

- разработка и производство шифровальных (криптографических) средств, защищенных с использованием шифровальных (криптографических) средств информационных систем, телекоммуникационных систем;

- выдача сертификатов ключей электронных цифровых подписей, регистрация владельцев электронных цифровых подписей, оказание услуг, связанных с использованием электронных цифровых подписей и подтверждением подлинности электронных цифровых подписей;

- выявление электронных устройств, предназначенных для негласного получения информации, в помещениях и технических средствах (за исключением случая, если указанная деятельность осуществляется для обеспечения собственных нужд юридического лица или индивидуального предпринимателя);

- разработка и (или) производство средств защиты конфиденциальной информации;

- техническая защита конфиденциальной информации;

- разработка, производство, реализация и приобретение в целях продажи специальных технических средств, предназначенных для негласного получения информации, индивидуальными предпринимателями и юридическими лицами, осуществляющими предпринимательскую деятельность.

Необходимо учитывать, что, согласно статье 1, действие данного Закона не распространяется на следующие виды деятельности:

- деятельность, связанная с защитой государственной тайны;
- деятельность в области связи;
- образовательная деятельность.

Подчеркнем в этой связи, что данный Закон не препятствует организации Интернет-Университетом учебных курсов по информационной безопасности (не требует получения специальной лицензии; ранее подобная лицензия была необходима). В свою очередь, Федеральный Закон "Об образовании" не содержит каких-либо специальных положений, касающихся образовательной деятельности в области ИБ.

2. Порядок выполнения работы

1.Ознакомиться с российским зарубежным законодательством в области ИБ..
2.Выполнить практическое задание.
3.Ответить на контрольные вопросы.

3. Практические задания

1. Разработать интерфейс пользователя «Обзор российского законодательства в области информационной безопасности».
2. Разработать интерфейс пользователя «Обзор зарубежного законодательства в области информационной безопасности».

4. Контрольные вопросы

1. Уголовный кодекс РФ не предусматривает наказания за:
2. Согласно Закону "Об информации, информатизации и защите информации", персональные данные — это:

3. Согласно Закону "О лицензировании отдельных видов деятельности", лицензия — это:
4. В законопроекте "О совершенствовании информационной безопасности" (США, 2001 год) особое внимание обращено на:

5. Содержание отчета

Отчет должен содержать:
1.Название работы.
2.Цель работы.
3.Задание и его решение.
4. Вывод по работе

Практическая работа № 17
Написать аннотации к событиям и новостям 1(час)
Цель работы: освоить основные приѐмы работы в социальных сетях.

Общие сведения

Социальные сети стремительно вошли в современную жизнь. Миллионы людей в разных странах зарегистрированы в социальных сетях и активно их используют. Число существующих социальных сетей велико, счѐт идѐт на десятки и даже сотни единиц.

Человечество представляет собой группы людей, связанных различными отношениями и интересами. Интернет предоставил для общения отдельным группам людей новые технические возможности. Отсутствие у современного человека доступа в социальную сеть является фактором, ограничивающим получение актуальной информации. Среди множества социальных сетей можно выделить насколько наиболее популярных в России: Facebook, Twitter, Одноклассники, ВКонтакте.

Социальные сети используются людьми для достижения различных целей: продвижение политических и религиозных идей, творческого само-выражения (стихи, проза, живопись, фотографии, музыка, видеоклипы), поиск информации, деловое сотрудничество, оперативное общение с друзьями, родственниками, сокурсниками,

коллегами и единомышленниками, расслабляющий досуг (игры, просмотр занимательных страниц).

Рассмотрим основные термины, используемые в этой работе.

Социальный — относящийся к жизни людей и их отношениям в обществе. Синоним для прилагательного **социальный** — общественный.

Сеть — это элементы и связи между ними. В контексте данной работы значение этого термина дуально (двойственно). С одной стороны, сеть – это люди и их общественные связи. С другой стороны, сеть — это система связи, состоящая из компьютеров (телефонов), объединённых каналами связи.

Социальная сеть — группы людей со сходными интересами, объединённые с помощью современных информационных технологий.

Аватар — визуальный образ (небольшая фотография или рисунок).

Аккаунт — учётная запись, содержащая сведения о пользователе.

Блог — сайт, на котором расположен дневник событий (новостей), по-
валяющий вести обсуждение контента (содержимого) с другими пользователями.

Блоггер – пользователь, ведущий (поддерживающий) свой блог.

Никнейм (англ. прозвище, кличка) — псевдоним человека, используемый им при входе на сайт.

Пост — сообщение, адресованное к аудитории страницы.

Профиль — основные сведения о пользователе, которые он сообщает о себе.

Твит — короткое сообщение (длиной до 140 символов), на которое есть гиперссылка.

Хэштег — метка, начинающаяся со знака «#», предназначенная для группировки сообщений одной тематики**.**

Дадим краткую характеристику некоторым социальным сетям.

Facebook (**«Фейсбук»**) — в настоящее время самая крупная социальная сеть в мире. Сеть основана в 2004 году Марком Цукербергом и его друзьями — Э. Саверином, Д. Московицем и К. Хьюзом. Сеть имеет более миллиарда зарегистрированных пользователей во всём мире.

«**Стена»** — это место на странице пользователя, на которой он и его друзья могут размещать новости, фотографии и различные публикации. Также к записям на «стене» можно писать комментарии.

Задания на выполнение лабораторной работы

Зарегистрироваться в социальной сети «Facebook»

Зарегистрироваться в сети Facebook, указав достоверную информацию о себе. Добавить на страницу рисунок в соответствии с вариантом (табл. 3.1). Рисунок необходимо скачать из Интернета. В отчете описать процедуру регистрации, а также поместить скриншот созданного профиля.

Вариант	Рисунок	Вариант	Рисунок
1	Кошка	9	Сорока
2	Голубь	10	Попугай
3	Белка	11	Лиса
4	Дельфин	12	Окунь
5	Собака	13	Черепаха
6	Заяц	14	Бабочка
7	Волк	15	Тигр

Добавить фотографии

Создать фотоальбом, выбрав название альбома в соответствии с вариантом (Приложение 2). Фотографию обложки подобрать с учѐтом тематики альбома. Поместить в альбом три фотографии по заданной тематике. В отчете описать порядок создания альбома и добавления фотографий, а также поместить скриншот созданного альбома.

Запланировать мероприятия

Организовать мероприятие в соответствии с вариантом (Приложение 1). В отчете описать порядок действий и поместить скриншот запланированного мероприятия.

Задание 2. Зарегистрироваться в сети «Google+»

Опубликовать сведения о пользователе

В окне ―Начало работы‖ заполнить данные о себе, добавить фотографию, указать место учебы и место проживания.

Оформить Ленту

На **Ленте** указать полное, фамилию, имя и группу, прикрепить фотографию с сайта, указанного преподавателем, в соответствии с вариантом (Приложение 2), указать источник (ссылку).

Присоединиться к сообществу

Необходимо присоединиться к сообществу, указанному преподавателем

Задание 3. Зарегистрироваться в социальной сети «Мой мир»

Создать фотоальбом и добавить фотографии

Используя функции сети «Мой мир», создать фотоальбом и поместить в него рисунок из папки, указанной преподавателем.

Название фотоальбома должно включать в себя фамилию студента. Название рисунка должно содержать в себя номер варианта и название персонажа, изображенного на рисунке.

Добавить уведомление в раздел «Праздники»

Используя функции проекта «Мой мир», создать уведомление о празднике. Праздник выбрать в соответствии с вариантом (см. Приложение 1)

Задание 4. Зарегистрироваться в социальной сети «Привет.ру»

Поместить запись в блоге

В соответствии с вариантом, разместить в своем блоге запись, содержащую афоризм (см. таблицу 3.4.1.), дополненный рисунком. Рисунок находится в контейнере, который соответствует номеру варианта. Отформатировать афоризм: гарнитура – Times New Roman, кегль 10. Указать заголовок записи – Задание 1. Тег записи – цитаты, Задание. Категория записи – Наука.

Вар.	**Афоризмы**
1	Ум — это хлеб, который насыщает; шутка — это специя, Вызывающая аппетит *Л. Берне*
2	Умный человек не только никогда не скажет ничего глупого, но даже никогда и не услышит ничего глупого *Л. Берне*
3	Немногие умы гибнут от износа, большей частью они

	ржавеют от неупотребления *П. Бомарше*
4	Орлиный взгляд страстей проникает в туманную пропасть грядущего, равнодушие же слепо и тупо от рождения. *Гельвеций*
5	Глупые мысли бывают у всякого, только умный их не высказывает *В. Буш*
6	Главное преимущество ума — не в живости, а в точности, как и достоинство часов – не в быстроте, а в точности хода *Л. Вовенарг*
7	Можно уступить силе, но безропотно покоряются только разуму. *Л. Бланки*
8	Радость видеть и понимать есть самый прекрасный дар пр и-роды. *А. Эйнштейн*
9	Разумный гонится не за тем, что приятно, а за тем, что избавляет от неприятностей *Аристотель*
10	Ум есть драгоценный камень, который более красиво играет в оправе скромности *М. Горький*

Сменить стандартный фон страницы

Используя настройку страницы, сменить стандартный фон страницы на фотографию. Фотографию скачать с сайта, адрес которого укажет преподаватель.

Задание 5. Зарегистрироваться в сети «ФотоКто

Найти пользователя в базе в соответствии с критерием.

Вар.	Критерий	Жанр
1	Фотолюбитель из Израиля	Архитектура. Интерьер
2	Флорист из Финляндии	Детский мир
3	Дизайнер из США	Животные
4	Модель из России	Мобильное фото
5	Фотограф из Бразилии	Натюрморт
6	Ассистент из Эстонии	Пейзаж

7	Визажист из России	Портрет
8	Оформитель из Чехии	Повседневная жизнь
9	Ретушѐр из Великобритании	Природа
10	Фотолюбитель из Греции	Спорт
11	Модель из Молдовы	Техника
12	Ассистент из Колумбии	Путешествия
13	Визажист из Израиля	Макро
14	Стилист из Франции	Жанровый портрет
15	Ретушѐр из Латвии	Животные
16	Оформитель из Египта	Пейзаж

Задание 6. Зарегистрироваться в сети «ВКонтакте»

Загрузить фотографии

На своей странице, разместить пять фотографий, импортированных с сайта samara.psati.ru. Тематика фотографий выбирается в соответствии с вариантом

Выполнить настройки страницы

Сделать свою страницу закрытой «от действий случайного пользователя

Задание 7. Зарегистрироваться в социальной сети «Twitter»

Создать список из пяти тем

Добавить страницу в список «читаемое»

Добавить в свой список «читаемое» новую страницу

Таблица

Вар.	**Страница**	**Адрес**
1	Официальная страница Twitter для потребительских продуктов корпорации Майкрософт	https://twitter.com/Microsoft
2	Официальная страница Twitter Samsung	https://twitter.com/Samsungtweets
3	Официальная страница Windows Azure.	https://twitter.com/WindowsAzure
4	Официальная страница Chris	https://twitter.com/cdixon

	Dixon	
5	Глава Webspam команды Google.	https://twitter.com/mattcutts
6	Генеральный директор Apple	https://twitter.com/tim_cook
7	Официальная страница Билла Гейтса	https://twitter.com/BillGates
8	Официальная страница Эрика Шмидта	https://twitter.com/ericschmidt
9	Официальная страница Рида Харпера	https://twitter.com/harper
10	Официальная страница Google	https://twitter.com/google
11	Официальная twitter-страница корпорации Intel в России	https://twitter.com/IntelRussia
12	Официальная страница россий-ского представительства Microsoft.	https://twitter.com/RusMicrosoft
13	Официальная страница Twitter для журнала LinuxFormat	https://twitter.com/linuxformat
14	Профессиональная соцсеть для IT-специалистов	https://twitter.com/rootFront
15	IT технологии, новости интер-нета, новинки техники и элек-троники.	https://twitter.com/it_net_news
16	ИТ Новости со всего мира	https://twitter.com/ICTnws

Составить сообщение для студентов Вашей группы Подготовить приветствие, в котором указать свою фамилию. При помощи хэштега указать номер своей студенческой группы.

Задание 8. Зарегистрироваться в социальной сети «LiveLib»

Поместить в список три прочитанные книги

Поместите в список три художественные книги, прочитанные Вами за последнее время.

Найти рецензию на книгу

Найти одну рецензию на книгу, недавно прочитанную Вами.

Требования к отчѐту

Отчѐт подготавливается в электронном виде (MS Word). Он должен содержать постановки задач, скриншоты, которые показывают порядок ре-шения задач, необходимые комментарии.

6. Контрольные вопросы

1.Что называется социальной сетью?
2.Перечислите социальные сети, о которых Вы слышали.
3.Что такое блог (микроблог)?
4.Кто такой блоггер?
5.Какие технические средства и сервисы Интернет были предшественниками социальных сетей?
6.Что такое аватар?
7.Каково ограничение на длину сообщений в твиттере?

5. Содержание отчета

Отчет должен содержать:
1.Название работы.
2.Цель работы.
3.Задание и его решение.
4. Вывод по работе

Практическая работа № 18

Работа с агрегаторами новостей, электронными подписками, социальными сетями, форумами

(1 час)

Цель работы: выработать практические навыки работы с форумами, регистрации, настройки и работы в системах

2. Оборудование, приборы, аппаратура, материалы: персональный компьютер с выходом в Интернет, инсталляторы программ Skype, ICQ, интернет-браузер.

3. Краткие теоретические сведения.

Форум - это тематическое общение. В отличие от чата, на форуме обсуждают какую-то определенную тему. Можно сказать, что форум - это клуб по интересам. То есть форум - это такое место в Интернете, где собираются люди, которых объединяет одно увлечение или идея, и общаются на интересующую их тему. Они помогают друг другу советами и подсказками, обмениваются жизненным опытом, поддерживают друг друга.

Для того чтобы найти форум на интересующую тему, можно воспользоваться поисковой системой. Например, открыть сайт yandex.ru и напечатать в строке поиска «форум интересующая тема». Например, «форум кошки».

Для общения в системе мгновенных сообщений ICQ каждому пользователю необходимо иметь специальный идентификационный номер, называемый ICQ UIN.

ICQ - служба передачи мгновенных сообщений в Интернете. Регистрация в системе ICQ

1. Перейдите на страницу http://www.icq.com/join/ru
2. Перейдя на страницу регистрации ICQ, вы увидите стандартные поля, которые вы должны будете заполнить и после нажать кнопку Регистрация. Для успешной регистрации заполнять придётся все поля. Рекомендуем обращать внимание на всплывающие подсказки справа - они достаточно полезны при возникновении трудностей.
   - имя, Фамилия - до 20 символов в каждое поле;
   - адрес электронной почты может быть использован для входа в систему или восстановления забытого пароля;
   - Пароль - у большинства при регистрации возникают проблемы с его выбором. Происходит это из-за того, что сервис ICQ установил некие рамки для вводимого пароля - он не может быть короче 6 и длиннее 8 символов включительно. Он может состоять из заглавных и строчных латинских букв и цифр;
   - Дата рождения - эта информация необходима для большей безопасности вашего ICQ UIN, она будет доступна только вашим друзьям (изменить это правило можно в настройках приватности ICQ);
   - Пол;
   - Защита от роботов - 5-6 цифр, обычно раза с 2-3 получается распознать их. о Заполнив все поля, нажмите кнопку Регистрация.
3. Если все поля были заполнены верно, вы увидите страницу, на которой написано, что для завершения процесса регистрации номера аськи нужно нажать на ссылку в письме и чуть ниже кнопку для перехода в свой почтовый ящик - жмите её.
4. В своей почте во Входящих должно появиться новое письмо от ICQ Support, откройте его и нажмите ссылку в этом письме. Обычно оно приходит в течение 10 минут. Если письмо так и нет во Входящих, поищите его во вкладке Спам.

5. Итак, вы перешли по ссылке, подтвердив тем самым регистрацию и теперь видите страницу, на которой вас информируют о том, что вы успешно зарегистрировались в ICQ.
6. Для того, чтобы узнать какой номер UIN вами зарегистрирован, нужно нажать Скачать в верхнем меню сайта и на открывшейся странице в правом верхнем углу вы увидите свою фамилию и имя. Кликнув по этой надписи и вы увидите какой ICQ номер вы только что зарегистрировали.
7. После успешной регистрации, чтобы пользоваться новым ICQ номером, вам необходимо скачать бесплатную версию ICQ.

Skype - программное обеспечение с закрытым кодом, обеспечивающее шифрованную голосовую связь и видеосвязь через Интернет между компьютерами а также платные услуги для звонков на мобильные и стационарные телефоны.

Программа также позволяет совершать конференц-звонки (до 25 голосовых абонентов, включая инициатора), видеозвонки (в том числе видеоконференции до 10 абонентов), а также обеспечивает передачу текстовых сообщений (чат) и передачу файлов. Есть возможность вместо изображения с веб-камеры передавать изображение с экрана монитора Регистрация в скайп:

1. Для начала вам необходимо скачать программу Скайп. После того как программа загрузилась, нажмите на файл установки «SkypeSetup».
2. Далее после распаковки должно открыться окно, в котором надо выбрать русский язык и нажать на кнопку «Я согласен - установить».
3. Дожидаемся конца установки.
4. В открывшимся окне, предварительно проверив соединение с интернетом, нажмите на надпись «У вас нет логина?».
5. Далее появится окно, в котором и произойдет регистрация Скайп. Вам необходимо заполнить все поля (Имя, пароль, электронная почта, а также надо будет придумать уникальный логин) и нажать на кнопку «Я согласен (-на). Создать учетную запись».

6. В появившимся окне вводим свой логин и пароль, который указали при регистрации.

Настройка Скайпа - основные настройки Скайпа включают в себя настройку аудио параметров (микрофон и наушники) и видео (веб-камера). Обычно пользователям самостоятельно не приходиться в ручную настраивать Скайп, все необходимые настройки происходят автоматически. Но, не стандартный, старый и слабый микрофон или наушники могут потребовать вашего вмешательства. Для начала попробуйте тестовый звонок, он совершенно бесплатен. Вам предложить прослушать сообщение что бы оценить качества звука через наушники или колонки, после этого Вам предложат оставить свое голосовое сообщение, которое Вы же потом и прослушаете. Это позволяет оценить качество работы вашего микрофона и качество передачи звука через интернет. Если есть проблемы с качеством звука или качеством интернет-соединения, то обычно Скайп сам вам об этом сообщит после тестового звонка и предложит пути решения проблемы.

Если все ж вас не устроило качество, то имеет смысл попытаться отключить автоматическую настройку микрофона и вручную установить уровень звука Настройка камеры в Скайпе

Если камера уже работала до Скайпа, то проблем обычно не возникает, Скайп сам корректно найдет и настроит веб-камеру. Если веб-камера подключается в первые, то следует подключить камеру, а после установить драйвера с диска который шел с камерой.

4. Задание

Задание 1. Найти с помощью одной из поисковых систем Интернета форумы по следующим темам:

- Компьютеры
- Информатика
- Информационные технологии на транспорте
- Информационные технологии железнодорожников и т.п. Зарегистрироваться на форуме. Предложить на форуме обсуждение интересующего вас вопроса по теме форума. Сохранить скрин окна форума в текстовом документе.

Задание 2. Зарегистрироваться в системе ICQ, настроить систему, найти в системе троих одногруппников, передать им текстовые сообщения.

Задание ***3.*** Зарегистрироваться в системе Scype, настроить систему, найти в системе трех одногруппников. Добавить их свои Контакты. Осуществить видеозвонок одному из них. Выполнить видео-сессию с тремя одногруппниками одновременно.

5. Содержание отчета

Отчет должен содержать:

1. Название работы.
2. Цель работы.
3. Описание порядка регистрации на форумах, в ISQ, в scype.
4. Вывод по работе.

6. Контрольные вопросы

1. Какие формы общения в реальном времени существуют в Интернете?
2. Порядок регистрации в ICQ.
3. Как добавить пользователя в ICQ?
4. Как установить статус в ICQ?
5. Порядок регистрации в Scype.
6. Как осуществить настройку web-камеры в Scype?
7. Как добавить пользователя в Scype?

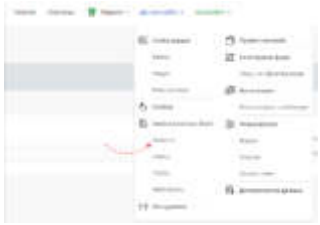
Практическая работа № 19

Технологии организации и ведения новостных лент, RSS-каналов, электронных подписок, рассылок по электронной почте (1 час)

Цель: Создание ленты новостей

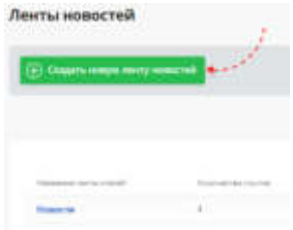
Шаг 1

Раскройте список "Данные сайта" и нажмите на пункт "Новости".



Шаг2

Откроется список всех созданных лент. Нажмите "Создать новую ленту новостей".



Шаг 3

Откроется окно добавления новостной ленты.

- Название ленты новостей - название новостной ленты не отображается на сайте и используется для вашего удобства при работе с новостными лентами через раздел "Данные сайта". На сайте отображается то название, которое вы указали при создании страницы новостей.
- Вид отображения ленты новостей - Укажите наиболее предпочтительный способ отображения новостей в ленте.
    - Выберите "В ленте отображается только анонс новости", если желаете, чтобы в ленте новостей отображалось лишь краткое описание новости;
    - Выберите "Новости отображаются целиком на ленте", если желаете, чтобы в ленте новостей пользователи могли ознакомиться со всем текстом, не переходя на страницу новости.
    - Выберите "Показывать только ленту новостей", если желаете, чтобы на странице ленты новостей должны отображались заголовки и тексты новостей целиком, без возможности перейти к странице конкретной новости. Иными словами, в таком режиме новости будут выводиться одной монолитной лентой, с полными текстами всех новостей.
- Способ сортировки статей в ленте - доступна сортировка только по дате, по убыванию (от новых к старым) и по возрастанию (от старых к новым).
- Количество новостей на одной странице - какое число статей должно отображаться на одной странице новостей (от 1 до 100).
- Формат изображений в новостных публикациях - все иллюстрации, прикрепленные к новостям, будут отображаться в указанном разрешении.
- Отключить индексацию всех новостей - если вы хотите, чтобы все новости перестали индексироваться на сайте, выберите "Да" (на собственные настройки новостей это никак не повлияет - они все просто не будут индексироваться и попадать в карту сайта). Если же вам не нужно полное отключение индексации содержимого - оставьте в настройке значение "Нет" (кроме того, для отключения индексации вы можете воспользоваться отдельным инструментом на SEO-панели).

В конце не забудьте нажать "Сохранить и закрыть".

5. Содержание отчета

Отчет должен содержать:
1.Название работы.
2.Цель работы.
3.Задание и его решение.
4. Вывод по работе

Практическая работа № 20
Ведение диалога и управление дискуссиями с участием большого числа активных пользователей (1 час)

Цель:

- Ознакомление с аппаратным обеспечение локальной компьютерной сети;
- Получение навыков работы в локальной компьютерной сети;
- Научиться устанавливать права доступа к сетевым ресурсам, работать с информацией, расположенной на компьютерах локальной сети.

Оборудование и ПО: ПК, ОС Windows 10

Теоретический материал:
Компьютерная сеть – объединение двух или более компьютеров с помощью телекоммуникационного оборудования.
Для создания компьютерной сети необходимо специальное аппаратное и программное обеспечение.

Назначение компьютерных сетей – обеспечение совместного доступа к общим ресурсам (аппаратным, программным и информационным).
Под аппаратным ресурсом понимается устройства ПК, которые можно использовать совместно, например принтер, жёсткий диск и т.д.
Программный ресурс – совокупность программного обеспечения, которое можно использовать совместно.
Информационный ресурс – совокупность данных, хранящихся на удалённых компьютерах, доступ к которым возможен пользователям сети.
Условно компьютерную сеть разделяют на локальную и глобальную.

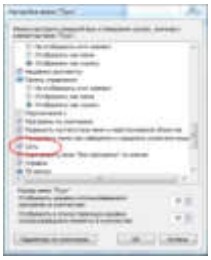
Разделение ведётся по территориальному признаку локальные сети более компактные. Это могут быть сети одного помещения, здания. Глобальные сети могут объединять как отдельные компьютеры, так и локальные сети.
Для того обеспечения совместимости компьютеров в сети действуют специальные стандарты, которые называются протоколами.
Протоколы бывают аппаратные (определяют взаимодействие компонентов сети) и программные (определяют взаимодействие программ и данных).
В локальных сетях обычно используется одинаковый набор протоколов, в глобальных – разный.
Основные понятия локальных компьютерных сетей
Локальные компьютерные сети бывают двух типов:

- одноранговые;
- с выделенным сервером.

Одноранговые сети – сети, в которых все компьютеры имеют равные права.
В сетях с выделенным файловым сервером один из компьютеров предназначен для совместной эксплуатации другими участниками локальной сети.
Группа участников, работающих над одним проектом, называется рабочей группой. Участники одной рабочей группы могут иметь разные права доступа к ресурсам сети. В одной локальной сети может быть несколько рабочих групп.
Совокупность приёмов разделения и ограничения прав участников компьютерной сети называется политикой сети.
Управление сетевыми политиками называется администрированием сети. Человек, который отвечает за организацию работы в локальной сети каждого участника называется системным администратором.
IP адрес - уникальный сетевой адрес узла в компьютерной сети, построенной по протоколу IP. В сети Интернет требуется глобальная уникальность адреса; в случае работы в локальной сети требуется уникальность адреса в пределах сети.
IP - адрес записывается в виде четырёх десятичных чисел (от 0 до 255), разделённых точками, например, 192.168.0.1 или 10.32.123.46. Когда речь идет о сетевом адресе, обычно имеется в виду IP – адрес.

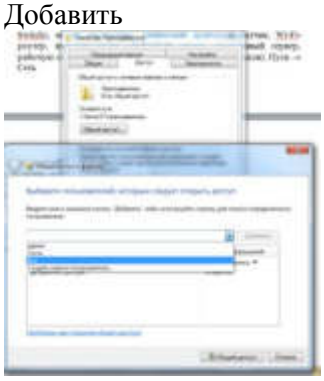
IP – адрес бывает статическим и динамическим.

IP-адрес называют динамическим (непостоянным, изменяемым), если он назначается автоматически при подключении устройства к сети и используется в течение ограниченного промежутка времени, указанного в сервисе назначавшего IP-адрес.
IP-адрес называют статическим (постоянным, неизменяемым), если он назначается пользователем в настройках устройства, либо если назначается автоматически при подключении устройства к сети и не может быть присвоен другому устройству.
Практические задания:
1. Отобразить ярлык Сеть в Главном меню (ПК на кнопке Пуск → Настроить → Сеть)



2. Ознакомиться с содержимым локальной компьютерной сети (открыть папку Сеть). Найдите в вашей сети сетевой адаптер, концентратор (HUB или Switch), модем, волоконно- оптический приёмопередатчик, Wi-Fi-роутер, интернет-сервер, файловый сервер, выделенный сервер, рабочую станцию (покажите преподавателю, что вы нашли). Скриншот окна разместить в документе Word
3. На диске D: создать папку с вашей фамилией и поместить в неё 2 любых файла.
4. На диске D: задать общий доступ для вашей папки
   4.1. Задание общего доступа папке:

ПК на папке → Свойства → Доступ → Общий доступ → из списка пользователей выбрать Все → Добавить

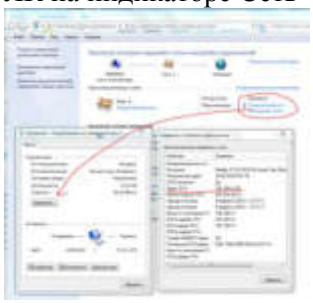


настроить доступ Чтение и запись → Общий доступ

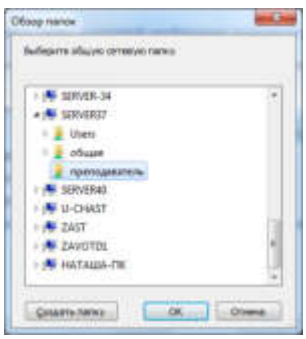
5. Проверить доступ к папке. Для этого открыть папку D:\ ваша папка на любом другом компьютере, входящем в вашу рабочую группу. Поместить скриншот содержимого вашей папки в документ Word.
6. Прерывание общего доступа папке:

ПК на папке → Общий доступ → Никому из пользователей

7. Определите IP адрес вашего персонального компьютера.
   7.1. Для определения IP адреса воспользуемся командной строкой. Для этого ЛК на кнопке Пуск и в поле Поиск ввести в поле команду cmd. Далее ввести команду ipconfig и найти свой IP адрес.
   7.2. Скрин разместить в отчёте
   7.3. ЛК на индикаторе Сеть → Центр управления сетями и общим доступом →



   7.4. Скрин окон (как в методичке) разместить в отчёте. Вырезать часть экрана с помощью инструмента Ножницы и пером отметить IP адрес.
8. Создать сетевой диск из папки Преподаватель, расположенной на ПК SERVER37.
   8.1. Пуск → ПК на ярлыке Сеть → Подключить сетевой диск → Обзор → выбрать ПК и выбрать папку → ОК .



8.2. Просмотреть как отображается сетевой диск в папке Мой компьютер
8.3. Скопировать скриншот окна Мой компьютер в отчёт.
8.4. Отправить отчёт по локальной сети в папку Преподаватель.
8.5. Найдите в сети Интернет информацию о назначении сетевого диска и выпишите в тетрадь.
9. Отключите сетевой диск Преподаватель.

Контрольные вопросы

1. Что такое компьютерная сети и её назначение
2. Классификация сетей по территориальному признаку
3. Основные понятия локальной компьютерной сети
4. Что такое IP адрес и для чего он предназначен
5. Как просмотреть свой IP адрес
6. Как назначить папке общий доступ? Как отключить общий доступ?
7. Что такое сетевой диск и как его подключить.
8. В чём отличие сетевого диска от папки с общим доступом?

5. Содержание отчета

Отчет должен содержать:
1.Название работы.
2.Цель работы.
3.Задание и его решение.
4. Вывод по работе

Практическая работа № 21

Владение функциональными особенностями популярных социальных сетей (1 час)

Цели:

1. Закрепить основные знания и умения регистрации в сетях
2. Развитие способностей планирования последовательности своих действий
3. Воспитание потребности осуществлять самоконтроль учебного процесса

Материально техническое и методическое обеспечение

ПК, Интернет, методические указания по практической работе.

Краткие сведения из теории

Социальная сеть — платформа, онлайн-сервис или веб-сайт, предназначенные для построения, отражения и организации социальных взаимоотношений, визуализацией которых являются социальные графы.

Характерными особенностями социальной сети являются:

- создание личных профилей (публичных или полу-публичных), в которых зачастую требуется указать реальные персональные данные и другую информацию о себе (место учёбы и работы, хобби, жизненные принципы и др.);
- предоставление практически полного спектра возможностей для обмена информацией (размещение фотографий, видеозаписей, размещение текстовых записей (в режиме блогов или микроблогов), организация тематических сообществ, обмен личными сообщениями и т. п.);
- возможность задавать и поддерживать список других пользователей, с которыми у него имеются некоторые отношения (например, дружбы, родства, деловых и рабочих связей и т. п.)

Самые популярные социальные сети:

- Одноклассники
- В контакте
- Мой мир
- Facebook (Фейсбук)
- My Space (Май Спейс)

Порядок выполнения работы (занятия)

Задание 1. Регистрация в контакте

Для регистрации в контакте достаточно иметь мобильный телефон, номер которого вы и прикрепляете к своей страничке в этой социальной сети. Именно номер вашего телефона и будет логином для входа на сайт ВКонтакте.

Регистрация в контакте бесплатная

Бесплатная регистрация в контакте начинается с посещения этой сети по адресу http://vk.com/. Это основная страница сайта ВК, с которой и начинаются все дальнейшие действия и отношения пользователей.

В верхней строчке набираете имя, а в строчке под ней, свою фамилию. После чего нажимаете на кнопку «Зарегистрироваться». После чего переходим к следующим действиям.

1. Указываем страну своей школы.
2. После указываем город, в котором школа указанная вами находится.
3. После чего укажите и год выпуска, или убытия из данной школы.
4. И нажмите на букву класса, в котором вы учились (а, б, в, г, д, е).

Примечание: Этот шаг вы можете пропустить на начальном пути регистрации, а при желании заполнить позже.

После этого шага нажмите кнопку «Перейти к следующему шагу».

Обратите внимание еще и на такой момент. Добавить друзей ВКонтакте, вы сможете уже в процессе регистрации, например из списка учеников школы, или ВУЗа. Но при желании сможете это сделать и после того, как зарегистрируетесь в контакте.

Дальше вы можете добавить, или не делать этой операции совсем, как добавление ВУЗа.

Далее вам придется указать свой реально рабочий сотовый телефон. Ведь именно туда вам придет код, который вам отошлет администрация контактов, который вы будете вынуждены подтвердить обязательно. Ведь если вы не подтвердите его или введете неправильные цифры, то регистрация в контактах для вас будет закрыта.

И кроме всего ваш номер мобильного, будет еще и логином для входа ВКонтакт. И так называемым гарантом безопасности от взлома вашей странички. Сеть ВКонтакте будет высылать вам бесплатные SMS – сообщения на указанный вами номер.

Еще этот шаг регистрации в контакте будет и одновременным началом создания вашей странички ВК.

Вводите страну, вводите номер телефона, и получаете свой код регистрации в контактах.

Если вас пригласили:

1) Вам приходит SMS на телефон, где указан ваш логин и пароль
2) Заходите на http://vkontakte.ru/
3) Справа вверху вводите логин и пароль на нажимаем "Войти"

Задание 2. Регистрация в Фотостране

Фотострана – это огромное интернет-государство, где есть президент, законы и даже собственная валюта.

Заходите на главную страницу Фотостраны. Слева вы увидите форму для входа, а под ней находится активная ссылка на форму регистрации (синенькое слово «Зарегистрируйтесь»).

Нажимаете на это слово, перед вами появляется несложная форма регистрации, которую и нужно заполнить. Вам предложат указать имя (никнейм), пол, дату рождения, адрес электронной почты на Мейле и пароль для доступа на сайт. Будьте осторожны с выбором пароля, постарайтесь сделать так, чтобы ваш пароль невозможно было угадать или подобрать. Для этого используйте набор букв, цифр и различных символов. Можете чередовать прописные и строчные буквы.

Когда все данные введены, нажмите «Зарегистрироваться».

Далее будет предложено ввести капчу: выбрать картинку, подходящую под описание. Капча – это подтверждение того, что вы живой человек, а не робот. Если ошибетесь с картинкой, система откажет в регистрации, и вам придется заполнять все заново. Поэтому внимательнее читайте задание и не спешите с ответом.

Следующий шаг – подтверждение своего е-мейла. Проверьте правильность адреса и нажимайте «Выслать письмо». Через пару минут на вашу почту с сайта Фотостраны придет письмо. Перейдите по указанной там ссылке.

Задание 3. Регистрация в Одноклассники

1) Заходите на http://odnoklassniki.ru/
2) Нажимаете кнопку "Зарегистрироваться"
3) В открывшемся окне заполняете анкету (имя, фамилия, дата рождения и т.д.) Придумываете логин и пароль.
4) Теперь нажимаете кнопку "Зарегистрироваться".
5) Заполняете остальную информацию о себе (е-маил, место учёбы, фото)
6) После этого нужно активировать свой профиль, для этого нужно ввести номер своего телефона.
7) На него придёт код активации, который нужно вписать в нужную строчку и нажать кнопку "Активировать".
8) Если нужно, то вводим свой логин и пароль и входим на свою страницу.

После чего добавляем всю остальную информацию о себе.

Задание 4. Регистрация в Мой мир.

1) Заходите на http://mail.ru/
2) Нажимаете "Регистрация в почте" (справа)
3) На странице регистрации заполняете все поля
4) После этого вводите свой номер телефона
5) Внизу ставите галочку на "Создать личную страницу на Мой Мир@Mail.Ru"
6) Вам на телефон придёт SMS с кодом активации, который нужно ввести в нужное место. Регистрация окончена. Теперь нужно нажать сверху кнопку Мой Мир и заполнить свой профиль.

Задание 5. Регистрация в Facebook (Фейсбук).

1) Заходите на http://www.facebook.com/
2) Заполняете все поля в приведенной форме и жмёте по кнопке "Регистрация":
3) Подключаете свои аккаунты из других мест
4) Добавляете информацию о себе и своё фото
5) Подтверждаете регистрацию через почтовый ящик

Регистрация окончена. Осталось заполнить свой профиль.

Задание 6. Регистрация в Myspace (Майспейс).

1) Заходите на http://ru.myspace.com/
2) Сверху нажимаете кнопку "Зарегистрироваться"
3) Вводите всю информацию и нажимаете "Бесплатная регистрация"
4) Подтвердите e-mail

Задание 7. Регистрация в Twitter (Твиттер).

1) Заходите на http://twitter.com/
2) Нажимаете кнопку "Регистрация"
3) Вводите всю информацию в соответствующие поля и жмёте "Создать учётную запись" Заполняйте профиль

Контрольные вопросы

1. Что такое социальные сети?
2. Приведите примеры популярных социальных сетей?
3. Что такое логин?
4. Этапы регистрации в соцсетях?

5. Содержание отчета

Отчет должен содержать:

1.Название работы.
2.Цель работы.
3.Задание и его решение.
4. Вывод по работе

Практическая работа № 24
«Работа с HTML5, CSS, JavaScript (3 час)

4. Цели и задачи учебного занятия

Цель: сформировать практические умения по программированию объектов JavaScript.

Задачи

- продолжить работу по формированию умений внедрять объекты JavaScript в HTML-документ;
- продолжить работу по формированию умений использования инструментальных средств программирования ИС,
- продолжить работу по формированию умений нахождения ошибок кодирования в разрабатываемых модулях ИС

Планируемые результаты

В результате прохождения занятия «Практическая работа «Программирование объектов на JavaScript» студент должен:

знать/понимать

- спецификации языка, этапы создания графического интерфейса;

уметь

- программировать в соответствии с требованиями технического задания;
- использовать языки структурного программирования и языка сценариев для создания независимых программ
-формулировать задачи проекта, выполнять управление проектом с использованием инструментальных средств;

владеть

- языковыми структурами и конструкциями для проектирования web-страниц.

5. Тип учебного занятия: урок совершенствования умений и навыков.

Форма проведения: практическая работа, решение проблемных задач.

Форма познавательной деятельности обучающихся: индивидуальная, фронтальная.

Технология: проблемное обучение, электронное обучение.

Материально-техническое обеспечение: персональные компьютеры по числу учащихся с предустановленной программой Notepad++ и браузером; мультимедийное устройство; раздаточный материал в печатном и электронном виде.

Практическая работа представлена в электронном виде в виде кода:

```
<!DOCTYPE html>
 <html>
 <meta charset=utf-8>
   <head>
     <title>Практическая работа</title>
   </head>
    <body>
        <font color=black>
        <p align=center>Практическая работа</p><br>
        Задание 1. Дополните и прокомментируйте код с учетом следующих параметров: <br>
1.      Использовать только язык HTML;<br>
2.      Установить фоновое изображение из предложенных в папке «Фон»;<br>
3.      Установите цвет шрифта таким образом, чтобы он был видим на выбранном фоне;<br>
4.      Вставьте изображение из предложенных в папке «Вставка изображений» с помощью тега img
src;<br>
5.      Удалите текст задания 1 из кода страницы, оставив надпись «Практическая работа» по центру
экрана<br>
6.      Прокомментируйте код, покажите результат в виде запущенной страницы преподавателю.<br>
<br><br>

Задание 2. Дополните и прокомментируйте код с учетом следующих параметров:<br>
1.      Запустите файл Пример1 и рассмотрите работу кода JavaScript;<br>
2.      На созданной в первом задании странице создайте надпись «This page was created using JavaScript» и
расположите её по центру;<br>
3.      Создайте кнопку «Перевод»;<br>
4.      Припишите код JavaScript, чтобы при нажатии на кнопку надпись менялась на «Эта страница была
создана с помощью JavaScript»;<br>
5.      Удалите текст задания 2 из кода страницы;<br>
6.      Прокомментируйте код, покажите результат в виде запущенной страницы преподавателю.<br>
```


Задание 3. Дополните и прокомментируйте код с учетом следующих параметров:
1. Запустите файл Пример2 и рассмотрите работу кода JavaScript;
2. На созданной в первом задании странице расположите после надписи рисунок smile1.jpg;
3. Создайте кнопки «Жёлтый», «Фиолетовый»;
4. Припишите код JavaScript, чтобы при нажатии на кнопки изображение менялось, используя вторым файлом smile2.jpg;
5. Удалите текст задания 3 из кода страницы;
6. Прокомментируйте код, покажите результат в виде запущенной страницы преподавателю.

Задание 4. Дополните и прокомментируйте код с учетом следующих параметров:
1. Запустите файлы Пример3, Пример4, Пример5 и рассмотрите работу кода JavaScript;
2. Используя полученные данные, создайте кнопку «Увеличить шрифт», которая увеличивала бы размер надписи «JavaScript», расположенной после задания 3;
3. Создайте две кнопки «Спрятать» и «Показать», которые бы скрывали и показывали текст задания 4;
4. Не удаляйте текст задания 4 из кода страницы;
5. Прокомментируйте код, покажите результат в виде запущенной страницы преподавателю.

</body>
</html>

5. Содержание отчета

Отчет должен содержать:
1.Название работы.
2.Цель работы.
3.Задание и его решение.
4. Вывод по работе

Практическая работа № 25

"Создание интерактивного контента с помощью HTML5, CSS, JavaScript"(3 часа)

Программа работы:

1. Взаимодействие с файлами
2. Использование мультимедиа
3. Реагирование на дислокацию и контекст обозревателя
4. Отладка и профилирование веб-приложений
5. Внедрение видео

Контрольные вопросы:

1Как можно описать синтаксис JavaScript и его использование в HTML5

2Как следует использовать JavaScript для управления DOM и обработкой событий

3 Опишите использование библиотеки jQuery для упрощения кода на JavaScript

5. Содержание отчета

Отчет должен содержать:
1.Название работы.
2.Цель работы.
3.Задание и его решение.
4. Вывод по работе

Практическая работа № 26

"Регистрация почтового ящика электронной почты, создание и отправка сообщения"(1час)

Цель: Проверить умения учащихся по регистрации ящика почтовой электронной почты и созданию и отправке сообщения.
Задание 1: Зарегистрировать в mail, yandex или rambler свой почтовый электронный ящик. Электронный адрес своего ящика сообщить учителю.
Задание 2: создать сообщение, прикрепить текстовый файл (графический файл или что-то другое). Добавить эмоциональности электронному письму с помощью смайликов.
Задание 3: отправить сообщение по адресу, указанному учителем (своему однокласснику).
Задание 4: проверить электронную почту на свое имя. Если есть сообщение, то составить ответ и отправить его
Задание 5: Отправить сообщение на адрес учителя (с уведомлением о получении).
Рекомендации по проведению практической работы:
а) На выполнение работы отводится 45 минут.
б) Учащиеся, имеющие на домашнем компьютере выход в Интернет, как правило, хорошо умеют работать с электронной почтой, так как постоянно обмениваются друг с другом сообщениями. Для таких учащихся данная практическая работа не составляет труда. Но есть учащиеся, не имеющие дома выхода в Интернет, поэтому для них эта работа имеет важное значение.

5. Содержание отчета

Отчет должен содержать:
1.Название работы.
2.Цель работы.
3.Задание и его решение.
4. Вывод по работе

Практическая работа №27

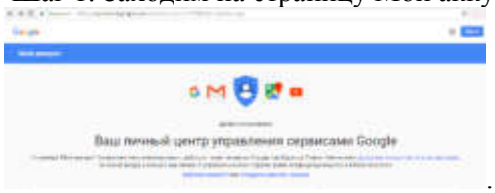
"Поиск информации в сети интернет. Ссылки и переходы" (1 час)"

Цель работы: изучить основные правила поиска в сети Интернет, синтаксис поиска в сети; научится на практике использовать полученные знания.

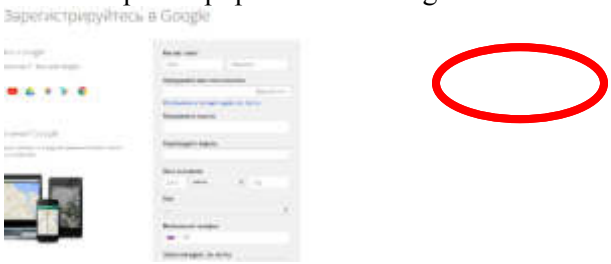
Ход выполнения работы

1. Войти в свой аккаунт Google (если аккуанта нет, то создаем)

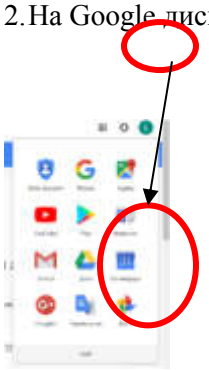
Шаг 1. Заходим на страницу Мой аккуант Google.

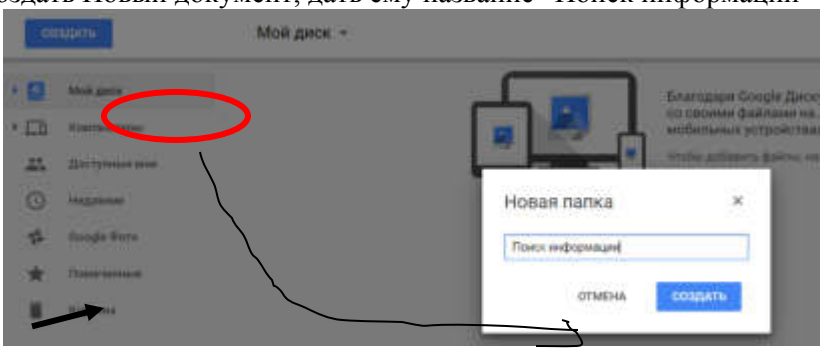


Шаг 2. Зарегистрироваться в Google.



2. На Google диске создать Новый документ, дать ему название “Поиск информации”





3. В настройках доступа **Разрешить Всем** доступ к Вашему документу

Краткие теоретические сведения

Точность поиска – выражает отношение совокупности релевантных откликов, к количеству всех выданных документов.

Правила поиска информации в сети Интернет?

1. Учитывать особенности естественного языка.
2. Не допускать орфографических ошибок.
3. Избегать поиска по одному слову, использовать необходимый и достаточный набор слов.
4. Не писать большими буквами.
5. Исключать из поиска не нужные слова.

Понятие языка запросов.

Синтаксис язы	**Значение**	**Пример**
!	Запрет перебора всех словоф	! педагогическая система (из поиска будут исключе слова педагогические системы)
+	Обязательное присутствие сл	Педсовет по+пятница (должны быть выбраны страни где встречаются слово не только педсовет) но

	найденных документах	обязательное условие наличие слова «пятница»
&	Обязательное вхождение сло одно предложение	Педагогическая & система
«»	Поиск устойчивых словосочет	«педагогическая система» (учитывается строгая последовательность слов, слово «система педагогичес будет исключеная)

Задание №1.

Определить провайдера, с помощью которого Ваш компьютер подключен к Интернету и найти в каком году он появился на рынке.

Задание №2.

Указание адреса страницы.

а) Запустить обозреватель.

б) Ввести в адресную строку

http://top140.com/fantasy/library/tolkien.htm

в) По полученным материалам выяснить, где и когда родился Дж.Р.Р.Толкиен (автор книги «Властелин кольца»).

г) Скопировать полученные данные и вставить в файл текстового документа.

Задание №3.

а) Назовите автора строк "Кто владеет информацией - тот владеет миром"

Ответ ввести в формате: фамилия. При вводе ответа не вводите лишних пробелов перед словом и после него. Фамилию вводить на русском языке!

б) Скопировать и вставить в документ ссылку с адресом сайта, где взят ответ

Задание №4

Используя любой поисковый сервер, найдите ответы на следующие вопросы:

1. Существует три типа речных окатанных камней: гравийные (диаметр от 1 до 10 мм), галичниковые (диаметр — от 10 до 100 мм). Как называется третий тип камней?

2. Говорят, что мало - кто из современников Эйнштейна понимали его теорию. Макс Планк, утешая Эйнштейна, говорил: "Новые теории никогда не принимаются. Они или опровергаются, или их противники...". Закончите мысль.

3. Считалось, что носить большие и красивые манжеты на рубашках у средневековых студентов значило прослыть трусом. Почему?

Оформите информацию в текстовом редакторе Word, сделав ссылки на сайты, содержащие ответы на вопросы.

4. Найдите в Интернет два закона РФ, регулирующие деятельность в области информационных технологий.

Задание №5

С помощью поисковой системы Google вы можете найти в сети имеющееся у вас изображение и определить по его описанию то, где оно сделано, кто его автор, кто на нем изображен и т.д

а) Для того, чтобы воспользоваться поиском по изображениям, зайдите на сайт поисковой системы google.com и перейдите в раздел **Картинки**, который находится в правой верхней части страницы.

б) В открывшемся окне щелкните по значку фотоаппарата в правой части поисковой строки. Далее укажите ссылку на изображение, которое вы ищите или загрузите файл изображения со своего компьютера.

в) Нажмите на кнопку Поиск по картинке и поисковая система предложит вам варианты найденных изображений. перейдите по одной из предложенных ссылок для того, чтобы ознакомиться с результатами поиска.

г) Для того, чтобы сохранить картинку с веб-страницы на своем компьютере для дальнейшего поиска информации о ней, щелкните на картинке правой кнопкой мыши и выберите пункт меню "Сохранить картинку как..."д) Картинку из текста скопировать в Папку своей группы.

е) Найдите с помощью поиска Google ответ на вопрос:

ИТОГИ РАБОТЫ!!!

1.Выполнение всех заданий отображается в MS Word.
2.Сохраняем результат на свой Google-диск.
3.Отправляем ссылку своего Google-диска преподавателю

Контрольные вопросы:

1.Как следует использовать технологию Drag and Drop и API для взаимодействия с файлами (File API)
2.Как следует использовать элементы аудио и видео
3.Как можно определить местоположение пользователя, запустившего Веб-приложение, с помощью API геолокации
4.Как можно отлаживать и персонифицировать веб-приложения, используя Web Timing API и Internet Explorer Developer Tools

5. Содержание отчета

1.Название работы.
2.Цель работы.
3.Задание и его решение.
4. Вывод по работе

Практическая работа №28
"FTP-клиенты. Загрузка файла из FTP-хранилища" (1 час)

Цель: Запуск FTP из командной строки

Исходные данные/сценарий

FTP (протокол передачи файлов) входит в набор протоколов TCP/IP. Протокол FTP используется для передачи файлов от одного сетевого устройства к другому. В ОС Windows входит клиентское FTP-приложение, которое можно запустить из командной строки. Существуют также бесплатные версии FTP с графическим интерфейсом пользователя (GUI), которые можно загрузить из Интернета. Работать с ними гораздо проще, чем набирать текст в командной строке. Протокол FTP часто используется для передачи файлов, размер которых слишком велик для пересылки по электронной почте.

Необходимые ресурсы

Компьютер под управлением Windows 10, с доступом к командной строке, интернет-подключением и установленной программой WS_FTP LE

Запуск FTP из командной строки

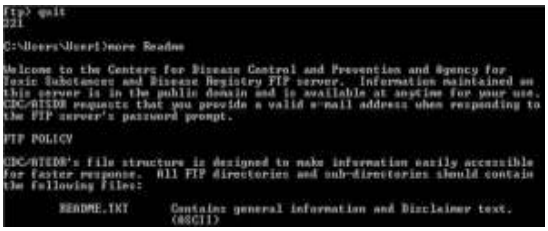
В Нажмите кнопку **Windows Пуск**, введите в поле поиска **cmd** и нажмите клавишу ВВОД, чтобы открыть окно командной строки.

В При появлении приглашения C:\> введите **ftp ftp.cdc.gov**. При появлении приглашения **Пользователь (ftp.cdc.gov:(none)):** введите **anonymous**. Пароль не указывайте. Нажмите клавишу ВВОД, чтобы войти как анонимный пользователь.

Обратите внимание, что теперь вместо C:\> в качестве приглашения отображается ftp>. Введите команду **ls** , чтобы отобразить список файлов и папок. На момент составления лабораторной работы на сайте был доступен файл справки Readme.

1. В командной строке введите **get Readme**. После этого на локальный компьютер с анонимного FTP-сервера будет загружен файл, подготовленный Центром по контролю и профилактике заболеваний. Файл будет скопирован в каталог, указанный в приглашении C:\> (в данном случае

2. Введите **quit**, чтобы выйти из FTP-сервера и вернуться к приглашению C:\>. Введите **more Readme**, чтобы отобразить содержимое документа.



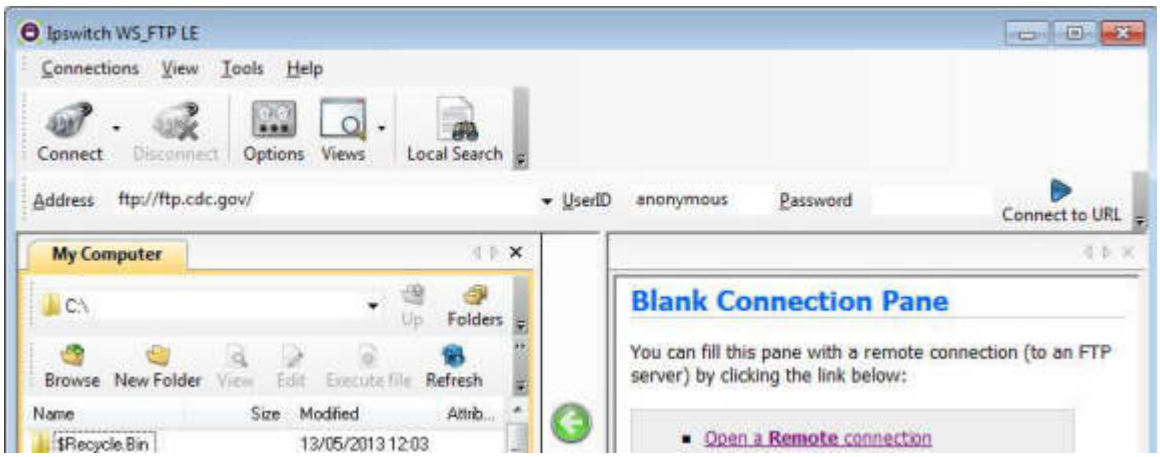
2. В чём недостаток доступа к FTP из командной строки?

Загрузка FTP-файла с помощью программы WS_FTP LE

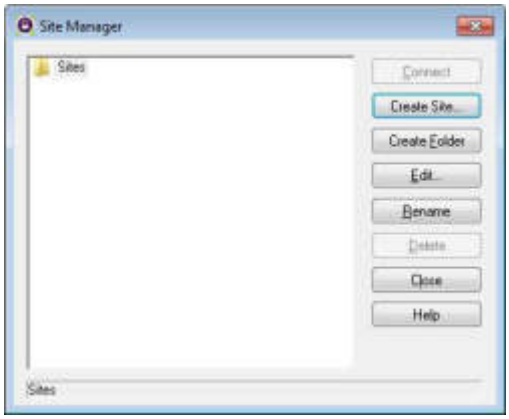
В части 2 вы загрузите файл, используя WS_FTP LE — бесплатный FTP-клиент.

В Запустите программу **WS_FTP LE**. Если откроется окно Ipswitch WS_FTP LE, нажмите кнопку **Next** (Далее), чтобы продолжить и перейти к шагу C. Если окно не откроется, нажмите на ссылку **Open a Remote Connection** (Открыть удалённое подключение).

7. Корпорация Cisco и/или её дочерние компании, 2014. Все права защищены.
В данном документе содержится общедоступная информация корпорации Cisco.



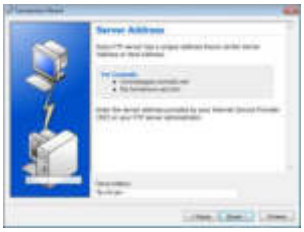
1 Нажмите кнопку **Create Site…** (Создать сайт).



1. В поле **Site Name** (Имя сайта) введите **Center for Disease Control** (Центр по контролю и профилактике заболеваний) и нажмите кнопку **Next**.

d. Откройте раскрывающийся список **Connection Type** (Тип подключения), выберите **FTP** (тип подключения, используемый по умолчанию) и нажмите кнопку **Next**.

В поле **Server Address** (Адрес сайта) введите **ftp.cdc.gov** и нажмите кнопку **Next**.

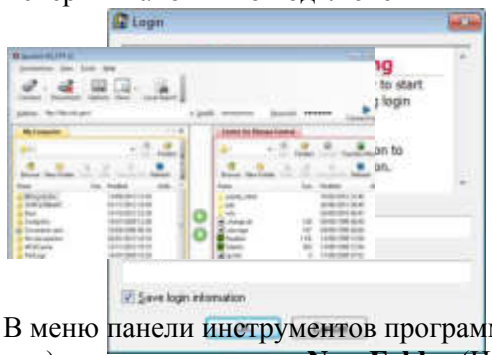


f. В поле **User Name** (Имя пользователя) введите **anonymous** и оставьте поле для ввода пароля пустым. Нажмите кнопку **Next** (Далее).

g. Нажмите кнопку **Finish** (Готово).

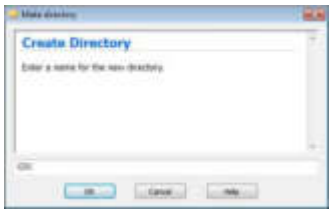
h. В открывшемся окне Login Information Missing (Отсутствует информация для входа в систему) нажмите кнопку **OK**. Не вводите пароль в поле **Password** (Пароль).

i. Теперь вы анонимно подключены к FTP-сайту Центра по контролю и профилактике заболеваний.

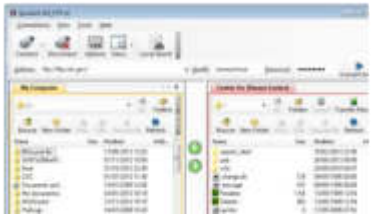


j. В меню панели инструментов программы WS_FTP LE на вкладке My Computer (Мой компьютер) нажмите на значок **New Folder** (Новая папка), чтобы создать папку на локальном диске **C:** своего компьютера.

k. В окне Make Directory (Создать каталог) укажите **CDC** в качестве имени папки и нажмите кнопку **OK**. **Примечание**. Если такая папка уже существует, можете использовать её или создать новую папку с другим именем. В случае использования уже имеющейся папки CDC можно заменить старый файл Readme загруженным файлом.



6. Перейдите на вкладку **My Computer** (Мой компьютер) и дважды нажмите на созданную папку, чтобы её открыть.



1. Перетащите файл **Readme** из правой панели приложения (удалённого FTP-сервера CDC) в папку CDC на локальном диске **C:**.

n. Дважды нажмите на файл **Readme** в папке **C:\CDC** на локальном диске **C:**. Если программа спросит, в каком приложении открыть документ, выберите любой текстовый редактор. Появится сообщение следующего содержания.

```
Welcome to the Centers for Disease Control and Prevention and
Agency for Toxic Substances and Disease Registry FTP server.
Information maintained on this server is in the public domain
and is available at anytime for your use.
```

o. Как было проще работать с FTP — из окна командной строки **cmd** или с помощью программы

WS_FTP LE? _____

p. По завершении работы с FTP нажмите кнопку **Disconnect** (Отключить), чтобы отключиться от сайта ftp.cdc.gov.

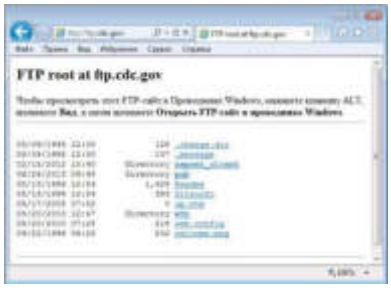
q. Удалённый узел будет исключён из списка сохранённых FTP-сайтов. В окне Ipswitch WS_FTP LE нажмите ссылку **Open a Remote Connection** (Открыть удалённое подключение). Выберите сайт **Center for Disease Control** (Центр по контролю и профилактике заболеваний) и нажмите кнопку **Delete** (Удалить), чтобы удалить FTP-сайт. Нажмите кнопку **Yes** (Да), чтобы подтвердить удаление. Нажмите кнопку **Close** (Закрыть), чтобы выйти из менеджера сайтов.

r. Удалите папку **C:\CDC**.

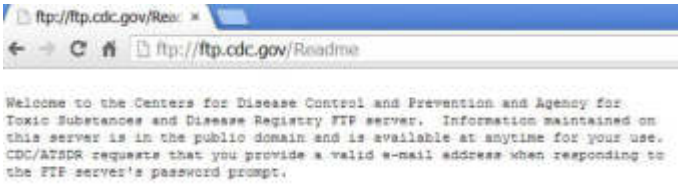
Запуск FTP в браузере

В качестве клиента для анонимного доступа к FTP можно использовать браузер.

a. В адресной строке браузера введите **ftp://ftp.cdc.gov/**.



b. Нажмите на файл **Readme**.



Контрольные вопросы

1. Какой из способов доступа к FTP было легче всего использовать?

2. Какие преимущества предоставляет доступ к FTP из командной строки и из программы WS_FTP LE?

5. Содержание отчета

Отчет должен содержать:
1.Название работы.
2.Цель работы.
3.Задание и его решение.
4. Вывод по работе

Практическая работа №29
"Торрент-клиенты. Загрузка торрент файла (1час)

Цель. Научиться работать с **Торрент-клиенты. Загрузка торрент файла**

Торрент (Torrent) *— это сервис обмена информацией по сетевому протоколу Bittorrent, данная технология позволяет обмениваться файлами на высокой скорости. Огромным преимуществом данной технологии стала возможность установки загрузки на паузу, а продолжить скачивание можно в удобное для пользователя время.*

Следует уделить особое внимание скорости скачивания через сервисы торрента, в большей части пользователям и полюбился данный вид загрузки из-за возможности быстро скачать нужную информацию.

Как работает торрент трекер?

Чтобы понять почему торрент клиенты лучше для скачивания информации необходимо понимать принцип их работы.

Начинается все с того, что Вы или любой другой пользователь хотите поделиться каким-то файлом. Находите его у себя на компьютере и вместо привычной загрузки на какой-нибудь файлообменник делаете загрузку через специальную программу торрент клиент, при этом файл как находился у вас на компьютере так и остается на своем месте не куда не выгружаясь, программа автоматически формирует список с адресом к данному файлу и создает отдельный файлик — которым мы и будем делиться всеми возможными способами.

Главное, вы должны понять, что при загрузке через торрент трекер сам файл или папка не куда не выгружаются, просто создается специальный список, в котором указано описание файла и адрес его размещения.

Когда любой пользователь получит ваш список файлов и запустит его через свой торрент клиент, программа автоматически определит, где находится файл и начнет его скачку напрямую с вашего компьютера.

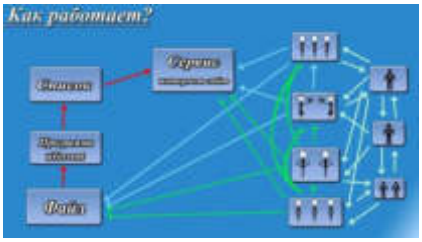
После того, как пользователь начнет скачивать ваш файл, он автоматически станет участником сложного процесса, потому что в этот момент он сам начнет передавать ваш файл следующему пользователю, как эстафета, тем самым файл уже раздаете вы и еще один пользователь, потом вас станет трое, четверо и т.д. каждый будет автоматически участвовать в передаче файла. Чем больше людей скачивает данный файл, тем быстрее будет скорость загрузки.

Достигается это способом передачи файла частями. Например, файл через торрент скачивает 100 человек, присоединился еще один пользователь и начал закачку.

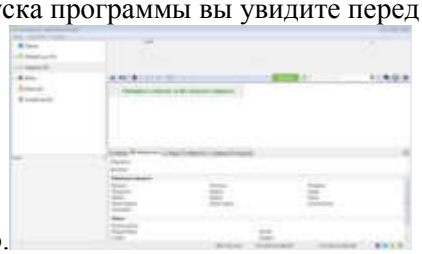
С помощью торрент программы он может скачивать по кусочку информацию у каждого из сотни пользователей.

К примеру, если при скачке целого файла скорость маленькая, программа выделит определенный кусочек информации и пойдет к следующему пользователю, у которого скорость выше, чтобы скачать еще один кусочек и так будет искать максимально комфортный вариант скачки, разделяя сам файл на огромное количество частей.

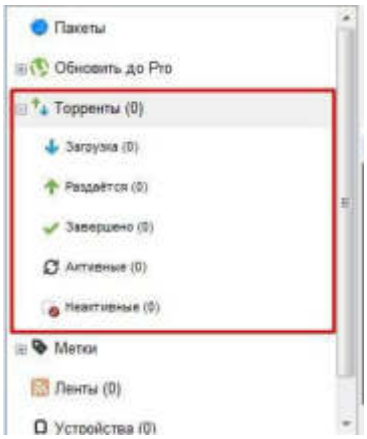
После того, как будут собраны все кусочки информации, программа их соберет в один единый файл, и загрузка будет завершена.



Краткий обзор программы uTorrent

После запуска программы вы увидите перед собой окно приложения, в котором на первый взгляд ничего не понятно. 

Если вы обычный пользователь, которому нужно уметь скачивать, то необходимо знать всего несколько моментов и где их посмотреть. для начала обратите внимание на таблицу слева в ней будут отображать списки загружаемых файлов, уже загруженных и раздающихся.



Теперь обратите внимание на основную область программы, на ней вы сможете увидеть кнопки управления, такие как пауза, стоп, продолжить скачку, создать раздачу, они могут немного меняться в зависимости от версии приложения. Так же тут можно будет увидеть название файла, который вы скачиваете, его размер, общий прогресс скачки в процентах и примерное время до полной загрузки.

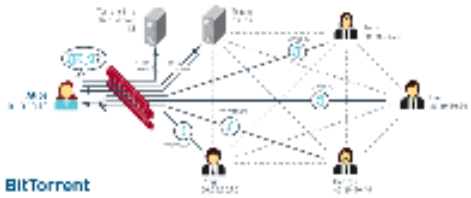
Опустившись чуть ниже, можно посмотреть подробное описание многих процессов.

Во время работы с программой вы можете встретить интересные название, значение которых не понятны — это Сиды и Пиры.

Сиды — это люди, которые загрузили файл уже на 100% и добровольно продолжают делиться им с пользователями.

Пиры — это люди, которые скачали файл меньше, чем на 100%, но уже раздают другим пользователям те частички, которые успели скачать.

Еще в общении с пользователями интернета часто можно услышать выражение торрент трекер, тоже не надо пугаться — это обычные интернет-сайты, на которых опубликованы списки торрент файлов с кратким описанием, характеристиками и картинками.



Когда скачивание почти завершено, клиент входит в режим, именуемый end game. Он запрашивает все оставшиеся сегменты у всех раздающих клиентов, что позволяет избежать замедления почти завершенной закачки.

При получении полного файла клиент становится сидом, то есть переходит в специальный режим работы, в котором он только отдаёт данные. Далее сид периодически информирует трекер об изменениях в состоянии торрентов и обновляет списки IP-адресов.

Принцип скачивания с помощью торрентов

Особенностью такого обмена является то, что протокол сам выбирает самый быстрый источник, а если же с этим источником что-то произойдет (отключится или резко снизится скорость), выбор падет на другой источник. Причем файлы скачиваются различными частями, а не последовательно.

Таким образом, как только один пользователь начинает скачивание, то автоматически включается раздача, что позволяет остальным пользователям скачивает те сегменты файла, которые уже скачаны другими.

Всю работу на себя берет специальная программа BitTorrent-клиент.

На самом сервере, где происходит раздача файлов (торрент-трекеров), фактически файл не находится. Все они лежат у пользователей. Сервер предоставляет всего лишь список файлов и координирует раздачи.

Из-за этого иногда выходит так, что файл есть в списке, но скачивание не происходит, потому что нет пользователей, которые бы его раздавали. Часто выходит, что днем у вас не получается скачать файл, а вот ночью происходит скачивание, потому что пользователи, у которых есть этот файл, подключились.

Загрузка файлов происходит в фоновом режиме, поэтому компьютер не тратит на это серьезных ресурсов. Вы выбираете файл, начинаете загрузку и при этом занимаетесь всем, чем вам нужно, программа сама все закачивает. По этой причине BitTorrent – очень удобный способ для обмена файлами между пользователями.

Для того, чтобы подключиться к торрент-клиенту и начать скачивание, нужно скачать и установить непосредственно саму программу и пройти регистрацию на торрент-трекере. Затем выбираете файл, который вы хотите получить. Все остальное сделает программа.

Если вы решите самостоятельно выкладывать файлы для скачивания, вам нужно внимательно ознакомиться с инструкцией торрент-трекера, ведь на каждом свои правила. Хотя вы можете просто не

удалять из списка раздач скачанные файлы, тем самым позволяя скачивать их у вас другим пользователям.

Как работает torrent

BitTorrent, или простоTorrent – это система обмена пользователей файлами через интернет. Термин BitTorrent относится к программе и протоколу для файлобмена, а сервер, координирующий этот процесс, называется BitTorrent-трекер.

Инструкция

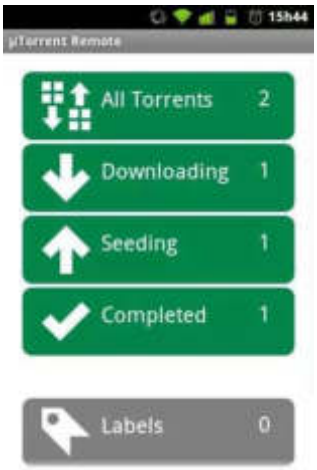
1. Для передачи файлов через интернет сетевой протокол Torrent разбивает их на мелкие части. Программа Torrent-client, установленная на компьютере пользователя, загружает эти части и собирает их в единый файл, она же занимается раздачей уже скаченных фрагментов. Таким образом, файлы через систему BitTorrent раздаются с большей скоростью. Кроме того, основной источник раздачи – сидер может не участвовать в процессе после того, как файл загрузят несколько других пользователей.
2. Сайт торрент-трекер не хранит файлы, но через него осуществляется управление файлообменным процессом. Все раздаваемые данные содержатся на компьютерах пользователей. Информация о частях скачиваемого фильма, книги или музыкальной композиции, их количестве и порядке размещения содержится в файле с расширением torrent, с которыми работает программа Torrent-client.
3. Чтобы скачать файл с одного из BitTorrent-трекеров, необходимо загрузить на свой компьютер и установить программу Torrent-client. Затем нужно найти и сохранить на свой жесткий диск файл с расширением torrent, который предлагается пользователям в раздаче фильмов, игр, музыки и другой цифровой продукции на сайте торрент-трекера. Запустите загруженный файл двойным щелчком мышки, и программа Torrent-client начнет загрузку выбранного вами контента.
4. Если пользователь удаляет торрент-файл или загруженный контент, он выходит из раздачи через Torrent-client. Участвуя в работе BitTorrent-трекера, желательно оставаться некоторое время на раздаче файла, чтобы дать другим пользователям возможность его загрузить. На ряде торрент-трекеров с регистрацией за участие в раздаче пользователь увеличивает свой рейтинг, позволяющий ему скачивать больше контента.
5. Загрузка и раздача файла через сайт BitTorrent-трекер также осуществляется с помощью программы Torrent-client. Выберите файл, которым вы хотите поделиться с другими пользователями. В программе-клиенте найдите этот документ в разделе «Выбрать файл». Убедитесь, что в графах Preserve file order, Start seeding и Private torrent нет галочек. Иначе файл будет недоступен для общего скачивания или раздача будет создаваться очень медленно, поскольку она одновременно будет автоматически раздаваться. Нажмите на кнопку OK. Процедура создания торрента-файла завершена.
6. Созданный файл с расширением torrent необходимо залить на сетевой ресурс. На разных торрент-трекерах существуют свои правила организации раздачи, с которыми необходимо предварительно ознакомиться. На большинстве ресурсов есть возможность создания новой темы, в которой по шаблону заполняется описание контента и прикрепляется созданный торрент-файл.

Видео по теме

Торрент-программы для мобильных устройств

Что касается мобильных устройств, которые в последнее время стали очень популярными и мощными, практически для всех операционных систем можно найти соответствующие приложения.

Исключением, пожалуй, могут стать «яблочные» устройства, поскольку сама компания Apple борется с компьютерным пиратством очень жестко и не представляет подобные программные продукты в своих Интернет-маркетах.



Если же говорить об остальных ОС, то установка или ответ на вопрос о том, как пользоваться торрентом, здесь еще проще, чем в случае с компьютером, а скорость загрузки будет напрямую зависеть только от наличия свободного места на внутренней флэш-памяти или на съемной карте, а также скорости беспроводного соединения типа Wi-Fi.

Соответственно, в настройках самих систем предварительно нужно будет указать место сохранения загружаемых данных.

Как и в компьютерных программах, мобильные приложения позволяют использовать большинство стандартных настроек.

5. Содержание отчета

Отчет должен содержать:
1.Название работы.
2.Цель работы.
3.Задание и его решение.
4. Вывод по работе

Практическая работа №30
"Работа с облачными хранилищами. OneDrive. Google Drive. Yandex Disc (1 час)
Цель: Работа с облачными хранилищами. OneDrive. Google Drive. Yandex Disc

Каждый пользователь ПК или смартфона знает, что такое облачное хранилище. Для тех же, кто не в курсе, коротко расскажу, что это. В любом почтовом аккаунте, будь то mail.ru или Яндекс, предоставляется небольшое дисковое пространство для хранения ваших фото, видео, документов и т.д. К тому же в «облаке» сейчас очень удобные видеоплееры, просмотрщики фото, редакторы Excel и Word, там также можно создавать папки, удалять, редактировать и выполнять те же процедуры, которые вы делаете на обычном компьютере. Более подробно об этом вы можете почитать тут - Облако майл - и тут - Яндекс диск.

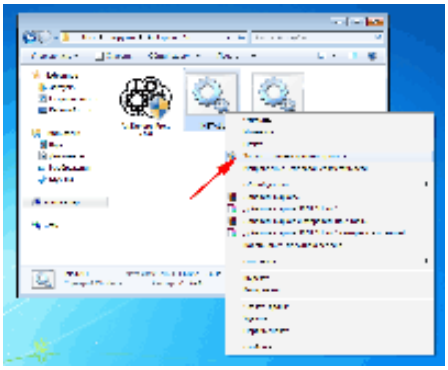
Действительно, облачные сервисы — это самые надежные источники хранения ваших файлов, поскольку на таких серьезных серверах файлы дублируются и занимаются этим специалисты. Ну а пользовательские источники хранения информации, такие как винчестеры, DVD, флешки и т.п. в любой момент могут выйти из строя, никто не застрахован. Поэтому всегда нужно делать дубликаты важных фото и документов.

Облачных хранилищ существует очень много. Google предоставляет нам - «Google Drive», Microsoft - «OneDrive», Яндекс, Майл ру, DropBox - это только малая часть из сервисов, которые бесплатно предоставляют данную услугу. К тому же еще существуют гиганты, которые предоставляют аж 50 гигабайт дискового пространства (возможно сейчас поменьше), например - MEGA. Кстати, если хотите зарегистрироваться и получить в пользование облако, можете пройти по приведеным выше ссылкам.

С одной стороны количество «хранилищ информации» очень радует, но с другой стороны из-за обилия сервисов мы часто забываем что и куда сохраняем. Наверняка каждый имеет несколько почтовых аккаунтов, в которые уже сохраняли и документы и фото с видео. Иногда начинаешь искать куда же ты залил важный вордовский документ, начинаешь рыться в логинах и паролях - в общем хватает головной боли. А так хотелось бы все систематизировать, сгруппировать в одном месте, чтобы в случае необходимости быстро «прошерстить» папки и получить нужное.

Давно уже приходили мысли о подобной «фрагментации» и наконец мысли материализовались)). Недавно случайно натолкнулся на программу **Air Explorer Pro**, которая специально и была создана для упорядочивания облачных сервисов. В этой статье будет небольшой обзор ее функциональности.

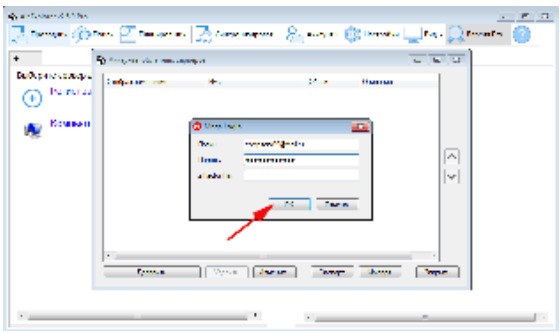
Для начала бесплатно скачиваем **Air Explorer Pro** по ссылке, находящейся внизу поста программы. После загрузки архива - **Air Explorer 2.6.zip** - распаковываем его в текущую папку. Вы увидите три файла, щелкаем по файлу - **INSTALL.cmd** - правой кнопкой мыши и выберите - **Запуск от имени администратора**.



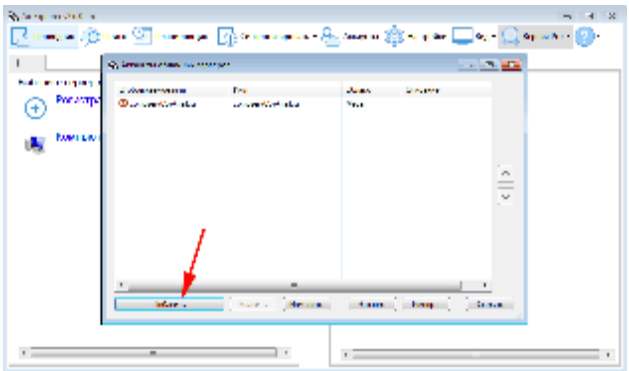
Это самый простой и надежный способ быстро и надежно установить программу. Можно, конечно, выбрать портативную установку, если щелкнуть по файлу - **PORTABLE.cmd**, но как показывает практика, портативные версии не всегда корректно работают.

После установки программы в тихом режиме на рабочем столе появится ярлык программы - **Air Explorer Pro**, запускаем его. После запуска программы появится главное окно с иконками популярных облачных сервисов. Красными стрелками я выделил те, к которыми я подключу свои аккаунты.

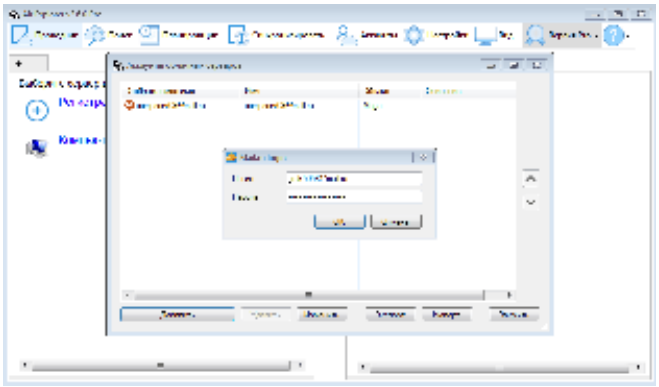
Начнем, пожалуй, с самого большого по объему, с МЕГИ. Щелкаем по одноименной иконке, в появившемся окне вводим данные от своего зарегистрированного ранее аккаунта. Последнюю, третью графу оставляем пустой.



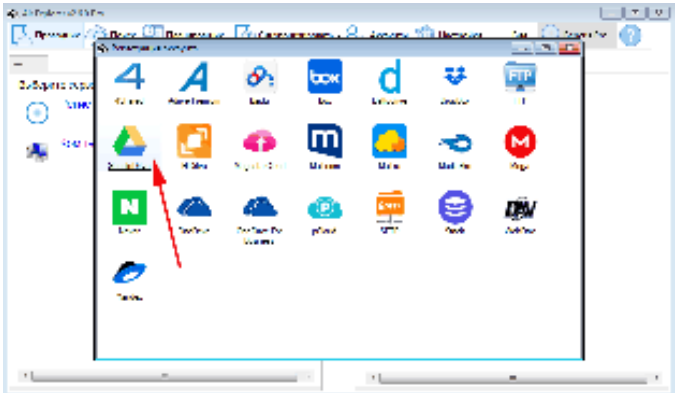
После заполнения и нажатия кнопки - **ОК** - увидите окно с добавленным аккаунтом Меги. Давайте сразу же добавим еще несколько, от других сервисов.

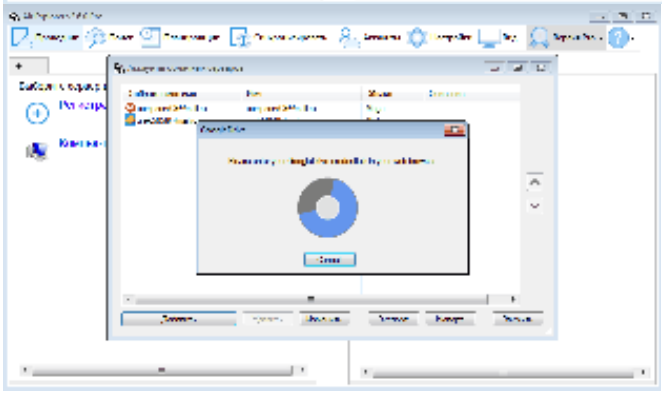


После щелчка по кнопке - **Добавить** - снова увидите иконки хранилищ, я выбрал Mail.ru. Подобным способом заполняем форму авторизации.

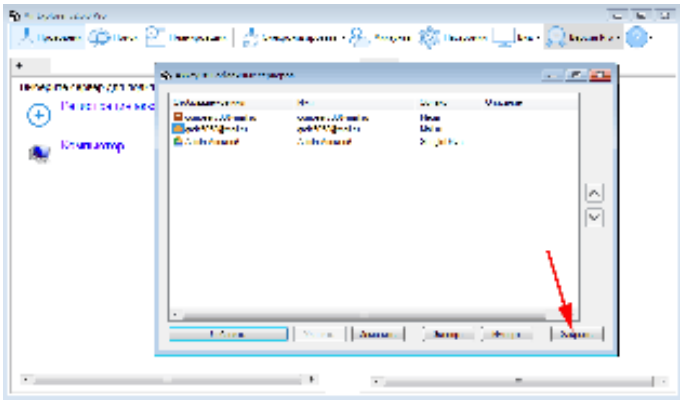


Следующим я выбрал - **Goggle Drive**. И вот тут есть одно, но. Прежде чем авторизоваться в аккаунте гугла, нужно авторизоваться в самом браузере. Собственно, после щелчка по иконке программа сама перекинет нас на нужную страницу авторизации.

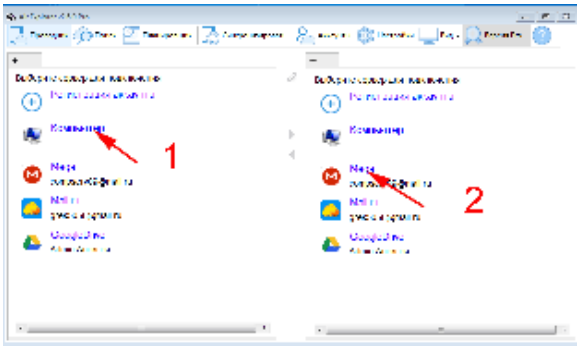




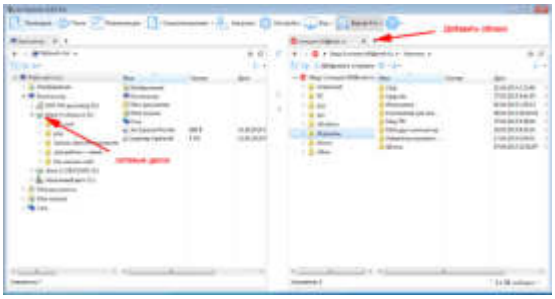
Все везде подтверждаем. Я добавил три аккаунта, другие, которые нужно, добавляем подобным образом. В окне вы видите мои подключенные аккаунты. Закрываем окно.



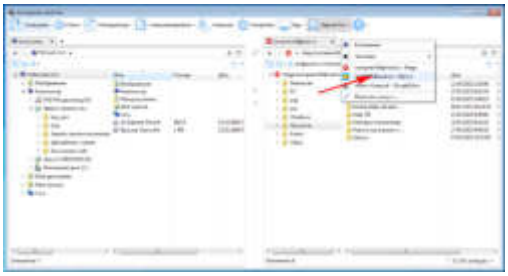
В появившемся окне мы видим подобие файлового менеджера с двумя окнами. В первом выбираем - **Компьютер**, а во втором - **Mega**.



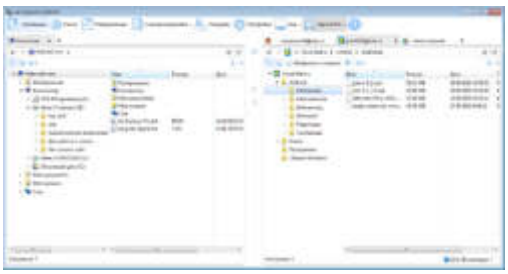
Увидим файлы компьютера в левом окне и закаченные ранее файлы в облако в правом. Все действия я произвожу на виртуальной машине - Virtual Box, поэтому у меня такой скудный запас файлов на компьютере.



Мы можем перетаскивать файлы с облака в компьютер прямо мышкой. Думаю не нужно объяснять как это делать. Давайте подключим еще одно облако Майл. Нажимаем на плюсик в правом окне сверху.

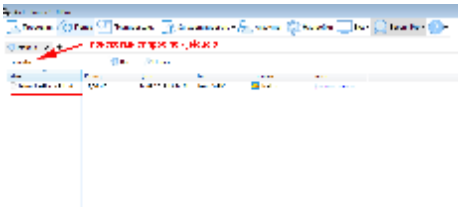


Я сразу подключил и Google Drive. Таким образом, **нажимая на плюсик**, можно добавить все подключенные аккаунты и легко перемещаться по вкладкам, делая копирование, удаление и другие действия, присущие компьютерным.

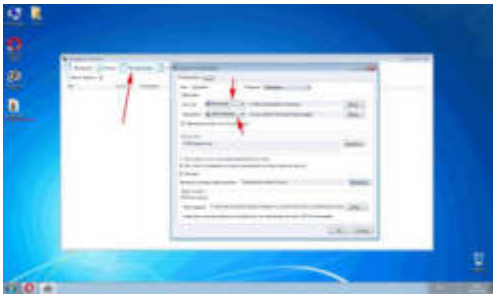


Настройки программы

Ну а теперь коротко рассмотрим основные пункты (вкладки) программы. «**Поиск**» - обычный поиск, можно быстро найти файл по имени. На скриншоте ниже я нашел файл - **JetAudio.apk**.



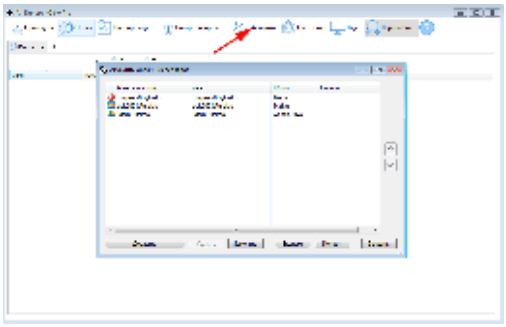
Вкладка - «**Планировщик**» - очень полезна например для системных администраторов. Можно запрограммировать **синхронизацию / Копирование** в определенное время. Интерфейс понятный. Я создал папку - **Архивация** - на рабочем столе и запрограммировал ежедневное копирование в 8-00 утра.



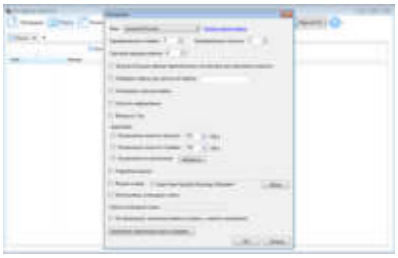
Таким образом, с помощью данной программы можно существенно облегчить труд, например офисных работников, которым ежедневно приходится копировать важные документы, дабы не потерять их в случае сбоя оборудования. Для тех, кто занимается программированием, созданием сайтов или другой творческой работой, в которой иногда каждый написанный символ имеет значение, полезнее будет включить - **синхронизацию** - вместо копирования. Таким образом любые изменения редактируемого документа, мгновенно будут произведены и в облаке, то есть с помощью данной процедуры вы сводите к минимуму потерю ваших данных.

Вкладка - «**Синхронизация**» - запускает копирование данных в планировщике.

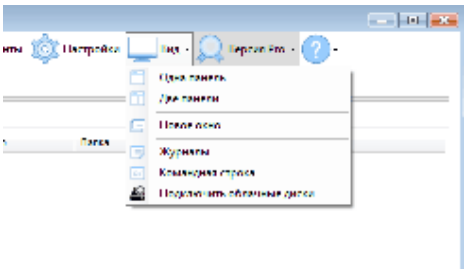
Вкладка - «**Аккаунты**» - появляется окно, куда нужно добавить новый облачный сервис.



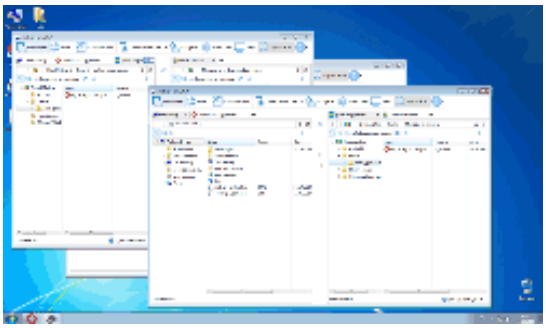
Вкладка - «Настройки» - позволяет манипулировать различными рычагами программы, многие из которых общие для всех программ. Например можно сменить язык интерфейса, поставить пароль на запуск программы и т.д. Иногда полезно поставить галочку - «Показывать скрытые файлы» - чтобы видеть все файлы, которые находятся в папке, чтобы случайно не пропустить что-то. Можно ограничить скорость скачивания / синхронизации, поставив нужно значение и галочку в разделе - «**Speed limits**». В общем настройки каждый подбирает под себя как хочет. В большинстве случаев трогать вообще ничего не нужно, просто оставить все по умолчанию (как есть).



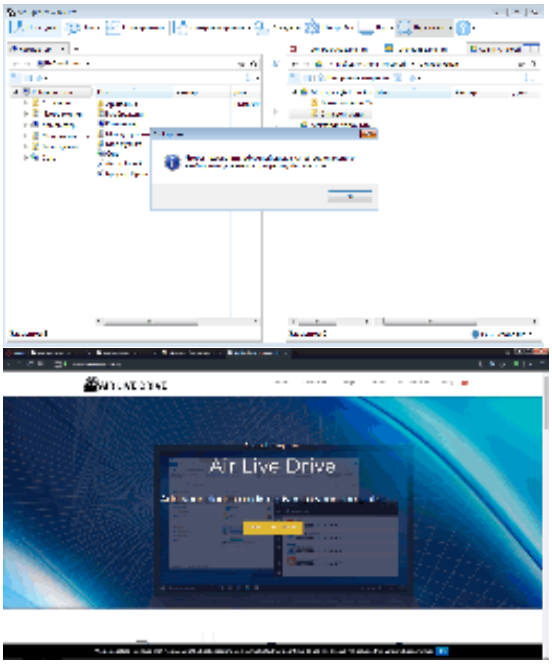
Вкладка - «**Вид**» - позволяет сделать окно программы 2-х оконным, если кому-то не нравится 4-х.



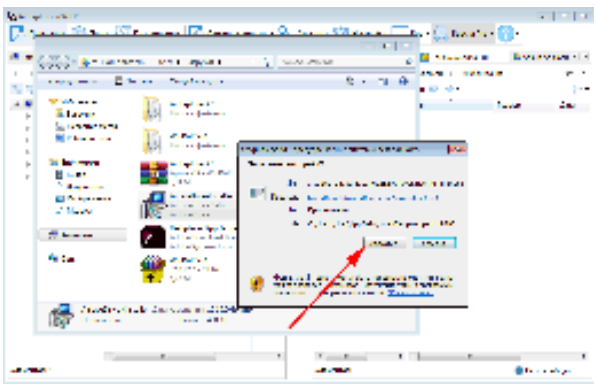
Если выбрать опцию - «**Новое окно**» - вы продублируете окно программы. Таким образом можно копировать, перенося мышкой файлы с одного в другой. Но в большинстве случаев достаточно стандратного окна программы.



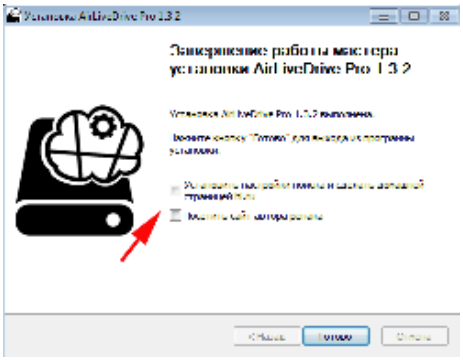
Последняя опция - «**Подключить облачные диски**» - позволяет сделать ваши «**облака**» обычными локальными дисками компьютера. Для этого программа предлагает установить нам утилиту - «**Air Live Drive**» и перебросит вас на сайт, где можно скачать и установить данный софт.



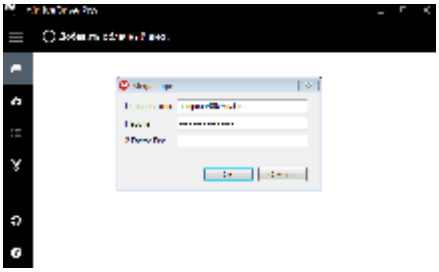
Скачиваете, запускаете программу и ваши облака превратятся в локальные диски.



Может так случится, что стандартный установщик программы выдаст ошибку. Тем более, если не ошибаюсь, там бесплатная версия действует ограниченное время. Поэтому специально для вас я выложил все дополнительные утилиты, если таковые не установлены на вашем ПК. Добро пожаловать на страницу загрузки программы - **AirLiveDrive Pro 1.3.2**. В описании подробно написан порядок установки. Не забудьте только убрать галочки после установки программы, чтобы не менять домашнюю страницу и не переходить на сайт автора.

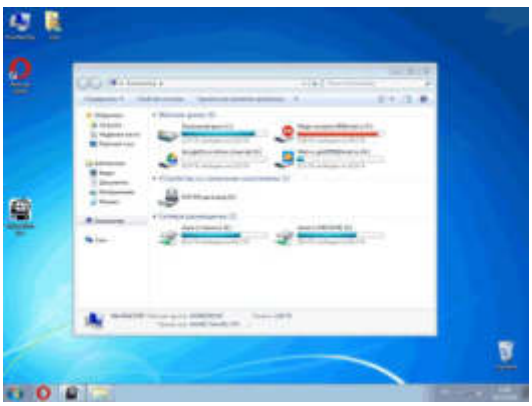


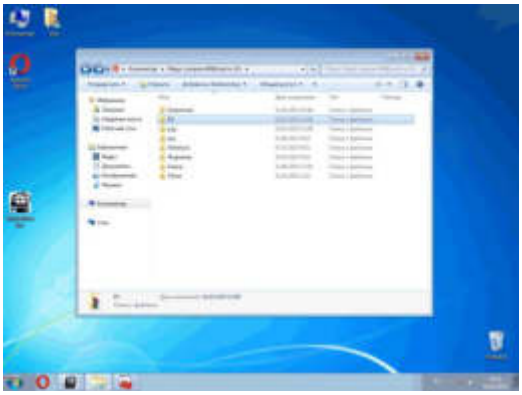
После установки программы нужно подключить аккаунты облачных хранилищ. Процедура аналогична выше приведенной. Также вводим логины и пароли от зарегистрированных ранее облаков.



После авторизации и добавления аккаунтов увидите примерно следущее окно.

Нажимаем везде кнопку - «**Подключить**» и добавляем таким образом локальные диски в систему.





Таким образом вы можете бродить по папкам и запускать видео, читать книги, то есть делать все что обычно используете на обычном компьютере. Очень удобно и быстро.

5. Содержание отчета

Отчет должен содержать:
1.Название работы.
2.Цель работы.
3.Задание и его решение.
4. Вывод по работе

Практическая работа №31
"Создание интерактивной веб-страницы с использованием HTML, CSS и JavaScript" (5 часов)

Цель: **Создание интерактивной веб-страницы с использованием HTML, CSS и JavaScript**

Задание № 1. ***Создание простейшего файла HTML***

- © Создайте личную папку, куда вы будете сохранять все файлы своего сайта.
- © Запустите программу *Блокнот* (*Notepad*).
- © Наберите в окне программы простейший файл HTML.

```
<HTML>
<HEAD>
<TITLE>Учебный файл HTML</TITLE>
</HEAD>
<BODY>
Расписание занятий на вторник
</BODY>
</HTML>
```

e. Сохраните файл под именем **RASP.HTML** (обязательно укажите тип файла HTML при сохранении) в личной папке.

f. Для просмотра Web-страницы используйте любую программу браузера (*Internet Explorer*, *Opera*, *Mozilla Firefox* или другую). Для этого, не покидая программу Блокнот (сверните окно на панель задач), откройте личную папку и двойным кликом по файлу RASP.HTML откройте окно браузера.

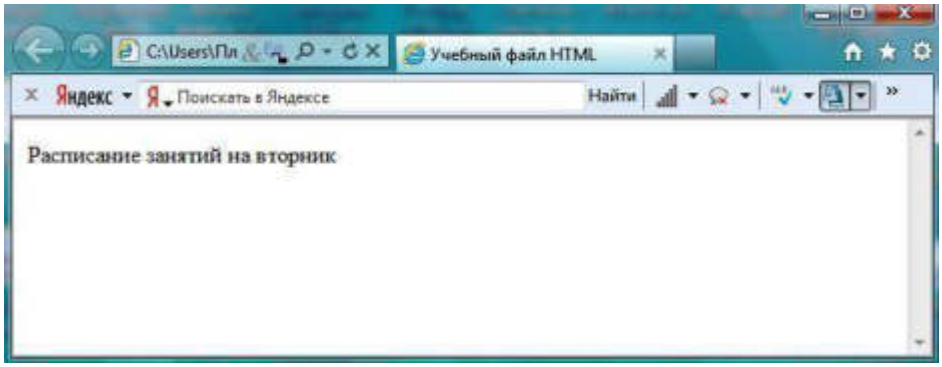


Рис.1

На экране вы увидите результат работы, изображенный на рисунке 1.

Задание № 2. ***Управление расположением текста на экране***

- © При необходимости откройте текст Web-страницы в Блокноте (1 щелчок правой клавишей мыши по файлу RASP.HTML, в контекстном меню выбрать команду *Открыть с помощью...* и выбрать программу *Блокнот*). При необходимости открыть файл в браузере

– двойной клик по значку файла левой клавишей мыши.

- © Внести изменения в файл RASP.HTML, расположив слова *Расписание*, *занятий*, *на вторник* на разных строках.

```
<HTML>
<HEAD>

<TITLE>Учебный файл HTML</TITLE>
</HEAD>
<BODY>
Расписание
занятий
на вторник
</BODY>
</HTML>
```

e. Сохраните текст с внесенными изменениями в файле RASP.HTML (меню Файл | Сохранить). Если у вас уже отображается Web-страница, то вам достаточно переключиться на панели задач на программу браузера и обновить эту страницу (кнопка). Изменилось ли отображение текста на экране?

Не удивляйтесь тому, что внешний вид вашей Web-страницы не изменился.

Не забывайте каждый раз сохранять текст Web-страницы при ее корректировке в программе *Блокнот* и обновлять страницу при ее просмотре в программе браузера.

Задание № 3. ***Некоторые специальные команды форматирования текста***

Существуют специальные команды, выполняющие перевод строки и задающие начало нового абзаца. Кроме того, существует команда, запрещающая программе браузера изменять каким-либо образом изменять форматирование текста и позволяет точно воспроизвести на экране заданный фрагмент текстового файла.

Тег перевода строки **
** отделяет строку от последующего текста или графики.

Тег абзаца **<P>** тоже отделяет строку, но еще добавляет пустую строку, которая зрительно выделяет абзац.

Оба тега являются одноэлементными, тег <P> – двойной, т.е. требуется закрывающий тег.

g. Внесите изменения в файл RASP.HTML

```
<HTML>
<HEAD>
<TITLE>Учебный файл HTML</TITLE>

</HEAD>
<BODY>
<P>Расписание</P>
<BR>занятий<BR>

на вторник
</BODY>
</HTML>
```

© Сохраните внесенные изменения, переключитесь на панели задач на программу браузера, обновите Web-страницу.
Как изменилось отображение текста на экране? Выглядеть ваша Web-страница будет примерно так, как показано на рисунке 2.

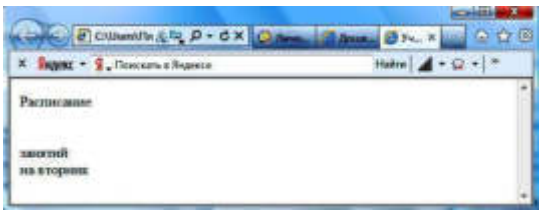


Рис. 2

Задание № 4. ***Выделение фрагментов текста***

h. Внести изменения в текст файла RASP.HTML

```
<HTML>

<HEAD>
<TITLE>Учебный файл HTML</TITLE>
</HEAD>
<BODY>

<B>Расписание</B>
<I> занятий</I>
<U> на вторник</U>

</BODY>
</HTML>
```

i. Посмотрите полученную Web-страницу.

Возможно использование комбинированных выделений текста.

<I><B>Расписание</B></I> <I><U> занятий</U></I> <U> на вторник</U>

Но при этом необходимо помнить следующее правило использования комбинированных тегов:

<Тег_1><Тег_2> ... </Тег_2></Тег_1> – ***правильная запись***.

<Тег_1><Тег_2> ... </Тег_1></Тег_2> – ошибочная запись.

Обратите внимание на «вложенность» тегов, она напоминает «вложенность» скобок.

Задание № 5. ***Задание размеров символов Web-страницы***

Существует два способа управления размером текста, отображаемого браузером:

© использование стилей заголовка,

© задание размера шрифта основного документа или размера текущего шрифта. Используется шесть тегов заголовков: от **<H1>** до **<H6>** (тег двойной, т.е. требует закрытия). Каждому тегу соответствует конкретный стиль, заданный параметрами настройки браузера.

ii. Внесите изменения в файл RASP.HTML

```
<HTML>
<HEAD>

<TITLE>Учебный файл HTML</TITLE>
```

```
</HEAD>
<BODY>
<P><H1>Расписание</H1></P>

<I> занятий</I><U> на вторник</U> </BODY>
</HTML>
```

1. Просмотрите свою Web-страницу. На экране вы увидите то, что отображено на рисунке 3.

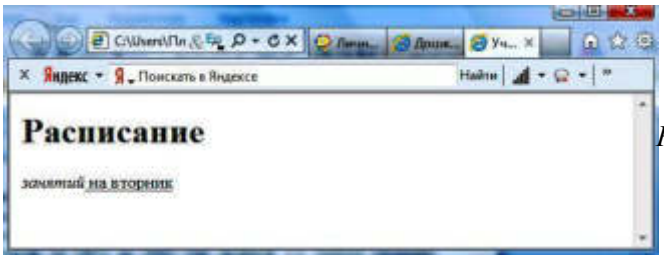


Рис. 3

Задание № 6. ***Установка размера текущего шрифта***

Тег шрифта **<FONT>** позволяет задавать размер текущего шрифта в отдельных местах текста в диапазоне от 1 до 7.

7. Внесите изменения в текст RASP.HTML

```
<HTML>
<HEAD>
<TITLE>Учебный файл HTML</TITLE>
</HEAD>
<BODY>
<FONT SIZE="7">Расписание</FONT>
занятий на вторник
</BODY>
</HTML>
```

6. Самостоятельно измените размер текста «занятий на вторник», используя тег <FONT>.

7. Измените оформление текста HTML-документа, используя тег выделения фрагментов и тег перевода строки и абзаца.

Задание № 7. ***Установка гарнитуры и цвета шрифта***

Тег **<FONT>** предоставляет возможности управления гарнитурой, цветом и размером текста. Изменение гарнитуры текста выполняется простым добавлением к тегу **<FONT>** атрибута **FACE**. Например, для отображения текста шрифтом *Arial* необходимо записать:

<FONT FACE=” ARIAL”>

Для изменения цвета шрифта можно использовать в теге **<FONT>** атрибут **COLOR=”X”**. Вместо **“X”** надо подставить английское название цвета в кавычках (“ ”), либо его шестнадцатеричное значение. При задании цвета шестнадцатеричным числом необходимо представить этот цвет разложенным на три составляющие: красную (*R – Red*), зелѐную (*G – Green*), синюю (*B – blue*), каждая из которых имеет значение от **00** до **FF**. В этом случае мы имеем дело с так называемым форматом **RGB**.

Примеры записи текста в формате **RGB** приведены в Таблице 1:

Таблица 1

Запись текста в формате RGB

Цвет	RRGGBB	Цвет	RRGGBB	Цвет	RRGGBB
Black Черный	000000	Purple Фиолетовый	FF00FF	Green Зеленый	00FF00
White Белый	FFFFFF	Yellow Желтый	FFFF00	Azure Бирюзовый	00FFFF

Red Красный		FF0000	Brown Коричневый		996633	Blue Синий	0000FF
Orange Оранжевый		FF8000	Violet Лиловый		B000FF	Gray Серый	A0A0A0

2. Внесите изменения в файл RASP.HTML

```
<HTML>
<HEAD>
<TITLE>Учебный файл HTML</TITLE>
</HEAD>
<BODY>
<U><I><B><FONT COLOR="#FF0000" FACE="ARIAL" SIZE="7"> Расписание</FONT></B></I></U>
занятий на вторник
</BODY>
</HTML>
```

o. Самостоятельно измените размер, цвет, гарнитуру стиль текста документа.

Задание № 8. ***Выравнивание текста по горизонтали***

© Внесите изменения в файл RASP.HTML

```
<HTML>
<HEAD>
<TITLE>Учебный файл HTML</TITLE>
</HEAD>
<BODY>
<P ALIGN="CENTER">
<FONT COLOR="#008080" SIZE="7">
<B>Расписание</B></FONT><BR>
<FONT SIZE="6"><I> занятий на вторник</I></FONT> </P>
</BODY>
</HTML>
```

s. Просмотрите изменения в браузере. На экране вы увидите то, что показано на рисунке 4.

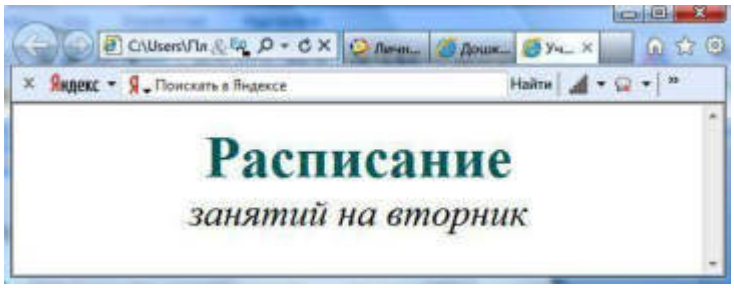


Рис. 4

Задание № 9. ***Задание цвета фона и текста***

При изображении фона и цвета браузеры используют цвета, установленные по умолчанию, – они заданы параметрами настройки браузера. Если вы хотите задать другие цвета, то это надо сделать в начале файла HTML в теге **<BODY>**. Атрибут **BGCOLOR=** определяет цвет фона страницы, атрибут **TEXT=** задает цвет текста для всей страницы, атрибуты **LINK=** и **VLINK=** определяют соответственно цвета не просмотренных и просмотренных ссылок (последние два примера будут рассмотрены позже).

b. Внесите изменения в файл RASP.HTML

```
<HTML>
<HEAD>
<TITLE>Учебный файл HTML</TITLE> </HEAD>
<BODY BGCOLOR="#FFFFCC" TEXT="#330066"> <P ALIGN="CENTER">
<FONT COLOR="#008080" SIZE="7">
<B>Расписание</B></FONT><BR>
<FONT SIZE="6"><I> занятий на вторник</I></FONT> </P>
</BODY>
</HTML>
```

© Просмотрите изменения Web-страницы в браузере.

Задание № 10. ***Размещение графики на Web-странице***

Тег **<IMG>** позволяет вставить изображение на Web-страницу. Оно появится в том месте документа, где находится этот тег. Тег **<IMG>** является одиночным.

Необходимо помнить, что графические файлы должны находиться в той же папке, что и файл HTML, описывающий страницу. Графика в Web, как правило, распространяется в трех форматах: GIF, JPG, PNG. Для выполнения следующего задания поместите файл с именем CLOCK.JPG (или другим именем) в рабочую папку.

Следует помнить, что для браузера важно, в каком регистре вы задаете описание имени и типа файла. Выработайте для себя определенное правило и строго следуйте ему. Если вы размещаете файл графического изображения во вложенной папке, то при описании изображения необходимо указывать путь доступа к файлу изображения, отображая вложенность папок.

c. Внесите изменения в файл RASP.HTML

```
<HTML>
<HEAD>
<TITLE>Учебный файл HTML</TITLE> </HEAD>
<BODY BGCOLOR="#FFFFCC" TEXT="#330066"> <P ALIGN="CENTER">
<FONT COLOR="#008080" SIZE="7">
<B>Расписание</B></FONT><BR>
<FONT SIZE="6"><I> занятий на вторник</I></FONT> <BR><BR>
<IMG SRC="CLOCK.PNG">
</P>
</BODY>
</HTML>
```

3. Просмотрите изменения вашей Web-страницы в браузере.

На экране вы увидите те, что показано на рисунке 5.

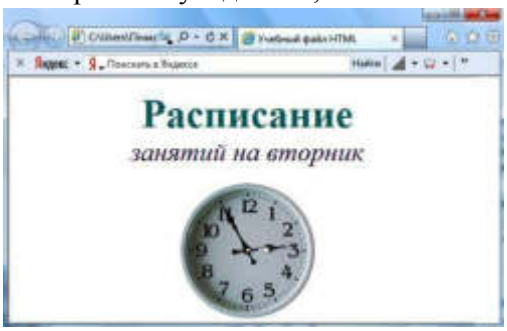


Рис. 5

Тег <IMG> имеет немало атрибутов, описанных в таблице 2. Эти атрибуты можно задавать дополнительно и располагаться они могут в любом месте тега после кода **IMG**.

Таблица 2

Атрибуты изображения

Атрибут	Формат	Описание
ALT	<IMG SRC="CLOCK.JPG" ALT="Картинка">	Задает текст, заменяющий изображение в том случае, если браузер не воспринимает изображение
BORDER	<IMG SRC="CLOCK.JPG" BORDER="3">	Задает толщину рамки вокруг изображения. Измеряется в пикселях
ALIGN	<IMG SRC="CLOCK.JPG" ALIGN="TOP">	Задает выравнивание изображения относительно текста: l относительно текста выровнена верхняя часть изображения – **"TOP"**, l относительно текста выровнена нижняя часть изображения – **"BOTTOM"**, l относительно текста выровнена средняя часть изображения – **"MIDDLE"**.

HEIGHT	<IMG SRC="CLOCK.JPG" HEIGHT="111">	Задает вертикальный размер изображения внутри окна браузера
WIDTH	<IMG SRC="CLOCK.JPG" WIDTH="210">	Задает горизонтальный размер изображения внутри окна браузера
VSPACE	<IMG SRC="CLOCK.JPG" VSPACE="8">	Задает добавление верхнего и нижнего пустых полей
HSPACE	<IMG SRC="CLOCK.JPG" HSPACE="8">	Задает добавление левого и правого пустых полей

Задание № 11. ***Использование атрибутов изображения***

- © Самостоятельно внесите изменения в текст файла RASP.HTML: опробуйте использование таких атрибутов графики, как **ALT**, **BORDER**, **ALIGN**, **HEIGHT**, **WIDTH**, **VSPACE**, **HSPACE**.

Всегда обращайте внимание на размер графического файла (в байтах), так как это влияет на время загрузки Web-страницы.

2. Просмотрите изменения вашей Web-страницы в браузере.

Задание № 12. ***Установка фонового изображения на Web-странице***

Фоновое изображение – это графический файл с небольшим рисунком, который многократно повторяется, заполняя все окно браузера независимо от его размеров. Графика, используемая в качестве фоновой, задается в теге **<BODY>**.

1. Внесите изменения в файл RASP.HTML, предварительно подготовив и сохранив в рабочей папке графический файл фонового рисунка (FON.PNG).

```
<HTML>
<HEAD>
<TITLE>Учебный файл HTML</TITLE> </HEAD>
<BODY BACKGROUND="FON.PNG" TEXT="#330066"> <P ALIGN="CENTER">
<FONT COLOR="#008080" SIZE="7">
<B>Расписание</B></FONT><BR>
<FONT SIZE="6"><I> занятий на вторник</I></FONT> <BR><BR>
<IMG SRC="CLOCK.PNG" ALIGN="MIDDLE"> </P>
</BODY>
</HTML>
```

На экране вы увидите то, что изображено на рисунке 6.

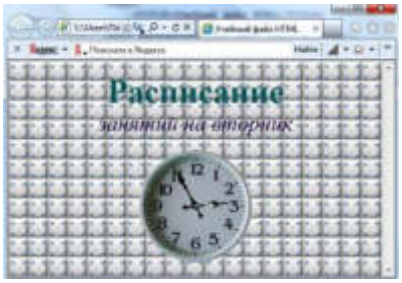


Рис. 6

Рисунок, который использовался в качестве фонового, имеет вид

Рис. 7

2. Поэкспериментируйте с фоновым рисунком Web-страницы и выберите оптимальный с вашей точки зрения.

Задание № 13. ***Создание таблицы***

Таблица является частью HTML-документа. Она представляет собой прямоугольную сетку, состоящую из вертикальных столбцов и горизонтальных строк. Пересечение строки и столбца называется ячейкой таблицы. Ячейка может содержать в себе текст, графику или другую таблицу.

Таблица состоит из трех основных частей:

- названия таблицы,
- заголовков столбцов,
- ячеек таблицы.

Таблица в Web-документе заполняется по строкам (слева направо по строке, затем переход на новую строку). **Каждая ячейка таблицы должна быть заполнена** (хотя бы пробелом, которые используются для создания пустых ячеек).

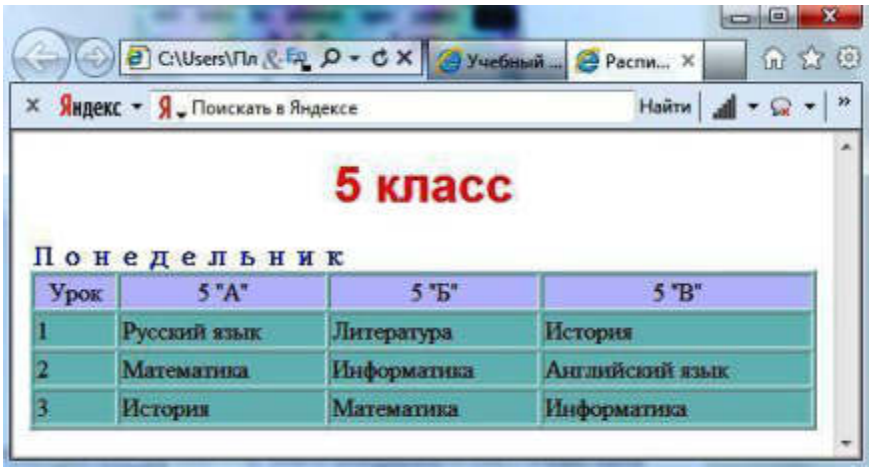
1. Запустите программу *Блокнот* и наберите текст следующей Web-страницы. Применяйте приемы копирования при создании таблицы, работая в программе *Блокнот*.

```
<HTML>
<HEAD>
<TITLE>Расписание занятий 5 классов</TITLE> </HEAD>
<BODY BGCOLOR="FFFFFF">
<P ALIGN="CENTER">
<FONT COLOR="RED" SIZE="6" FACE="ARIAL"> <B>5
класс</B></FONT><BR></P>
<FONT COLOR="BLUE" SIZE="4" FACE="COURIER">
<B>Понедельник</B></FONT><BR>
                        <TABLE BORDER="1" WIDTH=100% BGCOLOR="#99CCCC">
                                <TR BGCOLOR="#CCCCFF" ALIGN="CENTER">
<TD>Урок</TD> <TD>5 "А"</TD> <TD>5 "Б"</TD>

<TD>5 "В"</TD>
</TR>
<TD>1</TD> <TD>Русский язык</TD>
<TD>Литература</TD> <TD>История</TD> </TR>
<TD>2</TD> <TD>Математика</TD>
<TD>Информатика</TD> <TD>Английский язык</TD> </TR>
<TD>3</TD> <TD>История</TD>
<TD>Математика</TD> <TD>Информатика</TD> </TR>
</TABLE>
</BODY>
</HTML>
```

2. Сохраните файл в личной рабочей папке под именем 5.HTML
3. Для просмотра созданной Web-страницы в окне личной рабочей папки двойным щелчком левой клавиши мыши загрузите браузер.

На экране вы увидите то, что показано на рисунке 8.



Задание № 14. ***Построение гипертекстовых связей***

Важнейшим средством языка HTML является возможность включения в документ ссылок на другие документы.
Возможны ссылки:

- на удаленный HTML-файл,
- на некоторую точку в текущем HTML-документе,
- на любой файл, не являющийся HTML-документом.

В качестве ссылки можно использовать любой текст или графику.
Ссылки в пределах одного документа
Такие ссылки требуют двух частей: метки и самой ссылки. Метка определяет точку, на которую происходит переход по ссылке. Ссылка использует имя метки. Ссылки выделяют цветом или подчеркиванием в зависимости от того, как настроен браузер. Для изменения цвета ссылки используются атрибуты **LINK=** и **VLINK=** тега **<BODY>**.
Описание ссылки
<A HREF="#ПН">Понедельник</A>
Перед именем метки (ПН), указывающей, куда надо перейти по ссылке, ставится символ #. Между символами “>” и “<” располагается текст (“Понедельник”), на котором должен быть произведен щелчок для перехода по ссылке.
Определим метку
<A NAME="ПН">Понедельник</A>

1. Дополните файл 5.HTML описанием таблицы, содержащей названия дней недели, поместив его в начало Web-страницы.

```
<TABLE WIDTH=100%>

<TR>
<TD>Понедельник</TD>
<TD>Вторник</TD>
<TD>Среда</TD>
<TD>Четверг</TD>
<TD>Пятница</TD>
<TD>Суббота</TD>
</TR>
</TABLE>
<BR>
```

...Вставьте в файл 5.HTML метку, указывающую на понедельник.

```
<FONT COLOR="BLUE" SIZE="4" FACE="COURIER"><B>
<A NAME="ПН">Понедельник</A></B></FONT><BR>
```

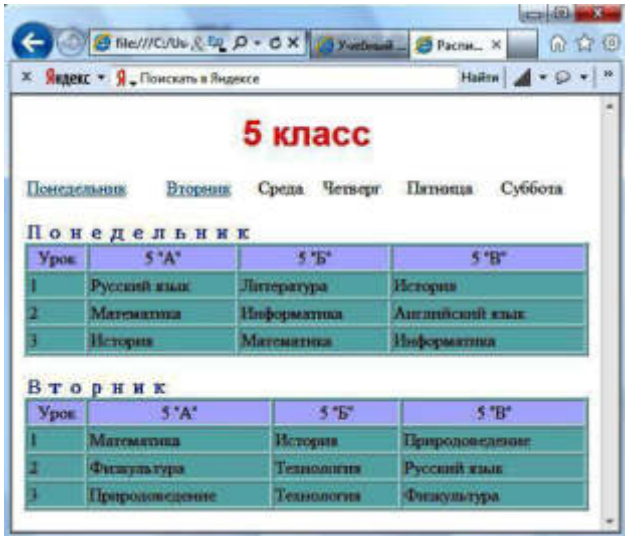
3. Вставьте в таблицу с названиями дней недели ссылку для выбранной метки:

```
<TABLE WIDTH=100%>
<TR>
<TD><A HREF="#ПН">Понедельник</A></TD>
<TD>Вторник</TD>

<TD>Среда</TD>
...
```

4. Создайте таблицы расписаний для других дней недели.

5. Сохраните файл 5.HTML в личной рабочей папке.
6. Просмотрите полученную Web-страницу.



Задание № 15. ***Создание ссылки на другой HTML-документ***

Ссылки позволяют щелчком на выделенном слове или фразе перейти к другому файлу.
Опишем ссылку:

<A HREF="5.HTML">5 класс</A>

После имени файла (5.HTML) между символами «>» и «<» располагается текст («5 класс»), на котором должен быть произведен щелчок для перехода к этому файлу.

1. Внесите изменения в файл RASP.HTML

```
<HTML>
<HEAD>
<TITLE>Учебный файл HTML</TITLE> </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="#330066"> <P ALIGN="CENTER">
<FONT COLOR="#008080" SIZE="7">
<B>Расписание</B></FONT><BR>
<FONT SIZE="6"><I> занятий на вторник</I></FONT> <BR><BR>
<IMG SRC="CLOCK.PNG" ALIGN="TOP">
</P>
<CENTER>
<TABLE WIDTH=60%>
<TR><TD><A HREF="5.HTML">5 класс</A></TD> <TD>6 класс</TD> </TR> <TR><TD>7
класс</TD>
<TD>8 класс</TD>                                    </TR>
<TR><TD>9 класс</TD>
<TD>10 класс</TD>                                    </TR>
<TR><TD>11 класс</TD>                                    </TR>

</TABLE>
</CENTER>
</BODY>
</HTML>
```

2. Сохраните файл RASP.HTML
3. Просмотрите полученную Web-страницу.

На экране вы увидите то, что изображено на рисунке 10.

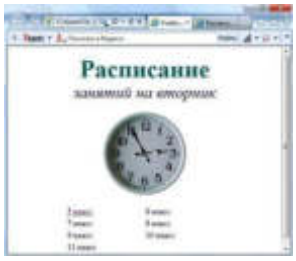


Рис. 10

Задание № 17. ***Самостоятельное итоговое задание***
Разработайте Web-страницы, рассказывающие о вашем классе. На головной странице разместите рассказ о классе, классном руководителе. Рассказы об учениках разместите на отдельных Web-страницах. Укажите ссылки на страницы учеников с головной Web-страницы. Не забудьте разместить ссылки возврата на головную страницу.

Как подготовить хорошую Web-страницу

1. Следует обратить внимание на простоту и логичность расположения информации на ваших страницах. Один из способов сделать информацию более легкой для восприятия – оставить на странице достаточно свободного места, не содержащего ни текста, ни рисунков. Страница, содержащая много информации, только отпугнет посетителя.

2. Постарайтесь представить информацию в виде списков или таблиц так, чтобы можно было достаточно легко найти важные сведения.

3. Не размещайте одно изображение сразу за другим. Попробуйте распределить их по документу, оставив достаточно свободного пространства.

4. Информация должна размещаться частями, легкими для восприятия. Обратите внимание на длину абзацев. Если абзац слишком длинный, разбейте его на несколько небольших абзацев.

5. Если Web-страница имеет большой объем, то, возможно, вам следует вставить ссылки, позволяющие пользователю быстро перемещаться между частями одного документа. Иногда имеет смысл вместо одного длинного документа подготовить одну страницу, содержащую перечень тем, каждую из которых раскрыть на отдельной Web-странице, и установить ссылки на соответствующие Web-страницы.

6. Использование графики может дополнительно привлечь пользователей. Но необходимо помнить о времени загрузки вашей страницы, которое определяется количеством и объемом графической информации. Красивая картинка не произведет никакого впечатления, ели для того, чтобы ее увидеть, придется долго ждать, пока она загрузится.

5. Содержание отчета

Отчет должен содержать:
1.Название работы.
2.Цель работы.
3.Задание и его решение.
4. Вывод по работе

Практическая работа № 32

Владение различными методами поиска информации в Интернет (по ключевым словам, с помощью каталогов) (1 час)

Цель: научиться пользоваться электронными словарями, изучить методы поиска нужных слов в электронных словарях; познакомиться с основными возможностями программы-браузера Internet Explorer; освоить навыки работы с программой-браузером, изучение процесса поиска информации в Интернет, зная адрес страницы, на которой она расположена.

Оснащение: персональный компьютер, конспекты, карточки-задания, учебники.

ХОД РАБОТЫ

Теоретическая часть:

Для связи с Internet используется специальная программа – браузер.

Сегодня наиболее популярными браузерами являются Opera, Google, Firefox и Internet Explorer. Основное назначение браузера – загрузка Веб-страницы из Интернета и отображение её на экране.

Служба Worldwide Web (WWW) – это единое информационное пространство, состоящее из сотен миллионов взаимосвязанных электронных документов. Отдельные документы, составляющие пространство Web, называют Web-страницами. Группы тематически объединенныхWeb-страниц, называют Web-узлами (сайтами).

Адресация в сети:

IP-адрес	DNS-адрес
Состоит из четырех блоков цифр, разделенных точками. Он может иметь такой вид: 32.87.56.5	http://www.yandex.ru http – протокол передачи данных www – сервер находится на WWW yandex – название сервера ru – сервер находится в России Часть адреса разделенная точками называется доменом.

Если вы изучаете иностранный язык, сталкиваетесь с переводом текстов с иностранного языка или ведете переписку с зарубежными друзьями, то без помощи словаря вам не обойтись. Для того, чтобы найти определение какого-либо термина, мы также можем воспользоваться услугами электронных словарей. Если рассматривать такой вариант, когда у вас нет возможности заглянуть в книгу, существуют электронные словари. Некоторые вещи, как например, поиск в огромных массивах текста, они делают удивительно быстро.

Практическая часть 1:

В тетрадь запишите следующую таблицу:

Адрес сайта	Назначение	Страна
help.belhost.by		
www.national-lottery.co.uk		
index.all-hotels.in.ua		
www.microsoft.com		
www.house.gov		
acorda.kz		
britain.uz		
klassica.ru		
en.beijing2008.cn		

Рассмотрите веб-страницы, определите назначение сайта, определите государство, в котором сделан этот сайт.

Откройте Веб страницу с адресом: www.detstvo.ru. Найдите гиперссылку праздники, нажмите на нее, дождитесь загрузки страницы, перепишите ее адрес. Далее найдите ссылку фото, перепишите и ее адрес.

Ответьте на вопрос: каким образом адресуются страницы одного сайта?

Практическая часть 2:

Откройте программу Enternet Explorer. Загрузите страницу электронного словаря В. Даля – www.slovardalya.ru. В текстовое поле **Поиск по словарю:** введите слово, лексическое значение которого вам нужно узнать, например, «рутина». Нажмите на кнопку**Искать**. Дождитесь результата поиска. Запишите результат к себе в тетрадь.

Самостоятельно найдите лексическое значение следующих слов и запишите результат к себе в тетрадь.

- Новости,
- Статья,

- Учитель,
- Техника,
- Команда.

Найдите лексические значения и этимологию следующих слов и запишите результат к себе в тетрадь:

- Метонимия,
- Видеокарта,
- Железо,
- Папирус,
- Скальпель,
- Дебет (бухг. термин).

Практическая часть 3:

Используя, словарь-переводчик сделайте перевод следующих слов на английский, испанский и японский язык. Вы можете воспользоваться любыми словарями из предложенного списка:

Массив	**Коммуникация**
Функция	**Алгоритм**
Диаграмма	**Аксиома**
Логика	**Сервис**
Память	**Матрица**
Графика	**Информация**
Таблица	
Файл	
Программа	
Вирус	
Мультимедиа	
Модель	
Формализация	
Топология	

Толковые словари	**Словари-переводчики**
www.golossary.ru	Lingvo.yandex.ru
Slovari.gramota.ru	translate.google.ru
www.efremova.info	
Mega.km.ru	
www.ozhegov.org	
www.golossary.ru	
Slovari.gramota.ru	
www.efremova.info	
www.chtotakoe.info	
www.slovari.ru	

5. Содержание отчета

Отчет должен содержать:

1.Название работы.

2.Цель работы.

3.Задание и его решение.

4. Вывод по работе