

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»  
Кафедра «Техносферная и информационная безопасность»**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ  
ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

**Б1.Б.05. Информатика**

**Направление подготовки (специальность) 20.03.01 Техносферная  
безопасность**

**Профиль образовательной программы Безопасность жизнедеятельности в  
техносфере**

**Форма обучения очная**

## СОДЕРЖАНИЕ

<b>1</b>	<b>Конспект лекций</b>	<b>4</b>
1.1	Лекция №1 <i>Информатика как наука</i>	4
1.2	Лекция №2 <i>Аппаратные средства реализации информационных процессов</i>	10
1.3	Лекция №3 <i>Состав и назначение основных элементов компьютера</i>	15
1.4	Лекция №4 <i>Классификации периферийных устройств, устройство и их назначение</i>	19
1.5	Лекция №5 <i>Программное обеспечение для компьютера</i>	21
1.6	Лекция №6 <i>Понятие информации</i>	26
1.7	Лекция №7 <i>Общая характеристика информационных процессов</i>	27
1.8	Лекция №8 <i>Системы исчисления</i>	32
1.9	Лекция №9 <i>Арифметические действия в различных системах исчисления</i>	33
1.10	Лекция №10 <i>Основы теории защиты информации</i>	35
1.11	Лекция №11 <i>Методы защиты информации</i>	38
1.12	Лекция №12 <i>Алгоритмизация</i>	41
1.13	Лекция №13 <i>Основные алгоритмические структуры</i>	43
1.14	Лекция №14 <i>Моделирование как метод познания</i>	44
1.15	Лекция №15 <i>Программирование</i>	50
1.16	Лекция №16 <i>Классификация языков программирования</i>	54
1.17	Лекция №17 <i>Основные понятия алгоритмического языка Pascal</i>	62
1.18	Лекция №18 <i>Программная реализация основных алгоритмических и циклических структур на языке Pascal</i>	66
1.19	Лекция №19 <i>Локальные и глобальные сети ЭВМ</i>	72
1.20	Лекция №20 <i>Компоненты компьютерных сетей</i>	77
1.21	Лекция №21 <i>Глобальная компьютерная сеть Интернет</i>	82
<b>2</b>	<b>Методические указания по выполнению лабораторных работ</b>	<b>87</b>
2.1	Лабораторная работа №1 <i>Элементы рабочего окна в Windows. Технология работы с окнами в Windows</i>	87
2.2	Лабораторная работа №2 <i>Технология работы с файлами и папками в Windows</i>	93
2.3	Лабораторная работа №3 <i>Технология работы с файлами и папками в Windows commander</i>	100
2.4	Лабораторная работа №4 <i>Технология работы в Microsoft Word. Набор, редактирование и форматирование текста</i>	108
2.5	Лабораторная работа №5 <i>Технология работы в текстовом процессоре Microsoft Word. Создание нумерованных и маркированных списков</i>	131
2.6	Лабораторная работа №6 <i>Технология работы с таблицами в Microsoft Word. Ввод и редактирование формул</i>	136
2.7	Лабораторная работа №7 <i>Создание гипертекста</i>	143
2.8	Лабораторная работа №8 <i>Создание графических объектов в текстовом процессоре Microsoft Word</i>	145
2.9	Лабораторная работа №9 <i>Создание презентаций в Power Point на базе шаблонов, использование эффектов и анимации</i>	150
2.10	Лабораторная работа №10 <i>Создание и форматирование HTML-документов</i>	157
2.11	Лабораторная работа №11 <i>Добавление графических объектов и анимации в HTML-документы</i>	161
2.12	Лабораторная работа №12 <i>Создание и форматирование таблиц</i>	163
2.13	Лабораторная работа №13 <i>Создание и заполнение базы данных в Microsoft Access. Связывание таблиц, создание форм, запросов, отчетов</i>	165

<b>2.14</b>	<b>Лабораторная работа №14</b> <i>Интерфейс Microsoft Excel. Набор текстовой информации в Excel. Ввод и редактирование формул в Microsoft Excel. Построение диаграмм. Логические функции. Создание составных интегрированных документов</i>	175
<b>2.15</b>	<b>Лабораторная работа №15</b> <i>Основы программирования на языке Pascal. Линейный алгоритмы</i>	181
<b>2.16</b>	<b>Лабораторная работа №16</b> <i>Основы программирования на языке Pascal. Логические функции</i>	184
<b>2.17</b>	<b>Лабораторная работа №17</b> <i>Основы программирования на языке Pascal. Разветвляющие алгоритмы</i>	185
<b>2.18</b>	<b>Лабораторная работа №18</b> <i>Основные математические вычисления в MathCad</i>	187
<b>3</b>	<b>Методические указания по проведению практических занятий</b>	192
<b>3.1</b>	<b>Практическое занятие №1 ПЗ-1</b> <i>Системы счисления</i>	192
<b>3.2</b>	<b>Практическое занятие №2 ПЗ-2</b> <i>Основные понятия и программная реализация алгоритмического языка Pascal</i>	194
<b>4</b>	<b>Методические указания по проведению семинарских занятий</b>	195

# 1. КОНСПЕКТ ЛЕКЦИЙ

## 1. 1 Лекция №1 (2 часа)

Тема: «Информатика как наука»

### 1.1.1 Вопросы лекции:

1. Что такое информатика.
2. Основные этапы развития информатики и вычислительной техники.
3. Поколения ЭВМ.

### 1.1.2 Краткое содержание вопросов:

1. Что такое информатика

Термин "**информатика**" (франц. informatique) происходит от французских слов *information* (информация) и *automatique* (автоматика) и дословно означает "**информационная автоматика**".

Широко распространён также англоязычный вариант этого термина — "**Computer science**", что означает буквально "**компьютерная наука**".

**Информатика** — это основанная на использовании компьютерной техники дисциплина, изучающая структуру и общие свойства информации, а также закономерности и методы её создания, хранения, поиска, преобразования, передачи и применения в различных сферах человеческой деятельности.

В 1978 году международный научный конгресс официально закрепил за понятием "**информатика**" области, связанные с разработкой, созданием, использованием и материально-техническим обслуживанием систем обработки информации, включая компьютеры и их программное обеспечение, а также организационные, коммерческие, административные и социально-политические аспекты компьютеризации — массового внедрения компьютерной техники во все области жизни людей.

Таким образом, информатика базируется на компьютерной технике и немыслима без нее.

Информатика — комплексная научная дисциплина с широчайшим диапазоном применения.

Её приоритетные направления:

- **разработка вычислительных систем и программного обеспечения;**
- **теория информации**, изучающая процессы, связанные с передачей, приёмом, преобразованием и хранением информации;
- **математическое моделирование**, методы вычислительной и прикладной математики и их применение к фундаментальным и прикладным исследованиям в различных областях знаний;
- **методы искусственного интеллекта**, моделирующие методы логического и аналитического мышления в интеллектуальной деятельности человека (логический вывод, обучение, понимание речи, визуальное восприятие, игры и др.);
- **системный анализ**, изучающий методологические средства, используемые для подготовки и обоснования решений по сложным проблемам различного характера;
- **биоинформатика**, изучающая информационные процессы в биологических системах;
- **социальная информатика**, изучающая процессы информатизации общества;
- **методы машинной графики, анимации, средства мультимедиа;**
- **телекоммуникационные системы и сети**, в том числе, глобальные компьютерные сети, объединяющие всё человечество в единое информационное сообщество;
- **разнообразные приложения**, охватывающие производство, науку, образование, медицину, торговлю, сельское хозяйство и все другие виды хозяйственной и общественной деятельности.

Российский академик А.А. Дороднин выделяет в информатике три неразрывно и существенно связанные части — **технические средства, программные и алгоритмические**.

**Технические средства**, или аппаратура компьютеров, в английском языке обозначаются словом **Hardware**, которое буквально переводится как "твёрдые изделия".

Для обозначения **программных средств**, под которыми понимается **совокупность всех программ, используемых компьютерами, и область деятельности по их созданию и применению**, используется слово **Software** (буквально — "мягкие изделия"), которое подчеркивает равнозначность самой машины и программного обеспечения, а также способность программного обеспечения модифицироваться, приспособливаться и развиваться.

Программированию задачи всегда предшествует **разработка способа ее решения в виде последовательности действий, ведущих от исходных данных к искомому результату**, иными словами, **разработка алгоритма решения задачи**. Для обозначения части информатики, связанной с разработкой алгоритмов и изучением методов и приемов их построения, применяют термин **Brainware** (англ. brain — интеллект).

**Роль информатики в развитии общества чрезвычайно велика. С ней связано начало революции в области накопления, передачи и обработки информации. Эта революция, следующая за революциями в владении веществом и энергией, затрагивает и коренным образом преобразует не только сферу материального производства, но и интеллектуальную, духовную сферы жизни.**

Прогрессивное увеличение возможностей компьютерной техники, развитие информационных сетей, создание новых информационных технологий приводят к значительным изменениям во всех сферах общества: в производстве, науке, образовании, медицине и т.д.

## 2. Основные этапы развития информатики и вычислительной техники

1. Ручной период автоматизации вычислений начался на заре человеческой цивилизации. Он базировался на использовании пальцев рук и ног. Счет с помощью группировки и перекладывания предметов явился предшественником счета на абаке — наиболее развитом счетном приборе древности. Аналогом абака на Руси являются счеты. В начале 17 века Дж. Непер ввел логарифмы, что оказало революционное влияние на счет. Изобретенная им логарифмическая линейка успешно использовалась еще 15 лет назад, более 360 лет прослужив инженерам.

2. Развитие механики в 17 веке стало предпосылкой создания вычислительных устройств и приборов, использующих механический способ вычислений. 1623г.- немецкий ученый В. Шиккард описывает и реализует в единственном экземпляре механическую счетную машину, предназначенную для выполнения 4-х арифметических операций над шестиразрядными числами. 1642г.-Б.Паскаль строит восьмиразрядную действующую модель счетной суммирующей машины. Было создано еще 50 таких машин. 1673г.-немецкий математик Г. Лейбниц создает первый арифмометр, позволяющий выполнять 4 арифметических операций. 1881г. – организация серийного производства арифмометров. Они использовались вплоть до 60-х годов 20 века. 1822г. 1 проект английского математика Ч. Беббиджа – разностная машина – 16-ти разрядный калькулятор, способный печатать цифровые таблицы. Эта машина имела арифметическое устройство, устройство управления, ввода и печати. работала на паровом двигателе и заносила результаты на металлическую пластины. 2 проект Беббиджа- аналитическая машина, использовавшая принцип программного управления и предназначавшаяся для вычисления любого алгоритма не был реализован, но получил широкую известность и высокую оценку ученых. Аналитическая машина состояла из 4-х основных частей: блок хранения исходных, промежуточных и

результатирующих данных (склад-память), блок обработки данных (мельница-арифметическое устройство), блок управления последовательностью вычислений (устройство управления), блок ввода исходных данных и печати результатов (устройство ввода-вывода). Вместе с Беббиджем работала Ада Лавлейс – первая программистка – писала первые программы для машины, заложила многие идеи и ввела ряд понятий и терминов, сохранившихся до настоящего времени.

3. Электромеханический этап развития вычислительной техники. 1887г. – создание Г. Холлеритом в США первого счетно-аналитического комплекса. Его использовали для обработки результатов переписи населения в нескольких странах, в том числе и в России. 30-е годы – разработка счетно-аналитических комплексов, которые состоят из 4-х основных устройств: перфоратор, контрольник, сортировщик и табулятор. В это же время развиваются аналоговые машины. 1930г. – В. Буш разрабатывает дифференциальный анализатор, использованный в дальнейшем в военных целях. 1937г. – Дж. Атанасов и К. Бери создают электронную машину АВС. 1944г. – Г. Айкен создает управляемую вычислительную машину МАРК-1. 1957г. – последний крупнейший проект релейной вычислительной техники – в СССР создана РВМ-1, которая эксплуатировалась до 1965г.

4. Электронный этап (создание ЭВМ) начинается с созданием в 1945г. в США электронной вычислительной машины ENIAC. В истории развития ЭВМ 5 поколений, которые отличаются в элементарной базе, логической архитектуре и программном обеспечении, различаются по быстродействию, оперативной памяти, способам ввода и вывода информации.

### 3 Поколения ЭВМ.

Начиная с 1950 года, каждые 7-10 лет кардинально обновлялись конструктивно-технологические и программно-алгоритмические принципы построения и использования ЭВМ. В связи с этим правомерно говорить о поколениях вычислительных машин. Условно каждому поколению можно отвести 10 лет.

ЭВМ проделали большой эволюционный путь в смысле элементной базы (от ламп к микропроцессорам) а также в смысле появления новых возможностей, расширения области применения и характера их использования.

Деление ЭВМ на поколения – весьма условная, нестрогая классификация вычислительных систем по степени развития аппаратных и программных средств, а также способов общения с ЭВМ.

К первому поколению ЭВМ относятся машины, созданные на рубеже 50-х годов: в схемах использовались электронные лампы. Команд было мало, управление – простым, а показатели объема оперативной памяти и быстродействия – низкими. Быстродействие порядка 10-20 тысяч операций в секунду. Для ввода и вывода использовались печатающие устройства, магнитные ленты, перфокарты и перфоленты.

Ко второму поколению ЭВМ относятся те машины, которые были сконструированы в 1955-65 гг. В них использовались как электронные лампы, так и транзисторы. Оперативная память была построена на магнитных сердечниках. В это время появились магнитные барабаны и первые магнитные диски. Появились так называемые языки высокого уровня, средства которых допускают описание всей последовательности вычислений в наглядном, легко воспринимаемом виде. Появился большой набор библиотечных программ для решения различных математических задач. Машинам второго поколения была свойственна программная несовместимость, которая затрудняла организацию крупных информационных систем, поэтому в середине 60х годов наметился переход к созданию ЭВМ, программно совместимых и построенных на микроэлектронной технологической базе.

Третье поколение ЭВМ. Это машины, создаваемые после 60х годов, обладающих единой архитектурой, т.е. программно совместимых. Появились возможности

мультипрограммирования, т.е. одновременного выполнения нескольких программ. В ЭВМ третьего поколения применялись интегральные схемы.

Четвертое поколение ЭВМ. Это нынешнее поколение ЭВМ, разработанных после 1970 г. Машины 4го поколения проектировались в расчёте на эффективное использование современных высокоуровневых языков и упрощение процесса программирования для конечного пользователя.

В аппаратурном отношении для них характерно использование больших интегральных схем как элементной базы и наличие быстродействующих запоминающих устройств с произвольной выборкой, объемом несколько Мбайт.

Машины 4-го поколения- многопроцессорные, многомашинные комплексы, работающие на внеш. память и общее поле внеш. устройств. Быстродействие достигает десятков миллионов операций в сек, память - нескольких млн. слов.

Переход к пятому поколению ЭВМ уже начался. Он заключается в качественном переходе от обработки данных к обработке знаний и в повышении основных параметров ЭВМ. Основной упор будет сделан на "интеллектуальность".

На сегодняшний день реальный «интеллект», демонстрируемый самыми сложными нейронными сетями, находится ниже уровня дождевого червя, однако, как бы ни были ограничены возможности нейронных сетей сегодня, множество революционных открытий, могут быть не за горами.

## **1. Первое поколение ЭВМ 1950-1960-е годы**

Логические схемы создавались на дискретных радиодеталях и электронных вакуумных лампах с нитью накала. В оперативных запоминающих устройствах использовались магнитные барабаны, акустические ультразвуковые ртутные и электромагнитные линии задержки, электронно-лучевые трубы (ЭЛТ). В качестве внешних запоминающих устройств применялись накопители на магнитных лентах, перфокартах, перфолентах и штекерные коммутаторы.

Программирование работы ЭВМ этого поколения выполнялось в двоичной системе счисления на машинном языке, то есть программы были жестко ориентированы на конкретную модель машины и "умирали" вместе с этими моделями.

В середине 1950-х годов появились машинно-ориентированные языки типа языков символьического кодирования (ЯСК), позволявшие вместо двоичной записи команд и адресов использовать их сокращенную словесную (буквенную) запись и десятичные числа. В 1956 году был создан первый язык программирования высокого уровня для математических задач - язык Фортран, а в 1958 году - универсальный язык программирования Алгол.

ЭВМ, начиная от UNIVAC и заканчивая БЭСМ-2 и первыми моделями ЭВМ "Минск" и "Урал", относятся к первому поколению вычислительных машин.

## **2. Второе поколение ЭВМ: 1960-1970-е годы**

Логические схемы строились на дискретных полупроводниковых и магнитных элементах (диоды, биполярные транзисторы, тороидальные ферритовые микротрансформаторы). В качестве конструктивно-технологической основы использовались схемы с печатным монтажом (платы из фольгированного гетинакса). Широко стал использоваться блочный принцип конструирования машин, который позволяет подключать к основным устройствам большое число разнообразных внешних устройств, что обеспечивает большую гибкость использования компьютеров. Тактовые частоты работы электронных схем повысились до сотен килогерц.

Стали применяться внешние накопители на жестких магнитных дисках1 и на флоппи-дисках - промежуточный уровень памяти между накопителями на магнитных лентах и оперативной памятью.

В 1964 году появился первый монитор для компьютеров - IBM 2250. Это был монохромный дисплей с экраном 12 x 12 дюймов и разрешением 1024 x 1024 пикселов. Он имел частоту кадровой развертки 40 Гц.

Создаваемые на базе компьютеров системы управления потребовали от ЭВМ более высокой производительности, а главное - надежности. В компьютерах стали широко использоваться коды с обнаружением и исправлением ошибок, встроенные схемы контроля.

В машинах второго поколения были впервые реализованы режимы пакетной обработки и телеобработки информации.

Первой ЭВМ, в которой частично использовались полупроводниковые приборы вместо электронных ламп, была машина SEAC (Standarts Eastern Automatic Computer), созданная в 1951 году.

В начале 60-х годов полупроводниковые машины стали производиться и в СССР.

### **3. Третье поколение ЭВМ: 1970-1980-е годы**

В 1958 году Роберт Нойс изобрел малую кремниевую интегральную схему, в которой на небольшой площади можно было размещать десятки транзисторов. Эти схемы позже стали называться схемами с малой степенью интеграции (Small Scale Integrated circuits -SSI). А уже в конце 60-х годов интегральные схемы стали применяться в компьютерах.

Логические схемы ЭВМ 3-го поколения уже полностью строились на малых интегральных схемах. Тактовые частоты работы электронных схем повысились до единиц мегагерц. Снизились напряжения питания (единицы вольт) и потребляемая машиной мощность. Существенно повысились надежность и быстродействие ЭВМ.

В оперативных запоминающих устройствах использовались миниатюрнее ферритовые сердечники, ферритовые пластины и магнитные пленки с прямоугольной петлей гистерезиса. В качестве внешних запоминающих устройств широко стали использоваться дисковые накопители.

Появились еще два уровня запоминающих устройств: сверхоперативные запоминающие устройства на триггерных регистрах, имеющие огромное быстродействие, но небольшую емкость (десятки чисел), и быстродействующая кэш-память.

Начиная с момента широкого использования интегральных схем в компьютерах, технологический прогресс в вычислительных машинах можно наблюдать, используя широко известный закон Мура. Один из основателей компании Intel Гордон Мур в 1965 году открыл закон, согласно которому количество транзисторов в одной микросхеме удваивается через каждые 1,5 года.

Ввиду существенного усложнения как аппаратной, так и логической структуры ЭВМ 3-го поколения часто стали называть системами.

Так, первыми ЭВМ этого поколения стали модели систем IBM (ряд моделей IBM 360) и PDP (PDP 1). В Советском Союзе в содружестве со странами Совета Экономической Взаимопомощи (Польша, Венгрия, Болгария, ГДР и др1.) стали выпускаться модели единой системы (ЕС) и системы малых (СМ) ЭВМ.

В вычислительных машинах третьего поколения значительное внимание уделяется уменьшению трудоемкости программирования, эффективности исполнения программ в машинах и улучшению общения оператора с машиной. Это обеспечивается мощными операционными системами, развитой системой автоматизации программирования, эффективными системами прерывания программ, режимами работы с разделением машинного времени, режимами работы в реальном времени, мультипрограммными режимами работы и новыми интерактивными режимами общения. Появилось и эффективное видеотерминальное устройство общения оператора с машиной - видеомонитор, или дисплей. Большое вниманиеделено повышению надежности и достоверности функционирования ЭВМ и облегчению их технического обслуживания. Достоверность и надежность обеспечиваются повсеместным использованием кодов с автоматическим обнаружением и исправлением ошибок (корректирующие коды Хемминга и циклические коды).

Модульная организация вычислительных машин и модульное построение их операционных систем создали широкие возможности для изменения конфигурации вычислительных систем. В связи с этим возникло новое понятие "архитектура" вычислительной системы,

определяющее логическую организацию этой системы с точки зрения пользователя и программиста.

#### **4. Четвертое поколение ЭВМ: 1980-1990-е годы**

Революционным событием в развитии компьютерных технологий третьего поколения машин было создание больших и сверхбольших интегральных схем (Large Scale Integration - LSI и Very Large Scale Integration - VLSI), микропроцессора (1969 г.) и персонального компьютера. Начиная с 1980 года практически все ЭВМ стали создаваться на основе микропроцессоров. Самым востребованным компьютером стал персональный.

Логические интегральные схемы в компьютерах стали создаваться на основе униполярных полевых CMOS-транзисторов с непосредственными связями, работающими с меньшими амплитудами электрических напряжений (единицы вольт), потребляющими меньше мощности, нежели биполярные, и тем самым позволяющими реализовать более прогрессивные нанотехнологии (в те годы - масштаба единиц микрон).

Оперативная память стала строиться не на ферритовых сердечниках, а также на интегральных CMOS-транзисторных схемах, причем непосредственно запоминающим элементом в них служила паразитная емкость между электродами (затвором и истоком) этих транзисторов.

Первый персональный компьютер создали в апреле 1976 года два друга, Стив Джобе (1955 г. р.) - сотрудник фирмы Atari, и Стефан Возняк (1950 г. р.), работавший на фирме Hewlett-Packard. На базе интегрального 8-битного контроллера жестко запаянной схемы популярной электронной игры, работая вечерами в автомобильном гараже, они сделали простенький программируемый на языке Бейсик игровой компьютер "Apple", имевший бешеный успех. В начале 1977 года была зарегистрирована Apple Сотр., и началось производство первого в мире персонального компьютера Apple.

#### **5. Пятое поколение ЭВМ: 1990-настоящее время**

Особенности архитектуры современного поколения компьютеров подробно рассматриваются в данном курсе.

Кратко основную концепцию ЭВМ пятого поколения можно сформулировать следующим образом:

1. Компьютеры на сверхсложных микропроцессорах с параллельно-векторной структурой, одновременно выполняющих десятки последовательных инструкций программы.
2. Компьютеры с многими сотнями параллельно работающих процессоров, позволяющих строить системы обработки данных и знаний, эффективные сетевые компьютерные системы.

Шестое и последующие поколения ЭВМ

Электронные и оптоэлектронные компьютеры с массовым параллелизмом, нейронной структурой, с распределенной сетью большого числа (десятка тысяч) микропроцессоров, моделирующих архитектуру нейронных биологических систем.

#### **Заключение**

Все этапы развития ЭВМ принято условно делить на поколения.

Первое поколение создавалось на основе вакуумных электроламп, машина управлялась с пульта и перфокарт с использованием машинных кодов. Эти ЭВМ размещались в нескольких больших металлических шкафах, занимавших целые залы.

Второе поколение появилось в 60-е годы 20 века. Элементы ЭВМ выполнялись на основе полупроводниковых транзисторов. Эти машины обрабатывали информацию под управлением программ на языке Ассемблер. Ввод данных и программ осуществлялся с перфокарт и перфолент.

Третье поколение выполнялось на микросхемах, содержавших на одной пластинке сотни или тысячи транзисторов. Пример машины третьего поколения - ЕС ЭВМ. Управление работой этих машин происходило с алфавитно-цифровых терминалов. Для управления использовались языки высокого уровня и Ассемблер. Данные и программы вводились как с терминала, так и с перфокарт и перфолент.

Четвертое поколение было создано на основе больших интегральных схем (БИС). Наиболее яркие представители четвертого поколения ЭВМ - персональные компьютеры (ПК). Персональной называется универсальная однопользовательская микроЭВМ. Связь с пользователем осуществлялась посредством цветного графического дисплея с использованием языков высокого уровня.

Пятое поколение создано на основе сверхбольших интегральных схем (СБИС), которые отличаются колоссальной плотностью размещения логических элементов на кристалле. Предполагается, что в будущем широко распространится ввод информации в ЭВМ с голоса, общения с машиной на естественном языке, машинное зрение, машинное осязание, создание интеллектуальных роботов и робототехнических устройств.

## 1. 2 Лекция №2 (2 часа)

**Тема:** Аппаратные средства реализации информационных процессов»

### 1.2.1 Вопросы лекции:

1. Понятие архитектуры компьютера.
2. Принципы построения ЭВМ.
3. Взаимодействие процессора и памяти при выполнении команд и программ.

### 1.2.2 Краткое содержание вопросов:

1. Понятие архитектуры компьютера

**Архитектура компьютера** — набор типов данных, операций и характеристик каждого отдельно взятого уровня называется архитектурой. Архитектура связана с программными аспектами. Аспекты реализации (например, технология, применяемая при реализации памяти) не являются частью архитектуры<sup>[1]</sup>.

Выделяют несколько уровней организации компьютера (компьютерной архитектуры), от двух и более:

Схема, иллюстрирующая многоуровневую структуру компьютера

#### Уровень 0

Цифровой логический уровень, это аппаратное обеспечение машины, состоящий из вентилей. См. также Логические элементы (зашелки), триггеры, регистры

#### Уровень 1

Микроархитектурный уровень, интерпретация (микропрограммы) или непосредственное выполнение. Электронные схемы исполняют машинно-зависимые программы. Совокупность регистров процессора формирует локальную память. Смотрим также арифметико-логическое устройство.

#### Уровень 2

Уровень архитектуры системы команд, трансляция (ассемблер).

#### Уровень 3

Уровень операционной системы, трансляция (ассемблер). Это гибридный уровень: одна часть команд интерпретируется операционной системой, а другая — микропрограммой. См. также виртуальная память, файлы.

#### Уровень 4

Уровень языка ассемблера, трансляция (компилятор). Четвертый уровень и выше используется для написания прикладных программ, с первого по третий — системных программ. Программы в удобном для человека виде транслируются на язык уровней 1-3.

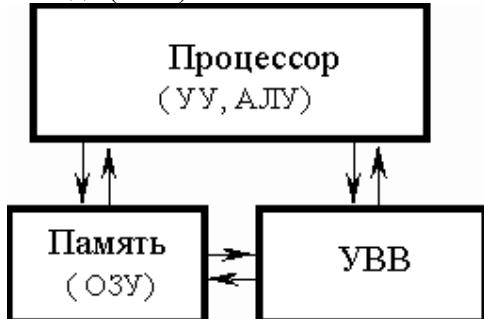
#### Уровень 5

Язык высокого уровня. Программы на языках высокого уровня транслируются обычно на уровнях 3 и 4.

## 2. Принципы построения ЭВМ.

**Основные принципы построения ЭВМ** были сформулированы американским учёным Джоном фон Нейманом в 40-х годах 20 века:

1. Любой ЭВМ образуют три основные компоненты: процессор, память и устройства ввода-вывода (УВВ).



2. Информация, с которой работает ЭВМ делится на два типа:

- набор команд по обработке (программы);
- данные подлежащие обработке.

3. И команды, и данные вводятся в память (ОЗУ) – **принцип хранимой программы**.

4. Руководит обработкой процессор, устройство управления (УУ) которого выбирает команды из ОЗУ и организует их выполнение, а арифметико-логическое устройство (АЛУ) проводит арифметические и логические операции над данными.

5. С процессором и ОЗУ связаны устройства ввода-вывода (УВВ).

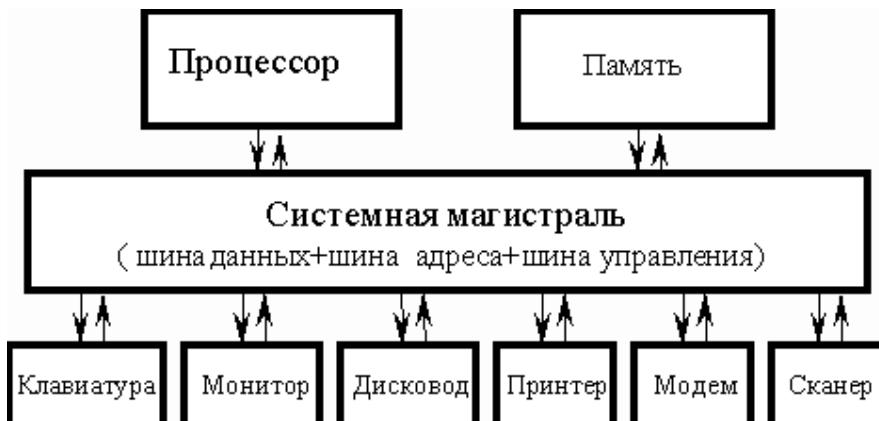
Архитектура современных персональных компьютеров основана на **магистрально-модульном принципе**. Информационная связь между устройствами компьютера осуществляется через **системную шину** (другое название - системная магистраль).

Шина - это кабель, состоящий из множества проводников. По одной группе проводников - **шине данных** передаётся обрабатываемая информация, по другой - **шине адреса** - адреса памяти или внешних устройств, к которым обращается процессор. Третья часть магистрали - **шина управления**, по ней передаются управляющие сигналы (например, сигнал готовности устройства к работе, сигнал к началу работы устройства и др.).

Системная шина характеризуется **тактовой частотой и разрядностью**. Количество одновременно передаваемых по шине бит называется **разрядностью шины**. **Тактовая частота** характеризует число элементарных операций по передаче данных в 1 секунду. Разрядность шины измеряется в битах, тактовая частота – в мегагерцах.

Всякая информация, передаваемая от процессора к другим устройствам по шине данных, сопровождается **адресом**, передаваемым по адреснойшине. Это может быть адрес ячейки памяти или адрес периферийного устройства. Необходимо, чтобы разрядность шины позволила передать адрес ячейки памяти. Таким образом, словами разрядность шины ограничивает объем оперативной памяти ЭВМ, он не может быть больше чем  $2^n$ , где  $n$  – разрядность шины. Важно, чтобы производительности всех подсоединеных к шине устройств были согласованы. Неразумно иметь быстрый процессор и медленную память или быстрый процессор и память, но медленный винчестер.

Ниже представлена схема устройства компьютера, построенного по магистральному принципу:



В современных ЭВМ реализован **принцип открытой архитектуры**, позволяющий пользователю самому комплектовать нужную ему конфигурацию компьютера и производить при необходимости её модернизацию. **Конфигурацией** компьютера называют фактический набор компонентов ЭВМ, которые составляют компьютер. Принцип открытой архитектуры позволяет менять состав устройств ЭВМ. К информационной магистрали могут подключаться дополнительные периферийные устройства, одни модели устройств могут заменяться на другие.

Аппаратное подключение периферийного устройства к магистрали на физическом уровне осуществляется через специальный блок - контроллер (другие названия - адаптер, плата, карта). Для установки контроллеров на материнской плате имеются специальные разъёмы - **слоты**.

Программное управление работой периферийного устройства производится через программу - **драйвер**, которая является компонентой операционной системы. Так как существует огромное количество разнообразных устройств, которые могут быть установлены в компьютер, то обычно к каждому устройству поставляется драйвер, взаимодействующий непосредственно с этим устройством.

Связь компьютера с внешними устройствами осуществляется через **порты** – специальные разъёмы на задней панели компьютера.

Различают **последовательные** и **параллельные** порты. Последовательные (СОМ – порты) служат для подключения манипуляторов, модема и передают небольшие объёмы информации на большие расстояния. Параллельные (LPT - порты) служат для подключения принтеров, сканеров и передают большие объёмы информации на небольшие расстояния. В последнее время широкое распространение получили последовательные универсальные порты (USB), к которым можно подключать различные устройства.

Минимальная конфигурация компьютера включает в себя: системный блок, монитор, клавиатуру и мышь.

### 3 Взаимодействие процессора и памяти при выполнении команд и программ.

Теперь от рассмотрения абстрактных проблем перейдем к рассмотрению конструктивных особенностей компьютера, не вдаваясь при этом глубоко в технические детали. Сейчас нас интересует принципиальная сторона вопроса. В компьютере хранение данных и их обработка пространственно разделены. Ранее было сказано, что устройство, предназначенное для хранения данных, называется памятью компьютера. Устройство, производящее различные вычисления и управляющее работой компьютера, называется центральным процессором (ЦП). Полное состояние компьютера определяется той информацией, которая хранится в памяти компьютера. Удобно делить память на три основных раздела: адресуемая память (которая и называется оперативной памятью), регистры процессора и ячейки ввода-

вывода (последний раздел является условным, так как он просто отображает процессы обмена информации с внешней средой).

Оперативная память (или оперативное запоминающее устройство - ОЗУ) в качестве составной части содержит постоянную память (или постоянное запоминающее устройство - ПЗУ). Постоянная память обладает той особенностью, что в нее нельзя записывать информацию. Нули и единицы в устройствах оперативной памяти изображаются электрическими сигналами, и поэтому информация в оперативной памяти бесследно исчезает при выключении питания. Однако постоянная память основана на других принципах хранения информации: в ней нули и единицы кодируются электрическими соединениями. При включении питания эту информацию можно прочитать.

Те условные ячейки, через которые информация может перемещаться из внешней среды в оперативную память и обратно, называются портами ввода-вывода. Для описания работы компьютера на принципиальном уровне можно считать, что в порты ввода из внешней среды (независимо или по запросу процессора) помещаются некоторые данные. Наоборот, при необходимости переслать данные куда-либо или сохранить их во внешней памяти нужно просто поместить эти данные в определенный порт вывода. Обо всем остальном позаботятся вспомогательные устройства компьютера. О некоторых из них пойдет речь ниже.

В некоторые порты информация передается порциями по одному биту. Такой порт называется последовательным. Порт, в который информация передается порциями по одному или более байт, называется параллельным. Например, модем подключается к последовательному порту, а принтер – к параллельному.

Процессор – это центральное устройство компьютера. На него возложены две основные функции: во-первых, производить определенные вычисления, и, во-вторых, управлять работой всех узлов компьютера. Эти функции выполняют различные составляющие процессора: арифметико-логическое устройство (АЛУ) и устройство управления. Конструктивно процессор состоит из огромного количества электронных микросхем, сосредоточенных в микроскопическом объеме. Быть может, процессор является самым сложным устройством в мире. Весь прогресс в компьютерной индустрии связан с совершенствованием процессоров: расширением списка выполняемых ими функций, уменьшением объема и одновременным увеличением скорости выполнения операций (быстродействием), увеличением надежности. Именно для увеличения быстродействия данные перед непосредственным вычислением перемещаются из оперативной памяти в специальные ячейки, называемые регистрами процессора. Этих регистров несколько десятков и они выполняют различные функции.

Быстродействие компьютера измеряется количеством выполняемых элементарных операций в секунду - флопов. Например, один гигафлоп означает миллиард операций в секунду. Единицей же действия компьютера является машинная команда. Число различных видов машинных команд, выполняемых процессором, конечно. Для каждого вида машинной команды в процессоре есть отвечающая за нее электронная микросхема. Различные машинные команды в зависимости от сложности состоят от нескольких до нескольких десятков элементарных операций. Все машинные команды делятся на группы. Команда каждой группы выполняет одно из следующих действий:

- “ перемещает блок информации из оперативной памяти в регистры процессора;
- “ перемещает блок информации из регистров процессора в оперативную память компьютера;
- “ получает блок информации извне через один из портов и помещает ее в регистр процессора;
- “ выводит блок информации из регистра процессора через один из портов во внешнюю среду;
- “ инициирует операцию вычисления, выполняемую процессором: операция совершается над величинами, хранящимися в регистрах процессора, и результат помещается в другие или те же регистры;

“ оповещает процессор о некоторой нештатной ситуации, называемой прерыванием: по этой команде процессор прекращает выполнение текущей программы и запускает другую программу, которая должна работать в случае прерывания данного типа.

Все содержательные операции совершаются непосредственно процессором над величинами, которые находятся в регистрах процессора. В частности, это арифметические и логические операции, а также операции преобразования. Эти операции фактически состоят из большого числа более мелких элементарных операций, каждая из которых производится над содержимым одного из разрядов одного из регистров. Многие из этих операций совершаются процессором параллельно (то есть по времени одновременно). Для этого в компьютерах используется генератор тактовых импульсов, который синхронизирует работу различных частей компьютера. Каждая машинная команда состоит из определенного числа тактов. За время одного такта процессор выполняет несколько элементарных операций, которые могут выполняться независимо. Если одна операция требует результатов другой, то они выполняются в последовательных тактах. После выполнения всех тактов команды в одном из регистров процессора образуется результат выполнения всей машинной команды. Уже было сказано, что разные машинные команды требуют при выполнении различного количества тактов. В частности, число тактов при умножении чисел значительно больше, чем при сложении.

Перемещение информации между оперативной памятью и процессором и между оперативной памятью и портами происходит по системе соединений, которая называются шиной данных. Для увеличения скорости передачи биты информации передаются одновременно по нескольким линиям шины. Количество линий называется разрядностью шины. В персональных компьютерах используются 32-разрядные и 64-разрядные шины данных. По первой одновременно идет 4 байта информации, по второй – 8 байтов. Однако для правильной организации работы компьютера процессор и память должны обмениваться не только данными, но и управляющими сигналами. Для этого в компьютере предусмотрены кроме шины данных еще две шины: шина адреса и шина управления (на самом деле есть еще шины питания, по которым на все устройства компьютера подается питание).

Шина адреса нужна для того, чтобы локализовать те ячейки оперативной памяти или те порты ввода-вывода, которые должны непосредственно участвовать в операции. Все байты оперативной памяти перенумерованы числами от нуля до максимального номера байта (последний зависит от объема оперативной памяти). Аналогично, перенумерованы также все порты ввода-вывода (обычно от 0 до 65535). Адресом байта в оперативной памяти считается его номер. Адресом участка памяти, состоящего из нескольких байтов (области памяти) служит адрес начального байта. Адресом порта ввода-вывода также служит его номер. При посылке предписания процессора к выполнению некоторой операции адрес того байта, который должен участвовать в операции, посыпается процессором по шине адреса. При прохождении адреса по шине адреса активизируется именно тот байт памяти, номер которого совпадает с посланным адресом. Остальные управляющие сигналы, необходимые для правильного выполнения операции, посыпаются по шине управления.

Для характеристики компьютера очень важна разрядность шины адреса. Например, у прежних персональных компьютеров использовалась 20-разрядная шина адреса. Максимальный адрес, который можно послать по такой шине, равен  $2^{20}-1 = 1\text{Мб}$ , то есть байту оперативной памяти с адресом большим 1Мб предписание по шине адреса отправить невозможно. В таких компьютерах объем оперативной памяти принципиально не мог быть больше 1Мб. В процессорах этих компьютеров использовалась специальная система определения адреса, ориентированная на такое ограничение. В результате все программы, написанные в то время, предусматривали адреса до 1Мб.

Современные персональные компьютеры включают 32-разрядную шину адреса. При такойшине максимальный объем оперативной памяти равен  $2^{32}=4\text{Гб}$ . При 32-разряднойшине можно обратиться к любому байту оперативной памяти в пределах 4Гб. Новые программы

так и делают. Однако, к сожалению, необходимо предусмотреть возможность выполнения программ, написанных для старых процессоров. Поэтому в современных процессорах предусмотрены два режима работы: один режим, называемый реальным, имитирует работу старых процессоров, и в этом режиме осуществляется доступ только к 1Мг оперативной памяти; другой режим, называемый защищенным, имеет доступ ко всей оперативной памяти. В настоящее время происходит переход от 32-разрядной к 64-разрядной шине адреса. Любопытно, что 64 разрядной шины должно хватить уже навсегда.

По шине управления идут сигналы, которые выполняют различные вспомогательные функции, необходимые для правильного выполнения операций. Всего линий в шине управления может быть более ста. Перечислим некоторые основные линии управления. Например, существует линия переключения между оперативной памятью и портами ввода-вывода. Дело в том, что когда по шине адреса идет сигнал, то он может восприниматься и как номер байта оперативной памяти, и как номер порта ввода-вывода. Как именно воспринимать этот адрес, зависит от сигнала, который параллельно с адресом идет по управляющей линии (например, нуль на управляющей линии обозначает оперативную память, единица – порт). По другой управляющей линии идет сигнал, который задает направление перемещения информации (нуль – информация читается из памяти или из порта в регистр процессора, единица – пишется из регистра в память или порт). По третьей управляющей линии передаются сигналы от тактового генератора. Эти сигналы позволяют синхронизировать операции, которые должны одновременно выполняться сразу несколькими устройствами компьютера (например, подготовиться к очередной операции). По специальным линиям передаются сигналы прерывания от различных устройств.

Все так прочие (так называемые периферийные) устройства компьютера подключаются к одному из портов ввода-вывода. Очень часто одна и та же шина данных используется для обмена данными между процессором и всеми внешними входами и выходами. Такая шина называется общей шиной. Это означает, что в процессе работы компьютера процессор посылает сигналы, которые идут по общей шине и которые в принципе могут прочесть все подключенные к шине устройства. Однако с помощью управляющих сигналов и шины адреса реально их воспримет только то устройство, для которого эта информация предназначена.. Следует отметить, что эта простая однозначная схема часто бывает усложнена. Реально устройства подключаются к общей шине не непосредственно. Некоторые порты могут быть подсоединенны к одной из вспомогательных шин, которая в свою очередь прикрепляется к общей шине. Однако это не меняет принципиальной схемы работы компьютера.

### **1. 3 Лекция №3 (2 часа)**

**Тема: «Состав и назначение основных элементов компьютера»**

#### **1.3.1 Вопросы лекции:**

1. Внутреннее и внешнее устройство компьютера.
2. Материнская плата.
3. Процессор.
4. ОЗУ.
5. ПЗУ.
6. Типы разъемов.

#### **1.3.2 Краткое содержание вопросов:**

#### **УСТРОЙСТВО КОМПЬЮТЕРА.**

Компьютер – от анг. Computer – вычислитель, представляет собой программируемое

устройство, способное обрабатывать данные и производить вычисления  
Базовая компьютерная конфигурация.

- системный блок (представляет собой основной узел, внутри которого установлены наиболее важные компоненты);
- монитор –
- клавиатура и мышь – для ввода информации в компьютер;

Устройства компьютера, не принадлежащие к основным, называются **периферийными**.

**Внутренне устройство компьютера.**

**Материнская плата** это самая главная схема компьютера, на ней располагаются:

**1. Процессор** – Самым главным элементом в компьютере, его мозг – небольшая в несколько сантиметров электронная схема выполняющая все вычисления и обработку информации. Процессор умеет производить сотни различных операций и делает это со скоростью сотен миллионов операций в секунду. сущ. процессоры различных фирм например Intel, AMD, Сурих. Основными характеристиками процессора **Разрядность процессора** Количество одновременно обрабатываемых данных. **Тактовая частота** Количество операций, выполняемых за единицу времени

**2. Оперативная память** (ОЗУ) набор микросхем, предназначенный для временного хранения данных, когда компьютер включен. Именно из нее процессор берет программы и исходные данные для обработки, в нее они записывают полученные результаты. Название это она получила потому что работает очень быстро и процессору не приходится ждать при чтении данных из памяти или записи в память. (64, 32 Мб)

**3. Шины** – набор проводников, по которым происходит обмен сигналами между внутренними устройствами компьютера.

**4. ПЗУ** – микросхема предназначеннная для длительного хранения данных, в том числе когда компьютер выключен.(в момент включения компьютера в оперативной памяти комп ничего нет, но процессору необходимо команды, программы находящиеся в ПЗУ, называют **зашитыми**- их записывают туда на этапе изготовления микросхемы, этот пакет программ образует базовую систему ввода-вывода данных BIOS, их основное назначение, проверить состав и работоспособность компьютера и обеспечить взаимодействие между основными устройствами компьютера, работу этих программ мы видим на экране в виде диагностического сообщения сопровождающего запуск комп.)

**5. Разъемы** для подключения дополнительных устройств (слоты)

К системному блоку можно подключить различные внешние и внутренние устройства:

Внешние устройства обеспечивают взаимодействие компьютера с окружающей средой – пользователями, объектами управления и другими компьютерами. Подключаются к компьютеру через специальные разъемы-порты ввода-вывода. Порты ввода-вывода бывают различных типов: параллельные (обозначаемые LPT1 – LPT4) – обычно используются для подключения принтеров; последовательные (обозначаемые COM1 – COM4) – обычно к ним подключаются мышь, модем и другие устройства. USB порты. (является более высокоскоростным и в последнее время вытесняет COM и LPT)

**К внешним устройствам относятся:**

1. устройства ввода информации;
2. устройства вывода информации;
3. диалоговые средства пользователя;
4. средства связи и телекоммуникации.

**К устройствам ввода информации относятся:**

1. клавиатура — устройство для ручного ввода в компьютер числовой, текстовой и управляющей информации; (все клавиши клавиатуры разделены на пять групп: алфавито-цифровых, функциональных, управления курсором, дополнительных клавиш, служебные клавиши)

2. графические планшеты (дигитайзеры) — для ручного ввода графической информации, изображений путем перемещения по планшету специального указателя (пера); при перемещении пера автоматически выполняется считывание координат его местоположения и ввод этих координат в компьютер;
3. сканеры (читающие автоматы) — для автоматического считывания с бумажных носителей и ввода в компьютер машинописных текстов, графиков, рисунков, чертежей; цифровые фотокамеры
4. устройства указания (манипуляторы) — для ввода графической информации на экран монитора путем управления движением курсора по экрану с последующим кодированием координат курсора и вводом их в компьютер (джойстик, мышь инфракрасные (беспроводные) мыши); **трекболы** (в отличие от стационарной мыши, движение шарика производится ладонью руки, **точпады** - сенсорные пластины, реагирующие на движение пальца по поверхности, удар по поверхности, воспринимается как нажатие кнопки, недостаток невысокая точность,
5. сенсорные экраны — для ввода отдельных элементов изображения, программ или команд с полиэкрана дисплея в компьютер).

**К устройствам вывода информации относятся:**

1. графопостроители (плоттеры) — устройство, которое чертит графики, рисунки или диаграммы под управлением компьютера. Плоттеры используются для получения сложных конструкторских чертежей, архитектурных планов, географических и метеорологических карт, деловых схем. Плоттеры рисуют изображения с помощью пера.
2. принтеры — печатающие устройства для вывода информации на бумажный носитель.

**Виды принтеров:**

- Матричные – принцип печати: печатающая головка принтера содержит вертикальный ряд тонких металлических стержней иголок, головка движется вдоль печатаемой строки, а стержни в нужный момент ударяют по бумаге через красящую ленту. Качество печати зависит от количества иголок (9, 18 и 24 игольчатые принтеры) Недостатками этих недорогих принтеров являются их шумная работа и невысокое качество печати,
- Струйные – в них изображение формируется микрокаплями специальных чернил выдавливаемых на бумагу с помощью сопел. Скорость печати до 500 знаков в секунду принтеры требовательны к качеству бумаги. Цветные струйные принтеры создают цвета, комбинируя чернила четырех основных цветов — ярко-голубого, пурпурного, желтого и черного.
- Лазерные принтеры в них используется принцип ксерографии: Компьютер формирует в своей памяти "образ" страницы текста и передает его принтеру, информация о странице проецируется с помощью лазерного луча на вращающийся барабан и затем переносится на бумагу. Широко используются цветные лазерные принтеры
- Светодиодные принтеры – принцип действия как у лазерного принтера, только вместо лазера используется линейка светодиодов

**К диалоговым средствам пользователя относятся:**

- Мониторы- устройство визуального представления данных, одно из главных устройств вывода информации. Сейчас наиболее распространены мониторы двух типов: на основе электронно-лучевой трубы (ЭЛТ) и плоские жидкокристаллические. ЭЛТ-мониторы обеспечивают лучшее качество изображения, но в пользу жидкокристаллических мониторов говорит их компактность, небольшой вес и идеально плоская поверхность. Размер монитора измеряется между противоположными углами видимой части экрана по диагонали, единица измерения дюйм. Стандартные размеры: 14, 15, 17, 19, 20, 21. Сейчас наиболее универсальными являются мониторы 15-17 дюймов, а для операций с графикой желательно мониторы ЭЛТ 19-21 дюйм. Так же для монитора есть еще одна важна характеристика – частота регенерации (обновления, измеряют в герцах) изображения показывает, сколько раз в течении секунды монитор может полностью сменить изображение. Чем выше частота, тем

устойчивее изображение, тем меньше утомляемость глаз. При частоте 60 Гц, мелкое мерцание заметно невооруженным глазом. Для ЭЛТ-мониторов нормальное значение 85 Гц, комфортное 100 Гц и более. Для жидкокристаллических мониторов изображение более инерционно, так что мерцание подавляется автоматически, для них комфортной считается частота в 75 Гц

— устройства речевого ввода-вывода информации. К ним относятся различные микрофонные акустические системы, а также различные синтезаторы звука, выполняющие преобразование цифровых кодов в буквы и слова, воспроизводимые через динамики или звуковые колонки, подсоединенные к компьютеру.

### **Средства связи и телекоммуникации**

используются для подключения компьютера к каналам связи, другим компьютерам и компьютерным сетям. К этой группе прежде всего относятся сетевые адаптеры. В качестве сетевого адаптера чаще всего используются модемы (модулятор-демодулятор). модем — для обмена информацией между компьютерами через телефонную сеть; факс-модем — сочетает возможности модема и телефакса;

Многие из названных выше устройств относятся к условно выделенной группе — средствам мультимедиа.

**Средства мультимедиа** — это комплекс аппаратных и программных средств, позволяющих человеку общаться с компьютером, используя самые разные естественные для себя среды: звук, видео, графику, тексты, анимацию и др. К средствам мультимедиа относятся:

1. устройства речевого ввода и вывода информации;
2. микрофоны и видеокамеры, акустические и видеовоспроизводящие системы с усилителями, звуковыми колонками, большими видеоэкранами;
3. звуковые и видеоплаты, платы видеозахвата, снимающие изображение с видеомагнитофона или видеокамеры и вводящие его в компьютер;
4. сканеры;
5. внешние запоминающие устройства большой емкости на оптических дисках, часто используемые для записи звуковой и видеинформации.

### **НАКОПИТЕЛИ НА ДИСКАХ.**

**Жесткий диск** (винчестер) предназначены для постоянного хранения информации. На самом деле это не один диск, а группа соосных дисков, имеющих магнитное покрытие и вращающихся с большой скоростью, над каждым находится считающая головка. Основной параметр винчестера его емкость, т.е. сколько информации помещается на диске (1, 10, -80 Гбайт). Несмотря на свое название, жесткий диск является очень хрупким, чувствительным к перегрузкам, ударам и толчкам.

**Гибкий диск** (дискета) — устройство для хранения небольших объемов информации, представляющее собой гибкий пластиковый диск в защитной оболочке, используемый для переноса данных с одного компьютера на другой. В пластиковом корпусе есть отверстие, которое прикрыто сдвигающейся шторкой для защиты от влаги, грязи и пыли, через которое считывается информация. В нижней части дискеты находятся два окошечка, если они оба открыты то дискета защищена (на нее ничего нельзя записать и удалить), для разрешения записи окошко открывают. Для закрытия окошечка используется специальная задвижка. Сейчас используются дискеты размером 3,5 дюйма емкостью 1,44 Мбайт, (раньше 5,25 дюймов).

Гибкие диски считаются малонадежными носителями информации. Пыль, грязь, влага, температурные перепады и электромагнитные поля часто становятся причиной порчи диска и утери данных.

При использовании диска следует придерживаться следующих правил:

1. Беречь от пыли, грязи, влаги, темп перепадов и электромагнитных полей
2. записывать данные в двойном экземпляре
3. не работать с данными на диске, нужно скопировать их на жесткий диск.

Перед первым использованием дискету необходимо специальным образом инициализировать (т.е. отформатировать) дискету.

*Форматирование диска в Windows.*

1. Открыть папку Мой компьютер
2. Вызвать контекстное меню нажатием пр. кнопки мыши по значку Диск 3,5 (A)
3. В появившемся меню выбрать Форматирование
4. Появится окно форматирования диска в нем поставить галочку напротив Полное и нажать кнопку начать.
5. После того как появится сообщение форматирование успешно завершено наж Ок

**Накопители на компакт дисках** – при малых физических размерах CD обладают высокой информационной емкостью, примерно 650 Мбайт (примерно 500 маленьких дискет), считывание информации происходит с высокой скоростью. В целях сохранения информации CD-ROM диски необходимо оберегать от загрязнения.

Сейчас очень популярными стали *цифровые видеодиски DVD*. Эти диски имеют тот же размер, что и обычные CD, но вмещают 4,7 Гбайт данных, т.е. по объёму заменяют семь стандартных дисков CD-ROM.

Помимо них есть еще *стримеры* (кассеты с магнитной лентой ёмкостью 1 - 2 Гбайта и больше, Недостатком стримеров является их сравнительно низкая скорость записи, поиска и считывания информации), *Zip-накопители, флэш-диски* (очень популярные, не требуют установки драйверов и при подсоединении распознаются как съемный диск).

#### **1. 4 Лекция №4 (2 часа)**

**Тема:** «Классификации периферийных устройств, устройство и их назначение»

##### **1.4.1 Вопросы лекции:**

1. Устройства ввода-вывода информации.
2. Диалоговые средства.
3. Средства мультимедиа.
4. Средства связи и телекоммуникации

##### **1.4.2 Краткое содержание вопросов:**

1. Устройства ввода-вывода информации.

Периферийные устройства персонального компьютера подключаются к его интерфейсам и предназначены для выполнения вспомогательных операций. Благодаря им компьютерная система приобретает гибкость и универсальность.

По назначению периферийные устройства можно подразделить на:

- устройства ввода данных;
- устройства вывода данных;
- устройства хранения данных;
- устройства обмена данными.

При этом устройства ввода данных подразделяются на устройства ввода знаковых данных, устройства ввода графических данных, устройства командного управления.

Устройства вывода данных в понимании некоторых авторов [1] представляют собою принтеры. Хотя к устройствам вывода можно, несомненно, отнести и монитор, и различные устройства, предназначенные для вывода звуковой информации (колонки, например).

Далее выделяют устройства хранения данных (к ним относятся дисковые накопители, CD-ROM и иные устройства, обеспечивающие хранение информации).

К устройствам обмена информацией относят прежде всего модемы, с помощью которых такой обмен происходит.

Некоторые устройства, в зависимости от ситуации применения, могут являться входными и выходными [4]. Так, с помощью дисковода или модема в компьютер может быть введена информация, а может быть и выведена с целью передачи на другой компьютер. Не могут быть выходными устройства ввода информации (сканер, клавиатура, мышь), а так же устройства вывода (принтер, монитор) не могут являться входными устройствами.

## 2. Диалоговые средства.

### **К устройствам ввода информации относятся:**

1. клавиатура — устройство для ручного ввода в компьютер числовой, текстовой и управляющей информации; (все клавиши клавиатуры разделены на пять групп: алфавито-цифровых, функциональных, управления курсором, дополнительных клавиш, служебные клавиши)
2. графические планшеты (дигитайзеры) — для ручного ввода графической информации, изображений путем перемещения по планшету специального указателя (пера); при перемещении пера автоматически выполняется считывание координат его местоположения и ввод этих координат в компьютер;
3. сканеры (читающие автоматы) — для автоматического считывания с бумажных носителей и ввода в компьютер машинописных текстов, графиков, рисунков, чертежей; цифровые фотокамеры
4. устройства указания (манипуляторы) — для ввода графической информации на экран монитора путем управления движением курсора по экрану с последующим кодированием координат курсора и вводом их в компьютер (джойстик, мышь инфракрасные (беспроводные) мыши)); **трекболы** (в отличие от стационарной мыши, движение шарика производится ладонью руки, **точпады** - сенсорные пластины, реагирующие на движение пальца по поверхности, удар по поверхности, воспринимается как нажатие кнопки, недостаток невысокая точность,
5. сенсорные экраны — для ввода отдельных элементов изображения, программ или команд с полизэкрана дисплея в компьютер).

### **К устройствам вывода информации относятся:**

1. графопостроители (плоттеры) — устройство, которое чертит графики, рисунки или диаграммы под управлением компьютера. Плоттеры используются для получения сложных конструкторских чертежей, архитектурных планов, географических и метеорологических карт, деловых схем. Плоттеры рисуют изображения с помощью пера.
2. принтеры — печатающие устройства для вывода информации на бумажный носитель.

### **Виды принтеров:**

- Матричные – принцип печати: печатающая головка принтера содержит вертикальный ряд тонких металлических стержней иголок, головка движется вдоль печатаемой строки, а стержни в нужный момент ударяют по бумаге через красящую ленту. Качество печати зависит от количества иголок (9, 18 и 24 игольчатые принтеры) Недостатками этих недорогих принтеров являются их шумная работа и невысокое качество печати,
- Струйные – в них изображение формируется микрокаплями специальных чернил выдавливаемых на бумагу с помощью сопел. Скорость печати до 500 знаков в секунду принтеры требовательны к качеству бумаги. Цветные струйные принтеры создают цвета, комбинируя чернила четырех основных цветов — ярко-голубого, пурпурного, желтого и черного.
- Лазерные принтеры в них используется принцип ксерографии: Компьютер формирует в своей памяти "образ" страницы текста и передает его принтеру, информация о странице проецируется с помощью лазерного луча на вращающийся барабан и затем переносится на бумагу. Широко используются цветные лазерные принтеры
- Светодиодные принтеры – принцип действия как у лазерного принтера, только вместо лазера используется линейка светодиодов

### **К диалоговым средствам пользователя относятся:**

- Мониторы- устройство визуального представления данных, одно из главных устройств вывода информации. Сейчас наиболее распространены мониторы двух типов: на основе электронно-лучевой трубы (ЭЛТ) и плоские жидкокристаллические. ЭЛТ-мониторы обеспечивают лучшее качество изображения, но в пользу жидкокристаллических мониторов говорит их компактность, небольшой вес и идеально плоская поверхность. Размер монитора измеряется между противоположными углами видимой части экрана по диагонали, единица измерения дюйм. Стандартные размеры: 14, 15, 17, 19, 20, 21. Сейчас наиболее универсальными являются мониторы 15-17 дюймов, а для операций с графикой желательно мониторы ЭЛТ 19-21 дюйм. Так же для монитора есть еще одна важна характеристика – частота регенерации (обновления, измеряют в герцах) изображения показывает, сколько раз в течении секунды монитор может полностью сменить изображение. Чем выше частота, тем устойчивее изображение, тем меньше утомляемость глаз. При частоте 60 Гц, мелкое мерцание заметно невооруженным глазом. Для ЭЛТ-мониторов нормальное значение 85 Гц, комфортное 100 Гц и более. Для жидкокристаллических мониторов изображение более инерционно, так что мерцание подавляется автоматически, для них комфортной считается частота в 75 Гц
- устройства речевого ввода-вывода информации. К ним относятся различные микрофонные акустические системы, а также различные синтезаторы звука, выполняющие преобразование цифровых кодов в буквы и слова, воспроизводимые через динамики или звуковые колонки, подсоединенные к компьютеру.

### **Средства связи и телекоммуникации**

используются для подключения компьютера к каналам связи, другим компьютерам и компьютерным сетям. К этой группе прежде всего относятся сетевые адаптеры. В качестве сетевого адаптера чаще всего используются модемы (модулятор-демодулятор). модем – для обмена информацией между компьютерами через телефонную сеть; факс-модем – сочетает возможности модема и телефакса;

Многие из названных выше устройств относятся к условно выделенной группе — средствам мультимедиа.

**Средства мультимедиа** — это комплекс аппаратных и программных средств, позволяющих человеку общаться с компьютером, используя самые разные естественные для себя среды: звук, видео, графику, тексты, анимацию и др. К средствам мультимедиа относятся:

1. устройства речевого ввода и вывода информации;
2. микрофоны и видеокамеры, акустические и видеовоспроизводящие системы с усилителями, звуковыми колонками, большими видеоэкранами;
3. звуковые и видеоплаты, платы видеозахвата, снимающие изображение с видеомагнитофона или видеокамеры и вводящие его в компьютер;
4. сканеры;
5. внешние запоминающие устройства большой емкости на оптических дисках, часто используемые для записи звуковой и видеинформации.

## **1. 5 Лекция №5 (2 часа)**

**Тема:** «Программное обеспечение для компьютера»

### **1.5.1 Вопросы лекции:**

1. Программное обеспечение.
2. Классификация программного обеспечения (прикладные, инструментальные и системные).

3. Краткий разбор основного программного обеспечения: операционная система Windows. Программы оболочки: Проводник и Far. Текстовые редакторы и процессоры (Word Pad, Microsoft Word). Табличные процессоры.
4. Файловая структура компьютера. Понятие файла и каталога.

### **1.5.2 Краткое содержание вопросов:**

#### **1. Программное обеспечение**

Последовательность действий, записанная на специальном языке и предназначенная для выполнения компьютером, - это **программа**; Конечная цель любой компьютерной программы – управление аппаратными средствами, даже если на первый взгляд программа никак не взаимодействует с оборудованием.

**Программное обеспечение** — это совокупность программ, позволяющих осуществить на компьютере автоматизированную обработку информации.

#### **2. Классификация программного обеспечения (прикладные, инструментальные и системные).**

**Программы, работающие на компьютере можно разделить на три категории:**

1. **прикладные программы**, обеспечивают выполнение необходимых пользователем работ: редактирование текстов, картинок. (Word, Excel).

Различают следующие типы пакетов прикладных программ:

- общего назначения — ориентированы на автоматизацию широкого круга задач пользователя (текстовые процессоры, табличные редакторы, системы управления базами данных, графические процессоры, издательские системы, системы автоматизации проектирования и т. д.);-методо-ориентированные — реализация разнообразных экономико-математических методов решения задач (математического программирования, сетевого планирования и управления, теории массового обслуживания, мат. статистики и т. д.);  
-проблемно-ориентированные — направлены на решение определенной задачи (проблемы) в конкретной предметной области (банковские пакеты, пакеты бух. учета, финансового менеджмента, правовых справочных систем и т. д.).

2. **Системные программы**, обеспечивают взаимодействие различных программ компьютерной системы и непосредственно с аппаратным обеспечением, т.е. выполняют посреднические (вспомогательные) функции, например, создание копий используемой информации, выдача справочной информации о компьютере, проверка работоспособности устройств компьютера, и др.;

– Среди всех системных программ особое место занимает **операционная система** – программа которая загружается при включении компьютера. Она осуществляет диалог с пользователем, управление компьютером, его ресурсами (оперативной памятью, местом на дисках и т.д.), запускает другие прикладные программы. Самая распространенная ОС MS DOS, LINUX, UNIX. Таким образом опер. система является программным расширением устройства управления компьютера.

– Популярный класс системных программ составляют **программы-оболочки**, они обеспечивают более удобный и наглядный способ общения с компьютером, чем с помощью командной строки DOS. NC, VC, WinCom.

– Операционные оболочки – в отличии от обычных программ оболочек не только дают более наглядные средства для часто используемых действий, но и представляют новые возможности для запускаемых программ. Windows

– Важным классом системных программ явл-ся программы драйверы. Без помощи драйверов невозможно подключение к компьютеру новых устройств, драйверы обеспечивают взаимосвязь оборудования с другими программами

– Так же к системным программам можно отнести утилиты (программы которые

выполняют работу по проверке, наладке и настройке компьютерной системы), программы упаковщики - программы для сжатия информации.

– Антивирусные программы – предназначены для предотвращения заражения компьютерным вирусом и ликвидации последствий заражения вирусом. И др.

3. **Инструментальные программы** (системы программирования) обеспечивает создание новых программ для компьютера. Инструментальные программные средства включают в свой состав средства написания программ (текстовые редакторы); средства преобразования программ в вид, пригодный для выполнения на компьютере (ассемблеры, компиляторы, интерпретаторы, загрузчики и редакторы связей), средства контроля и отладки программ. С, Basic, Pascal.

### **3. Файловая система.**

**Файловая система** — это базовая структура, предоставляющая средства именования, хранения и упорядочения файлов. Вся информация в компьютере хранится в кластерах, это четко нарезанные участки дискового пространства. Файл, конечно, в одном кластере не помещается, а проживает сразу в нескольких, и не обязательно чтобы они находились рядом, и чтобы не заблудится компьютер создает в самом начале жесткого диска существует специальный путеводитель по его содержанию FAT – таблица размещения файлов, именно там хранятся все сведения о том, какие именно кластеры занимает тот или иной файл или папка. Помимо этих файловых систем есть еще файловая система FAT32 и NTFS. Сейчас на большинстве компьютеров, где установлена операц. система 2000 XP используется NTFS.

Информация на дисках хранится в файлах.

Файл в переводе с английского языка это некоторая совокупность информации, документ, лист.

**Файл** – это поименованная область на диске или на другом машинном носителе.

В файлах может хранится тексты программ, рисунки, готовые к выполнению программы.(единица хранения данных)

Существует множество типов файлов. Это исполняемые файлы (программы), файлы конфигурации, содержащие параметры, необходимые для работы программы, файлы библиотеки, содержащие описания типовых процедур, файлы документы. Как же все их отличать? Да по имени. Каждый файл имеет свое имя, состоящее из двух частей: собственно имя файла – название и индентификатор , определяющий его, тип называемый расширением Каждый файл на диске имеет обозначение которое состоит из двух частей: имени и расширения (все вместе называется именем файла). Имя может быть выбрано произвольно, а вот расширение привязано к файлу и изменять его противопоказано.

До появления операционной системы Windows: имя файла состояло из 8 символов, а его расширение из трех – «короткое имя» (и имя и расширение могли включать только алфавитно цифровые символы латинского алфавита), Недостаток коротких имен их низкая содержательность.

С появлением Windows, было введено понятие длинного имени. Оно может содержать от 1 до 256 символов (все знаки кроме :\*?><|, можно использовать пробелы и несколько точек). Расширение идет после последней точки, и букв может быть больше трех. Но общепринято от 1 до 3 символов.

В компьютерном мире существует бесчисленное множество расширений и запомнить все просто не реально, однако основных не так уж много:

**Com, exe** – «исполняемый файл», содержит в себе готовые к выполнению прог-мы.

**Bat** – так называемый «пакетный файл», предназначенный для пользовательского запуска нескольких программ (по сути это текстовый файл, в котором набраны названия программных файлов, которые вы хотите выполнить в определенном порядке, например autoexec/bat автоматически выполняющийся файл при загрузке)

**Txt, doc** – текстовый файл

**Bmp, jpg, tif** – графические файлы (рисунков)

**Xls** – Excel , электронные таблицы

**Dll** – называемая динамически подключающаяся библиотека данных, к которой могут обратиться по мере надобности сразу несколько программ

Arj, Zip, Rar – файлы архивов, т.е. сжатой с помощью специальных программ информации.

### **Атрибуты файла**

**Скрытый** – эти файлы обычно не видны пользователю, как правило эти файлы очень важны для функционирования системы, хотя опытному пользователю не составит труда настроить программу просмотра таким образом, чтобы эти файлы были видны

**Только для чтения** – эти файлы всегда доступны взору, но изменить их содержание нельзя, по крайней мере без специальной команды пользователя.

**Системный** – этим атрибутом, отмечены самые важные файлы в операционной системе, отвечающие за загрузку компьютера. Их повреждение и удаление всегда влечет за собой тяжкие последствия и часто такие файлы награждены еще двумя атрибутами только для чтения и скрытый

**Архивный** – файл архива

Имена файлов регистрируются на магнитных дисках в каталогах (или директориях или папках)

**Каталог (директория, папка)** – это специальное место на диске в котором хранятся имена файлов, сведения о размерах, времени их последнего обновления.

Если каталог находится внутри какого-нибудь каталога то он называется подкаталогом этого каталога.

При записи пути к файлу все промежуточные каталоги разделяются между собой \

C:\мои документы\рефераты\реф.doc

### **Технология работы в Windows commander.**

1.Запустите Windows commander. (Пуск, Программы, Windows commander. Windows commander 32).

При загрузке программы окно программы делится на две панели: левую и правую. Их содержание может быть одинаковым, а может быть различным. Чтобы перейти из одной панели в другую достаточно щелкнуть мышкой по одной из них либо нажать клавишу **Tab**. Для выбора диска (A, C, D) на левой панели нажмите **Alt+F1** и в появившемся окне выберите нужный диск. На правой панели **Alt+F2**.

**Текстовые редакторы**, предназначены для набора и редактирования текста и иногда включает в себя простейшие элементы форматирования. (Word Pad, Блокнот)

**Текстовые процессоры**. Программы позволяющие не только вводить редактировать и форматировать текстовые данные, но так же обеспечивать взаимодействие текста, графики, таблиц.

**Редактирование** текста это изменение структуры текста (размещение)

**Форматирование** текста это процесс оформления внешнего вида документа в целом или его фрагмента.

Одним из самых известных и мощных текстовых процессоров является Word.

Для того чтобы войти в среду текстового процессора Word. Для этого щелкните по кнопке Пуск и в Главном меню выберите команду **Программы, Microsoft Word**.

### **Рассказать основные элементы диалогового окна Word**

Строка меню – главное меню: Файл, Правка...

Панель инструментов – Стандартная, Форматирование.

Рабочее поле

Строка состояния.

В текстовом процессоре Word имеется большое количество различных меню. Часть из них все время находится на экране, часть скрыта и появляется только после вызова.

**Главное меню** содержит девять раскрывающихся меню, которые позволяют выполнять большинство встречающихся на практике процедур работы с документами.

**Файл.** Создает новые файлы, сохраняет файлы. Находит и открывает ранее созданные файлы. Обеспечивает форматирование текста на странице, а также просмотр и печать документов.

**Правка.** Перемещает, копирует, удаляет фрагменты текста и объекты. Ищет и заменяет фрагменты текста.

**Вид.** Позволяет просматривать документ и его структуру различными способами; изменять масштаб отображения документа; оформлять колонтитулы. Вызывает на экран или скрывает панели инструментов, выполняет настройку панелей.

**Вставка.** Позволяет вставлять объекты, автоматизирует добавление специальной информации (номеров страниц, даты, сносок и ссылок). Обеспечивает автоматическое составление оглавления документа.

**Формат.** Позволяет выполнить нужное форматирование документа.

**Сервис.** Позволяет выбрать язык, проверить правописание документа, выполнить автозамену. Позволяет просмотреть статистику документа. Здесь также можно изменить опции программы, настроить параметры.

**Таблица.** Автоматизирует процедуры создания, правки и преобразования таблиц.

**Окно.** Переключает окна различных документов, организует удобное размещение открытых окон.

**Справка.** Здесь находится различная информация, содержащая подсказки пользователю.

Вывод панелей инструментов производится с помощью команды: **Вид, Панели инструментов.** Можно настроить свою панель инструментов: **Вид, Панели инструментов,**

**Настройка.** Если подвести указатель мыши к любому из значков то компьютер выведет подсказку, о том что означает данная клавиша.

#### **Режимы отображения документов:**

Установка режима производится соответствующей командой меню Вид (для режима предварительного просмотра — меню Файл). Кроме того, четыре первых режима можно вызывать, нажав соответствующую кнопку, находящуюся слева внизу экрана.

В обычном режиме представляется только содержательная часть документа без элементов оформления (этот режим используется на начальном этапе набора, редактирования текста)

В режиме Web-документа можно просматривать и редактировать документ в том виде, в каком он отображается в обозревателе.

Режим схемы документа предназначен для просмотра документов и их схемы на экране дисплея. Для облегчения чтения текст в каждом абзаце представлен по возможности крупнее и размещен точно в окне текста. Слева от окна текста текущего фрагмента документа отображается схема документа, где представлены заголовки подразделов разных документов. Щелкнув мышью по соответствующему заголовку, можно переместиться в нужный подраздел. Уровень иерархии выводимых разделов можно изменять.

Режим структуры, отображает только заголовки документа (этот режим полезен когда разработку документа начинают с создания плана, позволяет автоматизировано создать оглавление документа и формировать заголовки документа)

Режим разметки, предназначен для основной работы с текстом, в этом режиме документ представлен в том виде в котором будет распечатан.

Режим предварительного просмотра позволяет просмотреть печатные страницы документа в уменьшенном виде для проверки правильности разбиения текста на страницы. При вызове команды на экране появляется панель инструментов просмотра печати, с помощью которой можно указывать требуемый масштаб увеличения изображения, выводить на экран несколько печатных страниц и т. д. В режиме предварительного просмотра возможна также корректировка печатного текста.

В режиме просмотра во весь экран можно видеть текст документа без лишних экранных

элементов. В данном режиме выбирать команды можно только с помощью «горячих» клавиш или из контекстного меню.

## 1. 6 Лекция №6 (2 часа)

Тема: «Понятие информации»

### 1.6.1 Вопросы лекции:

1. Информация. Место и роль информации в информатике.
2. Свойства информации.
3. Сигналы. Данные.
4. Виды информации. Формы представления и передачи информации.
5. Количество информации, единицы измерения информации.

### 1.6.2 Краткое содержание вопросов:

1. Информация. Место и роль информации в информатике.

Слово информация происходит от латинского *informatio* что в переводе означает сведения, разъяснение, ознакомление. До начала промышленной революции, определение сути информации оставалось прерогативой философов. И рассматривать теорию информации стала лишь в 70 года наука кибернетика.

В настоящее время не существует единого термина информации

На бытовом уровне. **Информация** это— некоторые сведения, данные, знания которые кого-либо интересуют. *Например*: сообщение о каких-либо событиях, о чьей-либо деятельности.

Информатика рассматривает информацию как связанные между собой сведения, изменяющие наши представления о явлении или объекте окружающего мира.

Применительно к компьютерной обработке данных, **под информацией** понимается некая последовательность символов (букв, цифр, звуков и т.п.), несущую смысл и представленную в понятном для компьютера виде.

«Информировать» означает «сообщить нечто неизвестное ранее». Одно и тоже сообщение (статья, объявление, письмо, чертеж) может содержать разное количество информации для разных людей – в зависимости от уровня понимания этого сообщения и интереса к нему (так сообщение на японском языке не несет никакой новой информации человеку не знающему этого языка).

Способы восприятия (Визуальная , Аудиальная , Тактильная , Обонятельная, Вкусовая)

Информация может существовать в различных формах:

- текстовая- представленная в виде в виде символов
- числовая- в виде цифр и знаков, обозначающих математические операции
- графическая- в виде изображений и графиков
- звуковая
- а так же она может быть представлена в виде световых сигналов, радиоволн, жестов и мимики, запахов и ощущений.

Информация передается от источника к получателю посредством какой-нибудь среды – она называется **канал связи**. Сообщение которое передается по каналам связи называется **сигналом**.

### 2. Свойства информации:

1. достоверность верность информации, не вызывающая сомнений. (не достоверная информация может привести к неправильному пониманию и принятию неправильных решений, достоверная информация со временем может стать не достоверной –устареть)

2. полнота или точность, т.е. она содержит минимальный, достаточный для принятия решения набор показателей (как неполная так же как и избыточная информация может привести к неправильному пониманию и принятию неправильных решений)
3. ценность или актуальность – степень соответствия ее текущему времени (важность информации для решения задачи)
4. своевременность (преждевременная подача (когда она еще не может быть усвоена) нежелательна)
5. понятность и адекватность- степень соответствия реально полученной информации и ее ожидаемого содержимого (непонятная информация не может быть усвоена, напр. на другом языке)
6. доступность мера возможности получать ту или иную информацию (по уровню восприятия, поэтому одни темы в разных учебниках объясняются по разному)

#### Предназначение:

Массовая- содержит сведения и понятия понятные большей части людей.

Специальная – необходимая и понятная в рамках узкой социальной группы

Личная – сведения о какой либо личности.

Опр. Информационные процессы – это отбор (процесс когда информация приобретает определенное представление или форму), хранение (распространение информации во времени), передача (распространение в пространстве), преобразование и использование информации.

#### **3. Единицы измерения количества информации.**

За количество информации принято брать объем компьютерной памяти и объем данных, передаваемых по каналам связи.

За количество информации, принимают кол-во символов необходимых для ее представления в двоичном алфавите. Единица информации **бит**. Кол-во информации при этом подходе всегда целое число. Бит слишком маленькая единица, чаще применяется более крупная – **БАЙТ** = 8 Бит. Именно восемь бит требуется для того чтобы закодировать 256 символов алфавита клавиатуры.

1 байт = 8 бит.

1 килобайт = 1024 байт =  $2^{10}$  байт

1 Мегабайт = 1024 килобайт =  $2^{20}$  байт

1 Гигабайт = 1024 мегабайт =  $2^{30}$  байт

### **1. 7 Лекция №7 (2 часа)**

**Тема:** «Общая характеристика информационных процессов»

#### **1.7.1 Вопросы лекции:**

1. Информационный процесс.
2. Носители информации.
3. Кодирование информации.
4. Алгоритмы сжатия данных.

#### **1.7.2 Краткое содержание вопросов:**

1. Информационный процесс.

Опр. Информационные процессы – это отбор (процесс когда информация приобретает определенное представление или форму), хранение (распространение информации во времени), передача (распространение в пространстве), преобразование и использование информации.

#### **Единицы измерения количества информации.**

За количество информации принято брать объем компьютерной памяти и объем данных, передаваемых по каналам связи.

За количество информации, принимают кол-во символов необходимых для ее представления

в двоичном алфавите. Единица информации *бит*. Кол-во информации при этом подходе всегда целое число. Бит слишком маленькая единица, чаще применяется более крупная – БАЙТ = 8 Бит. Именно восемь бит требуется для того чтобы закодировать 256 символов алфавита клавиатуры.

1байт = 8 бит.

1 килобайт = 1024 байт =  $2^{10}$  байт

1 Мегабайт = 1024 килобайт =  $2^{20}$ байт

1 Гигабайт = 1024 мегабайт =  $2^{30}$ байт

## 2. Кодирование информации.

**Цель кодирования** – представление информации в более компактной форме.

**Кодирование информации** – это представление ее с помощью конечного набора знаков, отличных друг от друга, которые называются буквами.

Упорядоченное множество знаков называется алфавитом.

**Примеры**: русский алфавит, английский алфавит, азбука Морзе.

Совокупность правил построения кодовых обозначений объекта называется **системой кодирования**.

Как мы говорили ранее В ЭВМ для кодирования используются 2 символа- ноль и единица (0 и 1), и объем информации, необходимый для запоминания одного из этих двух символов называется 1 бит. Итак, если у нас есть один бит, то с его помощью мы можем закодировать один из двух символов- либо 0, либо 1. Если же есть 2 бита, то из них можно составить один из четырех вариантов кодов: 00, 01, 10, 11. N бит – то  $2^N$  вариантов.

СТАНДАРТНЫЙ НАБОР ИЗ 256 СИМВОЛОВ НАЗЫВАЕТСЯ ASCII (произносится "аски", означает "Американский Стандартный Код для Обмена Информацией"- англ. American Standard Code for Information Interchange). Он включает в себя большие и маленькие русские и латинские буквы, цифры, знаки препинания и арифметических действий и т.п.

КАЖДОМУ СИМВОЛУ ASCII СООТВЕТСТВУЕТ 8-БИТОВЫЙ ДВОИЧНЫЙ КОД, НАПРИМЕР: А - 01000001, В - 01000010, С - 01000011, D - 01000100, и т.д.

Объем информации, необходимый для запоминания одного символа ASCII называется 1 байт, поскольку под один стандартный ASCII-символ отводится 8 бит.

Пример: 1 килобайт соответствует примерно половине страницы текста, 1 мегабайт соответствует примерно 500 страницам текста, 1 гигабайт соответствует примерно 2 комплектам энциклопедии, 1 терабайт соответствует примерно 2000 комплектам энциклопедии.

Таким образом, если человек создает текстовый файл и записывает его на диск, то на самом деле каждый введенный человеком символ хранится в памяти компьютера в виде набора из восьми нулей и единиц.

Набор ASCII был разработан в США Американским Национальным Институтом Стандартов (ANSI), может быть использован и в других странах, если есть специальные программы, которые часть символов заменяют на другие, в частности на символы национального алфавита, в нашем случае - буквы кириллицы. Поэтому например, передавать по электронной почте за границу тексты, содержащие русские буквы, бессмысленно. В англоязычных странах на экране дисплея вместо русской буквы Ъ будет высвечиваться символ английского фунта стерлинга, вместо буквы р - греческая буква альфа, вместо буквы л - одна вторая и т.д.

Аналогом отечественной таблицы кодов является таблица КОИ-8.

## 1 Сжатие данных.

Сообщение записанное с помощью любого алфавита обладает избыточностью. Т.е. количество информации в вероятностном смысле меньше чем в объемном смысле. Степень избыточности зависит от типа данных. Например, видеоданные обладают более высокой степенью избыточности, чем графические данные, а графические данные, в свою очередь обладают более высокой степенью избыточности, чем текстовых данные.

Другим фактором, влияющим на степень избыточности является принятая система кодирования. Так, установлено, что кодирование текстовых данных с помощью русского языка дает в среднем избыточность на 20-25% большую, чем кодирование аналогичных данных средствами английского языка.

Для человека избыточность данных часто связана с качеством информации, поскольку избыточность, как правило, улучшает понятность и восприятие информации. Однако, когда речь идет о хранении и передаче информации средствами компьютерной техники, то избыточность играет отрицательную роль, поскольку она приводит к возрастанию стоимости хранения и передачи информации. Особенно актуальна эта проблема в случае обработки огромных объемов информации при незначительных объемах носителей данных. При работе с компьютером возникает проблема: непрерывный рост объема информации, которую нужно хранить на диске. В связи с этим, постоянно возникает необходимость уменьшения избыточности или сжатия данных.

Если методы сжатия данных применяются к готовым файлам, то часто употребляют термин "архивация данных", сжатый вариант данных называют **архивом**, а программные средства, которые реализуют методы сжатия называются **архиваторами**.

В зависимости от того, в каком объекте размещены данные, подлежащие сжатию различают:

1. Сжатие (архивация) файлов: используется для уменьшения размеров файлов при подготовке их к передаче каналами связи или к транспортированию на внешних носителях маленькой емкости;
2. Сжатие (архивация) папок: используется как средство уменьшения объема папок перед долгим хранением, например, при резервном копировании;
3. Сжатие (уплотнение) дисков: (как правило, средствами операционной системы). Это сжатие используют, если объем жесткого диска недостаточен для хранения требуемого объема информации, если какая-то информация не используется длительное время, но удалять ее нецелесообразно, поскольку она может потребоваться позже.

Сжатие всего диска используют редко, поскольку, во-первых, оно замедляет работу (при любом обращении к диску информацию нужно или сжимать при записи или возвращать к нормальному состоянию при считывании), во-вторых, информацию на таком диске сложнее восстановить при каких-либо сбоях, например при заражении вирусами. Архивацию, т.е. выборочное сжатие определенных файлов, применяют гораздо чаще.

Существует много практических алгоритмов сжатия данных, но все они базируются на трех теоретических способах уменьшения избыточности данных. Первый способ состоит в изменении содержимого данных, второй - в изменении структуры данных, а третий - в одновременном изменении как структуры, так и содержимого данных.

Если при сжатии данных происходит изменение их содержимого, то метод сжатия называется **необратимым**, то есть при восстановлении (разархивировании) данных из архива не происходит полное восстановление информации. Такие методы часто называются методами сжатия с регулированными потерями информации. Понятно, что эти методы можно применять только для таких типов данных, для которых потеря части содержимого не приводит к существенному искажению информации. К таким типам данных относятся видео- и аудиоданные, а также графические данные. Эти методы сжатия обеспечивают большую степень сжатия, но их нельзя применять к текстовым данным.

Примерами форматов сжатия с потерями информации могут быть:

- JPEG - для графических данных;
- MPG - для видеоданных;
- MP3 - для аудиоданных.

Если при сжатии данных происходит только изменение структуры данных, то метод сжатия называется **обратимым**. В этом случае, из архива можно восстановить информацию полностью. Обратимые методы сжатия можно применять к любым типам данных, но они дают меньшую степень сжатия по сравнению с необратимыми методами сжатия. Примеры

форматов сжатия без потери информации:

- GIF, TIFF - для графических данных;
- AVI - для видеоданных;
- ZIP, ARJ, RAR, CAB, LH - для произвольных типов данных.

Самые распространенные форматы сжатия ARJ, RAR, ZIP и соответствующие им программы-архиваторы, использующиеся на практике. (WinArj.exe WinRar.exe WinZip.exe). (Операционная система MS DOS Arj.exe–Arj.exe , Rar.exe–Unrar.exe, Pkzip.exe –Pkunzip.exe) Современные архиваторы предоставляют пользователю полный спектр услуг для работы с архивами, основными из которых являются: создание нового архива; добавление файлов в существующий архив; распаковывание файлов из архива; создание самораспаковывающихся архивов; выбор и настройка коэффициента сжатия; защита архивов паролями от несанкционированного доступа; просмотр содержимого файлов разных форматов без предварительного распаковывания; поиск файлов и данных внутри архива; проверка на вирусы в архиве к распаковыванию; создание распределенных архивов фиксированного размера для носителей маленькой емкости;

Существует много разных практических методов сжатия без потери информации, в основе этих методов лежат три теоретических алгоритма:

- алгоритм RLE (Run Length Encoding);
- алгоритм Хаффмана.
- алгоритмы группы KWE (KeyWord Encoding);

#### **Алгоритм RLE**

В основе алгоритма RLE лежит идея выявления повторяющихся последовательностей данных и замены их более простой структурой, в которой указывается код данных и коэффициент повторения.

Например, пусть задана такая последовательность данных, что подлежит сжатию: 1 1 1 1 2 2 3 4 4 4

В алгоритме RLE предлагается заменить ее следующей структурой: 1 4 2 2 3 1 4 3, где первое число каждой пары чисел - это код данных, а второе - коэффициент повторения. (Если для хранения каждого элемента последовательности отводится 1 байт, то вся последовательность будет занимать 10 байт памяти, а (сжатый вариант) будет занимать 8 байт памяти.)

Коэффициент сжатия, характеризующий степень сжатия, можно вычислить по формуле:

$$k = \frac{V_x}{V_n} \cdot 100\% = \frac{8}{10} \cdot 100\% = 80\%$$

где  $V_x$  - объем памяти, необходимый для хранения выходной (результатирующей) последовательности данных,  $V_n$  - входной последовательности данных.

Чем меньше значение коэффициента сжатия, тем эффективней метод сжатия. Понятно, что алгоритм RLE будет давать лучший эффект сжатия при большей длине повторяющейся последовательности данных. В случае рассмотренного выше примера, если входная последовательность будет иметь такой вид: 1 1 1 1 1 1 3 4 4 4, то коэффициент сжатия будет равен 60%. В связи с этим большая эффективность алгоритма RLE достигается при сжатии графических данных (особенности для однотонных изображений).

#### **Алгоритм Хаффмана**

В основе алгоритма Хаффмана лежит идея кодирования битовыми группами. Сначала устанавливается частота вхождения каждого символа, встречающегося в последовательности. После этого, символы сортируются по уменьшению частоты вхождения.

Основная идея состоит в следующем: чем чаще встречается символ, тем меньшим количеством бит он кодируется. Результат кодирования заносится в словарь, необходимый для декодирования. Рассмотрим простой пример, иллюстрирующий работу алгоритма Хаффмана.

Пусть задан текст, в котором бурва 'A' входит 10 раз, буква 'B' - 8 раз, 'C' - 6 раз, 'D' - 5 раз, 'E'

и 'F' - по 4 раза. Тогда один из возможных вариантов кодирования по алгоритму Хаффмана приведен в таблицы 1.

Таблица 1.

его для в	Символ	Частота вхождения	Битовый код
	A	10	00
	B	8	01
	C	6	100
	D	5	101
	E	4	110
	F	4	111

Алгоритм Хаффмана универсальный, можно применять для сжатия данных любых типов, но он малоэффективен файлов маленьких размеров (за счет необходимости сохранение словаря).

#### Алгоритмы группы KWE

основе алгоритма сжатия по ключевым словам положен принцип кодирования лексических единиц группами байт фиксированной длины. Примером лексической единицы может быть обычное слово.

На практике, на роль лексических единиц выбираются повторяющиеся последовательности символов, которые кодируются цепочкой символов (кодом) меньшей длины. Результат кодирования помещается в таблице, образовывая так называемый словарь.

Существует довольно много реализаций этого алгоритма, среди которых наиболее распространенными являются алгоритм Лемпеля-Зива (алгоритм LZ) и его модификация алгоритм Лемпеля-Зива-Велча (алгоритм LZW). Словарем в данном алгоритме является потенциально бесконечный список фраз. Алгоритм начинает работу с почти пустым словарем, который содержит только одну закодированную строку, так называемая NULL-строка. При считывании очередного символа входной последовательности данных, он прибавляется к текущей строке. Процесс продолжается до тех пор, пока текущая строка соответствует какой-нибудь фразе из словаря. Но рано или поздно текущая строка перестает соответствовать какой-нибудь фразе словаря. В момент, когда текущая строка представляет собой последнее совпадение со словарем плюс только что прочитанный символ сообщения, кодер выдает код, который состоит из индекса совпадения и следующего за ним символа, который нарушил совпадение строк. Новая фраза, состоящая из индекса совпадения и следующего за ним символа, прибавляется в словарь. В следующий раз, если эта фраза появится в сообщении, она может быть использована для построения более длинной фразы, что повышает меру сжатия информации.

Алгоритм LZW построен вокруг таблицы фраз (словаря), которая заменяет строки символов сжимаемого сообщения в коды фиксированной длины. Таблица имеет так называемое свойством опережения, то есть для каждой фразы словаря, состоящей из некоторой фразы w и символа K, фраза w тоже заносится в словарь. Если все части словаря полностью заполнены, кодирование перестает быть адаптивным (кодирование происходит исходя из уже существующих в словаре фраз).

Алгоритмы сжатия этой группы наиболее эффективны для текстовых данных больших объемов и малоэффективны для файлов маленьких размеров (за счет необходимости сохранение словаря).

На практике программные средства сжатия данных синтезируют эти три "чистых" алгоритмы, поскольку их эффективность зависит от типа и объема данных.

## Тема: «Системы исчисления»

### 1.8.1 Вопросы лекции:

1. История развития систем счисления.
2. Позиционные и непозиционные системы счисления.
3. Двоичная, десятичная, восьмеричная и шестнадцатеричная система счисления.

### 1.8.2 Краткое содержание вопросов:

#### 1. Системы счисления.

**Система счисления** – это способ записи чисел с помощью заданного набора специальных знаков (цифр). Существуют позиционные и непозиционные системы счисления.

В **непозиционных системах** **вес цифры** (т.е. тот вклад, который она вносит в значение числа) **не зависит от ее позиции** в записи числа. Так, в римской системе счисления в числе XXXII (тридцать два) вес цифры X в любой позиции равен просто десяти.

В **позиционных системах** счисления **вес каждой цифры изменяется в зависимости от ее положения** (позиции) в последовательности цифр, изображающих число. Например, в числе 757,7 первая семерка означает 7 сотен, вторая – 7 единиц, а третья – 7 десятых долей единицы.

Сама же запись числа 757,7 означает сокращенную запись выражения  
 $700 + 50 + 7 + 0,7 = 7 \cdot 10^2 + 5 \cdot 10^1 + 7 \cdot 10^0 + 7 \cdot 10^{-1} = 757,7$ .

Любая позиционная система счисления характеризуется своим **основанием**.

#### 2. Основание позиционной системы счисления — это количество различных знаков или символов, используемых для изображения цифр в данной системе.

За основание системы можно принять любое натуральное число — два, три, четыре и т.д. Следовательно, **возможно бесконечное множество позиционных систем**: двоичная, троичная, четверичная и т.д. Запись чисел в каждой из систем счисления с основанием  $q$  означает сокращенную запись выражения

$$a_{n-1} q^{n-1} + a_{n-2} q^{n-2} + \dots + a_1 q^1 + a_0 q^0 + a_{-1} q^{-1} + \dots + a_{-m} q^{-m},$$

где  $a_i$  – цифры системы счисления;  $n$  и  $m$  – число целых и дробных разрядов, соответственно. Например:  $1001_2 = 1 \cdot 2^0 + 0 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3$

#### Какие системы счисления используют специалисты для общения с компьютером?

3. Кроме десятичной широко используются системы с основанием, являющимся **целой степенью числа 2**, а именно:

- **двоичная** (используются цифры 0, 1);
- **восьмеричная** (используются цифры 0, 1, ..., 7);
- **шестнадцатеричная** (для первых целых чисел от нуля до девяти используются цифры 0, 1, ..., 9, а для следующих чисел — от десяти до пятнадцати – в качестве цифр используются символы A, B, C, D, E, F).

Полезно запомнить запись в этих системах счисления первых двух десятков целых чисел:

Данные и команды в компьютере представляются в виде двоичных последовательностей различной структуры и длины.

Почему в компьютере информация кодируется двоичным кодом. Это связано с тем, что информацию, представленную в таком виде, легко технически смоделировать, например в виде электрических сигналов. Если в какой-то момент времени по проводнику идет ток, то по нему передается единица, если тока нет – ноль. Поэтому компьютеры используют двоичную систему. Существует еще ряд **преимуществ перед другими системами**:

10 - я	2 - я	8 - я	16 - я	10 - я	2 - я	8 - я	16 - я
0	0	0	0	10	1010	12	A
1	1	1	1	11	1011	13	B

2	10	2	2	12	1100	14	C
3	11	3	3	13	1101	15	D
4	100	4	4	14	1110	16	E
5	101	5	5	15	1111	17	F
6	110	6	6	16	10000	20	10
7	111	7	7	17	10001	21	11
8	1000	10	8	18	10010	22	12
9	1001	11	9	19	10011	23	13

- представление информации посредством только двух состояний надежно и помехоустойчиво;
- двоичная арифметика намного проще десятичной. Недостаток двоичной системы — быстрый рост числа разрядов, необ

ходимых для записи чисел.

Почему люди пользуются десятичной системой, а компьютеры — двоичной?

Люди предпочитают десятичную систему, вероятно, потому, что с древних времен считали по пальцам, а пальцев у людей по десять на руках и ногах. Не всегда и не везде люди пользуются десятичной системой счисления. В Китае, например, долгое время пользовались пятеричной системой счисления. Двоичная система, удобная для компьютеров, для человека неудобна из-за ее громоздкости и непривычной записи.

Недостаток двоичной системы — **быстрый рост числа разрядов**, необходимых для записи чисел.

Двоичная система, удобная для компьютеров, для человека неудобна из-за ее *громоздкости и непривычной записи*.

## 1. 9 Лекция №9 (2 часа)

Тема: «Арифметические действия в различных системах исчисления»

### 1.9.1 Вопросы лекции:

1. Перевод чисел из одной системы счисления в другую и обратно.
2. Арифметические действия над числами в различных системах счисления.

### 1.9.2 Краткое содержание вопросов:

1. Перевод восьмеричных и шестнадцатеричных чисел в двоичную систему очень прост: достаточно каждую цифру заменить эквивалентной ей двоичной триадой (тройкой цифр) или тетрадой (четверкой цифр).

$$01101100 = 6C$$

$$00111001 = 39$$

$$00100011 = 23$$

$$01101100 = 154$$

$$111001 = 71$$

Для переводе целого десятичного числа в систему с основанием  $q$  его необходимо последовательно делить на  $q$  до тех пор, пока не останется остаток, меньший или равный  $q-1$ . Число в системе с основанием  $q$  записывается как последовательность остатков от деления, записанных в обратном порядке, начиная с последнего.

При переводе числа из двоичной (восьмеричной, шестнадцатеричной) системы в десятичную надо это число представить в виде суммы степеней основания его системы счисления.

Как перевести целое число из десятичной системы в любую другую позиционную систему счисления?

При переводе целого десятичного числа в систему с основанием  $q$  его необходимо

последовательно делить на  $q$  до тех пор, пока не останется остаток, меньший или равный  $q-1$ . Число в системе с основанием  $q$  записывается как последовательность остатков от деления, записанных в обратном порядке, начиная с последнего.

**Пример:** Перевести число 75 из десятичной системы в двоичную, восьмеричную и шестнадцатеричную:

**Ответ:**  $75_{10} = 1001011_2 = 113_8 = 4B_{16}$ .

59, 111, 99, - в двоичную

193, 702, 59, 172 – в 16-ти рич

**Как перевести число из двоичной (восьмеричной, шестнадцатеричной) системы в десятичную?**

При переводе числа из двоичной (восьмеричной, шестнадцатеричной) системы в десятичную надо это число представить в виде суммы степеней основания его системы счисления.

**Примеры:**

1100011, 10001001, 1001010, 10101100 из двоичной

13B, C1, 5F, AD. – из 16-рич

## 2. Арифметические операции в позиционных системах счисления?

Рассмотрим основные арифметические операции: *сложение, вычитание, умножение*.

### Сложение

При сложении цифры суммируются по разрядам, и если при этом возникает избыток, то он переносится влево.

#### Сложение в двоичной системе

$$\begin{array}{r} 11011 \\ + 11011 \\ \hline 110110 \end{array}$$

1.  $1+1=0$  (1 на след разряд)
2.  $1+1+1=1$  (1 на след разряд)
3.  $0+0+1=1$
4.  $1+1=0$  (1 на след разряд)
5.  $1+1+1=1$  (1 на след разряд)

### Вычитание

2. **Обратный код.** Получается инвертированием всех цифр двоичного кода абсолютной величины числа, включая разряд знака: нули заменяются единицами, а единицы — нулями. Например:

3. **Дополнительный код.** Получается образованием обратного кода с последующим прибавлением единицы к его младшему разряду. Например:

Получен правильный результат в дополнительном коде. Единицу переноса из знакового разряда компьютер **отбрасывает**.

#### Сложение и вычитание в шестнадцатиричной системе

### Умножение

Выполняя умножение многозначных чисел в различных позиционных системах счисления, можно использовать обычный алгоритм перемножения чисел в столбик, но при этом результаты перемножения и складываются в той системе счисления в какой мы решаем

#### Сложение в восьмеричной системе

$$\begin{array}{r} 1357 \\ + 5573 \\ \hline 7252 \end{array}$$

1.  $7+3=10$  (больше чем 8-рич десяток на 2 единицы, пишем ноль, а две единицы на след. разряд)
- 2.

## 1. 10. Лекция №10 (2 часа)

Тема: «Основы теории защиты информации»

### 1.10.1 Вопросы лекции:

1. Необходимость защиты информации.
2. Компьютерные вирусы. История возникновения.
3. Классификация вирусов.

### 1.10.2 Краткое содержание вопросов:

1. Необходимость защиты информации.

Информация, с точки зрения информационной безопасности, обладает категориями:

- **конфиденциальность** – гарантия, что конкретная информация доступна только тому кругу лиц, для кого она предназначена; нарушение этой категории называется хищением/раскрытием информации
- **целостность** – гарантия, что в информации при ее хранении или передаче не было произведено несанкционированных изменений; нарушение этой категории - фальсификация сообщения
- **аутентичность** – гарантия, что источником информации является именно то лицо, которое заявлено как ее автор; нарушение этой категории - фальсификация, но уже автора сообщения
- **апеллируемость** – гарантия, что при необходимости можно будет доказать, что автором сообщения является именно заявленный человек, и не может являться никто другой; отличие этой категории от предыдущей в том, что при подмене автора, кто-то другой пытается заявить, что он автор сообщения, а при нарушении апеллируемости – сам автор пытается "откреститься" от своих слов

Информационные системы обладают категориями:

- **надежность** – гарантия, что система ведет себя в нормальном и внештатном режимах так, как запланировано
- **точность** – гарантия точного и полного выполнения всех команд
- **контроль доступа** – гарантия, что различные группы лиц имеют различный доступ к информационным объектам, и эти ограничения доступа постоянно выполняются
- **контролируемость** – гарантия, что в любой момент может быть произведена полноценная проверка любого компонента программного комплекса
- **контроль идентификации** – гарантия, что клиент, подключенный в данный момент к системе, является именно тем, за кого себя выдает
- **устойчивость к умышленным сбоям** – гарантия, что при умышленном внесении ошибок в пределах заранее оговоренных норм система будет вести себя так, как оговорено заранее

### Необходимость защиты информации:

сейчас, с одной стороны, сильно расширилось использование компьютерных сетей, по которым передаются большие объемы информации государственного, военного, коммерческого и частного характера, не допускающего возможность доступа к ней посторонних лиц. С другой стороны, появление новых мощных компьютеров, технологий сетевых и нейронных вычислений сделало возможным дискредитацию криптографических систем еще недавно считавшихся практически не раскрываемыми.

### 2. Компьютерные вирусы. История возникновения.

Можно привести массу фактов, свидетельствующих о том, что угроза информационному ресурсу возрастает с каждым днем, подвергая в панику ответственных лиц в банках, на

предприятиях и в компаниях во всем мире. И угроза эта исходит от компьютерных вирусов, которые искажают или уничтожают жизненно важную, ценную информацию, что может привести не только к финансовым потерям, но и к человеческим жертвам. Вирус - это специально написанная небольшая по размерам программа, которая может "приписывать" себя к другим программам (т.е. "заражать" их), а также выполнять различные нежелательные действия на компьютере. Программа, внутри которой находится вирус, называется "зараженной". Когда такая программа начинает работу, то сначала управление получает вирус. Вирус находит и "заражает" другие программы, а также выполняет какие-нибудь вредные действия (например, портит файлы или таблицу размещения файлов на диске, "засоряет" оперативную память и т.д.). Для маскировки вируса действия по заражению других программ и нанесению вреда могут выполняться не всегда, а, скажем, при выполнении определенных условий. После того как вирус выполнит нужные ему действия, он передает управление той программе, в которой он находится, и она работает также, как обычно. Тем самым внешне работа зараженной программы выглядит так же, как и незараженной. Разновидности вирусов устроены так, что при запуске зараженной программы вирус остается резидентно, т.е. до перезагрузки DOS, компьютера и время от времени заражает программы и выполняет вредные действия на компьютере. Компьютерный вирус может испортить, т.е. изменить ненадлежащим образом, любой файл на имеющихся в компьютере дисках. Но некоторые виды файлов вирус может "заразить". Это означает, что вирус может "внедриться" в эти файлы, т.е. изменить их так, что они будут содержать вирус, который при некоторых обстоятельствах может начать свою работу. Следует заметить, что тексты программ и документов, информационные файлы без данных, таблицы табличных процессоров и другие аналогичные файлы не могут быть заражены вирусом, он может их только испортить. В настоящее время известно более 87800 вирусов, число которых непрерывно растет. Известны случаи, когда создавались учебные пособия, помогающие в написании вирусов.

Основные виды вирусов: загрузочные, файловые, файлово-загрузочные. Наиболее опасный вид вирусов - полиморфные. Из истории компьютерной вирусологии ясно, что любая оригинальная компьютерная разработка заставляет создателей антивирусов приспосабливаться к новым технологиям, постоянно усовершенствовать антивирусные программы.

Причины появления и распространения вирусов скрыты с одной стороны в психологии человека, с другой стороны - с отсутствием средств защиты у операционной системы.

Основные пути проникновения вирусов - съемные диски и компьютерные сети. Чтобы этого не случилось, соблюдайте меры по защите. Также для обнаружения, удаления и защиты от компьютерных вирусов разработано несколько видов следствием не вполне ясного понимания предмета.

Вирус - программа, обладающая способностью к самовоспроизведению. Такая способность является единственным средством, присущим всем типам вирусов. Но не только вирусы способны к самовоспроизведению. Любая операционная система и еще множество программ способны создавать собственные копии. Копии же вируса не только не обязаны полностью совпадать с оригиналом, но, и могут вообще с ним не совпадать!

Вирус не может существовать в «полной изоляции»: сегодня нельзя представить себе вирус, который не использует код других программ, информацию о файловой структуре или даже просто имена других программ. Причина понятна: вирус должен каким-нибудь способом обеспечить передачу себе управления. 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. Файловые вирусы могут внедряться и в другие типы файлов, но, как правило, записанные в таких файлах, они никогда не получают управление и, следовательно, теряют способность к

размножению. Загрузочные вирусы внедряются в загрузочный сектор диска (Boot-с) или в сектор, содержащий программу загрузки системного диска (Master Boot Record). Файловые загрузочные вирусы заражают как файлы, так и загрузочные сектора дисков.

По способу заражения вирусы делятся на резидентные и нерезидентные.

Резидентный вирус при заражении (инфицировании) компьютера оставляет в оперативной памяти свою резидентную часть, которая потом перехватывает обращение операционной системы к объектам заражения (файлам, загрузочным секторам дисков и т. п.) и внедряется в них. Резидентные вирусы находятся в памяти и являются активными вплоть до выключения или перезагрузки компьютера.

Нерезидентные вирусы не заражают память компьютера и являются активными ограниченное время.

По степени воздействия вирусы можно разделить на следующие виды: неопасные, не мешающие работе компьютера, но уменьшающие объем свободной оперативной памяти и памяти на дисках, действия таких вирусов проявляются в каких-либо графических или звуковых эффектах опасные вирусы, которые могут привести к различным нарушениям в работе компьютера очень опасные, воздействие которых может привести к потере программ, уничтожению данных, стиранию информации в системных областях диска.

По особенностям алгоритма вирусы трудно классифицировать из-за большого разнообразия. Простейшие вирусы - паразитические, они изменяют содержимое файлов и секторов диска и могут быть достаточно легко обнаружены и уничтожены. Можно отметить вирусы-репликаторы, называемые червями, которые распространяются по компьютерным сетям, вычисляют адреса сетевых компьютеров и записывают по этим адресам свои копии. Известны вирусы-невидимки,

называемые стелс-вирусами, которые очень трудно обнаружить и обезвредить, так как они перехватывают обращения операционной системы к пораженным файлам и секторам дисков и подставляют вместо своего тела незараженные участки диска.

Наиболее трудно обнаружить вирусы-мутанты, содержащие алгоритмы шифровки-расшифровки, благодаря которым копии одного и того же вируса не имеют ни одной повторяющейся цепочки байтов. Имеются и так называемые квазивирусные или «троянские» программы, которые хотя и не способны к самораспространению, но очень опасны, так как, маскируясь под полезную программу, разрушают загрузочный сектор и файловую систему дисков.

### **Проявление наличия вируса в работе на ПЭВМ.**

Все действия вируса могут выполняться достаточно быстро и без выдачи каких-либо сообщений, поэтому пользователю очень трудно заметить, что в компьютере происходит что-то необычное.

Пока на компьютере заражено относительно мало программ, наличие вируса может быть практически незаметно. Однако по прошествии некоторого времени на компьютере начинает твориться что-то странное, например:

- \* некоторые программы перестают работать или начинают работать неправильно;
- \* на экран выводятся посторонние сообщения, символы и т.д.;
- \* работа на компьютере существенно замедляется;
- \* некоторые файлы оказываются испорченными и т.д.

К этому моменту, как правило, уже достаточно много (или даже большинство) программ являются зараженными вирусом, а некоторые файлы и диски - испорченными. Более того, зараженные программы с одного компьютера могли быть перенесены с помощью дискет или по локальной сети на другие компьютеры.

Некоторые виды вирусов ведут себя еще более коварно. Они вначале незаметно заражают большое число программ или дисков, а потом причиняют очень серьезные повреждения, например, формируют весь жесткий диск на компьютере. А бывают вирусы, которые

стараются вести себя как можно более незаметно, но понемногу и постепенно портят данные на жестком диске компьютера.

Таким образом, если не предпринимать мер по защите от вируса, то последствия заражения компьютера могут быть очень серьезными.

## **1. 11 Лекция №11 (2 часа)**

**Тема: «Методы защиты информации»**

### **1.11.1 Вопросы лекции:**

1. Физические методы защиты информации.
2. Программные методы защиты информации.
3. Антивирусные программы.

### **1.11.2 Краткое содержание вопросов:**

#### **1. Физические методы защиты информации.**

Физические средства защиты – это разнообразные устройства, приспособления, конструкции, аппараты, изделия, предназначенные для создания препятствий на пути движения злоумышленников.

К физическим средствам относятся механические, электромеханические, электронные, электронно-оптические, радио- и радиотехнические и другие устройства для воспрещения несанкционированного доступа (входа, выхода), проноса (выноса) средств и материалов и других возможных видов преступных действий.

Эти средства применяются для решения следующих задач:

- 1) охрана территории предприятия и наблюдение за ней;
- 2) охрана зданий, внутренних помещений и контроль за ними;
- 3) охрана оборудования, продукции, финансов и информации;
- 4) осуществление контролируемого доступа в здания и помещения.

Все физические средства защиты объектов можно разделить на три категории: средства предупреждения, средства обнаружения и системы ликвидации угроз. Охранная сигнализация и охранное телевидение, например, относятся к средствам обнаружения угроз; заборы вокруг объектов – это средства предупреждения несанкционированного проникновения на территорию, а усиленные двери, стены, потолки, решетки на окнах и другие меры служат защитой и от проникновения, и от других преступных действий (подслушивание, обстрел, бросание гранат и взрывпакетов и т. д.). Средства пожаротушения относятся к системам ликвидации угроз.

#### **2. Программные методы защиты информации.**

К аппаратным средствам защиты информации относятся самые различные по принципу действия, устройству и возможностям технические конструкции, обеспечивающие пресечение разглашения, защиту от утечки и противодействие несанкционированному доступу к источникам конфиденциальной информации.

Аппаратные средства защиты информации применяются для решения следующих задач:

- 1) проведение специальных исследований технических средств обеспечения производственной деятельности на наличие возможных каналов утечки информации;
- 2) выявление каналов утечки информации на разных объектах и в помещениях;
- 3) локализация каналов утечки информации;
- 4) поиск и обнаружение средств промышленного шпионажа;

5) противодействие несанкционированному доступу к источникам конфиденциальной информации и другим действиям.

Системы защиты компьютера от чужого вторжения весьма разнообразны и классифицируются, как:

- 1) средства собственной защиты, предусмотренные общим программным обеспечением;
- 2) средства защиты в составе вычислительной системы;
- 3) средства защиты с запросом информации;
- 4) средства активной защиты;
- 5) средства пассивной защиты и другие.

Можно выделить следующие направления использования программ для обеспечения безопасности конфиденциальной информации, в частности такие:

- 1) защита информации от несанкционированной доступа;
- 2) защита информации от копирования;
- 3) защита программ от копирования;
- 4) защита программ от вирусов;
- 5) защита информации от вирусов;
- 6) программная защита каналов связи.

По каждому из указанных направлений имеется достаточно количество качественных, разработанных профессиональными организациями и распространяемых на рынках программных продуктов.

Программные средства защиты имеют следующие разновидности специальных программ:

- 1) идентификации технических средств, файлов и аутентификации пользователей;
- 2) регистрации и контроля работы технических средств и пользователей;
- 3) обслуживания режимов обработки информации ограниченного пользования;
- 4) защиты операционных средств ЭВМ и прикладных программ пользователей;
- 5) уничтожения информации в защитные устройства после использования;
- 6) сигнализирующих нарушения использования ресурсов;
- 7) вспомогательных программ защиты различного назначения.

### 3 Антивирусные программы.

Средства защиты от копирования предотвращают использование ворованных копий программного обеспечения и являются в настоящее время единственным надежным средством – как защищающим авторское право программистов-разработчиков, так и стимулирующих развитие рынка. Под средствами защиты от копирования понимаются средства, обеспечивающие выполнение программой своих функций только при опознании некоторого уникального некопируемого элемента. Таким элементом (называемым ключевым) может быть дискета, определенная часть компьютера или специальное устройство, подключаемое к персональному компьютеру. Защита от копирования реализуется выполнением ряда функций, являющихся общими для всех систем защиты:

1. Идентификация среды, из которой будет запускаться программа (дискета или ПК);
2. Аутентификация среды, из которой запущена программа;
3. Реакция на запуск из несанкционированной среды;
4. Регистрация санкционированного копирования;
5. Противодействие изучению алгоритмов работы системы.

### **Защита программ и данных от компьютерных вирусов**

Вредительские программы и, прежде всего, вирусы представляют очень серьезную опасность при хранении на ПЭВМ конфиденциальной информации. Недооценка этой опасности может

иметь серьезные последствия для информации пользователей. Знание механизмов действия вирусов, методов и средств борьбы с ними позволяет эффективно организовать противодействие вирусам, свести к минимуму вероятность заражения и потерю от их воздействия.

«Компьютерные вирусы» – это небольшие исполняемые или интерпретируемые программы, обладающие свойством распространения и самовоспроизведения (репликации) в компьютерной системе. Вирусы могут выполнять изменение или уничтожение программного обеспечения или данных, хранящихся в ПЭВМ. В процессе распространения вирусы могут себя модифицировать.

### **Классификация компьютерных вирусов**

---

В настоящее время в мире насчитывается более 40 тысяч только зарегистрированных компьютерных вирусов. Так как подавляющее большинство современных вредительских программ обладают способностью к саморазмножению, то часто их относят к компьютерным вирусам. Все компьютерные вирусы могут быть классифицированы по следующим признакам:

- по среде обитания вируса,
- по способу заражения среды обитания,
- по деструктивным возможностям,
- по особенностям алгоритма вируса.

Массовое распространение вирусов, серьезность последствий их воздействия на ресурсы компьютеров вызвали необходимость разработки и использования специальных антивирусных средств и методов их применения. Антивирусные средства применяются для решения следующих задач:

- обнаружение вирусов в КС,
- блокирование работы программ-вирусов,
- устранение последствий воздействия вирусов.

Обнаружение вирусов желательно осуществлять на стадии их внедрения или, по крайней мере, до начала осуществления деструктивных функций вирусов. Необходимо отметить, что не существует антивирусных средств, гарантирующих обнаружение всех возможных вирусов.

При обнаружении вируса необходимо сразу же прекратить работу программы-вируса, чтобы минимизировать ущерб от его воздействия на систему.

Устранение последствий воздействия вирусов ведется в двух направлениях:

- удаление вирусов,
- восстановление (при необходимости) файлов, областей памяти.

Для борьбы с вирусами используются программные и аппаратно-программные средства, которые применяются в определенной последовательности и комбинации, образуя методы борьбы с вирусами.

Самым надежным методом защиты от вирусов является использование аппаратно-программных антивирусных средств. В настоящее время для защиты ПЭВМ используются специальные контроллеры и их программное обеспечение. Контроллер устанавливается в разъем расширения и имеет доступ к общейшине. Это позволяет ему контролировать все обращения к дисковой системе. В программном обеспечении контроллера запоминаются области на дисках, изменение которых в обычных режимах работы не допускается. Таким образом, можно установить защиту на изменение главной загрузочной записи, загрузочных секторов, файлов конфигурации, исполняемых файлов и др.

При выполнении запретных действий любой программой контроллер выдает соответствующее сообщение пользователю и блокирует работу ПЭВМ.

Аппаратно-программные антивирусные средства обладают рядом достоинств перед программными:

- работают постоянно;
- обнаруживают все вирусы, независимо от механизма их действия;
- блокируют неразрешенные действия, являющиеся результатом работы вируса или неквалифицированного пользователя.

Недостаток у этих средств один – зависимость от аппаратных средств ПЭВМ. Изменение последних ведет к необходимости замены контроллера.

Современные программные антивирусные средства могут осуществлять комплексную проверку компьютера на предмет выявления компьютерных вирусов. Для этого используются такие антивирусные программы как – Kaspersky Anti-Virus (AVP), Norton Antivirus, Dr. Web, Symantec Antivirus. Все они имеют антивирусные базы, которые периодически обновляются.

### **Криптографические методы и средства защиты информации**

Криптография как средство защиты (закрытия) информации приобретает все более важное значение в мире коммерческой деятельности.

Криптография имеет достаточно давнюю историю. Вначале она применялась главным образом в области военной и дипломатической связи. Теперь она необходима в производственной и коммерческой деятельности. Если учесть, что сегодня по каналам шифрованной связи только у нас в стране передаются сотни миллионов сообщений, телефонных переговоров, огромные объемы компьютерных и телеметрических данных, и все это не для чужих глаз и ушей, становится ясным: сохранение тайны этой здесь крайне необходимо.

Криптография включает в себя несколько разделов современной математики, а также специальные отрасли физики, радиоэлектроники, связи и некоторых других смежных отраслей. Ее задачей является преобразование математическими методами передаваемого по каналам связи секретного сообщения, телефонного разговора или компьютерных данных таким образом, что они становятся совершенно непонятными для посторонних лиц. То есть криптография должна обеспечить такую защиту секретной (или любой другой) информации, что даже в случае ее перехвата посторонними лицами и обработки любыми способами с использованием самых быстродействующих ЭВМ и последних достижений науки и техники, она не должна быть дешифрована в течение нескольких десятков лет. Для такого преобразования информации используются различные шифровальные средства – такие, как средства шифрования документов, в том числе и портативного исполнения, средства шифрования речи (телефонных и радиопереговоров), телеграфных сообщений и передачи данных.

## **1. 12 Лекция №12 (2 часа)**

**Тема: «Алгоритмизация»**

### **1.12.1 Вопросы лекции:**

1. Понятие алгоритма и алгоритмической системы.
2. Свойства алгоритмов.
3. Формы представления алгоритмов.
4. Основные алгоритмические структуры.

### **1.12.2 Краткое содержание вопросов:**

1. Понятие алгоритма и алгоритмической системы
- Алгоритм** – это точное предписание, определяющее последовательность действий,

обеспечивающее получение требуемого результата из исходных данных.

необходимо сказать ученикам о месте алгоритмов в жизни людей. например, что все люди с детства привыкли следовать тем или иным правилам, выполнять разнообразные инструкции и указания.

Сознание алгоритма доступно всем живым существам, а не только человеку. Другое дело - реализация уже имеющегося алгоритма. Её можно поручить субъекту или объекту, который не обязан вникать в существо дела, а возможно, и не способен его понять. Такой субъект или объект принято называть формальным исполнителем. Пример стиральная машина – автомат. Человек тоже нередко выступает в роли формального исполнителя, Но в первую очередь формальными исполнителями являются различные автоматические устройства, и компьютер в том числе. Исполнитель алгоритма — это некоторая абстрактная или реальная (техническая, биологическая или биотехническая) система, способная выполнить действия, предписываемые алгоритмом.

Для составления программы, предназначеннной для решения на ЭВМ какой-либо задачи, требуется составление алгоритма ее решения.

В информатике понятие алгоритма является одним из фундаментальных понятий. Применительно к ЭВМ алгоритм определяет вычислительный процесс, начинающийся с обработки некоторой совокупности исходных данных и направленный на получение определенных этими данными результатов. Термин «вычислительный процесс» распространяется и на обработку всех видов информации (символьной, графической, звуковой).

## 2. Свойства алгоритма

**Алгоритм должен обладать рядом свойств:**

- ✓ Дискретность – описываемый процесс должен быть разбит на несколько шагов, т.е. представлять собой упорядоченную совокупность четко разделенных друг от друга предписаний, образующих прерывную (дискретную) структуру алгоритма: только выполнив требования одного предписания можно переходить к следующему.
- ✓ Понятность – предписания должны быть написаны на понятном исполнителю языке.
- ✓ Результативность – алгоритм должен приводить к результату после выполнения конечного количества операций.
- ✓ Определенность – состоит в совпадение получаемых результатов независимо от пользователя и применяемых технических средств.
- ✓ Массовость – возможность применения алгоритма к целому классу однотипных задач, отличающихся конкретными значениями исходных данных.

## 3. Формы представления алгоритмов

**Формы записи алгоритмов**

Существует несколько способов записи алгоритмов: словесно-формульный, графически – в виде блок-схем, с помощью таблиц и др. Для обеспечения возможности реализации на ЭВМ алгоритм должен быть записан на языке понятном компьютеру, т.е. на языке программирования.

**Блок-схема** алгоритма представляет собой совокупность геометрических фигур, помеченных порядковыми номерами и соединенных между собой связями (линиями потока), отражающими последовательность выполнения действий.

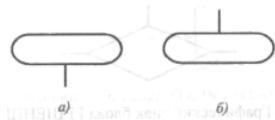
Геометрические фигуры называют блоками. Каждый блок имеет определенное смысловое значение.

## 4. Основные алгоритмические структуры

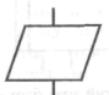
Рассмотрим графические знаки основных блоков, применяемых в блок-схемах алгоритмов.

1. Для обозначения начала или окончания алгоритма используют блоки, именуемые

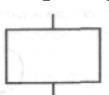
## НАЧАЛО-КОНЕЦ



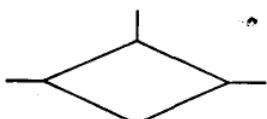
Процесс преобразования данных в форму, пригодную для обработки в ЭВМ или для отображения результатов обработки, изображается с помощью блока ВВОД-ВЫВОД данных.



Блок, в котором происходит обработка данных (выполнение операции или группы операций) (например,  $y = 5$ ,  $y = 2x + 3$  и т. д.), носит название ПРОЦЕСС (или БЛОК ДЕЙСТВИЯ)



Процесс выбора направления выполнения алгоритма в зависимости от некоторых переменных условий изображается с помощью блока РЕШЕНИЕ, имеющего один вход и два выхода.



Если блок-схема алгоритма вычислительного процесса занимает много места и требуется переносить часть блок-схемы на другой лист (или в другое место на том же листе), то используется блок с названием СОЕДИНИТЕЛЬ. Внутри блока указывается номер того блока, к которому (от которого) ведет разорванная линия потока.



## 1. 13 Лекция №13 (6 часов)

Тема: «Основные алгоритмические структуры»

### 1.13.1 Вопросы лекции:

1. Линейные алгоритмы.
2. Разветвляющие алгоритмы.
3. Циклические алгоритмы.

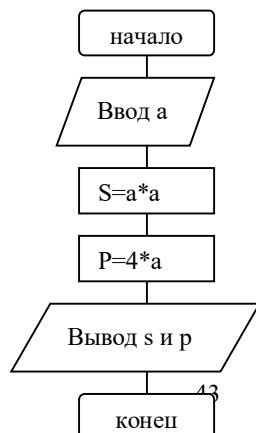
### 1.13.2 Краткое содержание вопросов:

#### Базовые алгоритмические структуры:

Любой вычислительный процесс может быть представлен как комбинация трех основных видов алгоритма: линейные, ветвящиеся, циклические.

##### 1. Линейные алгоритмы.

Линейным принято называть алгоритм, в котором операции выполняются последовательно, в порядке их записи. Каждая операция является самостоятельной и обозначается на схеме отдельным блоком.



Пример: по заданной стороне найти площадь и периметр квадрата.

## 2. Разветвляющийся алгоритм

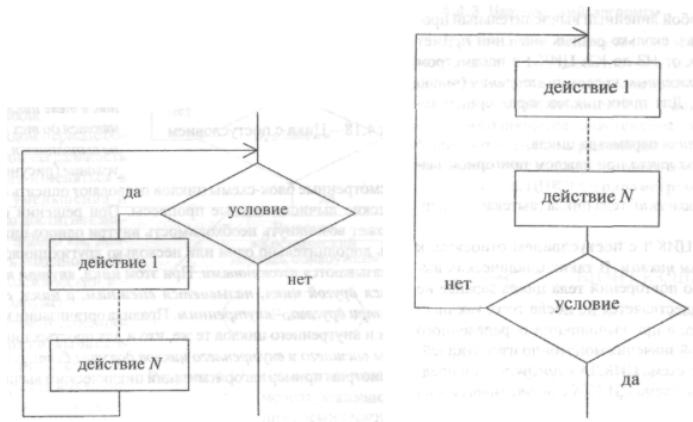
Разветвляющимся называется алгоритм, направление вычислений в котором зависит от результата проверки некоторого условия (условий).

Примеры:

## 3. Циклическим называется

алгоритм, в котором получение результата обеспечивается путем многократного повторения некоторой последовательности действий.

Различают: **ЦИКЛ с предусловием** и **ЦИКЛ с постусловием**. В таких циклических вычислительных процессах число повторений тела цикла заранее не известно. Выход из цикла осуществляется не после того, как цикл повторится заданное число раз, а при выполнении определенного условия, связанного с проверкой значения монотонно изменяющейся в теле цикла величины.



Кратко суть алгоритма ЦИКЛА с предусловием можно изложить следующим образом: пока выполняется условие - повторять действия. В таких циклах возможны ситуации, когда тело цикла не выполняется ни разу (например, если при первой же проверке не выполняется условие, то сразу происходит выход из цикла).

В ЦИКЛЕ с постусловием тело цикла выполняется не менее одного раза. При этом действия, предусмотренные в теле цикла, выполняются до тех пор, пока не выполнится заданное условие

Рассмотренные блок-схемы циклов позволяют описать простые циклические вычислительные процессы. При решении сложных задач может возникнуть необходимость внутри одного цикла организовать дополнительно один или несколько других циклов. Такие циклы называются вложенными. При этом цикл, внутри которого создается другой цикл, называется внешним, а цикл, создаваемый внутри другого, - внутренним

## 1. 14 Лекция №14 (4 часа)

Тема: «Моделирование как метод познания»

### 1.14.1 Вопросы лекции:

1. Понятие модели. Классификация моделей.
2. Методы и технология моделирования.
3. Интеллектуальные системы.
4. Примеры моделей.

### 1.14.2 Краткое содержание вопросов:

## 1. Понятие модели. Классификация моделей.

Сам по себе процесс моделирования в полной мере не формализован, большая роль в этом принадлежит опыту инженера. Но, тем не менее, рассматриваемый в теме процесс создания модели в виде шести этапов может стать основой для начинающих и с накоплением опыта может быть индивидуализирован.

*Математическая модель*, являясь абстрактным образом моделируемого объекта или процесса, не может быть его полным аналогом. Достаточно сходства в тех элементах, которые определяют цель исследования. Для качественной оценки сходства вводится понятие адекватности модели объекту и, в связи с этим, раскрываются понятия изоморфизма и изофункционализма. Формальных приемов, позволяющих автоматически, "бездумно", создавать адекватные математические модели, нет. Окончательное суждение об адекватности модели дает практика, то есть сопоставление модели с действующим объектом. И, тем не менее, усвоение всех последующих тем пособия позволит инженеру справляться с проблемой обеспечения адекватности моделей.

Завершается тема изложением требований к моделям, которые были сформулированы Р. Шеноном на заре компьютерного моделирования тридцать лет назад в книге "Имитационное моделирование систем - искусство и наука". Актуальность этих требований сохраняется и в настоящее время.

Практика свидетельствует: самое лучшее средство для определения свойств объекта - *натурный эксперимент*, т. е. исследование свойств и поведения самого объекта в нужных условиях. Дело в том, что при проектировании невозможно учесть многие факторы, расчет ведется по усредненным справочным данным, используются новые, недостаточно проверенные элементы (прогресс нетерпелив!), меняются условия внешней среды и многое другое. Поэтому натурный эксперимент - необходимое звено исследования. Неточность расчетов компенсируется увеличением объема натурных экспериментов, созданием ряда опытных образцов и "доводкой" изделия до нужного состояния. Так поступали и поступают при создании, например, телевизора или радиостанции нового образца.

Однако во многих случаях натурный эксперимент невозможен.

Например, наиболее полную оценку новому виду вооружения и способам его применения может дать война. Но не будет ли это слишком поздно?

Натурный эксперимент с новой конструкцией самолета может вызвать гибель экипажа.

Натурное исследование нового лекарства опасно для жизни человека.

Натурный эксперимент с элементами космических станций также может вызвать гибель людей.

Время подготовки натурного эксперимента и проведение мероприятий по обеспечению безопасности часто значительно превосходят время самого эксперимента. Многие испытания, близкие к граничным условиям, могут протекать настолько бурно, что возможны аварии и разрушения части или всего объекта.

Из сказанного следует, что натурный эксперимент необходим, но в то же время невозможен либо нецелесообразен.

*Выход* из этого противоречия есть и называется он "*моделирование*".

**Моделирование** - это замещение одного объекта другим с целью получения информации о важнейших свойствах объекта-оригинала.

Отсюда следует.

**Моделирование** - это, во-первых, процесс создания или отыскания в природе объекта, который в некотором смысле может заменить исследуемый *объект*. Этот промежуточный *объект* называется **моделью**. Модель может быть материальным объектом той же или иной природы по отношению к изучаемому объекту (оригиналу). Модель может быть мысленным объектом, воспроизводящим оригинал логическими построениями или математическими формулами и компьютерными программами.

**Моделирование**, во-вторых, это *испытание*, исследование модели. То есть, *моделирование* связано с экспериментом, отличающимся от натурного тем, что в процесс познания включается "промежуточное звено" - модель. Следовательно, **модель** является одновременно *средством эксперимента и объектом эксперимента*, заменяющим изучаемый *объект*.

**Моделирование**, в-третьих, это перенос полученных на модели сведений на оригинал или, иначе, приписывание *свойств модели* оригиналу. Чтобы такой перенос был оправдан, между моделью и оригиналом должно быть сходство, *подобие*.

Подобие может быть физическим, геометрическим, структурным, функциональным и т. д. **Степень подобия** может быть разной - от тождества во всех аспектах до сходства только в основном. Очевидно, модели не должны воспроизводить полностью все стороны изучаемых объектов. Достижение абсолютной одинаковости сводит *моделирование* к натуральному эксперименту, о возможности или целесообразности которого было уже сказано.

Остановимся на основных **целях моделирования**.

**Прогноз** - оценка поведения системы при некотором сочетании ее управляемых и неуправляемых параметров. Прогноз - главная цель *моделирования*.

**Объяснение и лучшее понимание объектов.** Здесь чаще других встречаются задачи оптимизации и анализа чувствительности. *Оптимизация* - это точное определение такого сочетания факторов и их величин, при котором обеспечиваются наилучший показатель качества системы, наилучшее по какому-либо критерию достижение цели моделируемой системой. *Анализ чувствительности* - выявление из большого числа факторов тех, которые в наибольшей степени влияют на функционирование моделируемой системы. Исходными данными при этом являются результаты экспериментов с моделью.

Часто модель создается для применения в качестве *средства обучения*: модели-тренажеры, стенды, учения, деловые игры и т. п.

*Моделирование* как метод познания применялось человечеством - осознанно или интуитивно - всегда. На стенах древних храмов предков южно-американских индейцев обнаружены графические модели мироздания. Учение о моделировании возникло в средние века. Выдающаяся роль в этом принадлежит Леонардо да Винчи (1452-1519).

Гениальный полководец А. В. Суворов перед атакой крепости Измаил тренировал солдат на модели измаильской крепостной стены, построенной специально в тылу.

Наш знаменитый механик-самоучка И. П. Кулибин (1735-1818) создал модель одноарочного деревянного моста через р. Неву, а также ряд металлических моделей мостов. Они были полностью технически обоснованы и получили высокую оценку российскими академиками Л. Эйлером и Д. Бернулли. К сожалению, ни один из этих мостов не был построен.

Огромный вклад в укрепление обороноспособности нашей страны внесли работы по моделированию взрыва - генерал-инженер Н. Л. Кирпичев, моделированию в авиастроении - М. В. Келдыш, С. В. Ильин, А. Н. Туполев и др., моделированию ядерного взрыва - И. В. Курчатов, А.Д. Сахаров, Ю. Б. Харитон и др.

Широко известны работы Н. Н. Моисеева по моделированию систем управления. В частности, для проверки одного нового метода математического моделирования была создана *математическая модель* Синопского сражения - последнего сражения эпохи парусного флота. В 1833 году адмирал П. С. Нахимов разгромил главные силы турецкого флота. *Моделирование* на вычислительной машине показало, что Нахимов действовал практически безошибочно. Он настолько верно расставил свои корабли и нанес первый удар, что единственное спасение турок было отступление. Иного выхода у них не было. Они не отступили и были разгромлены.

Сложность и громоздкость технических объектов, которые могут изучаться методами моделирования, практически неограничены. В последние годы все крупные сооружения исследовались на моделях - плотины, каналы, Братская и Красноярская ГЭС, системы дальних электропередач, образцы военных систем и др. объекты.

Поучительный пример недооценки моделирования - гибель английского броненосца "Кэптен" в 1870 году. В стремлении еще больше увеличить свое тогдашнее морское могущество и подкрепить империалистические устремления в Англии был разработан суперброненосец "Кэптен". В него было вложено все, что нужно для "верховной власти" на море: тяжелая артиллерия во врачающихся башнях, мощная бортовая броня, усиленное парусное оснащение и очень низкими бортами - для меньшей уязвимости от снарядов противника. Консультант инженер Рид построил математическую модель устойчивости "Кэптена" и показал, что даже при незначительном ветре и волнении ему грозит опрокидывание. Но лорды Адмиралтейства настояли на строительстве корабля. На первом же учении после спуска на воду налетевший шквал перевернул броненосец. Погибли 523 моряка. В Лондоне на стене одного из соборов прикреплена бронзовая плита, напоминающая об этом событии и, добавим мы, о тупоумии самоуверенных лордов Британского Адмиралтейства, пренебрегших результатами моделирования.

### **Классификация моделей и моделирования**

Каждая модель создается для конкретной цели и, следовательно, уникальна. Однако наличие общих черт позволяет сгруппировать все их многообразие в отдельные классы, что облегчает их разработку и изучение. В теории рассматривается много признаков классификации, и их количество не установлено. Тем не менее, наиболее актуальны следующие **признаки классификации**:

- характер моделируемой стороны объекта;
- характер процессов, протекающих в объекте;
- способ реализации модели.

#### *Классификация моделей и моделирования по признаку "характер моделируемой стороны объекта"*

В соответствии с этим признаком модели могут быть:

- функциональными (кибернетическими);
- структурными;
- информационными.

**Функциональные модели** отображают только поведение, функцию моделируемого объекта. В этом случае моделируемый объект рассматривается как "черный ящик", имеющий входы и выходы. Физическая сущность объекта, природа протекающих в нем процессов, структура объекта остаются вне внимания исследователя, хотя бы потому, что неизвестны. При **функциональном моделировании** эксперимент состоит в наблюдении за выходом моделируемого объекта при искусственном или естественном изменении входных воздействий. По этим данным и строится модель поведения в виде некоторой математической функции.

Компьютерная шахматная программа - функциональная модель работы человеческого мозга при игре в шахматы.

**Структурное моделирование** - это создание и исследование модели, структура которой (элементы и связи) подобна структуре моделируемого объекта. Как мы выяснили ранее, подобие устанавливается не вообще, а относительно цели исследования. Поэтому она может быть описана на разных уровнях рассмотрения. Наиболее общее описание структуры - это топологическое описание с помощью теории графов.

Учение войск - **структурная модель** вида боевых действий.

#### *Классификация моделей и моделирования по признаку "характер процессов, протекающих в объекте"*

По этому признаку модели могут быть детерминированными или стохастическими, статическими или динамическими, дискретными или непрерывными или дискретно-непрерывными.

**Детерминированные модели** отображают процессы, в которых отсутствуют случайные воздействия.

**Стохастические модели** отображают вероятностные процессы и события.

**Статические модели** служат для описания состояния объекта в какой-либо момент времени.

**Динамические модели** отображают поведение объекта во времени.

**Дискретные модели** отображают поведение систем с дискретными состояниями.

**Непрерывные модели** представляют системы с непрерывными процессами.

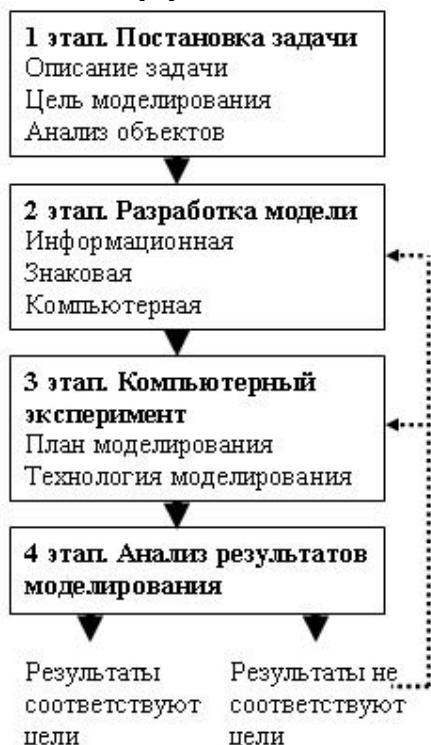
**Дискретно-непрерывные модели** строятся тогда, когда исследователя интересуют оба эти типа процессов.

## 2. Методы и технология моделирования.

Методы моделирования зависят от типов применяемых моделей и разделяются на:

- *Предметное* моделирование, в ходе которого исследование ведется на моделях, воспроизводящей основные геометрические, физические и функциональные характеристики «оригинала»;
- *Физическое* моделирование состоит в замене изучения некоторого объекта или явления экспериментальным исследованием его модели, имеющей ту же физическую природу;
- *Аналоговое* моделирование основано на аналогии (изоморфизме) явлений, имеющих различную физическую природу, но описываемые одинаковыми математическими уравнениями;
- *Знаковое* моделирование, при котором моделями служат знаковые образования: схемы, графики, чертежи, графы, слова и предложения в некотором алфавите;
- *Мысленное* («интуитивное») моделирование – разновидность знакового, при котором реальное построение модели может заменяться мысленно-наглядным представлением знаков или операций над ними.

### Этапы информационного моделирования



#### 1 Этап:

Определение целей моделирования. Они таковы:

- **понимание** – модель нужна для того, чтобы понять, как устроен конкретный объект, какова его структура, основные свойства, законы развития и взаимодействие с окружающим миром;
- **управление** – модель нужна, чтобы научиться управлять объектом (или процессом) и определить наилучшие способы управления при заданных целях и критериях;
- **прогнозирование** – модель нужна для того, чтобы научиться прогнозировать прямые и косвенные последствия реализации заданных способов и форм воздействия на объект.

Ответить на вопросы:

- Что будет, если...? – цель, определение последствий воздействия на объект и принятия правильного решения.
- Как сделать, чтобы...? – цель, создание объекта с заданными свойствами.

## 2 Этап:

**Ранжирование** – разделение входных параметров по степени важности влияния их изменений на выходные.

Выбор наиболее существенной информации при создании модели и ее сложность обусловлены целью моделирования.

## 3 Этап:

Поиск **математического описания**, разработка алгоритмов, выбор программного средства и составление программы для ЭВМ.

**Компьютерная модель** – модель, реализованная средствами программной среды.

## 4 Этап:

Тестирование программы, исправление ошибок, численный эксперимент.

**Тестирование** – процесс проверки правильности модели.

**Тест** – набор исходных данных, для которых заранее известен результат.

**Порядок решения задач на компьютере (пример):**



## 3 Интеллектуальные системы

Идея создания искусственного подобия человеческого разума для решения сложных задач и моделирования мыслительной способности витала в воздухе с древнейших времен (Луллий, Лейбниц, Декарт, Н. Винер).

**Интеллект** – способность человека мыслить, предусматривать возможные последствия своей и чужой деятельности, находить оптимальные варианты, создавать новые знания.

Термин **искусственный интеллект** предложен в 1956 г. в США на семинаре с аналогичным названием.

**Искусственный интеллект** – способность прикладного процесса обнаруживать свойства, ассоциируемые с разумным поведением человека.

**Искусственный интеллект** – раздел информатики, занимающийся вопросами имитации мышления человека с помощью компьютера.

**Интеллектуальная система** – система или устройство с программным обеспечением, имеющие возможность с помощью встроенного процессора настраивать свои параметры в зависимости от состояния внешней среды.



**Знания** в информатике – вид информации, отражающей опыт специалиста (эксперта) в определенной предметной области, его понимание множества текущих ситуаций и способы перехода от одного описания объекта к другому.

Существуют десятки моделей представления знаний для различных предметных областей. Основой любой интеллектуальной системы является *база знаний* – исключительно дорогие информационные массивы.

**Экспертная система (ЭС)** – система искусственного интеллекта, аккумулирующая знания специалистов в определенной узкой предметной области и тиражирующие этот эмпирический опыт для консультаций менее квалифицированных пользователей, т.е. способная предлагать и объяснять пользователю разумные решения. Экспертная система состоит из базы знаний, механизма логического вывода и подсистемы объяснений. В настоящее время все больше возрастает интерес к ЭС среди экономистов, финансистов, преподавателей, инженеров, медиков, психологов, программистов, лингвистов.

## 1. 15 Лекция №15 (2 часа)

Тема: «Программирование»

### 1.15.1 Вопросы лекции:

1. Компьютер как исполнитель алгоритма.
2. Языки программирования.
3. Предпосылки создания языков программирования.
4. Эволюция языков программирования

### 1.15.2 Краткое содержание вопросов: (тезисно изложить основное содержание рассматриваемых вопросов)

1. Компьютер как исполнитель алгоритма

**Алгоритм** – последовательность действий, описывающая процесс преобразования объекта из начального состояния в конечное, записанная с помощью понятных исполнителю команд.

**Исполнителем** алгоритма может быть человек или автоматическое устройство – компьютеры, роботы, станки, спутники, сложная бытовая техника и даже детские игрушки. Каждый алгоритм создается в расчете на вполне конкретного исполнителя.

Компьютер, как исполнитель, любую работу выполняет по программе. Программы пишут люди, а компьютер формально их выполняет.



Разработчики систем искусственного интеллекта пытаются научить машину, подобно человеку, самостоятельно строить программу своих действий, исходя из условия задачи.



Ставится цель превращения компьютера из формального исполнителя в интеллектуального исполнителя.

*Работа обоих исполнителей состоит из четырёх блоков, но формальный исполнитель работает по уже готовой программе, а интеллектуальный – сам составляет программу и получает результат.*

*Информация для компьютера – **данные**, представленные в форме, приемлемой для её передачи и обработки на компьютере.*

*Для работы с данными компьютеру необходимы инструкции (**команды**, правила действия). Команды формируются в перечень команд.*

*Алгоритм – это последовательность действий (команд) для достижения цели.*

В XIX веке английским математиком и инженером Чарльзом Бэббиджем был разработан проект вычислительной машины, которая предназначалась для автоматического проведения длинных цепочек вычислений. Главной особенностью конструкции этой машины является программный принцип работы.

Чарльза Бэббиджа считают изобретателем компьютера – он впервые соединил механический арифмометр с идеей программного управления.

По своему назначению компьютер – это универсальный прибор для работы с информацией.

В основу работы компьютеров положен программный принцип управления. Любой компьютер представляет собой автоматическое устройство, работающее по заложенным в него программам.

Первая вычислительная машина, способная хранить программу в своей памяти, разрабатывалась в 1943—1948 гг. в США под руководством Джона Мочли и Преснера Эккера.

В 1945 г. к работе был привлечен знаменитый математик Джон фон Нейман, который сформулировал общие принципы функционирования универсальных вычислительных устройств.

Первый компьютер, в котором были полностью реализованы эти принципы, был построен в 1949 г. английским исследователем Морисом Уилкском. Изменяется элементная база, компьютеры становятся все более и более мощными, но до сих пор большинство из них соответствуют тем принципам, которые изложил в своем докладе в 1945 г. Джон фон Нейман.

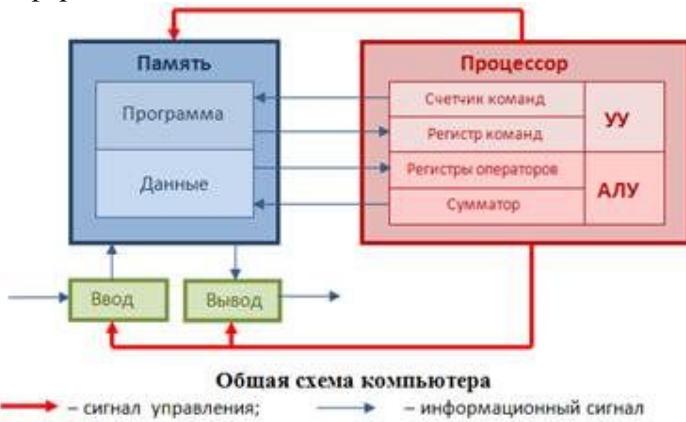
Согласно фон Нейману, ЭВМ состоит из следующих основных блоков:

- арифметико-логическое устройство, выполняющее арифметические и логические операции;
- устройство управления, которое организует процесс выполнения программ;
- запоминающее устройство, или память, для хранения программ и данных;
- внешние устройства для ввода-вывода информации.

В современных компьютерах это:

- память (запоминающее устройство — ЗУ), состоящая из перенумерованных ячеек;
- процессор, включающий в себя устройство управления (УУ) и арифметико-логическое устройство (АЛУ);
- устройство ввода;
- устройство вывода.

Эти устройства соединены между собой каналами связи, по которым передается информация.



**Функции памяти:** - прием информации из других устройств; - запоминание информации; - выдача информации по запросу в другие устройства машины.

**Функции процессора:** - обработка данных по заданной программе путем выполнения арифметических и логических операций; - программное управление работой устройств компьютера.

Одна часть процессора, которая выполняет команды, называется *арифметико-логическим устройством*, а другая его часть, выполняющая функции управления устройствами, — *устройством управления*. Обычно эти устройства выделяются чисто условно, конструктивно они не разделены.

В составе процессора имеется ряд специализированных дополнительных ячеек памяти, называемых *регистрами*. Регистр выполняет функцию кратковременного хранения числа или команды. Основным элементом регистра является электронная схема, называемая *триггером*.

Регистр представляет собой совокупность триггеров, связанных друг с другом определенным образом общей системой управления.

Существует несколько типов регистров, отличающихся видом выполняемых операций. Некоторые важные регистры имеют свои названия, например:

- *сумматор* — регистр АЛУ, участвующий в выполнении каждой операции;
- *счетчик команд* — регистр УУ, содержимое которого соответствует адресу очередной выполняемой команды. Он служит для автоматической выборки программы из последовательных ячеек памяти;
- *регистр команд* — регистр УУ для хранения кода команды на период времени, необходимый для ее выполнения. Часть его разрядов используется для хранения кода операции, остальные — для хранения кодов адресов operandов.

Компьютер является универсальным исполнителем по обработке информации. Значит, для него, как для любого исполнителя, существует определённая система команд (СКИ). Такая система команд для компьютера называется **языком машинных команд (ЯМК)**

**Программа для компьютера** – это алгоритм, разработанный на ЯМК. Или, **Программа управления компьютером** – это последовательность команд ЯМК, где каждая команда – директива для процессора на выполнение определённого действия.

Рассмотрим этапы выполнения программы.



Согласно принципам Джона фон Неймана, программа во время её исполнения и данные, которые она обрабатывает, находятся в оперативной памяти (принцип хранимой в памяти программы). Процессор исполняет программу начиная с первой команды и заканчивая последней.

Для компьютера вся информация должна быть представлена в двоичных кодах, т.е. необходим способ перевода. Такой способ перевода называется **трансляцией**, а выполняет это транслятор.

**Вывод:** Устройством, которое обрабатывает информацию в компьютере, является процессор, следовательно, алгоритм должен использовать систему команд процессора, или другими словами записан на машинном языке, представляющем собой последовательности нулей и единиц

Сначала программисты, работавшие на компьютерах первого поколения (50-е – 60-е г.г.), составляли программы на ЯМК (в двоичных кодах), но это довольно сложная работа, поэтому для облегчения программирования были созданы языки программирования высокого уровня (ЯПВУ) - это искусственно созданные языки с несколькими десятками слов (операторов) и строгими правилами синтаксиса. Составление программ на ЯПВУ намного проще. Примеры ЯПВУ: Фортран, Паскаль, Бейсик, Си и др.

Для того чтобы процессор мог выполнить программу, написанную на языке программирования, она и данные с которыми она работает должны быть загружены в оперативную память. Программа написана и загружена в оперативную память и для того чтобы процессор ее выполнил в оперативной памяти, должна быть еще и программа переводчик (транслятор), который переводит программу с языка высокого уровня на язык машинных команд

Таким образом, цепочка событий от составления программы на ЯПВУ до получения результатов решения задачи выглядит так



Человек всегда должен понимать ограниченность возможность компьютера как исполнителя, необходимость предусмотреть все тонкости команд, поручаемых компьютеру. Человек

разрабатывает алгоритм, записывает его на ЯПВУ и анализирует результаты выполнения программы.

Компьютер является формальным исполнителем программ.

Итак, компьютер не может обойтись без программы и исходных данных, подготовить их может только человек.

Поэтому можно говорить, **что решение задач компьютером** - это формальное исполнение алгоритма (программы), а компьютер является формальным исполнителем.

*Компьютер может быть использован для решения самых разнообразных задач, поэтому, исходя из условия задачи, человек решает, каким программным средством пользоваться. Если в состав ПО входят программы, подходящие для решения задач человека, то удобнее ими воспользоваться (текстовый редактор, электронные таблицы, базы данных, презентации).*

*В случае, если нельзя воспользоваться готовым программным обеспечением, приходится прибегать к программированию (операционные системы, доработка ОС, трансляторы, драйверы, архиваторы, антивирусы).*

## 2. Наименование вопроса № 2

**Язык программирования** — формальная языковая система, предназначенная для записи компьютерных программ. Язык программирования определяет набор лексических, синтаксических и семантических правил, задающих внешний вид программы и действия, которые выполнит исполнитель (компьютер) под её управлением.

Со времени создания первых программируемых машин человечество придумало более двух с половиной тысяч языков программирования (включая абстрактные и нестандартные языки). Каждый год их число увеличивается. Некоторыми языками умеет пользоваться только небольшое число их собственных разработчиков, другие становятся известны миллионам людей. Профессиональные программисты иногда применяют в своей работе более десятка разнообразных языков программирования.

Создатели языков по-разному толкуют понятие *язык программирования*. К наиболее распространённым утверждениям, признаваемым большинством разработчиков, относятся следующие:

- *Функция*: язык программирования предназначен для написания компьютерных программ, которые применяются для передачи компьютеру инструкций по выполнению того или иного вычислительного процесса организации управления отдельными устройствами.
- *Задача*: язык программирования отличается от естественных языков тем, что предназначен для передачи команд данных от человека к компьютеру, в то время как естественные языки используются для общения людей между собой. Можно обобщить определение «языков программирования» — это способ передачи команд, приказов, чёткого руководства к действию; тогда как человеческие языки служат также для обмена информацией.
- *Исполнение*: язык программирования может использовать специальные конструкции для определения и манипулирования структурами данных и управления процессом вычислений.

## 1. 16 Лекция №16 (2 часа)

**Тема:** «Классификация языков программирования»

### 1.16.1 Вопросы лекции:

1. Коды, ассемблеры, алгоритмические языки программирования.
2. Понятие языка высокого уровня.

3. Структурное программирование.

4. Объектно-ориентированное программирование.

**1.16.2 Краткое содержание вопросов:** (тезисно изложить основное содержание рассматриваемых вопросов)

### 1. Коды, ассемблеры, алгоритмические языки программирования

Разные типы процессоров имеют разные наборы команд. Если язык программирования ориентирован на конкретный тип процессора и учитывает его особенности, т.е. является машинно-зависимым, он называется *языком программирования низкого уровня*. В данном случае «низкий уровень» не значит «плохой». Имеется в виду, что операторы языка близки к машинному коду и ориентированы на конкретные команды процессора.

*Языками низкого уровня* являются все языки ассемблеров, которые представляет каждую команду машинного кода, не в виде чисел, а с помощью условных символьных обозначений, называемых *мнемониками*. Однозначное преобразование одной машинной инструкции в одну команду ассемблера называется *транслитерацией*. Так как наборы инструкций для каждой модели процессора отличаются, конкретной компьютерной архитектуре соответствует свой язык ассемблера, и написанная на нем программа может быть использована только в этой среде.

С помощью языков низкого уровня создаются очень эффективные и компактные программы, так как разработчик получает доступ ко всем возможностям процессора, но при этом:

- требуется очень хорошо понимать устройство компьютера,
- затрудняется отладка больших приложений,
- результирующая программа не может быть перенесена на компьютер с другим типом процессора.

Подобные языки обычно применяют для написания небольших системных приложений, драйверов устройств, модулейстыковки с нестандартным оборудованием, когда важнейшими требованиями становятся компактность, быстродействие и возможность прямого доступа к аппаратным ресурсам.

*Языки программирования высокого уровня* значительно ближе и понятнее человеку, нежели компьютеру. Особенности конкретных компьютерных архитектур в них не учитываются, поэтому создаваемые программы на уровне исходных текстов легко переносимы на другие платформы, для которых создан транслятор этого языка. Разрабатывать программы на языках высокого уровня с помощью понятных и мощных команд значительно проще, а ошибок при создании программ допускается гораздо меньше.

Старейшим языком высокого уровня, не утратившим своей актуальности и сегодня, является **ФОРТРАН** (FORTRAN), хотя от исходной версии в нем мало что сохранилось. Название ФОРТРАН составлено из словосочетания «FORmula TRANslation» (трансляция формул), поэтому его принято писать прописными буквами (это касается многих языков программирования). ФОРТРАН разработан в IBM в середине 1950-х для компьютеров серии 704 и долгое время интенсивно использовался в научном и инженерном

программировании. Он особенно удобен для математических расчетов благодаря обширнейшей поддержке операций с плавающей точкой, включая возможность работы с комплексными числами. Для него было создано огромное количество библиотек, начиная от статистических комплексов и, кончая пакетами управления спутниками. Все это позволяет активно использовать его и сейчас.

**АЛГОЛ** (ALGOL). Его имя – тоже сокращение, составленное из «ALGOrithmic Language» (алгоритмический язык). Он во многих отношениях – прямой предок многих распространенных языков, созданных позднее. Даже в наши дни иногда приходится слышать о «языках программирования типа АЛГОЛа». Первую версию – АЛГОЛ 58 – разработал в 1957-58 гг. международный комитет программистов. Два года спустя был выпущен усовершенствованный вариант – АЛГОЛ 60, а затем и АЛГОЛ 68.

Первыми программистами были в основном ученые и инженеры, которые умели формулировать свои задачи на языке математики, положенном в основу ФОРТРАНа и АЛГОЛА. Однако на протяжении всей истории языков программирования неоднократно предпринимались попытки разработать язык, который могли бы использовать и люди, не столь близко знакомые с математикой.

Одним из первых языков, специально предназначенных для бизнеса, был **КОБОЛ** (COBOL), созданный в конце 1950-х комитетом из представителей промышленности и Министерства обороны США. КОБОЛ широко применяется и сегодня. Его название расшифровывается как «COmmon Business Oriented Language» (язык, ориентированный на общие коммерческие задачи). Одно из основных требований, предъявлявшихся к КОБОЛу, заключалось в том, чтобы менеджеры, сами не занимавшиеся программированием, могли хотя бы читать программы и убеждаться, что они делают именно то, что должны делать. В КОБОЛе имеются обширные возможности по чтению записей (records) и созданию отчетов (reports).

В середине 1960-х в IBM разработали для компьютеров System/360 язык **PL/I** (Programming Language I, язык программирования №1). Предполагалось, что в PL/I будут объединены модульная структура программ на АЛГОЛе, обширный математический аппарат ФОРТРАНа и средства КОБОЛА для работы с записями. Но уровня популярности ФОРТРАНа или КОБОЛА этот язык так и не достиг.

Язык **БЕЙСИК** (BASIC, Beginner™s All-purpose Symbolic Instruction Code, универсальный символьный программный код для начинающих) разработан в 1964 г. Джоном Кемени (John Kemeny) и Томасом Курцем (Thomas Kurtz) из Дартмутского университета для студентов, которые не были ни математиками, ни инженерами, поэтому их не стоило заставлять возиться с перфокартами или сложными языками программирования. Вместо этого студент, сидя перед терминалом, набирал простую программу прямо на экране. Если строка начиналась с номера, она считалась строкой программы на БЕЙСИКе. Стока без номера считалась командой для системы. В отличие от АЛГОЛа в БЕЙСИКе программист не должен был указывать тип переменной. Большинство переменных по умолчанию считались вещественными. Во многих последующих реализациях БЕЙСИКА использовались не компиляторы, а интерпретаторы. На домашних компьютерах дебют БЕЙСИКА состоялся в 1975 г., когда Билл Гейтс (Bill Gates) и Пол Аллен (Paul Allen) написали интерпретатор БЕЙСИКА для компьютера «Альтаир 8800». Этот интерпретатор стал первым продуктом основанной ими корпорации Microsoft.

Язык программирования **Паскаль** (Pascal) унаследовал структуру АЛГОЛА и средства КОБОЛА для работы с записями. Он разработан в конце 1960-х швейцарским профессором информатики Николасом Виртом (Niklaus Wirth). Среди программистов компьютеров IBM PC Паскаль был очень популярен, правда, только в одной специфической реализации – Turbo Pascal фирмы Borland. Эта программа, написанная Андерсом Хейлсбергом (Anders Hejlsberg) из Дании, поступила в продажу в 1983 г. Она представляла собой интегрированную среду разработки (Integrated Development Environment, IDE) – текстовый редактор и компилятор были объединены в единую программу, что существенно облегчало разработку кодов.

На Паскале частично основан язык программирования **Ада**, разработанный для Минобороны США. Он назван в честь Августы Ады Байрон, первой женщины программиста.

**С (Си).** Этот чрезвычайно популярный язык был создан в 1969-1973 гг. в основном усилиями Денниса Ритчи (Dennis Ritchie) из Bell Telephone Laboratories. Его предшественником был язык B, который в свою очередь был упрощенным вариантом BCPL (Basic CPL), основанного на CPL (Combined Programming Language, комбинированный язык программирования). Первоначально планировался для замены ассемблера, чтобы иметь возможность создавать столь же эффективные и компактные программы, и в то же время не зависеть от конкретного типа процессор. До его появления ОС, как правило, писались на языке ассемблера для конкретного процессора. В 1973 г. ОС UNIX была написана (точнее, переписана) на С, и с тех пор язык и система идут по жизни рука об руку.

Все перечисленные языки относятся к алгоритмическим процедурным языкам, предназначенным для компьютеров с архитектурой Неймана.

## 2. Понятие языка высокого уровня.

### Высокоуровневый язык программирования —

язык программирования, разработанный для быстроты и удобства использования программистом. Основная черта высокоуровневых языков — это абстракция, то есть введение смысловых конструкций, краткоописывающих такие структуры данных и операции над ними, описания которых на машинном коде (или другом низкоуровневом языке программирования) очень длинны и сложны для понимания.

Так, высокоуровневые языки стремятся не только облегчить решение сложных программных задач, но и упростить портирование программного обеспечения. Использование разнообразных трансляторов интерпретаторов обеспечивает связь программ, написанных при помощи языков высокого уровня, различными операционными системами и оборудованием, в то время как их исходный код остаётся, виде але, неизменным.

Такого рода оторванность высокоуровневых языков от аппаратной реализации компьютера и омимомножества плюсов имеет и минусы. В частности, она не позволяет создавать простые и точные инструкции используемому оборудованию. Программы, написанные на языках высокого уровня, проще для понимания программистом, но менее эффективны, чем их аналоги, создаваемые при помощи низкоуровневых языков. Одним из следствий этого стало добавление поддержки того или иного языка низкого уровня (языка ассемблера) в ряд современных профессиональных высокоуровневых языков программирования.

Примеры: C++, Visual Basic, Python, Perl, Delphi (Pascal), строковых типов, объектов, операций файлового ввода-вывода и т. п.

Первым языком программирования высокого уровня считается компьютерный язык Plankalkyl разработанный немецким инженером Конрадом Цузе еще в период 1942 —

[1946](#) гг. Однако, широкое применение высокоуровневых языков началось с возникновением [Фортрана](#) и созданием [компилятора для этого языка](#) ([1957](#)).

### 3 Структурное программирование.

*Процедурное* или *императивное* (от лат. *imperativus* – повелительный) программирование есть отражение фон Неймановской архитектуры компьютера. Программа на процедурном языке состоит из последовательности команд, определяющих процедуру решения задачи. Основным является *оператор присваивания*, предназначенный для определения и изменения содержимого памяти компьютера. Концепция памяти как места хранения данных, значения которых можно изменять операторами программы, является фундаментальным в императивном программировании.

Выполнение программы сводится к последовательному выполнению операторов с целью преобразования исходного состояния памяти, т.е. программа последовательно обновляет содержимое памяти, изменяя его от исходного состояния до результирующего.

Одним из первых *процедурных языков программирования* высокого уровня стал *Фортран* (**FOR**mul**A****TRAN**slation), созданный в начале 50-х гг. в США фирмой IBM. Первая публикация о нем появилась в 1954 г. Основное назначение языка – программирование научно-технических задач. Объектами языка являются целые и вещественные числа и числовые переменные. Выражения в нем формируются с помощью четырех арифметических действий: возведения в степень, логических операций И, ИЛИ, НЕ, операций отношения и круглых скобок. Основные операторы Фортрана – ввод, вывод, присваивание, условный и безусловный переход, цикл, вызов подпрограмм. Долгие годы он был одним из самых распространенных языков в мире. За это время накоплена огромная библиотека программ, написанных на Фортране. И сейчас ведутся работы над очередным стандартом Фортрана. Многие средства Фортрана использованы в языках PL-1 и Бейсик.

*Кобол* (**CO**mmon **B**usiness **O**riented **L**anguage – обще принятый деловой язык) – язык программирования, ориентированный на решение задач обработки данных. Широко используется для решения учетно-экономических и управлеченческих задач. Разработан в США в 1958–1960 гг. Программа на Коболе имеет вид ряда предложений на английском языке и напоминает обычный текст. Группы последовательно записанных операторов объединяются в предложения, предложения – в параграфы, параграфы – в секции. Программист присваивает параграфам и секциям имена (метки), что облегчает непосредственное обращение к нужному участку программы. В СССР был принят русский вариант языка. В Коболе были реализованы мощные средства работы с большими объемами данных, хранящимися на различных внешних носителях. На этом языке создано много приложений, некоторые из них активно эксплуатируются и сейчас. Достаточно сказать, что одной из высокооплачиваемых категорий граждан в США являются программисты на Коболе.

*Алгол* (**AZG**Orithmic **L**anguage) разработан группой зарубежных специалистов в 1960 г., явился результатом международного сотрудничества конца 50-х гг. (Алгол-60). Алгол предназначался для записи алгоритмов, построенных в виде последовательности процедур, применяемых при решении поставленных задач. Специалисты-практики воспринимали этот язык неоднозначно, но тем не менее, он как признанный международный язык сыграл большую роль в становлении основных понятий программирования и для обучения программистов. В нем впервые введены понятия «блочная структура программы», «динамическое распределение памяти». Внутри блока в Алголе можно вводить локальные обозначения, которые не зависят от остальной части программы. Несмотря на свое

интернациональное происхождение, Алгол-60 получил меньшее распространение, чем Фортран. Например, не на всех зарубежных ЭВМ имелись трансляторы с Алгола-60. В 1968 г. в результате дальнейшего развития и усовершенствования Алгола-60 была создана версия Алгол-68. Это многоцелевой универсальный расширенный язык программирования. Последнее свойство позволяло с помощью одной и той же программы транслятора осуществлять трансляцию с различных расширенных версий языка без дополнительных затрат на приспособление этого языка к различным категориям пользователей, на получение проблемно-ориентированных диалектов языка. По своим возможностям Алгол-68 и сегодня опережает многие языки программирования, однако из-за отсутствия эффективных компьютеров для него не удалось своевременно создать хорошие компиляторы. В нашей стране в те годы под руководством академика Андрея Петровича Ершова был создан транслятор Альфа, который представлял достаточно удачную русифицированную версию Алгола.

В середине 60-х гг. сотрудники математического факультета Дартмутского колледжа Томас Курц и Джон Кемени создали специализированный язык программирования, который состоял из простых английских слов. Новый язык называли *универсальным символическим кодом для начинающих (BeginnersAll-purposeSymbolicInstructionCode)* или сокращенно *BASIC (Бейсик)*. 1964 г. считают годом рождения этого языка. Он получил самое широкое распространение при работе на персональных компьютерах в режиме интерактивного диалога. Популярность Бейсика объясняется как простотой его освоения, так и наличием достаточно мощных универсальных средств, пригодных для решения научных, технических и экономических задач, а также задач бытового характера, игровых и т.д. Согласно концепциям, заложенным в Бейсике, в нем широко распространены различные правила умолчания, что считается плохим тоном в большинстве языков программирования подобного типа. Возникло множество версий языка, зачастую мало совместимых друг с другом. Однако, зная одну из версий, можно без особого труда освоить любую другую. Бейсик активно поглощает многие концепции и новинки из других языков. Первоначально интерактивный режим осуществлялся с использованием интерпретатора, в настоящее время для этого языка имеются также и компиляторы.

В начале 60-х гг. каждый из существующих языков программирования был ориентирован на разные классы задач, но в той или иной мере привязан к конкретной архитектуре ЭВМ. Были предприняты попытки преодолеть этот недостаток путем создания универсального языка программирования. *ПЛ/1 (PL/1 – ProgrammingLanguageOne)* — первый многоцелевой универсальный язык, разработан в США фирмой IBM в 1963—1966 гг. Это один из наиболее распространенных универсальных языков, он хорошо приспособлен для решения задач в области вычислительной техники: исследования и планирования вычислительных процессов, моделирования, решения логических задач и исследования логических схем, разработки систем математического обеспечения. При разработке PL/1 были широко использованы основные понятия и средства языков Фортран, Алгол-60, Кобол. PL/1 — богатый и гибкий язык, дает возможность производить вставки, исправлять текст программы в процессе ее отладки. Язык получил широкое распространение, трансляторы с него имеются для многих типов компьютеров. Компания IBM сегодня продолжает поддерживать этот язык.

*Паскаль (Pascal)* является одним из наиболее популярных процедурных языков программирования, особенно для персональных компьютеров. Созданный как учебный язык программирования в 1968—1971 гг. Никлаусом Виртом в Высшей технической школе в Цюрихе (Швейцария), он был назван в честь французского математика и философа Блеза Паскаля (1623—1662). Целью работы Н. Вирта было создание языка, который

- строился бы на небольшом количестве базовых понятий;
- имел простой синтаксис;
- допускал перевод программ в машинный код простым компилятором.

Лингвистическая концепция Паскаля пропагандирует системный подход, выражющийся, в частности, в расчленении крупных задач на меньшие по сложности и размеру, легко поддающиеся решению. К основным принципам Паскаля следует отнести:

- **Структурное программирование.** Суть его заключается в оформлении последовательности команд как замкнутых функций или процедур и в объединении данных, связанных по смыслу, в сложные структуры данных. Благодаря этому повышается наглядность текста и упрощается его отладка.
- **Программирование сверху вниз**, когда задача разбивается на простые, после чего каждая решается в отдельности. Затем компонуются результаты проектирования простых задач, и поставленная задача решается сверху вниз в целом.

В основу разработки языка Паскаль был положен Алгол-60, но в нем ужесточен ряд требований к структуре программы и имеются возможности, позволяющие успешно применять его для создания крупных проектов, например, программ-трансляторов. Паскаль реализован для всех типов компьютеров, в настоящее время используется во многих учебных заведениях для обучения программированию, а также для создания больших реальных проектов.

Период с конца 60-х до начала 80-х гг. характеризуется бурным ростом числа различных языков программирования, сопровождавшим, как это ни парадоксально, кризис программного обеспечения. Этот кризис особенно остро переживало военное ведомство США. В январе 1975 г. Пентагон решил навести порядок среди бесчисленного множества трансляторов и создал комитет для разработки одного универсального языка. На конкурсной основе комитет рассмотрел сотни проектов и выяснил, что ни один из существующих языков не может удовлетворить их требованиям, для окончательного рассмотрения было оставлено два проекта. В мае 1979 г. был объявлен победитель – группа ученых во главе с Жаном Ихбиа. Победивший язык назвали АДА, в честь Ады Лавлейс, дочери великого поэта Байрона. Она в юности была увлечена идеями Чарльза Бэббиджа и помогала ему составлять описание машины, а в начале 40-х гг. XIX в. разработала первую в мире программу для вычислительной машины. Язык АДА – прямой наследник Паскаля. Он предназначен для создания и длительного сопровождения больших программных систем, управления процессами в реальном масштабе времени. В языке четко выражена модульность его конструкций, причем обеспечивается удобство организации разнообразных связей между модулями. Важным его достоинством является возможность параллельного программирования ветвей программы, которые затем могут реализоваться на многопроцессорных компьютерах. Язык АДА сложен для изучения.

Язык программирования С (*Cи*) был разработан в лаборатории Bell для реализации операционной системы UNIX в начале 70-х гг. и не рассматривался как массовый. Он планировался для замены Ассемблера, чтобы иметь возможность создавать столь же эффективные и компактные программы, и в то же время не зависеть от конкретного типа процессора. В Си сочетаются достоинства современных высоковысоких языков в части управляющих конструкций и структур данных с возможностями прямого доступа к аппаратным средствам компьютера. Синтаксис языка Си обеспечивает краткость программы, его компиляторы генерируют эффективный объектный код. Одна из наиболее существенных особенностей Си состоит в том, что различия между выражениями и операторами

нивелируются, это приближает его к функциональным языкам. Например, выражение может обладать побочным эффектом присваивания, а также может использоваться в качестве оператора. Нет четкого различия между процедурами и функциями, более того, понятие процедуры вообще не вводится. Синтаксис языка затрудняет программирование и восприятие составленных программ. Отсутствует строгая типизация данных, что предоставляет дополнительные возможности программисту, но не способствует созданию надежных программ. Язык Си приобрел большую популярность среди системных и прикладных программистов. В настоящее время этот язык реализован для большинства компьютерных платформ.

#### 4. Объектно-ориентированное программирование

Пионером данного направления явился язык *Смояток* (*Smalltalk*), первоначально предназначенный для реализаций функций машинной графики. Работа над языком началась в 1970 г. в исследовательской лаборатории XEROX (США), а закончилась в 1980 г. окончательным вариантом интерпретатора *Smalltalk-80*. Данный язык оригинален тем, что его синтаксис очень компактен и базируется исключительно на понятии объекта. В нем отсутствуют операторы или данные, все, что входит в Смояток, является объектами, а объекты общаются друг с другом исключительно с помощью сообщений. В настоящее время версия *VisualAge for Smalltalk* активно развивается компанией IBM.

Основой объектно-ориентированного программирования (ООП) является понятие **объект**. Его сущность выражается формулой «объект = данные + процедуры». Каждый объект содержит некоторую структуру данных и доступные только ему процедуры (методы) обработки этих данных. Используя эту методологию, можно создать свой собственный абстрактный тип и отобразить проблемную область в эту созданную абстракцию вместо традиционного ее отображения в предопределенные управляющие структуры и структуры данных языка программирования. Объединение данных и свойственных им процедур обработки в одном объекте называется *инкапсуляцией* и присуще ООП.

Другим фундаментальным понятием ООП является **класс**. Класс – это шаблон, на основе которого может быть создан конкретный программный объект, он определяет свойства и методы объекта, принадлежащего этому классу, соответственно, любой созданный объект становится *экземпляром класса*. Класс обеспечивает скрытие данных, их гарантированную инициализацию, неявное преобразование типов для типов, определенных пользователем, контролируемое пользователем управление памятью и механизмы перегрузки операций.

ООП является более естественным, так как предоставляет возможность выбрать имеющиеся или создать новые объекты и организовать взаимодействия между ними. Следовательно, объектно-ориентированные языки по сравнению с процедурными являются языками более высокого уровня.

## 1. 17 Лекция №17 (4 часа)

Тема: «Основные понятия алгоритмического языка Pascal»

### 1.17.1 Вопросы лекции:

1. История развития алгоритмического языка Pascal.
2. Основная структура программы языка Pascal.
3. Основные операторы языка Pascal.

### 1.17.2 Краткое содержание вопросов:

1. История развития алгоритмического языка Pascal.

Алгоритмический язык Паскаль был разработан в 1971г. швейцарским математиком Н. Виртом. Язык получил название в честь французского математика и философа Блеза Паскаля (1623–1662). С момента создания и до сегодняшних дней язык играет особую роль в его изучении и в практическом программировании. Автор реализовал в языке принцип структурного программирования.

Паскаль стал первым языком, с которым знакомится большинство будущих программистов. Существует много версий языка Паскаль.

В 80-е годы на основе Паскаля был разработан Turbo Pascal. Turbo – это торговая марка разработчика фирмы Borland.

2. Основная структура программы языка Pascal.

Под конструкцией любого алгоритмического языка высокого уровня понимают все его составляющие: алфавит, данные, стандартные функции и процедуры, операторы.

#### *Алфавит*

- 1) латинский шрифт;
- 2) русский шрифт;
- 3) цифры (0 ÷ 9);
- 4) символы:
  - а) знаки арифметических операций (+ – \* /), нет возведения в степень;
  - б) знаки логических отношений (<, >, <= вместо ≤, >= вместо ≥, <> вместо ≠);
  - в) разделители (, . ; :)
  - г) прочие символы.

#### *Данные и типы данных*

Данные могут быть разделены на:

- 1) Константы – const.
- 2) Переменные – var.

**Константам и переменным даётся имя, которое называется идентификатором.** С другой стороны в зависимости от вида данных (число, текст, символ и т.д.) в Паскале имеет значение тип данных.

#### **Понятие типа – одно из фундаментальных понятий Turbo Pascal.**

Паскаль – это типизированный язык, который характеризуется разветвленной структурой типов данных, построен на основе строгого соблюдения типов. Язык Turbo Pascal предоставляет большие возможности создания сложных типов, однако все они строятся на основе элементарных (стандартных) типов.

Для начала можно ограничиться стандартными типами данных (4 типа). Соответственно можно выделить следующие данные: числовые, символьные, логические. Числовые данные подразделяются на целые и вещественные:

1. INTEGER – целочисленные данные, во внутреннем представлении занимают два байта; диапазон возможных значений от -32768 до +32767.
2. REAL – вещественные данные, занимают 6 байт; диапазон возможных значений модуля от 2.9E-39 до 1.7E+38; точность представления данных – 11...15 значащих цифр.

Вещественные данные в паскале могут записываться в двух форматах:

а) Формат с фиксированной точкой.

Пример: Число 34,5 в паскале записывается 34.5.

б) Формат с плавающей запятой.

Пример: Число 34,5 в паскале можно записать 0.345E2 или 3.45E1.

Где символ **E** называется десятичной экспонентой, означает число 10, а после записывается степень этого числа.

3. CHAR – символьные данные, занимает 1 байт.

4. BOOLEAN – логический тип, занимает 1 байт и имеет два значения: FALSE (ложь) и TRUE (истина).

### *Стандартные функции*

Стандартные функции подразделяются на числовые, символьные, строковые и т.д.

Числовые стандартные функции представлены в таблице 14.

После имени стандартной функции в скобках записывается аргумент, который может быть:

- константой: например  $\cos(1.3)$ ;
- переменной: например  $\cos(x)$ ;
- арифметическим выражением: например  $\cos(x+y)$ ;
- стандартной функцией: например  $\cos(\ln(x))$ .

Таблица 14

№	Запись на Паскале	Запись в математике	Тип результата	Примечание
1	$\sin(x)$	$\sin x$	вещественный	$x$ -угол в радианах
2	$\cos(x)$	$\cos x$	вещественный	$x$ -угол в радианах
3	$\arctan(x)$	$\arctg x$	вещественный	$x$ - в радианах
4	$\exp(x)$	$e^x$	вещественный	$e=2,7182\dots$ -основание натурального логарифма
5	$\ln(x)$	$\ln x$	вещественный	
6	$\text{sqr}(x)$	$x^2$	зависит от типа $x$	Квадрат числа $x$
7	$\text{sqrt}(x)$		вещественный	Корень квадратный
8	$\text{abs}(x)$	$ x $	вещественный	Модуль числа $x$
9	$\text{trunc}(x)$		целый	Целая часть ( $x$ )
10	$\text{int}(x)$		вещественный	Целая часть ( $x$ )
11	$\text{frac}(x)$		вещественный	Дробная часть ( $x$ )
12	$\text{round}(x)$		целый	Округление ( $x$ )
13	$\text{odd}(x)$		целый	Если $x$ -нечётное, то функция $\text{true}$
14	$\pi$		вещественный	$\pi=3,1415\dots$

Аргумент тригонометрической функции должен быть задан в радианах. Если он задан в градусах, то его следует перевести в радианы по формуле: ;

Логарифмические функции:

Обратные тригонометрические функции:

; ; ;

Гиперболические функции:

; ; ; ;

Возведение в степень:

;

Тригонометрические функции:

- $\text{tg } x = \sin x / \cos x;$

- $\operatorname{ctg} x = \cos x / \sin x$ .

### **Арифметические, логические, символьные выражения**

#### *a) Арифметические выражения*

Пример арифметического выражения.

В Турбо Паскале есть все 4 арифметические операции над числовыми переменными:

- + сложение;
- - вычитание;
- \* умножение;
- / деление вещественное;

Для данных типа INTEGER в Турбо Паскале есть еще операции деления:

- MOD получение остатка от целочисленного деления,
- DIV частное от целочисленного деления.

Пример. Найти частное A/Z. На Паскале частное A/Z имеет вид: *A div Z*.

Пример. Найти остаток от деления A/Z. На Паскале остаток от деления A/Z имеет вид: *A mod Z*.

*F:=17 DIV 5*; деление нацело, ответ: *F:=3*;

*R:=17 MOD 5*; остаток от деления нацело, ответ: *R:= 2*.

#### *б) Логические выражения*

Пример логических выражений:

*(A>0) and (B>0)* означает (*A* и *B* больше нуля).

*(A>0) or (B>0)* означает (*A* или *B* больше нуля).

В Турбо Паскале определены следующие логические операции из алгебры логики:

- *not* – логическое НЕ (логическое отрицание);
- *and* – логическое И (конъюнкция или логическое умножение);
- *or* – логическое ИЛИ (дизъюнкция или логическое сложение);
- *xor* – исключающее ИЛИ;
- *eqv* – эквивалентность;
- *IMP* – импликация (если..., то...).

## **5.4. Структура программы на языке Паскаль**

Структура программы на языке Turbo Pascal представлена в виде таблицы 15.

Таблица 15

№	Структура программы на языке Turbo Pascal	Комментарий
1	PROGRAM Pr;	Заголовок не обязательен
2	<b>Раздел описаний</b>	Не исполняемая часть программы
3	Begin	<b>Начало</b> раздела операторов
4	<b>Раздел операторов</b>	Исполняемая часть программы
5	END.	<b>Конец</b> раздела операторов

Рассматривая структуру программы, выделяют два раздела в программе:

#### **1. Раздел описаний.**

В разделе описаний задаётся описание констант ключевым словом **const**, переменные в этом разделе задаются ключевым словом **var**, описание нового типа переменных задаётся ключевым словом **type**.

Этот раздел является **исполняемой частью программы**. Чтобы отделить раздел описаний от раздела операторов между ними вставляется слово **begin**, которое означает **начало** исполняемой части программы. Раздел операторов заканчивается словом **end.**, обязательно в конце должна быть точка.

Пара (**begin...end.**) называется операторными скобками.

Такая структура обязательна для любой программы, что является следствием жесткого требования языка: любой нестандартный для языка Турбо Паскаль идентификатор,

используемый в исполняемых операторах, должен быть предварительно описан в разделе описаний.

Описать идентификатор – это значит указать тип связанного с ним объекта программы (константы или переменной).

### 3 Основные операторы языка Pascal.

#### ***Оператор присваивания***

Пример: Представлен оператор присваивания:  $R := \cos(x) + \ln(y);$

Оператор присваивания выполняется в два этапа:

1. Первый этап – выполнение правой части, т.е. в примере вычисляется арифметическое выражение.

2. Второй этап – присвоение результата левой части, т.е. в примере переменной  $R$  присваивается число, полученное при вычислении арифметического выражения.

Примечание. Недопустима запись оператора присваивания в виде:  $\cos(x) + \ln(y) := R;$

#### ***Операторы ввода***

В Паскале нет специальных операторов ввода-вывода. Для обмена информацией в программах Паскаля используются специальные встроенные процедуры, которые не нуждаются в предварительном описании. Таким образом, все операторы ввода-вывода являются операторами обращения к встроенным процедурам ввода или вывода данных.

По операторам READ, READLN вызывается встроенная процедура ввода данных и программа останавливается в ожидании ввода.

Пример: `readln (x,y);`

Следует набрать на клавиатуре два числа через пробел и нажать клавишу «Ввод».

#### ***Операторы вывода***

Основное назначение этих операторов – вывод результатов выполнения программы.

Оператор вывода WRITE выводит строку на экран и оставляет курсор в конце выведенной строки. Если в программе несколько операторов WRITE, то вывод осуществляется в одну строку.

Оператор вывода WRITELN выводит в отдельную строку, после вывода результата осуществляет перевод строки и устанавливает курсор в начало следующей строки экрана. Пример записи оператора вывода переменных X, Y:

`writeln (x,y);`

Если в программе необходимо вывести текст на экран, следует этот текст заключить в апострофы. В частности подсказка на экран для ввода данных записывается оператором:

`writeln ('ввести X,Y,Z');`

Пример записи оператора вывода переменной в формате с фиксированной точкой:

`writeln ('z=', z: 7: 3);`

где: 7 – количество позиций под число  $z$ , 3 – количество позиций под дробную часть числа.

#### ***Комментарий***

Комментарий в Турбо Паскале – это произвольная последовательность любых символов, обрамленная фигурными скобками. Комментарий разрешается вставлять в любое место программы, где по смыслу должен стоять пробел. В качестве ограничителей комментария допускается использование фигурных скобок «{» и «}», а также пары символов «(\*» – слева от комментария и «\*)» – справа от него:

{Это – комментарий}. (\*Это тоже комментарий\*).

Пример: Написать программу линейного алгоритма (рис.8). Вычислить и вывести на экран значение функции:  $z = (x-y)/x + y^2$

Решение: Программа линейного алгоритма имеет вид:

`PROGRAM PR1;`

`VAR`

```

z, x, y:real;
BEGIN
writeln ('ввести x, y'); {На экран выводится подсказка-текст в скобках}
read (x, y); {Ввод с клавиатуры переменных x, y}
z:= (x-y)/x +y*y;
writeln ('z=', z: 7: 3); {Вывод переменной z }
END.

```

В программе после слова BEGIN в фигурных скобках даются комментарии, поясняющие действия операторов.

В примере вывод переменной записан в формате с фиксированной точкой.

Примечание. Необходимо учесть последовательность действий при выполнении арифметического выражения с учётом приоритета арифметических действий:

- а) вычисляются скобки.
- б) операция возведения в степень.
- в) операция деления,
- г) операция сложения.

Назначение операторов передачи управления заключается в организации ветвлений в программе: условных или безусловных. С помощью этих операторов вычислительный процесс передается в указанную оператором точку программы по указанному в операторе условию либо без условия.

## 1. 18 Лекция №18 (4 часа)

**Тема:** «Программная реализация основных алгоритмических и циклических структур на языке Pascal»

### 1.18.1 Вопросы лекции:

1. Программная реализация линейных и разветвляющихся алгоритмов на языке Pascal
2. Программная реализация цикла с пред и пост-условием.

### 1.18.2 Краткое содержание вопросов:

1. Программная реализация линейных и разветвляющихся алгоритмов на языке Pascal

#### *Оператор безусловного перехода*

Действие оператора GOTO состоит в передаче управления соответствующему оператору. Структура оператора:

GOTO метка;

Метка в Турбо Паскале – это произвольный идентификатор, позволяющий именовать некоторый оператор программы и таким образом ссылаться на него. Метка располагается непосредственно перед помечаемым оператором и отделяется от него двоеточием. Перед тем как появиться в программе, метка должна быть задана в разделе описания. Описание меток состоит из зарезервированного слова LABEL (метка), за которым следует список меток.

Пример:

```

LABEL 1; {в разделе описания};
goto 1; {в разделе операторов} {перейти на метку 1}
1:read(x,y); {строка с меткой 1 в разделе операторов}

```

При исполнении меток необходимо руководствоваться следующими правилами:

1. метка, на которую ссылается оператор GOTO, должна быть задана в разделе описаний и она обязательно должна встретиться где-нибудь в разделе операторов программы;

2. метки, описанные в процедуре (функции), локализуются в ней, поэтому передача управления извне процедуры (функции) на метку внутри неё невозможна.

Однако в программировании **не рекомендуется использование оператора GOTO**, т.к. это затрудняет понимание программ, делает ее запутанной и сложной в отладке. Современная технология структурного программирования основана на принципе программирования без GOTO.

### **Операторы условного перехода**

Структура условного оператора имеет следующий вид:

IF <условие> THEN <оператор 1> ELSE <оператор 2>;

где: IF, THEN, ELSE – зарезервированные слова (если, то, иначе);

<условие> – произвольное выражение логического типа;

<оператор 1>, <оператор 2> – любые операторы языка Турбо Паскаль.

Условный оператор работает по следующему алгоритму. Вначале вычисляется условное выражение <условие>. Если результат есть *TRUE* (истина), то выполняется <оператор 1>, а <оператор 2> пропускается; если результат есть *FALSE* (ложь), наоборот, <оператор 1> пропускается, а выполняется <оператор 2>.

Поскольку любой из операторов <оператор 1> и <оператор 2> может быть любого типа, в том числе и условным, а в то же время не каждый из «вложенных» условных операторов может иметь часть ELSE <оператор 2>, то возникает неоднозначность трактовки условий. Эта неоднозначность в Турбо Паскале решается следующим образом: любая встретившаяся часть ELSE соответствует ближайшей к ней «сверху» части THEN условного оператора. Условный оператор позволяет проверить некоторое условие и в зависимости от результатов проверки выполнить то или иное действие. Таким образом, условный оператор – это средство ветвления вычислительного процесса.

Операторы условного перехода подразделяются на: простые и составные, короткие и полные.

### **Простой, короткий IF (если)**

Структура оператора имеет вид:

IF (условие) THEN (оператор или метка);

Пример: Написать программу алгоритма ветвления.

Вычислить  $y := \ln x$ , если  $x > 0$ .

Решение: Программа алгоритма ветвления имеет вид:

```
Program PR2;
var y, x : real;
begin
writeln('ввести x');
Readln (x);
IF x > 0 THEN y := ln(x); {простой, короткий IF}
writeln ('x= ', x: 7:2, 'y= ', y:7 :2);
end.
```

### **Простой, полный IF**

Пример: Написать программу алгоритма ветвления. Вычислить  $y = \ln x$ , если  $x > 0$ , иначе  $y = \cos x$ . В примере рассматривается не только вариант «тогда», но и «иначе».

Решение: Программа алгоритма ветвления имеет вид:

```
Program PR3;
var x, y: real;
begin
writeln('ввести X');
Readln (X);
if x > 0 THEN y := ln (x) ELSE y := cos(x); {простой, полный IF}
writeln ('x = ', x:6:2, 'y = ', y:7:2)
```

end.

Если  $x > 0$ , тогда выполняется оператор за словом THEN (тогда), иначе выполняется оператор, следующий за словом ELSE (иначе).

### **Составной, короткий IF**

**Составной оператор** – это последовательность произвольных операторов программ, заключенная в операторные скобки – зарезервированные слова BEGIN...END.

Составные операторы – важный инструмент Турбо Паскаля, дающий возможность писать программы по современной технологии структурного программирования (без перехода GOTO).

Язык Турбо Паскаль не накладывает никаких ограничений на характер операторов, входящих в составной оператор.

Пример: Вычислить  $y = \ln x$ ,  $z = y - 5x$ , если  $x > 0$ .

Оператор условия запишется в виде:

```
IF x>0 then Begin y:=Ln(x); z:=y-5*x;  
Writeln ('y=', y:7:2, 'z=', z:8:3);end;
```

### **Составной, полный IF**

Пример: Вычислить  $y = \ln x$ ,  $z = y - 5x$ , если  $x > 0$ . Вывод производить для каждого условия.

Решение : Оператор условия запишется в виде:

```
IF x>0 then Begin  
Y:=ln (x);  
Writeln ('x = ', x:6:2 , 'y = ', y:7:2);  
End  
Else begin  
Y:=cos (x);  
Writeln ('x = ', x:6:2,'y = ', y:7:2);  
End;
```

В примере оператор условия составной, так как после слов then, else операторы заключены в операторные скобки.

### **Структурированный (вложенный) IF**

Среди условных операторов можно выделить структурированный, который предполагает проверку условий путём вложения.

### **Структурированный, короткий, простой IF.**

В структурированном операторе содержится последовательная проверка вложенных условий.

Пример: Вычислить  $r = \ln(x+y+z)$ , если  $x > 0$ ,  $y > 0$ ,  $z > 0$ .

Решение: Структурированный оператор условия запишется в виде:

```
IF x>0 then  
IF y>0 then  
IF z>0 then  
R:=LN(X+Y+Z);
```

Пример: Можно этот пример записать иначе коротким, простым оператором IF с помощью логического выражения:

```
IF (x>0) and (y>0) and (z>0) then R:=LN(X+Y+Z);
```

Пример: Вычислить  $r = x + y + z$ , если выполняется хотя бы одно из условий  $x > 0$ ,  $y > 0$ ,  $z > 0$ .

Решение: Оператор условия запишется в виде:

```
IF (x>0) or (y>0) or (z>0) then R:=(x+y +z);
```

### **Структурированный, полный, простой IF.**

Пример: Вычислить:

```
r=ln(x+y+z), если x>0, y>0, z>0;
```

$r = \ln(x+y) + z$ , если  $x > 0, y > 0$ ;  
 $r = \ln(x) + y + z$ , если  $x > 0$ , иначе  $r = x + y + z$ .

Решение: Оператор условия для трёх строк задачи запишется в виде:

```
IF x>0 then
IF y>0 then
IF z>0 then r:=ln (x+y+z)
else r:= ln (x+y)+z
else r:= ln (x)+y+z
else r:= x+y+z;
```

Вначале проверяются три условия. Если они выполняются, то вычисляется

$r = \ln(x+y+z)$ . Иначе выполняются первые два условия, а последнее не выполняется и  $z \leq 0$  (первое слово else относится к последнему условию). В этом случае вычисляется  $r = \ln(x+y)+z$ .

Если из двух условий выполняется только первое, то вычисляется  $r = \ln x + y + z$  (второе слово else относится ко второму условию) и в этом случае  $y \leq 0$ . Последнее слово else относится к первому условию и в этом случае  $x \leq 0$ . В этом случае вычисляется  $r = x + y + z$ .

Пример: Написать программу разветвляющегося алгоритма рис.9.

При выполнении условия  $x > 0$  вычисляется функция:  $z = \ln x + y$ , иначе, а именно, когда  $x = 0$  или  $x < 0$  вычисляется функция:  $z = x + y^2$ .

Решение: Программа алгоритма ветвления имеет вид:

```
PROGRAM PR4;
VAR
x, y, z:real;
BEGIN
Writeln ('ввести x, y'); {На экран выводится подсказка-текст в скобках}
Read (x, y); {Ввод с клавиатуры переменных x, y }
if x>0 then z:=ln (x)+ y
else z=x+y*y;
Writeln ('z=', z: 7: 3); {Вывод результата}
END.
```

Пример: Написать программу разветвляющегося алгоритма рис.10. Найти максимальное число из трех разных целых чисел.

Решение: Программа алгоритма ветвления, поиск максимального из трех разных целых чисел, имеет вид:

```
program max;
var
x, y, z: integer;
begin
writeln ('введите x, y, z');
readln (x, y, z);
if x> y then
if x> z then max:=x
else max:=z
else
if y>z then max:=y
else max:=z;
writeln ('max=', max) ;
readln;
end.
```

Оператор условия выделен курсивом.

## 2. Программная реализация цикла с пред и пост-условием

### **Оператор цикла с параметрами**

Счетный оператор цикла FOR имеет структуру:

FOR  $i:=a$  TO  $b$  DO <оператор>;

FOR, TO, DO – зарезервированные слова (для, до, выполнить);

$i$  – переменная цикла типа INTEGER; (счётчик циклов)

$a$  – начальное значение счётчика циклов (тип INTEGER);

$b$  – конечное значение счётчика циклов (тип INTEGER);

<оператор> – произвольный оператор Турбо Паскаля.

#### **Шаг изменения параметра цикла равен единице.**

Алгоритм выполнения оператора цикла с параметрами при выполнении оператора

FOR:

1. счётчику циклов присваивается начальное значение  $i:=a$ ;
2. проверяется условие  $i>b$  (счётчик циклов больше конечного значения);
3. если условие  $i>b$  выполняется, то на пункт 7, иначе на пункт 4
4. выполняется тело цикла;
5. счётчик увеличивается на единицу:  $i:=i+1$ ;
6. переход на 2;
7. при выполнении условия  $i > b$  цикл заканчивается.

Пример: Найти сумму значений переменной цикла.

Решение: Фрагмент программы с оператором цикла запишется в виде:

For  $i:= 1$  to  $10$  do  $s:=s+i$ ;

Writeln('s=', s);

Счётный оператор цикл FOR может иметь такую структуру:

FOR  $i:= b$  DOWNTO  $a$  DO <оператор>;

Замена зарезервированного слова TO на DOWNTO означает, что шаг наращивания переменной цикла равен (-1).

Пример: Найти сумму значений переменной цикла.

Фрагмент программы с оператором цикла запишется в виде:

For  $i:=10$  downto  $1$  do  $s:=s+i$ ;

Writeln('s=', s);

{Результат получится тот же, что и в примере с циклом For..to..do}.

При работе с оператором FOR следует соблюдать ряд правил:

1. Нельзя войти в цикл, минуя оператор FOR.
2. Нельзя изменять параметры цикла ( $a, b$ ) внутри цикла.
3. Параметры цикла и переменная цикла должны быть целыми.
4. Шаг цикла может быть единица или минус единица.
5. Естественное окончание цикла осуществляется при условии  $i>b$  при шаге=1.
6. Из цикла можно выйти до естественного окончания цикла по условию.

### **Оператор цикла WHILE с предусловием**

Структура оператора имеет вид:

WHILE <условие> DO <оператор>;

WHILE, DO – зарезервированные слова (WHILE – пока; DO – выполнить);

<условие> – выражение логического типа;

<оператор> – произвольный оператор Турбо Паскаля.

Если выражение <условие> имеет значение TRUE, то выполняется <оператор>, после чего вычисление выражения <условие> и его проверка повторяются. Если <условие> имеет значение FALSE, оператор WHILE прекращает свою работу.

Пример: Переписать фрагмент примера с циклом While...do, используя оператор цикла с предусловием.

Решение: Фрагмент программы с оператором цикла запишется в виде:

```

s:=0; i:=1;
while i<=10 do
Begin
s:=s+i;
i:=i+1;
End;
Writeln('s=', s);

```

В примере рассматривается составной оператор цикла, тело цикла заключено в операторные скобки.

### **Оператор цикла REPEAT...UNTIL с постусловием**

Структура оператора имеет вид:

REPEAT <тело\_цикла> UNTIL <условие>;

REPEAT, UNTIL – зарезервированные слова (повторять до тех пор, пока не будет выполнено условие);

<тело\_цикла> – произвольная последовательность операторов Турбо Паскаля;

<условие> – выражение логического типа.

Операторы, входящие в <тело\_цикла>, выполняются хотя бы один раз, после чего вычисляется выражение <условие>: если его значение есть FALSE, операторы <тело\_цикла> повторяются, в противном случае оператор REPEAT...UNTIL завершает свою работу.

**Пример:** Переписать фрагмент примера с циклом While...do, используя оператор цикла с постусловием.

Решение: Фрагмент программы с оператором цикла запишется в виде:

```

s:=0; i:=1;
repeat
s:=s+i;
i:=i+1;
Until i>10;
Writeln('s=',s);

```

В пример цикл выполняется пока переменная  $i \leq 10$ , при  $i > 10$  цикл закончится.

**Пример:** Написать программу циклического алгоритма рис.11.

В цикле вычислить значение функции  $z = x^*y$  при условии, что одна из переменных - x меняется в каждом цикле, а другая переменная- y не меняется и может быть любым целым числом.

Решение: Программа алгоритма цикла со счётчиком имеет вид:

```

PROGRAM PR5;
Var
x, y, z, i, n :integer;
begin
{циклический алгоритм}
writeln ('ввести x, y, количество циклов-n');
readln (x, y, n);
for i:=1 to n do {оператор цикла с параметрами}
begin
z:=x*y;
Writeln ('x= ', x, ' y= ', y, ' z= ', z);
x:=x+1;
End; {конец оператора цикла с параметрами}
End.

```

Если неизвестно количество циклов, то следует выбрать любой из двух операторов цикла: с предусловием или с постусловием.

**Пример:** Пока  $y > x$  вычислить  $y = y - x$ , , если  $y = 30$ ,  $x = 4$ . Вывести на экран количество циклов и значения переменной  $y$  в цикле. Алгоритм этой задачи представлен в виде блок-схемы рис.12.

**Решение:** Программа алгоритма цикла с предусловием имеет вид:

PROGRAM PR6;

Var

$i, x, y$ : integer;

begin

{циклический алгоритм}

$x := 4$ ;  $y := 30$ ;  $i := 0$ ;

{оператор цикла с предусловием}

while  $y > x$  do

begin

$y := y - x$ ;

$i := i + 1$ ;

readln ('i = ',  $i$ , 'x = ',  $x$ , 'y = ',  $y$ );

end; {конец оператора}

end.

В примере используется оператор цикла с предусловием, который работает при условии  $y > x$ . Условие проверяется при входе в цикл.

**Пример:** Составить программу для алгоритма, представленного на рисунке 13, используя оператор цикла с постусловием. Цикл выполняется при условии  $y > x$ , но в конце оператора проверяется условие выхода из цикла ( $y \leq x$ ).

**Решение:** Программа алгоритма цикла с постусловием имеет вид:

PROGRAM PR7;

Var

$i, x, y$ : integer;

begin

{циклический алгоритм}

$x := 4$ ;  $y := 30$ ;  $i := 0$ ;

{оператор цикла с постусловием}

repeat

$y := y - x$ ;

$i := i + 1$ ;

readln ('i = ',  $i$ , 'x = ',  $x$ , 'y = ',  $y$ );

until  $y \leq x$ ; {конец оператора}

end.

## 1. 19 Лекция №19 (2 часа)

**Тема:** «Локальные и глобальные сети ЭВМ»

### 1.19.1 Вопросы лекции:

1. Понятие и виды сетей.
2. Топологии локальных сетей.

**1.19.2 Краткое содержание вопросов:** (тезисно изложить основное содержание рассматриваемых вопросов)

## 1. Понятие и виды сетей

Сегодняшняя статья открывает новую рубрику на блоге, которая будет называться “Сети”. В данной рубрике будет освещаться широчайший круг вопросов, касающихся **компьютерных сетей**. Первые статьи рубрики будут посвящены разъяснению некоторых базовых понятий, с которыми вы столкнетесь при работе с сетью. А сегодня мы поговорим о том, какие компоненты потребуются для создания сети и какие существуют **виды сетей**.

**Компьютерная сеть** – это совокупность компьютерного и сетевого оборудования, соединенного с помощью каналов связи в единую систему. Для создания компьютерной сети нам потребуются следующие компоненты:

- компьютеры, имеющие возможности для подключения к сети (например, сетевая карта, которая есть в каждом современном ПК);
- передающая среда или каналы связи (кабельные, спутниковые, телефонные, волоконно-оптические и радиоканалы);
- сетевое оборудование (например, коммутатор или роутер);
- сетевое программное обеспечение (как правило, входит в состав операционной системы или поставляется вместе с сетевым оборудованием).

Компьютерные сети принято подразделять на два основных вида: глобальные и локальные.

**Локальные сети** (Local Area Network – LAN) обладают замкнутой инфраструктурой до выхода на поставщиков услуг интернета. Термин “локальная сеть” может описывать и маленькую офисную сеть, и сеть большого завода, занимающего несколько гектаров. Применительно к организациям, предприятиям, фирмам используется термин **корпоративная сеть** – локальная сеть отдельной организации (юридического лица) независимо от занимаемой ею территории.

Корпоративные сети являются сетями закрытого типа, доступ к ним разрешен только ограниченному кругу пользователей (например, сотрудникам компании). Глобальные сети ориентированы на обслуживание любых пользователей.

**Глобальная сеть** (Wide Area Network – WAN) охватывает большие географические регионы и состоит из множества локальных сетей. С глобальной сетью, которая состоит из нескольких тысяч сетей и компьютеров, знакомы все – это Интернет.

Системному администратору приходится иметь дело с локальными (корпоративными) сетями. Обычный пользовательский компьютер, подключенный к локальной сети, называется **рабочей станцией**. Компьютер, предоставляющий свои ресурсы для общего использования другим компьютерам сети, называется **сервером**; а компьютер, обращающийся к совместно используемым ресурсам на сервере – **клиентом**.

Существуют различные **виды серверов**: файловые (для хранения общих файлов), серверы баз данных, серверы приложений (обеспечивающие удаленную работу программ на клиентах), web-серверы (для хранения web-контента) и другие.

Загрузка сети характеризуется параметром, называемым трафиком. **Трафик** – это поток сообщений в сети передачи данных. Под ним понимают количественное измерение числа проходящих по сети блоков данных и их длины, выраженное в битах в секунду. Например, скорость передачи данных в современных локальных сетях может быть 100Мбит/с или 1Гбит/с.

В настоящее время в мире насчитывается огромное количество всевозможного сетевого и компьютерного оборудования, позволяющего организовать самые различные компьютерные сети. Все многообразие компьютерных сетей можно разделить на несколько видов по различным признакам:

**По территории:**

- локальные – охватывают небольшие территории и располагаются внутри отдельных офисов, банков, корпораций, домов;
- региональные – образуются путем объединения локальных сетей на отдельных территориях;
- глобальные (интернет).

**По способу связи компьютеров:**

- проводные (компьютеры соединяются посредством кабеля);
- беспроводные (компьютеры обмениваются информацией посредством радиоволн. например, по технологии WI-FI или Bluetooth).

**По способу управления:**

- с централизованным управлением – для управления процессом обмена данных в сети выделяется одна или несколько машин (серверов);
- децентрализованные сети – не содержат в своем составе выделенных серверов, функции управления сетью передаются по очереди от одного компьютера другому.

**По составу вычислительных средств:**

- однородные – объединяют однородные вычислительные средства (компьютеры);
- неоднородные – объединяют различные вычислительные средства (например: ПК, торговые терминалы, веб-камеры и сетевое хранилище данных).

**По типам среды передачи** сети разделяются на оптоволоконные, с передачей информации по радиоканалам, в инфракрасном диапазоне, через спутниковый канал и т.д.

Вы можете встретить и другие классификации компьютерных сетей. Как правило, системному администратору приходится иметь дело с локальными проводными сетями с централизованным, либо децентрализованным управлением.

## 2. Топологии локальных сетей.

Все компьютеры в локальной сети соединены линиями связи. Геометрическое расположение линий связи относительно узлов сети и физическое подключение узлов к сети называется физической топологией. В зависимости от топологии различают сети: шинной, кольцевой, звездной, иерархической и произвольной структуры.

Различают физическую и логическую топологию. Логическая и физическая топологии сети независимы друг от друга. Физическая топология - это геометрия построения сети, а логическая топология определяет направления потоков данных между узлами сети и способы передачи данных.

В настоящее время в локальных сетях используются следующие физические топологии:

- физическая "шина" (bus);
- физическая "звезда" (star);
- физическое "кольцо" (ring);
- физическая "звезда" и логическое "кольцо" (Token Ring).

### **Шинная топология**

Сети с шинной топологией используют линейный моноканал (коаксиальный кабель) передачи данных, на концах которого устанавливаются оконечные сопротивления (терминалы). Каждый компьютер подключается к коаксиальному кабелю с помощью Т-разъема (Т - коннектор). Данные от передающего узла сети передаются по шине в обе стороны, отражаясь от оконечных терминалов. Терминалы предотвращают отражение сигналов, т.е. используются для гашения сигналов, которые достигают концов канала передачи данных.

Таким образом, информация поступает на все узлы, но принимается только тем узлом, которому она предназначается. В топологии логическая шина среда передачи данных

используются совместно и одновременно всеми ПК сети, а сигналы от ПК распространяются одновременно во все направления по среде передачи. Так как передача сигналов в топологии физическая шина является широковещательной, т.е. сигналы распространяются одновременно во все направления, то логическая топология данной локальной сети является логической шиной.

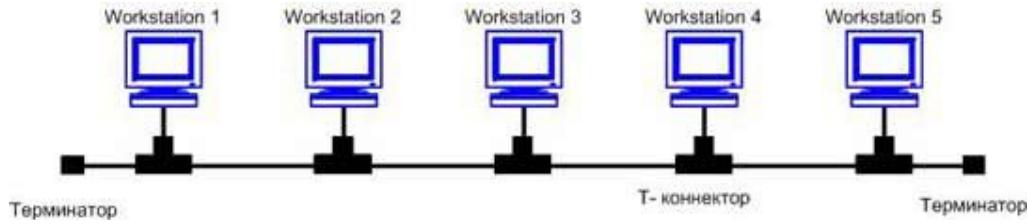


Рис. 1.

Данная топология применяется в локальных сетях с архитектурой Ethernet (классы 10Base-5 и 10Base-2 для толстого и тонкого коаксиального кабеля соответственно).

Преимущества сетей шинной топологии:

- отказ одного из узлов не влияет на работу сети в целом;
- сеть легко настраивать и конфигурировать;
- сеть устойчива к неисправностям отдельных узлов.

Недостатки сетей шинной топологии:

- разрыв кабеля может повлиять на работу всей сети;
- ограниченная длина кабеля и количество рабочих станций;
- трудно определить дефекты соединений.

### **Топология типа “звезда”**

В сети построенной по топологии типа “звезда” каждая рабочая станция подсоединяется кабелем (витой парой) к концентратору или хабу (*hub*). Концентратор обеспечивает параллельное соединение ПК и, таким образом, все компьютеры, подключенные к сети, могут общаться друг с другом.

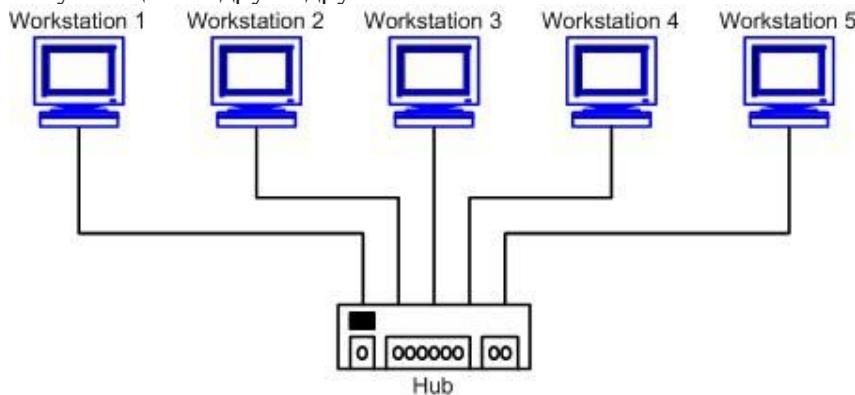


Рис. 2.

Данные от передающей станции сети передаются через хаб по всем линиям связи всем ПК. Информация поступает на все рабочие станции, но принимается только теми станциями, которым она предназначается. Так как передача сигналов в топологии физическая звезда является широковещательной, т.е. сигналы от ПК распространяются одновременно во все направления, то логическая топология данной локальной сети является логической шиной.

Данная топология применяется в локальных сетях с архитектурой 10Base-T Ethernet.

Преимущества сетей топологии звезда:

- легко подключить новый ПК;
- имеется возможность централизованного управления;
- сеть устойчива к неисправностям отдельных ПК и к разрывам соединения отдельных ПК.

Недостатки сетей топологии звезды:

- отказ хаба влияет на работу всей сети;
- большой расход кабеля.

### **Топология “кольцо”**

В сети с топологией кольцо все узлы соединены каналами связи в неразрывное кольцо (необязательно окружность), по которому передаются данные. Выход одного ПК соединяется со входом другого ПК. Начав движение из одной точки, данные, в конечном счете, попадают на его начало. Данные в кольце всегда движутся в одном и том же направлении.

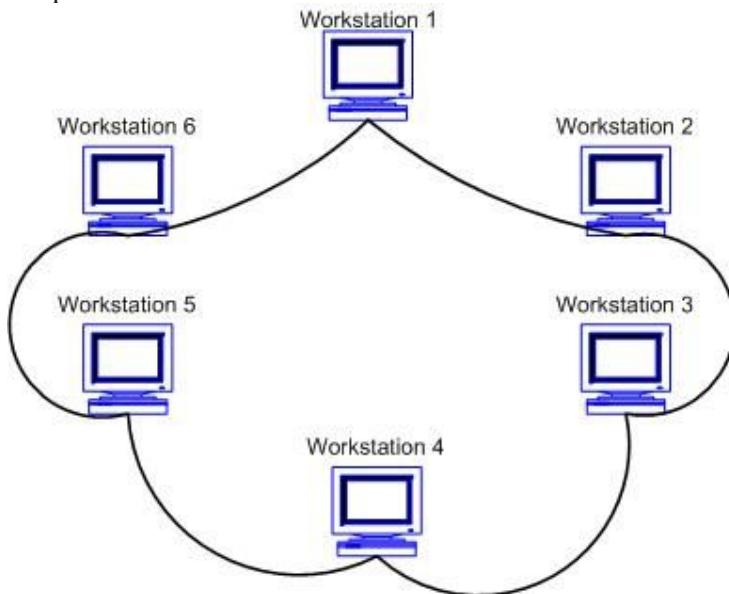


Рис. 3.

Принимающая рабочая станция распознает и получает только адресованное ей сообщение. В сети с топологией типа физическое кольцо используется маркерный доступ, который предоставляет станции право на использование кольца в определенном порядке. Логическая топология данной сети - логическое кольцо. Данную сеть очень легко создавать и настраивать.

К основному недостатку сетей топологии кольцо является то, что повреждение линии связи в одном месте или отказ ПК приводит к неработоспособности всей сети.

Как правило, в чистом виде топология “кольцо” не применяется из-за своей ненадёжности, поэтому на практике применяются различные модификации кольцевой топологии.

### **Топология Token Ring**

Эта топология основана на топологии "физическое кольцо с подключением типа звезды". В данной топологии все рабочие станции подключаются к центральному концентратору (Token Ring) как в топологии физическая звезда. Центральный концентратор - это интеллектуальное устройство, которое с помощью перемычек обеспечивает последовательное соединение выхода одной станции со входом другой станции.

Другими словами с помощью концентратора каждая станция соединяется только с двумя другими станциями (предыдущей и последующей станциями). Таким образом, рабочие станции связаны петлей кабеля, по которой пакеты данных передаются от одной станции к другой и каждая станция ретранслирует эти посланные пакеты. В каждой рабочей станции имеется для этого приемо-передающее устройство, которое позволяет управлять прохождением данных в сети. Физически такая сеть построена по типу топологии “звезда”.

Концентратор создаёт первичное (основное) и резервное кольца. Если в основном кольце произойдёт обрыв, то его можно обойти, воспользовавшись резервным кольцом, так как

используется четырёхжильный кабель. Отказ станции или обрыв линии связи рабочей станции не влечет за собой отказ сети как в топологии кольцо, потому что концентратор отключает неисправную станцию и замкнет кольцо передачи данных.

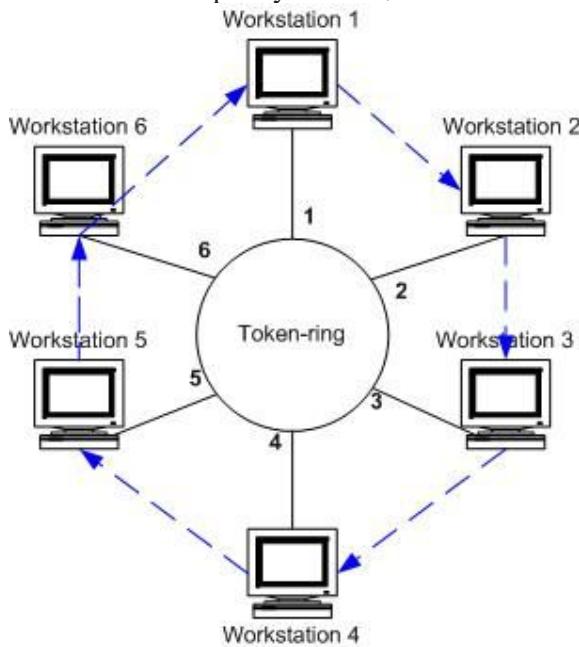


Рис. 4.

В архитектуре Token Ring маркер передаётся от узла к узлу по логическому кольцу, созданному центральным концентратором. Такая маркерная передача осуществляется в фиксированном направлении (направление движения маркера и пакетов данных представлено на рисунке стрелками синего цвета). Станция, обладающая маркером, может отправить данные другой станции.

Для передачи данных рабочие станции должны сначала дождаться прихода свободного маркера. В маркере содержится адрес станции, пославшей этот маркер, а также адрес той станции, которой он предназначается. После этого отправитель передает маркер следующей в сети станции для того, чтобы и та могла отправить свои данные.

Один из узлов сети (обычно для этого используется файл-сервер) создаёт маркер, который отправляется в кольцо сети. Такой узел выступает в качестве активного монитора, который следит за тем, чтобы маркер не был утерян или разрушен.

Преимущества сетей топологии Token Ring:

- топология обеспечивает равный доступ ко всем рабочим станциям;
- высокая надежность, так как сеть устойчива к неисправностям отдельных станций и к разрывам соединения отдельных станций.

Недостатки сетей топологии Token Ring: большой расход кабеля и соответственно дорогостоящая разводка линий связи.

## 1. 20 Лекция №20 (2 часа)

Тема: «Компоненты компьютерных сетей»

### 1.20.1 Вопросы лекции:

1. Программные компоненты компьютерных сетей.
2. Аппаратные компоненты компьютерных сетей

**1.20.2 Краткое содержание вопросов:** (тезисно изложить основное содержание рассматриваемых вопросов)

## 1. Программные компоненты компьютерных сетей

Весь комплекс программно-аппаратных средств сети может быть описан многослойной моделью. В основе любой сети лежит аппаратный слой стандартизованных компьютерных платформ, т.е. система конечного пользователя сети, в качестве которого может выступать компьютер или терминальное устройство (любое устройство ввода-вывода или отображения информации). Компьютеры в узлах сети иногда называют хост-машинами или просто хостами. В настоящее время в сетях широко и успешно применяются компьютеры различных классов - от персональных компьютеров до мэйнфреймов и суперЭВМ. Набор компьютеров в сети должен соответствовать набору разнообразных задач, решаемых сетью. Второй слой - это коммуникационное оборудование. Хотя компьютеры и являются центральными элементами обработки данных в сетях, в последнее время не менее важную роль стали играть коммуникационные устройства. Кабельные системы, повторители, мосты, коммутаторы, маршрутизаторы и модульные концентраторы из вспомогательных компонентов сети превратились в основные наряду с компьютерами и системным программным обеспечением как по влиянию на характеристики сети, так и по стоимости. Сегодня коммуникационное устройство может представлять собой сложный специализированный мультипроцессор, который нужно конфигурировать, оптимизировать и администрировать. Третьим слоем, образующим программную платформу сети, являются операционные системы (ОС). От того, какие концепции управления локальными и распределенными ресурсами положены в основу сетевой ОС, зависит эффективность работы всей сети. При проектировании сети важно учитывать, насколько просто данная операционная система может взаимодействовать с другими ОС сети, насколько она обеспечивает безопасность и защищенность данных, до какой степени она позволяет наращивать число пользователей, можно ли перенести ее на компьютер другого типа и многие другие соображения. Самым верхним слоем сетевых средств являются различные сетевые приложения, такие как сетевые базы данных, почтовые системы, средства архивирования данных, системы автоматизации коллективной работы и др. Очень важно представлять диапазон возможностей, предоставляемых приложениями для различных областей применения, а также знать, насколько они совместимы с другими сетевыми приложениями и операционными системами.

## 2. Аппаратные компоненты компьютерных сетей

Основными аппаратными компонентами компьютерной сети (рис. 1) являются:

- Серверы;
  - Рабочие станции;
  - Каналы (линии) связи;
  - Аппаратура передачи данных.



Рис. 1. Основные аппаратные компоненты компьютерной сети

### *Серверы и рабочие станции*

Серверы являются достаточно мощными компьютерами, так как должны обеспечить высокую скорость передачи данных и обработки запросов. Сервер - источник ресурсов сети, компьютер с большой емкостью оперативной памяти, жесткими дисками большой емкости и дополнительными накопителями информации. Серверов в сети может быть много.

Сервер работает под управлением сетевой операционной системы, которая обеспечивает одновременный доступ пользователей сети к расположенным на нем данным. Требования к серверу определяются задачами, которые на него возложены в конкретной сети. Успешность выполнения задач сервером зависит от установленного программного обеспечения. Серверы могут осуществлять хранение данных, пересылку почтовых сообщений, управление базами данных, удаленную обработку заданий, доступ к web-страницам, печать заданий и ряд других функций, потребность в которых может возникнуть у пользователей сети.

Компьютер, подключенный к сети и имеющий доступ к ее ресурсам, называется *рабочей станцией*.

Роли сервера и рабочей станции могут быть различными в сетях.

Например, файл-сервер выполняет следующие функции:

- хранение данных;
- архивирование данных;
- синхронизацию изменений данных различными пользователями;
- передачу данных.

Файл-сервер получает запрос на доступ к файлу от рабочей станции. Файл отсылается на рабочую станцию. Пользователь на рабочей станции обрабатывает данные. Затем файл возвращается серверу обратно.

Существует и другое разделение ролей между компьютерами в сети, Например, сеть типа Клиент/Сервер.

*Клиентом* называют рабочую станцию, на которой установлено программное обеспечение, обеспечивающее решение задач, сформированных в процессе работы пользователя.

В процессе обработки данных клиент формирует запрос серверу для выполнения различных задач: пересылки сообщения, просмотр web-страниц, и т. д.

Сервер, выполняет запрос, поступивший от клиента. Результаты выполнения запроса передаются клиенту. Часть задач может выполняться на стороне клиента. Обмен данными, обработка запросов и обработка данных продолжаются между сервером и клиентом, пока они не выполнят задачу. Обработка данных может выполняться как сервером, так и клиентом.

Сервер обеспечивает хранение данных общего пользования, организует доступ к этим данным и передает данные клиенту.

Клиент обрабатывает полученные данные и представляет результаты обработки в виде, удобном для пользователя.

### ***Каналы связи***

*Канал связи* (или *линия связи*) - физическая среда, по которой передаются информационные сигналы аппаратуры передачи данных.

Среда передачи данных может базироваться на различных физических принципах действия. Например, это может быть кабель и соединительные разъемы. Физической средой передачи данных может служить земная атмосфера или космическое пространство, через которые распространяются информационные сигналы.

В телекоммуникационных системах данные передаются с помощью электрического тока, радиосигналов или световых сигналов. Все эти физические процессы представляют собой колебания электромагнитного поля различной частоты и природы. Основной характеристикой физических каналов служит *скорость передачи данных*, измеряемая в количестве бит (Кбит, Мбит) в секунду.

В зависимости от физической среды линии связи могут быть классифицированы в виде следующих групп: проводные линии, кабельные линии, радиоканалы наземной и спутниковой связи.

*Проводные линии* - это неэкранированные провода, проложенные над землей по воздуху. По ним, в основном, передаются телефонные или телеграфные сигналы, но их можно использовать и для передачи данных, посыпаемых от одного компьютера к другому. Скорость передачи данных по таким линиям измеряется десятками Кбит/сек.

*Кабельные линии* - это совокупность изолированных разными слоями проводников. В основном, используются волоконно-оптические кабели и кабели на основе медных проводов: витая пара (скорость от 100 Мбит/сек до 1Гбит/сек) и коаксиальный кабель (скорость – десятки Мбит/сек). Кабели используются для внутренней и внешней проводки. Внешние кабели подразделяются на подземные, подводные и кабели воздушной проводки.

Наиболее качественным кабелем является волоконно-оптический кабель. Он состоит из гибких стеклянных волокон, по которым распространяются световые сигналы. Он обеспечивает передачу сигнала с очень высокой скоростью (до 10 Гбит/сек и выше). Этот тип кабеля является надежным, так как хорошо защищает данные от внешних помех.

*Радиоканалы наземной и спутниковой связи*, представляют собой канал, образованные между передатчиком и приемником радиоволн. Радиоканалы различаются используемыми частотными диапазонами и дальностью канала. Они обеспечивают различную скорость передачи данных. Спутниковые каналы и радиосвязь используются в случаях, когда нельзя использовать кабельный канал, например, в малонаселенных местностях, для связи с пользователями мобильной радиосети.

В компьютерных сетях применяются все описанные типы физических сред передачи данных, но волоконно-оптический кабель представляется наиболее перспективным. Его уже начали широко использовать в качестве магистралей территориальных, городских сетей, а также используют на высокоскоростных участках локальных сетей.

### *Аппаратура передачи данных*

Аппаратура передачи данных служит для непосредственного присоединения компьютеров к линии связи. К ней относятся устройства передачи данных, которые отвечают за передачу информации в физическую среду (линию связи) и прием из нее данных: сетевая карта (адаптер), модемы, устройства подключения к цифровым каналам, терминальные адAPTERы сетей ISBN, мосты, маршрутизаторы, шлюзы и пр.

*Сетевая карта (адаптер)* указывает адрес компьютера. Компьютер, работающий в сети, должен быть правильно опознан, то есть, его адрес должен быть уникальным. Поэтому, производителям сетевых карт выделяют некоторое количество разных адресов, которые не совпадают между собой.



Рис. 2. Сетевой адаптер (карта)

*Модемы* - устройства для преобразования цифровых сигналов компьютера в аналоговые сигналы телефонной линии и обратно. Распространенная скорость передачи данных – 56 Кбит/сек.

*Терминальные адAPTERы сетей ISBN* ( Integrated Services Digital Network ) – телефонная сеть с интеграцией услуг. Основой такой сети является цифровая обработка сигналов. Абоненту предоставляется два канала для голосовой связи и передачи данных со скоростью 64 Кбит/сек.

*Устройства подключения к цифровым каналам* предназначены для улучшения качества сигналов и создания постоянного составного канала между двумя абонентами сети. Используются, в основном, на линиях связи большой протяженности.

*Мосты* - устройства, соединяющие две сети и использующие одинаковые методы передачи данных.

*Маршрутизаторы* или *роутеры* - устройства, соединяющие сети разного типа, но использующие одну операционную систему.

*Шлюзы* - устройства, позволяющие организовать обмен данными между двумя сетями, использующими различные правила взаимодействия, например, подключить локальную вычислительную сеть к глобальной.

Мосты, маршрутизаторы, шлюзы могут работать как в режиме полного выделения функций, так и в режиме совмещения их с функциями рабочей станции вычислительной сети.

К аппаратуре передачи данных относятся также:

- Усилители - устройства, повышающие мощность сигналов;

- Регенераторы, восстанавливающие форму импульсных сигналов, искаженных при передаче на большие расстояния;
- Коммутаторы – аппаратура для создания между двумя абонентами сети долговременного непрерывного составного канала из отрезков физической среды с усилителями.

Невидимая пользователям сеть с промежуточной аппаратурой канала связи образует сложную сеть, которая называется первичной сетью. Она не поддерживает никаких служб для пользователя, а лишь служит основой для построения других сетей.

## 1. 21 Лекция №21 (2 часа)

**Тема:** «Глобальная компьютерная сеть Интернет»

### 1.21.1 Вопросы лекции:

1. Глобальные компьютерные сети.
2. История развития Интернет.
3. Принцип построения сети Интернет.

### 1.21.2 Краткое содержание вопросов:

1. Глобальные компьютерные сети.

**Глобальная компьютерная сеть**, ГКС ([англ. Wide Area Network, WAN](#)) — [компьютерная сеть](#), охватывающая большие территории и включающая в себя большое число компьютеров.

ГКС служат для объединения разрозненных сетей так, чтобы пользователи и компьютеры, где бы они ни находились, могли взаимодействовать со всеми остальными участниками глобальной сети.

Некоторые ГКС построены исключительно для частных организаций, другие являются средством коммуникации корпоративных [ЛВС](#) с сетью Интернет или посредством Интернет с удалёнными сетями, входящими в состав корпоративных. Чаще всего ГКС опирается на выделенные линии, на одном конце которых [маршрутизатор](#) подключается к ЛВС, а на другом коммутатор связывается с остальными частями ГКС. Основными используемыми протоколами являются [TCP/IP](#), [SONET/SDH](#), [MPLS](#), [ATM](#) и [Frame relay](#). Ранее был широко распространён протокол [X.25](#), который может по праву считаться прародителем [Frame relay](#).

Связывает компьютеры, рассредоточенные на расстоянии сотен и тысяч километров. Часто используются уже существующие не очень качественные линии связи. Более низкие, чем в [локальных сетях](#), скорости передачи данных (десятки килобит в секунду) ограничивают набор услуг передачей файлов, преимущественно не в оперативном, а в фоновом режиме, с использованием [электронной почты](#). Для стойкой передачи дискретных данных применяются более сложные методы и оборудование, чем в локальных сетях.

Глобальные сети отличаются от локальных тем, что глобальные сети рассчитаны на неограниченное число абонентов и используют, как правило, некачественные каналы связи и невысокую скорость передачи.

В глобальных сетях намного более важно не качество связи, а сам факт её существования. Правда, в настоящий момент уже нельзя провести четкий и однозначный предел между локальными и глобальными сетями. Большинство локальных сетей имеют выход в глобальную сеть, но характер переданной информации, принципы организации обмена, режимы доступа к ресурсам внутри локальной сети, как правило, сильно отличаются от тех,

что приняты в глобальной сети. И хотя все компьютеры локальной сети в данном случае включены также и в глобальную сеть, специфику локальной сети это не отменяет. Возможность выхода в глобальную сеть остается всего лишь одним из ресурсов, поделенным пользователями локальной сети.

## 2. История развития Интернет.

В 1969 агентство ARPA (Агентство передовых исследовательских проектов) занялось исследованиями с целью создать экспериментальную сеть "с коммутацией пакетов". Эта сеть была создана и получила очень простое и аскетичное наименование ARPANET, т.е. сеть агентства... Сеть была построена для изучения технологий независимой передачи потока данных в системе связи.

Именно успешным развитием исследовательского проекта, мы обязаны тому, что многие организации, которые принимали участие в ее создании, начали применять ее в своих повседневных целях, и в 1975 году экспериментальную сеть объявили рабочей, причем ответственность за нее была возложена на DCA (Агентство Оборонной связи США). Одновременно, специалисты занялись разработкой основ TCP/IP (Transmission Control Protocol / Internet Protocol - Протоколы управления процессом передачи / Интернет-протокол).

TCP/IP были приняты, в качестве Военных Стандартов (MIL STD) в 1983 году, после чего от всех хостов (компьютеров), подключенных к APRANET стали требовать работать только с данными протоколами. Одновременно стал распространяться термин "Internet", в то время, как APRANET была разделена на две отдельных сети: MILNET (Военная Сеть) - несекретная часть Оборонной Сети Передачи данных (DDN) и новую (уменьшенных размеров) APRANET. Термин Internet употребляли тогда, когда имели в виду сразу обе сети.

В 1985 году, Фонд Национальной Науки (NSF) принял участие в создании собственной сети NSFNet, которая вскоре была подключена к интернет. Первоначально в состав NSF входили 5 супер-компьютерных центров, впрочем, меньше, чем в APRANET, а скорость передачи данных в каналах связи не превышала 56 кбит/с. В то же время, создание NSFNet стало заметным вкладом в развитие интернет, поскольку позволило по-новому взглянуть на то, как можно использовать интернет. Фонд поставил задачу, чтобы каждый ученый, каждый инженер в США оказались "подключены" к единой сети, а потому приступили к созданию сети с более быстрыми каналами, которая бы объединила многочисленные региональные и локальные сети.

Не так давно, в 1990 году APRANET формально завершила свое существование, а еще через 5 лет, NSFNet прекратила играть первую скрипку в оркестре интернет. Интернет вырос далеко за пределы того, каким его видели и проектировали, он перерос те агентства и организации, которые его создавали, они более не могли играть в его росте доминирующую роль. Сегодня это мощная всемирная сеть связи, основанная на распределенных коммутационных элементах - хабах и каналах связи. С 1983 года интернет растет по экспоненте, и едва ли ни одна деталь сохранилась с тех времен, - интернет все еще работает на основе набора протоколов TCP/IP.

Если термин "интернет" первоначально использовался для описания сети, построенной на базе интернет-протокола IP, то сейчас это слово приобрело глобальный смысл и лишь иногда применяется в качестве названия набора объединенных сетей. Строго говоря - интернет, это любой набор отдельных в физическом смысле сетей, которые соединены

между собой единым протоколом IP, что позволяет говорить о них, как об одной логической сети.

Бурный рост интернет, вызвал повышенный интерес к протоколам TCP/IP, а в итоге появились специалисты и компании, которые разобравшись в протоколе, нашли для него и ряд других приложений, прежде всего, использовав этот же протокол для построения локальных вычислительных сетей (LAN - Local Area Network) даже тогда, когда не предусматривалось их подключение к Интернет. Кроме того, TCP/IP стал применяться при создании корпоративных сетей, которые взяли на вооружение интернет-технологии, в том числе WWW (World Wide Web) - мировую паутину, чтобы наладить эффективный обмен внутрикорпоративной информацией. Эти корпоративные сети получили название "интранет" и могут подключаться, либо не быть подключены к интернет.

С 1990 года интернет до некоторой степени саморазвивающимся объектом, как говорят инженеры, в нем действует положительная обратная связь, т.е. по мере того, как больше ресурсов (информационных и физических) становятся доступными, так большее число людей и компаний стремятся получить доступ к этим ресурсам.

### 3 Принцип построения сети Интернет.

Сеть **Internet** - это сеть сетей. Она объединяет как локальные, так и глобальные сети. Центральным местом при обсуждении принципов построения сети является семейство протоколов **TCP/IP**.

Главной задачей **TCP/IP** является объединение в сеть пакетных подсетей через шлюзы. Каждая сеть работает по своим собственным законам, предполагается, что шлюз может принять пакет из другой сети и доставить его по указанному адресу. Реально пакет из одной сети передается в другую сеть через последовательность шлюзов, которые обеспечивают сквозную маршрутизацию пакетов по всей сети.

Такой механизм доставки становится возможным благодаря реализации во всех узлах сети межсетевого обмена **IP**. Базируясь на классификации **OSI**(Open System Integration-Взаимодействие открытых систем), всю архитектуру семейства **TCP/IP** можно представить в виде схемы:

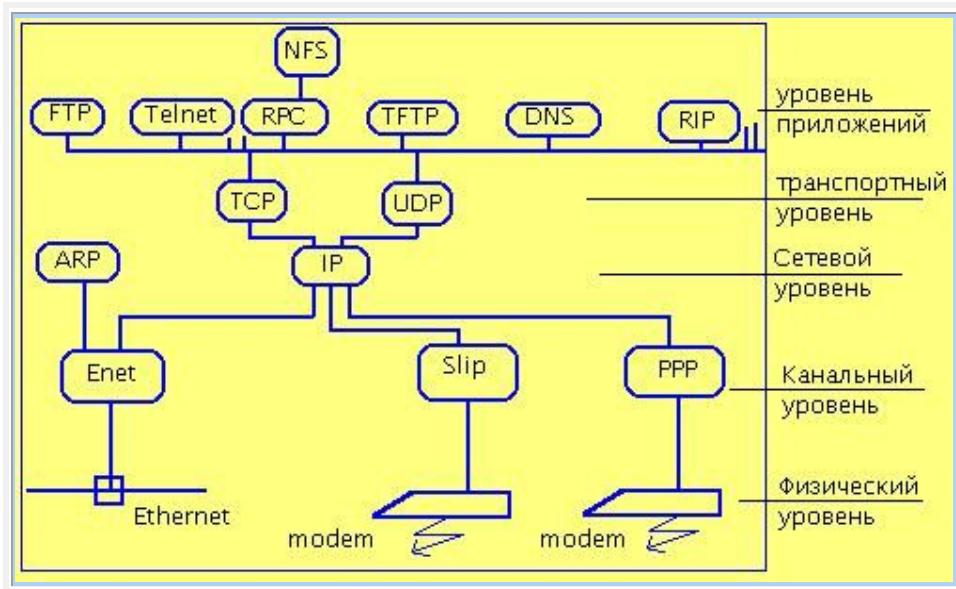


рис 1. Схема модулей, реализующих протоколы семейства TCP/IP в узле сети.

Вся схема называется стеком протоколов TCP/IP или просто *стеком TCP/IP*. Прямоугольниками на схеме обозначены модули, обрабатывающие пакеты, линиями- пути передачи данных. Схема является упрощенным представлением стека протоколов TCP/IP. Современные IP-сети гораздо сложнее.

*Драйвер*- программа, непосредственно взаимодействующая с сетевым адаптером.

*Модуль*- программа, взаимодействующая с драйвером, с сетевыми прикладными программами или другими модулями.

*Сетевой интерфейс*- физическое устройство, подключающее компьютер к сети.(в данном случае это карта Ethernet).

*Кадр*- это блок данных, который принимает/передает сетевой интерфейс.

*TCP*-Transmission Control Protocol- базовый транспортный протокол.

*UDP*- User Datagram Protocol- второй транспортный протокол семейства TCP/IP .

*IP-пакет*- это блок данных, которым обменивается модуль IP с сетевым интерфейсом.

*TCP-сегмент*- блок данных, которым обменивается модуль IP с модулем TCP.

*UDP-датаграмма*- блок данных, которым обменивается модуль Ipc модулем UDP.

*Прикладное сообщение*- блок данных, которым обмениваются программы сетевых приложений с протоколами транспортного уровня.

*ARP*- Address Resolution Protocol- протокол используется для определения соответствия IP-адресов и Ethernet-адресов.

Структура стека протоколов зависит от программ прикладного уровня. При работе с такими программами, как **FTP** или **Telnet**, образуется стек с использованием модуля **TCP/IP**:



При работе с прикладными программами, использующими транспортный протокол **UDP**, программные средства **NetWork File System(NFS)**, используется другой стек, где вместо модуля **TCP** будет использоваться модуль **UDP**:



При обслуживании блочных потоков данных модули **TCP**, **UDP** и драйвер **ENET** работают как мультиплексоры, т.е. перенаправляют данные с одного входа на несколько выходов и/или наоборот, с многих входов на один выход.

Так драйвер **ENET** может направить кадр либо модулю **IP**, либо **ARP**, в зависимости от значения поля "тип" в заголовке кадра. Модуль **IP** может направить IP-пакет либо модулю **TCP**, либо **UDP**, что определяется полем "протокол" в заголовке пакета.

Получатель **UDP**-датаграммы или **TCP**-сообщения определяется на основании значения поля "порт" в заголовке датаграммы или сообщения.

Все указанные выше значения прописываются модулями в заголовке сообщения на отправляющем компьютере. Так как схема протоколов- это дерево, то к его корню ведет только один путь, при прохождении которого каждый модуль добавляет свои данные в заголовок блока. Машина, принявшая пакет, осуществляет демультиплексирование в соответствии с этими отметками.

Технология **Internet** поддерживает разные физические среды, из которых самой распространенной является технология **Ethernet**. При использовании **Ethernet** и IP-пакета каждая машина имеет как минимум один адрес **Ethernet** и один IP-адрес. Собственно

Ethernet-адрес имеет несколько интерфейсов, то это автоматически означает, что каждому интерфейсу будет назначен свой Ethernet-адрес. IP-адрес назначается для каждого драйвера сетевого интерфейса. Грубо говоря, каждой сетевой карте Ethernet соответствуют один Ethernet-адрес и один IP-адрес. IP-адрес уникален в рамках всего Internet.

## **2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ**

### **2.1 Лабораторная работа №1 (2 часа)**

**Тема:** «Элементы рабочего окна в Windows. Технология работы с окнами в Windows»

**Цель работы:** приобретение навыков работать с окнами и объектами Windows; производить настройку рабочей среды Windows.

**Задачи работы:**

- *Получить навыки работы в операционной системе Windows.*
- *Изучить работу с окнами графического интерфейса.*
- *Уметь изменять параметры Рабочего стола – фон, рисунок, заставку и т.д.*

**Описание (ход) работы:**

Операционные системы семейства Windows являются наиболее распространенными ОС, которые установлены в домашних и офисных ПК.

Графическая оболочка ОС Windows обеспечивает взаимодействие пользователя с компьютером в форме диалога с использованием ввода и вывода на экран дисплея графической информации, управления программами с помощью пиктограмм, меню, окон, панелей (управления, задач, инструментов) и других элементов управления.

Основными элементами графического интерфейса Windows являются: Рабочий стол, Панель задач с кнопкой Пуск. Так как в Windows применен графический пользовательский интерфейса, то основным устройством управления программами является манипулятор мыши.

**Указатель мыши** – графический элемент, который передвигается по экрану синхронно с перемещением самого манипулятора по поверхности стола. Как правило, указатель мыши используется для выбора команд меню, перемещения объектов, выделения фрагментов текста и т.д.

Вид указателя мыши на экране зависит от выполняемой операции:

- при выборе объекта указатель имеет вид стрелки, направленной под углом вверх;
- в окне документа указатель напоминает английскую букву I;
- если операционная система занята выполнением каких-либо операций и не может работать с пользователем, указатель имеет форму песочных часов.

Для того чтобы указать объект, представленный на экране, необходимо переместить мышь так, чтобы острие стрелки указателя разместилось поверх этого объекта. Чтобы выбрать объект на экране, необходимо установить на него указатель, а затем нажать левую кнопку мыши.

**Щелчок** (кратковременное нажатие кнопки) – одна из основных операций мыши. Для ее выполнения следует быстро нажать и отпустить кнопку мыши. Щелчком выделяют объект в окне программы, выбирают нужную команду в меню и т.д.

При щелчке правой кнопкой мыши на элементах Windows отображается контекстное меню, кроме того, можно получить справку по элементам диалогового окна.

**Двойной щелчок** производится, когда указатель мыши установлен на определенном объекте и используется, как правило, для открытия файла.

**Перетаскивание объекта (Drag and Drop)** (перетаскивание, транспортировка, перемещение) – это операция, в результате выполнения которой изменяется местоположения объекта.

Для ее выполнения надо поместить указатель на нужном объекте (пиктограмме, папке, файле), нажать левую кнопку мыши и, не отпуская ее, переместить мышь так, чтобы этот объект перемещался вместе с ней.

Перетаскивать объект можно и правой кнопкой мыши. После того как кнопка мыши будет отпущена, на экране появится контекстное меню с набором возможных действий.

### **Основные элементы графического интерфейса Windows**

Основными элементами графического интерфейса Windows являются:

- Рабочий стол с пиктограммами.
- Панель задач, на которой размещаются программные кнопки, индикаторы, Панель быстрого запуска.
- Главное меню (кнопка Пуск).
- Контекстное меню (отображается при щелчке правой кнопкой мыши по выбранному объекту).

### **Работа с окнами**

Окно - представляет собой область экрана, ограниченную прямоугольной рамкой. В нем отображается содержимое папки, работающая программа или документ.

Различают три варианта отображения окна на экране:

- окно стандартного размера занимает часть площади экрана. При желании можно переместить его или любую его границу в другое место экрана
- окно, развернутое на весь экран, имеет максимальный размер
- свернутое окно изображается в виде кнопки на панели задач.
- В свернутом окне программа продолжает выполняться. Чтобы открыть свернутое окно или свернуть уже открытое, нажмите кнопку окна на панели задач.

Окна можно классифицировать по типу:

- окно программы (окно папки)
- окно документа
- окно диалога
- окно справки

## **Окна программ**

Окна программ – это окна, в которых отображаются программы.

Операции над окнами:

- Открыть и закрыть окно программы
- Переместить
- Изменить размеры окон
- Упорядочить окна на экране

Элементы окна программы:

- строка заголовка (слева – системное меню, справа – кнопки переключения режимов отображения на экране)
- строка меню
- панель инструментов
- рабочая область
- полосы прокруток
- строка состояния

## **Диалоговые окна**

Диалоговые окна в Windows используется для задания параметров и настроек ОС, оборудования и программ.

**Основные элементы окна диалога:**

- Вкладка
- Кнопка
- Переключатель
- Текстовое поле
- Список
- Кнопка раскрытия списка
- Флажок
- Индикатор
- Ползунок

## **Меню в Windows**

В ОС Windows применяются четыре типа меню (**меню** – это список команд, выводимых на экран и предлагаемых пользователю для выбора):

- Главное меню (открывается кнопкой Пуск)
- Стока меню в окнах приложения (все программы, входящие в стандартный пакет поставки Windows, имеют строку меню)
- Системное меню в окнах приложения (для изменения размеров окна и его положения)
- Контекстное меню

## **Работа с файлами**

Все файлы, документы и программы в Windows хранятся в папках. В электронной папке, как правило, хранят файлы, сгруппированные по какому-либо признаку, типу и другие папки.

**Папка** – это контейнер для программ и файлов в графических интерфейсах пользователя, отображаемый на экране с помощью значка, имеющего вид канцелярской папки. Windows предоставляет средства для управления файлами и папками.

К таким средствам относятся программа *Проводник* и окно *Мой компьютер*. Приложение Проводник является главным инструментом Windows для просмотра файлов и папок, хранящихся на жестких и гибких дисках и других носителях информации.

Проводник отображает иерархическую структуру файлов, папок и дисков на ПК. В левой части проводника Windows использует иерархическое представление папок, файлов и других ресурсов, подключенных к компьютеру или сети.

*Мой компьютер* – программа, используемая для работы с файлами и папками, хранящимися на дисках компьютера. *Мое сетевое окружение* – программа, используемая для работы с сетевыми ресурсами в рабочей группе.

## Манипулирование файлами и папками:

- Создание нового файла и папки
- Присвоение имени
- Переименование
- Перемещение и копирование
- Удаление
- Восстановление
- Поиск
- Просмотр и изменение свойств файлов и папок
- Создание ярлыка на рабочем столе (правой кнопкой мыши, с проводника, с главного меню, с папки мой компьютер)

### Задание 1

Выполните стандартные действия с окнами:

1. Переместите окно.
2. Измените размеры окна.
3. Измените размеры нескольких окон и расположите их в различных местах экрана (мозаикой).
4. Раскройте из строки меню любое меню со списком команд.
5. Просмотрите содержимое окна с помощью полос прокрутки.
6. Сверните окно в кнопку на *Панели задач*.
7. Разверните окно, расположенное на *Панели задач*.
8. Расположите окна на рабочем столе различными способами.
9. Закройте все раскрытие на *Рабочем столе* окна.

## Технология работы

1. Раскройте окна разного типа.
2. Переместите окно с помощью мыши:
  - установите указатель мыши на заголовок окна;
  - нажмите левую кнопку мыши и, не отпуская, переместите окно.
3. Измените размеры окна следующими вариантами:

### Вариант 1

- щелкните левой кнопкой мыши на кнопке **Развернуть**
- как только окно распахнется на весь экран, кнопка **Развернуть** будет заменена кнопкой **Восстановить**, которая позволяет Восстановить прежние размеры окна.

### Вариант 2

- подведите указатель мыши к границе окна. Форма указателя изменится, он превратится в двунаправленную стрелку, которая показывает направление, в котором возможно изменение размеров окна;
  - нажмите левую кнопку мыши и, не отпуская ее, переместите указатель — Windows покажет новый контур окна;
  - перемещайте указатель, пока контур не примет желаемый размер.
4. Перемещая окна и изменения их размеры, расположите окна на рабочем столе мозаикой;
  5. Раскройте из **строки меню** любое меню со списком команд:

- выберите окно, содержащее строку меню;
- подведите указатель мыши к пункту меню, Windows выделит его;
- щелкните левой кнопкой мыши, появится Меню со списком команд;
- подведите указатель мыши к необходимой команде, Windows выделит ее синим цветом;
- для того чтобы отказаться от выполнения выбранной команды, щелкните левой кнопкой мыши за областью этого меню.

**Внимание!** Если Windows не может выполнить некоторую команду в данных условиях, то эта команда является неактивной и написана блеклым шрифтом. Активная команда выделяется ярким цветом, и ее можно выполнить.

6. Прокрутите экран в границах окна, используя команды управления полосой прокрутки из таблицы 1:

Таблица 1. Команды для управления вертикальной полосой прокрутки

Команда	Результат выполнения команды
Щелкнуть на верхней стрелке	Перемещение содержимого окна на одну строку вверх
Щелкнуть на нижней стрелке	Перемещение содержимого окна на одну строку вниз
Щелкнуть на поле полосы над ползунком	Прокрутка содержимого окна на одну страницу вверх
Щелкнуть на поле полосы под ползунком	Прокрутка содержимого окна на одну страницу вниз
Перетащить ползунок вверх (вниз)	Прокрутка содержимого окна на часть, соответствующую перемещению ползунка относительно всей полосы

7. Сверните окно в кнопку:

- щелкните на кнопке **Свернуть**. Окно свернется в кнопку на панели в нижней части экрана;
- этот же результат может быть получен, если использовать команду **контекстного меню** - щелкните правой кнопкой мыши в поле заголовка окна и выполните команду **Свернуть**;
- сверните в кнопки все раскрытие окна.

**Внимание!** Для того чтобы *Свернуть сразу все окна, достаточно нажать комбинацию клавиш *CTRL + D**

8. Разверните окно:

- щелкните на кнопке свернутого окна, окно развернется на экране;
- разверните все окна, расположенные на **Панели задач**.

9. Расположение окон на рабочем столе различными способами:

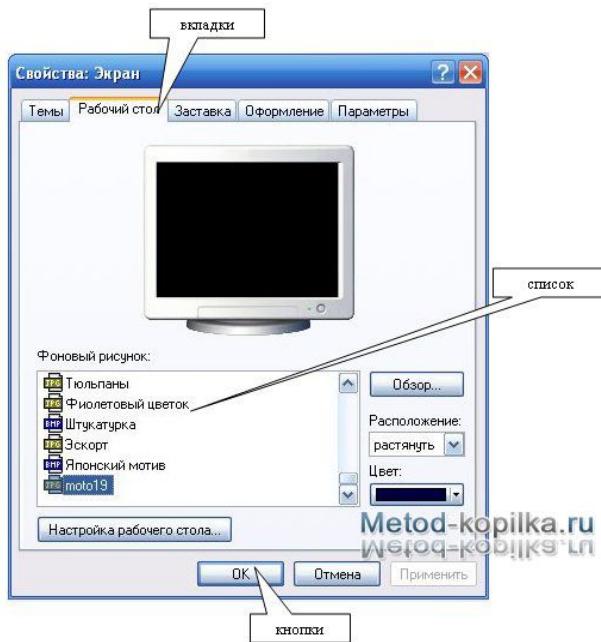
- щелкните на **Панели задач** правой кнопкой мыши, откроется контекстное меню;
- выберите опцию **Каскад**;
- посмотрите результат;
- щелкните на **Панели задач** правой кнопкой мыши, откроется контекстное меню;
- выберите опцию **Сверху вниз**;
- посмотрите результат;
- щелкните на **Панели задач** правой кнопкой мыши, откроется контекстное меню;
- выберите опцию **Слева направо**;
- посмотрите результат.

10. Закройте все раскрытие на экране окна, используя разные способы:

- кнопку **Закрыть** окно;
- команду **Закрыть** контекстного меню;
- команду **Выход** меню **Файл**;
- Нажать кнопки **Alt + F4**.

Задание 2

Измените параметры Рабочего стола: фон и рисунок, цветовую схему, заставку.



### Технология работы

#### 1. Измените параметр Рабочего стола — Фон:

- вызовите контекстное меню для объекта **Рабочий стол**, щелкнув правой кнопкой мыши на любом свободном месте **Рабочего стола**, но не на объектах стола, иначе будет вызвано контекстное меню, связанное с данными объектами;
- в раскрывшемся меню выберите команду **Свойства**. Перед вами появится окно **Свойства: Экран**;
- выберите вкладку **Рабочий стол**, щелкнув левой кнопкой мыши;
- просмотрите с помощью полосы прокрутки список графических файлов в окне **Фоновый рисунок**;
- щелчком мыши на значке графического файла выберите рисунок;
- выберите из списка способ расположения рисунка на Рабочем столе щелчком на кнопке **Расположение: по центру; замостить; растянуть**.
- в окне предварительного просмотра посмотрите, как будет выглядеть Рабочий стол с выбранным вами рисунком;
- щелкните на кнопке **Применить**, чтобы увидеть результат изменений на экране (окно **Свойства: Экран** останется раскрытым).

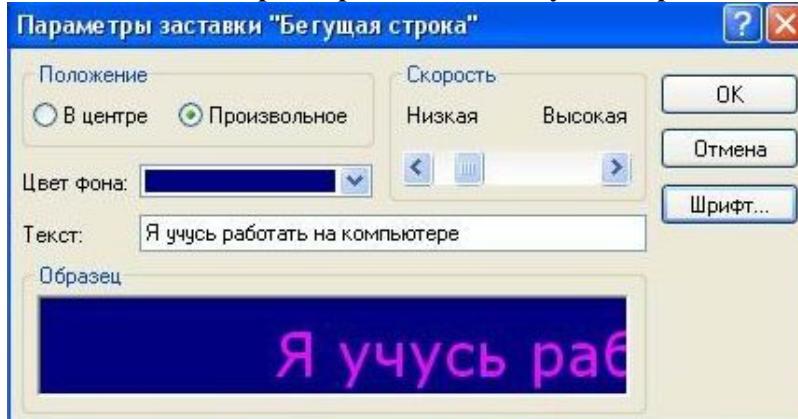
#### 2. Измените параметр Рабочего стола — цветовая схема экрана, выполнив следующие действия:

- выберите вкладку **Оформление** щелчком левой кнопки мыши;
- просмотрите в окне **Цветовая схема** предоставляемый список: *оливковая; серебристая; стандартная (голубая)*;
- выберите желаемое сочетание цветов, щелкнув мышью на названии схемы;
- в окне предварительного просмотра посмотрите, как будет выглядеть Рабочий стол с выбранной вами цветовой схемой;
- щелкните на кнопке **Применить**.

#### 3. Измените параметр Рабочий стол — заставка экрана. Во вкладке **Заставка** вы можете выбрать экранную заставку, которая появляется на экране, если в течение некоторого времени вы не пользуетесь клавиатурой или мышью. При нажатии любой клавиши на клавиатуре или перемещении мыши на экране восстановится прежнее изображение **Рабочего стола**. Выполните следующие действия:

- в окне **Свойства: Экран** выберите вкладку **Заставка** щелчком левой кнопки мыши;

- в раскрывающемся списке **Заставка** выберите название заставки, которую вы увидите в окне предварительного просмотра;
  - для полного представления заставки нажмите кнопку **Просмотр**;
  - щелчком левой кнопки мыши вернитесь из **Просмотра** в окно **Свойства: Экран**;
  - щелкните на кнопке **Применить**.
4. Каждая заставка характеризуется своими параметрами. Измените, например, параметры заставки **Бегущая строка**:
- выберите заставку **Бегущая строка**;
  - вызовите окно **Параметры заставки Бегущая строка** щелчком на кнопке;



- выберите **Положение** текста;
- выберите **Цвет фона**;
- в поле ввода «**Текст**» введите любой текст;
- измените шрифт текста, щелкнув по кнопке **Шрифт**;
- для установки скорости движения текста на экране переместите с помощью мыши ползунок **Скорость** на линейке;
- щелкните на кнопке **OK** в окне **Параметры заставки «Бегущая строка»**;
- щелкните на кнопке **OK** в окне **Свойства: Экран**.

## 2.2 Лабораторная работа №2 (2 часа)

**Тема:** «Технология работы с файлами и папками в Windows»

**Цель работы:** изучение приемов работы с программой Проводник

**Описание (ход) работы:**

### 1. Краткие теоретические сведения

В составе операционной системы Windows XP имеется диспетчер файлов Windows Explorer (Проводник), к помощи которого целесообразно прибегать в случае, если необходимо выполнить серию операций с папками и файлами.

Исполняемый файл диспетчера Проводник именован как Explorer.exe и записан в папке Windows.

Окно Проводника напоминает окно папки, но имеет две панели (рис. 1).

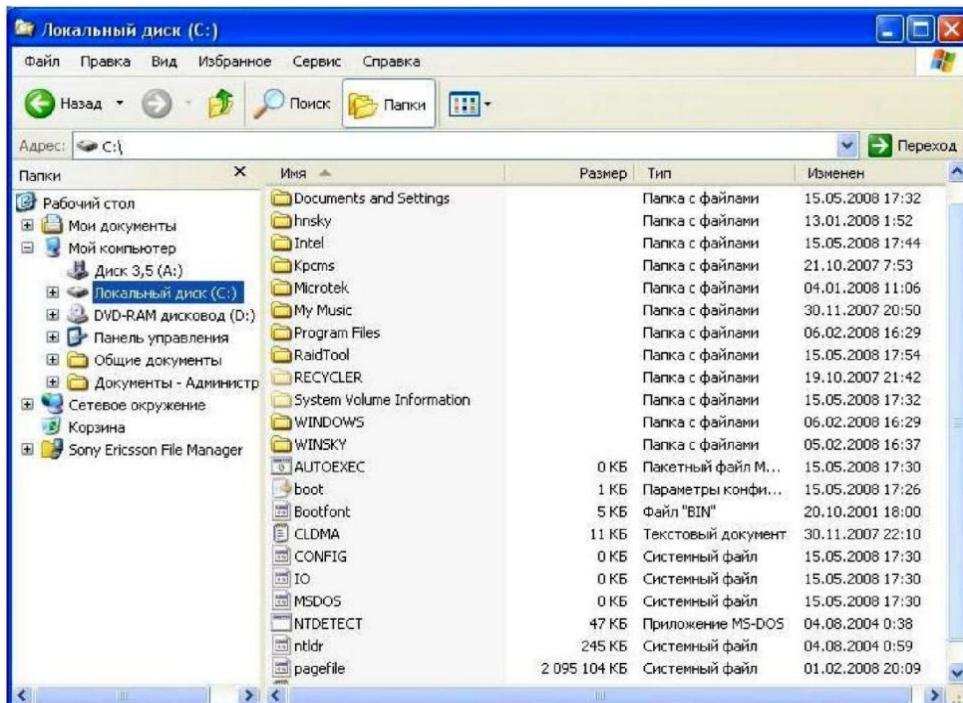


Рис. 1

На левой панели показывается иерархия папок в точности в том виде, в котором она поддерживается Windows.

На правой панели отображается содержимое папки, которая на левой панели выделена (открыта). Сразу после активации Проводника на левой панели оказывается открытой главная папка диска С:, либо папка, применительно к которой Проводник был запущен.

Одна из панелей Проводника пассивна, другая активна. Активная панель отличается тем, что имеет выделенный объект. Панель активизируется, если щелкнуть по одному из представленных на ней объектов или по любому ее месту.

Для смены активности можно пользоваться клавишей F6.

Система меню Проводника во многом повторяет систему меню окна папки, но содержит дополнительное меню Сервис. Настраивается меню Проводника подобно меню папки. Меню Файл Проводника является обобщенным контекстным меню объекта, если объект выделен в правой панели.

Папки в дереве левой панели можно сворачивать и разворачивать. Слева от папки имеется квадратик. Если в нем «+», то папка свернута – содержащиеся в ней папки в иерархии не показаны. Если стоит «-», то папка развернута.

Открытая папка имеет соответствующую пиктограмму.

Развернуть или свернуть папку можно щелкнув мышью на расположеннем слева квадрате, либо дважды щелкнуть мышью по самой папке.

Дополнительные возможности Проводника:

- иерархия папок на левой панели окна диспетчера позволяет проследить взаимосвязи между папками и открыть любую из них;
- новые окна диспетчера не открываются – все операции и изменения выполняются в уже открытом окне.

Способ и вариация представления объектов на правой панели Проводника задается командами меню Вид, совпадающим с ними командами подменю Вид контекстного меню свободного поля папки, а также инструментальными кнопками.

В окнах папок программы Проводник объекты можно сортировать, то есть упорядочивать по некоторому критерию. Для сортировки используется подменю «Упорядочить значки» из

меню Вид. Упорядочить файлы и папки можно по имени, по типу (в лексикографическом порядке), по размеру (в порядке возрастания), по дате создания (объекты с более поздней датой создания выносятся вперед).

Программа Проводник обеспечивает один из наиболее удобных способов управления файлами. В случае, когда требуется открыть, скопировать, переместить, удалить, переименовать или переупорядочить папки или (и) файлы, есть возможность либо воспользоваться программой Проводник, либо работать с папками и файлами непосредственно на Рабочем столе. Основное преимущество программы Проводник, по сравнению с окнами папок на Рабочем столе, состоит в том, что в программе Проводник можно одновременно работать с содержимым правой панели окна и со всей структурой файловой системы вашего компьютера – левой панели. Это значительно облегчает копирование файлов из правой панели на логический диск или в папку, находящиеся на левой панели. При этом нет необходимости открывать большое число окон, как это пришлось бы делать на Рабочем столе. Наиболее удобным приемом является открытие двух окон программы Проводник, упорядоченных по вертикали или по горизонтали.

Работа с программой Проводник ничем не отличается от работы с большинством других окон и программ, за исключением того, что вам предоставляется возможность просмотра иерархической структуры файловой системы. Вы можете щелкнуть правой кнопкой мыши по какой-либо папке и просмотреть ее меню или щелкнуть по любой пиктограмме и перетащить ее с одного места на другое.

## 2. Порядок выполнения работы

### Задание

Для успешного управления файлами на компьютере необходимо освоить следующие операции: запуск Проводника, просмотр каталога диска, просмотр содержимого папок, разворачивание и сворачивание папки, копирование, перемещение и удаление папок, файлов и приложений, поиск папок, файлов и приложений, сортировку объектов.

В ходе лабораторной работы выполните следующие задания:

- 1) запустите приложение программы Проводник каждым из известных вам способов;
- 2) выполните с окном программы Проводник (рис. 1) стандартные операции:
  - уменьшение размеров окна;
  - перемещение окна в пределах Рабочего стола;
  - изменение соотношения собственных панелей программы Проводник(панель просмотр);
  - сворачивание окна программы Проводник;
  - разворачивание окна программы Проводник;
  - работа программы Проводник в режиме двух окон.
- 3) изучите состав меню окна Проводник;
- 4) ознакомьтесь с названиями кнопок Панели инструментов, указывая поочередно указателем мыши на каждую из них. Названия кнопок поочередно появляются под кнопками Панели инструментов;
- 5) расположите объекты окна по имени, типу, размеру или дате. Найдите приложение Explorer.exe;
- 6) ознакомьтесь с содержанием рабочего логического диска, просмотрев все ветви на соответствующей панели программы Проводник, и получите информацию о свойствах логического диска;
- 7) создайте одним из способов на сервере в ресурсе Public персональную рабочую папку, именованную своей фамилией;
- 8) создайте в своей персональной папке ветвь из трех вложенных подпапок с именами «1», «2» и «3»;

- 9) скопируйте все файлы рисунков из системной папки Windows (или Windows\Web\Wallpaper) в папку «3»;
- 10) переместите в папку «2» все файлы из папки «3», размер которых не превышает 45 Кбайт, используя при этом разные способы перемещения;
- 11) переименуйте папку «1», присвоив ей ваше собственное имя;
- 12) создайте в переименованной папке копию папки C:\Windows\Cursors различными способами;
- 13) присвойте полученной папке имя «Курс 1»;
- 14) откройте полученную папку разными способами;
- 15) предъявите преподавателю полученные результаты;
- 16) удалите созданные папки, используя различные способы;
- 17) восстановите удаленные объекты и закройте окно «Проводник».

#### Технология работы

Запуск диспетчера файлов Проводник можно выполнить несколькими способами:

- командой Проводник из подменю Программы главного меню системы;
- командой Проводник из контекстного меню некоторой папки (Мой компьютер, Сетевое окружение) в окне папки или в диалоговом окне;
- двойным щелчком мыши при нажатой клавише Shift по предварительно выделенной папке в окне папки;
- командой Проводник из меню Файл окна папки, предварительно выделив одну из папок в этом окне;
- командой Проводник из меню, открываемого щелчком правой кнопки мыши по пиктограмме управляющего меню окна папки;
- командой Проводник из контекстного меню кнопки Пуск;
- командной строкой по команде Выполнить из главного меню системы.

Для выполнения стандартных операций с изображенным на рис. 1 окном программы Проводник используйте:

- в правом верхнем углу окна кнопки <Свернуть>, <Развернуть>, <Восстановить>, <Закрыть>;
- маркеры растяжки (двойная черная стрелка, в которую превращается курсор при помещении его на бордюр окна).

Внимание! Для создания «двуоконного режима» программы Проводник дважды откройте эту программу, щелкнув правой кнопкой мыши по Панели задач и выбрав опцию Сверху вниз или Слева направо.

Установите указатель мыши на меню Файл, нажмите и отпустите левую кнопку мыши. Ознакомьтесь с командами меню Файл и переведите указатель мыши на очередное меню. Запишите основные команды меню Правка, Вид, Сервис.

Вызовите из меню Вид команду Панель инструментов (установите флажок).

Выполните сортировку объектов при помощи команды меню Вид,

Упорядочить значки.

Для поиска объекта вызовите команду Сервис, Поиск, Файлы и папки:

- в окне «Найти» в закладках Имя и размещение, Дата изменения, Дополнительно сделайте соответствующие установки;
- для получения информации об объекте используйте:
- команду меню Файл, Свойства;
- кнопку Панели инструментов <Свойства>;
- контекстное меню (правую кнопку мыши).

Для создания папки используйте следующие способы:

- команду меню Файл, Создать;

- контекстное меню (щелкните правой кнопкой мыши в пространстве правого окна программы Проводник).

Для копирования используйте следующие способы:

- команды меню Правка, Копировать и Правка, Вставить;
- кнопки панели управления программы Проводник <Копировать в буфер>, <Вставить из буфера>;
- контекстное меню (правая кнопка мыши);
- метод перетаскивания drag and drop (левая кнопка мыши при нажатой клавише <Ctrl>).

Наиболее удобно выполняется в двухоконном режиме.

Внимание! Отменить копирование объекта можно командой меню Правка, Отменить копирование.

Поместить объект в папку можно любым способом копирования.

Для переименования используйте следующие способы:

- команду меню Файл, Переименовать;
- контекстное меню (правая кнопка мыши).

Для перемещения я используйте следующие способы:

- команды меню Правка, Вырезать и Правка, Копировать;
- кнопки панели управления программы Проводник <Копировать в буфер>, <Вставить из буфера>;
- контекстное меню (правая кнопка мыши);
- метод перетаскивания drag and drop (левая кнопка мыши).

Внимание! Отменить перемещение объекта можно командой меню Правка, Отменить перемещение.

Чтобы открыть папку, используйте:

- двойной щелчок левой кнопки мыши;
- клавишу <Enter>;
- команду меню Файл, Открыть;
- контекстное меню (правую кнопку мыши).

Чтобы удалить объект (выбросить в корзину), используйте:

- клавишу <Delete>;
- команду меню Файл, Удалить;
- кнопку Панели инструментов <Удалить>;
- контекстное меню (правую кнопку мыши).

Внимание! Отменить удаление объекта можно командой меню Правка, Отменить удаление.

Для того чтобы удалить объект окончательно, удерживайте нажатой клавишу <Shift>.

Для восстановления удаленного объекта (выброшенного в корзину) запустите программу Корзина одним из трех способов:

- командой меню Файл, Восстановить;
- командой меню Правка, Отменить удаление;
- через контекстное меню (с помощью правой кнопки мыши).

Внимание! При копировании, перемещении, удалении папки или файла значок выбранного объекта должен быть активизирован. Если заданная команда выполняется с помощью контекстного меню, курсор должен располагаться в области значка выбранного объекта.

В Windows предусмотрены средства, позволяющие удобно организовать работу с документами и программами. Файлы можно не только создавать и хранить в папках, копировать, переименовывать и перемещать их из одной папки в другую — система позволяет также осуществлять поиск файлов и папок.

Каждый файл и папка имеет контекстное меню, которое вызывается при нажатии правой кнопки мыши. Контекстные меню содержат команды, наиболее часто употребляющиеся при работе с объектом.

Для работы с папками и файлами в Windows используются **Мой компьютер** (значок запуска расположен на рабочем столе) и **Проводник** (запускается из меню запуска программ) меню которых также содержит команды работы с файлами и папками.

**Мой компьютер** удобен для просмотра содержимого одной папки или диска. По двойному щелчку мыши на значке «Мой компьютер» на экране в новом окне появляются доступные на компьютере диски. Теперь, если дважды щелкнуть значок диска, в окне будут показаны имеющиеся на этом диске папки. Для просмотра содержимого папки следует дважды щелкнуть ее значок.

Если вам удобно просматривать файлы в виде иерархической структуры, пользуйтесь приложением **Проводник** Windows. Чтобы не открывать диски и папки в отдельных окнах, этот режим позволяет перемещаться по ним в одном окне. Левая область проводника Windows содержит список дисков и каталогов, а справа отображается содержимое текущего объекта. Меню Вид позволяет изменить вид значков в правой области.

#### **Задание 1. Знакомство с папкой *Мой компьютер*.**

- Откройте *Мой компьютер*, выполнив двойной щелчок мышью на соответствующем значке рабочего стола.
- Ознакомьтесь с элементами окна *Мой компьютер*: областью для выбора папок, командами горизонтального меню, панелью инструментов, полем со списком для выбора папки в качестве текущей.

#### **Задание 2. Переход по дискам и папкам в окне *Мой компьютер*.**

- Переместитесь в корень дерева файлов (в папку *Рабочий стол- Desktop*)
- Разверните содержимое папки *Мой компьютер*, щелкнув на его значке.
- Разверните содержимое диска C:\ щелкнув на значке диск (C:)
- Перемещаясь по дереву файлов окна найдите папку Temp.
- Раскройте содержимое папки Temp, щелкнув по ней два раза

#### **Задание 3. Знакомство с приложением *Проводник (Explorer)***

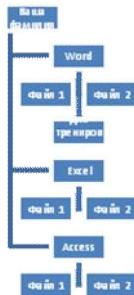
- Запустите программу *Проводник*, выполнив команду Главного меню *Программы-> Проводник*.
- Ознакомьтесь с элементами окна *Проводника*: областью для выбора папок и областью для просмотра содержимого выбранной папки, панелью инструментов, полем со списком для выбора папки в качестве текущей, горизонтальным меню.

#### **Задание 4. Переход по дискам и папкам в *Проводнике*.**

- Переместитесь в корень дерева файлов (в папку *Рабочий стол- Desktop*)
- Сверните все открытые папки в левой области окна проводника, щелкнув на информационных значках (знак -).
- Разверните содержимое папки *Мой компьютер*, щелкнув на значке +
- Разверните содержимое диска C:\ щелкнув на значке +
- Перемещаясь по дереву файлов в левой области окна, найдите папку Temp.
- Раскройте содержимое папки Temp, щелкнув по ней два раза в дереве файлов.

#### **Задание 5. Создание новой папки на диске C:\.**

- Перемещаясь по дереву файлов посредством **Моего компьютера** или **Проводника** в окне найдите C:\Мои Документы на котором надо создать новую папку и раскройте его двойным щелчком мыши.
- Создайте новую папку на диске C:\Мои Документы. Для этого выполните команду меню **Файл/Создать/Папка (File/New/Folder)**. После чего должен появится значок новой папки. Введите название новой папки в поле названия папки (Вашу фамилию) и нажмите клавишу ENTER. В этой папке будут храниться все Ваши файлы.
- Создать в Вашей папке следующую структуру папок:



Задание 6. Создание нового текстового файла в Вашей папке.

- Раскройте Вашу папку, щелкнув на ней два раза.
- Создайте новый текстовый файл и разместите его в Вашей папке. Для этого выполните команду меню **Файл/Создать/Текстовый документ (File/New/Text Document)**. Выберите из списка типов файлов *Текстовый документ*. На правой панели должен появится значок нового файла с выделенным названием. Введите название нового файла и нажмите клавишу **ENTER**.
- Откройте двойным щелчком текстовый Документ.
- В запущенном стандартном приложении *Блокнот* введите текст:

**"Для ввода прописных букв используется одновременное нажатие клавиши Shift, для ввода длинной последовательности прописных букв клавиатуру можно переключить с помощью клавиши Caps Lock. Для переключения между русскими и английскими символами используется индикатор языка, если он отображен на панели задач, или специальная комбинация клавиш, установленная на Вашем компьютере. Обычно это комбинация клавиш Alt+Shift или Ctrl+Shift.**

**Ваша фамилия, номер группы/подгруппы"**

- Сохраните набранный текст в текущем файле, выполнив команду **Файл/Сохранить как** в нужной вам директории.
- Закройте приложение *Блокнот*.

**Задание 7. Копирование и перемещение файлов с помощью буфера обмена**

Способы выполнения операций с файлами

Операция	Клавиши клавиатур	Команды меню	Кнопки панели инструментов
Копировать в	Ctrl+C	Копировать	Копировать
Переместить в	Ctrl+X	Вырезать	Вырезать
Вставить из	Ctrl+V	Вставить	Вставить
Удалить	Delete	Удалить	Удалить
Поиск	Ctrl+F	Вызов контекстного меню	Поиск
Выделение	Ctrl+A	Выделение всего	Правка – Выделить всё
Закрытие активного окна	Alt+F4	Закрытие активного	

**Задание 8. Копирование и перемещение файлов и папок перетаскиванием мышью.**

Найдите в папке *Мой компьютер* или в окне Проводника нужный файл или папку, подлежащий копированию или перемещению.

Убедитесь, что место, куда предполагается перетащить объект, присутствует на экране.

Перетащите объект в нужное место с помощью мыши.

Результат этого действия зависит от типа объекта и выбранного места. Если при перетаскивании объекта удерживать не левую, а правую кнопку мыши, на экране появится меню с набором возможных действий.

Перетаскивание файла в папку, находящуюся на том же диске, приводит к перемещению папки. Если вторая папка находится на другом диске, объект будет скопирован.

Выбрать нужный тип действия можно с помощью указанных ниже клавиш.

Чтобы переместить файл, удерживайте клавишу SHIFT.

Чтобы скопировать файл, удерживайте клавишу CTRL.

Чтобы создать ярлык к файлу, удерживайте клавиши CTRL+SHIFT.

- Переместите текстовый файл, созданный в задании 6 из Вашей папки в папку .../Мои Документы/ваша фамилия/Word/Для тренировок.

**Задание 9. Поиск документа в Моём Компьютере и в определённой папке.**

- найдите файл proba.doc, скопируйте его в .../Мои Документы/ваша фамилия/Word/файл1

Задание 10. Копирование файла на другой диск

Скопируйте файл proba.doc из .../Мои Документы/ваша фамилия/Word/файл1 в .../Мои Документы/ваша фамилия/Word/файл2

**Задание 11. Переименование файла.**

• Переименуйте текстовый файл из .../Мои Документы/ваша фамилия/Word/Для тренировок. Для этого необходимо щелкнуть по значку файла правой кнопкой и в контекстном меню выбрать команду **Переименовать** или выделить значок файла и выполнить команду горизонтального меню **Файл/Переименовать (File/Rename)**. Введите новое имя New.doc в выделенную рамку имени значка и нажмите клавишу Enter.

**Задание 12. Удаление файла.**

- Удалите файл New.doc Для этого щелкните по нему в правой части окна и нажмите клавишу Delete клавиатуры.

**Задание 13. Восстановление файла.**

- Восстановите удаленный файл, с помощью кнопки Отмена.

• Восстановите файл, воспользовавшись Корзиной. Откройте Корзину двойным щелчком в левой области окна. Справа появится содержимое корзины. Выделите нужный файл, выберите функцию Восстановить.

**Задание 14. Архивация файлов и папок.**

Используя встроенный в Windows архиватор, создать архивный файл Word, содержащий папку ../Word1, и поместить созданный архив в папку .../Мои Документы/ваша фамилия/Word/Для тренировок.

- Извлечь из архива Фамилия текстовый файл и поместить его в папку .../Мои Документы/ваша фамилия/Word/Для тренировок.

**Задание 14. Архивация файлов и папок.**

- Скопировать папку с Вашей фамилией со всеми вложенными папками по адресу //server/Практика/номер вашей группы

## 2.3 Лабораторная работа №3 (2 часа)

**Тема:** «Технология работы с файлами и папками в Windows commander»

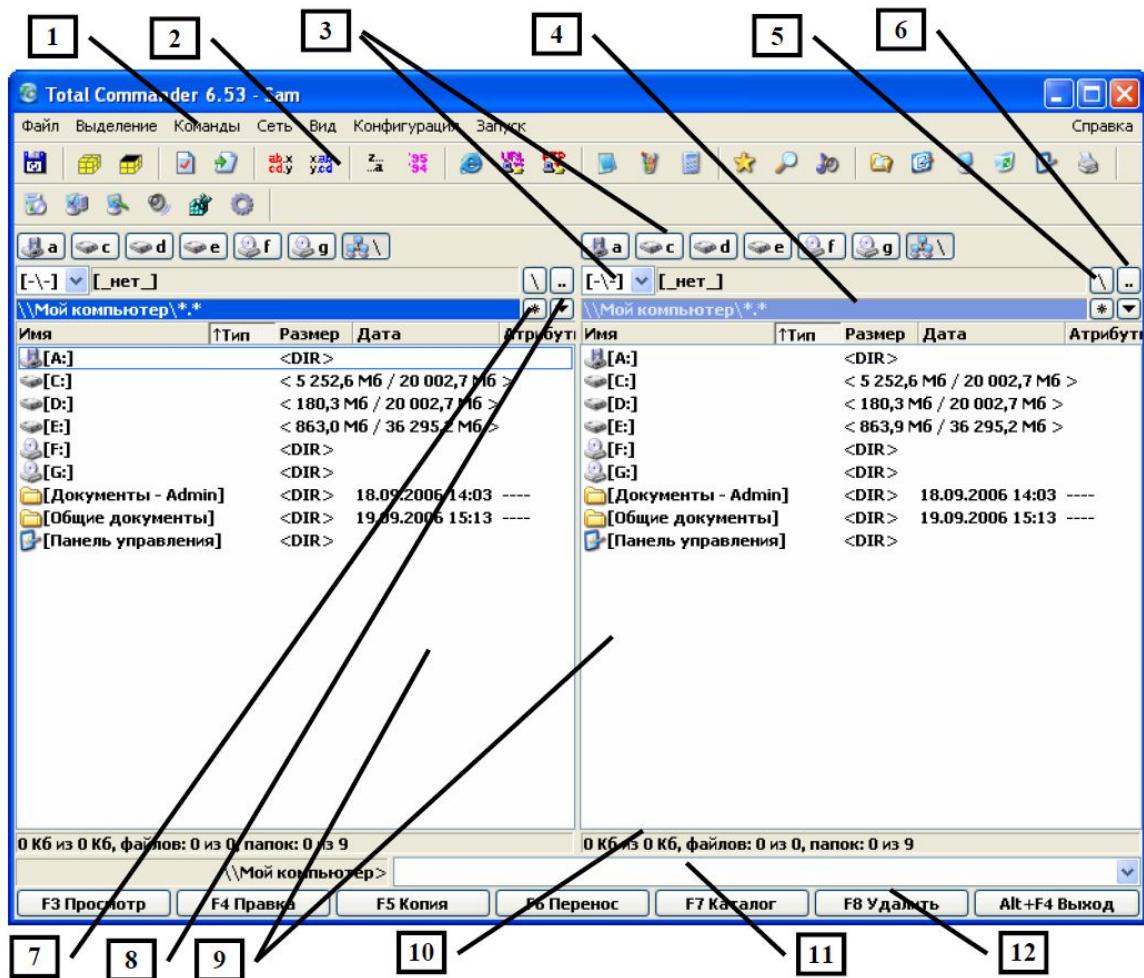
**Цель работы:** Освоить основные приемы работы с файлами и папками в Windows commander

**Перечень приборов, материалов, используемых в лабораторной работе:**

1. Компьютеры с установленным программным обеспечением
2. Методические указания к выполнению лабораторной работы
3. Рабочая тетрадь.

**Описание (ход) работы:**

Файловый менеджер **Total Commander** (TC) представляет удобный доступ к файлам и папкам, позволяет осуществлять все операции, используемые при работе с файлами и папками. Внешний вид файлового менеджера показан на рисунке 1.



- 1 – главное меню;  
 2 – панель инструментов;  
 3 – кнопки выбора дисков;  
 4 – текущий путь;  
 5 – переход в корневой каталог; 10 – информационная строка;  
 6 – переход на один уровень вверх;  
 7 – избранные каталоги;  
 8 – история;  
 9 – файловые панели;  
 11 – командная строка;  
 12 – панель функциональных клавиш.

Рисунок 1 – Файловый менеджер **Total Commander**

### Основные операции с файлами:

#### 1) Выделение файлов

Чтобы выделить файлы или каталоги, просто щёлкните по ним мышью или переместитесь на них с помощью клавиш курсора и нажмите **INSERT**. Если вы выбираете каталог, используя клавишу **ПРОБЕЛ**, показывается размер дискового пространства, занятого этим каталогом. Все описанные ниже операции доступны из меню **Выделение**.

#### 2) Выделение нескольких последовательных объектов

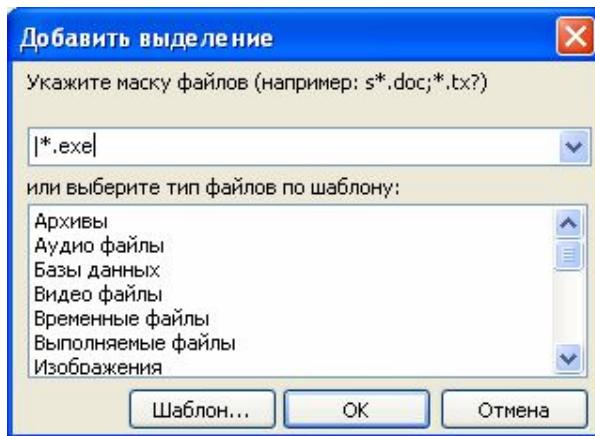
Щёлкните по первому файлу или каталогу, предназначенному для выделения. После этого нажмите клавишу **SHIFT** и, удерживая её, щёлкните левой кнопкой мыши на последнем объекте, который хотите выделить.

### 3) Выделение нескольких несмежных объектов

Выделяйте левой кнопкой мыши любые несмежные файлы или каталоги, держа при этом нажатой клавишу **CTRL** (снятие выделения с отдельного файла/каталога выполняется точно так же).

### 4) Выделение и отмена выделения определённых типов файлов

Нажмите клавишу **Num+** (или **Num-**) или выберите одну из команд выделения **Выделить группу** → **Снять выделение группы** в меню **Выделение**. Затем в появившемся диалоге введите нужный вам тип файла (например, `*.txt`). Вы можете также указать несколько типов файлов, и даже те типы файлов, которые не должны быть выделены. Их следует отделить символом вертикальной черты `"|"`.



Пример 1: `w*.*|*.bak *.old` Выделить все файлы, которые начинаются с `w` и не заканчиваются `.bak` или `.old`.

Пример 2: `|*.exe` Выделить все файлы, кроме программ.

### 5) Выделить всё / Снять всё выделение

Нажмите сочетание клавиш **CTRL+Num+** (или, соответственно, **CTRL+Num-**) или выберите команду **Выделить всё** / **Снять всё выделение** в меню **Выделение**. Выделить всё содержимое файловой панели можно также при помощи комбинации клавиш **CTRL+A**.

### 6) Выделить по расширению / Снять выделение по расширению

Выбрав файл с нужным вам расширением, нажмите сочетание клавиш **ALT+Num+** (или **ALT+Num-**), чтобы выделить все файлы с таким же расширением в текущей панели или, соответственно, снять выделение с этих файлов.

### 7) Инвертировать выделение

Эта команда отметит все файлы в исходном каталоге, которые не были отмечены, и снимет выделение у ранее отмеченных файлов. Для вызова команды нажмите клавишу **Num\*** (умножение).

### 8) Изучить самостоятельно пункты:

Сохранить выделение, Восстановить выделение, Сохранить выделение в файл, Загрузить выделение из файла. Определить горячие клавиши соответствующих команд.

### Обновление содержимого панели:

Нажмите **CTRL+R**. При этом обновится текущая панель. Эту операцию следует выполнить, например, после смены гибкого диска, чтобы обновить содержимое файловой панели.

### Просмотр содержимого файлов:

Выберите файлы, которые хотите просмотреть, и нажмите **F3**. Встроенная программа просмотра файлов **Lister** показывает содержимое файла под курсором.

Интегрированный просмотрщик файлов позволяет просматривать файлы в текстовом, двоичном или шестнадцатеричном формате, Unicode-файлы и HTML-страницы, файлы растровой графики, мультимедиа и файлы RTF. Быстрый просмотр выделенного файла можно выполнить при помощи нажатия **Ctrl + Q**.

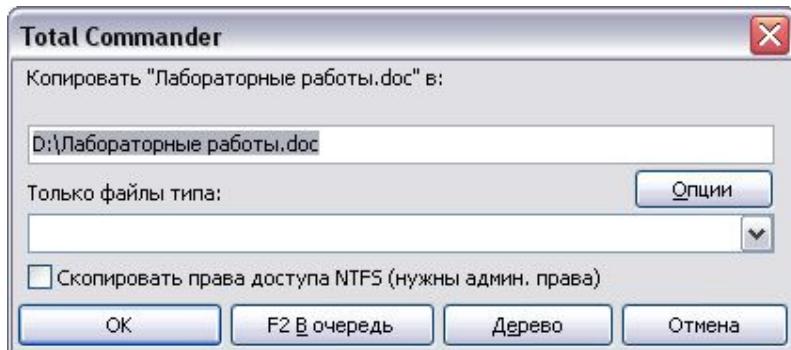
### Правка (F4)

Поместите курсор на файл, который вы хотите редактировать, и нажмите **F4**. При этом запускается выбранный вами в диалоге настройки редактор, а в него загружается выбранный файл. По умолчанию запускается стандартный Блокнот Windows. Он может работать только с текстовыми файлами ограниченного размера.

Если вы хотите редактировать файлы других типов, просто дважды щёлкните на файле или нажмите **ENTER**. Запустится программа, ассоциированная с файлом.

### Копирование (F5) и Перемещение (F6)

Эта команда копирует файлы и целые каталоги из исходного каталога в каталог на другой панели. Выделите файлы, которые вы хотите скопировать, и нажмите **F5**. При этом откроется диалоговое окно, в котором вы можете ввести каталог назначения и маску файлов.



Чтобы скопировать файл в тот же самый каталог (под другим именем), нажмите **SHIFT+F5**. Ярлык для файла можно создать комбинацией клавиш **CTRL+SHIFT+F5**.

Кнопкой **Дерево** вы можете выбрать каталог назначения из дерева каталогов.

При нажатии кнопки **F2 В очередь** выбранные файлы будут добавлены в список последнего открытого диспетчера фоновой пересылки. Это полезно при копировании нескольких больших файлов друг за другом, что более эффективно, чем параллельное копирование их всех в фоновом режиме.

Кнопка **Опции** позволяет установить параметры для автоматического копирования. По умолчанию ТС выводит запрос о перезаписи файлов. Эта кнопка позволяет, например, установить по умолчанию "Заменить все", "Пропустить все" или "Заменить все старые".

В поле "Только файлы типа", вы можете указать, какие файлы копировать, причём это распространяется также на файлы из подкаталогов.

#### Примеры:

\*.txt \*.doc Будут копироваться только файлы .doc и .txt.  
\*.\* | \*.bak \*.old Будет копироваться всё, кроме файлов .bak и .old.  
Для перемещения файлов воспользуйтесь клавишей **F6**.

## Архивы

Если вы хотите создать новый архив и упаковать в него выделенные файлы, просто нажмите кнопку  на панели инструментов или сочетание **ALT+F5**. Откроется диалоговое окно упаковки файлов. При использовании сочетания **ALT+SHIFT+F5** файлы после упаковки будут удалены.

Если вы хотите распаковать архив под курсором (или выделенные архивы), нажмите кнопку  на панели инструментов или **ALT+F9**. После указания каталога назначения (и при необходимости – маски файлов), все файлы из архива будут распакованы.

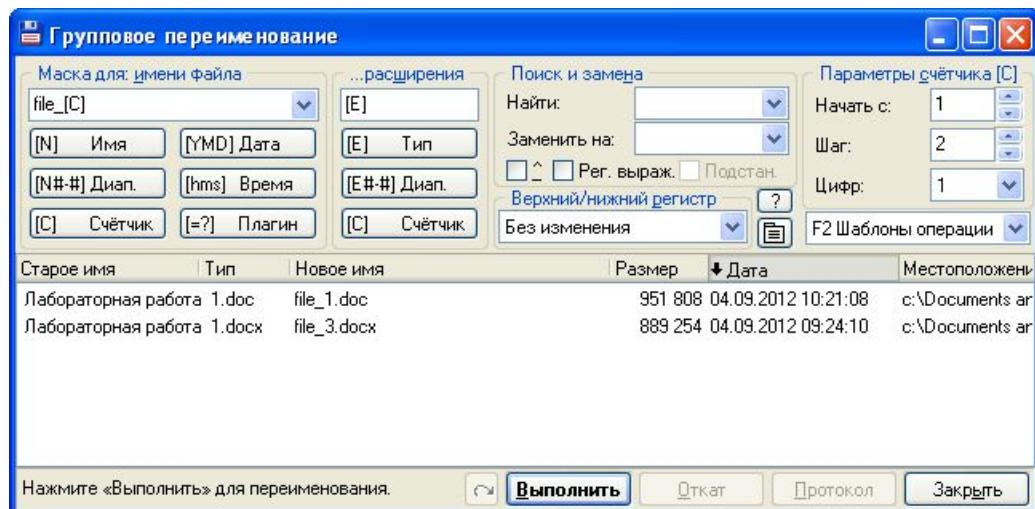
Также для выполнения данных операций можно пользоваться пунктами главного меню **Файл → Упаковать** и **Файл → Распаковать** или контекстным меню группы выделенных файлов (**Добавить в архив..**) или архива (**Извлечь файлы...**)

## Переименование файлов и каталогов

Для переименования файла или каталога под курсором нажмите правую кнопку мыши и выберите пункт **Переименовать**. Также можно воспользоваться сочетанием **SHIFT+F6**.

### Групповое переименование файлов

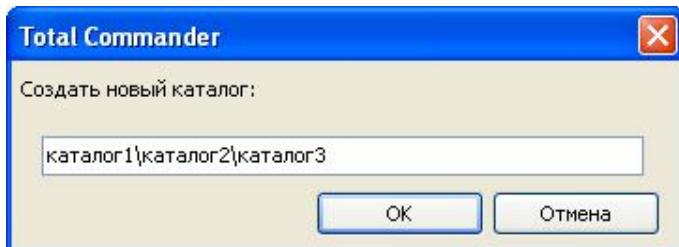
Выделите несколько файлов и выберите пункт **Файл → Групповое переименование**. Также можно воспользоваться сочетанием **CTRL+M**. В открывшемся диалоговом окне можно настроить формат имени группы файлов. Например:



После нажатия на клавишу «выполнить» файлы будут переименованы в соответствии с заданным форматом.

## Создание каталога (F7)

Эта команда создаёт новый подкаталог в исходном каталоге. После нажатия **F7** просто введите желаемое имя каталога. Можно также создавать и несколько подкаталогов за одну операцию. Просто отделите подкаталоги обратной косой чертой (обратный слэш), например:



Проверьте, что произойдет, если в поле имени каталога ввести **каталог1\каталог2\каталог3**.

### Удаление (F8)

Выделите файлы и/или каталоги, которые хотите удалить, и нажмите **F8**. После подтверждения файлы удаляются. Процесс может быть прерван в любой момент кнопкой '**Отмена**'. Для каждого непустого каталога будет запрашиваться подтверждение в дополнительном диалоговом окне. Предупреждение: все файлы И ПОДКАТАЛОГИ в этом каталоге будут удалены.

### Поиск файлов

Используйте пункт меню **Инструменты** → **Поиск файлов** или сочетание клавиш **ALT+F7**.

В открывшемся диалоговом окне можно задать маску для поиска файлов. Имена, содержащие пробелы, ДОЛЖНЫ быть помещены в двойные кавычки, например, "Письмо к Иванову.doc", иначе Total Commander искал бы каждую часть имени по отдельности. Для удаления ненужных записей из истории поиска, вы можете использовать сочетание **Shift+Del**.

#### Примеры:

**\*.ini** находит, например, **win.ini**

**Иванов** находит "Письмо к Иванову.doc"

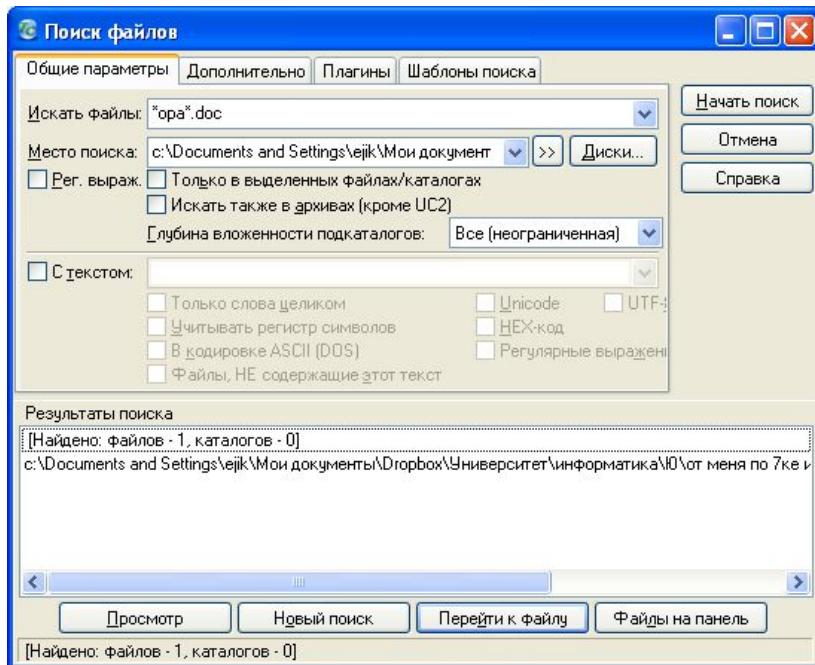
**\*.bak \*.sik \*.old** находит все файлы резервных копий с этими расширениями

**\*n.ini** теперь находит имена, которые **обязательно** содержат 'n' перед точкой.

**\*n?.ini** находит имена, которые **обязательно** содержат 'n' и ещё *ровно* один символ перед точкой. Например, *Wcmd\_eng.ini*.

**w\*.\*|\*.bak \*.old** находит файлы, которые начинаются с *w* и не заканчиваются на *.bak* или *.old*

На вкладке «Дополнительно» можно указать такие параметры поиска, как размер файла, дата создания. На вкладке «Шаблоны поиска» можно сохранить созданный вами запрос или открыть уже введённый шаблон.



<P

Ре

## Результаты поиска

Содержит список всех найденных файлов. Подробности (дата и время модификации, размер) показаны под списком из-за недостатка свободного пространства. Дважды щёлкните на файле, чтобы перейти в каталог, в котором этот файл находится. Чтобы скопировать весь список файлов в буфер обмена, щёлкните в этом окне и нажмите **Ctrl+C**. Щелчок правой кнопкой на найденном файле открывает системное контекстное меню для него.

**Просмотр** Загружает выделенный файл во внутреннюю программу просмотра (Lister).

**Новый поиск** Закрывает нижнюю часть диалога для нового поиска.

**Перейти к файлу** Если вы выбрали файл в списке, вы можете перейти в каталог, где находится этот файл, нажав "Перейти к файлу". Удерживая нажатой клавишу **Shift**, вы откроете этот каталог в новой вкладке.

**Файлы на панель** Передаёт найденные файлы в исходную файловую панель, где они могут быть скопированы, перемещены или удалены. Это работает, только когда поиск внутри архивов НЕ включён! Удерживая нажатой клавишу **Shift**, вы откроете результаты поиска в новой вкладке.

С помощью **F2** или **Ctrl+R** вы можете вернуться к обычному режиму отображения файлов.

## Сравнение файлов по содержимому

Выделите файлы, которые нужно сравнить. Используйте пункт меню **Файлы → Сравнить по содержимому..**

## Задания для самостоятельной работы:

Задания, отмеченные \*, можно пропустить, если вы не успеваете выполнить лабораторную работу.

1. Создайте в своем каталоге папку **Лабораторная\_ТС**.

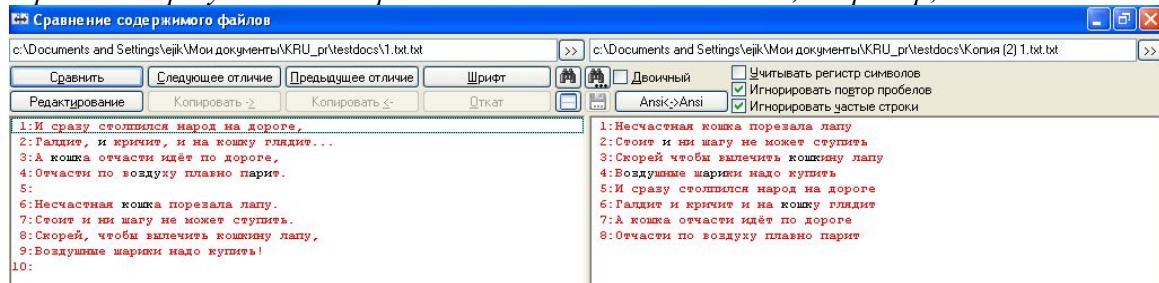
2. За одно нажатие клавиши F7 в папке **Лабораторная\_ТС** создайте каталог **Учеба**, с подкаталогами **Информатика** и **Обществознание**.

В отчёт включите текст команды, которую вы использовали. Например, при выполнении команды **папка1\папка2\папка3** в папке **Лабораторная\_ТС** будут созданы каталоги **папка1** и **папка3**, находящиеся внутри **Лабораторная\_ТС**, и **папка2**, находящийся внутри каталога **папка1** (Проверьте ;-)).

3. Создайте в каталоге **Информатика** 3 текстовых файла **file01.txt**, **file02.txt**, **file03.txt**. Введите в них текст (не менее 30 символов) при помощи Блокнота. Создайте 2 изображения **img1.bmp**, **img2.bmp**.

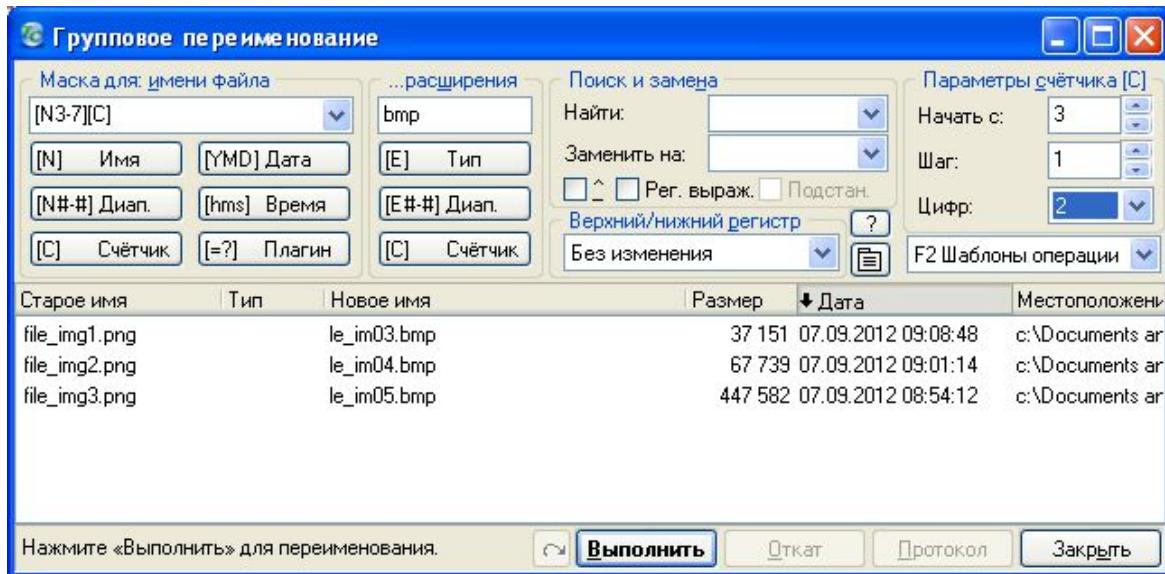
4. Скопируйте файл **file03.txt** в каталог **Обществознание**, переместите файл **file02.txt** в эту же папку.

5. Выполните сравнение файлов **file03.txt** и **file02.txt** по содержимому. В отчёт включите скриншот с результатами сравнения. Он может выглядеть, например, так:



6. Переименуйте файлы из папки **Обществознание** в: **new\_file001.txt**, **new\_file003.txt**, используя групповое переименование файлов.

В отчёт включите скриншот диалогового окна с настройками для переименования файлов. Например, настройки для переименования группы файлов **file\_img1.png**, **file\_img2.png**, **file\_img3.png** в **le\_im03.bmp**, **le\_im04.bmp**, **le\_im05.bmp** могут выглядеть так:



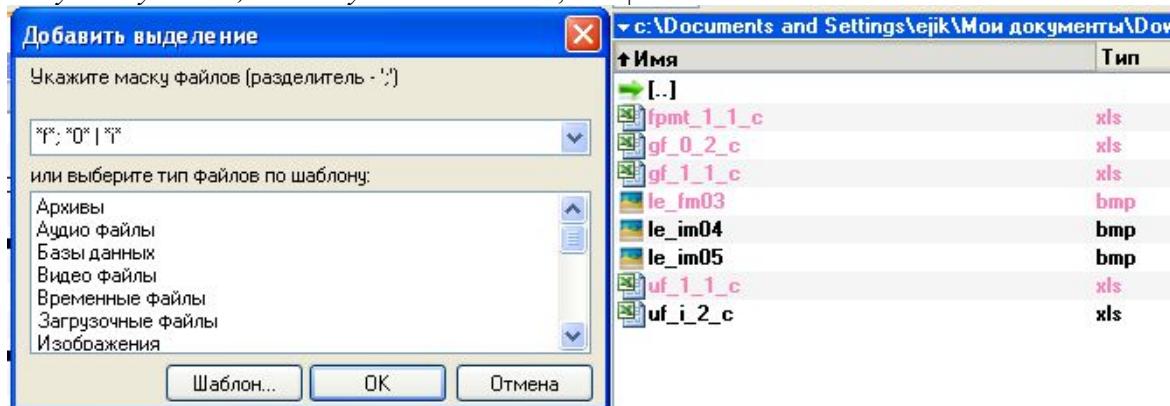
В заданиях 7-9 в отчёт включите текст маски и скриншот с результатами её применения для выделения файлов.

7. Создайте маску для выделения изображений.

8. Создайте маску для выделения файлов с именем, начинающимся с символов "fil".

9\*. Создайте маску для выделения файлов с расширением .txt , в названии которых присутствует сочетание “img”, но отсутствует сочетание “fil”.

Например, чтобы выделить файлы, в названии которых присутствует “0” или “f”, но отсутствует “i”, используется маска \*f\*; \*0\* | \*i\*



В заданиях 10-14 в отчёт включите скриншот окна с настройками и результатами поиска.

10. Найдите все файлы на диске **Z**, созданные в течение трех дней.

11. Найдите все файлы с расширением .txt, имена которых начинаются на “fi” на диске **Z**.

12. Найдите все файлы с расширением .txt, имена которых начинаются на “fi” или содержат в имени файла сочетание “img” на диске **Z**.

13. Найдите на диске **Z** все изображения, используя шаблоны поиска.

14\*. Найдите на диске **Z** все файлы, содержащие в названии строку “file”, после которой находится ровно 2 символа, а затем следует расширение. Подсказка: например, найдётся файл **new\_file01.txt**.

15. Заархивируйте папку **Лабораторная\_ТС**. Создайте **.zip** и **.rar** – архивы;

самораспаковывающийся архив (какое у него расширение?); архив, защищённый паролем. Результаты продемонстрируйте преподавателю. В отчёте внесите заполненную таблицу.

Размер несжатой папки, Кб	Размер zip-архива, Кб	Размер rar-архива, Кб	Размер самораспаковывающегося архива, Кб	Размер zip-архива, защищённого паролем, Кб

16. Продемонстрируйте отчёт и результаты выполнения лабораторной работы преподавателю.

## 2.1 Лабораторная работа №4 (4 часа)

**Тема:** «Технология работы в Microsoft Word. Набор, редактирование и форматирование текста»

**Цель работы:** Изучить основные принципы работы с простыми текстовыми документами с использованием текстового редактора MS Word.

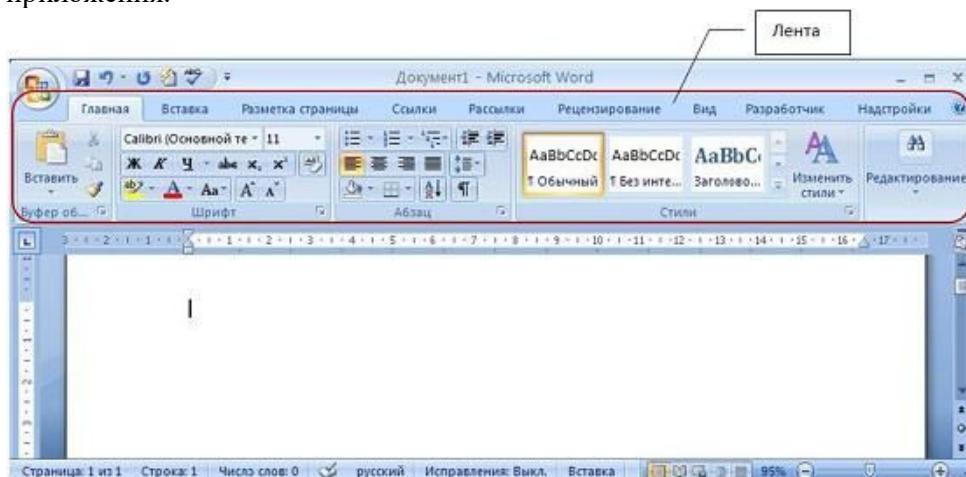
**Перечень приборов, материалов, используемых в лабораторной работе:**

1. Компьютеры с установленным программным обеспечением
2. Методические указания к выполнению лабораторной работы
3. Рабочая тетрадь.

**Описание (ход) работы:**

**Задание 1.** Запустите приложение MS Word и изучите специальные элементы интерфейса.

Запустите электронный процессор MS Word из главного меню Windows (**Пуск – Программы – Microsoft Word**). При запуске программы Word 2007 открывается окно приложения.

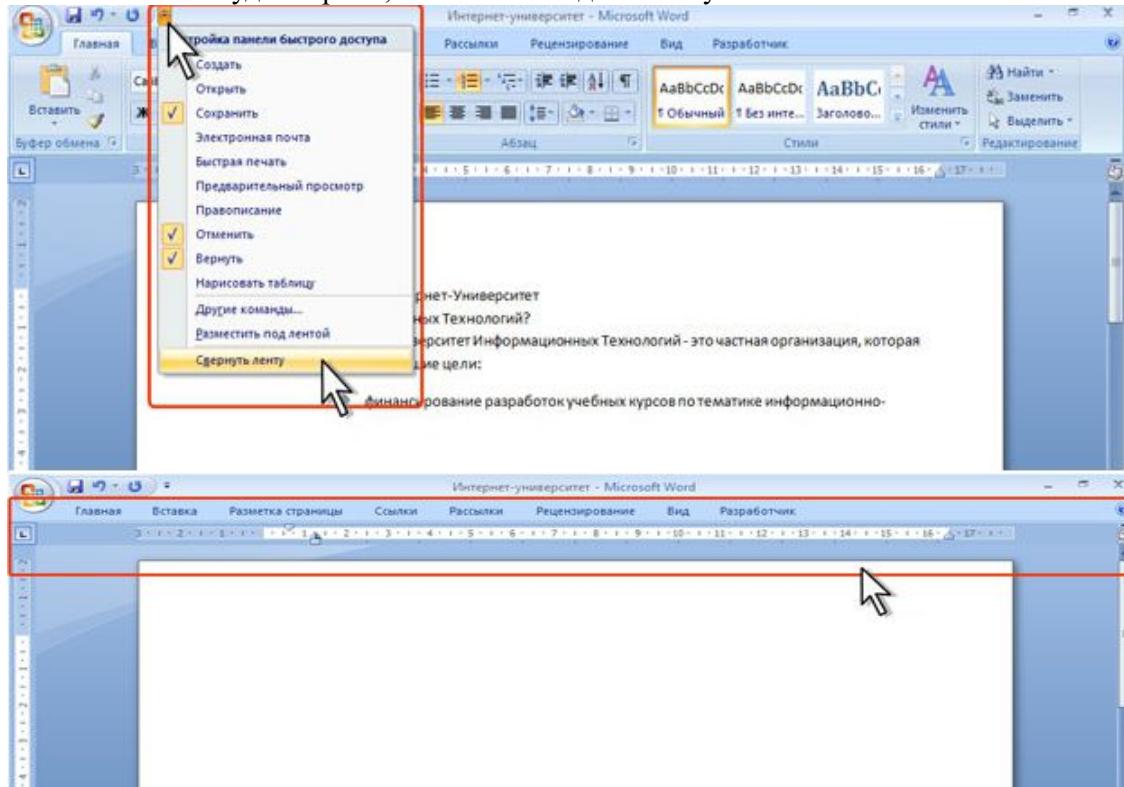


По умолчанию приложение открывается на вкладке Главная, на которой отображаются все требуемые средства для ввода текста или вставки текста из буфера обмена, его редактирования и форматирования. Основу среди Word 2007 составляют визуальные средства (команды в виде кнопок, полей для ввода информации или меню), расположенные на Ленте, для управления содержимым документа в процессе его создания и обработки. Необходимо отметить, что Лента состоит из 9 стандартных встроенных вкладок, корешки которых отображаются в окне приложения Word 2007: Главная, Вставка, Разметка страницы, Ссылки, Рассылки, Рецензирование, Вид, Разработчик и Надстройка.

Команды упорядочены в логические группы, собранные на вкладках.

Удалить ленту также нельзя. Однако чтобы увеличить рабочую область, ленту можно скрыть (свернуть).

1. Нажмите кнопку Настройка панели быстрого доступа
2. В меню выберите команду Свернуть ленту.
3. Лента будет скрыта, названия вкладок останутся



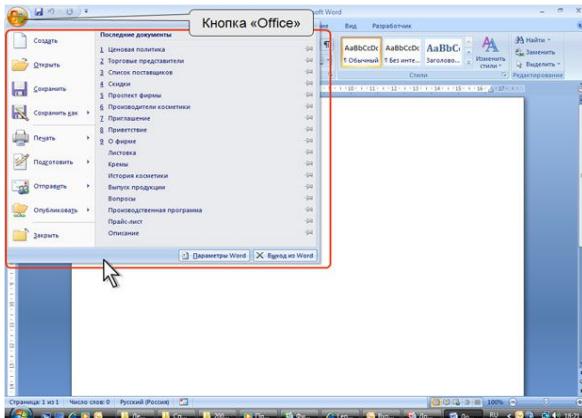
Для использования ленты в свернутом состоянии щелкните по названию нужной вкладки, а затем выберите параметр или команду, которую следует использовать.

Чтобы быстро свернуть ленту, дважды щелкните имя активной вкладки. Для восстановления ленты дважды щелкните вкладку.

Чтобы свернуть или восстановить ленту, можно также нажать комбинацию клавиш **Ctrl + F1**. Содержание ленты для каждой вкладки постоянно и неизменно. Нельзя ни добавить какой-либо элемент на вкладку, ни удалить его оттуда.

Кнопка «Office».

Кнопка «Office» расположена в левом верхнем углу окна. При нажатии кнопки отображается меню основных команд для работы с файлами, список последних документов, а также команда для настройки параметров приложения (например, Параметры Word) (рис. 4).

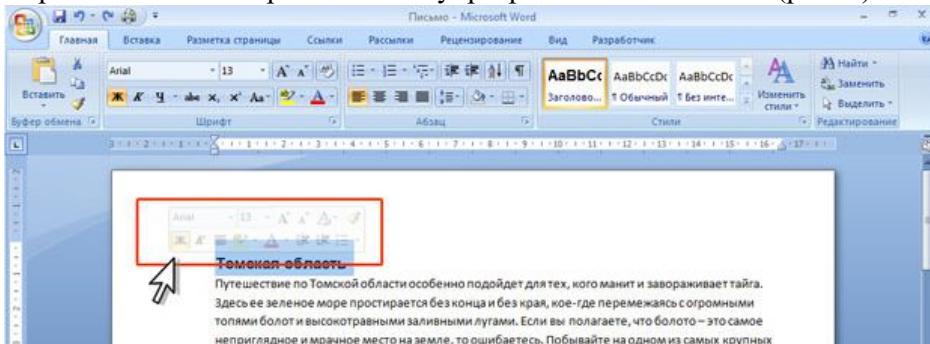


Некоторые из команд меню кнопки «Office» имеют подчиненные меню.

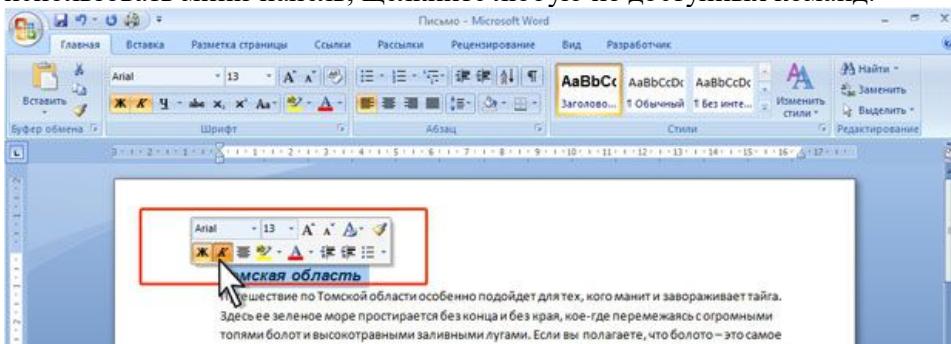
### Мини-панель инструментов

Мини-панель инструментов содержит основные наиболее часто используемые элементы для оформления текста документа.

Мини-панель появляется автоматически при выделении фрагмента документа. Первоначально отображается полупрозрачная мини-панель (рис. 5).



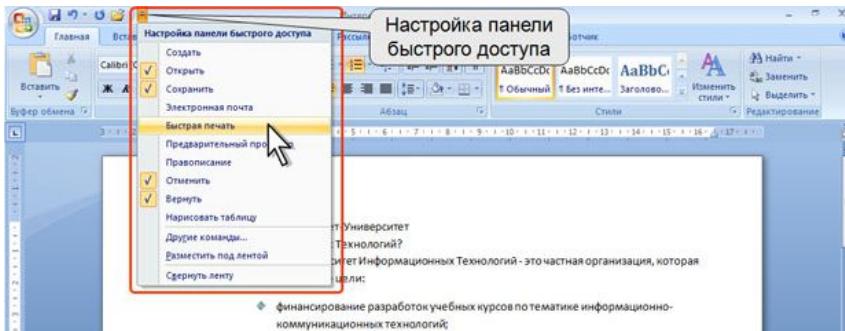
Мини-панель станет яркой, как только на нее будет наведен указатель мыши. Чтобы использовать мини-панель, щелкните любую из доступных команд.



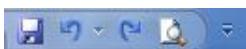
Состав элементов мини-панели инструментов – постоянный и неизменный.

### Задание 2. Настройка панели быстрого доступа.

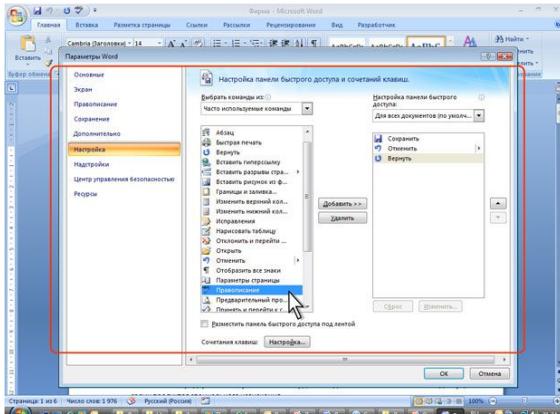
Панель быстрого доступа по умолчанию расположена в верхней части окна Word и предназначена для быстрого доступа к наиболее часто используемым функциям. По умолчанию панель содержит всего три кнопки: Сохранить, Отменить, Вернуть (Повторить). Панель быстрого доступа можно настраивать, добавляя в нее новые элементы или удаляя существующие.



Нажмите кнопку **Настройка панели быстрого доступа**. В меню выберите наименование необходимого элемента. Элементы, отмеченные галочкой, уже присутствуют на панели. Например, если щелкнуть на команде «Предварительный просмотр», то пиктограмма этой команды появится на Панели быстрого доступа (рис. 8).



По умолчанию Панель быстрого доступа размещена над Лентой, но ее можно разместить под Лентой. Для добавления элемента, отсутствующего в списке, выберите команду *Другие команды*.



Добавьте на панель быстрого доступа дополнительные элементы: предварительный просмотр, создать, параметры страницы.

### Задание 3. Настройка строки состояния.

Строка состояния находится в нижней части окна приложения. В строке состояния пользователь всегда видит номер строки документа и номер столбца, где в текущий момент находится курсор. Положение курсора можно изменить клавишами управления курсора или мышью только в пределах набранного текста.

Меню «Настройка строки состояния» (рис.10) или контекстное меню вызывается щелчком правой клавиши мыши на строке состояния. Активизация команд в меню «Настройка строки состояния» осуществляется щелчком левой кнопки мыши на команде.

### Задание 4. Знакомство с технологией создания, редактирования и сохранения текстовых документов в среде текстового процессора Word.

#### 4.1. Создание нового документа

Создать пустой документ в Word можно несколькими способами:

- нажатием клавиш *Control+N*. В результате создается новый пустой документ. При этом если вы уже

Настройка строки состояния	
Форматированный номер страницы	1
Раздел	1
Номер страницы	1 из 1
Вертикальное положение на странице	1,9 см
Номер строки	1
Столбец	1
Число слов	0
Проверка правописания	Ошибки не найдены
Язык	Английский (США)
Подписи	Отключен
Политика управления данными	Отключен
Разрешения	Отключен
Исправления	Выкл.
Caps Lock	Отключен
Замена	Вставка
Режим выделения	
Запись макроса	Нет записи
Всплывки режимов просмотра	
Масштаб	268%
Ползунок масштаба	

работаете с открытым документом, то вновь созданный документ появляется в новом окне.

– выбором кнопку Office → **Создать...** Результат будет такой же, как и при нажатии клавиши *Control+N*.



– щелчком по иконке **Создать** на панели быстрого доступа (если кнопка вынесена, см. Настройка панели быстрого доступа).

Создайте новый документ Word, используя меню кнопку Office → **Создать**

#### 4.2. Настройка параметров страницы

##### Использование линейки

– Чтобы отобразить или скрыть линейки, выберите вкладку: Вид → **Линейка** (если напротив команды **Линейка** расположен флажок, то горизонтальная и вертикальная линейки отображаются в окне документа, в противном случае – нет, поэтому необходимо установить его). Либо справа вверху полосы прокрутки кнопка **Линейка**.

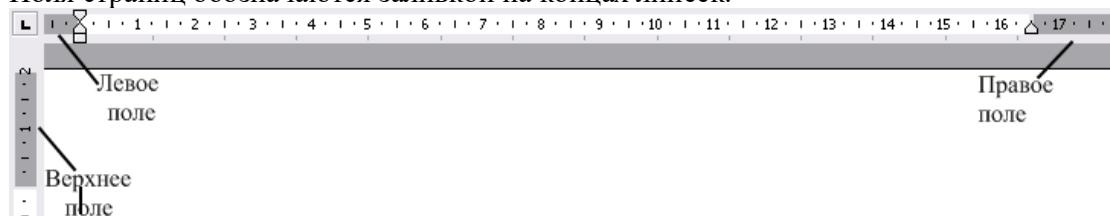
##### Использование горизонтальной и вертикальной полос прокруток

– Чтобы отобразить полосы прокрутки используем либо кнопку настройки панели быстрого доступа → **Другие команды** → **Дополнительно** → **Экран** (выберите команду: снимите щелчком мыши флажок **Горизонтальную полосу прокрутки** → **OK**).

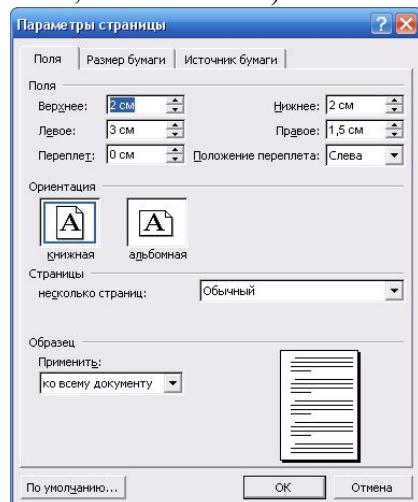
– Либо кнопка Кнопка "Office" → **Параметры Word** → **Дополнительно** → **Экран**

#### 4.3. Настройка полей страницы

Поля страниц обозначаются заливкой на концах линеек.



Для установки параметров (поляй) страницы выполните следующие действия: кликните дважды на конце линейки в появившемся диалоговом окне выберите вкладку **Поля** (рис. 12) → в области **Поля** установите стандартные параметры (Левое –3 см, Правое –1,5 см, Верхнее -2 см, Нижнее -2 см) → **OK**.



Также можно вынести кнопку **Параметры страницы** на панель быстрого



доступа  
(см. Настройка панели быстрого доступа)

Параметры страницы

Обратите внимание на поля страницы документа, отображающиеся на вертикальной и горизонтальной линейках, они должны соответствовать выбранным вами параметрам. Настройте параметры страницы: Верхнее – 2,5; Левое – 3; Нижнее – 2; Правое – 1,5; Ориентация – книжная.

## РЕДАКТИРОВАНИЕ ТЕКСТА В WORD

### Задание 5. Ввод текста в документ.

Наберите следующий выделенный текст **не нажимая клавишу Enter** (с одной строки на другую Word будет переносить текст автоматически, а клавиша <Enter> используется для перехода на новую строку и отмечает новый абзац. Обратите внимание, что после знаков препинания пробел ставить *надо*, а до знаков препинания – *не надо*

#### 5.1. Основные операции по редактированию

##### При наборе текста в Word **применимые правила**

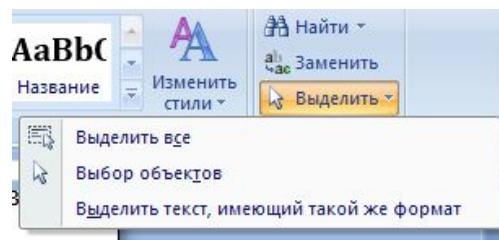
рекомендуется делать более одного пробела между словами; при вводе знаков пунктуации (точек, запятых, двоеточия) пробел перед ними после слова не ставят, а ставят сразу после знака перед следующим словом; нельзя нажимать клавишу <Enter> после каждой строки абзаца, она используется только при переходе к новому абзацу; не следует использовать клавишу пробела для создания «красной строки» абзаца и вставки неформатированного текста.

Если в процессе набора текста, Вы – допустили ошибки исправьте их, **Мой компьютер** → диск Students → Факультет → используя клавиши удаления символов специальность → Курс → Папка под вашей <Del> (стирает символы справа от фамилией → **Текст1.doc** → **Открыть**). курсора) и <Backspace> (стирает символы слева от курсора). Перед тем как совершить какую-либо операцию по редактированию текста или его фрагмента, необходимо его выделить. Выделение осуществляется с использованием клавиатуры или мыши.

– Выделите фрагмент текста – «**При наборе текста в Word придерживаются следующих правил:**» с помощью левой клавиши мыши, для этого щелкните мышью в его начале и, не отпуская клавиши, протяните до конца фрагмента.

**При наборе текста в Word придерживаются следующих правил:** не рекомендуется делать более одного пробела между словами; при вводе знаков пунктуации (точек, запятых, двоеточия) пробел перед ними после слова не ставят, а ставят сразу после знака перед следующим словом; нельзя нажимать клавишу <Enter> после каждой строки абзаца, она используется только при переходе к новому абзацу; не следует использовать клавишу пробела для создания «красной строки» абзаца и для размещения текста по центру строки.

- Для выделения слова «**рекомендуется**» - сделайте по нему двойной щелчок левой клавишей мыши.
- Для выделения всего текста можно использовать либо тройной щелчок мышью в любом месте текста, либо клавиши <Ctrl+A>, либо протаскиванием выделения удерживая левую клавишу мыши, либо на вкладке Главная → Выделить → Выделить все



— Для выделения непоследовательных элементов текста с использованием мыши необходимо:

- выделить первый фрагмент текста, например, «При наборе текста в Word»
- удерживая нажатой клавишу <Ctrl> выделить мышью следующий фрагмент текста, например, «При наборе текста в Word».
- повторить операцию столько раз, сколько необходимо.

**При наборе текста в Word придерживаются следующих правил: не рекомендуется делать более одного пробела между словами; при вводе знаков пунктуации (точек, запятых, двоеточия) пробел перед ними после слова не ставят, а ставят сразу после знака перед следующим словом; нельзя нажимать клавишу <Enter> после каждой строки абзаца, она используется только при переходе к новому абзацу; не следует использовать клавишу пробела для создания «красной строки» абзаца и для размещения текста по центру строки.**

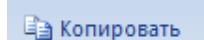
- далее можно работать с выделенным текстом (копировать его, удалить его, изменить стиль и пр.).
- Режим выделения:
- выделить первый фрагмент текста – «При наборе текста в Word придерживаются следующих правил:»
- вызовите контекстное меню строки состояния и установите флашок на Режим выделения;
- нажать клавиши Shift+F8. (Это переведет Word в режим выделения);
- Используйте клавиши со стрелками переместите курсор к началу следующей порции выделяемого текста – «при вводе знаков пунктуации». Удерживая нажатой клавишу Shift, выделите эту порцию текста.
- Повторите указанные действия для текстового фрагмента – «не следует использовать клавишу пробела для создания «красной строки» абзаца и для размещения текста по центру строки»
- Теперь вы можете работать с выделенным текстом.
- Нажмите клавишу Esc, чтобы выйти из этого режима.

## 5.2. Вырезание, копирование и вставка текста

Вырезание и копирование текста в Word выполняется так же, как и в других приложениях. Для этих операций можно использовать манипулятор «мышь» или клавиатуру.

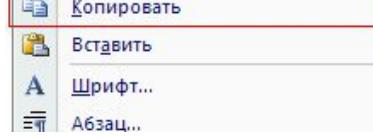
 на вкладке Главная, либо сочетание клавиш *Control+X*, либо через контекстное меню



 на *Control+C*, либо через

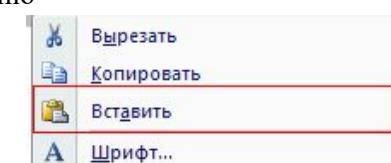


 на вкладке  
либо через контекстное меню



вкладке Главная, либо клавиши контекстное меню

Главная, либо клавиши *Control+V*,



### 5.3. Быстрое копирование и перемещение фрагментов текста

- Поместите курсор внутрь последнего абзаца.
- Выполните тройной щелчок левой клавишей мыши (*в результате этих действий абзац становится выделенным*) → на выделенном абзаце нажмите и удерживайте нажатыми левую кнопку мыши и клавишу *Control* переместите указатель мыши на новую строку. В результате таких действий появится копия выделенного текстового фрагмента.
- Выделите текстовый фрагмент – **«При наборе текста в Word придерживаются следующих правил:»**, находящийся в последнем абзаце текста.
- Удерживая нажатой левую клавишу мыши на выделенном фрагменте переместите указатель мыши в конец абзаца. При этом происходит перемещение фрагмента без его копирования, а результат операции выглядит следующим образом:

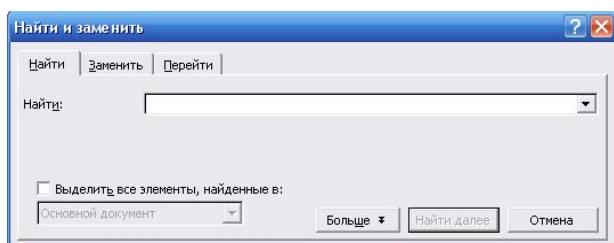
не рекомендуется делать более одного пробела между словами; при вводе знаков пунктуации (точек, запятых, двоеточия) пробел перед ними после слова не ставят, а ставят сразу после знака перед следующим словом; нельзя нажимать клавишу *<Enter>* после каждой строки абзаца, она используется только при переходе к новому абзацу; не следует использовать клавишу пробела для создания «красной строки» абзаца и для размещения текста по центру строки **При наборе текста в Word придерживаются следующих правил:**

- Отмените выполненное действие, щелкнув на кнопку **Отменить**  панели инструментов *Стандартная*.

### Задание 6. Поиск и замена текста

В Word на вкладке главная есть команды *Найти* и *Заменить*, с помощью которых можно автоматизировать процесс поиска и замены текста в документе.

Чтобы отобразить диалоговое окно *Найти и заменить*, используйте клавиши *Control+F* или выберите **Главная** → **Найти (Заменить)**. Появится диалоговое окно *Найти и заменить* (рис.13):

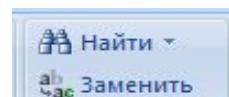


#### Диалоговое окно Найти и заменить

- Замените все предлоги «в» на предлоги «на». Для этого перейдите на вкладку *Заменить* и в поле *Найти* введите текст, который вам нужно найти, в частности, предлог **- в**.
- Для замены этого текста на другой введите в поле *Заменить на* новый текст, в частности, предлог **- на** (можно выбрать дополнительные параметры поиска, такие как *учет регистра, только слово целиком или поиск подобных слов*) → установите флагки **Учитывать регистр** и **Только слово целиком** → **Найти** (при этом будет найден первый элемент) → **Заменить** (необходимо произвести столько замен, сколько найденных элементов текста).

**Замечание:** При нажатии кнопки **Найти все** Word выделяет все экземпляры отыскиваемого в документе текста. Аналогично, при нажатии кнопки **Заменить все**, Word заменит все найденные экземпляры.

Произведите обратную замену.



Аналогично замените текст «Enter» на текст «ENTER»: Главная → Найти и заменить → установите флажки Учитывать регистр и Только слово целиком → Найти: *Enter* → Заменить: *ENTER* → Заменить все → OK → Закрыть.

### **Задание 7. Сохранение документа.**

Большинство текстовых процессоров используют следующие три операции сохранения данных.

– *Сохранить и продолжить редактирование*. Эта операция выполняется периодически, гарантируя сохранность более свежей копии проектируемого документа на случай возможной его утраты.

– *Сохранить и выйти*. Мы используем эту операцию для сохранения отредактированного документа и выхода в операционную систему.

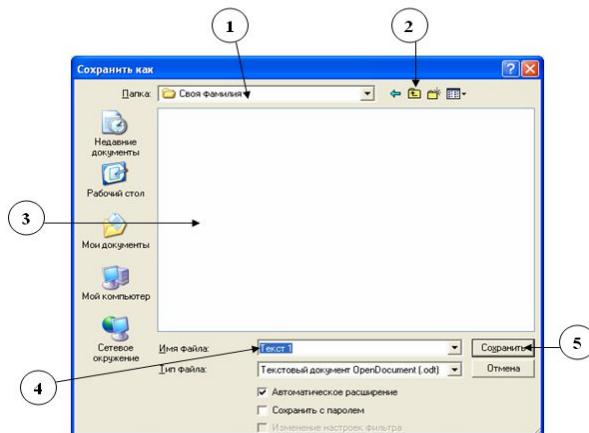
– *Выйти без сохранения*. Эта операция позволяет выйти в операционную систему без сохранения документа, с которым работали.

Сохраните текстовый документ в свою системную папку под именем *Текст 1\_1*.

Сохраните набранный текст в файле с именем *Текст1* в свою системную папку: **Кнопка Office**→**Сохранить как** →**Мой компьютер**→ диск Students → Факультет → специальность→ Курс→ Папка под вашей фамилией → Имя файла *Текст1*→ **Сохранить**.

В диалоговом окне **Сохранить как** (рис.14) довольно много элементов управления. Основными являются:

1. Список папок, в котором показывается текущая папка (по умолчанию Word сохраняет файлы в папке **C:\Мои документы**).
2. Кнопка выхода в предыдущую папку (родительскую) (в показанном примере, для **C:\Мои документы\Своя фамилия**, родительской является папка **\Мои документы**).
3. Список с содержимым текущей папки.
4. Стока ввода имени файла с которым будет сохранен документ. В этой строке вы **обязательно должны** ввести корректное имя, прежде чем сохранить файл.
5. Кнопка **Сохранить**, которую необходимо щелкнуть для записи документа в файл с указанным именем (это действие может быть заменено нажатием клавиши <Enter>).



Имя файлу дается при первом сохранении документа. Впоследствии при выборе команды кнопка **Office** → **Сохранить** новый вариант документа будет **заменять** старый в файле с тем же именем.

## **ОСНОВНЫЕ ОПЕРАЦИИ ПО ФОРМАТИРОВАНИЮ ТЕКСТОВЫХ ДОКУМЕНТОВ WORD.**

**Цель работы:** изучить правила форматирования текстового документа, научиться применять параметры форматирования текстового документа в приложении MS Word

**Задания для выполнения и методические рекомендации:**

При редактировании документа изменяется его содержимое, а, форматируя документ, вы изменяете его внешний вид.

**Форматирование** - это способность текстового процессора изменять оформление документа на странице. В Word различают **форматирование символов** и **форматирование абзацев**.

При форматировании символов, как правило, задаются параметры шрифта: гарнитура, размер, начертание, тип подчеркивания и прочее.

**Гарнитура шрифта** - это термин, которым определяется общая форма символов. Для любого фрагмента документа (слова, строки, абзаца, предложения или всего документа) можно задать шрифт. Понятие шрифта включает в себя совокупность следующих параметров:

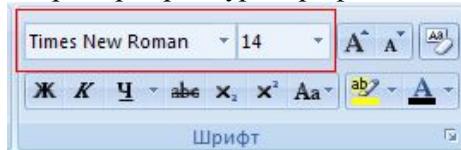
- тип шрифта (или гарнитура). Это может быть Таймс, Курьер и т.д.;
- размер шрифта. Задается в пунктах. Например 14 пт, 16 пт и т.д.;
- начертание (обычный, полужирный, курсив, полужирный курсив);
- тип подчеркивания (одинарное, двойное, волнистое и т.д.);
- цвет шрифта;
- эффекты (верхний и нижний индекс, зачеркивание, тень и т.д.);

#### **Задание 8. Настройка параметров форматирования символов.**

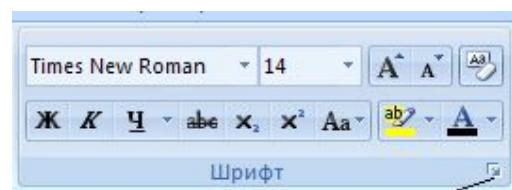
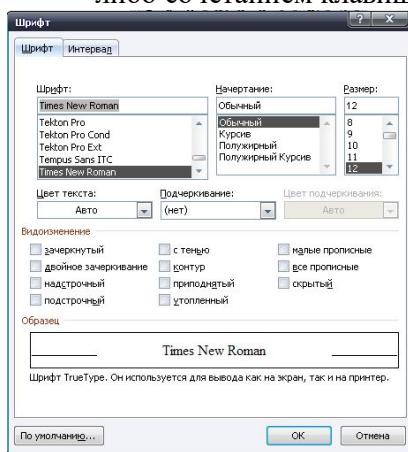
Откройте созданный файл *Текст1.doc* **Мой компьютер** → диск *Students* → Факультет → специальность → Курс → Папка под вашей фамилией → *Текст1* <Открыть>.

Форматирование символов можно производить несколькими способами:

- с помощью вкладки Главная → группа Шрифт и выбрать необходимые параметры, например гарнитура шрифта и кегель (размер)



- с помощью диалогового окна Шрифт, которое можно вызвать либо через вкладку Главная → группа Шрифт
- либо сочетанием клавиш **Ctrl+D**.



Открытие диалогового окна "Шрифт"

Отформатируйте текст одним из предложенных способов применив следующие параметры: Шрифт - Times New Roman, кегель – 14, начертание – обычное.

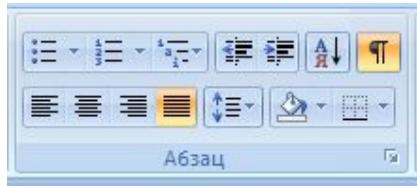
**Замечания.** Если вы хотите вставить специальный символ, которого нет на клавиатуре (п, у, V, oo, €, ® и т. п.), нужно выбрать вкладку *Вставка -Символы*. Перед вами откроется окно, в котором можно выбрать необходимые символы и нажать кнопку *Вставить*.

Если в документ требуется вставить формулу, нужно выбрать вкладку *Вставка - Символы* пункт *Формула*. В появившемся окне можно выбрать имеющуюся формулу, либо составить новую, нажав на кнопку *Вставить новую формулу* и использовав средства контекстной ленты *Работа с формулами - Конструктор*.

### **Настройка параметров форматирования абзацев.**

Форматировать **абзацы**, так же как и символы, можно одновременно с вводом текста или позже, когда текст уже набран. При форматировании абзацев кроме параметров шрифта задаются параметры расположения абзаца: выравнивание и отступы относительно полей страницы, интервалы между абзацами и между строками внутри абзаца, а также положение самого абзаца на странице.

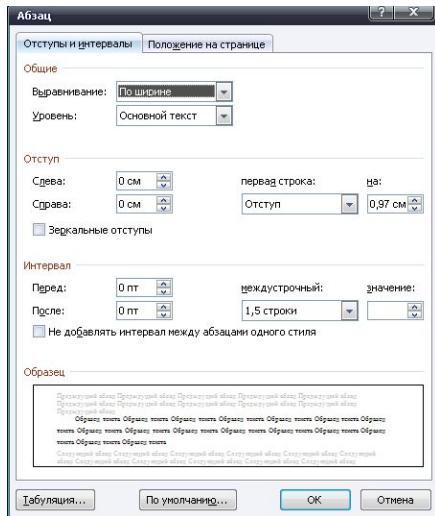
Для абзацного форматирования предназначены: группа кнопок панели *Абзац* вкладки *Главная*



и диалоговое окно *Абзац*, вызываемое с панели группы *Абзац*.



Открытие диалогового окна "Абзац"



**Рис. Окно диалога Абзац**

На вкладке *Отступы и интервалы* данного окна можно задать:

- *выравнивание текста в абзаце* (по левому краю, по центру, по правому краю, по ширине), для этого также можно использовать кнопки на панели *Абзац*;
- *отступы текста, выступы краев страницы, строки абзаца «красная строка»*, также можно применять кнопки
- *межстрочный интервал в абзаце*, можно использовать выпадающий список установки межстрочного интервала

На вкладке *Положение на странице* диалогового окна *Абзац* можно задать:

- запрет висячих строк (обеспечивает перенос всего абзаца на следующую страницу без оставления одной строки на предыдущей);
  - неразрывность абзаца, запрет отрыва от предыдущего;
  - положение абзаца с новой страницы;
  - запрет расстановки переносов в абзаце и нумерации его строк.
- С помощью диалогового окна *Абзац* установите для каждого абзаца файла *Текст 1* следующие параметры: Отступ слева «0 см», справа «0 см», первая строка *отступ «1,25 см»* → Интервал перед: «0», после: «0», межстрочный → «полуторный». Выравнивание→по ширине→Последняя строка→По левому краю→OK.

### **Задание 9. Формат по образцу**

Полезной при форматировании текста в документе является кнопка  **Формат по образцу**, которая переносит параметры форматирования указанного объекта на выделяемый фрагмент. Чтобы перенести все заданные параметры форматирования на новый абзац необходимо:

- установить курсор в любом месте абзаца, параметры форматирования которого мы хотим использовать;
- нажать кнопку *Формат по образцу* на вкладке *Главная - Буфер обмена* (если необходимо форматировать за один раз несколько разных фрагментов, следует сделать двойной щелчок на кнопке);
- выделить текст, на который надо перенести форматирование (если был сделан двойной щелчок на кнопке *Формат по образцу*, то можно выделять последовательно нужные фрагменты текста; по завершении всей операции форматирования надо один раз щелкнуть на кнопке *Формат по образцу*, чтобы «отжать» ее).

### **Задание 10. Работа со списками**

Списки - это фрагменты текста, пункты которого отмечены специальными знаками. Списки могут быть маркированными, нумерованными и многоуровневыми.

Для работы со списками служат пять верхних кнопок панели *Абзац* вкладки



*Главная*, также *Нумерованный* и *маркированный* список могут быть созданы с использованием команд *Маркеры*, *Нумерация* по нажатию на тексте правой кнопки мыши.

При формировании *многоуровневого списка*, чтобы задать создание маркеров очередного уровня, можно использовать клавишу *Tab* (либо кнопку *Увеличить отступ* на панели *Абзац*). Вернуться к вводу данных предыдущего уровня можно, нажав сочетание *Shift+Tab* (либо кнопку *Уменьшить отступ* на панели *Абзац*).

При необходимости редактирования многоуровневого списка, щелкните кнопкой мыши на кнопке *Многоуровневый список* - *Определить новый многоуровневый список*. Здесь можно настроить формат номера, расстояние, тип шрифта и другие параметры списка.

Если необходимо сформировать новый стиль списка, то нужно воспользоваться пунктом *Определить новый стиль списка*. В появившемся окне можно настроить все необходимые параметры стиля, а также задать область действия нового формата.

**Задание 10.1.** Наберите следующий текст, разделяя его на абзацы (согласно образца), используя клавишу **[Enter]**:

Установите следующие параметры форматирования: заголовок текста и название клавиш - *Полужирным*, заголовок текста - *По центру*, основной текст – *По ширине*, оформите маркированный список. С помощью команды *Регистр*  оформите заголовок текста прописными буквами.

Наиболее часто используемые клавиши: [Enter]

[Enter]

Enter – переход к новому абзацу, пропуск пустой строки; [Enter]

Delete – удаление выделенного фрагмента, если нет выделенного фрагмента, то удаление символа справа от курсора; [Enter]

BackSpace – удаление символа слева от курсора; [Enter]

Shift – удержание в нажатом виде приводит к смене режима прописных или строчных букв, используется в других специально оговоренных случаях; [Enter]

Caps Lock - назначение основным режима прописных или строчных букв (подсвеченный индикатор в правой части клавиатуры указывает на режим ПРОПИСНЫХ букв); [Enter]

Ctrl, Alt – вспомогательные клавиши, расширяющие возможности манипулирования клавиатурой, [Enter]

Home, End – быстрая установка клавиатурного курсора в начало или конец строки, [Enter]

Page Up, Page Down – клавиши листания текста к началу или концу [Enter]

Сохраните файл под именем *Текст 2* в своей папке. Закройте документ.

**Задание 11. Работа со стилями в документе и настройки собственного стиля.**

Стиль — это именованный и сохраненный набор параметров форматирования. Например, стиль может включать в себя: - шрифт Arial размером 14 пунктов, отступ величиной 1 см, двойной межстрочный интервал и выравнивание по обоим краям. Определив стиль, можно быстро применить его к любому тексту документа. Использование стилей позволяет ускорить подготовку больших документов со сложным оформлением.

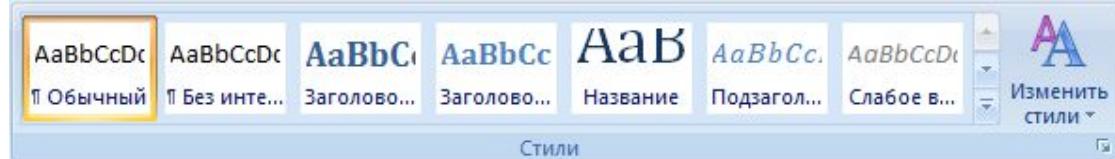
Стили документов Word 2007 можно:

- изменить имя существующего стиля;
- применять другие стили к абзацу или фрагменту текста (осуществлять наложение стилей);
- изменять существующие стили;
- создавать новые.

Внимание! Все операции со стилями (переименовать, применять, изменять и создавать) могут выполняться только над выделенными абзацами (фрагментами текста) или над абзацем, в котором установлен курсор.

По умолчанию во вновь создаваемые документы Word вводит текст в стиле **Обычный** (Экспресс-стиль). В этом стиле установлены основные параметры форматирования абзаца: шрифт - (Calibri - по умолчанию) + Основной текст, выравнивание символов - По левому краю, междустрочный интервал - множитель 1,15 ин., интервал После абзаца - 10 пт., Запрет висячих строк.

Экспресс-стили — это наборы различных вариантов форматирования, которые отображаются в виде эскизов в коллекции экспресс-стилей. При наведении указателя мыши на эскиз экспресс-стиля можно увидеть, как выделенный фрагмент текста или абзац, в котором установлен курсор, примет выбранное форматирование.



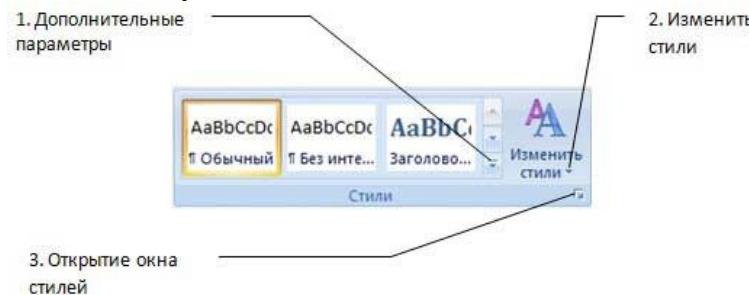
Существует три основных типа стилей:

- стиль текста задает параметры форматирования выделенного фрагмента текста внутри абзаца (обозначение - a);
- стиль абзаца содержит параметры форматирования абзацев (обозначение - Т);
- стиль таблицы содержит параметры форматирования таблиц (обозначение - Таблица).

Абзацы, содержащие параметры форматирования фрагмента текста и абзацев, обозначаются - a 

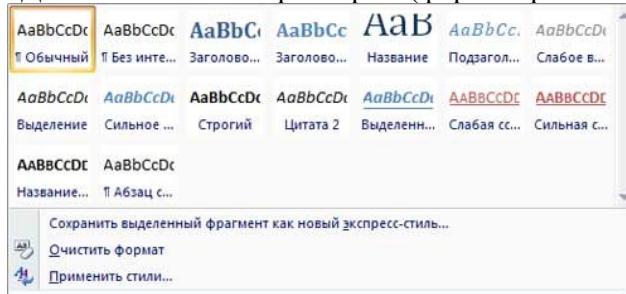
В группе Стили на вкладке Главная помещены три кнопки для работы со стилями:

- Дополнительные параметры (форматирование заголовков и другого текста);
- Изменить Стили (изменение набора или коллекции стилей, цветов и шрифтов, используемых в документе);
- Открытие окна стилей.

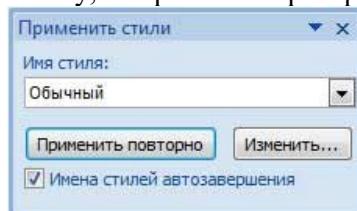


Первая кнопка в группе Стили – «Дополнительные параметры».

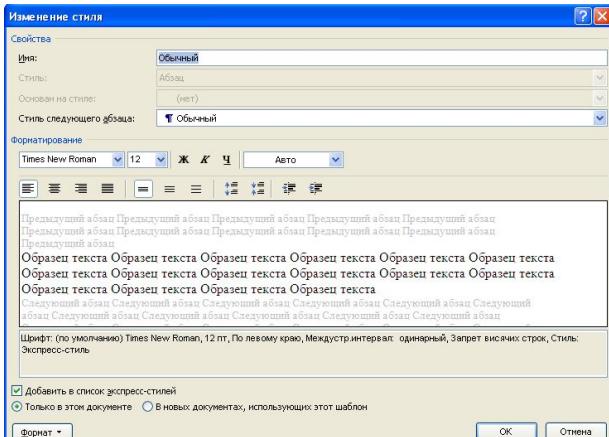
Для отображения всей коллекции Экспресс-стилей надо щелкнуть на кнопке «Дополнительные параметры» (форматирование заголовков и другого текста).



В нижней части коллекции размещены три команды. Команда «Сохранить выделенный фрагмент как новый экспресс-стиль» позволяет переименовать Стили абзаца или переформатировать абзац, т.е. создать новый стиль и сохранить его под уникальным именем или **изменить** стиль. Команда очистить формат, которая преобразует любой формат в «Обычный». А третья команда «Применить стили» позволяет применить другой Стили к тексту, выбрав его из раскрывающегося списка



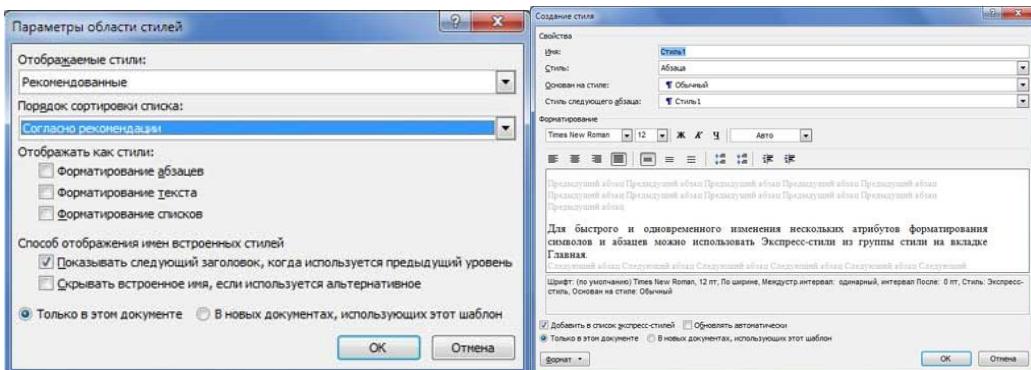
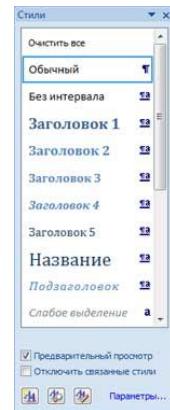
или изменить существующий стиль. Для изменения стиля надо в окне диалога щелкнуть на кнопке «Изменить». Откроется окно диалога «Изменение стиля», в котором можно ввести некоторые параметры вновь отформатировать и установить флагок «Добавить в список экспресс-стилей», а переключатели установить в режим «Только в этом документе» или «В новых документах, использующих этот шаблон» и щелкнуть OK.



При щелчке на кнопке «Открытие окна стилей» открывается область задач «Стили», в которой отображаются стили абзацев и текста. Эти стили можно применить к абзацам и фрагментам текста, в которых установлен курсор. При наведении указателя мыши на стили будут отображаться их параметры.

Кроме того, щелкнув на первой нижней кнопке AA панели стилей можно вызвать окно «Создание стиля». Для создания новых стилей необходимо заполнить окно диалога «Создание стиля» и щелкнуть на кнопке OK.

Для настройки панели стилей надо щелкнуть на команде «Параметры...» в области стилей, откроется окно диалога «Параметры области стилей».



В этом окне можно установить отображаемые стили в области стилей (Рекомендованные, Используемые, В текущем документе, Все), порядок сортировки (По алфавиту, Согласно рекомендации и т.д.), установить флажки в группе «Отображать как стили» и т.д.

В окне диалога «Инспектор стилей», вызываемом нажатием второй нижней кнопки («А») в области стилей, отображаются параметры форматирования стилей абзаца и текста. Кроме того, щелкнув на третьей нижней кнопке в области стилей («AA»), можно вызвать окно

диалога «Управление стилями» для настройки «Автозамены», «Автоформата» и выполнения других настроек.

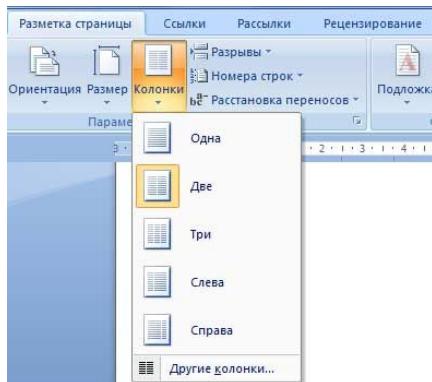
**Задание 10.2.** В диалоговом окне Стили измените стиль Обычный → вызвать подменю →Изменить: шрифт - Times New Roman, размер -14 пт, выравнивание - по ширине, интервал - одинарный→Добавить в шаблон→OK.

Аналогично измените стиль Заголовок 1 (шрифт - Times New Roman, размер -14 пт, выравнивание –

по центру, начертание – полужирный; интервал - одинарный). Примените стиль Заголовок 1 к названию документа Текст 2.

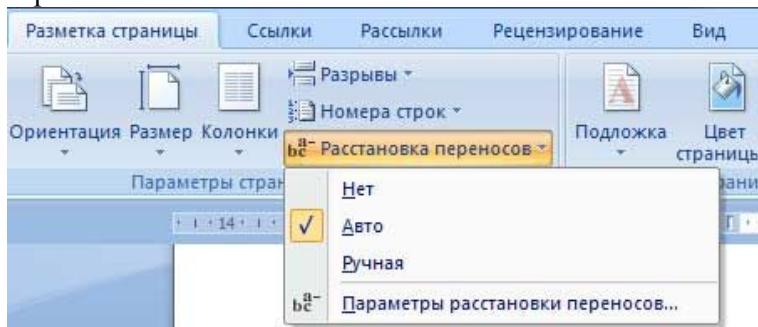
**Задание 11. Размещение данных в столбцах.**

Текст в столбцах переходит с низа одной колонки наверх другой, для принудительного перехода с одной колонки в другую надо выполнить разрыв колонки (команда «Разметка страницы/Разрывы/Столбец»). Создавать столбцы можно с помощью пиктограммы «Колонки» на вкладке «Разметка страницы» в группе «Параметры страницы».



Или с помощью окна диалога «Колонки», которое можно вызвать, щелкнув на команде «Другие колонки».

Для того чтобы между словами не было больших разрывов можно выполнить ручную или автоматическую расстановку переносов. Для расстановки переносов надо на вкладке «Разметка страницы» в группе «Параметры страницы» открыть список и выбрать требуемый вариант.



В столбцах можно размещать рисунки, таблицы, диаграммы и другие объекты документа. Необходимо отметить, что размеры объектов документов Word не должны превышать ширину столбца.

## ЗНАКИ ФОРМАТИРОВАНИЯ

Знаки форматирования (Непечатаемые символы) используются для форматирования текста и не выводятся на печать. Команда Знаки форматирования отображает непечатаемые символы в тексте, такие как знак абзаца, разрыв строки, позиция табуляции и пробел.

Для получения доступа к этой команде выберите на стандартной панели инструментов щелкните значок **Знаки форматирования** ¶.

Чтобы указать, какие непечатаемые символы должны отображаться, выберите **кнопка Office – Параметры Word – Экран** и выберите те, которые необходимо отображать.

**Задание 12.** Наберите следующий текст по образцу. Обратите внимание на непечатаемые символы, отображаемые в образце документа.

## ПЕЧЕНЬЕ¶

Сырки, муку, маргарин, поставить в холодильник на 2 часа.¶

Тонко раскатать тесто. Смазать белком, взбитым с сахаром, и свернуть рулетом. Нарезать ломтиками и сложить на смазанный противень. Печь примерно 25 минут.¶

2 сырка по 100 г, 1 пачка маргарина, 2 яйца, 1 стакан сахара, 0,5 чайной ложки соды, погашенной уксусом, 2,5 стакана муки.¶

Проведите соответствующее форматирование:

- Заголовок текста оформите стилем **Заголовок 1**, по центру листа;
- Текст рецепта – стиль **Основной текст**, по ширине листа;
- Последний абзац текста оформите курсивом;
- Установите отступ первой строки абзаца *1,25 см*, отбивка *После абзаца 0,50 см*, *Межстрочный интервал – Одинарный*. Сохраните файл в своей папке под название **Рецепт**.

**Задание 13.** Нерастяжимый пробел и принудительный конец строки и возможности их применения.

Создайте форматированный документ по образцу (рис. 34), используя полученные ранее знания.

В тексте используются новые элементы форматирования: *отступ абзаца слева*, *принудительный конец строки* ↵ (позволяет переместиться на следующую строку без образования нового абзаца) и *нерастяжимые пробелы* ⏐ (позволяют при форматировании не «отрывать» слова друг от друга).

355000, г. Ставрополь, ул. Мира, 112, кв. 7-  
Петровой Ирине Владимировне.¶

Согласно заключенному с Вами договору от 25 февраля 2007 г. Вы обязаны  
возвратить мне, Иванову Сергею Владимировичу, взятые Вами взаймы 250 000  
(двести пятьдесят тысяч) рублей в срок до 25 февраля 2008 г.¶

Сообщаю, что в настоящее время я проживаю по адресу: 355000, ул.  
Ворошилова, 12, кв. 75.¶

Прошу Вас выплатить мне указанную сумму почтовым переводом за мой счет по  
моему адресу.¶

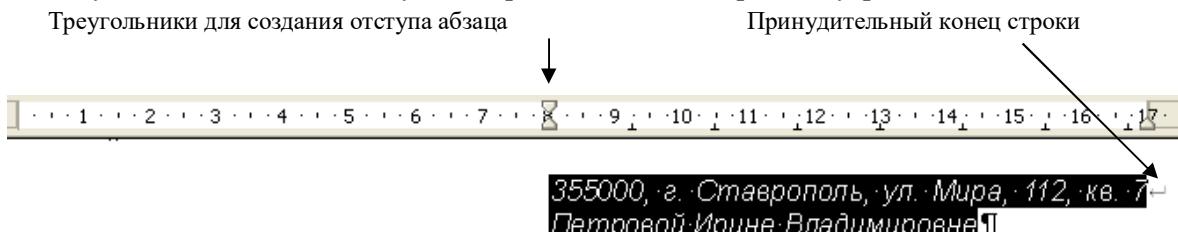
26 февраля 2007 г.

С.В. Иванов

¶

Для выполнения отступа абзаца слева проще всего воспользоваться маленькими треугольниками, расположенными на горизонтальной линейке. Необходимо выделить абзац и переместить треугольники при помощи мыши в нужное положение, или в диалоговом окне **Абзац** – вкладка **Отступы и интервалы** установить *Отступ Слева* (например 8,00 см).

Принудительный конец строки задается нажатием клавиш **[Shift+Enter]** и употребляется в том случае, когда вам самим нужно определить конец строки внутри абзаца.



При создании подписи к документу необходимо воспользоваться нерастяжимыми пробелами.

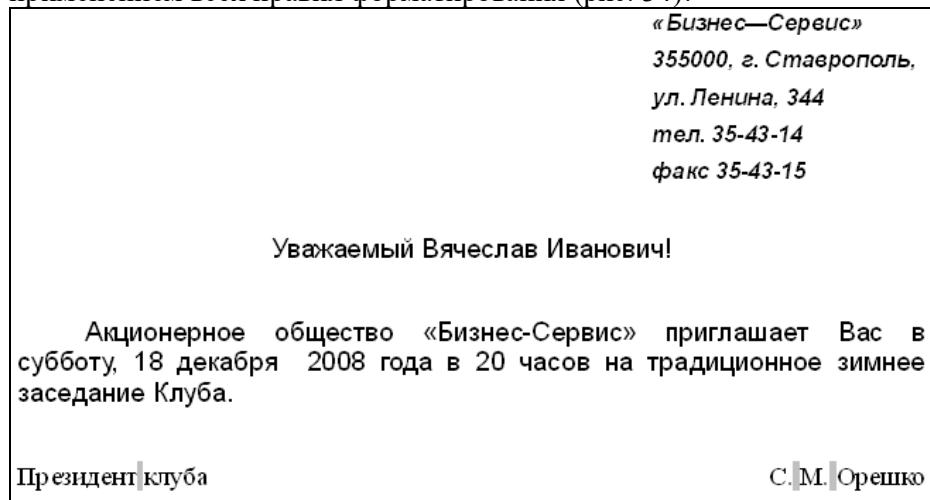
Нерастяжимый (неразрывный) пробел **[Ctrl+Shift+Пробел]** не позволяет слова, между которыми он вставлен, располагать на разных строчках и сохраняет этот пробел фиксированным (по ширине) при любом выравнивании абзаца (не растягивается в отличие от обычного пробела). Создайте подпись к документу по образцу.



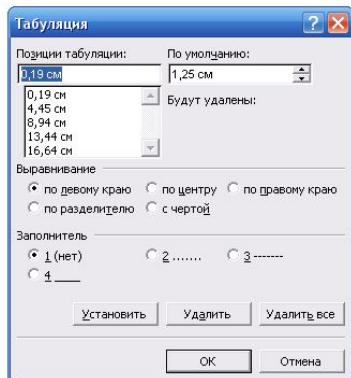
*!!! Абзац выровнен по ширине листа*

Сохраните файл в своей папке под именем **Расписка**.

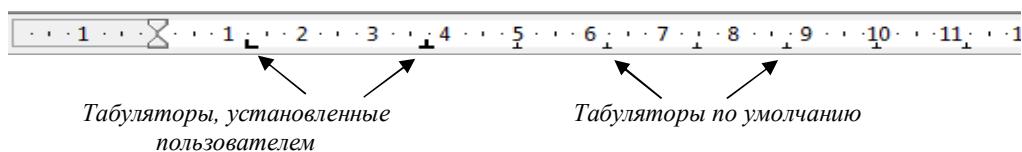
**Задание 14.** Используя полученные знания, создайте документ **Приглашение** по образцу с применением всех правил форматирования (рис. 34):



**Задание 15. Технология создания табулированных списков**



Координатная линейка может быть использована для создания, просмотра и удаления позиций табуляции. По умолчанию в документах Word позиции табуляции располагаются с интервалом 1,27 см. Эти позиции отмечаются штрихами в нижней части линейки. При определении позиции табуляции пользователем Word помещает на линейку более контрастные маркеры. Для позиции табуляции могут быть выбраны различные режимы выравнивания и заполнители в окне диалога **Табуляция**, которое вызывается по средствам диалогового окна Абзац → кнопка Табуляция.

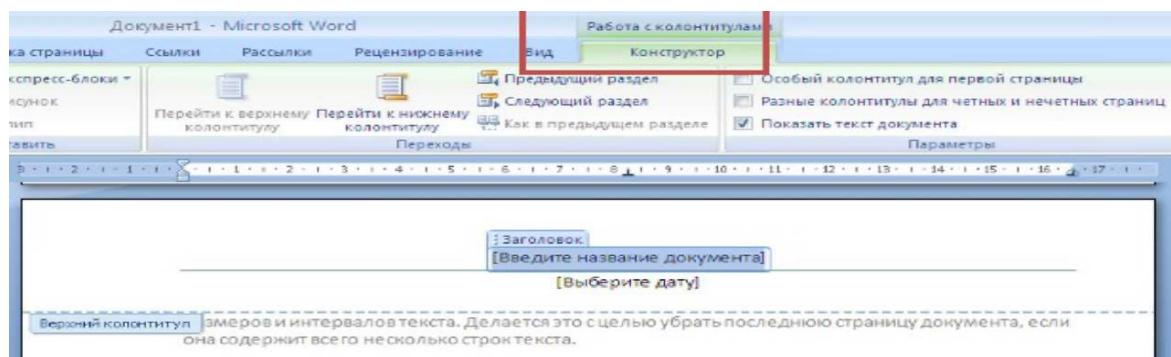


**Задание 16.** Создайте табулированный список по образцу, приведенному ниже. Обратите внимание на табуляторы установленные для списка на координатной линейке. Табуляторы проставляются вручную, щелчком мыши в области линейки. Начиная со второй строки списка, к табуляторам установлены заполнители в диалоговом окне **Табуляция**.

ФИО	Средний балл	Стипендия	Подпись
Иванов.....	4.....	1000.....	.....
Петров.....	5.....	1500.....	.....
Смирнова.....	4,5.....	1250.....	.....
Воронов.....	5.....	1500.....	.....

### Задание 26. Создание колонтитулов.

Для работы с колонтитулами предназначена панель *Колонтитулы* - вкладка *Вставка*. После



вставки колонтитул доступен для редактирования, при этом появляется контекстная вкладка *Конструктор (Работа с колонтитулами)*.

1. Выберите вкладку Вставка, в группе Колонтитулы выберите необходимый вам
2. В появившейся ниспадающем меню выберите вариант отображения колонтитула.
3. В появившейся области колонтитула введите текст: ФИО, факультет, специальность, курс, группа.

Быстрый переход между колонтитулами и основным текстом документа можно осуществлять двойным щелчком мыши на нужном элементе (верхнем/нижнем колонтитуле или на основном тексте).

Для удаления колонтитулов предназначен пункт *Удалить верхний/нижний колонтитул*



соответствующих кнопок колонтитулов.

Колонтитулы можно *редактировать* как обычный текст, применять различные стили для его оформления. Отредактированный колонтитул можно добавить в галерею колонтитулов при помощи опции *Сохранить выделенный фрагмент в коллекцию верхних/нижних колонтитулов*.

### **Задание 27. Нумерация страниц**

Для нумерации страниц служит кнопка *Номер страницы* (вкладка *Вставка* - панель *Колонтитулы*).

- Необходимо выбрать вариант размещения номера на самой странице и при необходимости настроить формат самого номера.
- При необходимости элементы номеров страницы можно сохранять, добавляя в коллекцию стандартных блоков. Для этого, вставив и настроив номер, нажмите кнопку *Номер страницы* и выберите команду *Вверху/внизу страницы* - *Сохранить выделенный фрагмент как номер страницы*.
- Чтобы убрать номер с первой страницы в документе, нужно войти в область колонтитулов и выбрать на вкладке *Конструктор* пункт *Параметры - Особый колонтитул для первой страницы*.

Установите нумерацию страниц в текстовом документе *Текст\_1*.

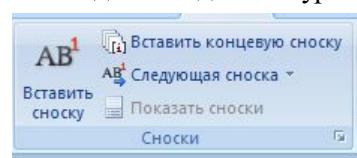
### **Задание 28. Применение в тексте сноски, ссылок, перекрестных ссылок, закладок в тексте.**

#### **Сноски**

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 вкладка *Ссылки* панель *Сноски*

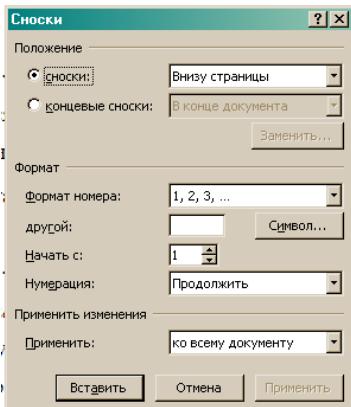
Для вставки обычной сноски необходимо:

1. Нажать кнопку *Вставить сноски*. В тексте, в том месте где находился курсор появится значок сноски, а внизу страницы - горизонтальная разделительная линия и номер сноски.



2. Для вставки концевой сноски предназначена кнопка *Вставить концевую сноsku*.

Для более точных настроек сносков служит диалоговое окно, вызываемое с 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**

Использование перекрестных ссылок полезно при указании ссылок на таблицы, заголовки, разделы, к которым нужно обратиться для получения более подробной информации. Они служат для быстрого перехода к нужному элементу.

Можно создавать перекрестные ссылки на следующие элементы: заголовки, сноски, закладки, названия, нумерованные абзацы, таблицы, рисунки, формулы. Чтобы создать перекрестные ссылки необходимо:

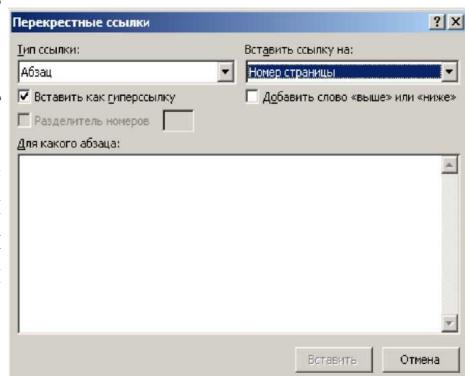
- Выбрать вкладку *Вставка* панель *Связи*.
- Нажмите кнопку *Перекрестная ссылка*,
- В открывшемся окне в выпадающем списке *Тип ссылки* надо выбрать тип элемента, на который будем ссылаться;
- В выпадающем списке *Вставить ссылку на...* надо указать данные, которые следует добавить в документ. Для того, чтобы иметь возможность перехода к ссылаемому элементу, флажок *Вставить как гиперссылку* должен быть установлен. Флажок *Добавить слово «выше» или «ниже»* позволяет вставить перед ссылкой данные слова.

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

### **Закладки**

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

- Установить курсор в нужном месте документа и на вкладке *Вставка* на панели *Связи* нажать кнопку *Закладка*.



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

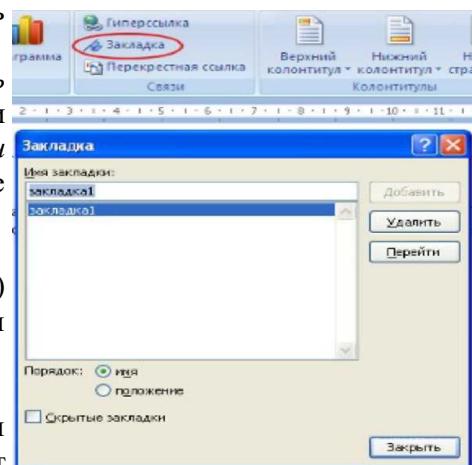
Перемещаться по закладкам, добавлять новые и удалять ненужные можно также при помощи этой же кнопки и окна, либо по нажатию клавиши F5 в окне *Найти и заменить* - *Перейти* - объект *Закладка* выбрать в списке нужную закладку.

#### **Добавление примечаний в документ**

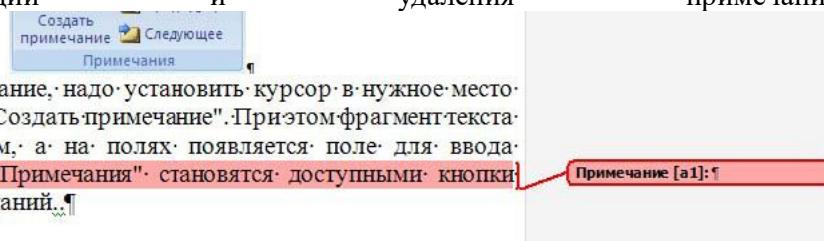
Для добавления (и последующего управления) примечаний предназначена панель *Примечания* вкладки *Рецензирование*.

Чтобы создать примечание, надо

- Установить курсор в нужное место документа и нажать кнопку *Создать примечание*. При этом фрагмент



текста выделяется красным цветом, а на полях появляется поле для ввода примечания, а на панели *Примечания* становятся доступными кнопки навигации и удаления примечаний.



#### **Задание 29. Создание к документу автоматического оглавления, предметного указателя, списка литературы и титульного листа.**

##### **Оглавление**

Для того чтобы быстро сделать оглавление, документ должен быть отформатирован согласно встроенным форматам уровней структуры или стилям заголовков. Для этого оформите все заголовки документа Текст\_1.doc (заголовки в тексте выделены полужирным шрифтом) стандартными стилями в соответствии с их уровнем (*Стили Заголовок 1, Заголовок 2, Заголовок 3*). Для чего выделить соответствующий фрагмент текста и выберет соответствующий стиль окне стилей.

Чтобы создать оглавление:

- установив курсор в месте вставки оглавления, на вкладке *Ссылки* на панели *Оглавление* нажмите кнопку *Оглавление*;
- в открывшемся окне выберите нужный формат оглавления (автособираемое или ручное). При необходимости дополнительных настроек оглавления, нажмите кнопку *Оглавление...* окна и настройте количество уровней оглавления, заполнитель, отображение и положение номеров страниц (рис. 4.8).

**Замечание.** По умолчанию в оглавление включаются текст, отформатированный стилями:

- Заголовок 1 (первый уровень оглавления),
- Заголовок 2 (второй уровень оглавления) и т. д. Для того, чтобы включить в оглавление заголовки других стилей и назначить им другие уровни, нужно, нажав кнопку *Параметры*, выполнить соответствующую настройку стилей уровням оглавления.

Для быстрой правки уже существующего оглавления сделайте щелчок в поле оглавления. Кнопка *Обновить таблицу* служит для обновления в оглавлении номеров страниц элементов.

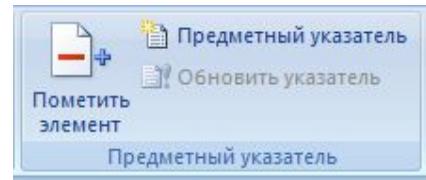
### **Предметный указатель встречающихся терминов**

Предметный указатель - это список терминов, 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* вкладки *Ссылки*.

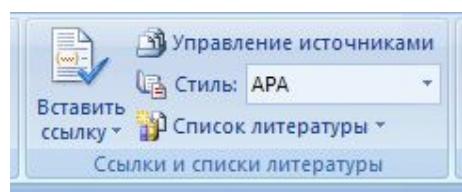
Чтобы использовать в качестве предметного указателя какой-либо фрагмент текста, его необходимо выделить, затем нажать кнопку *Пометить элемент*. При пометке текста в документе добавляется специальное скрытое поле.



Для окончательной сборки предметного указателя нажмите кнопку *Предметный указатель* и при необходимости в появившемся окне *Указатель* произведите окончательные настройки. Создайте в Тексте\_1 предметный указатель ссылок.

### **Список литературы**

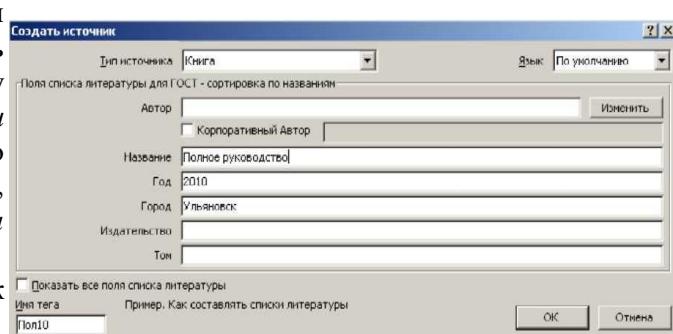
*Библиография* - это список литературных или других источников, которые использовались при подготовке документа. Как правило, она помещается в конце текста. При вставке цитат надо указывать источник, откуда они взяты, поэтому понятия «библиография» и литературы «цитаты» тесно взаимосвязаны. Для работы с библиографией и цитатами служит панель *Ссылки и списки литературы* вкладки *Ссылки*.



Для добавления нового источника нужно нажать кнопку *Вставить ссылку* панели и выбрать команду *Добавить новый источник*. В окне создания источника заполнить необходимые атрибуты.

После добавления источника ссылка на него будет помещена там, где был установлен курсор.

После завершения работы с документом список литературы можно создать автоматически, используя кнопку *Список литературы* на панели *Ссылки и списки литературы*. Перед этим можно просмотреть и откорректировать список, нажав кнопку *Управление источниками* данной панели.



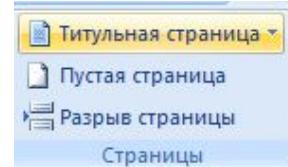
Создайте в Тексте\_1 Список литературы.

### **Титульный лист**

*Титульный лист* - первая страница, на которой указывают название работы, автора и другие необходимые сведения.

Для создания титульного листа существуют специальные заготовки. Для вставки в документ титульного листа

- Перейдите на вкладку *Вставка* и выберите *Титульная страница* в группе *Страницы*.
- В появившемся окне выберите нужный шаблон титульного листа.
- После вставки в документ выбранного листа, введите в



поля нужный текст, ненужные элементы удалите, щелкнув на них и два раза нажав клавишу *Delete*.

Чтобы удалить титульную страницу, выполните команду *Удалить текущую титульную страницу* в меню кнопки *Титульная страница*.

## 2.5 Лабораторная работа №5 (2 часа)

**Тема:** «Технология работы в текстовом процессоре Microsoft Word. Создание нумерованных и маркированных списков»

**Цель работы:** получить навыки оформления списков в документах, использования режима границ и заливки, использования готовых и создания новых стилей, использования табуляции и газетного стиля.

**Перечень приборов, материалов, используемых в лабораторной работе:**

1. Компьютеры с установленным программным обеспечением
2. Методические указания к выполнению лабораторной работы
3. Рабочая тетрадь.

**Описание (ход) работы:**

Абзацы можно преобразовывать в списки (нумерованные, маркированные, многоуровневые). Word предоставляет возможность задания различных вариантов границ и заливки абзацев. В Word реализованы два способа форматирования текста – прямое (или непосредственное) и форматирование с использованием стилей и шаблонов.

Стиль – это набор параметров форматирования, который формируется заранее и имеет уникальное имя. Стиль используется при форматировании символов и абзацев.

Загрузите текстовый процессор WORD

Списки

Ведите следующий текст, выделите:

*Понедельник*

*Вторник*

*Среда*

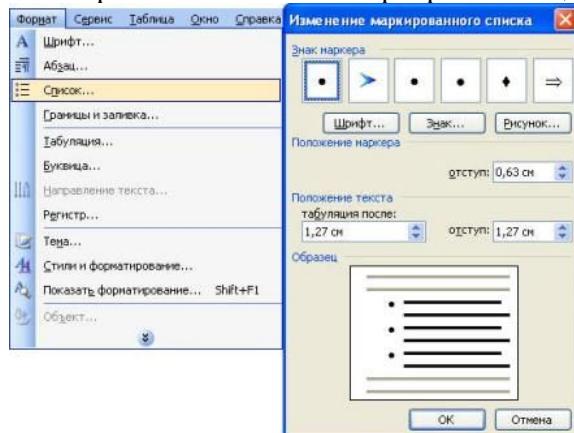
*Четверг*

*Пятница*

*Суббота*

Оформите введенный текст как Маркированный список, для этого:

- выделите весь текст
- выберите Формат• Список
- выберите закладку Маркированный
- выберите любой способ маркировки и подтвердите Ok



Измените способ маркировки (самостоятельно)

Измените символ маркировки, для этого:

- выберите Формат• Список
- выберите закладку Маркированный
- щелкните по кнопке Изменить
- в появившемся диалоговом окне щелкните по кнопке Маркер
- в таблице выберите любой символ и подтвердите свой выбор Ok

Снимите маркировку: Формат• Список, Маркированный, Нет

Оформите текст как Нумерованный список (самостоятельно)

Отредактируйте список, для этого:

- добавьте новую строку в список: установите курсор в конец любой строки и нажмите Enter
- введите любой текст
- просмотрите результат нумерации
- удалите введенную строку
- просмотрите результат нумерации

Результат показать преподавателю и снять нумерацию.

Добавьте с новой строки следующий текст:

*Математика*

*Химия*

*Литература*

*Физика*

*Этика*

Выделите школьные предметы, увеличьте отступ для этих строк (кнопка на панели форматирования)

Скопируйте выделенные строки, вставив их после каждого дня недели

Оформите весь текст как Многоуровневый список, для этого:

- выделите весь текст
- выберите Формат• Список
- выберите закладку Многоуровневый
- выберите любой способ оформления и подтвердите Ok

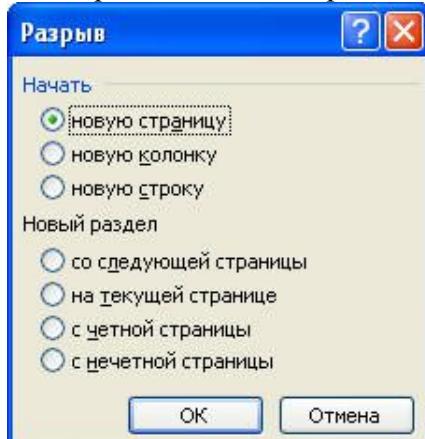
Результат показать преподавателю и снять нумерацию.

Выделите любые строки текста и оформите их маркированным или нумерованным списком, используя соответствующие кнопки на панели Форматирования (самостоятельно)

Границы и заливка

Вставить новую страницу в документ:

- поставьте курсор в пустой строке после всего текста
- выберите Вставка• Разрыв• Начать новую страницу



Введите нижеприведенный текст согласно образцу

**ЧТО НЕ НАДО ДЕЛАТЬ,  
ЕСЛИ КОМПЬЮТЕРА У ВАС НЕТ,  
НО ВЫ СОБИРАЕТЕСЬ ЕГО КУПИТЬ  
НЕ ПОКУПАЙТЕ КОМПЬЮТЕР БЕЗ ЦЕЛИ  
НЕ ПОКУПАЙТЕ ДЕШЕВЫЙ КОМПЬЮТЕР  
НЕ ПОКУПАЙТЕ ДОРОГОЙ КОМПЬЮТЕР  
НЕ ЭКОНОМЬТЕ НА МОНИТОРЕ**

Оформите заливку для первых трех строк, для этого:

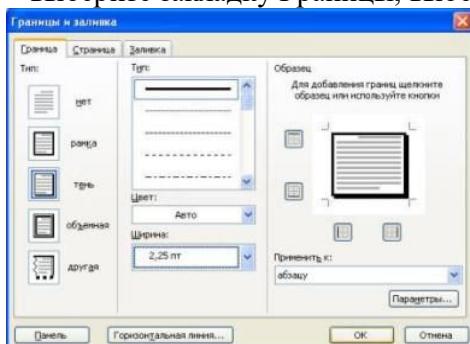
- выделите указанный текст
- выберите Формат• Границы и Заливка
- выберите закладку Заливка
- выберите любой способ заливки и подтвердите Ok
- снимите выделение с текста

Оформите границы для четвертой строки:

- выделите строку
- выберите Формат• Границы и Заливка
- выберите закладку Границы
- выберите тип Рамка и подтвердите Ok
- снимите выделение с текста

Оформите следующую строку:

- выделите строку
- выберите Формат• Границы и Заливка
- выберите закладку Границы, выберите тип Тень, измените Ширину (раскройте список)



- выберите закладку Заливка, выберите любой цвет, подтвердите Ok
- с помощью Горизонтальной линейки измените Отступ Слева и Справа для выделенной строки
- снимите выделение с текста

Остальные строки оформите по своему усмотрению, изменения Цвет, Тип и Ширину рамки, а также добавляя или убирая часть границ, используя Образец.

Результат показать преподавателю.

Отмените границы и заливку для всех строк, для этого:

- выделите весь текст
- выберите Формат• Границы и Заливка
- выберите закладку Границы, режим Нет
- выберите закладку Заливка, режим Нет
- подтвердите Ok

Оформите Границы для всей страницы, используя закладку Страница(самостоятельно)

Использование и создание стилей

Вставить новую страницу в документ.

Ведите текст по следующему образцу:

## Часть 1. Начальные сведения

### Глава 1. Что такое компьютер

Слово «компьютер» означает «вычислитель», т.е. устройство для вычислений. В основу работы компьютера заложен принцип фон Неймана. Компьютер может обрабатывать только информацию, представленную в числовой форме.

### Глава 2. Архитектура компьютера

Обычно персональные компьютеры состоят из трех основных частей:

Системного блока;

Клавиатуры;

Монитора.

В системном блоке располагаются все основные узлы компьютера: блок питания, процессор, оперативная память, контроллеры устройств, винчестер, дисковод и др.

Отформатируйте текст, используя встроенные стили, для этого:

- установите курсор в первой строке или выделите ее.
- выберите Формат• Стили и форматирование
- в списке выберите стиль Заголовок 1 и подтвердите Применить
- отформатируйте строки Глава 1 и Глава 2, используя стиль Заголовок 2 (самостоятельно)

Создайте свой стиль форматирования, для этого:

- выделите абзац между Главой 1 и Главой 2.
- отформатируйте его, используя панель форматирования: Размер 14, Цвет - синий, Отступ для первой строки, курсив, Выравнивание по ширине.
- выберите Формат• Стили и форматирование• Создать
- введите Имя для стиля (например Мой стиль)
- выберите стиль для абзаца (если он не установлен).
- подтвердите Ok и Применить

Отформатируйте текст после строки Глава 2, используя созданный стиль, для этого:

- выделите указанные строки
- выберите Формат• Стили и форматирование
- в списке выберите свой стиль и подтвердите: Применить

Результат показать преподавателю.

Измените созданный стиль, для этого:

- выделите любой фрагмент текста
- выберите Формат• Стили и форматирование
- в списке выберите свой стиль и щелкните по кнопке Изменить
- выберите Формат и измените некоторые параметры форматирования по своему усмотрению
- подтвердите Ok и Применить

Отформатируйте другой фрагмент текста встроенным стилем, используя список стилей на панели форматирования.

Удалите созданный стиль из списка стилей:

- выберите Формат• Стили и форматирование
- выделите свой стиль
- щелкните по кнопке Удалить и подтвердите свой выбор.

Сохраните документ под именем Практическая работа № 4\_1

## Табуляция

Создайте новый документ.

Подготовьте список студентов как табулированный текст

СТИПЕНДИАЛЬНАЯ ВЕДОМОСТЬ

№ группы      Фамилия И.О.

Стипендия      Подпись

31-----	Смирнов А.В.-----	200-----	-----
32-----	Соколов А.А.-----	200-----	-----
33-----	Иванов А.А.-----	250-----	-----
34-----	Иванов А.П.-----	250-----	-----

ИТОГО

Ключ к заданию:

Введите название документа шрифтом Times New Roman Cyr, 11 пт, разряженный интервал 5 пт, выровнивание по центру.

Выберите для последующих строк выравнивание по левому краю.

С помощью горизонтальной линейки задайте форматы табуляторов для абзаца с заголовками колонок. Для этого:

- выберите тип табулятора По левому краю слева на линейке;
- устанавливая указатель мыши в нужное место на линейке, щелкайте левую кнопку мыши.

Позиции табуляторов: 3,5 см. – выравнивание влево, 10 см. – выравнивание влево, 14 см. – выравнивание влево.

Введите заголовки колонок, используя клавишу Tab для перемещения курсора к следующей позиции.

С помощью команды Формат• Табуляция для абзацев списка задайте форматы табуляторов:

- 3,5 см. – выравнивание по левому краю, заполнитель – точки 2, Установить;
- 10 см. - выравнивание по левому краю, заполнитель – точки 2, Установить;
- 13 см. - выравнивание по левому краю, заполнитель – точки 2, Установить;
- 15,5 см. - выравнивание по левому краю, заполнитель – линия 4, Установить;
- остальные форматы табуляторов Удалить;
- нажать Ok.

Введите текст для каждой строки документа, используя клавишу Tab для перемещения курсора к следующей позиции.

Перейдите в абзац итоговой строки и снимите с горизонтальной линейки с помощью мыши табуляторы в следующих позициях: 3,5 см, 10 см, 15,5 см. Для этого надо подвести курсор к табулятору, нажать левую кнопку мыши и протащить мышь в левое поле.

Выделите абзацы списка и с помощью горизонтальной линейки и мыши перенесите табулятор из позиции 3,5 см в позицию 4 см.

Выделите абзацы списка и с помощью команды Формат• Табуляция измените тип линии заполнения для табуляторов в позициях 4 см, 10 см и 13 см на тип 3 (пунктир). Не забывайте для каждого табулятора после изменения нажимать кнопку Установить.

Результат показать преподавателю.

### Использование колонок

Вставить новую страницу в документ.

Подготовьте текст газетной статьи:

Создание и использование газетных колонок в среде текстового процессора Word  
Смирнов Андрей Анатольевич

Для текстов газетного типа выполняется набор в несколько колонок, после заполнения левой колонки (по высоте страницы или до установленного ограничения) курсор автоматически переходит в правую колонку. Текст колонок формируется по общим

правилам, внутри колонок можно вставлять кадры и размещать в них графику. Любые вставки или удаления текста и графики внутри колонок автоматически обеспечивают «перетекание» текста из колонки в колонку.

Ключ к заданию:

Введите название статьи – «Создание и использование газетных колонок в среде текстового процессора Word» и нажмите Enter.

Введите фамилию автора статьи – произвольную и нажмите Enter.

Выполните команду Формат• Колонки и задайте: Количество колонок – 3. Разделитель колонок – флажок. Применить – до конца документа.

Введите текст статьи – содержание нескольких абзацев справки по газетному стилю.

Для текстов газетного типа выполняется набор в несколько колонок, после заполнения левой колонки (до установленного разделителя раздела) курсор автоматически переходит в левую колонку.

Текст колонок формируется по общим правилам, внутри колонок можно вставлять кадры и размещать  в  них  графику.

Любые вставки или удаления текста и графики внутри колонок автоматически обеспечивают «перетекание» текста из колонки в колонку.

– нажмите Enter.

Вставьте разрыв раздела Для этого:

- установите курсор в первой пустой строке после текста;
- выполните команду Вставка• Разрыв и в диалоговом окне выберите Новый раздел на текущей странице. Вставляемый разрыв должен обеспечивать расположение текста статьи равномерно в три колонки.

На панели Стандартная нажмите кнопку Непечатаемые символы  и найдите в тексте разделители разделов: =====. При правильном форматировании колонок в тексте присутствуют два разделителя разделов документа. Всего в документе три раздела.

Установите в последнем разделе печать в одну колонку. Для этого:

- установите курсор после второго разделителя разделов;
- выполните команду Формат• Колонки и задайте: Количество колонок – 1. Применить – к текущему разделу.

Введите заключительный текст статьи – список литературы.

Установите курсор в колонки с текстом и выполните команду Формат• Колонки, указав: Количество колонок: 2, Разделитель колонок – флажок. Применить – к текущему разделу.

Задайте буквицу:

- установите курсор в первую строку первой колонки и выполните Формат• Буквица;
- выберите положение В тексте;
- высота в строках – 2;
- расстояние от текста - 0,5;
- нажмите кнопку ОК.

Результат показать преподавателю.

## 2.6 Лабораторная работа №6 (4 часа)

**Тема:** «Технология работы с таблицами в Microsoft Word. Ввод и редактирование формул»

**Цель работы:** изучить правила создания и удаление таблиц, научиться редактировать таблицы и работать с данными таблицы в приложении MS Word.

**Задачи работы:**

1. Научиться создавать и удалять таблицы в MS Word.
2. Изучить правила редактирования таблиц.
3. Изучить правила использование формул в таблицах и вычисления в документах MS Word.
4. Научится работать с данными таблицы.

**Перечень приборов, материалов, используемых в лабораторной работе:**

1. Компьютеры с установленным программным обеспечением
2. Методические указания к выполнению лабораторной работы

### 3. Рабочая тетрадь.

#### Описание (ход) работы:

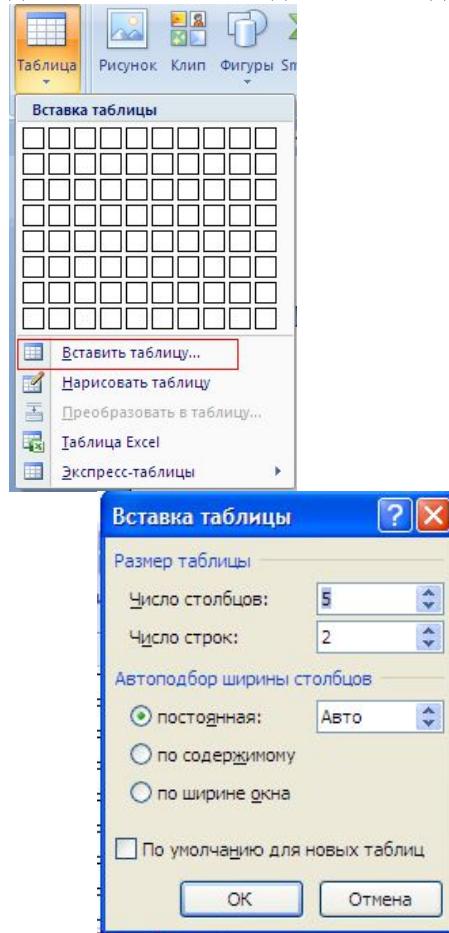
**Таблица** представляет собой набор ячеек, расположенных в виде строк и столбцов. Каждая ячейка таблицы может содержать текстовую, числовую или графическую информацию. Можно производить любые манипуляции с данными таблицы, например, вставлять и удалять, форматировать данные, не нарушая при этом расположения столбцов.

#### Создание таблицы

Для вставки таблицы служит кнопка *Таблицы*, расположенная на панели *Таблицы* вкладки *Вставка*. При нажатии на эту кнопку можно в интерактивном режиме выбрать необходимое количество строк и столбцов для будущей таблицы

Если таблица очень большая и количество предлагаемых нужно воспользоваться опцией *Вставить таблицу* (рис. 39) в диалоговом окне задать необходимое количество строк и

ячеек недостаточно, 6.2) и в появившемся столбцов

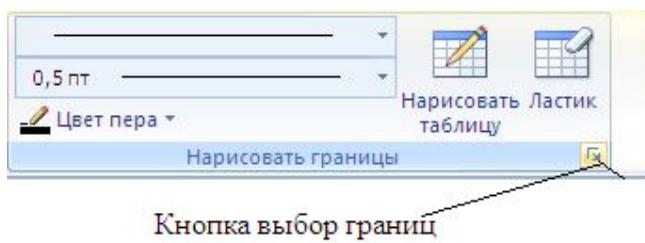


#### Создание таблицы

**Задание 17.** Создайте таблицу из **5 столбцов** и **6 строк** через диалоговое окно **Вставка таблицы**, заполните шапку таблицы, как на рис. 39, задайте границы таблицы.

№ п/п	Фамилия	Имя	Оценка	Стипендия

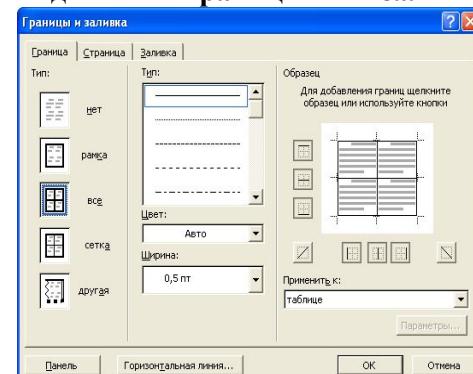
**Образец Таблицы 1.**



#### *Отображение и скрытие линии сетки, внешние границы*

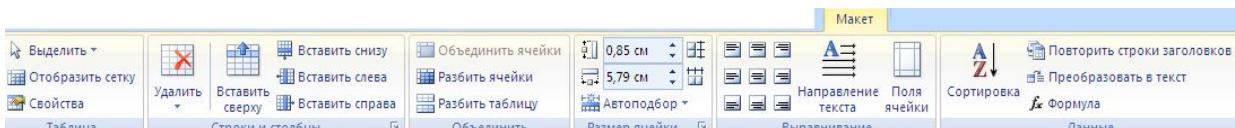
По умолчанию режим отображения рамки таблицы и разделительных линий включен. Тем не менее, даже если линии сетки будут удалены с использованием команды Границы и заливка, Word по-прежнему будет отображать линии

сетки, разделяющие ячейки таблицы, эти линии не отображаются на печати. Для добавления границ и заливки в таблицах используется окно диалога Границы и заливка открывющееся из контекстной вкладки Конструктор → панель Нарисовать таблицу → кнопка Выбор границ или средства работы с границами и заливкой, расположенными во вкладке Конструктор → панель Стили таблиц → Границы → Границы и заливка (рис.40).



#### **Задание 18. Работа с таблицей.** По завершении создания таблицы в нее необходимо ввести текст.

Текст вводится в каждую из ячеек таблицы в отдельности, причем для его форматирования подходит большинство стандартных функций форматирования Word. Для перемещения по ячейкам таблицы используют клавиши управления курсором, манипулятор мыши и специальные клавиши (Tab). При оформлении таблицы пользователь может установить свойства выбранной таблицы, например, выравнивание, интервалы, ширину столбцов,



границы и заливку через контекстную вкладку Макет.

#### **Вкладка Макет**

Изучите возможности, предоставляемые контекстными вкладками Конструктор и Макет.

#### **Задание 19. Изменение размеров строк и столбцов. 3Выравнивание текста и формирование шрифта в ячейках таблицы.**

Наиболее простым способом изменения размеров столбцов и строк является перетаскивание их границ вручную. Нужно подвести указатель мыши к линии, разделяющей два столбца и в тот момент, когда указатель примет вид двойной стрелки, нажать левую клавишу мыши и перенести границу столбца в сторону (рис. 42.).

№ п/п	Фамилия	Имя
		Настроить столбец таблицы

**Рис. 42. Настройка ширины столбцов**

Для указания числовых значений размеров столбца или строки используется вкладка Макет → Размер ячейки, здесь же пользователь может установить **автоподбор** ширины и высоты ячеек

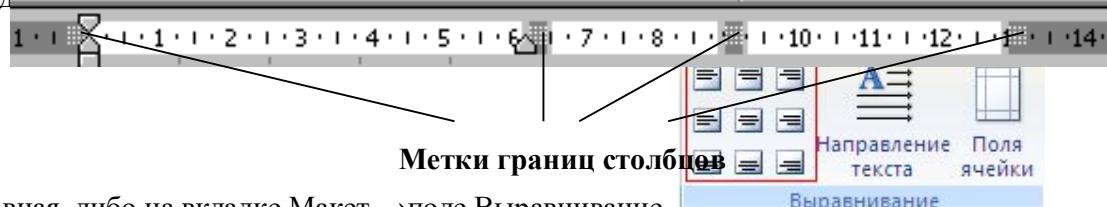
Для изменения ширины столбцов таблицы можно воспользоваться горизонтальной линейкой. На горизонтальной линейке располагаются метки границ столбцов таблицы (рис. 6.). Установите курсор на метку, и когда появятся стрелки изменения размеров, нажмите левую кнопку мыши и, удерживая ее, переместите границу ячейки таблицы.

**Рис. 43. Управление шириной столбцов при помощи координатной линейки**

Сузьте столбцы, в котором введены № п/п, оценка и стипендия и расширьте столбец, в котором введена фамилия.

Перед тем, как начать форматирование данных таблицы, ячейки необходимо выделить. Пользуясь вкладкой Макет → поле Таблица → Выделить, можно выделить ту строку или столбец, в которых предварительно установлен курсор, а если для выделения хотите воспользоваться мышью - подведите указатель мыши левее строки или выше столбца и щелкните мышью в тот момент, когда указатель примет форму стрелки, указывающей на выделяемый участок. Выделите первую строку таблицы, заполненную текстом.

Выравнивание текста и форматирование шрифта в ячейках таблицы производится традиционными способами с использованием инструментов, расположенных на вкладке



Главная, либо на вкладке Макет → поле Выравнивание.

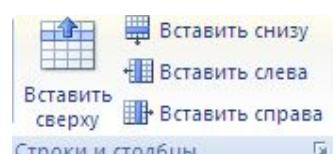
**Рис. 44. Выравнивание текста в ячейке**

Установите для первой строки таблицы выравнивание по горизонтали и вертикали **по центру**. Для первой строки таблицы установите шрифт **Times New Roman, размер 14, жирный**.

#### **Задание 20. Добавление и удаление строк и столбцов**

Для вставки строки/столбца в имеющуюся таблицу необходимо установить курсор в строку или столбец перед или после которых необходимо добавить строку/столбец. На вкладке Макет выбрать поле Строки и столбцы и добавить нужный элемент

Чтобы добавить строку в конце установить курсор в последней правом нижнем углу таблицы, и



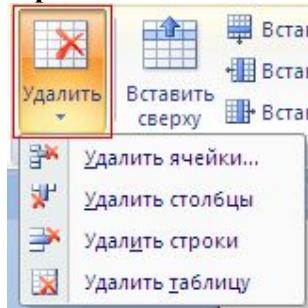
таблицы, необходимо ячейке, расположенной в нажать клавишу Tab.

Добавьте **один** столбец после столбца **Имя**. Внесите заголовок столбца **Отчество**. Заполните таблицу данными как на рис. 45. Столбец № **n/n** заполните при помощи нумерованного списка, выбрав его на вкладке главная в поле Абзац. Добавьте в конце таблицы одну пустую строку.

№ п/п	Фамилия	Имя	Отчество	Оценка	Стипендия
1.	Иванов	Андрей	Алексеевич	5	1250
2.	Сергеева	Мария	Ивановна	4	1000
3.	Воронов	Петр	Ильич	4	1000
4.	Коротков	Евгений	Семенович	3	0
5.	Андреева	Вера	Петровна	5	1250
6.					

### Заполненная таблица

Удаление строки или столбца осуществляется командой **Удалить строки/столбцы/ячейки** на вкладке Макет



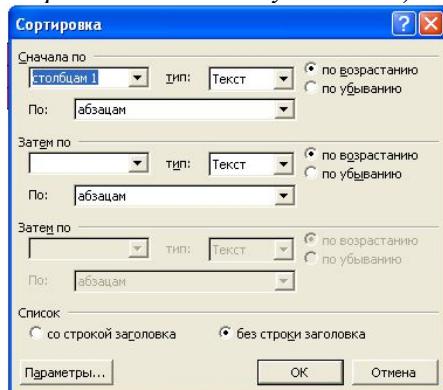
### Задание 21. Сортировка данных в таблицах.

Выполнить сортировку по возрастанию или по убыванию позволяют кнопка **Сортировка по**



**возрастанию/убыванию**, расположенная на панели Главная в поле Абзац . Сортировать можно как числовые, так и текстовые данные.

В окне диалога **Сортировка**, открывающегося вкладке Макет поле Данные можно установить параметры для сортировки данных, используя до трех полей таблицы, указать тип сортировки (**текст, число, дата**) и порядок сортировки для каждого поля (**по возрастанию или по убыванию**).



Диалоговое окно Сортировка

Проведите сортировку по возрастанию ранее созданной таблицы по полю **Фамилия**. В открывшемся диалоговом окне **Сортировка** установите *первый ключ: Столбец – 2, Порядок – по возрастанию*.

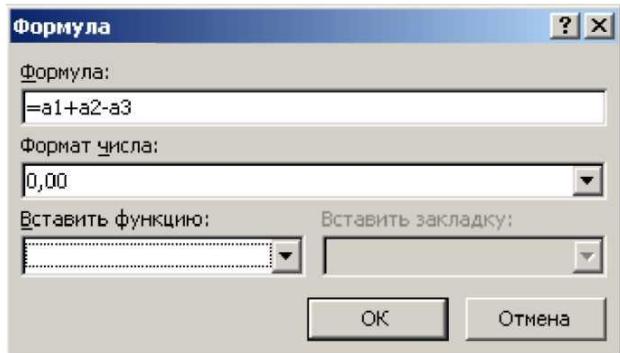
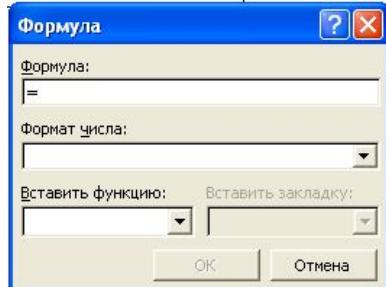
*Внимание! Если данные таблицы не выделить, а просто установить курсор в пределах сортируемого поля, то сортировка произойдет без привязки остальных данных таблицы.*

## ИСПОЛЬЗОВАНИЕ ФОРМУЛ В ТАБЛИЦАХ И ВЫЧИСЛЕНИЯ В ДОКУМЕНТАХ

**Задание 22.** Некоторые таблицы, создаваемые в Word предназначены для хранения числовых данных. Для выполнения вычислений над этими данными можно включать в таблицы формулы.

Для того чтобы использовать формулу необходимо:

- Установить курсор мыши в ячейку, где должен быть получен результат вычислений
- Выбрать вкладку **Макет – поле Данные →Формула**. При этом на экране появится окно диалога **Формула** (рис. 47).



- В списке **Вставить функцию** выбрать нужную функцию (выбранная функция добавится в поле **Формула**).
- В скобки введите направление расположения подсчитываемых ячеек (сверху- above, слева- left, справа-right).
- При необходимости можно выбрать формат отображения результатов вычисления в списке **Формат числа**.
- В результате Word вставит поле, содержащее формулу, в ячейку таблицы и отобразит результат вычисления.

**Также можно произвести подсчет следующим образом:**

Чтобы задать в ячейке формулу, нужно выполнить следующие действия:

- поместить курсор в ячейку таблицы или в то место документа, где должен выводиться результат;
- выполнить команду *Данные -Формула* ленты *Макет*;
- в появившемся окне **Формула** (рис. 48) в строке формулы

записать формулу, указать формат вывода результата в строке формата числа, при необходимости выбрать функцию в поле вставки функции и нажать *OK*.

**Замечание.** Диапазон ячеек указывается через двоеточие от верхней левой ячейки до нижней правой.

Вычислите средний балл студентов и общую сумму стипендии в созданной ранее таблице. Для чего установите курсор в пустую ячейку под полем **Оценка**, выберите вкладку **Макет – поле Данные →Формула**. В окне диалога Формула установите курсор в строку Формула, затем выберите функцию (SUM, AVERAGE).

**Задание 23. Объединение и разбиение ячеек**

Для объединения нескольких ячеек в одну, необходимо выделить объединяемые ячейки, а затем выбрать вкладку **Макет** → поле **Объединить ячейки**. Для разделения ячеек с использованием меню необходимо выделить одну или несколько ячеек и выбрать вкладку **Макет** → поле **Объединить** → **Разбить ячейки**. В появившемся окне диалога указать, на сколько столбцов или строк следует разделить выделенную ячейку.

Объедините первые четыре ячейки таблицы в последней строке. В объединенные ячейки введите текст: **Итоговые значения**. Подсчитайте общую сумму стипендии и среднюю оценку. В результате Вы должны получить таблицу как на рис. 49.

**Таблица 2.**

№ п/п	Фамилия	Имя	Отчество	Оценка	Стипендия
1.	Иванов	Андрей	Алексеевич	5	1250
2.	Сергеева	Мария	Ивановна	4	1000
3.	Воронов	Петр	Ильич	4	1000
4.	Коротков	Евгений	Семенович	3	0
5.	Андреева	Вера	Петровна	5	1250
<b>Итоговые значения:</b>				<b>4,2</b>	<b>4500</b>

**Рис. 49. Готовая таблица**

Сохраните созданную таблицу в своей паке под именем **Таблица 2**

Самостоятельно создайте таблицу (рис. 50) по образцу, подсчитайте столбец *Средний балл*.

**Таблица 2.**

№ п/п	ФИО	Предметы				Средний балл
		История	Экономика	Психология	Информатика	
1	Иванов А.П.	3	4	5	4	
2	Петров И.В.	3	3	5	5	
3	Сидоров А.К.	4	4	5	5	
4	Смирнов В.Г.	3	3	3	5	
<b>Общий средний балл группы</b>						

**Рис. 50. Вычисление среднего значения**

**Задание 24. Добавление названия к таблице.** Если в документе предполагается наличие множества таблиц, то для удобства пользователей существует возможность создания автоматической подписи к таблице с соответствующей перенумерацией при добавлении новых таблиц в документ.

Для добавления названия к таблице, выделите таблицу и выберите команду *Вставить названия* панели *Названия* вкладки *Ссылки*. В появившемся окне *Название* (рис. 51) укажите текст подписи, ее положение относительно таблицы. Если вы хотите, чтобы подпись к таблице добавлялась автоматически при вставке новых таблиц в документ, то используя кнопку *Автоназвание* настройте параметры подписи и укажите, что ее следует добавлять для всех объектов типа «Таблица Microsoft Word» (рис.52 ).

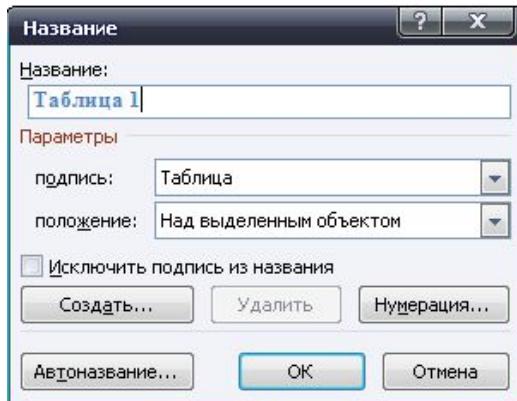
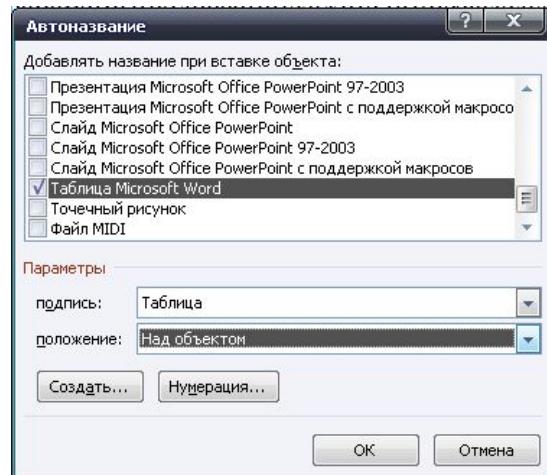


Рис. 51. Диалоговое окно название

Рис. 52. Вставка Автоназвания таблицы



## РАБОТА С ДАННЫМИ ТАБЛИЦЫ

### *Задание 25. Преобразование таблицы в текст.*

При возникшей необходимости получить из имеющейся таблицы структурированный текст, воспользуйтесь командой Данные - Преобразовать в текст ленты Макет. В итоге будет получен текст, разделенный в местах разделения ячейками таблицы выбранными разделителями.

#### *Преобразование текста в таблицу*

Можно превращать уже набранный текст в таблицу. Для этого необходимо

- Выделить нужный блок текста и выбрать опцию Преобразовать в таблицу кнопки Таблица.
  - В появившемся окне надо задать параметры будущей таблицы.
- Следует иметь ввиду, что фрагмент текста должен быть предварительно отформатирован символами-разделителями (например, табуляцией или абзацем), чтобы программа смогла различить ячейки таблицы.
- Задание для развития и контроля владения компетенциями:**
1. Какими способами можно создавать и удалять таблицы в MS Word.
  2. Какие существуют способы редактирования таблицы.
  3. Как произвести вычисления в документах MS Word в таблице используя формулы.
  4. Как добавить название к таблице.

## 2.7 Лабораторная работа №7 (2 часа)

### **Тема: «Создание гипертекста»**

**Цель работы:** Узнаете что такое гиперссылка; Научитесь связывать текст гиперссылками, изменять, удалять гиперссылки.

#### **Перечень приборов, материалов, используемых в лабораторной работе:**

1. Компьютеры с установленным программным обеспечением
2. Методические указания к выполнению лабораторной работы
3. Рабочая тетрадь.

#### **Описание (ход) работы:**

**Гиперссылка** - это выделенные области документа, позволяющие переходить к другому документу, содержащему связанную информацию. Простейшим примером может служить энциклопедия, в которой в каждой статье встречаются ссылки на другие статьи.

**Гиперссылки** - это наиболее из рациональных способов организации больших текстов, в которых разным пользователям могут быть интересны разные части.

**Задание «Пример создания гиперссылки».**

1. Набрать следующие тексты, сохраняя их в отдельных файлах. Имена файлов соответствуют названиям текстов.

### **Пряноароматические овощи**

Такие овощи выращиваются в огороде и используются для приготовления блюд, чтобы придать им аромат. Биологическая ценность таких овощей не столь важна, однако многие из них обладают лечебными свойствами. К наиболее распространенным видам пряноароматических овощей, которые можно вырастить в огороде, относятся укроп, анис, мелисса, мята, фенхель и т.д.

### **Мелисса**

Лимонный привкус и аромат мелиссы лекарственной определят сферу использования этого растения: там, где требуется лимонная цедра или лимонный сок. В листьях мелиссы содержатся эфирные масла и дубильные вещества. С помощью мелиссы ароматизируют уксус, майонез, мясные блюда, салаты, супы и соусы. Лечебные свойства у мелиссы аналогичны мяте: мелисса предотвращает вздутие кишечника, снижает давление и снимает мигрень. Из свежих и сушеных листочек готовят превосходный освежающий чай.

### **Анис**

Анис используется в медицине многие тысячелетия. Семена или эфирные масла аниса - составной компонент лекарств против кашля и простуды. В значительной степени он употребляется при изготовлении кондитерских изделий, печения, при консервировании овощей и производстве ликеров. Аналогичными свойствами обладает и фенхель, который легче выращивать, а по своим качествам он не уступает анису.

Сушеные листья мяты перечной используют при приготовлении самых различных блюд. Она вызывает аппетит, устраниет пучение живота, оказывая тем самым **Анис**

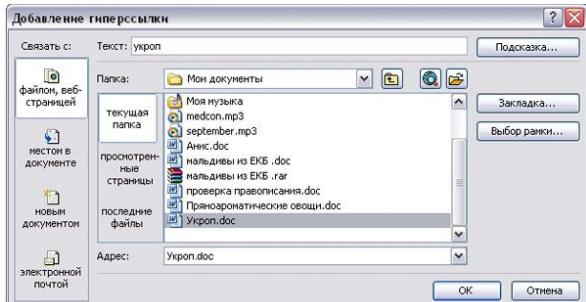
Анис используется в медицине многие тысячелетия. Семена или эфирные масла аниса - составной компонент лекарств против кашля и простуды. В значительной степени он употребляется при изготовлении кондитерских изделий, печения, при консервировании овощей и производстве ликеров. Аналогичными свойствами обладает и фенхель, который легче выращивать, а по своим качествам он не уступает анису.

благотворное воздействие на весь пищеварительный тракт, а также устраниет желудочные боли. Мята используется в виде лечебного чая (отваров) прежде всего при заболеваниях печени и желчного пузыря.

### **Фенхель**

Фенхель (укроп аптечный) - распространенная приправа. Помимо использования в домашнем хозяйстве, он применяется в пищевой промышленности при выпечке печения и кондитерских изделий. Эфирные масла, содержащие в фенхеле, благотворно действуют на пищеварительный тракт.

2. Нужно связать эти тексты в гипертекст. Для этого в первом тексте выделить слово "укроп". В меню **Вставка** выбрать команду **Гиперссылка**. На экране появится окно диалога **Добавление гиперссылки**, в котором нужно найти необходимый файл.



3. Щелкнуть по кнопке **OK**, и слово "укроп" изменит цвет и станет подчеркнутым.
  4. Повторить эти действия для слов: анис, мелисса - из первого документа, а затем для названия овощей, встречающихся в других документах. Названия каждого овоща связывайте с названием соответствующего файла.
  5. После того, как все названия овощей окажутся выделенными цветом и подчеркиванием, установить курсор на любой из них. Если курсор изменит свою форму и станет похож на правую руку с вытянутым указательным пальцем, значит, все сделано правильно. Задержите курсор на слове "укроп", рядом появится табличка с полным именем файла, который связан с этим словом. При щелчке по этому слову загрузится файл Укроп.doc.
  6. В загрузившемся файле появится панель инструментов Веб-узел. Вернитесь к исходному тексту, щелкнув по кнопке со стрелкой **Назад**. Слово "укроп" изменило цвет. Это означает, что эту ссылку просмотрели.
  7. Выполните п.4-5 с другими словами.
- Удаление или изменение гиперссылки**  
 Щелкнуть по гиперссылке правой кнопкой мыши и выбрать в контекстном меню команду **Гиперссылка**, в сплывающем меню выбрать команду **Изменить гиперссылку**. На экране появиться диалоговое окно **Изменение гиперссылки**. Для изменения нажмите кнопку **Удалить ссылку**.

## 2.8 Лабораторная работа №8 (4 часа)

**Тема:** Создание графических объектов в текстовом процессоре Microsoft Word»

**Цель работы:** изучить возможности работы с графикой в текстовых документах MS Word

**Задачи работы:**

1. Научиться работать с графическими объектами.
2. Изучить правила работы с графическими фигурами.
3. Научиться работать с объектами WordArt, SmartArt.
4. Изучить правила создания Диаграмм в приложении MS Word.

**Перечень приборов, материалов, используемых в лабораторной работе:**

1. Компьютеры с установленным программным обеспечением
2. Методические указания к выполнению лабораторной работы
3. Рабочая тетрадь.

**Описание (ход) работы:**

Microsoft Word предоставляет достаточно большие возможности работы с графикой. Наличие графических объектов в текстовых документах часто желательно, а в некоторых случаях просто необходимо.

Основные способы включения графики в документ - импорт графики из других приложений или создание графических объектов непосредственно в документе. Основные инструменты для работы с графикой находятся на ленте *Вставка* панель *Иллюстрации*.

**Задание. Вставьте графический объектов в документ из других приложений, из файла, из коллекции.**

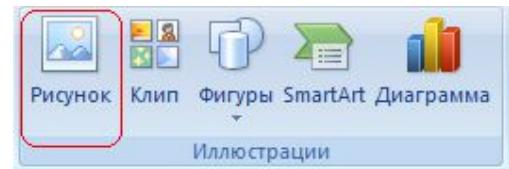
## РАБОТА С ИЗОБРАЖЕНИЯМИ

### **Вставка изображений из других приложений**

Графические объекты из других приложений в документ Word можно вставить, используя буфер обмена. Для этого нужно скопировать картинку из любого источника - веб-страницы, другого документа, другого приложения, а потом вставить из буфера обмена в нужное место текущего документа.

### **Вставка рисунков из файла**

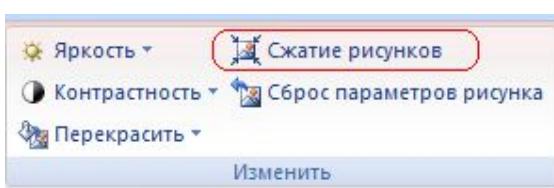
Для вставки рисунка из имеющегося графического файла, необходимо воспользоваться кнопкой *Рисунок* панели *Иллюстрации* на вкладке *Вставка*.



В появившемся окне найдите и выберите нужный графический файл. Изображение вставится в документ. **Замечание.** Следует учитывать, что вставленное изображение зачастую занимает значительный объем памяти. Поэтому выполнение некоторых операций будет занимать определенное время, причем, оно будет тем больше, чем больше размер вставляемого файла и ниже производительность компьютера. Чтобы работа с изображениями была более удобной, а итоговый размер текстового документа не был очень большим, целесообразно сделать компрессию изображения. Для этого на вкладке *Формат* на панели *Изменить* выбрать кнопку *Сжатие рисунка*.

### **Вставка рисунков непосредственно в программе из коллекции.**

Для работы с которой служит область задач *Клип*. Данная коллекция содержит подборку набора картинок текстового редактора.



Для вставки клипа необходимо нажать кнопку *Клип* на панели *Иллюстрации* вкладки *Вставка*. Справа появится панель *Клип* с кнопкой *Упорядочить клипы...*, нажав на которую мы попадем в окно *Организатор клипов*. В данном окне слева будет находиться каталог клипов, а справа - область просмотра выбранного раздела каталога.

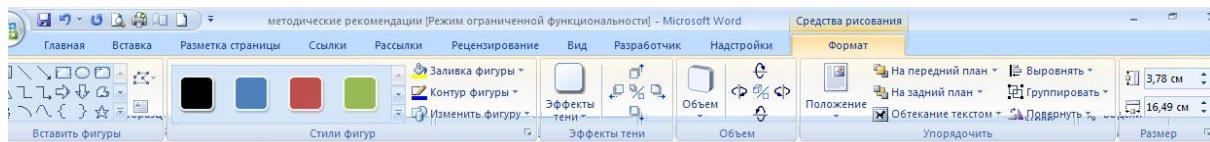
По умолчанию доступны три коллекции:

- Мои коллекции (в нее входят все папки жесткого диска, содержащие картинки);
- Коллекции Microsoft Office (собрание изображений, которое входит в поставку MS Office);
- Веб-коллекции (позволяет расширить собрание картинок Word при помощи изображений, доступных на интернет-ресурсе Office Online).



### **Редактирование изображений**

Для изменения каких-либо параметров изображений (рисунков), нужно выделить вставленное изображение, при этом появится новый контекстный инструмент *Средства рисования*, содержащий вкладку *Формат* с инструментами для обработки изображения.



С их помощью можно производить несложные операции редактирования рисунка - изменять яркость, контрастность, размер, вращать, выбирать стиль для рисунка (можно задать его форму, цвет границы, а также эффекты), указывать положение относительно текста.

1. Чтобы изменить яркость, контрастность, перекрасить рисунок в определенный цвет (например, сделать его менее ярким, чтобы использовать в качестве фона), на панели *Изменить вкладки Format* (Работа с рисунками) выберите соответствующие пункты.

2. Чтобы задать стиль оформления, изменить форму рисунка, задать вид его границ и эффекты (тень, отражение, свечение, сглаживание, рельеф, поворот), используйте инструменты с панели *Стили рисунков* вкладки *Format*. Также для оформления рисунков по нажатию правой кнопки мыши можно вызвать контекстное меню и выбрать кнопку *Format рисунка*.

3. Чтобы отменить все исправленные параметры на панели *Изменить* выберите кнопку *Сброс параметров рисунка*.

4. Чтобы задать нужный размер рисунка, можно, выделив его, изменить размер вручную, либо задать точные значения размера на панели *Размер*. На этой же панели доступна кнопка *Обрезка*, которая позволяет обрезать рисунок с каждой стороны. Важно учитывать, что Word не удаляет обрезанную часть рисунка, а просто перестает ее отображать.

5. Чтобы повернуть/отразить рисунок, используйте кнопку *Повернуть* панели *Упорядочить*.

6. Чтобы сгруппировать несколько рисунков в один (для более удобной работы с множеством изображений), используйте кнопку *Группировать* на панели *Упорядочить*.

7. Чтобы распределить графические объекты относительно друг друга и страницы, используйте кнопку *Выровнять*, *На задний план*, *На передний план* панели *Упорядочить*.

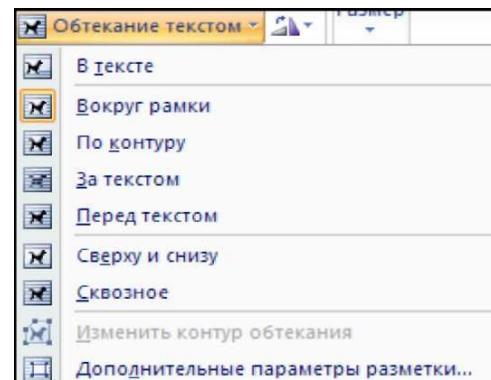
8. Чтобы отобразить сетку, которая позволяет более точно распределять объекты на странице, выберите пункт *Отображать сетку* кнопки *Выровнять* панели *Упорядочить*. Нажав кнопку *Параметры сетки*, можно настроить необходимые параметры работы с сеткой (рис 7.5). Так, объекты, привязанные к сетке, будут передвигаться только по соответствующим клеткам сетки и располагаться в них.

9. Для настройки расположения графического объекта относительно текста (обтекания), используйте кнопку *Обтекание текстом* панели *Упорядочить*. По умолчанию программа устанавливает режим обтекания «В тексте», при котором рисунок «разрывает» текст. Можно выбрать любой другой подходящий вариант обтекания:

- *Вокруг рамки* - рисунок вписан в прямоугольник, текст обтекает его по рамке этого прямоугольника;
- *По контуру* - текст обтекает рисунок по его границе;
- *За текстом* - рисунок будет играть роль фонового изображения;

- *Перед текстом* - рисунок будет помещен над текстом и закроет собой его часть;

- *Сверху и снизу* - текст будет располагаться выше и ниже рисунка;



При необходимости можно изменить контур обтекания текстом. Для этого выберите соответствующую команду меню кнопки *Обтекание текстом*. При этом вокруг объекта появится рамка с маркерами, перемещая которые, можно будет изменить контур.

Также можно задать готовый вид обтекания, *Положение панели Упорядочить*.

## РАБОТА С ФИГУРАМИ

Фигурами являются всевозможные простые готовые изображения (графические примитивы), которые можно вставлять в документ. Они являются объектами векторной графики, к ним можно применить множество визуальных эффектов.

### ***Создание графического примитива***

Кнопка *Фигуры* панели *Иллюстрации* вкладки *Вставка* служит для быстрого создания графических примитивов. Для создания нужного примитива выберите его из выпадающего списка и «нарисуйте» в документе протяжкой мыши с нажатой левой кнопкой.

#### *Замечания.*

- Для того чтобы фигура имела одинаковые пропорции, во время рисования надо удерживать нажатой кнопку Shift. При нажатой клавиши Ctrl фигура будет нарисована «от центра».
- При создании фигур можно использовать полотно, которое позволяет размещать на нем графические объекты и перемещать, вращать, удалять сразу все объекты, размещенные на нем. Чтобы создать полотно, выберите команду *Новое полотно* из меню кнопки *Фигуры* (вкладка *Вставка-Иллюстрации*).

При выделении фигуры по краям появляется рамка с маркерами. *Синие круглые маркеры* позволяют пропорционально изменить размеры фигуры, *синие* - непропорционально. *Желтый ромбовидный маркер* также служит для изменения геометрических размеров фигуры. Фигуру можно вращать, для этих целей служит *зеленый круглый маркер*, расположенный над фигурой. Для вращения примитива необходимо установить курсор мыши на маркер и, нажав левую кнопку, производить движения мышью. При этом фигура будет вращаться в ту или иную сторону.

### ***Форматирование фигур***

Когда фигура нарисована и выделена, появляется контекстный инструмент *Средства рисования* с лентой *Формат*.

1. Чтобы добавить новые автофигуры, используйте кнопки панели *Вставить фигуры* данной ленты.

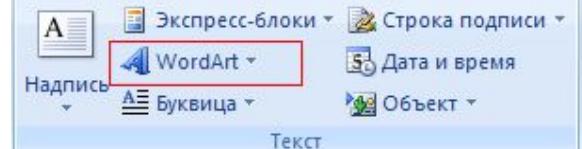
2. Чтобы задать для фигуры нужный цвет, обрамляющий контур, изменить форму уже отформатированной фигуры или выбрать оформление фигуры из имеющихся образцов, используйте соответствующие кнопки панели *Стили фигур*. Вызываемое с этой же панели окно *Формат автофигуры* содержит расширенные параметры форматирования фигур.

3. Чтобы применяйте *Эффекты* настройки расположенные в правой части данной панели. Кнопка *Объем* позволяет применить трехмерные эффекты к фигуре. При этом можно настраивать такие параметры как: *Цвет объемной фигуры*, *Глубина*, *Направление*, *Освещение*, *Поверхность*. Для интерактивной настройки объема служат кнопки, расположенные в правой части панели *Объем*.

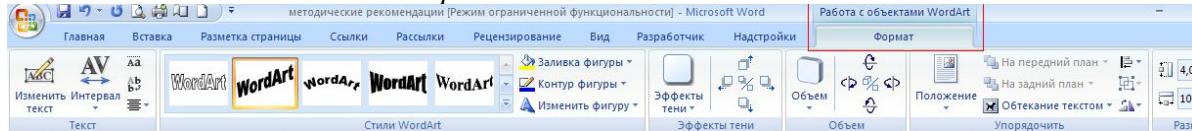
4. Чтобы настроить положение фигур относительно текста документа и друг друга, используйте инструменты, расположенные на панели *Упорядочить*.

5. Если с несколькими фигурами одновременно нужно произвести какие-либо действия (увеличить, уменьшить, переместить), либо получить один объединенный объект из множества, следует использовать *группировку*. Для *группировки* фигур, предварительно выделив их, используйте кнопку *Группировать* панели *Упорядочить*. После выполнения со сгруппированными объектами нужных действий при необходимости объекты можно разгруппировать или перегруппировать.

### ***Объекты WordArt***



*WordArt* - это красиво оформленный текст на основе готовых шаблонов, которые можно редактировать. Для вставки объекта WordArt предназначена кнопка *WordArt* на панели *Текст* вкладки *Вставка*, которая позволяет выбрать образец из коллекции и внести свой текст. После вставки объекта WordArt в окне программы появляется контекстный инструмент *Работа с объектами WordArt-Формат*.

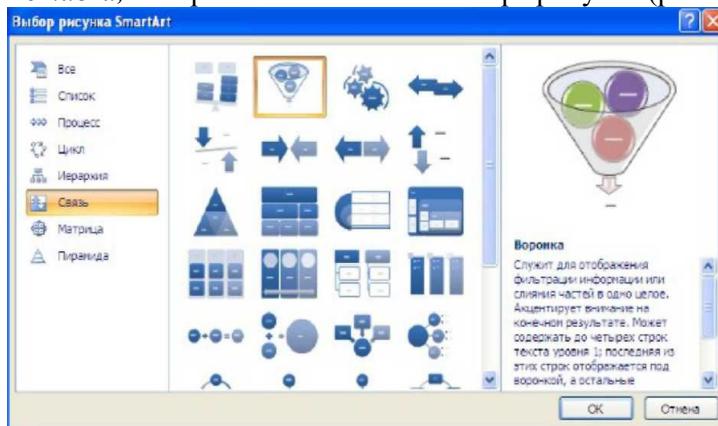


На панелях данной ленты представлены инструменты для работы с объектами WordArt, которые позволяют изменять форму, параметры и направление текста, менять стили объекта, выполнять настройку обтекания, расположения и т. д.

### **Объекты SmartArt**

Объекты SmartArt - новый тип графических элементов. Такие объекты являются чем-то средним между диаграммами и фигурами, они дают возможность представить разную информацию в виде удобных графических блоков -разнообразных красочных схем.

При выборе шаблонов SmartArt необходимо учитывать их первоначальное предназначение. Для вставки объекта SmartArt служит одноименная кнопка на панели *Иллюстрации* вкладки *Вставка*, которая вызывает окно выбора рисунка (рис. 63).



Выбрав шаблон, вы увидите его краткое описание. После добавления шаблона в документ в окне текстового процессора появится контекстный инструмент *Работа с рисунками SmartArt*, содержащий две ленты: *Конструктор* и *Формат*.

Для заполнения текстовых полей шаблона предназначены области, помеченные как *Текст*, либо левая панель SmartArt-объекта. Для добавления нового элемента в объект SmartArt надо просто нажать клавишу ввода. Пункты *Добавить фигуру выше* и *Добавить фигуру ниже* предназначены для вставки элемента другого уровня. Если какие-то кнопки неактивны, значит добавление нового элемента невозможно. Для удаления какого-либо элемента необходимо его выделить и нажать клавишу *Delete*. Кнопки *Повысить уровень* и *Понизить уровень* предназначены для изменения уровня выделенных элементов.

Объекты SmartArt форматируются как и обычный графический примитив, для этого используйте инструменты панелей ленты *Формат* контекстного инструмента *Работа с рисунками SmartArt*.

### **Диаграммы**

В документы Word можно вставлять диаграммы. При помощи диаграмм можно наглядно представить числовые данные в графическом виде, их гораздо легче воспринимать.

Для создания диаграммы необходимо выбрать вкладку *Вставка* и нажать кнопку *Диаграмма* на панели *Иллюстрации*. В появившемся окне *Вставка диаграммы* надо выбрать тип диаграммы и ее вид. После этого, автоматически открывается окно программы Excel 2007, с

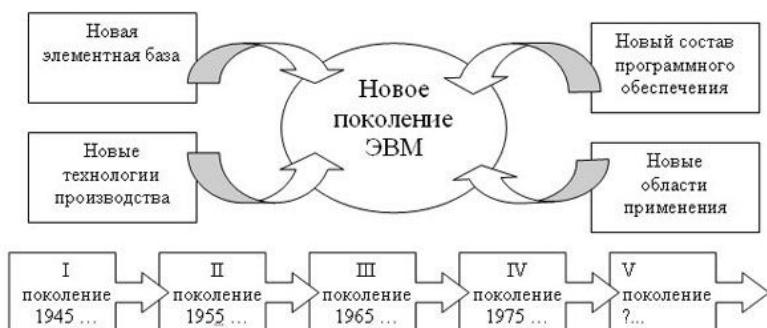
набором некоторых стандартных значений для построения диаграммы (рис. 65). Необходимо ввести свои данные для ее построения.

После ввода данных в документе Word появится построенная диаграмма. При этом в ленте главного меню появится контекстный инструмент *Работа с диаграммами*, содержащий три ленты: *Конструктор*, *Макет*, *Формат*.

- Лента *Конструктор* состоит из четырех панелей: *Тип*, *Данные*, *Макеты диаграмм*, *Стили диаграмм*. Основные операции, выполняемые этими инструментами: изменение типа и вида диаграммы, ее данных и стиля.
- Лента *Макет* содержит шесть панелей: *Текущий фрагмент*, *Вставить*, *Подписи*, *Оси*, *Фон*, *Анализ*. Эти инструменты предназначены для непосредственного оформления внешнего вида отдельных элементов диаграммы. Для выбора элемента диаграммы служит выпадающий список *Текущий фрагмент*.
- Лента *Формат* содержит инструменты для придания диаграмме окончательного вида.

Задания для самостоятельной работы

1. Постройте схему «История поколений ЭВМ» с помощью объектов SmartArt.



2. Создайте надпись «Информатика» при помощи объектов WordArt

3. Создайте диаграмму в приложении MS Word к таблице Успеваемость.

№ п/п	ФИО	Предметы				Средний балл
		История	Экономика	Психология	Информатика	
1	Иванов А.П.	3	4	5	4	
2	Петров И.В.	3	3	5	5	
3	Сидоров А.К.	4	4	5	5	
4	Смирнов В.Г.	3	3	3	5	
<b>Общий средний балл группы</b>						

## 2.9 Лабораторная работа №9 (4 часа)

**Тема:** «Создание презентаций в Power Point на базе шаблонов, использование эффектов и анимации»

**Цель работы:** научить создавать презентацию с помощью шаблона оформления в PowerPoint. Т.е. создавать слайды, добавлять в них таблицы, диаграммы, рисунки и демонстрировать презентацию.

## Перечень приборов, материалов, используемых в лабораторной работе:

1. Компьютеры с установленным программным обеспечением
2. Методические указания к выполнению лабораторной работы
3. Рабочая тетрадь.

## Описание (ход) работы:

PowerPoint — это программа для создания презентаций, причем одна из лучших. С помощью PowerPoint можно создать различные слайд-шоу: *Бизнес-презентации*, *Презентации продаж*, *Лекции*, *Слайд-шоу для Информационных станций*, *Информационные табло*, *Internet-презентации*. PowerPoint позволяет организовать презентацию даже в случае отсутствия докладчика.

Презентация PowerPoint состоит из одного или нескольких слайдов. Каждый слайд может содержать текст, графические и другие элементы. Программа предлагает ряд инструментов, которые позволяют легко форматировать и создавать привлекательные слайды.

## Добавление заметок

Заметки — это приложение к слайду. Они не появляются на самом слайде, а отображаются отдельно. С каждым слайдом презентации связана отдельная страница заметок.

В обычном режиме заметки скрыты в нижней части экрана, в отдельной панели заметок. Чтобы работать с заметками в обычном режиме, сначала необходимо расширить панель заметок. В PowerPoint предусмотрен также особый режим для работы с заметками — режим заметок (*Note Page View*) — рис. 1. Для перехода в этот режим используйте команду *Вид — Страницы заметок* (*View — Notes Page*). В верхней части каждой страницы заметок располагается уменьшенная копия слайда.

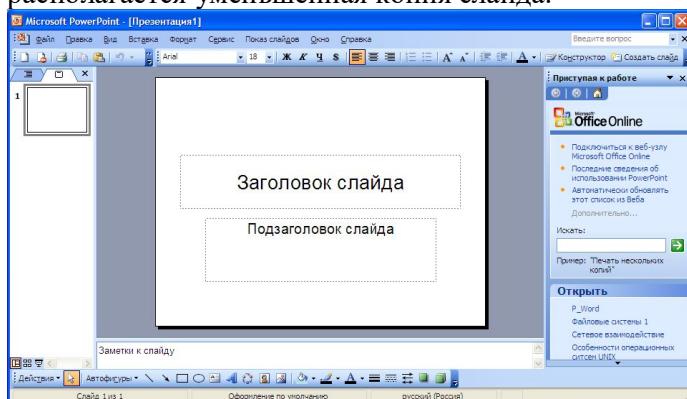


Рис. 1. Интерфейс программы MS PowerPoint

Чтобы добавить заметки к слайду, выполните следующие действия.

1. Находясь в обычном режиме, перейдите к слайду, из которого необходимо создать заметки.
2. Щелкните на границе области заметок и перетащите ее так, чтобы на экране появился текст заметок.
3. Щелкните в области с надписью *Заметки к слайду* (*Click to add notes*),
4. Введите текст.

## Форматирование текста и списков

В PowerPoint предусмотрено как незначительное, так и существенное изменение внешнего вида отдельных букв. Все параметры символов задаются с помощью диалогового окна *Шрифт* (*Font*), которое можно вызвать командой *Формат — Шрифт* (*Format — Font*).

Чтобы создать маркированный список из одного или нескольких абзацев, выполните следующие действия.

1. Выделите абзацы, которые будут маркироваться.
2. Щелкните на кнопке *Маркеры* (*Bullet*).

Если внешний вид маркеров вас не устраивает, их можно заменить другими символами, рисунками или даже клипами. Воспользуйтесь командой *Формат — Список* (*Format — List*).

*Bullet*) — откроется диалоговое окно *Список (Bullets and Numbering)*. В этом окне можно выбрать необходимый тип маркера, изменить его цвет или величину в соответствии с размером текста.

Чтобы включить в слайд нумерованный список, воспользуйтесь кнопкой *Нумерация (Numbering)* на панели *Форматирование*. Если вы щелкнете на кнопке, программа пронумерует выделенные абзацы.

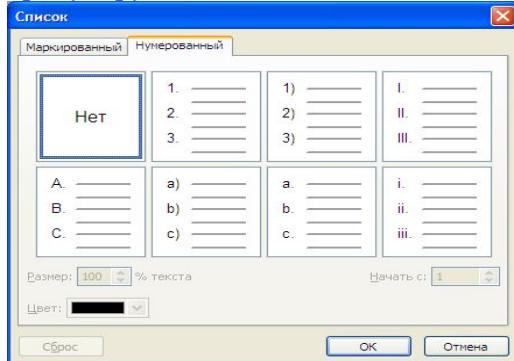


Рис. 2. Типы нумерованных списков

Для изменения формата списка выполните команду *Формат — Список (Format Bullets and Numbering)*, вызвав соответствующее диалоговое окно. Перейдите на вкладку *Нумерованный (Numbering)* (рис. 2), которая предлагает несколько видов нумерованных списков.

В программе PowerPoint предусмотрен целый ряд опций по выравниванию текста на слайде. Текст можно расположить по центру, выровнять по левому или правому краю или по ширине. Для доступа к этим командам выберите команду *Формат — Выравнивание (Format — Alignment)* и необходимый тип выравнивания. Вы также можете использовать одну из кнопок выравнивания на панели инструментов

Манипуляции с отступами производятся следующим образом:

1. Щелчком на кнопке *Обычный режим (Slide)* перейдите в обычный режим. При работе в режиме структуры (*Notes Page — View*) или сортировщика слайдов (*Slide Sorter*) операции с отступами и табуляцией не доступны.
2. Если линейка отключена, вызовите ее командой *Вид — Линейка (View — Ruler)*. Сверху и слева от окна презентации появятся линейки с маркерами текущих отступов и табуляции. Если линейки отсутствуют, вызовите их командой *Вид — Линейка*
3. Выделите текстовый объект, в котором требуется изменить отступы или табуляцию.
4. Щелчком на линейке установите табуляцию.
5. Для внесения изменений в отступы достаточно перетащить соответствующий маркер.

#### **Вставка картинок из коллекции и изображений из файлов**

Чтобы вставить в презентацию картинку из коллекции, выполните следующие действия:

1. Выделите слайд, в который планируете поместить картинку. Чтобы одно и то же изображение присутствовало на всех слайдах, командой *Вид — Образец — Образец слайдов (View — Master — Slide Master)* перейдите в режим *Образец слайдов (Slide Master View)*. Еще один вариант: щелкните на кнопке *Обычный режим (Slide View)*, удерживая нажатой клавишу *<Shift>*.
2. Выполните команду *Вставка — Рисунок — Картинки (Insert — Picture — Clip Art)*.
3. В правой части экрана появится область задач *Коллекция клипов (Insert Clip Art)*.
4. Введите ключевое слово в поле *Искать (Search Text)*, затем щелкните на кнопке *Начать (Search)*.
5. Щелкните на выбранной картинке.
6. Если добавлены все изображения, закройте область задач *Коллекция клипов*, щелкнув на кнопке с символом X, расположенной в ее правом верхнем углу.

Если изображение для вставки в презентацию уже находится на компьютере, выполните команду *Вставка – Рисунок – Из файла* (*Insert – Picture – From File*). В этом случае область задач выводиться на экран не будет, а появится диалоговое окно *Добавление рисунка* (*Insert Picture*). Итак, чтобы вставить картинку из файла, выполните следующие действия.

### ***Изменение цветовой схемы***

Если цветовая схема презентации вас не устраивает, просто поменяйте следующим образом:

1. Перейдите в обычный режим, если был активизирован другой. Щелкните на кнопке *Обычный режим* (*Normal View*) или выполните команду *Вид – Обычный* (*View – Normal*).
2. Выберите *Формат – Оформление слайда* (*Format – Slide Design*). Справа от слайда появится область задач *Дизайн слайда* (*Slide Design*).
3. Щелкните на ссылке *Цветовые схемы* (*Color Schemes*), которая находится в верхней части панели задач.
4. Выберите понравившуюся схему.

### ***Добавление к слайду даты, номера или колонтитула***

Чтобы добавить к слайду дату, номер или колонтитул, выполните следующие действия.

1. Выберите команду *Вид – Колонтитулы* (*View – Header and Footer*). На экране появится диалоговое окно *Колонтитулы* (рис. 3). (Если необходимо, перейдите на вкладку *Слайд* (*Slide*), чтобы получить доступ к параметрам колонтитулов слайда)
2. Для добавления к слайду даты установите флажок *Дату и время* (*Date and Time*). Затем выберите формат даты в раскрывающемся списке под переключателем *Автообновление* (*Update Automatically*). Чтобы на слайде отображался его номер, не забудьте установить флажок *Номер слайда* (*Slide Number*).
3. Если хотите, чтобы на каждом из слайдов присутствовал нижний колонтитул, установите флажок *Нижний колонтитул* (*Footer*), а затем наберите в поле запланированный текст.
4. Чтобы даты, номера страницы, колонтитулы были выведены на всех слайдах, кроме титульного, установите флажок *Не показывать на титульном слайде* (*Don't Show on Title Slide*).
5. Закройте диалоговое окно, щелкнув на кнопке *Применить ко всем*

### ***Создание презентаций на основе шаблона***

Чтобы создать новую презентацию на базе шаблона, выполните команду *Файл – Создать* (*File – New*), вызвав панель задач *Создание презентации* (*New Presentation*). В группе *Создание (New)* выберите опцию *Из шаблона оформления* (*From Design Template*). Появившаяся область задач *Дизайн слайда* (*Slide Design*). Щелкните на понравившемся шаблоне, чтобы применить его к презентации.

### ***Панель инструментов «Рисование»***

PowerPoint содержит довольно много инструментов для рисования, которые сгруппированы на панели *Рисование* (*Drawing*). Если панель Рисование не отображена на экране, выберите команду *Вид – Панели инструментов* (*View – Toolbars*) и установите флажок рядом с названием панели.

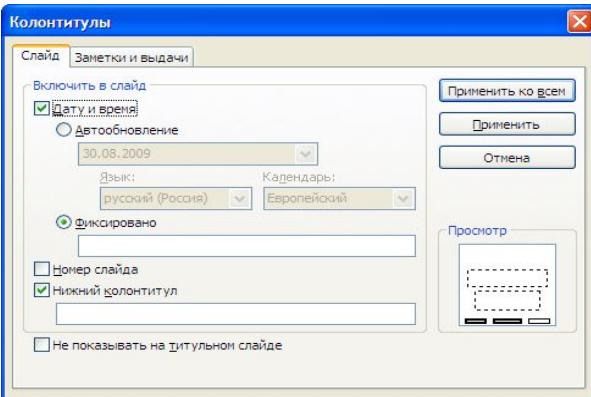


Рис. 3. Диалоговое окно Колонтитулы

Для добавления в презентацию нового слайда, содержащего диаграмму, выполните следующие действия:

1. Перейдите к слайду, за которым должен следовать новый слайд.
  2. Выберите команду *Вставка — Создать слайд* (*Insert — New Slide*) -так вы создадите новый слайд и вызовите панель *Создание слайда* (*Slide Layout*).
  3. Щелкните на одном из вариантов макета слайда, содержащего диаграмму. Выберите необходимый и щелкните на кнопке OK. Power Point добавит новый слайд выбранного типа. *Объект Диаграмма* (*Chart*) в этом случае станет местом размещения диаграммы. Для того чтобы придать ему значение, используйте Microsoft Graph. Некоторые макеты разметки слайда содержат элемент *Заголовок* (*Content*). При выборе такого типа разметки на слайде будет создано место для заголовка, в котором расположено шесть пиктограмм. Эти пиктограммы позволяют создавать таблицы, диаграммы, рисунки, рисунки из файла или видеоролики в зависимости от выбранной пиктограммы.
  4. Дважды щелкните на объекте *Диаграмма*, чтобы вызвать Microsoft Graph. Если выбран слайд с содержимым, просто щелкните на пиктограмме Диаграмма. В любом случае Power Point запустит Microsoft Graph, которая создаст пример диаграммы на основе предполагаемых данных (рис. 4). Обратите внимание: Microsoft Graph заменяет панели инструментов Power Point своими собственными.
  5. Измените данные примера на собственные.
  6. Вернитесь на слайд. Щелкните в любой части слайда за пределами диаграммы или таблицы, чтобы закрыть Microsoft Graph и вернуться на слайд.
- Добавление диаграммы на существующий слайд:
1. Перейдите на слайд, на который требуется поместить диаграмму.
  2. Выберите команду *Вставка — Диаграмма* (*Insert — Chart*). Можно также щелкнуть на кнопке *Добавление диаграммы* (*Insert Chart*) на панели инструментов *Стандартная* (*Standard*).
  3. Введите данные в таблицу.
  4. Щелкните за пределами диаграммы, чтобы вернуться к слайду.
  5. Разместите все элементы в необходимом порядке.

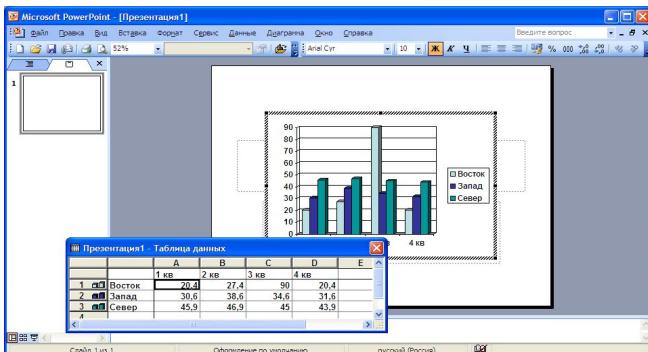


Рис. 4. Окно Microsoft Graph

Power Point содержит такое свойство, как Библиотека диаграмм (Diagram Gallery). Оно позволяет добавлять на слайд различные типы полезных диаграмм. С помощью библиотеки диаграмм вы можете создать организационную, циклическую, радиальную, пирамидальную, целевую диаграмму и диаграмму Венна.

Из шести типов диаграмм, которые создаются с помощью библиотеки, все (за исключением организационной диаграммы) реализуют единый принцип, отображая отношения между элементами диаграмм. После создания диаграммы вы имеете возможность изменить ее тип. Простейший способ создания диаграмм — вставка нового слайда, макет которого содержит место для диаграммы. Выполните следующие действия:

1. Выполните команду Вставка — Создать слайд (Insert — New Slide) или нажмите <Ctrl+M> для вставки нового слайда. Будет создан новый слайд, а затем откроется область задач, которая содержит макеты. Выберите макет, на котором предусмотрено место для диаграммы.
2. Выберите макет Диаграмма (Diagram) или Организационная диаграмма (Organization Diagram).
3. Дважды щелкните на элементе слайда, в котором должна размещаться диаграмма. Откроется диалоговое окно Библиотека диаграмм (Diagram Gallery), приведенное на рис. 5.
4. Определите тип диаграммы, который вы решили создать.
5. Щелкните на кнопке OK.

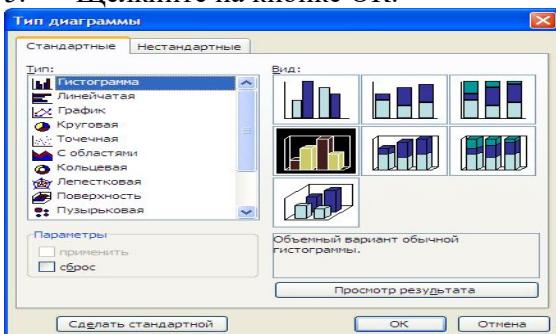


Рис. 5. Диалоговое окно Библиотеке диаграмм

6. Измените диаграмму. Подробнее об изменении диаграмм см. раздел "Работа с организационными диаграммами" и "Работа с диаграммами других типов" далее в этой главе.
7. Готово.

Диаграмму необходимо изменять и настраивать до тех пор, пока она не будет соответствовать всем требованиям.

### Использование таблиц

Таблицы — эффективное средство для представления большого количества информации. Вначале создайте слайд, содержащий таблицу. Для этого используйте макет слайда Заголовок и таблица (Title and Table).

1. Выберите команду Вставка — Создать слайд или нажмите <Ctrl+M>. Будет создан новый слайд и откроется область задач Создание слайда.
2. В области задач Создание слайда выберите макет Заголовок и таблица. Чтобы найти макет Заголовок и таблица, прокрутите список макетов почти до конца (макет находится в разделе Другие макеты (Other Laptop)).
3. Дважды щелкните на области таблицы в новом слайде. Откроется диалоговое окно Вставка таблицы (Insert Table).
4. Задайте число строк и столбцов новой таблицы и щелкните на кнопке ОК. Откроется таблица. Рядом с ней появится плавающая панель инструментов Таблицы и границы (Tables and Borders), на которой находятся кнопки, используемые для создания макета таблицы и ее форматирования.

### **Задание к лабораторной работе**

Создать презентацию предприятия

#### **Порядок выполнения работы**

- 1) Разработать содержание презентации согласно «Заданию к лабораторной работе»;
- 2) Оформить постановочный слайд;
- 3) Создать не менее 10 слайдов со следующим содержимым (содержимое должно соответствовать тематике презентации):
  - различными цветами фона;
  - примерами шаблонов оформления;
  - текстовыми вставками, с применением различного вида форматирования текста;
  - списками трех видов;
  - таблицами;
  - диаграммами;
  - организационными диаграммами любого вида;
  - элементами «Панели рисования»: WordArt, автофигурами, рисунками из коллекции и из файлов;
  - кнопками навигации по презентации;
- 4) Оформить заключительный слайд;
- 5) Добавить минимум к трем слайдам заметки;
- 6) Добавить к презентации колонтитулы.

#### **Список предприятий для выбора тем индивидуальных заданий**

№ п/п	Предприятие, учреждение
<b>Предприятия производственной сферы</b>	
1	Автомастерская
2	Автосалон
3	Ателье пошива и ремонта одежды
4	Мастерская по ремонту обуви
5	Мастерская по ремонту сложной техники

6	Фирма по сборке и продаже компьютеров
7	АЗС (автозаправочная станция)
8	Строительная компания
9	Книжное издательство

## 2.10 Лабораторная работа №10 (2 часа)

**Тема:** «Создание и форматирование HTML-документов»

**Цель работы:** Научиться создавать простейший гипертекстовый документ средствами текстового редактора Блокнот. Научиться использовать теги форматирования шрифта и абзаца.

**Перечень приборов, материалов, используемых в лабораторной работе:**

1. Компьютеры с установленным программным обеспечением
2. Методические указания к выполнению лабораторной работы
3. Рабочая тетрадь.

**Описание (ход) работы:**

1. Создать файл с гипертекстовым документом:

- Запустить редактор Блокнот, ввести текст:

**Приветствую Вас на моей первой web-страничке!**

- Сохранить файл в созданной папке. При сохранении, в окне диалога **Сохранить как...** в строке **Тип файла:** выбрать вариант **Все файлы (\*.\*)**, а в строке **Имя файла** задать имя с расширением **.htm**, например **1\_name.htm** (где **name** – ваше имя)

- Закрыть документ, найти его пиктограмму в окне **Мой компьютер** или в окне программы **Проводник**.

- Открыть файл. Проанализировать, *с помощью какого приложения отображается файл* и как выглядит введенная фраза.

2. Ввести теги, определяющие структуру html-документа:

- С помощью контекстного меню открыть файл с помощью редактора Блокнот. Ввести приведенные ниже теги, в разделе заголовка документа (между тегами **<TITLE> </TITLE>**) указать свою фамилию.

**<HTML>**

**<HEAD> <TITLE> Фамилия </TITLE>**

**</HEAD>**

**<BODY>**

**Приветствую Вас на моей первой web-страничке!**

**</BODY>**

**</HTML>**

- **Сохранить** документ под тем же именем, обновить его отображение в броузере (выполнить **Вид/Обновить** или нажать кнопку **Обновить** на панели инструментов). Проанализировать произошедшие изменения в отображении документа.

3. Отредактировать документ:

- Вызвать меню броузера **Вид/Просмотр HTML-кода** и добавить после текста **«Приветствую Вас на моей первой web-страничке!»** текст подписи:

## Студент группы NNN **Фамилия Имя**

Сохранить документ (но не закрывать) и обновить его просмотр в броузере.

- Используя одиночный тег **<BR>**, отредактировать документ так, чтобы подпись начиналась с новой строки, а **Фамилия Имя** – в следующей строке. Просмотреть в броузере новый вариант.

**Внимание!** После каждого изменения документ нужно сохранять, а просмотр в броузере начинать с обновления загрузки документа с помощью кнопки «Обновить» на панели инструментов.

4. Оформить фрагменты текста с помощью стилей **Заголовков**:

- Первую строку документа оформить стилем **Заголовок 1-го уровня** с помощью парного тега **<H1> ...</H1>**. Вторую строку оформить как **Заголовок 6-го уровня**, а третью как **Заголовок 4-го уровня**.

- Просмотреть документ в броузере, изменяя настройку отображения шрифтов (меню **Вид / Размер шрифта / Самый крупный, Средний, Мелкий и Самый мелкий**).

- Поменять стиль оформления первой строки на **Заголовок 2 уровня**, второй строки - на **Заголовок 5 уровня**, последней строки - на **Заголовок 3-го уровня**.

5. Выполнить форматирование шрифта:

- После строки **Фамилия Имя** добавить еще одну строку текста

**Нас утро встречает прохладой**

- Оформить фразу по приведенному ниже образцу.

**Нас *утро* встречает прохладой**

В слове УТРО все буквы должны иметь **разные цвета**. В слове ПРОХЛАДОЙ оформить буквы ПРО – **красным** цветом, ОЙ – **синим**.

- Оформить строку с подписью **(Студент группы NNN Фамилия Имя)** **курсивом**, размер шрифта задать относительным изменением. Использовать теги **<FONT SIZE=<+2>>** и **<I>**

- Просмотреть полученный документ в броузере.

6. Выполнить форматирование абзацев:

- Создать новый документ **2\_name.htm**, сохранить его в той же рабочей папке.
- Ввести текст (использовать копирование текста из документа **1\_name.htm**):

```
<HTML>
<HEAD> <TITLE> Фамилия </TITLE>
      </HEAD>
      <BODY>
```

Приветствую Вас на моей второй web-страничке! <BR> Монолог Гамлета

```
      </BODY>
</HTML>
```

- Выровнять текст **по центру**.

- Ввести текст:

Быть иль не быть - вот в чем вопрос. Что благороднее: сносить удары неистовой судьбы - иль против моря невзгод вооружиться, в бой вступить. И все покончить разом...

- Оформить выравнивание абзаца **по ширине**.

- Ограничить абзац горизонтальными разделительными линиями сверху и снизу, используя тег **<HR>**.

- Скопировать монолог и разбить его на абзацы. Выровнять **по центру**.

Быть	иль	не	быть	-	вот	в	чем	вопрос.
Что		благороднее:			сносить			удары
Неистовой		судьбы	-		иль	против		моря
Невзгод		вооружиться,		в		бой		вступить

И все покончить разом...

- Сохранить документ.
  - Просмотреть документ в окне броузера, изменяя размер окна.
7. Выполнить оформление списков:
- Создать новый документ **3\_name.htm**, сохранить его в той же рабочей папке жесткого диска.
  - Ввести текст:

```
<HTML>
<HEAD> <TITLE> Фамилия </TITLE>
      </HEAD>
      <BODY>
Приветствую Вас на моей третьей web-страничке!
</BODY>
</HTML>
```

- Дополнить текст документа (между тегами **<BODY>...</BODY>**) следующим текстом:  
**Я** **знаю** **как** **оформлять:**  
**Шрифты,**  
**Заголовки,**  
**Абзацы**
- Оформить три последние строки как **список нумерованный**. Для этого использовать следующую конструкцию тегов:

```
<OL>
<LI> Шрифты, </LI>
<LI> Заголовки, </LI>
<LI> Абзацы </LI>
</OL>
```

- Поменять оформление списка на **список маркированный**. Использовать теги **<UL>, </UL>**
- Создать «смешанный» список:

**Я знаю как оформлять:**

- Шрифты**
- Размер**
- Цвет**
- Гарнитуру**
- Индексы**
- Заголовки**
- От 1-го до 6-го уровня**
- Абзацы**
- Выравнивание**
- Разрыв строк внутри абзаца**
- С использованием переформатирования.**

8. Предъявить результаты работы преподавателю.

*Таблица основных тегов HTML-документа. Теги форматирования шрифта и абзаца*

Назначение	Вид тегов	Примечание
<b>Общая структура документа HTML</b>		
Тип документа	<HTML></HTML>	Начало и конец документа

Имя документа	<HEAD></HEAD>		Не отображается броузером
Заголовок	<TITLE></TITLE>		Содержимое строки заголовка окна броузера
Тело документа	<BODY></BODY>		Содержимое WEB-страницы
<b>Структура содержания документа</b>			
Внутренние заголовки различного уровня	<H <sub>№</sub> > текст </H <sub>№</sub> >		Где № – номер уровня заголовка (от 1 до 6). Например, <H1>...</H1> - заголовок 1-го уровня.
Заголовок с выравниванием	<H <sub>№</sub> ALIGN="LEFT  CENTER  RIGHT"> текст </H <sub>№</sub> >		LEFT - по левому краю, CENTER - по центру, RIGHT - по правому краю.
<b>Форматирование абзацев</b>			
Создание абзаца (параграфа)	<P> текст </P>		Абзацы отделяются двойным межстрочным интервалом
Перевод строки внутри абзаца	 		Одиночный тег
Выравнивание абзаца	<P ALIGN="LEFT">текст </P> <P ALIGN= "CENTER">текст </P> <P ALIGN="RIGHT"> текст</P> <P ALIGN="JUSTIFY"> текст </P>		LEFT - по левому краю CENTER - по центру  RIGHT - по правому краю JUSTIFY – по ширине
Разделительная горизонтальная линия между абзацами	<HR SIZE=<?>>		Одиночный тег. «?» - толщина линии в пикселях. Толщину линии можно не указывать.
<b>Форматирование шрифта</b>			
Жирный	<B> текст </B>		<B>Жирный</B>
Курсив	<I> текст </I>		<I> Курсив </I>
Подчеркнутый	<U> текст </U>		<U> Подчеркнутый </U>
Перечеркнутый	<S> текст </S>		<S> Перечеркнутый</S>
Увеличенный размер	<BIG> текст </BIG >		
Уменьшенный размер	<SMALL> текст </SMALL>		
Верхний индекс	<SUP> текст </SUP>		<SUP> Верхний индекс </SUP>
Нижний индекс	<SUB> текст </SUB>		<SUB> Нижний индекс </SUB>
Размер шрифта	<FONT SIZE=<?>> текст </FONT>		? - значения от 1 до 7 или относительное изменение (например, +2)
Базовый размер шрифта	<BASEFONT SIZE=<?>>		Одиночный тег ? – размер от 1 до 7; по умолчанию равен 3 и задается для всего документа в целом
Гарнитура шрифта	<FONT FASE=<название1, название2, ...>> текст </FONT>		Текст оформляется первым, установленным на компьютере шрифтом из списка названий
Цвет шрифта	<FONT COLOR="\$\$\$\$\$\$"> текст </FONT>		Цвет задается либо ключевым словом, либо шестнадцатеричным кодом с символом #

		RED –красный, #FF0000 – шестнадцатеричный код – красного цвета
<i>Создание списков</i>		
Нумерованный	<OL> элементы списка </OL>	<OL>
Маркированный	<UL> элементы списка </UL>	<LI> Элемент списка 1 </LI>
Элемент списка	<LI> элементы списка </LI>	<LI> Элемент списка 2 </LI> <LI> Элемент списка 3 </LI> </OL>

## 2.11 Лабораторная работа №11 (2 часа)

**Тема:** «Добавление графических объектов и анимации в HTML-документы»

**Цель работы:** Научиться выполнять вставку рисунков в HTML-документ. Научиться создавать закладки и гиперссылки

**Перечень приборов, материалов, используемых в лабораторной работе:**

1. Компьютеры с установленным программным обеспечением
2. Методические указания к выполнению лабораторной работы
3. Рабочая тетрадь.

**Описание (ход) работы:**

1. Скопировать из Интернета или какой-либо папки в личную папку файлы три графических файла (например, **Arrows1.wmf**, **Arrows2.wmf**, **Arrows3.wmf**).

Убедиться, что созданные ранее документы **1\_name.htm**, **2\_name.htm** и **3\_name.htm** также находятся в вашей папке на жёстком диске.

2. Вставка рисунков в документ.
  - Открыть в Блокноте документ **2\_name.htm**.
  - Вставить рисунок **Arrows1.wmf** в начало документа **2\_name.htm**. Для вставки использовать тег **IMG** с параметрами **WIDTH** и **HEIGHT** для установки размеров рисунка 50 пикселов по горизонтали и по вертикали.
  - Сохранить документ под именем **4\_name.htm**.
  - Просмотреть в браузере полученный результат.
  - Ввести в тег рисунка параметр **ALIGN** для выравнивания рисунка по правому краю. Просмотреть результат в браузере.
  - Вставить рисунок **Arrows2.wmf** в конец документа **4\_name.htm** перед, подобрать тип выравнивания рисунка на свое усмотрение. Установить размер рисунка 100 пикселов по горизонтали и по вертикали. С помощью параметра **ALT** создать всплывающую подсказку «**Рисунок 2**», появляющуюся при наведении курсора мыши на рисунок.
  - Просмотреть в браузере полученный результат.

3. Создание гиперссылок и закладок.

- В документе **3\_name.htm** закрепить гиперссылки за следующими элементами списка:

За словом **Шрифт** – гиперссылка на документ **1\_name.htm**.

За словом **Заголовки** – на документ **1\_name.htm**.

За словом **Абзацы** – на документ **2\_name.htm**.

- Создать закладку в документе **1\_name.htm** перед фразой «**Нас утро встречает прохладой**». Дать ей имя «**Morning**».

- Изменить первую гиперссылку (слово **Шрифт**) так, чтобы она указывала на закладку «**Morning**» в документе **1\_name.htm**.

- Создать закладку в начале текущего документа **3\_name.htm**. Присвоить ей имя «Hello».
  - Изменить вторую гиперссылку (на слове **Заголовки**), определив для неё переход в начало текущего документа на установленную закладку «Hello».
  - Создать закладку в документе **2\_name.htm** перед фрагментом монолога. Присвоить ей имя «Mono».
    - Установить на слово **реформирования** гиперссылку на закладку «Mono».
    - Проверить правильность переходов по всем гиперссылкам.
4. Закрепить гиперссылки за графическими файлами:
- Отредактировать тег вставки рисунка **Arrows1.wmf**, ввести в тег атрибут **ALT** для отображения текста подсказки «**Вернуться**». Просмотреть в браузере как реагирует рисунок на наведение курсора мыши.
  - Закрепить за рисунком **Arrows1.wmf** в документе **4\_name.htm** гиперссылку на документ **3\_name.htm**. Выполнить переход между документами.
5. Предъявить результат преподавателю.

#### Основные теги вставки рисунков, закладок и гиперссылок

<i>Вставка изображений</i>		
Вставка графического файла	<IMG SRC="файл" WIDTH="ширина" HEIGHT="высота" >	<i>Пример:</i> <IMG SRC="grafica.gif" WIDTH="550" HEIGHT="240" ALIGN="right" ALT="Графический файл">
Выравнивание картинки относительно текста	<IMG SRC="файл" ALIGN="left right top bottom">	
Вывод текста всплывающей подсказки при наведении курсора мыши на рисунок	<IMG SRC="файл" ALT="текст">	
<i>Вставка ссылок</i>		
Ссылки на другую страницу	<A HREF="страница"> текст </A>	<A HREF="str2.htm"> Ссылка1</A>
Ссылка на закладку в другом документе	<A HREF="#" страница # имя закладки"> текст</A>	<A HREF="#" index.htm #met1"> На главную страницу</A>"
Ссылка на закладку в том же документе	<A HREF="#имя закладки"> текст </A>	<A HREF="#" #metka1"> Ссылка2</A>
Определить закладку	<A NAME="имя закладки">текст</A>	<A NAME="metka1"></A>

Цвет фона, текста и ссылок	
Фоновая картинка	<BODY BACKGROUND="файл рисунка">
Цвет фона	<BODY bgcolor="#\$\$\$\$\$\$" >

## 2.12 Лабораторная работа №12 (2 часа)

Тема: «Создание и форматирование таблиц»

Цель работы: Научиться использовать таблицы для оформления WEB-страниц

Перечень приборов, материалов, используемых в лабораторной работе:

1. Компьютеры с установленным программным обеспечением
2. Методические указания к выполнению лабораторной работы
3. Рабочая тетрадь.

Описание (ход) работы:

1. Создать таблицу по приведенному образцу, сохранить документ под именем **tabl\_name.htm**. Сверху над таблицей разместить заголовок **Таблица №1**

При отображении таблицы в броузере должны удовлетворяться следующие условия:

- таблица должна быть выравнена по центру и быть правильной (симметричной) формы;
    - в центральной ячейке поместить символ \* (звездочка), остальные ячейки должны быть пустыми.
- Примечание.** Для отображения пустых ячеек в них нужно поместить символьный примитив пробела &nbsp;
2. В этом же документе создать копию таблицы №1, ввести заголовок **Таблица №2** и модифицировать ее:
    - В центральной ячейке разместить рисунок **Arrows3.wmf**
    - «Раскрасить» все остальные ячейки в различные цвета.

3. Создать еще одну копию таблицы – **Таблица №3** и отредактировать теги таблицы так, чтобы она соответствовала приведенному ниже образцу.



**Примечание.** Для объединения ячеек в тегах <TD> необходимо использовать параметры **colspan=** и **rowspan=**

4. Создать новый HTML-документ - **rasp\_name.htm** с расписанием занятий.

- Документ должен начинаться заголовком

**Расписание занятий гр. NNN на весенний семестр 2005 г.**

- Первая строка таблицы должна быть оформлена как заголовки полей (с использованием тегов <TH>).
- Таблица по ширине должна занимать полный размер окна. Ширину отдельных столбцов задать в относительных единицах (в %), с тем, чтобы при изменении ширины окна пропорции таблицы сохранялись.

День недели	Время	Предмет	Преподаватель	Аудитория
Понедельник	8:30-10:05	Математика (лек)	доц. Иванов А.А.	320

	10:15-11:50	Математика (пр)	преп. Петрова И.А.	302
	12:30-14:05	Физика (лаб)	доц. Сидоров О.И.	307
Вторник	8:30-10:05	История (лек)	проф. Громова О.А.	310
	10:15-11:50	История (сем)	преп. Попов М.А.	302
	12:30-14:05	Физика (лаб)	доц. Сидоров О.И.	307
...	...	...	...	...

- Просмотреть созданный документ в броузере при различных размерах окна и различных настройках размера шрифта.

5. Сохранить файл с расписанием под именем **rasp\_menu\_name.htm** и модифицировать его.

6. После заголовка создать таблицу, состоящую из одной строки меню с названиями дней недели.

## Расписание

<a href="#">Понед.</a>	<a href="#">Вторник</a>	<a href="#">Среда</a>	<a href="#">Четверг</a>	<a href="#">Пятн.</a>	<a href="#">Субб.</a>
------------------------	-------------------------	-----------------------	-------------------------	-----------------------	-----------------------

7. В таблице с расписанием установить закладки на названия дней недели.

8. В таблице меню создать гиперссылки на соответствующие дни недели.

9. Выполнить цветовое оформление каждой ячейки меню.

10. Проверить правильность выполнения переходов по гиперссылкам.

11. Создать группу web-страниц, объединенных меню:

- На рабочем диске создать папку **My\_raspisanie** для размещения файлов расписания.
- Поместить расписание на каждый день недели и таблицу с меню в отдельные файлы. Имена файлов: **menu.htm** – для главной страницы, названия дней недели – для остальных. Все документы разместить в папке **My\_raspisanie**.

- Отредактировать гиперссылки меню так, чтобы по ним выполнялись переходы на соответствующий документ.
- В конце каждого файла с расписанием на день организовать гиперссылку для возврата в главный документ с меню.
- Оформить фон каждого дня недели собственным цветом, совпадающим с цветом ячейки таблицы меню.

12. Предъявить результат преподавателю.

## Теги оформления таблиц

Определить таблицу	<TABLE></TABLE>	Пример
Строка таблицы	<TR> </TR>	<TABLE border="1" align="center" width="50%>
Выравнивание	<TR ALIGN=left   right   center   middle   bottom>	<TR>
Ячейка таблицы	<TD></TD>	<TH>Товар</TH>
Выравнивание по горизонтали	<TD ALIGN=LEFT   RIGHT   CENTER>	<TH>Цена</TH>
Выравнивание по вертикали	<TD VALIGN = TOP MIDDLE BOTTOM>	</TR>
		<TR>
		<TD>Радиотелефон</TD>

Установка ширины ячейки (в пикселях или %)	<TD WIDTH=<?>>	<TD>2000 </TD> </TR> </TABLE> <table border="1"> <tr> <td>Товар</td><td>Цена</td></tr> <tr> <td>Радиотелефон</td><td>2000</td></tr> </table>	Товар	Цена	Радиотелефон	2000
Товар	Цена					
Радиотелефон	2000					
Заливка цветом ячейки	<TD BGCOLOR = «# цвет»> </TD>	<TD BGCOLOR = «#FF0000»> </TD> красный цвет				
Заголовок столбца или строки	<TH>текст </TH>	Текст в ячейке выравнивается по центру, устанавливается жирный шрифт				

## 2.13 Лабораторная работа №13 (6 часов)

**Тема:** «Создание и заполнение базы данных в Microsoft Access. Связывание таблиц, создание форм, запросов, отчетов»

**Перечень приборов, материалов, используемых в лабораторной работе:**

1. Компьютеры с установленным программным обеспечением
2. Методические указания к выполнению лабораторной работы
3. Рабочая тетрадь.

**Описание (ход) работы:**

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).

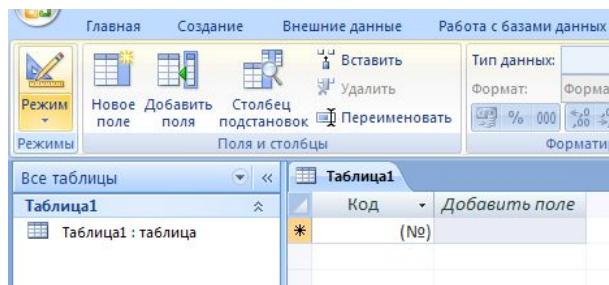


Рисунок 1

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.



Таблица 1.

Ведомость успеваемости	
Имя поля	Тип данных
Код	Счетчик
Фамилия	Текстовый
Имя	Текстовый
Математика	Числовой
Менеджмент	Числовой
Сервисная деятельность	Числовой
Информационные технологии	Числовой
Стандартизация	Числовой
Гостиничная индустрия	Числовой
Пропуски по неуважительной причине	Числовой
Пропуски по уважительной причине	Числовой

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	Математика	Менеджмент	Сервисная деятельность	Информационные технологии	Стандартизация	Гостиничная индустрия	Пропуски по неуважительной причине	Пропуски по уважительной причине
1	Иванникова	Анна								
2	Баранова	Ирина								
3	Корнилова	Ольга								
4	Воробьев	Алексей								
5	Воробьев	Олег								
6	Скоркин	Александр								
7	Володина	Нина								
8	Новоселов	Алексей								
9	Петрова	Елена								
10	Чернова	Кристина								
11	Терещинка	Инна								
12	Истратов	Максим								
13	Бондарь	Ольга								

14	Ревин	Олег							
15	Шарова	Оксана							

8. Выполните редактирование ячеек:

– Замените фамилию Иванникова на Иванова.

9. Отсортируйте:

а) *фамилии* – по алфавиту (поставьте маркер на любую фамилию в столбце Фамилия и щелкните 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 инструментов 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.

1) Откройте учебную базу данных **<Анкета ГС-31>**.

2) Создайте таблицу **<Преподаватели >** в *Режиме таблицы*. Для этого в меню Создание выберите кнопку **Таблица**. В появившейся таблице сделайте следующее:

- Добавьте два поля – Поле 1 и Поле 2, выполнив команду через контекстное меню.
- Переименуйте **<Поле 1>** на **<Предмет>**. Для этого поставьте курсор в любую ячейку столбца **<Поля 1>** и выполните команду **Переименовать столбец** из контекстного меню. Или щелкните два раза по имени поля, удалите старое название и впечатайте новое.
- Переименуйте аналогично **<Поле 2>** на **<Преподаватель>**.

3) Сохраните таблицу с именем **<Преподаватели>**, щелкнув по кнопке **<Сохранить>** (дискетка  на панели инструментов).

4) Перейдите в режим **<Конструктор>** и удалите строку с ключевым словом **Счетчик**. Посмотрите как заданы поля. Сделайте поле **<Предмет>** ключевым, поместив курсор на имя этого поля и щелкнув по кнопке  **ключевое поле**. Тип данных поля задайте **текстовым**.

5) Перейдите в *Режим таблицы* и заполните таблицу **<Преподаватели>** записями из *Таблицы3*.

**Таблица 3**| преподаватели | |
| --- | --- |
| предмет | преподаватель |
| Математика | Бекетова Н.И. |
| Менеджмент | Казумова Н.С. |
| Сервисная деятельность | Бессарабова Т.В. |
| Информационные технологии | Бабич О.А. |
| Стандартизация | Казарян Г.Г. |
| Гостиничная индустрия | Казарян Г.Г. |

Таблица 3	
предмет	имя
Математика	Бекетова Н.И.
Менеджмент	Казумова Н.С.
Сервисная деятельность	Бессарабова Т.В.
Информационные технологии	Бабич О.А.
Стандартизация	Казарян Г.Г.
Гостиничная индустрия	Казарян Г.Г.

- 6) Закройте таблицу <Преподаватели>, сохранив все изменения.
- 7) Используя <Шаблон таблицы>, создайте таблицу <Личные данные> студентов с ключевым полем. Для этого:
- Находясь на закладке <Создание> щелкните по кнопке <Шаблоны таблиц>, <Контакты>. Появится таблица уже с готовыми полями.
  - Переименуйте предложенные поля на следующие поля: <Код студента>, <Фамилия>, <Имя>, <Город>, <Адрес>, <Телефон>, <Дата рождения>, <Фотография>, <Любимый предмет>, лишние поля удалите.
  - Сохраните полученную таблицу под названием <Личные данные>. Ключевое поле задано автоматически.
- 8) Внесите данные в новую таблицу, заполнив поля <Фамилия>, <Имя>, <Город>, <Адрес>, <Телефон>, <Дата рождения>.

**ПРИМЕЧАНИЕ.** Поля <Фамилия> и <Имя> можно скопировать из таблицы <Ведомость успеваемости>. В поле <Город> внесите четыре разных города (например, Новороссийск, Геленджик, Анапа, Крымск)

9) Перейдите в режим <Конструктор> и назначьте типы данных: для поля <Телефон> - *числовой*, для поля <Дата рождения> - *дата/время*, для поля <Фотография> – *поле объекта OLE*, для остальных – *текстовый*.

Для поля <Любимый предмет> выполните свойство выбор предмета из списка с помощью *Мастера подстановок*. Для этого в строке <Любимый предмет> в поле *Тип данных – текстовый* щелкните по кнопке и в ниспадающем меню выберите команду <Мастер подстановок>.

- В диалоговом окне <Создание подстановки> поставьте флажок напротив способа <Будет введен фиксированный набор значений> и нажмите <Далее>.
- В следующем окне внесите в столбец все предметы (предметы из таблицы <Преподаватели>), нажмите <Далее>.
- В последнем окне, не изменяя имени столбца нажмите <Готово>.

10) Перейдите в режим таблицы и выберите для каждого студента с помощью кнопки списка любимый предмет.

11) Создайте *схему данных*, т.е. установите связи между таблицами.

Щелкните по кнопке - *Схема данных* на панели инструментов меню <Работа с базами данных>. В окне <Отобразить таблицу> выделите таблицу <Ведомость успеваемости> и щелкните по кнопке <Добавить>. Также добавьте таблицы <Преподаватели> и <Личные данные>. В окне <Схема данных> появится условный вид этих таблиц. Закройте окно <Добавление таблицы>.

- Поставьте мышку на имя поля <Предметы> в таблице <Преподаватели>, и не отпуская кнопку мыши перетащите его на поле <Любимый предмет> таблицы <Личные данные>. Отпустите мышку. Появится диалоговое окно <Связи>, в котором включите значки <Обеспечение целостности данных>, <Каскадное обновление связанных полей> и <Каскадное

*удаление связанных полей*. Щелкните по кнопке <Создать>. Появиться связь «**один-ко-многим**».

- Поставьте мышку на имя поля <Код студента> в таблице <Личные данные> и перетащите его, не отпуская мышки, на поле <Код> таблицы <Ведомость успеваемости>. В появившемся окне <Связи> включите значок «*Обеспечение целостности данных*» и щелкните по кнопке <Создать>. Появиться связь «**один-к-одному**».

- Закройте схему данных, сохранив ее.

12) Произведите фильтрацию данных в таблице <Личные данные> *по выделенному*.

- Откройте таблицу в режиме таблицы.
- Выберите студентов, проживающих в Новороссийске. Для этого поставьте курсор в одну из первых записей, где есть город Новороссийск и щелкните по кнопке  «*Фильтр по выделенному*» на панели инструментов. Выберите команду <Равно «Новороссийск»>. Access отобразит все записи, удовлетворяющие критерию фильтрации.
- Для отображения всех записей выполните команду <Удалить фильтр> для этого щелкните по соответствующей кнопке на панели инструментов .

13) Закончите работу с базой данных Access.

1) Откройте свою учебную базу данных.

2) Создайте запрос на выборку студентов, у которых по всем предметам только хорошие оценки с помощью *Мастера запросов*.

- На панели инструментов выберите команду <Мастер запросов>.
- В появившемся диалоговом окне выберите <Простой запрос> и щелкните по кнопке <OK>.
- В следующем окне выберите таблицу, по которой строится запрос (<Ведомость успеваемости>), и те поля, которые  входят в запросе. Перенесите их в правую часть окна с помощью кнопки , нажмите <Далее>. В следующем окне тоже нажмите <Далее>.
- В другом окне дайте название запроса «**Хорошисты**» и нажмите <Готово>.
- Появится таблица <Хорошисты>, в которой отражены фамилии всех студентов и изучаемые предметы.
- Откройте таблицу «**Хорошисты**», перейдите в режим <Конструктор>. Здесь в поле <Условия отбора> под каждым предметом поставьте условие  $>=4$  или **4OR5**.

**Примечание:** Галочки в каждом поле означают, что по вашему выбору можно включить или убрать любое поле на выборку.

- Перейдите в режим таблицы, ответив <Да> на вопрос о сохранении запроса. (В таблице должны остаться фамилии «хорошистов»).

3) С помощью <Конструктора запросов> создайте запрос на выборку по таблице <Личные данные>.

- Щелкните по таблице <Личные данные>, зайдите в меню <Создание>, выберите команду <Конструктор запросов>.
- Добавьте нужную таблицу в поле запроса. Выделите её в списке и щелкните по кнопке <Добавить>. Закройте окно <Добавление таблицы>.
- Выберите студентов, чьи фамилии начинаются на букву «В» и которые проживают в Анапе. Для этого:
  - добавьте в строку <Поле> два поля <Фамилия> и <Город>;
  - в строке <Условия отбора> в первом столбце укажите значение Like «**B \***», а во втором столбце с названием <Город> - «**Анапа**»;

- закройте запрос, сохранив его под названием “**BBB**” (у вас должны остаться в списке студенты, проживающие в Анапе). Рисунок 2.

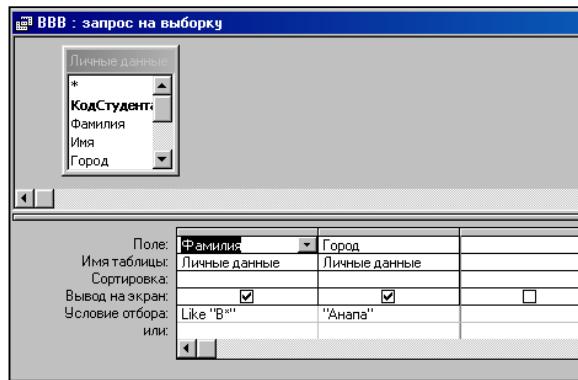


Рисунок 2.

### **Самостоятельное задание**

- Составьте запрос с названием <Запрос 1> на базе таблицы <Ведомость успеваемости>, в котором будут указаны студенты, имеющие по первым двум предметам оценки не менее «4». (Выполните запрос или через Конструктор запросов, или через Мастер запросов)
- Составьте <Запрос 2> на базе таблицы <Ведомость успеваемости>, в котором будут указаны студенты, имеющие не более 30 часов пропусков по неуважительной причине. Добавьте в этот запрос поле пропуски по уважительной причине в интервале от 30 часов до 45 часов (используйте оператор **Between... And...**)
- Составьте <Запрос> на базе таблицы <Личные данные>. Выведите список студентов, которым на данный момент, т.е. на сегодняшнее число, исполнилось уже 17 лет (используйте оператор **Between... And...**)

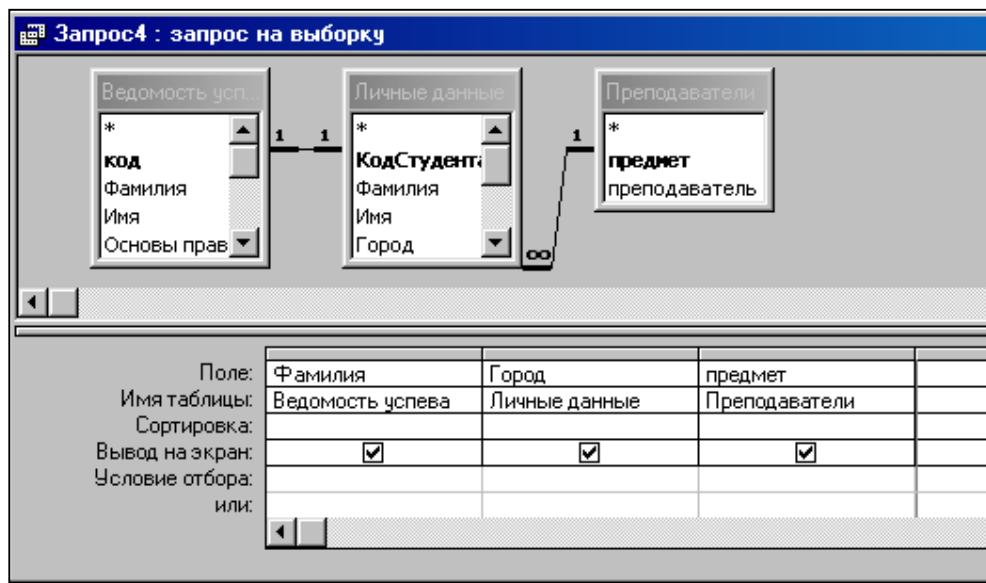
**Примечание:** Дата записывается с использованием символа #, например, #01.02.02.#

- Составьте запрос на базе трех таблиц <Ведомость успеваемости>, <Личные данные> и <Преподаватель>. Выберите студентов, которые проживают в Новороссийске и у которых любимый предмет «Менеджмент». Озаглавьте <Запрос 4>. Используйте <Конструктор запросов>.

В меню <Создание> выберите <Конструктор запросов>.

Добавьте все три таблицы в поле запроса. Закройте окно <Добавление таблицы>.

В первый столбец в строку <Поле> перетащите из первой таблицы с помощью мышки <Фамилия>, из второй таблицы во второй столбец <Город> и из третьей таблицы в третий столбец строки <Поле> - <Предмет> (Рисунок 3).



### *Рисунок 3*

В поле <Условия отбора> в столбце <Город> введите город «Новороссийск», в столбец <Предмет> введите «Менеджмент».

Сохраните запрос под именем <Запрос 4>.

Откройте запрос и проверьте результат проделанной работы.

Выполните запрос на создание новой таблицы, в которой должны быть поля <Фамилия>, <Имя>, <Пропуски по неуважительной причине>, <Город> и <Предмет>.

В меню <Создание> выберите <Конструктор запросов>.

Добавьте все три таблицы из списка окна <Добавление таблицы>. Закройте это окно.

В первую строчку <Поле> из первой таблицы перенесите в первый столбец поля <Фамилия>, во второй <Имя> и в третий <Пропуски по уважительной причине>, в четвертый столбец перетащите поле <Город> из второй таблицы и в последнем столбце будет поле <Предмет> из третьей таблицы.

Закройте запрос, сохранив его с именем <Запрос 5>.

Создайте *перекрестный запрос*.

Допустим, нужно посчитать для ведомости, сколько в группе человек получили по предмету «троек», «четверок» и «пятерок». Для этих целей используется *перекрестный запрос*.

В меню <Создание> выберите <Мастер запросов>.

В диалоговом окне выберите <Перекрестный запрос>, щелкните по кнопке <OK>.

В окне <Создание перекрестных запросов> выделите таблицу <Ведомость успеваемости> и щелкните <Далее>.

Выберите поля, значения которых будут использоваться в качестве заголовок строк – это <Фамилия> и <Имя>. Щелкните по кнопке <Далее>.

Выберите поле, значение которого будут использоваться в качестве заголовков столбцов, например <Менеджмент>. Щелкните по кнопке <Далее>.

Выберите функцию, по которой будут вычисляться значения ячеек на пересечении столбцов и строк (в данном случае **Count** – количество). Щелкните по кнопке <Далее>.

Задайте имя запроса <Итог по менеджменту> и щелкните по кнопке <Готово>.

### Самостоятельное задание

Составьте аналогичные запросы для оценок по трем другим предметам.

1) Откройте свою базу данных.

2) Создайте форму с помощью <Мастера форм> на базе таблицы <Ведомость успеваемости>.

Откройте таблицу <Ведомость успеваемости>.

Выберите закладку <Формы>, щелкните мышкой по кнопке <Другие формы>.



В появившемся диалоговом окне выберите <Мастер форм>.

В поле <Таблицы/Запросы> выберите таблицу <Ведомость успеваемости>, в поле <Доступные поля> выберите поля <Фамилия>, <Имя> и перенесите их стрелкой в поле <Выбранные поля>. Также перенесите поля с названием предметов, щелкните по кнопке <Далее>.

Выберите внешний вид формы – **Табличный**, щелкните по кнопке <Далее>.

Выберите требуемый стиль (н-р, **Обычная**), щелкните по кнопке <Далее>.

Задайте имя формы <Успеваемость> и щелкните по кнопке <Готово>. В результате получите форму, в которой можно менять данные и вводить новые значения.

Закройте форму.

3) Создайте форму на основе таблицы <Преподаватели>.

Откройте таблицу <Преподаватели>.

Выберите закладку <Формы>, щелкните мышкой по кнопке <Другие формы>. В появившемся диалоговом окне выберите <Мастер форм> .



Выберите внешний вид формы - <ленточный>.

Выберите любой стиль.

Получите готовую форму. Сохраните ее под именем <Преподаватели>.

Закройте форму.

4) Создайте форму <Личные данные> с помощью инструмента <Пустая форма>

- На вкладке **Создание** в группе **Формы** щелкните **Пустая форма**.

Access открывает пустую форму в режиме макета и отображает область **Список полей**.

- В области **Список полей** щелкните знак плюс (+) рядом с таблицей или таблицами, содержащими поля, которые нужно включить в форму.

• Чтобы добавить поле к форме, дважды щелкните его или перетащите его на форму. Чтобы добавить сразу несколько полей, щелкните их последовательно, удерживая нажатой клавишу CTRL. Затем перетащите выбранные поля на форму.

- Закройте окно списка полей.

- Перейдите в режим Конструктора

**Примечание 1** Размер окошка для названия поля и для его значений меняются мышкой.

Для этого выделите черный квадратик рамки (рамка станет цветной), установите курсор на границу рамки и с помощью двунаправленной стрелки измените размеры рамки.

**Примечание 2** С помощью кнопок панели инструментов **Шрифт** меняйте соответственно цвет фона, текста, линии/границы и т.д.

- Расположите элементы удобно по полю.

- Задайте размер текста поля <Фамилия> равным **24 пт**, шрифт - **синего цвета**.

- Увеличьте в высоту рамку поля <Фотография>.

- Сохраните форму с именем <Данные студентов>.

• Посмотрите все способы представления форм: в режиме **Конструктора**, режиме **Макета** и режиме **Форм**.

- Закройте форму.

5) Добавьте в таблицу <Личные данные> логическое поле <Институт> (т.е., собирается ли в дальнейшем учащийся поступать в институт). Значение этого поля <ДА> или <НЕТ>.

- Откройте таблицу <Личные данные> в режиме **Конструктор**. Добавьте поле с именем <Институт> и типом **Логический**. Закройте таблицу.

• Перейдите на закладку **Формы** и откройте форму <Данные студентов> в режиме **Конструктор**

- Щелкните по кнопке <Список полей> на панели инструментов, выделите название <Институт> и перетащите его мышкой в область данных, появиться значок надпись <Институт>.

- Расположите новые элементы по правилам оформления формы (с помощью мыши).

- Закройте <Список полей>

**Примечание 3** Если флагок установлен, поле в таблице имеет значение <ДА>, если снят, то <НЕТ>.

• Перейдите в режим <Раздельная форма> и посмотрите записи. Установите флагки у восьми разных учащихся.

- Закройте форму, ответив утвердительно на вопрос о сохранении.

6) Создайте кнопочную форму <Заставка> с помощью **Конструктора**.

- Щелкните по кнопке <Создать>.

• Выберите <Конструктор>. Появиться пустая форма. Задайте мышкой ширину формы, равную 10см, а высоту – 7см.

- Сохраните работу с именем <Заставка>.

- Откройте созданную форму <Заставка> в режиме Конструктора.
- Выберите на панели инструментов <Элементы управления> кнопку **Aa** – <Надпись>. Курсор мышки примет вид крестика с «приклеенной» буквой А. Щелкните мышкой по месту начала надписи и введите:

### База данных

#### «Гостиница»

#### группа ГС - 31

(после слов **База данных** нажмите одновременно комбинацию клавиш **Shift+Enter**.)

- Нажмите клавишу <Enter>. Выберите размер букв **18**, а выравнивание - **по центру**. Цвет фона – **голубой**. Растигните мышкой надпись на ширину окна.
- Выберите на панели элементов значок  **Кнопка**. Щелкните мышкой по тому месту области данных, где должна быть кнопка. Появится диалоговое окно <Создание кнопок>.
- Выберите категорию <Работа с формой>, а действие <Открыть форму>, и щелкните по кнопке <Далее>.
- Выберите форму <Успеваемость> , открываемую этой кнопкой щелкните по кнопке <Далее>. В следующем окне также щелкните по кнопке <Далее>.
- В следующем окне поставьте переключатель в положение <Текст>, наберите в поле слово <Успеваемость> (Рисунок 4) и щелкните по кнопке <Далее>.

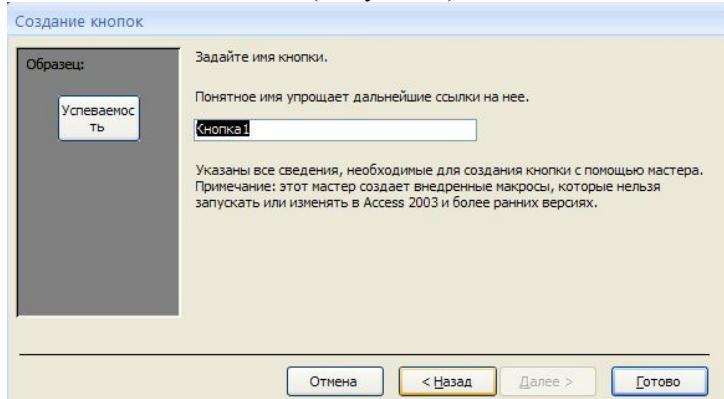


Рисунок 4

- Задайте имя кнопки <Успеваемость> и щелкните по кнопке <Готово>.

**Примечание 3** Размер и расположение кнопок можно менять мышкой в режиме Конструктор.

**Самостоятельно** создайте кнопки для форм <Личные данные> и <Преподаватели>.

- Перейдите в режим формы (Рисунок 5). Теперь при щелчке мышью по соответствующим кнопкам будут открываться соответствующие формы для работы.
- Закройте форму.

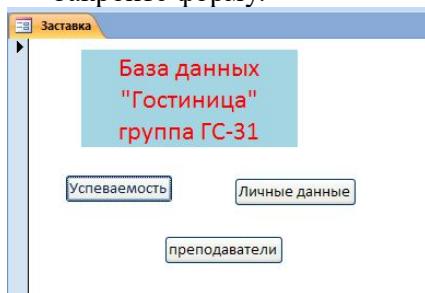


Рисунок 5

- 7) Создайте кнопочную форму при помощи **Диспетчера кнопочных форм**.

- ✓ Откройте вкладку **Работа с базами данных**, команда - **Диспетчер кнопочных форм**. Вы получите диалоговое окно, представленное на Рисунке 6.

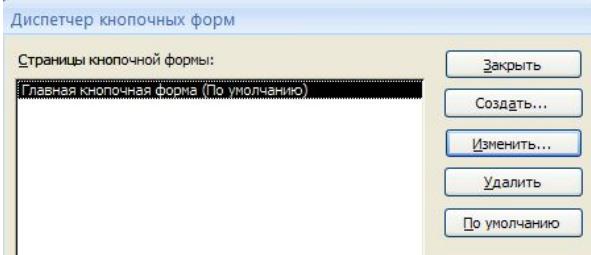


Рисунок 6

- ✓ Щелкните в этом окне по кнопке <Изменить>.
- ✓ В следующем окне щелкните по кнопке <Создать> и в появившемся окне измените содержимое полей в соответствии с Рисунком 7 (Команду и Форму выбирайте из списка, а не набирайте вручную). Щелкните по кнопке <OK>.

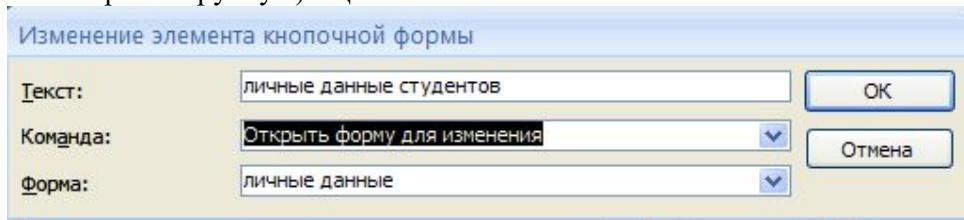


Рисунок 7

- ✓ Аналогично создайте еще три элемента кнопочной формы: <Успеваемость>, <Преподаватели> и <Заставка>.
- ✓ Добавьте кнопку закрытия базы данных. Для этого щелкните по кнопке <Создать>, наберите в поле **Текст** слово <Выход>, а в поле **Команда** выберите <Выйти из приложения>. Закройте диалоговые окна.
- ✓ Откройте окно <Кнопочная форма> в режиме Конструктора или Макета, измените цвет надписи и название вашей базы данных на **ГОСТИНИЦА**, сохраните форму.
- ✓ Украсьте вашу форму рисунком. Для этого щелкните по значку Эмблема и выберите в открывшемся окне папку с рисунками, выберите понравившийся и вставьте в свою кнопочную форму.
- ✓ Перейдите в режим формы, проверьте работу всех кнопок кнопочной формы. Завершите работу с базой данных, нажав на кнопку <Выход>.

**Задание 1.** Откройте свою базу данных.

**Задание 2.** Создайте отчет с помощью *Мастера отчетов*.

- Откройте вкладку *Создание*, меню *Отчеты*.
- Выберите *Мастер отчетов* и таблицу «Личные данные».
- Выберите нужные поля, которые будут участвовать в отчете, нажмите кнопку «Далее».
- В новом окне выберите поля для группировки так, чтобы сначала было указано поле «Фамилия», нажмите кнопку «Далее».
- На этом шаге отсортируйте данные по алфавиту, нажмите кнопку «Далее».
- Выберите вид макета *Ступенчатый* и щелкните по кнопке «Далее».
- Выберите стиль отчета: *Открытая* и щелкните по кнопке «Далее».
- Задайте имя отчета: «**Отчет1**» и щелкните по кнопке «Готово». Вы попадете в режим просмотра отчета.
- Закройте отчет согласившись с сохранением.

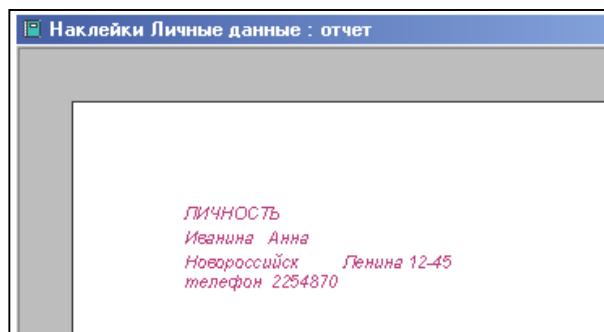
**Самостоятельно** Составьте еще два отчета по запросам – «Запрос 3» и «Запрос 5», выбирая из разных макетов: *блок*; *структура*, выбирая из разных стилей. Сохраните отчеты под именами «Отчет 2» и «Отчет 3».

**Задание 3.** Создайте Пустой отчет в столбец на базе таблицы «Ведомость успеваемости» и сохраните его с именем «Успеваемость».

С помощью Конструктора измените цвет букв заголовка, их размер и шрифт.

**Задание 4.** Создайте почтовые наклейки.

- Откройте вкладку *Создание*, меню *Отчеты*.
- Выберите таблицу «Личные данные», команда **Наклейки**.
- В следующем окне щелкните по кнопке «Далее».
- В следующем окне выберите шрифт, размер шрифта, насыщенность и цвет, вновь щелкните по кнопке «Далее».
- В следующем окне создайте прототип наклейки, напечатав слово **ЛИЧНОСТЬ** и выбрав соответствующие поля, щелкните по кнопке «Далее».
- В следующем окне укажите поля для сортировки (Фамилия, Имя), щелкните по кнопке «Далее».
- Введите имя отчета **«Наклейки»** и щелкните по кнопке «Готово».
- Просмотрите Наклейки.



## 2.14 Лабораторная работа №14 (12 часов)

**Тема:** «Интерфейс Microsoft Excel. Набор текстовой информации в Excel. Ввод и редактирование формул в Microsoft Excel. Построение диаграмм. Логические функции. Создание составных интегрированных документов»

**Перечень приборов, материалов, используемых в лабораторной работе:**

1. Компьютеры с установленным программным обеспечением
2. Методические указания к выполнению лабораторной работы
3. Рабочая тетрадь.

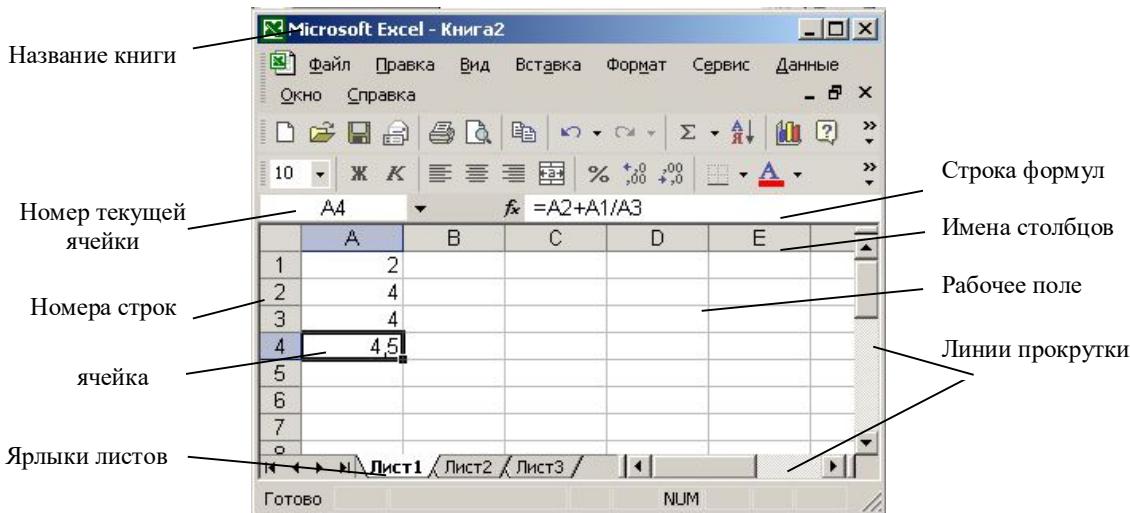
**Описание (ход) работы:**

Для расчетов данных и представления их в табличной форме, используют электронные таблицы EXCEL. Особенность электронных таблиц заключается в возможности применения формул, для описания связей между значениями различных ячеек. Расчет по заданным формулам выполняется автоматически. Изменение содержимого какой-либо ячейки приводит к пересчету значений всех ячеек, которые с ней связаны формулами отношениями. Документы созданные в среде EXCEL называют *рабочими книгами*. Рабочая книга представляет собой набор рабочих листов, каждый из которых имеет табличную

структуру и может содержать одну или несколько таблиц. Рабочие книги записываются как файлы с расширением xls.

Запуск программы EXCEL: Пуск, Программы, Microsoft Excel.

**Рабочий лист электронной таблицы EXCEL:**



Каждый рабочий лист имеет название, которое отображается на ярлычке листа, внизу листа. С их помощью можно переключаться с одного листа на другой. Чтобы **переименовать** его, надо щелкнуть два раза по его ярлыку.

*Рабочее поле EXCEL* – это электронная таблица, состоящая из столбцов и строк (ее размер 256 строк на 65536 строк), название столбцов – буква латинского алфавита, все строки пронумерованы. Стандартная рабочая книга состоит из трех листов. Листы можно добавить, выполнив команду – *Вставка, Лист*.

Пересечение строки и столбца образует **ячейку**. Местоположение ячейки называется **адресом**, образованным из имени строки и столбца (A1). Блок ячеек задается адресом A1:C1. Адреса ячеек используют при написании формул.

Перемещение между ячейками осуществляется с помощью мыши или нажатием клавиши TAB. Выделяют ячейки так же как и текст, а для выделения несмежных ячеек осуществляется с нажатой клавишей CTRL.

**Строка формул** – находится под панелью инструментов и она показывает номер ячейки и ее содержимое. (вывод строки формул – *Вид, Стока формул*)

## **ВВОД ТЕКСТОВОЙ ИНФОРМАЦИИ:**

После того, как написали текст нажмите клавишу Enter. Чтобы отменить внесенные изменения нажмите клавишу Esc на клавиатуре. Для очистки ячейки или выделенного диапазона нажмите клавишу Delete.

Чтобы исправить содержимое ячейки нужно два раза щелкнуть по ней мышкой или выделить ячейку и поводить исправления в строке формул. Когда вы вводите информацию в ячейку, то ее содержимое будет распространяться на другие ячейки до тех пор, пока они не заполнены. Ширину строки и столбца можно изменить захватив границу м.у. названиями строк или столбцов.

Чтобы установить **перенос по словам в ячейках**, нужно их выделить и выполнить команду *Формат ячеек* и во вкладке *выравнивание* установить галочку напротив *переносить по словам*.

## **ВВОД ФОРМУЛ:**

Любая формула начинается со знака = (формула пишется на английском языке, значение не имеет прописными или заглавными буквами)

Если ячейка содержит формулу, то в рабочем листе отображается текущий результат, а сама формула отображается в строке формул.

ПРИМЕР 1: Найти  $y=(a^3+b)*(a-b)$ . Вычислим это значение в ячейке B3.

	A	B	C	
1	a	3		
2	b	5		
3	y	$=(B1^3+B2)*(B1-B2)$		

При записи формул используют номера ячеек в которых находятся значения  $a$ ,  $b$ , и знаки +, -, \*, /, ^ (сложение, вычитание, умножение, деление и возвведение в степень). Если не поставлены скобки в формуле, то операции выполняются в таком порядке: ^, \*, /, +, -. Нерационально писать такую формулу  $=(3^3+5)*(3-5)$ , т.к. у рассчитывается для произвольных  $a$ ,  $b$ , а не для этих конкретных значений  $a$ ,  $b$ .

#### Абсолютные и относительные ссылки.

По умолчанию, ссылки в формулах рассматриваются как *относительные*, это означает, что при копировании ячейки формула в ней изменится.

При *абсолютной адресации* адреса ссылок при копировании не изменяются. Для изменения способа адресации при редактировании формулы надо выделить ссылку на ячейку и нажать F4. Элементы номера ячейки предваряются символом \$. Например, при последовательном нажатии клавиши F4 номер ячейки A1 будет записываться A1, \$A\$1, A\$1, \$A1, в двух последних случаях один из компонентов ячейки рассматривается как абсолютный, а другой – как относительный.

### ИСПОЛЬЗОВАНИЕ МАСТЕРА ФУНКЦИЙ.

Чтобы рассчитать например  $\sin a$ , нужно вызвать мастер функций: выполнив команду Вставка, Функция или нажать кнопку  $f(x)$  в строке формул.

В появившемся окне в списке Категории сначала выбрать категорию функций (математическую, статистическую, логическую или 10 недавно использовавшихся), далее в списке выбрать нужную функцию, как только вы выберите функцию, то внизу появится подсказка что эта функция делает. После этого нажимаете ОК

В следующем окне в строке число указываете адрес ячейки, в которой находится данное число.

Линия сетки таблицы EXCEL не видна при распечатки, чтобы оформить ваши вычисления в



таблицу нужно, выделить данные и на панели инструментов нажать значок границы (там выбрать нужный стиль обрамления).

### ПОСТРОЕНИЕ ДИАГРАММ И ГРАФИКОВ.

**Диаграмма – это графическое представление данных.**

Различают два вида диаграмм: *внедренные* – сохраняются на рабочем листе вместе с данными и диаграммные листы – диаграмма в полный экран на *отдельном листе*.

**Чтобы построить диаграмму нужно**

Выделить данные по которым строится диаграмма и выполнить команду Вставка,



диаграмма или нажать на панели инструментов кнопку мастер диаграмм

Построение диаграммы осуществляется в 4 шага.

**Шаг 1. Выбор типа диаграммы.**

Слева выбрать форму диаграммы (гистограмма, график...), а справа ее вид (простая, объемная и т.д.). нажать ДАЛЕЕ

## Шаг 2. Источник данных.

Во вкладке диапазон данных устанавливается диапазон, если вы его сразу не выделили. И то, как берутся значения по строчкам или по столбцам.

Во вкладке ряд устанавливаются подписи рядов и значений по оси ОХ.

Чтобы *подписать ряды* в строке *Ряд* выберите нужный ряд и напротив в строке имя напишите нужное имя.

Чтобы *подписать значения по оси ОХ*. Установи мигающий курсор в строке *подписи по оси ОХ* и выделите на рабочем листе ячейки в которых находятся подписи. и нажать **ДАЛЕЕ**.

## Шаг 3. Параметры диаграммы.

А) во вкладке заголовки вводим поочередно на соответствующую строку название диаграммы и подписи осей X, Y, Z и нажать **ДАЛЕЕ**

Б) во вкладке Легенда устанавливается расположение легенды.

## Шаг 4. Размещение диаграммы.

Указать где будет расположена диаграмма – *На отдельном* или *на этом листе*.

### **Редактирование диаграммы:**

Чтобы отредактировать любой элемент диаграммы щелкните по нему два раза мышкой и в появившемся окне выберите нужные вкладки или через контекстное меню, вызванному для любого элемента диаграммы (1-е четыре пункта меню позволяют вернуться к любому из четырех шагов (тип диаграммы, исходные данные, параметры диаграммы и размещение)).

### **Проверка вводимых данных.**

Вернемся к примеру1, поставим ограничения на вводимые данные в ячейках B1 и B2. Например, можно вводить только целые числа от -10 до 100. Для этого выделяем эти ячейки и выполняем команду *Данные, Проверка*. Во вкладке *Параметры* в строке *Тип данных* выбрать нужный тип (целое, действительное, дата и т.д.) и внизу указать минимальное и максимальное значение данных (минимум -10, максимум-100), Во вкладке *Сообщение для ввода* написать (например: введите целое число от -10 до 100), Во вкладке *Сообщение об ошибке* напишите текст который будет выходить в случае неправильного ввода данных (например: неправильно введены данные введите число от -10 до 100)

### **Условное форматирование:**

Например если в результате у нас получится число отрицательное то пусть выделит это число красным цветом. Выделяем ячейку и выполняем команду *Формат, Условное форматирование*. (в появившемся окне Условие1 указывают условие и формат (цвет размер начертание) ячейки как будет она выделяться среди других. Если нужно наложить форматирование на второе условие нажимаем кнопку *а также* и заполняем форматирование на условие 2.

### **Список.**

Часто при заполнении мы используем так называемые списки, например список номеров (1,2,...), дней недели (понедельник, вторник,...); месяцев (Январь, февраль,...). Чтобы постоянно не писать эти списки можно использовать автозаполнение. Мы пишем в одной ячейке Январь (не обязательно первый элемент списка, можно любой), затем устанавливаем курсор мыши в нижний правый угол и нажав левую кнопку протащить на нужное количество ячеек, когда мы отпустим мышь, то ячейки автоматически заполнятся. Можно создать свой список: например при заполнении ведомостей мы используем один и тот же список фамилий, и вот чтобы постоянно его не заполнять создадим свой. Это делается с помощью команды: *Сервис, Параметры* и во вкладке *Список* выбрать *Новый список* и напротив в строке *Элементы списка* ввести ваш список через запятую.

### **Фиксирование строк и столбцов.**

Часто при работе с большими документами нам нужно чтобы шапка таблицы была нам всегда видна, это можно сделать зафиксировав строки с шапкой таблицы: Выделить строку (столбец) до которой данные будут закреплены и выполнить команду: *Окно, закрепить области*.

## ЛОГИЧЕСКИЕ ФУНКЦИИ:

### ФУНКЦИЯ ЕСЛИ

Используется при проверке условий для значений и формул.

**ЕСЛИ (лог\_выражение;значение\_если\_истина;значение\_если\_ложь)**

Возращает одно значение, если заданное условие при вычислении дает значение ИСТИНА, и другое значение, если ЛОЖЬ.

**ПРИМЕР1:** Решить уравнение  $y = \begin{cases} x^2, & \text{если } x > 1 \\ x - 5, & \text{если } x \leq 1 \end{cases}$

	A	B
1	X	3
2	y	=Если (B1>1;B1^2; B1-5)

### ФУНКЦИЯ ИЛИ.

**ИЛИ (логическое\_значение1; логическое\_значение2; ...)** — от 1 до 30 проверяемых условий, которые могут иметь значение либо ИСТИНА, либо ЛОЖЬ.

Возращает значение ИСТИНА, если хотя бы один из аргументов имеет значение ИСТИНА; возвращает значение ЛОЖЬ, если все аргументы имеют значение ЛОЖЬ.

**ПРИМЕР2:** Если мы поменяется условие: если  $x=1$  или  $x=2$ , то  $y=x^2$ , иначе  $x-5$

$y = \begin{cases} x^2, & \text{если } x = 1 \text{ или } x = 2 \\ x - 5, & \text{если } x \neq 1 \text{ или } x \neq 2 \end{cases}$

Для выполнения этой задачи используются **ФУНКЦИЯ ИЛИ.**

	A	B
1	X	3
2	y	=Если (ИЛИ(B1=1;B1=2);B1^2; B1-5)

### ФУНКЦИЯ И.

**И (логическое\_значение1; логическое\_значение2; ...)** это от 1 до 30 проверяемых условий, которые могут иметь значение либо ИСТИНА, либо ЛОЖЬ.

Возращает значение ИСТИНА, если все аргументы имеют значение ИСТИНА; возвращает значение ЛОЖЬ, если хотя бы один аргумент имеет значение ЛОЖЬ.

**ПРИМЕР3:** А если проверяется двойное условие  $1 < x < 10$ , то  $y=x^2$ , иначе  $x+2$ , для выполнения этой задачи используются **ФУНКЦИЯ И.**

	A	B
1	X	3
2	y	=Если (И(B1>1;B1<10);B1^2; B1+2)

### ВЛОЖЕННАЯ ФУНКЦИЯ ЕСЛИ

А если будет не два а три, четыре и т.д. условий для этого используют **ВЛОЖЕННУЮ ФУНКЦИЮ ЕСЛИ**

$y = \begin{cases} x^2, & \text{если } x < 1 \\ x - 5, & \text{если } 1 \leq x \leq 10 \\ \sin x, & \text{если } x > 10 \end{cases}$

	A	B
1	X	3
2	y	=Если (B1<1;B1^2;Если(B1<=10;B1-5;sin(b1)))

До 7 функций ЕСЛИ могут быть вложены друг в друга в качестве значений аргументов.

## **СОЗДАНИЕ СОСТАВНЫХ ИНТЕГРИРОВАННЫХ ДОКУМЕНТОВ.**

Часто при создании документа возникает необходимость вставить в него фрагмент из документов созданных в других программных средствах. Например в текст созданый в текстовом редакторе Word нужно включить рисунок типа *bmp* (созданный в *paint*), а так же таблицу и диаграмму созданные в *Excel*.

*Составной (интегрированный) документ* – документ, в котором связанные между собой фрагменты имеют разные типы и создавались в разных приложениях или программах.

### **Создание составного документа через буфер обмена.**

- выделить объект
- выполнить команду *Правка, Копировать*,
- установить курсор в место вставки
- выполнить команду *Правка, Вставить*

### **Создание составного документа через буфер обмена по технологии OLE.**

Основное назначение технологии OLE (связывание и внедрение объекта) – это создание связи между разнотипными объектами Windows. Не все Windows-приложения поддерживают технологию OLE, например текстовый редактор Word Pad полностью обеспечивает работу по технологии OLE он предназначен для создания текстового документа в котором могут быть рисунки, звуки, таблицы, видеоклипы и пр. А графический редактор Paint может использоваться только как источник. Технология OLE обеспечивает пользователю возможность редактировать вставленный объект средствами, в которых он был создан. OLE-объект всегда сохраняет связь с источником (которая активизируется двойным щелчком).

Технология OLE предусматривает два варианта обмена данными:

- внедрение объекта при котором создается дубль объекта, который вставляется в документ при этом теряется связь с исходным документом, но связь с приложением остается. Внедрение объекта осуществляется с помощью команды Вставка, Объект. Чтобы внедрить объект из уже открытого документа нужно скопировать объект (Правка, Копировать), переходим в приложение и выполняем команду Правка, Специальная вставка (на экране появится диалоговое окно Специальная вставка, в котором нужно выбрать формат вставляемого объекта и указать способ вставки: обычный или в виде значка, флажок связь с файлом не устанавливать, для того чтобы содержимое файла оказалось внедренным).
- связывание объекта, при этом способе отличие объекта будет заметно только при редактировании (способ вставки такой же только устанавливаем флажок напротив связь с файлом). При просмотре составного документа связанный объект ни чем не будет отличаться от внедренного, но только при этом способе объект остается в документе источнике, а на него в документе имеется ссылка (указатель). Таким образом связанный объект хранится в одном экземпляре в документе источнике и любые изменения в нем сразу отразятся в документе-источнике, обратное не верно, заметим что при открытии такого составного документа нужно позаботится чтобы на диске присутствовали все файлы, которые связаны с ним.

Этот способ удобен в том случае если объект вставлен в несколько документов и любые изменения должны незамедлительно отражаться в этих документах и из-за экономия дискового пространства.

### **Вставка объекта как рисунок.**

Но если вам не нужны связи объекта с источником, то удобно вставить объект (например диаграмму) как рисунок (векторный (можно разгруппировать и механически удалить не нужные элементы) или растровый).

1. выделить объект
2. нажать клавишу SHIFT и удерживая ее выполнить команду *Правка, Копировать рисунок...*
3. в появившемся окне выбрать *внешний вид*: как на экране, и *формат рисунка* (векторный или растровый)

4. перейти в документ, установить мигающий курсор в место вставки и выполнить команду *Правка, Вставить*.

## 2.15 Лабораторная работа №15 (4 часа)

**Тема:** «Основы программирования на языке Pascal. Линейный алгоритмы»

**Перечень приборов, материалов, используемых в лабораторной работе:**

1. Компьютеры с установленным программным обеспечением
2. Методические указания к выполнению лабораторной работы
3. Рабочая тетрадь.

**Описание (ход) работы:**

Запустить интегрированную среду Turbo Pascal:

В окне редактирования введите текст программы:

<b>Program Tutor01;</b>	1
<b>Var A, B, SUMMA : Integer;</b>	2
<b>Begin</b>	3
<b>Write ('Введите целое число A - '');</b>	4
<b>Readln (A);</b>	5
<b>Write ('Введите целое число B - '');</b>	6
<b>Readln (B);</b>	7
<b>SUMMA := A+B;</b>	8
<b>Write ('Сумма чисел - ', SUMMA);</b>	9
<b>End.</b>	10

Пока текст программы имеет имя NONAME00, т.е. ему не присвоено конкретное имя. Сохраните текст программы на диске под именем T01.PAS, для чего клавишами Alt + F перейдите в меню, выберите пункт "Save as..." (записать под новым именем) и в окне ввода задайте имя программы T01.PAS.

Тоже самое можно выполнить, войдя в главное меню с помощью клавиши F10, выбрать пункт меню File, а затем "Save as...".

Откомпилируйте и выполните программу, используя клавиши Ctrl + F9.

Если при выполнении пункта 5 в программе были ошибки, то компилятор выдаст сообщение и установит курсор на первую ошибочную запись. Исправьте их и выполните пункт 5 еще раз.

Для того, чтобы просмотреть результат выполнения программы (экран пользователя) используйте клавиши Alt + F5.

Для возврата в окно редактирования нажмите клавишу Esc.

Выходите из интегрированной среды Turbo Pascal с помощью клавиш Alt + X или через меню.

Запустить интегрированную среду Turbo Pascal.

Загрузите в окно редактирования файл T01.PAS для чего клавишами Alt + F перейдите в меню, выберите пункт "Open" (открыть файл) или нажмите клавишу F3.

На экран компьютера будет выведено окно выбора открываемого файла из списка. Переход от одного элемента к другому осуществляется с помощью клавиши Tab. Каждый элемент, когда он является активным, выделяется.

Установите курсор на имени файла T01.PAS, после этого нажатием клавиши Tab выберите кнопку [OPEN].

Вставить комментарий после первой строки. Для этого подвести курсор к концу первой строки и нажать Enter, а на новой строке набрать текст:

**{ первая программа }**

Текст программы будет выглядеть:

Program Tutor01;	1
<b>{ первая программа }</b>	
Var A, B, SUMMA : Integer;	2
Begin	3
Write ('Введите целое число A - ');	4
Readln (A);	5
Write ('Введите целое число B - ');	6
Readln (B);	7
SUMMA := A+B;	8
Write ('Сумма чисел - ', SUMMA);	9
End.	10

Выделить блок комментарий и поместить его в *карман*.

Для этого подвести курсор к началу блока и клавишами Shift + → выделить необходимый блок, затем нажать клавиши Ctrl + Ins.

Скопировать текст в строку с номером три.

Для этого подвести курсор к нужной позиции строки с номером три и нажать клавиши Shift + Ins.

Удалить строку комментарий, находящуюся после первой строки.

Для этого подвести курсор к удаляемой строке и нажать клавиши Ctrl + Y.

Сохранить текст программы.

**Задание:**

Измените программу для нахождения суммы трех чисел.

**Вывод целых чисел**

**В окне редактора наберите программу:**

Program Tutor02;	1
Var A, B: Integer;	2
Begin	3
Write ('Введите числа A, B - ');	4
Readln (A, B);	5
WriteLn ('*****');	6
WriteLn (A);	7
WriteLn (B);	8
WriteLn (A, A);	9
End.	10

Примечание:

1. В тексте программы используется строка №6 для того, чтобы легче было подсчитывать количество выводимых знаков и номер выводимой позиции.

2. При записи результатов необходимо указывать пробелы от начала строки и между цифрами и символами.

Выполните программу, введя исходные значения A = 134, B = 5671.

Запишите в тетрадь полученные результаты в виде таблицы, заполнив первый и второй столбцы:

Выражение	Результат	Выражение	Результат
WriteLn (A)		WriteLn (A:6)	
WriteLn (B)		WriteLn (B:10)	
WriteLn (A,A)		WriteLn (A:8, A:6)	

Как изменятся результаты вывода, если внести изменения в программу, задав параметр форматного вывода указывающий на общее количество выводимых знаков? Запишите в тетрадь полученные результаты, заполнив третий и четвертый столбцы таблицы.

### **Вывод вещественных чисел**

Внесите изменения в текст программы:

**Var A, B: Real;**  
**WriteLn (A/2);**

вместо строки № 2  
вместо строки № 9

Ведите исходные данные A = 75.432, B = -1.9156e+01 и запишите в тетрадь полученные результаты в виде таблицы, заполнив первый и второй столбцы:

Выражение	Результа т	Выражение	Результа т	Выражение	Результа т
WriteLn (A)		WriteLn (A:15)		WriteLn (A:8:4)	
WriteLn (B)		WriteLn (B:10)		WriteLn (B:10:2)	
WriteLn (A /2)		WriteLn (A /2:12)		WriteLn (A/2:5:3)	

Как изменятся результаты вывода, если внести изменения в программу, задав параметр форматного вывода указывающий на общее количество выводимых знаков? Запишите в тетрадь полученные результаты, заполнив третий и четвертый столбцы таблицы.

Как изменятся результаты вывода, если внести изменения в программу, задав параметр форматного вывода указывающий дополнительно количество цифр после десятичной точки? Запишите в тетрадь полученные результаты, заполнив пятый и шестой столбцы таблицы.

### **Вывод данных символьного типа**

Внесите изменения в текст программы:

**Var A, B: Char;**  
**WriteLn (A, A,A);**

вместо строки № 2  
вместо строки № 9

Ведите исходные данные A='X', B='S' и запишите в тетрадь полученные результаты в виде таблицы, заполнив первый и второй столбцы:

Выражение	Результат	Выражение	Результат
WriteLn (A)		WriteLn (A:3)	
WriteLn (B)		WriteLn (B: 5)	
WriteLn (A , A, A)		WriteLn (A:2 , A:4, A:3)	

Как изменятся результаты вывода, если внести изменения в программу, задав параметр форматного вывода указывающий на общее количество выводимых знаков? Запишите в тетрадь полученные результаты, заполнив третий и четвертый столбцы.

### **Вывод данных логического типа**

Внесите изменения в текст программы (измененные строки выделены жирным шрифтом):

Program Tutor02;

**Var A, B: Boolean;**

Begin

1

2

3

WriteLn ('*****');	4
<b>A:= True; B:= False;</b>	5
WriteLn (A);	6
WriteLn (B);	7
<b>WriteLn (not A);</b>	8
End.	9

Ведите исходные данные и запишите в тетрадь полученные результаты в виде таблицы, заполнив первый и второй столбцы:

Выражение	Результат	Выражение	Результат
WriteLn (A)		WriteLn (A:6)	
WriteLn (B)		WriteLn (B:10)	
WriteLn (not A, B)		WriteLn (not A:7, B:3)	

Как изменятся результаты вывода, если внести изменения в программу, задав параметр форматного вывода указывающий на общее количество выводимых знаков? Запишите в тетрадь полученные результаты, заполнив третий и четвертый столбцы таблицы.

## 2.16.Лабораторная работа №16 (4 часа)

**Тема:** «Основы программирования на языке Pascal. Логические функции»

**Цель работы:** Познакомиться с основами программирования на языке Pascal.

**Перечень приборов, материалов, используемых в лабораторной работе:**

1. Компьютеры с установленным программным обеспечением
2. Методические указания к выполнению лабораторной работы
3. Рабочая тетрадь.

**Описание (ход) работы:**

1. Запустить интегрированную среду Turbo Pascal.
2. В окне редактора наберите программу:

Uses Crt;	1
Var I, START, FINISH : Integer;	2
Begin	3
ClrScr;	4
Write ('Введите начальное значение - '');	5
ReadLn (START);	6
Write ('Введите конечное значение - '');	7
ReadLn (FINISH);	8
For I := START To FINISH do	9
WriteLn ('Значение I = ', I)	10
End.	11

3. Выполните программу, введя различные исходные значения для переменных START, FINISH, и запишите в тетрадь полученные результаты в виде таблицы:

№ п/п	Значения START	Значения FINISH	Результат вывода на экран

1	2	5	
2	2	2	
3	4	3	

4. Исправьте программу, заменив строку с номером 9 на оператор:

**For I := START To FINISH do**

5. Выполните программу, введя различные исходные значения для переменных START, FINISH, и запишите в тетрадь полученные результаты в виде таблицы:

№ п/п	Значения START	Значения FINISH	Результат вывода на экран
1	2	5	
2	2	2	
3	4	3	

6. Что будет выведено на экран, после выполнения программы и почему:

```
Var I, FINISH : Integer;
Begin
  FINISH := 3;
  For I := 1 To FINISH do
    Begin
      I := I + 1 ;
      WriteLn ( 'Значение I = ' , I )
    End
  End.
```

### ***Вопросы для повторения***

1. Решение каких задач можно осуществлять, используя циклический алгоритм?
2. Для чего используется оператор For?
3. Чем отличается форма For - to от For - downto?
4. Каков порядок выполнения оператора For?
5. Какое значение (тип выражения) может принимать начальное и конечное значение переменной цикла?
6. При каком условии оператор цикла не будет выполняться?

### **2.17 Лабораторная работа №17 (4 часа)**

**Тема:** «Основы программирования на языке Pascal. Разветвляющие алгоритмы»

**Цель работы:** Познакомиться с основами программирования на языке Pascal.

**Описание (ход) работы:**

3. Запустить интегрированную среду Turbo Pascal.

### **Параметры-значения**

4. В окне редактора наберите программу:

Program Trial_1;	1
Var X,Y : Integer;	2
Procedure Reverse ( A, B : Integer );	3
Begin	4
A := A * 10; B := B + 5;	5
WriteLn ( B : 3, A : 3)	6
End;	7
Begin	8
X := 1; Y := 100;	9
Reverse (X, Y);	10
Reverse (4, 5);	11
Reverse (4 * X, 5 * Y);	12
End.	13

3. Выполните программу и запишите в тетрадь полученные результаты в виде таблицы:

№ п/п	Формальные параметры	Фактические параметры	Результат вывода на экран
1	A	B	X=1 Y=100
2			
3			

Выполните программу еще раз, но в пошаговом режиме с прослеживанием работы процедуры (F7). Задайте для просмотра в окне отладчика переменные X, Y и A, B. Обратите внимание как меняются значения этих переменных.

### Параметры-переменные

4. В окне редактора наберите программу:

Program Trial_2;	1
Var I : Integer;	2
Procedure Count ( X : Integer );	3
Begin	4
X := X + 1 ;	5
Write ( ' X=' , X );	6
End;	7
Begin	8
I := 5;	9
Write ( ' I1=' , I );	10
Count ( I );	11
Write ( ' I2=' , I );	12
End.	13

5. Выполните программу и запишите в тетрадь полученные результаты в виде таблицы, заполнив 1, 2 и 3 графы:

№ п/п	Переменная	Результат вывода на экран для процедуры Procedure Count (X : Integer );	Результат вывода на экран для процедуры Procedure Count (Var X: Integer);

1	I 1		
2	X		
3	I 2		

6. Внесите изменения в программу, заменив строку 3 на следующий текст:

Procedure Count ( Var X : Integer );

7. Выполните программу и запишите в тетрадь полученные результаты, заполнив в таблице графу 4.

8. Выполните программу еще раз, но в пошаговом режиме с прослеживанием работы процедуры (F7). Задайте для просмотра в окне отладчика переменные I и X. Обратите внимание как меняются значения этих переменных. Почему получены такие результаты, и как это объяснить?

## 2.18 Лабораторная работа №18 (6 часов)

**Тема:** «Основные математические вычисления в MathCad»

**Цель работы:** Освоить работу с процессом символьных вычислений на примере вычисления интегралов, производных, сумм, пределов. Изучить работу с векторами и матрицами

### 2.1.2 Задачи работы:

1. По заданным координатам точек A, B, C, D найти координаты векторов  $a=AB$  и  $b=CD$ .
2. Вычислить скалярное и векторное произведения найденных векторов.
3. Найти следующие произведения векторов на заданную матрицу M:  $a^*M$  и  $M^*b$ .
4. Вычислить определитель матрицы M.
5. Для заданного ряда вычислить  $i$ -частичную сумму и исследовать сходимость ряда.
6. Вычислить сумму ряда.
7. Найти первообразную неопределенного интеграла и выполнить проверку, полученного результата.
8. Вычислить значения определенного интеграла.

### Перечень приборов, материалов, используемых в лабораторной работе:

1. Компьютеры с установленным программным обеспечением
2. Методические указания к выполнению лабораторной работы
3. Рабочая тетрадь.

### Описание (ход) работы:

#### Упражнение 1. Вычислить:

$$\sqrt{100} = \quad | -10 | = \quad 10! = \quad .$$

Это и все остальные задания снабдить комментариями, используя команду **Вставка**  $\Rightarrow$  **Текстовая область**.

**Упражнение 2.** Определить переменные:  $a := 3.4$ ,  $b := 6.22$ ,  $c \equiv 0.149$  (причем переменную  $c$  - глобально) и выражения:

$$Z := \frac{2ab + \sqrt[3]{c}}{\sqrt{(a^2 + b^2 + c^2) \cdot c}} \quad N := e^{\sin c} \cos \frac{a}{b}.$$

- Вычислить выражения.
- С помощью команды **Формат**  $\Rightarrow$  **Результат**  $\Rightarrow$  **Формат** **чисел**  $\Rightarrow$  **Число** **знаков** изменить точность отображения результатов вычисления **глобально**.

**Упражнение 3.** Вывести на экран значение системной константы  $\pi$  и установить максимальный формат ее отображения локально.

**Упражнение 4.** Выполнить следующие операции с комплексными числами:

$$Z := -3 + 2i \quad |Z| = \quad \operatorname{Re}(Z) = \quad \operatorname{Im}(Z) = \quad \operatorname{arg}(Z) = \\ \sqrt{Z} = \quad \sqrt{-5} = \quad 2 \cdot Z = \quad Z1 := 1 + 2i \quad Z2 := 3 + 4i$$

$$Z1 + Z2 = \quad Z1 - Z2 = \quad Z1 \cdot Z2 = \quad Z1/Z2 =$$

**Упражнение 5.** Выполнить следующие операции:

$$i := 1 \dots 10 \quad \sum_i i = \quad \prod_i (i+1) = \quad \int_0^{0.4} x^2 \cdot \lg(x+2) dx = \quad \int_{0.8}^{1.2} \frac{\operatorname{ctg} 2x}{(\sin 2x)^2} dx = \quad x := 2 \\ \frac{d}{dx} x^5 = \quad \frac{d}{dx} \sin(x) =$$

**Упражнение 6.** Определить векторы  $d$ ,  $S$  и  $R$  через дискретный аргумент  $i$ . Отобразить графически таблично заданные функции  $S_i(d_i)$  и  $R_i(d_i)$ , используя команду **Вставка**⇒**График**⇒**X-Y**

**Зависимость.** Чтобы оформить график, необходимо выполнить следующие команды:

$i$	$d_i$	$S_i$	$R_i$
0	0.5	3.3	2
1	1	5.9	3.9
2	1.5	7	4.5
3	2	6.3	3.7
4	2.5	4.2	1.2

- Щелкнуть левой клавишей мыши на графике, чтобы выделить его. Затем щелкнуть правой клавишей мыши, при этом появится контекстное меню в котором необходимо выбрать команду **Формат** (появится диалоговое окно **“Formatting Currently Selected X-Y Plot”**).

- Нанести линии сетки на график (**Оси X-Y** ⇒ **Вспом. линии**) и отобразить легенду (**След**⇒**Скрыть легенду**)

- Отформатировать график так, чтобы в каждой узловой точке графика функции  $S_i(d_i)$  стоял знак вида (**След**⇒**Символ**⇒**box**), а график функции  $R_i(d_i)$  отобразить в виде гистограммы (**След**⇒**Тип**⇒**bar**).

**Упражнение 7.** Построить декартовы (**X-Y Зависимость**) и полярные (**Полярные Координаты**) графики следующих функций:

$$X(\alpha) := \cos(\alpha) \cdot \sin(\alpha)$$

$$Y(\alpha) := 1.5 \cos(\alpha)^2 - 1$$

$$P(\alpha) := \cos(\alpha).$$

Определить по графику **X-Y Зависимость** координаты любой из точек пересечения графиков  $Y(\alpha)$  и  $P(\alpha)$ , для этого необходимо:

- Выделить график и выбрать из контекстного меню **Масштаб** (появится диалоговое окно **“X-Y Zoom”**) для увеличения части графика в области точки пересечения.
- На чертеже выделить пунктирным прямоугольником окрестность точки пересечения графиков  $Y(\alpha)$  и  $P(\alpha)$ , которую нужно увеличить.
- Нажать кнопку **Масштаб+**, чтобы перерисовать график.
- Чтобы сделать это изображение постоянным, выбрать **OK**.
- Выбрать из контекстного меню **Трассировка** (появится диалоговое окно **“X-Y Trace”**).
- Внутри чертежа нажать кнопку мыши и переместить указатель мыши на точку, чьи координаты нужно увидеть.

Для этого необходимо определить  $\alpha$  как дискретный аргумент на интервале от 0 до  $2\pi$  с шагом  $\pi/30$ .

- Выбрать **Copy X** (или **Copy Y**), на свободном поле документа набрать  $Xper :=$  (или  $Yper :=$ ) и выбрать пункт меню **Правка**⇒**Вставка**. Вычислить значения функций  $X(\alpha)$  и  $Y(\alpha)$  при  $\alpha := \pi/2$ .

**Упражнение 8.** Используя команду **Вставка**⇒**Матрица** создать матрицу  $Q$  размером  $6 \times 6$ , заполнить ее произвольно и отобразить графически с помощью команды **Вставка**⇒**График**⇒**Поверхности**.

**Упражнение 9.** Построить график поверхности (**Поверхности**) и карту линий уровня (**Контурный**) для функции двух переменных

$X(t, \alpha) := t \cdot \cos(\alpha) \cdot \sin(\alpha)$ , двумя способами:

1. С помощью функции *CreateMesh* (сетка размером  $40 \times 40$ , диапазон изменения  $t$  от  $-5$  до  $5$ ,  $\alpha$  - от  $0$  до  $2\pi$ ).

2. Задав поверхность математически, для этого:

- Определить функцию  $X(t, \alpha)$
- Задать на осях переменных  $t$  и  $\alpha$  по 41 точке

$i := 0..40 \quad j := 0..40$

для переменной  $t_i$  со значениями, изменяющимися от  $-5$  до  $5$  с шагом  $0.25 \quad t_i := -5 + 0.25 \cdot i$ , а для переменной  $\alpha_j$  - от  $0$  до  $2\pi$  с шагом  $\pi/20 \quad \alpha_j := \pi/20 \cdot j$ .

- Определить матрицу  $M_{i,j} := X(t_i, \alpha_j)$  и отобразить ее графически.

С помощью команды **Формат** контекстного меню вызвать диалоговое окно “**Формат 3-Д графика**” и изменить:

- характеристики просмотра (**Общее**⇒**Вид**⇒**Вращение, Наклон**),
- цвета и линии поверхности (**Внешний Вид**⇒**Свойства линии, Свойства заливки**),
- параметры осей (**Оси**),
- вид заголовка графика (**Название**).

**Упражнение 10.** Отобразить графически пересечение поверхностей  $f1(x, y) := \frac{(x+y)^2}{10}$  и

$f2(x, y) := 5 \cdot \cos\left(\frac{x-y}{3}\right)$ . Матрицы для построения поверхностей задать с помощью функции *CreateMesh*, значения факультативных параметров не указывать. Выполнить однотонную заливку для поверхностей, выбрав из контекстного меню команду **Формат**. Также из контекстного меню выбрать эффекты **Туман, Освещение, Перспектива**.

**Упражнение 11.** Используя переменную **FRAME** и команду **Вид** ⇒ **Анимация**, создать анимационные клипы с помощью данных приведенных в Таблице 1.

**Упражнение 12.** Построить график функции  $f(x)$  (Таблица 1) и приблизительно определить один из корней уравнения. Решить уравнение  $f(x) = 0$  с точностью  $\varepsilon = 10^{-4}$  с помощью встроенной функции Mathcad *root*;

**Упражнение 13.** Для полинома  $g(x)$  (Таблица 2) выполнить следующие действия:

- 1) с помощью команды **Символы** ⇒ **Коэффициенты полинома** создать вектор  $V$ , содержащий коэффициенты полинома;
- 2) решить уравнение  $g(x) = 0$  с помощью функции *polyroots*;
- 3) решить уравнение символьно, используя команду **Символы** ⇒ **Переменные** ⇒ **Вычислить**.

**Упражнение 14.** Решить систему линейных уравнений (Таблица 3):

- 1) используя функцию *Find*;
- 2) матричным способом и используя функцию *Isolve*.

**Упражнение 15.** Преобразовать нелинейные уравнения системы из Таблицы 4 к виду  $f_1(x) = y$  и  $f_2(y) = x$ . Построить их графики и определить начальное приближение решения. Решить систему нелинейных уравнений с помощью функции *Minerr*.

**Упражнение 16.** Символьно решить системы уравнений:

$$\begin{cases} 3x + 4\pi y = a, \\ 2x + y = b, \\ 3y + x = c. \end{cases}$$

**Упражнение 17.** Используя операцию **Символы**  $\Rightarrow$  **Расчеты**  $\Rightarrow$  **С плавающей запятой...**, представьте:

- 1) число  $\pi$  в 7 позициях;
- 2) число 12,345667 в 3 позициях.

**Упражнение 18.** Выведите следующие числа в комплексной форме, используя операцию **Расчеты**  $\Rightarrow$  **Комплексные** меню **Символы**:

- 1)  $\sqrt{-7}$ ;
- 2)  $\operatorname{tg}(a\sqrt{-3})$ ;
- 3)  $e^{\frac{1+\pi i}{4}}$ ;

4) для выражения 3) последовательно выполните операции **Расчеты**  $\Rightarrow$  **Комплексные** и **Упростить** меню **Символы**.

**Упражнение 19.** Для полинома  $g(x)$  (см. Таблица 1) выполнить следующие действия:

- 1) разложить на множители, используя операцию **Символы**  $\Rightarrow$  **Фактор**;
- 2) подставьте выражение  $x = y + z$  в  $g(x)$ , используя операцию **Символы**  $\Rightarrow$  **Переменные**  $\Rightarrow$  **Замена** (предварительно скопировав подставляемое выражение в буфер обмена, выделив его и нажав комбинацию клавиш **Ctrl + C**);
- 3) используя операцию **Символы**  $\Rightarrow$  **Расширить**, разложите по степеням выражение, полученное в 2);
- 4) используя операцию **Символы**  $\Rightarrow$  **Подобные**, сверните выражение, полученное в 3), по переменной  $z$ .

**Варианты упражнения 3** **Упражнение 4.** Разложите выражения на элементарные дроби используя операцию **Символы**  $\Rightarrow$  **Переменные**  $\Rightarrow$  **Преобразование в частичные доли**:

- 1)  $\frac{6x^2 - x + 1}{x^3 - x}$ ;
- 2)  $\frac{3x^2 - 2}{(x^2 + x + 1)(x + 1)}$ ;
- 3)  $\frac{x + 1}{x(x - 1)^3}$ ;
- 4)  $\frac{5x^2 - 4x + 16}{(x^2 - x + 1)^2(x - 3)}$ .

**Упражнение 20.** Разложите выражения в ряд с заданной точностью, используя операцию **Символы**  $\Rightarrow$  **Переменные**  $\Rightarrow$  **Разложить на составляющие**:

- 1)  $\ln(1 + x)$ ,  $x_0 = 0$ , порядок разложения 6;
- 2)  $\sin(x)^2$ ,  $x_0 = 0$ , порядок разложения 6.

**Упражнение 21.** Найти первообразную аналитически заданной функции  $f(x)$  (Таблица 4), используя операцию **Символы**  $\Rightarrow$  **Переменные**  $\Rightarrow$  **Интеграция**.

**Упражнение 22.** Определить символьное значение первой и второй производных  $f(x)$  (Таблица 4), используя команду **Символы**  $\Rightarrow$  **Переменные**  $\Rightarrow$  **Дифференциалы**.

**Упражнение 23.**

- 1) Транспонируйте матрицу  $M$
- $$\begin{pmatrix} 1 & a & b \\ x & 2 & c \\ x^2 & 3 & d \end{pmatrix}$$

с помощью операции **Символы**  $\Rightarrow$  **Матрицы**  $\Rightarrow$  **Транспонирование**.

- 2) Инвертируйте матрицу

$$\begin{pmatrix} 1 & y \\ x & 2 \end{pmatrix}$$

с помощью операции **Символы**  $\Rightarrow$  **Матрицы**  $\Rightarrow$  **Инвертирование**.

3) Вычислите определитель матрицы  $M$

$$\begin{pmatrix} 1 & a & b \\ x & 2 & c \\ x^2 & 3 & d \end{pmatrix}$$

с помощью операции **Символы**  $\Rightarrow$  **Матрицы**  $\Rightarrow$  **Определитель**.

**Упражнение 24.** Вычислите пределы:

)  $\lim_{x \rightarrow 1} \frac{x^2 + 2 \cdot x + 5}{x^2 + 1}$  )  $\lim_{x \rightarrow \frac{\pi}{2}} (2 \cdot \sin(x) - \cos(x) + \operatorname{ctg}(x))$

)  $\lim_{h \rightarrow 0} \frac{(x+h)^3 - x^3}{h}$  )  $\lim_{x \rightarrow \infty} \frac{\sqrt[3]{x^2 - 3}}{\sqrt[3]{x^3 + 1}} \rightarrow 1$

)  $\lim_{x \rightarrow +\infty} x \cdot (\sqrt{x^2 + 1} - x)$  )  $\lim_{x \rightarrow -\infty} x \cdot (\sqrt{x^2 + 1} - x)$

)  $\lim_{x \rightarrow 0^+} (1+x)^{\frac{1}{x}}$  )  $\lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n$

**Упражнение 25.** Задайте операторы пользователя:

1) Для пересчета единиц электрической энергии (кВт·ч в Дж, эВ в Дж) если известно, что

1 кВт·ч =  $3,6 \cdot 10^6$  Дж;

1 эВ =  $1,602 \cdot 10^{-19}$  Дж.

2) Для пересчета единиц магнитной индукции (Вб/см<sup>2</sup> в Т, Гс в Т) если известно, что

1 Вб/см<sup>2</sup> =  $1 \cdot 10^4$  Т;

1 Гс =  $1 \cdot 10^{-4}$  Т.

3) Для пересчета единиц мощности (эр/с в Вт, кгс·м/с в Вт) если известно, что

1 эрг/с =  $1 \cdot 10^{-7}$  Вт;

1 кгс·м/с = 9,80665 Вт.

### 3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРОВЕДЕНИЮ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

#### 3.1 Практическое занятие №1 (2 часа).

Тема: «Системы счисления»

##### 3.1.1 Вопросы к занятию:

- Перевести данное число из десятичной системы счисления в двоичную, восьмеричную и шестнадцатеричную системы счисления.
- Перевести данное число в десятичную систему счисления.
- Сложить числа.
- Выполнить вычитание.
- Выполнить умножение.

##### 3.1.2 Краткое описание проводимого занятия:

#### Вариант 1

- a)  $666_{(10)}$ ; б)  $305_{(10)}$ ; в)  $153,25_{(10)}$ ; г)  $162,25_{(10)}$ ; д)  $248,46_{(10)}$
- a)  $1100111011_{(2)}$ ; б)  $10000000111_{(2)}$ ; в)  $10110101,1_{(2)}$ ; г)  $100000110,10101_{(2)}$ ; д)  $671,24_{(8)}$ ; е)  $41A,6_{(16)}$ .
- a)  $10000011_{(2)} + 1000011_{(2)}$ ; б)  $1010010000_{(2)} + 110111011_{(2)}$ ; в)  $110010,101_{(2)} + 1011010011,01_{(2)}$ ; г)  $356,5_{(8)} + 1757,04_{(8)}$ ; д)  $293,8_{(16)} + 3CC,98_{(16)}$ .
- a)  $100111001_{(2)} - 110110_{(2)}$ ; б)  $1111001110_{(2)} - 111011010_{(2)}$ ; в)  $1101111011,01_{(2)} - 101000010,0111_{(2)}$ ; г)  $2025,2_{(8)} - 131,2_{(8)}$ ; д)  $2D8,4_{(16)} - A3,B_{(16)}$ .
- a)  $1010010_{(2)}$ ; б)  $1011100_{(2)}$ ; в)  $1102,2_{(8)}$ ; г)  $135,2_{(8)}$ ; д)  $7,4_{(16)}$ ; е)  $1A,9_{(16)}$ .

#### Вариант 2

- a)  $164_{(10)}$ ; б)  $255_{(10)}$ ; в)  $712,25_{(10)}$ ; г)  $670,25_{(10)}$ ; д)  $11,89_{(10)}$
- a)  $1001110011_{(2)}$ ; б)  $1001000_{(2)}$ ; в)  $1111100111,01_{(2)}$ ; г)  $1010001100,101101_{(2)}$ ; д)  $413,41_{(8)}$ ; е)  $118,8C_{(16)}$ .
- a)  $1100001100_{(2)} + 1100011001_{(2)}$ ; б)  $110010001_{(2)} + 1001101_{(2)}$ ; в)  $11111111,001_{(2)} + 111111110,0101_{(2)}$ ; г)  $1443,1_{(8)} + 242,44_{(8)}$ ; д)  $2B4,C_{(16)} + EA,4_{(16)}$ .
- a)  $1001101100_{(2)} - 1000010111_{(2)}$ ; б)  $1010001000_{(2)} - 1000110001_{(2)}$ ; в)  $1101100110,01_{(2)} - 111000010,1011_{(2)}$ ; г)  $1567,3_{(8)} - 1125,5_{(8)}$ ; д)  $416,3_{(16)} - 255,3_{(16)}$ .
- a)  $1000000_{(2)}$ ; б)  $1001010_{(2)}$ ; в)  $1204,2_{(8)}$ ; г)  $106,6_{(8)}$ ; д)  $61,4_{(16)}$ ; е)  $28,7_{(16)}$ .

#### Вариант 3

- a)  $273_{(10)}$ ; б)  $661_{(10)}$ ; в)  $156,25_{(10)}$ ; г)  $797,5_{(10)}$ ; д)  $53,74_{(10)}$
- a)  $1100000000_{(2)}$ ; б)  $1101011111_{(2)}$ ; в)  $1011001101,00011_{(2)}$ ; г)  $1011110100,011_{(2)}$ ; д)  $1017,2_{(8)}$ ; е)  $111,B_{(16)}$ .
- a)  $1110001000_{(2)} + 110100100_{(2)}$ ; б)  $1001001101_{(2)} + 1111000_{(2)}$ ; в)  $111100010,0101_{(2)} + 1111111,01_{(2)}$ ; г)  $573,04_{(8)} + 1577,2_{(8)}$ ; д)  $108,8_{(16)} + 21B,9_{(16)}$ .
- a)  $1010111001_{(2)} - 1010001011_{(2)}$ ; б)  $1110101011_{(2)} - 100111000_{(2)}$ ; в)  $1110111000,011_{(2)} - 111001101,001_{(2)}$ ; г)  $1300,3_{(8)} - 464,2_{(8)}$ ; д)  $37C,4_{(16)} - 1D0,2_{(16)}$ .
- a)  $1100010_{(2)}$ ; б)  $101001_{(2)}$ ; в)  $600,3_{(8)}$ ; г)  $40,2_{(16)}$ ; д)  $1A,88_{(16)}$ .

#### Вариант 4

- a)  $105_{(10)}$ ; б)  $358_{(10)}$ ; в)  $377,5_{(10)}$ ; г)  $247,25_{(10)}$ ; д)  $87,27_{(10)}$

2. а)  $1100001001_{(2)}$ ; б)  $1100100101_{(2)}$ ; в)  $111110110,01_{(2)}$ ; г)  $11001100,011_{(2)}$ ; д)  $112,04_{(8)}$ ; е)  $334,A_{(16)}$ .  
 3. а)  $101000011_{(2)} + 110101010_{(2)}$ ; б)  $111010010_{(2)} + 101101110_{(2)}$ ; в)  $10011011,011_{(2)} + 1111100001,0011_{(2)}$ ; г)  $1364,44_{(8)} + 1040,2_{(8)}$ ; д)  $158,A_{(16)} + 34,C_{(16)}$ .  
 4. а)  $1111111000_{(2)} - 100010011_{(2)}$ ; б)  $111101110_{(2)} - 1100110_{(2)}$ ; в)  $1001100100,01_{(2)} - 10101001,1_{(2)}$ ; г)  $1405,3_{(8)} - 346,5_{(8)}$ ; д)  $3DD,4_{(16)} - 303,A_{(16)}$ .  
 5. а)  $1011000_{(2)} \bullet 101010_{(2)}$ ; б)  $1772,34_{(8)} \bullet 44,3_{(8)}$ ; в)  $67,B_{(16)} \bullet 16,58_{(16)}$ .

### Вариант 5

1. а)  $500_{(10)}$ ; б)  $675_{(10)}$ ; в)  $810,25_{(10)}$ ; г)  $1017,25_{(10)}$ ; д)  $123,72_{(10)}$   
 2. а)  $1101010001_{(2)}$ ; б)  $100011100_{(2)}$ ; в)  $1101110001,011011_{(2)}$ ; г)  $110011000,111001_{(2)}$ ; д)  $1347,17_{(8)}$ ; е)  $155,6C_{(16)}$ .  
 3. а)  $1000101101_{(2)} + 1100000010_{(2)}$ ; б)  $111011010_{(2)} + 111001100_{(2)}$ ; в)  $100100011,1_{(2)} + 10001101,101_{(2)}$ ; г)  $415,24_{(8)} + 1345,04_{(8)}$ ; д)  $113,B_{(16)} + 65,8_{(16)}$ .  
 4. а)  $1101111100_{(2)} - 100100010_{(2)}$ ; б)  $1011010110_{(2)} - 1011001110_{(2)}$ ; в)  $111011110,1101_{(2)} - 1001110111,1_{(2)}$ ; г)  $1333,2_{(8)} - 643,2_{(8)}$ ; д)  $176,7_{(16)} - E5,4_{(16)}$ .  
 5. а)  $1010000_{(2)} \bullet 1001101_{(2)}$ ; б)  $1467,72_{(8)} \bullet 31,56_{(8)}$ ; в)  $11,6_{(16)} \bullet 11,3_{(16)}$ .

### Вариант 6

1. а)  $218_{(10)}$ ; б)  $808_{(10)}$ ; в)  $176,25_{(10)}$ ; г)  $284,25_{(10)}$ ; д)  $253,04_{(10)}$   
 2. а)  $111000100_{(2)}$ ; б)  $1011001101_{(2)}$ ; в)  $10110011,01_{(2)}$ ; г)  $101011111,011_{(2)}$ ; д)  $1665,3_{(8)}$ ; е)  $FA,7_{(16)}$ .  
 3. а)  $11100000_{(2)} + 1100000000_{(2)}$ ; б)  $110101101_{(2)} + 111111110_{(2)}$ ; в)  $10011011,011_{(2)} + 110110100,01_{(2)}$ ; г)  $1041,2_{(8)} + 1141,1_{(8)}$ ; д)  $3C6,8_{(16)} + B7,5_{(16)}$ .  
 4. а)  $10110010_{(2)} - 1010001_{(2)}$ ; б)  $1101000000_{(2)} - 10000000_{(2)}$ ; в)  $1100101111,1101_{(2)} - 100111000,1_{(2)}$ ; г)  $1621,44_{(8)} - 1064,5_{(8)}$ ; д)  $1AC,B_{(16)} - BD,7_{(16)}$ .  
 5. а)  $10001_{(2)} \bullet 10110_{(2)}$ ; б)  $616,34_{(8)} \bullet 73,2_{(8)}$ ; в)  $64,98_{(16)} \bullet 2A,6_{(16)}$ .

### Вариант 7

1. а)  $306_{(10)}$ ; б)  $467_{(10)}$ ; в)  $218,5_{(10)}$ ; г)  $667,25_{(10)}$ ; д)  $318,87_{(10)}$   
 2. а)  $1111000111_{(2)}$ ; б)  $11010101_{(2)}$ ; в)  $1001111010,010001_{(2)}$ ; г)  $1000001111,01_{(2)}$ ; д)  $465,3_{(8)}$ ; е)  $252,38_{(16)}$ .  
 3. а)  $1000001101_{(2)} + 1100101000_{(2)}$ ; б)  $101001110_{(2)} + 10001000_{(2)}$ ; в)  $1100111,00101_{(2)} + 101010110,011_{(2)}$ ; г)  $520,4_{(8)} + 635,4_{(8)}$ ; д)  $2DB,6_{(16)} + 15E,6_{(16)}$ .  
 4. а)  $1101000101_{(2)} - 111111000_{(2)}$ ; б)  $11110101_{(2)} - 110100_{(2)}$ ; в)  $1011101011,001_{(2)} - 1011001000,01001_{(2)}$ ; г)  $1034,4_{(8)} - 457,44_{(8)}$ ; д)  $239,A_{(16)} - 9C,4_{(16)}$ .  
 5. а)  $1001101_{(2)} \bullet 110001_{(2)}$ ; б)  $1121,4_{(8)} \bullet 110,3_{(8)}$ ; в)  $54,8_{(16)} \bullet 40,18_{(16)}$ .

### Вариант 8

1. а)  $167_{(10)}$ ; б)  $113_{(10)}$ ; в)  $607,5_{(10)}$ ; г)  $828,25_{(10)}$ ; д)  $314,71_{(10)}$   
 2. а)  $110010001_{(2)}$ ; б)  $100100000_{(2)}$ ; в)  $1110011100,111_{(2)}$ ; г)  $1010111010,1110111_{(2)}$ ; д)  $704,6_{(8)}$ ; е)  $367,38_{(16)}$ .  
 3. а)  $10101100_{(2)} + 111110010_{(2)}$ ; б)  $1000000010_{(2)} + 110100101_{(2)}$ ; в)  $1110111010,10011_{(2)} + 1011010011,001_{(2)}$ ; г)  $355,2_{(8)} + 562,04_{(8)}$ ; д)  $1E5,18_{(16)} + 3BA,78_{(16)}$ .  
 4. а)  $1010110010_{(2)} - 1000000000_{(2)}$ ; б)  $1111100110_{(2)} - 101011111_{(2)}$ ; в)  $1101001010,101_{(2)} - 110011000,011_{(2)}$ ; г)  $1134,54_{(8)} - 231,2_{(8)}$ ; д)  $2DE,6_{(16)} - 12A,4_{(16)}$ .  
 5. а)  $110001_{(2)} \bullet 110001_{(2)}$ ; б)  $1017,1_{(8)} \bullet 6,2_{(8)}$ ; в)  $1F,A_{(16)} \bullet 55,4_{(16)}$ .

### Вариант 9

1. а) 342<sub>(10)</sub>; б) 374<sub>(10)</sub>; в) 164,25<sub>(10)</sub>; г) 520,375<sub>(10)</sub>; д) 97,14<sub>(10)</sub>.
2. а) 1000110110<sub>(2)</sub>; б) 111100001<sub>(2)</sub>; в) 1110010100,1011001<sub>(2)</sub>; г) 1000000110,00101<sub>(2)</sub>; д) 666,16<sub>(8)</sub>; е) 1C7,68<sub>(16)</sub>.
3. а) 1101010000<sub>(2)</sub>+1011101001<sub>(2)</sub>; б) 100000101<sub>(2)</sub>+1100001010<sub>(2)</sub>; в) 1100100001,01001<sub>(2)</sub>+1110111111,011<sub>(2)</sub>; г) 242,2<sub>(8)</sub>+1153,5<sub>(8)</sub>; д) 84,8<sub>(16)</sub>+27E,8<sub>(16)</sub>.
4. а) 1111110<sub>(2)</sub>-1111011<sub>(2)</sub>; б) 1111100000<sub>(2)</sub>-111110011<sub>(2)</sub>; в) 1111011111,1001<sub>(2)</sub>-1010111100,01<sub>(2)</sub>; г) 1241,34<sub>(8)</sub>-1124,3<sub>(8)</sub>; д) 15F,A<sub>(16)</sub>-159,4<sub>(16)</sub>.
5. а) 101000<sub>(2)</sub>• 1110001<sub>(2)</sub>; б) 712,3<sub>(8)</sub>• 64,2<sub>(8)</sub>; в) 3D,8<sub>(16)</sub>• 37,4<sub>(16)</sub>.

### Вариант 10

1. а) 524<sub>(10)</sub>; б) 222<sub>(10)</sub>; в) 579,5<sub>(10)</sub>; г) 847,625<sub>(10)</sub>; д) 53,35<sub>(10)</sub>.
2. а) 10111111<sub>(2)</sub>; б) 111100110<sub>(2)</sub>; в) 10011000,1101011<sub>(2)</sub>; г) 1110001101,1001<sub>(2)</sub>; д) 140,22<sub>(8)</sub>; е) 1DE,54<sub>(16)</sub>.
3. а) 1101010000<sub>(2)</sub>+11100100<sub>(2)</sub>; б) 100110111<sub>(2)</sub>+101001000<sub>(2)</sub>; в) 1111100100,11<sub>(2)</sub>+1111101000,01<sub>(2)</sub>; г) 1476,3<sub>(8)</sub>+1011,1<sub>(8)</sub>; д) 3E0,A<sub>(16)</sub>+135,8<sub>(16)</sub>.
4. а) 1010010100<sub>(2)</sub>-11101110<sub>(2)</sub>; б) 10000001110<sub>(2)</sub>-10011100<sub>(2)</sub>; в) 1110100111,01<sub>(2)</sub>-110000001,1<sub>(2)</sub>; г) 1542,5<sub>(8)</sub>-353,24<sub>(8)</sub>; д) 3EB,8<sub>(16)</sub>-3BA,8<sub>(16)</sub>.
5. а) 1001111<sub>(2)</sub>• 1000100<sub>(2)</sub>; б) 1017,3<sub>(8)</sub>• 73,44<sub>(8)</sub>; в) 56,2<sub>(16)</sub>• 4A,4<sub>(16)</sub>.

### 3.2 Практическое занятие №2 (2 часа).

Тема: «Основные понятия и программная реализация алгоритмического языка Pascal»

#### 3.1.1 Вопросы к занятию:

1. Перевести данное число из десятичной системы счисления в двоичную, восьмеричную и шестнадцатеричную системы счисления.
2. Перевести данное число в десятичную систему счисления.
3. Сложить числа.
4. Выполнить вычитание.
5. Выполнить умножение.

#### 3.1.2 Краткое описание проводимого занятия:

Вариант 1	Вариант 2	Вариант 3	Вариант 4
<p>1. По заданному целому неотрицательному значению <math>n</math> вычислить <math>n! = 1*2*3*...*n</math>. Учтите, что по определению <math>0! = 1</math>.</p> <p>2. Написать программу, которая выводит таблицу квадратов первых десяти целых положительных чисел.</p>	<p>1. Составить таблицу значений функции <math>y = -2,4x^2 + 5x - 3</math> в диапазоне от <math>-2</math> до <math>2</math> с шагом <math>0,5</math>.</p> <p>2. Начав тренировки, спортсмен в первый день пробежал <math>10</math> км. Каждый день он увеличивал дневную норму на <math>10\%</math> нормы предыдущего дня. Какой суммарный путь пробежит спортсмен за <math>7</math> дней?</p>	<p>1. Вычислить сумму натурального ряда чисел от <math>1</math> до <math>N</math>.</p> <p>2. Одноклеточная амёба каждые <math>3</math> часа делится на <math>2</math> клетки. Определить, сколько будет амёб через <math>3,6,9,12,...24</math> часа.</p>	<p>1. Вычислить сумму <math>n</math> слагаемых вида <math>x+i</math>, где <math>x</math> – данное число, а <math>i</math> меняется от <math>1</math> до <math>n</math>.</p> <p>2. Распечатать числа Фибоначчи от <math>1</math> до <math>n</math>. (Каждое число в последовательности, начиная с третьего, получается сложением двух предыдущих чисел: <math>1,1,2,3,5,8,13...</math>)</p>

<b>Вариант 5</b>	<b>Вариант 6</b>	<b>Вариант 7</b>	<b>Вариант 8</b>
<p>1. Подсчитать сумму всех нечетных чисел от 101 до 301.</p> <p>2. Даны действительное число <math>a</math>, натуральное число <math>n</math>. Вычислить:</p> $P=a*(a-n)*(a-2n)*\dots*(a-n^2).$	<p>1. Определить, сколько из <math>n</math> заданных точек принадлежат графику функции <math>y= x </math>.</p> <p>2. Дано натуральное <math>n</math>. Вычислить <math>y=1*3*5*\dots*(2n-1)</math>.</p>	<p>1. Составьте таблицу значений функции <math>y=5x^2-2x+1</math> на отрезке <math>[-5;5]</math> с шагом 2.</p> <p>2. Дано натуральное <math>n</math>. Вычислить <math>S=2/1 + 3/2 + 4/3 + \dots + (n+1)/n</math>.</p>	<p>1. Напечатать таблицу значений функции <math>y=x^3</math> на отрезке <math>[-1;1]</math> с шагом 0,1.</p> <p>2. Напечатать таблицу перевода 1,2,..., 20 долларов США в рубли по текущему курсу (значение курса вводится с клавиатуры).</p>

**4. МЕТОДИЧЕСКИЕ УКАЗАНИЯ  
ПО ПРОВЕДЕНИЮ СЕМИНАРСКИХ ЗАНЯТИЙ**  
*Не предусмотрено РУП*