

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»**

Факультет среднего профессионального образования

КОНСПЕКТ ЛЕКЦИЙ ПО ДИСЦИПЛИНЕ

ПУП.02 Информатика

Специальность 21.02.05 Земельно-имущественные отношения

Форма обучения очная

СОДЕРЖАНИЕ

<i>Введение</i>	3
<i>Раздел 1. Информация и информационные процессы</i>	3
Тема 1.1. Представление информации в компьютере.....	3
Тема 1.2. Способы измерения информации	7
Тема 1.3. Алгоритмы. Алгоритмические языки. Языки программирования.	9
Тема 1.4. Логические, физические и арифметические основы вычислительной техники	14
Логические законы и правила преобразования логических выражений	
<i>Раздел 2. Информационная деятельность человека</i>	<u>19</u>
Тема 2.1. Информационная деятельность человека.	19
<i>Раздел 3. Средства информационных и коммуникационных технологий</i>	20
Тема 3.1.Архитектура компьютеров	20
<i>Раздел 4. Технологии создания и преобразования информационных объектов</i>	26
Тема 4.1. Текстовый редактор (текстовый процессор MS Word)	26
Тема 4.2. Электронные таблицы (Табличный процессор MS Excel)	28
Тема 4.3. Информационные системы. Базы данных MS Access	30
Тема 4.4. Мастер презентаций MS Power Point	32
<i>Раздел 5 Телекоммуникационные технологии</i>	34
Тема 5.1. Компьютерные сети. Средства телекоммуникационных технологий.	34
Тема 5.2. Язык HTML. Основы WEB-дизайна	38

Введение

1. Роль дисциплины «Информатика и ИКТ» в подготовке специалиста;
2. Связь с другими дисциплинами;
3. Роль информационной деятельности в современном обществе: экономической, социальной, культурной, образовательной сферах;

Информатика – самостоятельная научная дисциплина, предметом которой стали свойства информации, её поведение в техногенных, социальных и биологических системах, а также методы и технологии, ориентированные на сбор, обработку, хранение, передачу и распространение информации, или, кратко, информационные технологии.

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

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.

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.

Информатика исследует следующие группы основных вопросов:

- технические, связанные с изучением методов и средств надежного сбора, хранения, передачи, обработки и выдачи информации;
- семантические, определяющие способы описания смысла информации, изучающие языки ее описания;
- прагматические, описывающие методы кодирования информации;
- синтаксические, связанные с решением задач по формализации и автоматизации некоторых видов научно-информационной деятельности, в частности индексирование, автоматическое реферирование, машинный перевод.

Раздел 1. Информация и информационные процессы

Тема 1.1. Представление информации в компьютере.

1. Основные подходы к определению понятия «информация».
2. Системы, образованные взаимодействующими элементами, состояния элементов, обмен информацией между элементами, сигналы, сообщения, данные. Дискретные и непрерывные сигналы. Носители информации.
3. Системы счисления: позиционные и непозиционные.
4. Преобразование чисел в системах счисления
5. Двоичная арифметика

Термин "информация" происходит от латинского слова "*informatio*", что означает **сведения, разъяснения, изложение**. Несмотря на широкое распространение этого термина, понятие информации является одним из самых дискуссионных в науке. Понятие *информация* получает различные смысловые наполнения в различных отраслях человеческой деятельности:

- **в обиходе** информацией называют любые данные или сведения, которые кого-либо интересуют. Например, сообщение о каких-либо событиях, о чьей-либо деятельности и т.п. "*Информировать*" в этом смысле означает "*сообщить нечто, неизвестное раньше*";
- **в технике** под *информацией* понимают сообщения, передаваемые в форме знаков или сигналов;
- **в кибернетике** под информацией понимает ту часть знаний, которая используется для ориентирования, активного действия, управления, т.е. в целях сохранения, совершенствования, развития системы (Н. Винер).

Клод Шеннон, американский учёный, заложивший основы теории информации — науки, изучающей процессы, связанные с передачей, приёмом, преобразованием и хранением информации, — **рассматривает информацию как снятую неопределенность наших знаний о чем-то**.

Современное научное представление об информации сформулировал **Норберт Винер**, "отец" кибернетики.

Информация — это обозначение содержания, полученного из внешнего мира в процессе нашего приспособления к нему и приспособления к нему наших чувств.

Люди обмениваются информацией в форме сообщений. Сообщение - это форма представления информации в виде речи, текстов, жестов, взглядов, изображений, цифровых данных, графиков, таблиц и т.п.

Одно и то же информационное сообщение (статья в газете, объявление, письмо, телеграмма, справка, рассказ, чертёж, радиопередача и т.п.) *может содержать разное количество информации для разных людей — в зависимости от их предшествующих знаний, от уровня понимания этого сообщения и интереса к нему.*

В случаях, когда говорят **об автоматизированной работе с информацией посредством каких-либо технических устройств**, обычно в первую очередь интересуются не содержанием сообщения, а тем, сколько символов это сообщение содержит.

Применительно к компьютерной обработке данных под информацией понимают некоторую последовательность символьических обозначений (букв, цифр, закодированных графических образов и звуков и т.п.), несущую смысловую нагрузку и представленную в понятном компьютеру виде. Каждый новый символ в такой последовательности символов **увеличивает информационный объём сообщения**.

Информация может существовать в виде:

- текстов, рисунков, чертежей, фотографий;
- световых или звуковых сигналов;
- радиоволн;
- электрических и нервных импульсов;
- магнитных записей;
- жестов и мимики;
- запахов и вкусовых ощущений;
- хромосом;

Сообщения передаются от некоторого **источника** информации к её **приёмнику** посредством **канала связи** между ними. Источник посыпает **передаваемое сообщение**, которое **кодируется в передаваемый сигнал**. Этот сигнал посыпается по **каналу связи**. В результате в приёмнике появляется **принимаемый сигнал**, который **декодируется** и становится **принимаемым сообщением**.

Передача информации по каналам связи часто сопровождается воздействием **помех**, вызывающих **искажение и потерю информации**.

- достоверность;
- полнота;
- ценность;
- понятность;
- доступность;
- краткость;

- своевременность;
- и др.

Информация достоверна, если она отражает истинное положение дел. Недостоверная информация может привести к неправильному пониманию или принятию неправильных решений.

Достоверная информация со временем может стать недостоверной, так как она обладает свойством устаревать, то есть перестаёт отражать истинное положение дел.

Информация полна, если её достаточно для понимания и принятия решений. Как неполная, так и избыточная информация сдерживает принятие решений или может повлечь ошибки.

Точность информации определяется степенью ее близости к реальному состоянию объекта, процесса, явления и т.п.

Ценность информации зависит от того, насколько она важна для решения задачи, а также от того, насколько в дальнейшем она найдёт применение в каких-либо видах деятельности человека.

Только своевременно полученная информация может принести ожидаемую пользу. Однаково нежелательны как преждевременная подача информации (когда она ещё не может быть усвоена), так и её задержка.

Если ценная и своевременная информация выражена непонятным образом, она может стать бесполезной.

Информация становится понятной, если она выражена языком, на котором говорят те, кому предназначена эта информация.

Машинные единицы измерения информации.

1 бит информации - в вычислительной технике наименьшая "порция" памяти, которая необходима для хранения одного из двух знаков "0" и "1" (используемых для внутримашинного представления данных и команд).

Термин "бит" предложен как аббревиатура от английского словосочетания "Binary digit", которое переводится как "двоичная цифра".

В компьютерной технике бит соответствует физическому состоянию носителя информации: намагничено - не намагничено, есть отверстие - нет отверстия. При этом одно состояние принято обозначать цифрой 0, а другое - цифрой 1. Выбор одного из двух возможных вариантов позволяет также различать логические истину и ложь. Последовательностью битов можно закодировать текст, изображение, звук или какую-либо другую информацию. Такой метод представления информации называется двоичным кодированием (binary encoding).

В информатике часто используется величина, называемая байтом (byte) и равная 8 битам. И если бит позволяет выбрать один вариант из двух возможных, то байт, соответственно, 1 из 256 (2^8).

В большинстве современных ЭВМ при кодировании каждому символу соответствует своя последовательность из восьми нулей и единиц, т. е. байт. Соответствие байтов и символов задается с помощью таблицы, в которой для каждого кода указывается свой символ.

Наряду с байтами для измерения количества информации используются более крупные единицы:

1 Кбайт (один килобайт) = 2^{10} байт = 1024 байта;

1 Мбайт (один мегабайт) = 2^{10} Кбайт = 1024 Кбайта = 2^{20} байт;

1 Гбайт (один гигабайт) = 2^{10} Мбайт = 1024 Мбайта = 2^{30} байт.

В последнее время в связи с увеличением объёмов обрабатываемой информации входят в употребление такие производные единицы, как:

1 Тбайт (один терабайт) = 2^{10} Гбайт = 1024 Гбайт = 2^{40} байт,

1 Пбайт(один петабайт) = 2^{10} Тбайт = 1024 Тбайт = 2^{50} байт.

Определение. Система счисления – это совокупность правил для обозначения и наименования чисел.

Все рассмотренные нами нечисловые системы счисления являются непозиционными.

Определение. Непозиционной называется такая система счисления, в которой количественный эквивалент каждой цифры не зависит от ее положения (места, позиции) в записи числа.

Итак, в непозиционных системах счисления позиция, которую цифра занимает в записи числа, роли не играет. Так, например, римская система счисления непозиционная. В числах XI и IX "вес" обоих цифр одинаков, несмотря на их месторасположение.

Система счисления, которой мы пользуемся в настоящее время, носит название **десятичной**, так как она основана на счете десятками. Исключительная роль десятка восходит к древнейшим

временам и, несомненно, связана со счетом по пальцам на двух руках. Для записи любых чисел в ней используется десять всем хорошо известных цифр (0,1,2,3,4,5,6,7,8,9). Поэтому ее и называют десятичной.

Определение. Основанием системы счисления называется количество знаков или символов, используемых для изображения числа в данной системе счисления.

Наименование системы счисления соответствует ее основанию (например, десятичной называется система счисления так потому, что ее основание равно 10, т.е. используется десять цифр).

Определение. Система счисления называется позиционной, если значение цифры зависит от ее места (позиции) в записи числа.

Системы счисления, используемые в компьютерах.

Двоичная система счисления. Для записи чисел используются только две цифры – 0 и 1. Выбор двоичной системы объясняется тем, что электронные элементы, из которых строятся ЭВМ, могут находиться только в двух хорошо различимых состояниях. По существу эти элементы представляют собой выключатели. Как известно выключатель либо включен, либо выключен. Третьего не дано. Одно из состояний обозначается цифрой 1, другое – 0.

Благодаря таким особенностям двоичная система стала стандартом при построении ЭВМ.

Восьмеричная система счисления. Для записи чисел используется восемь чисел 0,1,2,3,4,5,6,7.

Шестнадцатеричная система счисления. Для записи чисел в шестнадцатеричной системе необходимо располагать уже шестнадцатью символами, используемыми как цифры. В качестве первых десяти используются те же, что и в десятичной системе. Для обозначения остальных шести цифр (в десятичной они соответствуют числам 10,11,12,13,14,15) используются буквы латинского алфавита – A,B,C,D,E,F.

Перевод целых чисел из десятичной системы счисления в другую.

Правило перевода целых чисел из десятичной системы счисления в систему с основанием q :

- Последовательно выполнять деление исходного числа и получаемых частных на q до тех пор, пока не получим частное, меньшее делителя.
 - Полученные при таком делении остатки – цифры числа в системе счисления q – записать в обратном порядке (снизу вверх).

Пример1. Перевести 26_{10} в двоичную систему счисления. $A_{10} \rightarrow A_2$

Решение:

$$\begin{array}{r}
 - 26 \Big| 2 \\
 26 \Big| -13 \Big| 2 \\
 \hline
 0 \quad 12 \quad 6 \quad 3 \quad 2 \\
 \hline
 1 \quad 6 \quad 0 \quad 2 \quad 1 \\
 \hline
 \end{array}$$

Ответ: $26_{10} \equiv 110102$

Пример2. Перевести 19_{10} в троичную систему счисления. $A_{10} \rightarrow A_3$

Решение:

$$\begin{array}{r} - 19 \\ \hline 18 \\ \hline 1 \end{array}$$

Ответ: $19_{10} \equiv 201_3$

Пример4. Перевести 3627_{10} в шестнадцатеричную систему счисления. $A_{10} \rightarrow A_{16}$

Решение:

$$\begin{array}{r}
 -3627 \quad | \quad 16 \\
 -3616 \quad | \quad 226 \quad | \quad 16 \\
 \hline
 \textbf{11} \quad | \quad \textbf{224} \quad | \quad \textbf{14} \\
 \end{array}$$

Т.к. в шестнадцатеричной системе счисления $14 = E$, а $11 = B$, то получаем ответ $E2B_{16}$.

Ответ: $3627_{10}=E2B_{16}$

Перевод произвольных чисел из десятичной системы счисления в другую.

Перевод произвольных чисел, то есть чисел, содержащих целую и дробную части, осуществляют в два этапа. Отдельно переводится целая часть, отдельно – дробная. В итоговой записи полученного числа целая часть отделяется от дробной запятой.

Пример1. Перевести $26,25_{10}$ в двоичную систему счисления. $A_{10} \rightarrow A_2$

Решение:

переводим целую часть

$$\begin{array}{r} - 26 \\ \hline 26 \\ \hline 0 \\ \hline 13 \\ \hline 12 \quad | \quad 2 \\ \hline 1 \quad | \quad 6 \\ \hline 0 \quad | \quad 3 \\ \hline 2 \quad | \quad 1 \\ \hline \end{array}$$

переводим дробную часть

$$\begin{array}{r} 0, \quad 25 \\ \hline * \quad 2 \\ \hline 0 \quad | \quad 50 \\ \hline * \quad 2 \\ \hline 1 \quad | \quad 00000 \end{array}$$

Ответ: $26,25_{10}=11010,01_2$

Пример2. Перевести $123,5625_{10}$ в двоичную систему счисления. $A_{10} \rightarrow A_8$

Решение:

переводим целую часть

$$\begin{array}{r} - 123 \\ \hline 120 \quad | \quad 8 \\ \hline 3 \quad | \quad 15 \quad | \quad 8 \\ \hline 8 \quad | \quad 7 \quad | \quad 1 \\ \hline \end{array}$$

переводим дробную часть

$$\begin{array}{r} 0, \quad 5625 \\ \hline * \quad 8 \\ \hline 4 \quad | \quad 5000 \\ \hline * \quad 8 \\ \hline 4 \quad | \quad 00000 \end{array}$$

Ответ: $123,5625_{10}=173,44_8$

Современные ЭВМ обрабатывают не только числовую, но и текстовую, или алфавитноцифровую, информацию, содержащую буквы, цифры, знаки препинания, математические и другие символы. Именно такой характер имеет экономическая, плановая и учетная информация, а также тексты программ на алгоритмических языках.

Представление текстовой информации в ЭВМ

Для представления алфавитно-цифровой информации используется один байт. С помощью байта можно закодировать $2^8=256$ символов. Большое распространение в России для представления символьной информации в ЭВМ получили кодировочные таблицы: KOI8-R (Операционная система (ОС) - Unix), CP1251 (ОС - Ms Windows), ISO 8859-5 (Интернет), 866 кодовая страница (ОС – Ms DOS).

Символьная информация представляет собой последовательность знаков, представленных в различных кодировках.

Контрольная работа №1

Перевести числа из одной системы счисления в другую согласно своего варианта:

1 вариант	2 вариант
$()_{10}$	$()_2$
243_{10}	$()_{8,16}$
$()_{8,16}$	$()_2$
130_8	159_{10}
1423_{16}	$()_{8,16}$
	12043_8
	58_{16}

Тема 1.2. Способы измерения информации

1. Количество информации как мера уменьшения неопределенности знаний.
2. Алфавитный подход к определению количества информации

3. Вероятностный подход к определению количества информации

Вопрос «Как измерять информацию?» очень непростой. Ответ на него зависит от того, что понимать под информацией.

Существует несколько подходов к определению информации в зависимости от области знаний. Рассмотрим лишь три.

Определение 1. В **быту**, под информацией понимают сведения об окружающем мире и протекающих в нем процессах, воспринимаемые человеком или специальными устройствами.

Определение 2. В **теории информации**, под информацией понимают не любые сведения об объектах и явлениях окружающей среды, а лишь те, которые снимают полностью или уменьшают имеющуюся о них степень неопределенности, неполноты знаний.

Определение 3. В **технике**, под информацией понимают сообщения, передаваемые в форме знаков или сигналов.

Для измерения информации используется понятие «Энтропия». Энтропия — это общая мера неопределенности, которая уменьшается при получении информации.

Для измерения количества информации имеются единицы измерения, что позволяет некоторой порции информации приписывать числа — количественные характеристики.

Для измерения информации была предложена Клодом Шенноном в 1948 году минимальная единица измерения **бит**, которая равнялась 0 либо 1.

На сегодняшний день наиболее известны следующие способы измерения информации:

1. **ОБЪЕМНЫЙ** — является самым простым и грубым способом измерения информации. Объем информации в сообщении — это количество символов, содержащихся в сообщении, причем не зависит от системы счисления. Например, 21, 11001, XXI.
2. **ЭНТРОПИЙНЫЙ** — общая мера неопределенности (энтропия) характеризуется некоторой математической зависимостью от между количеством информации и количеством возможных событий. Количество информации в сообщении определяется тем, насколько уменьшается эта мера неопределенности после получения сообщения. Существует формула, которая связывает между собой количество возможных событий **N** и количество информации **I**:

По этой формуле можно легко определить количество возможных событий, если известно количество информации.

$$N = 2^I$$

3. **АЛФАВИТНЫЙ ПОДХОД** к определению количества информации заключается в том, что каждый символ сообщения несет некоторое количество битов информации. Количество информации в сообщении можно подсчитать, умножив количество информации, которое несет один символ, на количество символов в сообщении. Для определения используется следующая формула:

$$I_I = K * I, \text{ где}$$

I_I — информационный объем всего сообщения;

K — количество символов в сообщении;

I — информационный объем одного символа.

Есть алфавит, который можно назвать достаточным. Это алфавит мощностью 256 символов. Алфавит из 256 символов используется для представления текстов в компьютере. В этом алфавите можно поместить практически все необходимые символы: латинские и русские буквы, цифры, знаки арифметических операций, скобки, знаки препинания, знаки псевдографики. Поскольку $256 = 2^8$, то один символ этого алфавита «весит» 8 бит.

8 бит информации присвоили свое название — **байт**.

Байт — поле из 8 последовательных бит. Байт широко используется как единица измерения количества информации.

$$1 \text{ байт} = 8 \text{ бит}$$

Компьютерные текстовые редакторы работают с алфавитом мощности 256 символов. Поскольку в настоящее время при подготовке книг используются текстовые редакторы, легко посчитать объем информации в тексте.

Если один символ алфавита несет 1 байт информации, то надо просто сосчитать число символов, полученное значение даст информационный объем текста в байтах.

В любой системе единиц измерения существуют основные единицы и производные от них.

Для измерения больших объемов информации используются производные от байта единицы:

1 килобайт = 1 Кб = 2^{10} байт = 1024 байта

1 мегабайт = 1 Мб = 2^{10} Кб = 1024 Кб = 1048576 байт

1 гигабайт = 1 Гб = 2^{10} Мб = 1024 Мб = 1048576 Кб = 1073741824 байт

4. ВЕРОЯТНОСТНЫЙ ПОДХОД

Следует различать два вида вероятностных подхода:

- Равновероятностный:
- Неравновероятностный.

Для частных, но широко распространенных случаев, когда события равновероятны, используются следующие формулы:

$$I = \log_2 N \Rightarrow N = 2^I, \text{ где}$$

I — количество информации о событии;

N — количество возможных событий

$$P_I = K/N \text{ где}$$

P_I — вероятность равномерного события;

K — количество равномерных событий;

Для неравномерных событий, когда события неравновероятны, используются следующие формулы:

$$I = \log_2 (1/P_I) \Rightarrow (1/P_I) = 2^I, \text{ где}$$

P_I — вероятность неравномерного события;

$$P_I = K/N \text{ где}$$

P_I — вероятность равномерного события;

K — величина, показывающая сколько раз произошло интересующее нас событие;

Тема 1.3. Алгоритмы. Алгоритмические языки. Языки программирования.

1. Понятие «алгоритм». Свойства алгоритмов. Виды алгоритмов
2. Программирование на языке Turbo Pascal.
3. Типы данных. Структура программы.
4. Программы линейной структуры.
5. Программы разветвляющейся структуры.
6. Программы циклической структуры

Алгоритм — это точное предписание, которое определяет процесс, ведущий от исходных данных к требуемому конечному результату. Слово алгоритм происходит от algoritmi, являющегося латинской транслитерацией арабского имени Хорезмийского математика IX века аль-Хорезми. Применительно к ЭВМ алгоритм определяет вычислительный процесс, начинающийся с обработки некоторой совокупности возможных исходных данных и направленный на получение определенных этими исходными данными результатов. Термин вычислительный процесс

распространяется и на обработку других видов информации, например, символьной, графической или звуковой.

Алгоритм должен быть всегда результативен, иметь свойство повторяемости и рассчитан на конкретного исполнителя. В технике таким исполнителем является ЭВМ. Для обеспечения возможности реализации на ЭВМ алгоритм должен быть описан на языке, понятном ЭВМ, то есть на машинном языке, созданным с помощью языка программирования.

Таким образом, **Программа для ЭВМ** представляет собой описание алгоритма и данных на некотором языке, понятном ЭВМ и полученным с помощью языка программирования. Программа при этом предназначена для последующего автоматического выполнения.

После решения задачи обработки информации результат должен быть выдан конечным пользователям в требуемом виде. Эта операция реализуется в ходе решения задачи выдачи информации. Выдача информации, как правило, производится с помощью внешних устройств ЭВМ в виде текстов, таблиц, графиков.

Существует несколько форм представления алгоритмов:

- На естественном языке (словесная форма).
- На языке блок-схем.
- На алгоритмическом языке – программа.

В повседневной жизни алгоритм часто записывается в виде предложений, расположенных в порядке выполнения. Запись алгоритма с помощью слов называется **словесным представлением алгоритма**.

Если вы внимательно оглянитесь вокруг, то обнаружите множество алгоритмов которые мы с вами постоянно выполняем. Мир алгоритмов очень разнообразен. Несмотря на это, удается выделить общие свойства, которыми обладает любой алгоритм.

Свойства алгоритмов:

1. **Дискретность** (алгоритм должен состоять из конкретных действий, следующих в определенном порядке);
2. **Детерминированность** (любое действие должно быть строго и недвусмысленно определено в каждом случае);
3. **Конечность** (каждое действие и алгоритм в целом должны иметь возможность завершения);
4. **Массовость** (один и тот же алгоритм можно использовать с разными исходными данными);
5. **Результативность** (отсутствие ошибок, алгоритм должен приводить к правильному результату для всех допустимых входных значениях).

В алгоритме команды записаны одна за другой в определенном порядке. Исполняются они не обязательно в том же порядке. В зависимости от того, каков порядок исполнения команд, можно выделить три типа алгоритмов: линейный, разветвляющий, циклический. и вспомогательные.

Виды алгоритмов:

1. **Линейный алгоритм** (описание действий, которые выполняются однократно в заданном порядке);
2. **Циклический алгоритм** (описание действий, которые должны повторяться указанное число раз или пока не выполнено задание);
3. **Разветвляющий алгоритм** (алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий).

Для более наглядного представления алгоритма широко используется графическая форма – блок-схема, которая составляется из стандартных графических объектов.. Каждое графически

обозначенное предложение алгоритма называется блоком. В блок записывается только одна команда. Блоки (шаги) алгоритма соединены стрелочками.

Вид стандартного графического объекта	Назначение
	Начало и конец алгоритма
	Ввод/вывод данных
	Выполняемое действие записывается внутри прямоугольника
	Условие выполнения действий записывается внутри ромба
	Счетчик кол-во повторов
	Последовательность выполнения действий

Примеры записи алгоритмов в виде блок-схемы:

Линейный алгоритм.

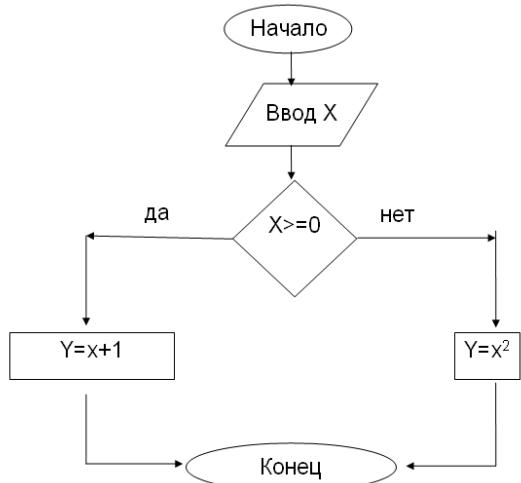
Вычислить площадь прямоугольника со сторонами А, В.



Разветвляющий алгоритм.

Вычислить:

$$Y = \begin{cases} X+1, & \text{если } X \geq 0 \\ X^2, & \text{если } X < 0 \end{cases}$$



Циклический алгоритм.

Стадии создания алгоритма:

1. Алгоритм должен быть представлен в форме, понятной человеку, который его разрабатывает.

2. Алгоритм должен быть представлен в форме, понятной тому объекту (в том числе и человеку), который будет выполнять описанные в алгоритме действия.

Объект, который будет выполнять алгоритм, обычно называют исполнителем.

Исполнитель – объект, который выполняет алгоритм.

Идеальными исполнителями являются машины, роботы, компьютеры...

Компьютер – автоматический исполнитель алгоритмов.

Алгоритм, записанный на “понятном” компьютеру языке программирования, называется **программой**.

В качестве примера использования ветвлений рассмотрим составление алгоритма для вычисления функции f в зависимости от конкретных значений x, a, b :

$$f = \begin{cases} x + a, & \text{при } x \leq 2.5; \\ x - a \cdot b, & \text{при } 2.5 < x < 10; \\ x - b, & \text{при } x \geq 10. \end{cases}$$

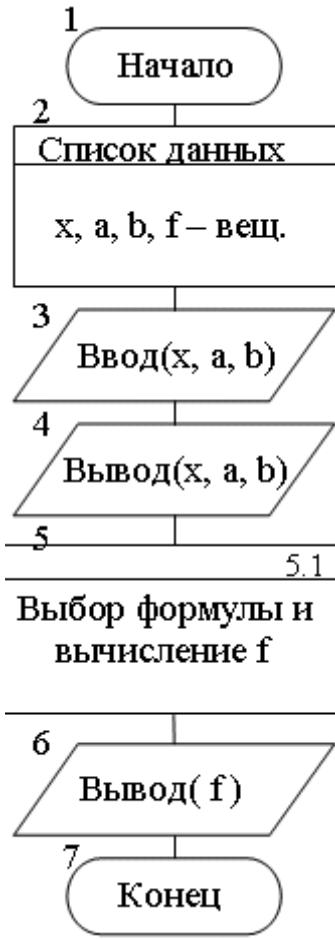


Рис. 10. Первый этап составления алгоритма.

Первое приближение алгоритма будет иметь вид, показанный на рис. 10. Анализ этого алгоритма показывает, что все его блоки, кроме блока 5, являются элементарными. Возможный вариант дальнейшей детализации блока 5 представлен на рис. 11. Все блоки второго этапа детализации являются элементарными, поэтому составление алгоритма практически закончено.

На заключительном этапе производится сборка алгоритма, т.е. содержимое рис. 11 помещается на рис. 10 вместо блока 5.

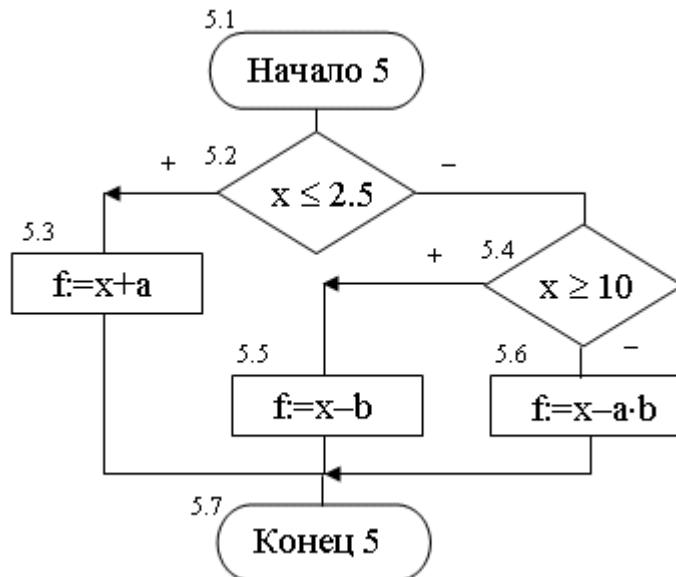


Рис. 11. Второй этап составления алгоритма.

Для представления алгоритма в виде, понятном компьютеру, служат языки программирования. Сначала разрабатывается алгоритм действий, а потом он записывается на одном из таких языков. В итоге получается текст программы - полное, законченное и детальное описание алгоритма на языке программирования. Затем этот текст программы специальными служебными приложениями, которые называются *трансляторами*, либо переводится в машинный код (язык нулей и единиц), либо исполняется.

Языки программирования - искусственные языки. От естественных они отличаются ограниченным числом "слов", значение которых понятно транслятору, и очень строгими правилами записи команд (*операторов*).

Для написания текста программы можно использовать обычный текстовый редактор (например, Блокнот), а затем с помощью компилятора перевести её в машинный код, т.е. получить исполняемую программу. Но проще и удобнее пользоваться специальными интегрированными средствами программирования.

Язык программирования Паскаль

Алгоритмический язык **Паскаль (Pascal)** разработан профессором Цюрихского технологического института **Никлаусом Виртом** в 1969-71 годах для обучения студентов структурному программированию. Язык получил свое название не от сокращения каких-то слов, как другие языки: он назван так в честь великого французского математика и физика **Блеза Паскаля**, который в 1642 г. изобрел счетную машину для арифметических операций – паскалево колесо, как называли её современники.

Идеи, заложенные в основу создания языка, позволили фирме Borland International значительно расширить алгоритмические средства языка, а удобный интерфейс (меню команд) и высокая скорость компиляции (процесса перевода текста программы в машинные коды) сделали

язык Турбо-Паскаль (Turbo-Pascal) одним из самых распространенных среди начинающих и профессиональных программистов.

Процесс программирования начинается, как правило с составления алгоритма - последовательности операций, описывающих процесс решения задачи. Графическая запись алгоритма представляется в виде блок-схемы. Составление алгоритма заключается в логическом описании процесса решения задачи и требует знания элементов математической логики. Программирование заключается в записи алгоритма на языке программирования и отладке программы.

Программа на языке Паскаль проходит три этапа обработки:

- создание текста программы;
- компиляция;
- исполнительная система.

Тема 1.4. Логические, физические и арифметические основы вычислительной техники

1. Алгебра логики. Формы мышления: понятие, высказывание (суждение), умозаключение, доказательство.
2. Алгебра высказываний: логический язык, имена, логические операции, кванторы, правила построения и семантика. Примеры записи утверждений на логическом языке.
3. Таблицы истинности. Основные законы (аксиомы) алгебра логики.

История логики насчитывает около двух с половиной тысячелетий. Первые учения о формах и способах мышления возникли в Древнем Китае и Индии. Основоположником формальной логики является Аристотель (384-322 гг. до н.э.) – древнегреческий философ, который впервые отделил логические формы мышления от его содержания. Алгебра логики – наука об операциях, аналогичных математическим, над высказываниями или над объектами, которые могут принимать только два значения – «ИСТИНА» или «ЛОЖЬ». В 1842 году английский математик Джорж Буль разработал математическую логику или алгебру логики, которую впоследствии стали называть «булевой алгеброй». Спустя 100 лет алгебра логики стала основой теории цифровых вычислительных машин, ее используют в компьютерной логике, электронике, в основе всех микропроцессорных операций. Многие философы и математики развивали отдельные положения логики и иногда даже намечали контуры современного исчисления высказываний, но ближе всех к созданию математической логики подошел уже во второй половине XVII века выдающийся немецкий ученый Готфрид Вильгельм Лейбниц (1646— 1716), указавший пути для перевода логики "из словесного царства, полного неопределенностей, в царство математики, где отношения между объектами или высказываниями определяются совершенно точно". Лейбниц надеялся даже, что в будущем философы, вместо того чтобы бесплодно спорить, станут брать бумагу и вычислять, кто из них прав. При этом в своих работах Лейбниц затрагивал и двоичную систему счисления. Уже в XIX веке стало понятно, что система Буля хорошо подходит для описания электрических переключательных схем. Ток в цепи может либо протекать, либо отсутствовать, подобно тому, как утверждение может быть либо истинным, либо ложным. А еще несколько десятилетий спустя, уже в XX столетии, ученые объединили созданный Джорджем Булем математический аппарат с двоичной системой счисления, заложив тем самым основы для разработки цифрового электронного компьютера.

Логика – это наука о формах и способах мышления, рассуждений и доказательств.

Мышление осуществляется через понятия, высказывания и умозаключения.

Понятие – это форма мышления, выделяющая существенные и отличительные признаки объекта.

Умозаключение – это форма мышления, с помощью которой из одного или нескольких простых высказываний (суждений) может быть получено новое составное высказывание (суждение).

Высказывание – это формулировка в форме утверждения или отрицания об объекте и его свойствах. Высказывание может быть истинным или ложным.

Алгебра высказываний

Алгебра в широком смысле этого слова — наука об общих операциях, аналогичных сложению и умножению, которые могут выполняться над различными математическими объектами (алгебра переменных и функций, алгебра векторов, алгебра множеств и т.д.). Объектами алгебры логики являются высказывания.

Алгебра логики отвлекается от смысловой содержательности высказываний. Ее интересует только один факт — истинно или должно данное высказывание, что дает возможность определять истинность или ложность составных высказываний алгебраическими методами.

Простые высказывания в алгебре логики обозначаются заглавными латинскими буквами:

$A = \{\text{Аристотель - основоположник логики}\}$

$B = \{\text{На яблонях растут бананы}\}.$

Истинному высказыванию ставится в соответствие 1, ложному — 0. Таким образом, $A = 1$, $B = 0$.

Составные высказывания на естественном языке образуются с помощью союзов, которые в алгебре высказываний заменяются на логические операции. Логические операции задаются таблицами истинности и могут быть графически проиллюстрированы с помощью диаграмм Эйлера-Венна.

Логическая операция КОНЬЮНКЦИЯ (логическое умножение):

- в естественном языке соответствует союзу и;
- в алгебре высказываний обозначение $\&$;
- в языках программирования обозначение **And**.

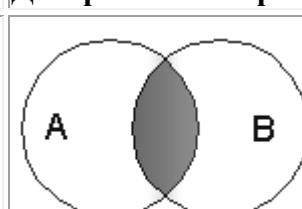
Конъюнкция — это логическая операция, ставящая в соответствие каждым двум простым высказываниям составное высказывание, являющееся истинным тогда и только тогда, когда оба исходных высказывания истинны.

В алгебре множеств конъюнкции соответствует операция *пересечения множеств*, т.е. множеству получившемуся в результате умножения множеств A и B соответствует множество, состоящее из элементов, принадлежащих одновременно двум множествам.

Таблица истинности

A	B	$A \& B$
0	0	0
0	1	0
1	0	0
1	1	1

Диаграмма Эйлера-Венна



Логическая операция ДИЗЬЮНКЦИЯ (логическое сложение):

- в естественном языке соответствует союзу или;
- обозначение \square ;
- в языках программирования обозначение **Or**.

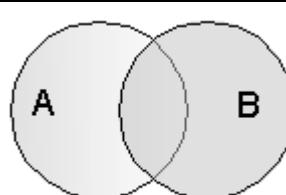
Дизьюнкция — это логическая операция, которая каждым двум простым высказываниям ставит в соответствие составное высказывание, являющееся ложным тогда и только тогда, когда оба исходных высказывания ложны и истинным, когда хотя бы одно из двух образующих его высказываний истинно.

В алгебре множеств дизьюнкции соответствует операция *объединения множеств*, т.е. множеству получившемуся в результате сложения множеств A и B соответствует множество, состоящее из элементов, принадлежащих либо множеству A , либо множеству B .

Таблица истинности

A	B	$A \square B$
0	0	0
0	1	1
1	0	1
1	1	1

Диаграмма Эйлера-Венна



Логическая операция ИНВЕРСИЯ (отрицание):

- в естественном языке соответствует словам **неверно**, **что...** и частице **не**;
- обозначение \bar{A} ;
- в языках программирования обозначение **Not**;

Отрицание - это логическая операция, которая каждому простому высказыванию ставит в соответствие составное высказывание, заключающееся в том, что исходное высказывание отрицается.

В алгебре множеств логическому отрицанию соответствует операция *дополнения до универсального множества*, т.е. множеству получившемуся в результате отрицания множества A соответствует множество \bar{A} , дополняющее его до универсального множества.

Таблица истинности		Диаграмма Эйлера-Венна
A	\bar{A}	

Логическая операция ИМПЛИКАЦИЯ (логическое следование):

- в естественном языке соответствует обороту **если ..., то ...**;
- обозначение \square .

Импликация - это логическая операция, ставящая в соответствие каждым двум простым высказываниям составное высказывание, являющееся ложным тогда и только тогда, когда условие (первое высказывание) истинно, а следствие (второе высказывание) ложно.

A	B	$A \square B$
0	0	1
0	1	1
1	0	0
1	1	1

Логическая операция ЭКВИВАЛЕНЦИЯ (равнозначность):

- в естественном языке соответствует оборотам речи **тогда и только тогда; в том и только в том случае;**
- обозначения \square, \sim .

Эквиваленция – это логическая операция, ставящая в соответствие каждым двум простым высказываниям составное высказывание, являющееся истинным тогда и только тогда, когда оба исходных высказывания одновременно истинны или одновременно ложны. Таблица истинности эквиваленции:

A	B	$A \square B$
0	0	1
0	1	0
1	0	0
1	1	1

Логические операции имеют следующий приоритет: действия в скобках, инверсия, $\&$, \square , \neg , \square .

Логические законы и правила преобразования логических выражений

Логические выражения называются *равносильными*, если их истинностные значения совпадают при любых значениях, входящих в них логических переменных.

В алгебре логики имеется ряд законов, позволяющих производить равносильные преобразования логических выражений. Приведем соотношения, отражающие эти законы.

1. Закон двойного отрицания:

$$A = \bar{\bar{A}}.$$

Двойное отрицание исключает отрицание.

2. Переместительный (коммутативный) закон:

— для логического сложения:

$$A \square B = B \square A;$$

— для логического умножения:

$$A \& B = B \& A.$$

Результат операции над высказываниями не зависит от того, в каком порядке берутся эти высказывания.

В обычной алгебре $a + b = b + a$, $a \square b = b \square a$.

3. Сочетательный (ассоциативный) закон:

— для логического сложения:

$$(A \square B) \square C = A \square (B \square C);$$

— для логического умножения:

$$(A \& B) \& C = A \& (B \& C).$$

При одинаковых знаках скобки можно ставить произвольно или вообще опускать.

В обычной алгебре:

$$(a + b) + c = a + (b + c) = a + b + c,$$

$$a \square (b \square c) = a \square (b \square c) = a \square b \square c.$$

4. Распределительный (дистрибутивный) закон:

— для логического сложения:

$$(A \square B) \& C = (A \& C) \square (B \& C);$$

— для логического умножения:

$$(A \& B) \square C = (A \square C) \& (B \square C).$$

Определяет правило выноса общего высказывания за скобку.

В обычной алгебре:

$$(a + b) \square c = a \square c + b \square c.$$

5. Закон общей инверсии (законы де Моргана):

— для логического сложения

$$\overline{A \vee B} = \overline{A} \& \overline{B};$$

— для логического умножения:

$$\overline{A \& B} = \overline{A} \square \overline{B}$$

6. Закон идемпотентности (от латинских слов *idem* — тот же самый и *potens* — сильный; дословно — равносильный):

— для логического сложения:

$$A \square A = A;$$

— для логического умножения:

$$A \& A = A.$$

Закон означает отсутствие показателей степени.

7. Законы исключения констант:

— для логического сложения:

$$A \square 1 = 1, \quad A \square 0 = A;$$

— для логического умножения:

$$A \& 1 = A, \quad A \& 0 = 0.$$

8. Закон противоречия:

$$A \& \overline{A} = 0.$$

Невозможно, чтобы противоречащие высказывания были одновременно истинными.

9. Закон исключения третьего:

$$A \square \overline{A} = 1.$$

Из двух противоречащих высказываний об одном и том же предмете одно всегда истинно, а второе — ложно, третьего не дано.

10. Закон поглощения:

— для логического сложения:

$$A \square (A \& B) = A;$$

— для логического умножения:

$$A \& (A \square B) = A.$$

11. Закон исключения (склеивания):

— для логического сложения:

$$(A \& B) \square (\bar{A} \& B) = B;$$

— для логического умножения:

$$(A \square B) \& (\bar{A} \square B) = B.$$

12. Закон контрапозиции (правило перевертывания):

$$(A \Leftrightarrow B) = (B \Leftrightarrow A).$$

Справедливость приведенных законов можно доказать табличным способом: выписать все наборы значений А и В, вычислить на них значения левой и правой частей доказываемого выражения и убедиться, что результирующие столбцы совпадут.

Пример 1. Найдите X, если $\overline{X \vee A} \square \overline{X \vee \bar{A}} = B$.

Для преобразования левой части равенства последовательно воспользуемся законом де Моргана для логического сложения и законом двойного отрицания:

$$(\bar{X} \& \bar{A}) \square (\bar{X} \& A)$$

Согласно распределительному закону для логического сложения:

$$\bar{X} \& (\bar{A} \square A)$$

Согласно закону исключения третьего и закона исключения констант:

$$\bar{X} \& 1 = \bar{X}$$

Полученную левую часть приравняем правой:

$$\bar{X} = B$$

Окончательно получим, что $X = \bar{B}$.

Пример 2. Упростите логическое выражение $(A \square B \square C) \& (\overline{A \vee \bar{B} \vee C})$

Правильность упрощения проверьте с помощью таблиц истинности для исходного и полученного логического выражения.

Согласно закону общей инверсии для логического сложения (первому закону Моргана) и закону двойного отрицания:

$$(A \square B \square C) \& (\overline{A \vee \bar{B} \vee C}) = (A \square B \square C) \& (\bar{A} \& B \& \bar{C})$$

Согласно распределительному (дистрибутивному) закону для логического сложения:

$$(A \square B \square C) \& (\bar{A} \& B \& \bar{C}) = (A \& \bar{A}) \square (B \& \bar{A}) \square (C \& \bar{A}) \square (A \& B) \square (B \& B) \square (C \& B) \square (A \& \bar{C}) \square (B \& \bar{C}) \square (C \& \bar{C})$$

Согласно закона противоречия:

$$(A \& \bar{A}) = 0; (C \& \bar{C}) = 0$$

Согласно закона идемпотентности

$$(B \& B) = B$$

Подставляем значения и, используя переместительный (коммутативный) закон и группируя слагаемые, получаем:

$$0 \square (A \& B) \square (\bar{A} \& B) \square B \square (C \& B) \square (\bar{C} \& B) \square (C \& \bar{A}) \square (A \& \bar{C}) \square 0$$

Согласно закона исключения (склеивания)

$$(A \& B) \square (\bar{A} \& B) = B$$

$$(C \& B) \square (\bar{C} \& B) = B$$

Подставляем значения и получаем:

$$0 \square B \square B \square B \square (C \& \bar{A}) \square (A \& \bar{C}) \square 0$$

Согласно закона исключения констант для логического сложения и закона идемпотентности:

$$0 \square B \square 0 \square B \square B = B$$

Подставляем значения и получаем:

$$B \square (C \& \bar{A}) \square (A \& \bar{C})$$

Согласно распределительному (дистрибутивному) закону для логического умножения:

$$(C \& \bar{A}) \square (A \& \bar{C}) = (C \square A) \& (C \square \bar{C}) \& (\bar{A} \square A) \& (\bar{A} \square \bar{C})$$

Согласно закона исключения третьего:

$$(C \square \bar{C}) = 1$$

$$(\bar{A} \square A) = 1$$

Подставляем значения и окончательно получаем:

$$B \& \bar{A} \& \bar{C}$$

Раздел 2. Информационная деятельность человека

Тема 2.1. Информационная деятельность человека.

1. Классификация информационных процессов.
2. Информационные ресурсы.
3. Этические и правовые нормы информационной деятельности человека.
4. Информационная безопасность.

Единство законов обработки информации в системах различной природы (физических, экономических, биологических и т.п.) является фундаментальной основой теории информационных процессов, определяющей ее обще значимость и специфичность. Объектом изучения этой теории является информация - понятие во многом абстрактное, существующее "само по себе" вне связи с конкретной областью знания, в которой она используется.

Информационная деятельность, информационные процессы и информационные технологии это совокупность методов и средств сбора, передачи, хранения и обработки информации. Условием эффективного функционирования информационных систем является согласованность методологического, информационного, методического, программно-технического, лингвистического и организационного обеспечения.

Как известно, процесс- это совокупность последовательных, действий для достижения какого-либо результата. В переводе с латинского processus — продвижение.

Исходя из вышеизложенного, можно предположить, что информационный процесс — это совокупность действий, направленных на использование либо продвижение информации.

Однако не все так просто, информационный процесс — явление неоднозначное, трактуемое по-разному и учеными, и даже законодателями. Например, в ст. 2 Федерального закона «Об информации, информатизации и защите информации» №24—ФЗ от 20 февраля 1995 г. информационные процессы определены как процессы сбора, обработки, накопления, хранения, поиска и распространения информации.

Понятие ресурс определяется в словаре русского языка С.И. Ожегова как запас, источник чего-нибудь. Что касается информационного ресурса, то это понятие является относительно новым и еще не до конца освоенным. Хотя в последние годы все чаще можно встретить термин «информационный ресурс» не только в научной литературе, но и в традиционном социальном диалоге. Причиной тому служит процесс информатизации общества, в котором роль информации выходит на первый план.

Согласно ст. 2 Федерального закона «Об информации, информатизации и защите информации» №24—ФЗ от 20 февраля 1995 г. **Информационные ресурсы** - это отдельные

документы и отдельные массивы документов, документы и массивы документов в информационных системах (библиотеках, архивах, фондах, банках данных, других информационных системах).

Держателями информационных ресурсов (владельцами и собственниками) могут быть как самостоятельные информационные центры, информационные организации, предприятия, учреждения, обладающие статусом юридического лица, так и отдельные информационные структуры в составе иных юридических лиц, а также физические лица.

Защита информации — комплекс мероприятий, направленных на обеспечение важнейших аспектов информационной безопасности (целостности, доступности и, если нужно, конфиденциальности информации и ресурсов, используемых для ввода, хранения, обработки и передачи данных).

Появление персонального компьютера произвело *информационную революцию*. За дисплей ЭВМ сел непрограммист. Информация стала ресурсом наравне с материалами, энергией и капиталом. Появилась новая экономическая категория - национальные информационные ресурсы. Истощение природных ресурсов привело к использованию воспроизводимых ресурсов, основанных на применении научных знаний. Затраты на получение профессиональных знаний, используемых в производстве наукоемких изделий на базе персональных компьютеров, составляют приблизительно 70% себестоимости, а число занятых в сфере обработки информации — 60-90% экономически активного населения индустриально развитых стран.

Профессиональные знания экспортируются посредством продажи наукоемкой продукции. Производство вновь становится мелкосерийным с быстрым ростом производительности труда и увеличением номенклатуры производимых изделий. *Информация стала стратегическим ресурсом*. Внедряются дистанционное обучение, автоматизированные офисы, всемирные каталоги изделий. Проектируются геоинформационные системы по управлению природными богатствами, экологией, информационной политикой правительства. Страны становятся зависимыми от источников информации, от уровня развития и эффективности использования средств передачи и переработки информации. Происходит информатизация общества.

Проблема обеспечения защиты информации является одной из важнейших при построении надежной информационной структуры учреждения на базе ЭВМ. Эта проблема охватывает как физическую защиту данных и системных программ, так и защиту от несанкционированного доступа к данным, передаваемым по линиям связи и находящимся на накопителях, являющегося результатом деятельности как посторонних лиц, так и специальных программ-вирусов. Таким образом, в понятие защиты данных включаются вопросы сохранения целостности данных и управления доступа к данным (санкционированность).

Система называется **безопасной**, если она, используя соответствующие аппаратные и программные средства, управляет доступом к информации так, что толькоенным образом авторизованные лица или же действующие от их имени процессы получают право читать, писать, создавать и удалять информацию.

Очевидно, что абсолютно безопасных систем нет, и здесь речь идет о надежной системе в смысле «система, которой можно доверять» (как можно доверять человеку). Система считается надежной, если она с использованием достаточных аппаратных и программных средств обеспечивает одновременную обработку информации разной степени секретности группой пользователей без нарушения прав доступа.

Раздел 3. Средства информационных и коммуникационных технологий

Тема 3.1.Архитектура компьютеров

1. Основные характеристики компьютеров
2. Виды программного обеспечения

Компьютер (англ. computer — «вычислитель»), ЭВМ (электронная вычислительная машина) — машина для проведения вычислений, а также приёма, переработки, хранения и выдачи информации по заранее определённому алгоритму (компьютерной программе).

На заре эры компьютеров считалось, что основная функция компьютера — вычисление. Однако в настоящее время полагают, что основная их функция — управление.

Джон фон Нейман обосновал структуру вычислительной машины. По фон Нейману основными блоками ЭВМ являются: **блок управления, арифметико-логическое устройство, память и устройства ввода-вывода** (рисунок 1).

Несмотря на то, что принципы фон-Неймана были сформулированы еще в середине прошлого столетия, на заре развития цифровой вычислительной техники, они являются жизненными и по настоящее время. Современные персональные ЭВМ реализованы также на основе названных принципов.

В современных ЭВМ обычно устройство управления (УУ) и арифметико-логическое устройство (АЛУ) объединяются в центральный процессор. Кроме

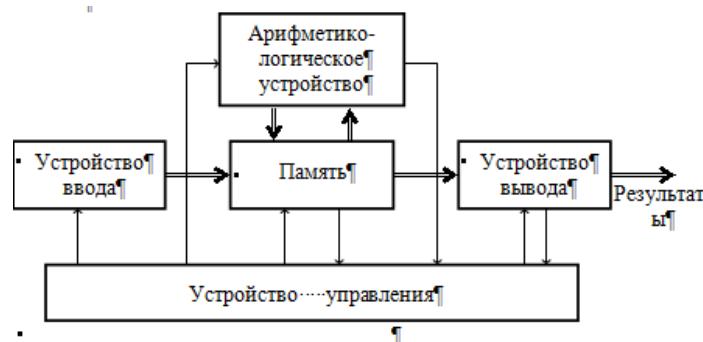


Рисунок 1. Обобщенная структурная схема ЭВМ

оперативной памяти используется внешняя память, предназначенная для хранения программ и данных. Схема устройства такой ЭВМ представлена на рисунке 2. Следует отметить, что внешняя память отличается от устройств ввода и вывода тем, что данные в нее заносятся в виде, удобном компьютеру, но недоступном для непосредственного восприятия человеком. Так, накопитель на магнитных дисках относится к внешней памяти, а клавиатура - устройство ввода, дисплей и печать - устройства вывода.

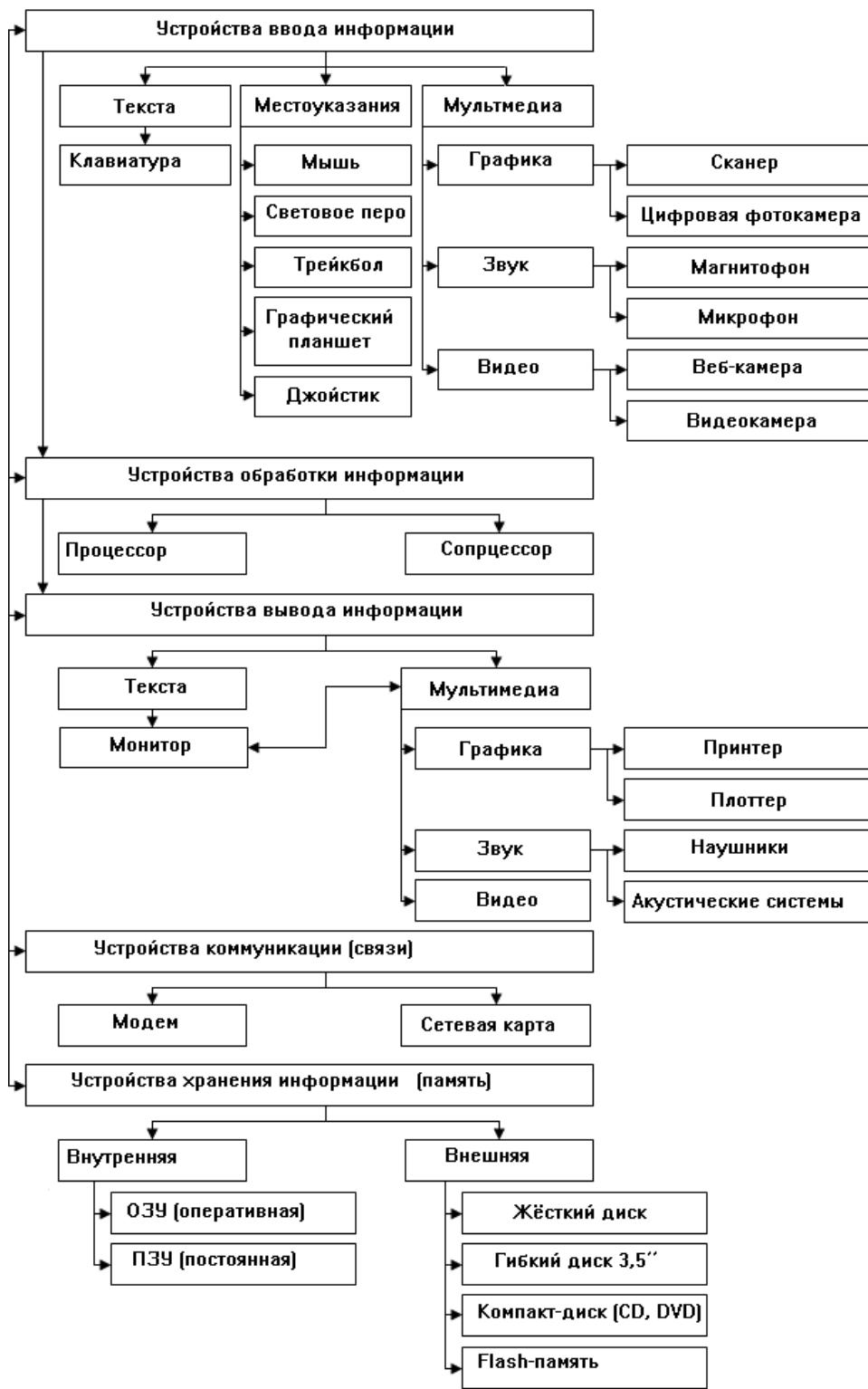


Рисунок 2. Обобщенная структура современной ЭВМ

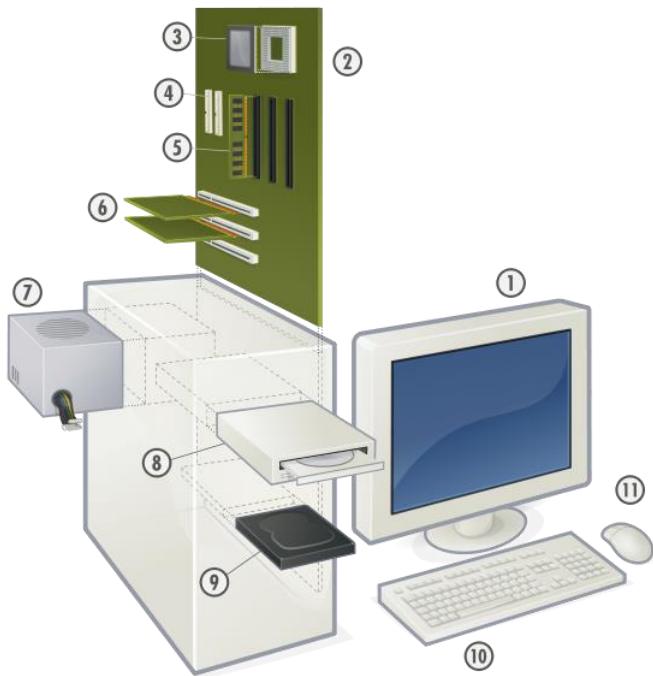
Процессор ЭВМ является **преобразователем информации**, поступающей из памяти и внешних устройств (сюда относятся **выборка команд** из памяти, **кодирование и декодирование**, **выполнение различных**, в том числе и арифметических, **операций**, **согласование работы узлов компьютера**). Более детально функции процессора будут обсуждаться ниже.

В построенной по описанной схеме ЭВМ происходит последовательное считывание команд из памяти и их выполнение. Номер (адрес) очередной ячейки памяти, из которой будет извлечена следующая команда программы, указывается специальным устройством - **счетчиком команд** в УУ. Его наличие также является одним из характерных признаков рассматриваемой архитектуры.

Аппаратное обеспечение компьютера



Структура аппаратного обеспечения ПК.



1. Монитор
2. Материнская плата
3. Процессор
4. IDE-слот
5. Оперативная память
6. Платы расширения (видео, звуковая...)
7. Блок питания
8. Привод для дисков (CD/ DVD)
9. Винчестер
10. Клавиатура
11. Мышь

Рис. Расположение основных устройств, входящих в состав ПК.

Таблица

Основные блоки	системный блок монитор устройства ввода-вывода
Устройства в составе системного блока	материнская плата центральный процессор оперативная память жёсткий диск графическая плата звуковая плата сетевая плата дисковод CD-привод DVD-привод TV-тюнер
Периферийные (внешние) устройства	принтер сканер графопостроитель (плоттер) модем микрофон акустика ИБП – источник бесперебойного питания клавиатура мышь графический планшет тачпад вебкамера фотокамера

Устройства, входящие в состав системного блока

Материнская плата

Материнская плата — печатная плата, на которой осуществляется монтаж большинства компонентов компьютерной системы. Название происходит от английского *motherboard*, иногда используется сокращение *MB* или слово *mainboard* — главная плата.

Материнская плата *обеспечивает связь между всеми устройствами ПК*, посредством передачи сигнала от одного устройства к другому.

На поверхности материнской платы имеется большое количество разъемов предназначенных для установки других устройств: *sockets* — гнезда для процессоров; *slots* — разъемы под оперативную память и платы расширения; *контроллеры портов ввода/вывода*.

Центральный процессор

Центральный процессор, или центральное процессорное устройство (ЦПУ) (англ. *central processing unit* — *CPU*) — основная микросхема компьютера, в которой и производятся все вычисления. ЦПУ имеет размеры $5 \times 5 \times 0,3$ см, устанавливается на материнской плате. С остальными устройствами компьютера, и в первую очередь с оперативной памятью, процессор связан несколькими группами проводников, называемых шинами. Основных шин три: *шина данных*, *адресная шина* и *командная шина*.

Адресная шина. У процессоров Intel Pentium (а именно они наиболее распространены на сегодняшний день в персональных компьютерах) адресная шина 32-разрядная, то есть состоит из 32 параллельных линий. В зависимости от того, есть напряжение на какой-то из линий или нет, говорят, что на этой линии выставлена единица или ноль. Комбинация из 32 нулей и единиц образует 32-

разрядный адрес, указывающий на одну из ячеек оперативной памяти. К ней и подключается процессор для копирования данных из ячейки в один из своих регистров.

Шина данных. По этой шине происходит копирование данных из оперативной памяти в регистры процессора и обратно. В компьютерах, собранных на базе процессоров Intel Pentium, шина данных 64-разрядная, то есть состоит из 64 линий, по которым за один раз на обработку поступают сразу 8 байтов.

Шина команд. Для того чтобы процессор мог обрабатывать данные, ему нужны команды. Он должен знать, что следует сделать с теми байтами, которые хранятся в его регистрах. Эти команды поступают в процессор тоже из оперативной памяти, но не из тех областей, где хранятся массивы данных, а оттуда, где хранятся программы. Команды тоже представлены в виде байтов. Самые простые команды укладываются в один байт, однако есть и такие, для которых нужно два, три и более байтов. В большинстве современных процессоров шина команд 32-разрядная, хотя существуют 64-разрядные процессоры и даже 128-разрядные.

Основные параметры процессоров

Основными параметрами процессоров являются: *рабочее напряжение, разрядность, рабочая тактовая частота, коэффициент внутреннего умножения тактовой частоты и размер кэш-памяти.*

Рабочее напряжение процессора обеспечивает материнская плата, поэтому разным маркам процессоров соответствуют разные материнские платы (их надо выбирать совместно).

Разрядность процессора показывает, сколько бит данных он может принять и обработать в своих регистрах за один раз (за один такт). Первые процессоры были 4-разрядными. Современные процессоры семейства Intel Pentium являются 32-разрядными, хотя и работают с 64-разряднойшиной данных (разрядность процессора определяется не разрядностью шины данных, а разрядностью командной шины).

Оперативная память

Оперативная память (ОЗУ — оперативное запоминающее устройство). Существует два типа *оперативной памяти* - память с произвольным доступом (*RAM - Random Access Memory*) и память, доступная только на чтение (*ROM - Read Only Memory*). Процессор ЭВМ может обмениваться данными с оперативной памятью с очень высокой скоростью, на несколько порядков превышающей скорость доступа к другим носителям информации, например дискам.

Оперативная память - энергозависимая, т. е. данные в ней хранятся только до выключения ПК. Для долговременного хранения информации служат диски, винчестеры, компакт-диски и т. п.

Жесткий диск

Накопитель на жёстких магнитных дисках, жёсткий диск или винчестер (англ. *Hard Disk Drive, HDD*) — энергонезависимое, перезаписываемое компьютерное запоминающее устройство. Является основным накопителем данных практически во всех современных компьютерах.

В отличие от «гибкого» диска (дискеты), информация в НЖМД записывается на жёсткие (алюминиевые или стеклянные) пластины, покрытые слоем ферромагнитного материала. Считывающие головки в рабочем режиме не касаются поверхности пластин благодаря прослойке воздуха, образуемой при быстром вращении дисков.

Характеристики

Интерфейс — способ, использующийся для передачи данных. Современные накопители могут использовать интерфейсы ATA (IDE, EIDE), Serial ATA, SCSI, SAS, FireWire, USB и Fibre Channel.

Ёмкость — количество данных, которые могут храниться накопителем. Ёмкость современных устройств может достигать до 1.5 Тб, в ПК сегодня распространены винчестеры ёмкостью 80, 120, 200, 320 Гб. В отличие от принятой в информатике системе приставок, обозначающих кратную 1024 величину (кило=1024), производителями при обозначении ёмкости жёстких дисков используются кратные 1000 величины. Так, напр., «настоящая» ёмкость жёсткого диска, маркированного как «200 Гб», составляет 186,2 Гб.

Физический размер — почти все современные накопители для персональных компьютеров и серверов имеют размер либо 3,5, либо 2,5 дюйма. Последние чаще применяются в ноутбуках.

Скорость вращения шпинделя — количество оборотов шпинделя в минуту. От этого параметра в значительной степени зависят время доступа и скорость передачи данных. В настоящее время выпускаются винчестеры со следующими стандартными скоростями вращения: 4200, 5400 и

7200 (ноутбуки), 7200 и 10000 (персональные компьютеры), 10000 и 15000 об./мин. (серверы и высокопроизводительные рабочие станции).

Графическая плата

Графическая плата (известна также как графическая карта, видеокарта, видеоадаптер) (англ. *videocard*) — устройство, преобразующее изображение, находящееся в памяти компьютера, в видеосигнал для монитора.

Обычно видеокарта является платой расширения и вставляется в специальный разъём (ISA, VLB, PCI, AGP, PCI-Express) для видеокарт на материнской плате, но бывает и встроенной.

Современная графическая плата состоит из следующих основных частей:

Графический процессор (GPU) — занимается расчетами выводимого изображения, освобождая от этой обязанности центральный процессор, производит расчеты для обработки команд трехмерной графики. Является основой графической платы, именно от него зависят быстродействие и возможности всего устройства. Современные графические процессоры по сложности мало чем уступают центральному процессору.

Видеоконтроллер — отвечает за формирование изображения в видеопамяти.

Видеопамять — выполняет роль буфера, в котором в цифровом формате хранится изображение, предназначенное для вывода на экран монитора. Ёмкость видеопамяти так же, как и оперативной памяти *кратна степени числа два* и на сегодняшний день измеряется в мегабайтах.

Цифро-аналоговый преобразователь (ЦАП) — служит для преобразования изображения, формируемого видеоконтроллером, в уровня интенсивности цвета, подаваемые на аналоговый монитор. Большинство ЦАП имеют разрядность 8 бит на канал — получается по 256 уровней яркости на каждый основной цвет RGB, что в сумме дает 16.7 млн. цветов.

Звуковая плата

Звуковая плата (также называемая звуковая карта, аудиоадаптер) используется для записи и воспроизведения различных звуковых сигналов: речи, музыки, шумовых эффектов.

Любая современная звуковая карта может использовать несколько способов воспроизведения звука. Одним из простейших является преобразование ранее оцифрованного сигнала снова в аналоговый. Глубина оцифровки сигнала (например, 8 или 16 бит) определяет качество записи и, соответственно, воспроизведения. Так, 8-разрядное преобразование обеспечивает качество звучания кассетного магнитофона, а 16-разрядное — качество компакт-диска.

В настоящее время звуковые карты чаще бывают встроенными в материнскую плату, но выпускаются также и как отдельные платы расширения.

Сетевая плата

Сетевая плата (также известная как сетевая карта, сетевой адаптер, Ethernet card, NIC (англ. *network interface card*)) — печатная плата, позволяющая взаимодействовать компьютерам между собой, посредством локальной сети.

На сетевой плате имеются разъёмы для подключения кабеля витой пары и/или BNC-коннектор для коаксиального кабеля.

Сетевая карта относится к устройствам коммуникации (связи). Кроме нее к устройствам коммуникации относится модем, но он служит для организации связи в глобальной сети (Интернет). Скорость передачи данных устройствами коммуникации измеряется в битах в секунду (а также в Кбит/с и Мбит/с). Модем, используемый для подключения домашнего компьютера к сети Интернет по телефонной линии, обычно обеспечивает пропускную способность до 56 Кбит/с, а сетевая карта — до 100 Мбит/с.

Программное обеспечение (произношение *обеспечéние* не рекомендуется) — наряду с аппаратными средствами, важнейшая составляющая информационных технологий, включающая *компьютерные программы и данные*, предназначенные для решения определённого круга задач и хранящиеся на машинных носителях.

Программное обеспечение представляет собой *алгоритм*, реализованный в виде последовательности инструкций для процессора.

В компьютерном жаргоне часто используется слово «софт» от английского *software*.



Рис. Иерархия программного обеспечения.

По назначению программное обеспечение разделяется на системное, прикладное и инструментальное.

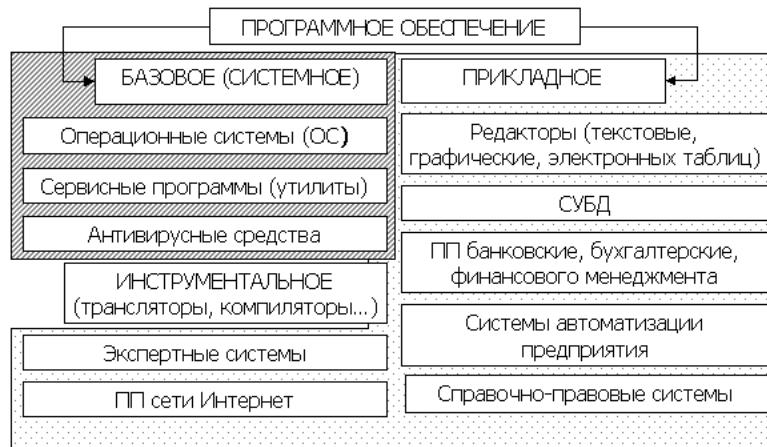


Рис. Классификация программного обеспечения.

Инструментальное ПО предназначено для использования в ходе проектирования, разработки и сопровождения программ. Это: интегрированные среды разработки, SDK, компиляторы, интерпретаторы, линковщики, ассемблеры, отладчики и т.п.

Прикладное ПО (приложения) — программы, предназначенные для выполнения определенных пользовательских задач и рассчитанные на непосредственное взаимодействие с пользователем. К прикладному ПО относятся: банковские и бухгалтерские программы, игры, текстовые и графические редакторы, Интернет-клиенты и т. п.

Системное ПО используется для обеспечения работы компьютера самого по себе и выполнения прикладных программ. Конкретные виды системного программного обеспечения включают загрузчики, операционные системы, драйверы устройств, утилиты (сервисные программы). Наиболее общая часть системного программного обеспечения — операционная система.

Раздел 4. Технологии создания и преобразования информационных объектов

Тема 4.1. Текстовый редактор (текстовый процессор MS Word)

1. Структура окна MS WORD.
2. Режимы работы с документом.
3. Способы выделения фрагментов текста, перемещение по тексту.

Microsoft Word - мощный текстовый процессор, предназначенный для выполнения всех процессов обработки текста: набор, модификация, сохранение, удаление.

Возможности WORD:

- набора текстовой информации на 21 языках,
- использование масштабированных шрифтов, использование цвета фона и цвета шрифта,
- автоматическая коррекция текста по границам,
- автоматический перенос слов,
- сохранение текста в определенный устанавливаемый промежуток времени,
- проверка правописания, подбор синонимов,
- вставка в текст графики в стандарте *.psx или *.bmp,

- использование мастеров и шаблонов, позволяющих в считанные минуты создать деловое письмо, факс, автобиографию, расписание, календарь и т.д.,
- поиск заданного слова или фрагмента текста,
- копирование, перемещение слова или фрагмента текста во внутренний буфер,
- установка пароля,
- возможность работать с несколькими документами одновременно,
- возможность разбить одно активное окно по горизонтали на два.

Обзор пунктов горизонтального меню.

Строка меню содержит пункты, объединяющие сходные по своему назначению команды.

Обычно в этой строке присутствует следующие пункты:

Файл- все операции, связанные с файлом, печатью и форматом страниц;

Правка- все операции с буфером, поиском и замены строк, а также отмена и повторение последних операций;

Вид- задает вариант просмотра документа, позволяет задать колонтитулы, настроить элементы окна;

Вставка- все операции, связанные со вставкой, включая внедрение объектов;

Формат- все, связанное с форматированием: вид абзаца, выбор шрифта, выравнивание;

Сервис- различные дополнительные функции, такие как проверка орфографии, синонимы, перенос слов и т. д., а также настройка Word;

Таблица- все операции, связанные с созданием и обработкой таблиц;

Окно- все операции, связанные со вторичными окнами Word, в том числе при работе с несколькими документами сразу;

?- все, связанное со справкой;

Панели инструментов.

Вкл/выкл панелей инструментов: **ВИД - ПАНЕЛИ ИНСТРУМЕНТОВ**. На экране появится диалоговое окно Панели инструментов. С помощью этого окна кнопки можно сделать большими или маленькими, цветными или нет, добавить новые кнопки в панели. Включить / выключить соответствующую панель на экране. Однако, не стоит экспериментировать с этим пунктом меню, пока нет достаточного опыта работы.

Кнопка свернуть используется для того, чтобы свернуть открытое окно в пиктограмму.

Кнопка восстановить используется для разворачивания окна на весь экран или для того, чтобы свернуть окно до первоначальных размеров.

Кнопка закрыть используется для закрытия окна.

Вертикальная полоса прокрутки текста используется для прокрутки текста по вертикали, если он не умещается в окне по высоте.

Горизонтальная полоса прокрутки текста используется для прокрутки текста по горизонтали, если он не умещается в окне по ширине.

Строка состояния- в ней отображается различная полезная информация, как то:

- текущая страница;
- текущий раздел;
- текущая страница / общее число страниц документа;

Далее отображается текущее положение курсора:

- расстояние от верхнего края документа;
- строка, считая от начала текущей позиции;
- колонка или количество символов слева от курсора.

В следующем поле отображается режимы редактирования:

- **ЗАП**- запись макрокоманды;
- **ИСПР**- активная маркировка исправлений;
- **ВДЛ**- включен режим выделения с помощью клавиши **F8**;
- **ЗАМ**- если темный, яркий цвет, то включен режим замены, блеклый- режим вставки.

В строке состояния выводится также краткая подсказка по функциям кнопок и опций меню или текущая операция. Вертикальная и горизонтальная линейки размера- обычные линейки, с помощью которых пользователь может лучше ориентироваться при работе с текстом. С их помощью можно также изменить левый, правый, нижний, верхний, абзацный отступы. Сделать видимыми или не видимыми линейки можно с помощью команды **ЛИНЕЙКА** из меню **ВИД**. Если возле этого

пункта стоит, то линейка будет присутствовать на экране. Для того, чтобы включить / выключить линейку надо просто щелкнуть мышью на этом пункте.

Рабочая область экрана - это поле, на котором мы будем набирать свой текст. Курсор это вертикальный штрих на этом поле. Курсор определяет место появления очередного символа на экране. Здесь может быть и горизонтальный штрих, показывающий конец текста.

Режим просмотра документа на экране, т. е. Вид его на экране можно задать с помощью кнопок переключения режимов, которые находятся на левом конце горизонтальной полосы прокрутки или с помощью команд (**ОБЫЧНЫЙ, СТРУКТУРА, РАЗМЕТКА СТРАНИЦЫ, ГЛАВНЫЙ ДОКУМЕНТ**) из меню **ВИД**.

Изображение документа на экране может быть увеличено, чтобы легче его было читать, или уменьшено, чтобы видеть всю страницу целиком. Для управления масштабом изображения можно пользоваться меню **ВИД / МАСШТАБ** или кнопкой **МАСШТАБ** на панели инструментов.

Способы выделения фрагментов документа Word

Фрагмент	Способ выделения
Слово	Два раза щелкнуть мышью по слову
Предложение	Щелкнуть мышью в любом месте предложения при нажатой клавиши Ctrl
Строка	Щелкнуть мышью в полосе выделения – слева от текста
Абзац	Дважды щелкнуть мышью в полосе выделения напротив абзаца или трижды щелкнуть внутри абзаца
Любой фрагмент	Щелкнуть мышью в начале выделяемого фрагмента, затем нажать клавишу Shift и щелкнуть по последнему символу выделяемого фрагмента
Вертикальный блок	При выделении держать нажатой клавишу Alt
Рисунок, кадр, вставленный объект	Щелкнуть мышью по объекту
Весь документ	Меню ПРАВКА => Выделить все
	Трижды щелкнуть мышью в полосе выделения
	Ctrl + щелчок в полосе выделения
	Нажать Ctrl + A или Ctrl + 5 (Num) на цифровой клавиатуре

Тема 4.2. Электронные таблицы (Табличный процессор MS Excel)

1. Структура окна MS EXCEL.
2. Создание формул, элементы формулы, виды ссылок.
3. Использование в вычислениях стандартных функций.

Окно программы

Microsoft Excel – табличный процессор, программа для создания и обработки электронных таблиц. Ярлык Microsoft Excel чаще всего имеет вид на рис.1.



Рис.1

Microsoft Excel позволяет работать с таблицами в двух режимах:

***Обычный** – наиболее удобный для выполнения большинства операций.

***Разметка страниц** – удобен для окончательного форматирования таблицы перед распечаткой. Границы между страницами в этом режиме отображаются синими пунктирными линиями. Границы таблицы – сплошной синей линией, перетягивая которую, можно изменять размеры таблицы.

Для перехода между режимами **Обычный** и **Разметка страниц** используются соответствующие пункты меню **Вид**.

Под панелями инструментов Microsoft Excel обычно находится строка формул, а в нижней части окна – строка состояния. Чтобы вывести или убрать эти строки, следует в меню **Вид** выбрать соответствующие пункты: **Строка формул** или **Строка состояния**.

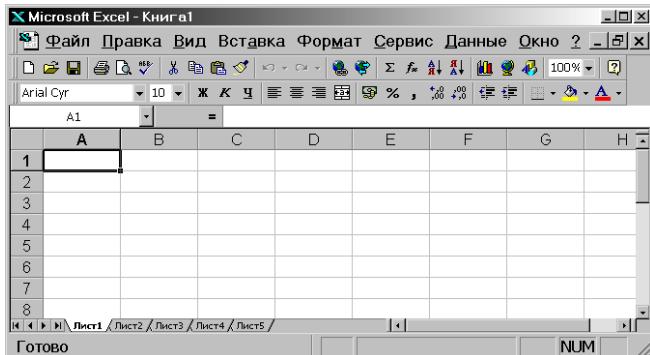


Рис.2. Окно Microsoft Excel



Рис.3

Заполнение ячеек

Для ввода данных в ячейку необходимо сделать ее активной и ввести данные с клавиатуры. Данные появятся в ячейке и в строке редактирования (рис.3). Для завершения ввода следует нажать **Enter** или одну из клавиш управления курсором. Процесс ввода данных закончится и активной станет соседняя ячейка.

Чтобы отредактировать данные в ячейке, необходимо:

- * сделать ячейку активной и нажать клавишу **F2** или дважды щелкнуть в ячейке мышью;
- * в ячейке появится текстовый курсор, который можно передвигать клавишами управления курсором;
- * отредактировать данные;
- * выйти из режима редактирования клавишей **Enter**.

Внимание! Перед выполнением любой команды *Microsoft Excel* следует завершить работу с ячейкой, т. е. выйти из режима ввода или редактирования.

Работа с формулами

Основные сведения

Вычисления в таблицах выполняются с помощью **формул**. **Формула** может состоять из математических операторов, значений, ссылок на ячейку и имена функций. Результатом выполнения формулы есть некоторое новое значение, содержащееся в ячейке, где находится формула. Формула начинается со знака равенства "**=**". В формуле могут использоваться арифметические операторы **+, -, *, /**. Порядок вычислений определяется обычными математическими законами.

Примеры формул: **= (A4+B8)*C6, =F7*C14+B12**.

Константы – текстовые или числовые значения, которые вводятся в ячейку и не могут изменяться во время вычислений.

Ссылка на ячейку или группу ячеек – способ, которым указывается конкретная ячейка или несколько ячеек. Ссылка на отдельную ячейку – ее координаты. Значение пустой ячейки равно нулю.

Ссылки на ячейки бывают двух типов:

относительные – когда ячейки обозначаются относительным смещением от ячейки с формулой (например: **F7**).

абсолютные – когда ячейки обозначаются координатами ячеек в сочетании со знаком **\$** (например: **\$F\$7**).

Комбинация предыдущих типов (например: **F\$7**).

При копировании формул относительные ссылки изменяются на размер перемещения.

Для обращения к группе ячеек используются специальные символы:

: (двоеточие) – формирует обращение к блоку ячеек. Через двоеточие указывается левая верхняя и правая нижняя ячейки блока. Например: **C4:D6** – обращение к ячейкам **C4, C5, C6, D4, D5, D6**.

; (точка с запятой) – обозначает объединение ячеек. Например, **D2:D4;D6:D8** – обращение к ячейкам **D2, D3, D4, D6, D7, D8**.

Для ввода формулы в ячейку следует ввести знак '=' и формулу для вычисления. После нажатия клавиши **Enter** в ячейке появится результат вычисления. При выделении ячейки, содержащей формулу, формула появляется в строке редактирования.

Тема 4.3. Информационные системы. Базы данных MS Access

1. Понятие «информационная система», понятие «модели данных», виды моделей данных: сетевая, иерархическая, реляционная.
2. Система управления данными: понятие, основные функции.
3. Характеристика основных объектов баз данных.

Необходимость хранения и обработки больших объемов информации привела к возникновению автоматизированных информационных систем, ориентированных на коллективное пользование.

Информационная система (ИС) – система, реализующая автоматизированный сбор, обработку и манипулирование данными и включающая технические средства обработки данных, программное обеспечение и соответствующий персонал.

Примером могут служить такие ИС, как документальные (библиотечные) информационно-поисковые системы, предоставляющие пользователю по его запросу перечень документов в интересующей его области, и системы резервирования авиа- и железнодорожных билетов типа «Сирена» и «Экспресс», в которых ответом на запрос пассажира является документ в виде билета либо сообщение об отсутствии свободных мест. Примером информационной системы является также и Internet.

ВИДЫ МОДЕЛЕЙ ДАННЫХ

Ядром любой базы данных является модель данных. Модель данных представляет собой множество структур данных, ограничений целостности и операций манипулирования данными. С помощью модели данных могут быть представлены объекты предметной области и взаимосвязи между ними.

Модель данных – совокупность структур данных и операций их обработки.

СУБД основывается на использовании иерархической, сетевой или реляционной модели, на комбинации этих моделей или на некотором их подмножестве.

Рассмотрим три основных типа моделей данных:

- **Иерархическую**: Иерархическая структура представляет совокупность элементов, связанных между собой по определенным правилам. Объекты, связанные иерархическими отношениями, образуют ориентированный граф (перевернутое дерево);
- **Сетевую**: В сетевой структуре при тех же основных понятиях (уровень, узел, связь) каждый элемент может быть связан с любым другим элементом.
- **Реляционную**: Реляционная модель ориентирована на организацию данных в виде двумерных таблиц. Каждая реляционная таблица представляет собой двумерный массив и обладает следующими свойствами:
 - ✓ каждый элемент таблицы – один элемент данных;
 - ✓ все столбцы в таблице однородные, т.е. все элементы в столбце имеют одинаковый тип (числовой, символьный и т.д.) и длину;
 - ✓ каждый столбец имеет уникальное имя;
 - ✓ одинаковые строки в таблице отсутствуют;
 - ✓ порядок следования строк и столбцов может быть произвольным

Основу любой информационной системы составляет база данных.

База данных (БД) – именованная совокупность данных, организованная по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования данными.

Для создания, ведения и совместного использования баз данных многими пользователями применяются совокупности программных и языковых средств - **системы управления базами данных (СУБД)**.

СУБД должна обеспечивать реализацию следующих требований:

- Позволять пользователям создавать новые *базы данных* и определять их *схемы (schemata)* - логические структуры данных с помощью *языка определения данных (DDL)*;
- предлагать пользователям возможности задания *запросов (queries)* - вопрос, затрагивающий те или иные аспекты информации, и модификации данных средствами соответствующего *языка запросов (query language)*, или *языка управления данными (DML)*;
- Поддерживать способность сохранения больших объемов информации - до многих гигабайт и более - на протяжении длительного периода времени, предотвращать опасность несанкционированного доступа к данным и гарантировать эффективность операций их просмотра и изменения;
- управлять единовременным доступом к данным со стороны многих пользователей, исключая возможность влияния действий одного пользователя на результаты, получаемые другим, и запрещая совместное обращение к данным, чреватое их порчей.

Объекты Access

СУБД Access ориентирована на работу с *объектами базы данных*, к которым относятся *таблицы, и запросы*, а также *объекты приложений* для работы с базой данных — *формы, отчеты, страницы, макросы и модули*.

Для типовых процессов обработки данных — просмотра, обновления, поиска по заданным критериям, получения отчетов — в Access имеются *средства конструирования* таких объектов, как *формы, запросы, отчеты и страницы*. Объекты приложений — *формы, отчеты, страницы* — состоят из графических элементов, называемых *элементами управления*. Основные элементы управления служат для связи объекта с записями таблиц, являющимися источниками данных.

При создании приложений пользователя также используются средства программирования, реализуемые объектами другого типа — *макросами и модулями* на языке программирования *Visual Basic for Application (VBA)*.

Каждый объект и элемент управления имеет свои свойства, определяя которые, можно настраивать объекты и элементы управления. С каждым объектом и элементом управления связывается набор событий. Их можно обрабатывать с помощью макросов или процедур на VBA.

Объекты представлены в окне базы данных Access. Все операции по работе с объектами базы данных и приложений начинаются в этом же окне.

Таблицы создаются пользователем для хранения данных об одном информационном объекте модели данных предметной области. Таблица состоит из *полей* (столбцов) и *записей* (строк). Каждое поле содержит одну характеристику объекта предметной области. В записи собраны сведения об одном экземпляре этого объекта.

Запросы создаются пользователем для выборки необходимых данных из одной или нескольких связанных таблиц. Результатом выполнения запроса является таблица, которая может быть использована наряду с другими таблицами базы данных при обработке данных. Запрос может формироваться с помощью *QBE — запросов по образцу* или с помощью *инструкции SQL — языка структурированных запросов*. Используя запрос, можно также обновить, удалить или добавить данные в таблицы или создать новые таблицы на основе уже существующих.

Формы являются основным средством создания диалогового интерфейса приложения пользователя. Форма может создаваться для ввода и просмотра взаимосвязанных данных базы на экране в удобном виде, который соответствует документу, привычному для пользователя. Формы также могут применяться для создания панелей управления в приложениях.

Отчеты предназначены для формирования выходных документов, содержащих результаты решения задач пользователя, и вывода их на печать.

Страницы доступа к данным являются диалоговыми Web-страницами, которые поддерживают динамическую связь с базой данных и позволяют просматривать, редактировать и вводить данные в базу, работая в окне броузера.

Макросы – это программы, которые содержат описание последовательности действий, выполняемых, как правило, при наступлении некоторого события в объекте или элементе управления приложения. Каждое действие реализуется макрокомандой. Создание макросов

осуществляется в диалоговом режиме путем выбора нужных макрокоманд и задания параметров, используемых ими при выполнении.

Модули содержат процедуры на языке Visual Basic for Applications. Могут создаваться *процедуры-функции*, которые разрабатываются пользователем для реализации нестандартных функций в приложении пользователя, *процедуры-подпрограммы* и *процедуры для обработки событий*.

Тема 4.4. Мастер презентаций MS Power Point

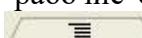
1. *Назначение программы. Режимы работы.*
2. *Форматирование слайда. Настройка анимации, смены слайдов.*
3. *Вставка объектов: рисунка, объекта WordArt, диаграммы, гиперссылки, таблицы.*

Microsoft PowerPoint (полное название — **Microsoft Office PowerPoint**) — это программа для создания и проведения презентаций, являющаяся частью Microsoft Office и доступная в редакциях для операционных систем Microsoft Windows и Mac OS. **PowerPoint** предоставляет пользователю мощные функции работы с текстом, включая отрисовку контура текста, средства для рисования, построение диаграмм, широкий набор стандартных иллюстраций и т.п.

Презентация - это набор слайдов и спецэффектов, сопровождающих их показ на экране, раздаточный материал, а также конспект и план доклада, хранящиеся в одном файле, созданным с помощью PowerPoint.

Слайд - это отдельный кадр презентации, который может включать в себя заголовок, текст, графику, диаграммы и т.д. Созданные средствами PowerPoint слайды можно распечатать на черно-белом или цветном принтере, либо изготовить 35-миллиметровые слайды на фотопленке, создав раздаточный материал. В качестве раздаточного материала могут служить распечатанные в компактном виде слайды презентации: два, четыре или шесть слайдов на одной странице. В процессе работы над презентацией в PowerPoint можно получить конспект доклада, при печати которого на каждой странице будет выведено уменьшенное изображение слайда и текст, поясняющий его содержание.

В PowerPoint имеется возможность отображения слайдов в трех основных режимах: обычный режим, режим сортировщика слайдов и режим просмотра слайдов. По умолчанию выбран обычный режим.

Обычный режим является основным режимом редактирования для создания и оформления презентации. В данном режиме имеются три рабочие области: область слева для редактирования структуры текста слайда (область **Структура** ) и самих слайдов, отображающихся в виде эскизов (область **Слайды** ), область слайда справа, в которой отображается текущий слайд в крупном масштабе и область заметок снизу.

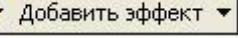
Для включения **Обычного режима** внизу на панели **Режим отображения** необходимо нажать кнопку .

Режим сортировщика слайдов — единственный режим, позволяющий просматривать слайды в виде эскизов.

По завершении создания и редактирования презентации сортировщик слайдов дает законченную картину всей презентации, упрощая ее реорганизацию, добавление или удаление слайдов и предварительный просмотр эффектов анимации и смены слайдов.

Для просмотра презентации в режиме сортировщика слайдов необходимо внизу на панели **Режим отображения** нажать кнопку .

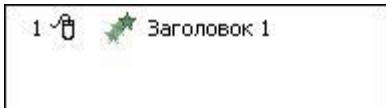
Добавление анимации текста или изображения

1. Выберите блок текста или изображение, к которому требуется применить анимацию.
2. Выберите пункт меню **Показ слайдов → Настройка анимации.**
3. На панели **Настройка анимации** нажмите кнопку  **Добавить эффект**  и выполните одно или несколько следующих действий.

- Если во время показа слайдов требуется ввести текст или объект в сопровождении определенного визуального эффекта, выберите команду **Вход** , а затем — нужный эффект.
- Если требуется добавить определенный визуальный эффект в текст или объект, находящиеся на самом слайде, выберите команду **Выделение** , а затем — нужный эффект.
- Если требуется добавить определенный визуальный эффект в текст или объект, который вызывает удаление текста или объекта со слайда в заданный момент, выберите команду **Выход** , а затем — нужный эффект.

Проведите настройку анимации текстового блока следующим образом.

1. На панели **Настройка анимации**, в списке настроек анимации, выберите анимированный текстовой объект.



2. Щелкните стрелку, выберите пункт **Параметры эффектов**, а затем выполните одно или несколько следующих действий.
 - Если к тексту требуется применить анимацию по буквам, на вкладке **Эффекты** в списке **Анимация текста** выберите команду **По буквам**.
 - Если к тексту требуется применить анимацию по отдельным словам, на вкладке **Эффекты** в списке **Анимация текста** выберите команду **По словам**.
 - Если требуется применить анимацию на уровне абзацев или маркеров абзаца, на вкладке **Анимация текста** в списке **Группировка текста** выберите нужную настройку.

Добавление эффекта смены слайдов

Для добавления одного и того же перехода на все слайды в демонстрации слайдов, выполните следующие действия:

1. В меню **Показ слайдов** выберите команду **Смена слайдов**.
2. В списке выберите нужный эффект смены слайдов.
3. Нажмите кнопку **Применить ко всем слайдам**.

Для добавления различных переходов между слайдами повторите следующие действия для всех слайдов, к которым требуется добавить переходы.

1. В обычном режиме в области **Слайды** выберите слайды, к которым требуется добавить переходы.
2. В меню **Показ слайдов** выберите команду **Смена слайдов**.
3. В списке выберите нужный эффект смены слайдов.

Изменение порядка слайдов

Выполните одно из следующих действий.

- На вкладке **Структура** в обычном режиме выделите один или несколько значков слайдов и перетащите выделенные значки на новое место.
- В области **Слайды** в обычном режиме выделите один или несколько эскизов слайдов и перетащите выделенные эскизы на новое место.

Чтобы выделить несколько слайдов из одного ряда, щелкните их значки или эскизы, удерживая клавишу клавиши **[Shift]**.

Вставка объектов: рисунка

Для обеспечения наглядности презентации рекомендуется использовать графические объекты, в том числе разнообразные рисунки. Рисунок можно выбрать из стандартной библиотеки - «Картинки» или графический объект, который создали Вы, для этого воспользуйтесь функцией «Из файла». Основное требование к рисунку обоснованность по отношению к содержанию презентации. Второе немаловажное требование: оптимизация по объему, для этого рекомендуется использовать программы графические редакторы, например, Photoshop

Вставка диаграммы

Если есть статистические данные по тематике презентации, то по ним можно построить диаграмму. Это можно осуществить при помощи функции «Диаграмма». После того как выбрана

эта функция, на слайде появляется макет диаграммы и таблица данных . Меняя данные в таблице, меняется вид диаграммы, надписи на осях и разметка.

Так же можно выбрать формат слайда при создании нового. Тогда в центре появится пиктограмма диаграммы, для активизации которой нужно кликнуть по ней мышью 2 раза. Далее все действия аналогичны описанным ранее

Вставка таблицы

Еще один инструмент наглядного представления информации на слайдах - таблица. Рекомендуется использовать для ограниченного количества информации, поскольку большая по наполненности таблица за собой необходимость использования мелкого шрифта. Но поскольку не бывает правил без исключений, таблица дает исключительные возможности по систематизации данных, то единственный выход из данной ситуации, дополнительные слайды с пояснениями. Итак, для вставки таблицы в слайд выбрать в меню **Вставка/ Таблица**

Настройка параметров таблицы (количество строк и колонок) стандартное для программ Microsoft. После нажатия **OK** на слайде появится таблица. При ее заполнении обращайте внимание, чтобы границы не ушли за границы слайда, иначе при демонстрации эти области буду не видимыми. Менять ширину колонок и ширину строк можно при помощи мыши.

Раздел 5 Телекоммуникационные технологии

Тема 5.1. Компьютерные сети. Средства телекоммуникационных технологий.

1. Назначение и классификация компьютерных сетей.
2. Классификация и топология локальных сетей.
3. Глобальная сеть Internet. Представление о структуре и системе адресации.
4. Поисковые системы.

Локальные и глобальные сети. Назначение и классификация компьютерных сетей. Абоненты сетей, станция. Физическая передающая среда. Режимы передачи. Средства передачи. Локальная одноранговая сеть. Сеть с выделенным сервером. Достоинства и недостатки разных типов сетей. Глобальная сеть Internet. Представление о структуре и системе адресации.

Электронная почта. Передача файлов с помощью протокола FTP. Всемирная паутина (Word Wide Web). Понятие гипертекста. Телеконференции. Электронные доски объявлений BBS.

Средства поиска информации в сети Internet. Обозреватель Microsoft Internet Explorer. Поисковые системы. Синтаксис языка запросов. Копирование информации.

Компьютерная сеть - совокупность взаимосвязанных через каналы передачи данных компьютеров, обеспечивающих пользователей средствами обмена информацией и коллективного использования ресурсов сети:

- аппаратных,
- программных,
- информационных.

Таблица 5.1.1.

	Локальные (ЛВС / LAN - Local Area Network)	Глобальные (Internet / Wan - Wide Area Network)
Функция	Связывает абонентов одного или нескольких близлежащих зданий одного предприятия	Объединяет абонентов, расположенных по всему миру
Канал передачи данных	<ul style="list-style-type: none">• витая паракоаксиальный кабельоптоволоконный кабельрадиоканалинфракрасный канал	оптический кабель телефонные линии спутниковые каналы
Расстояния между ЭВМ	до 20 км.	до 15000 км.

Локальная вычислительная сеть (ЛВС) - группа ЭВМ, а также периферийное оборудование, объединенные в пределах одного или нескольких близлежащих зданий автономными (не арендуемыми) высокоскоростными каналами передачи цифровых данных (проводными или беспроводными: медными, волоконно-оптическими, СВЧ или ИК-диапазона).

Служит для решения информационных задач (например, в рамках какой-либо организации), а также совместного использования объединенных информационных и вычислительных ресурсов.

ЛВС могут иметь в своем составе средства для выхода в Интернет.

Каждый компьютер, включаемый в локальную сеть должен иметь сетевую плату, в разъем которой и подключается связующий кабель. Кабели, выходящие из различных компьютеров объединяются в устройстве, называемом *сетевой концентратор* (switch, HUB). Сетевые концентраторы также могут иметь связь друг с другом, объединяя вместе подсети различных участков здания. Таким образом, обеспечивается прохождение сигналов между всеми устройствами,ключенными в сеть.

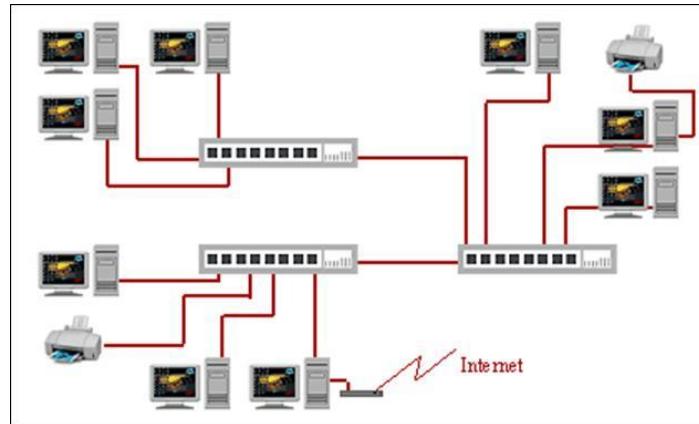


Рис. 5.1.1. Одноранговая ЛВС.

Преимущества работы в локальной сети:

- Возможность хранения данных персонального и общего использования на дисках файлового сервера.
- Возможность постоянного хранения ПО, необходимого многим пользователям, в единственном экземпляре на дисках файлового сервера.
- Обмен информацией между всеми компьютерами сети.
- Одновременная печать всеми пользователями сети на общесетевых принтерах.
- Обеспечение доступа с любого компьютера локальной сети к ресурсам Интернет, при наличии единственного коммуникационного узла глобальной сети.

В зависимости от принципов построения ЛВС подразделяются на следующие основные типы:

Одноранговая ЛВС (peer-to-peer, p2p LAN) - "безсерверная" организация построения сети.

Термин "одноранговая" означает, что все компьютеры, объединенные в сеть, имеют в ней одинаковые права. Каждый пользователь одноранговой сети может определить состав файлов, которые он предоставляет для общего использования. Таким образом, пользователи одноранговой сети могут работать как со всеми своими файлами, так и с файлами, предоставляемыми другими ее пользователями. Создание одноранговой сети обеспечивает также совместную эксплуатацию периферийных устройств. Серверные функции при этом распределены по сети. Компьютер, к которому подключен принтер, может являться для всех остальных сервером печати, компьютер, к которому присоединен модем – сервером Интернет... На рисунке 67 изображена схема одноранговой сети.

ЛВС с выделенным сервером (Dedicated Server Network: file-server или client-server architecture) – означает, что специально выделенный самый мощный компьютер (*сервер*) в сети берет на себя основные функции по ее обслуживанию: управляет созданием, поддержкой и использованием общих информационных ресурсов, включая доступ к ее базам данных и отдельным файлам, а также их защите и аудит.

К нему же присоединяются основные периферийные устройства: модем, принтер. Т.е. серверные функции не распределены по сети, а централизованы. Один и тот же компьютер-сервер может являться и файловым сервером, и Интернет-сервером, и сервером печати. Остальные компьютеры сети тогда именуют "клиентами" или "рабочими станциями".

Слово *Интернет* (*Internet*) происходит от словосочетания *Interconnected networks* (связанные сети), это глобальное сообщество малых и больших сетей.

Обращаясь в Интернет, мы пользуемся услугами *Интернет-провайдера* или *ISP* (Internet Service Provider - поставщик услуг Интернета).

ISP - это организация, которая имеет собственную высокоскоростную сеть, объединенную с другими сетями по всему земному шару. Провайдер подключает к своей сети клиентов, которые становятся частью сети данного провайдера и одновременно частью всех объединенных сетей, которые и составляют Интернет.

В основе объединения малых и больших сетей (которые и составляют Интернет) лежит цепь договорных соглашений. Каждый пользователь Интернета имеет договор с определенным провайдером о подключении к его сети. В простейшем случае этот договор может выглядеть как одноразовая карта доступа, содержащая всю необходимую информацию для подключения пользователя в локальную сеть провайдера: номера телефонов модемного пула провайдера, имя и пароль пользователя для доступа в сеть. В свою очередь провайдеры договариваются о соединении их сетей.

В более широком смысле *Интернет* - это информационное пространство, распределенное среди миллионов компьютеров во всем мире, которые постоянно обмениваются данными. Основная задача Интернета - это связь. Связь круглогодичная, высоконадежная. Для того чтобы осуществлять такую высоконадежную связь, была разработана специальная Интернет-технология доставки данных.

Когда вы получаете на свой ПК Web-страницы с удаленного сервера, происходит совсем другой процесс. Послание разбивается на отдельные порции данных - группы пакетов. Каждый пакет посыпается на место назначения по наиболее оптимальному из доступных путей. Если какой-то пакет теряется, система посылает его заново. Поэтому, даже если какой-то участок Сети окажется нарушенным, это не повлияет на доставку пакета, который будет направлен по альтернативному пути. Таким образом, во время доставки данных между двумя пользователями нет необходимости в фиксированной линии связи.

Такую надежность сеть Интернет унаследовала от своего прототипа - сети *ARPAnet*, разработанной по заказу Министерства обороны США. Она задумывалась именно как сеть, устойчивая к повреждениям, например, в случае бомбовых ударов, и способная продолжать нормальное функционирование при выходе из строя любой ее части. Первое компьютерное сообщение по ARPANet было передано в 1969 году.

Изначально Сеть использовалась преимущественно в научных проектах. Однако со временем и коммерческие компании поняли, что Интернет - это возможность общения с огромной аудиторией потенциальных клиентов и покупателей. С этих пор стало активно развиваться направление электронной коммерции. Сеть стала не только источником информации, но и огромной витриной для заказа самых различных товаров.

Адресация в Интернете

IP-адресация

Чтобы компьютеры, объединенные в сеть, могли обмениваться сообщениями, каждый из них должен иметь уникальный адрес. В сети Интернет это 32-х разрядный (т.е. 32-х битный = 4-х байтный) адрес, называемый *IP-адрес*.

Таблица 5.1.2.

IP-адрес двоичный	11011100	11010111	00001110	00010110
IP-адрес десятичный	220	215	14	22

В точечно-десятичной нотации IP-адрес может выглядеть, например, так: 220.215.14.22. Каждая часть, разделенная точкой, представляет собой один байт, и, следовательно, максимальное десятичное число, которое может быть представлено одним байтом 255 ($2^8=256$, от 0 до 255).

Но, для человека такая система адресации сложна, так же как нам сложно помнить, набирать и диктовать одиннадцатизначные телефонные номера, поэтому в 1984 году Полом Мокапетрисом была разработана *надстройка над IP-адресацией*, называемая системой *DNS* (*domain name system, система доменных имен*).

DNS - система доменных имен

Доменные имена системы DNS – синонимы IP-адреса, так же, как имена в адресной книжке вашего телефона – синонимы телефонных номеров. Они символные, а не числовые; они удобнее для запоминания и ориентации; они несут смысловую нагрузку.

www.irnet.ru -> таблицы DNS -> 193.232.70.36

Доменные имена также уникальны, т.е. нет в мире двух одинаковых доменных имен. Доменные имена, в отличие от IP-адресов необязательны, они приобретаются дополнительно.

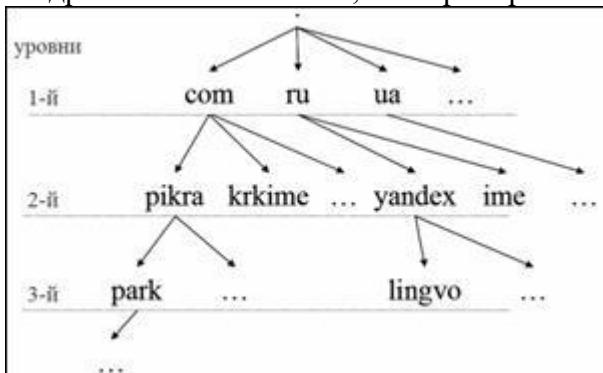


Рис. 5.1.2. Иерархия в системе DNS.

Так же уникальны адреса, которые указываются на конвертах при доставке писем обычной почтой. В мире нет стран с одинаковыми названиями. И если названия городов иногда и повторяются, то в сочетании с делением на более крупные административные единицы типа районов и областей они становятся уникальными. А названия улиц не должны повторяться в пределах одного города. Таким образом, адрес на основе географических и административных названий однозначно определяет точку назначения.

Домены имеют аналогичную иерархию. Имена доменов отделяются друг от друга точками: lingvo.yandex.ru, krkime.com.

Домены первого уровня разделяются на *тематические* и *географические*.

Таблица 5.1.3. Тематические домены 1-ого уровня

COM	Commercial (для коммерческих организаций)
NET	Networks (Интернет, телекоммуникационные сети)
ORG	Organizations (некоммерческие организации либо организации, не попадающие в другие категории)
INFO	Information (открытый для всех домен)
BIZ	Business Organizations (аналог com)
NAME	Personal (для частных лиц)
INT	International Organizations (международные организации)
EDU	Educational (образовательные проекты США)
MIL	US Dept of Defense (департамент безопасности США)
GOV	US Government (правительство США)
MUSEUM	Museums (музеи)
AERO	Air-transport industry (воздушно-транспортная индустрия)
COOP	Cooperatives (кооперативы)

Регистрация доменного имени второго уровня в тематических доменных зонах доступна для организаций или частного лица независимо от географического положения. Стоимость владения доменным именем в такой зоне не превышает \$35 в год. Юридического оформления владения не требуется, требуется только выбрать подходящее доменное имя из числа незанятых и перечислить деньги. Одна из организаций, осуществляющих регистрацию имен в этих доменных зонах – Network Solution (<http://www.netsol.com>).

Территориальные домены первого уровня, в отличие от тематических, всегда двухбуквенные.

Исторически сложилось так, что Россия владеет двумя национальными доменами: .RU и .SU. Последний остался за РФ после раз渲ала Советского Союза. Впрочем, в настоящий момент ведется пересмотр территориальных доменов, и в ближайшем будущем Россия может лишиться зоны .SU.

Регистрация доменных имён второго уровня в зонах .RU и .SU производится организацией RU-CENTER (<http://www.nic.ru>). Стоимость доменного имени в зоне .RU - \$20+НДС в год, в зоне .SU - \$100+НДС в год. При регистрации требуется юридическое оформление договорных отношений, которое занимает около месяца, с учетом пересылки документов в Москву и из Москвы.

Система адресации URL

Чтобы найти в Интернете какой-либо документ, достаточно знать *ссылку* на него - так называемый *универсальный указатель ресурса* (URL - Uniform Resource Locator), который определяет местонахождение каждого файла, хранящегося на компьютере, подключенном к Интернету.

Адрес URL является сетевым расширением понятия полного имени ресурса в операционной системе. В URL, кроме имени файла и директории, где он находится, указывается сетевое имя компьютера, на котором этот ресурс расположен, и протокол доступа к ресурсу, который можно использовать для обращения к нему. Система адресации URL и адресация почтовой службы имеют сходную структуру.

Тема 5.2. Язык HTML. Основы WEB-дизайна

Интернет — это сложная электронная информационная структура, представляющая собой глобальную сеть, которая позволяет связывать между собой компьютеры в любой точке земного шара.

WWW — World Wide Web («Всемирная паутина») — это общемировая гипертекстовая информационная система (является частью Интернета).

Web — страница — это отдельный комбинированный документ сети WWW, который может содержать текст, графику, анимацию, звуковые и другие объекты. Хранится в файле *.html.

Сайт (веб-сайт, ресурс) — это место в интернете, которое определяется своим адресом (URL), имеет своего владельца и состоит из веб-страниц, которые воспринимаются как единое целое.

Создание сайтов — составной процесс, состоящий из нескольких этапов:

1. разработка дизайна,
2. вёрстка,
3. программирование,
4. безопасность.

Сопровождение сайтов:

- это техническая поддержка сайта;
- помошь в обновлении контента;
- внесение корректировок в работу ресурса.

Методы создания и сопровождения сайтов:

- вручную на языке HTML (в БЛОКНОТе);
- с помощью редакторов сайтов (HEFS, DreamWeaver и др.);
- с помощью Конструктора сайтов на основе готового шаблона (ucoz.ru, narod.ru и др.);
- с помощью систем управления сайтов (Joomla, 1С Битрикс и др.).

Этапы создания WEB-страницы:

1. Разработка проекта (Постановка задачи);
 - Главная тема страницы.
 - Текстовое содержание (грамотный язык).
 - Планировка размещения информации на странице (верстка).
 - Графика (набор рисунков, анимации).
 - Стиль дизайна (сочетания цветов, фоны и т. п.)
2. Алгоритм заполнения страницы.
3. Программирование.

Основным тегом языка HTML является тег (пара тэгов) **<html> ... </html>**

Они обозначают, что находящиеся между ними строки представляют единый HTML-документ. Это важно, так как сам по себе документ является обычным текстовым файлом в формате ASCII. Без этих тегов ни один Интернет-браузер не в состоянии идентифицировать формат документа и правильно его интерпретировать. Следующим тегом, который относится к разметке структуры документа, является тег заголовка документа <head> ... </head>

Задача заголовка документа – предоставление необходимой информации для программы, интерпретирующей документ. Элементы находящиеся внутри заголовка (кроме <title>) не отображаются на экране во время просмотра документа.

Раздел <title> ... </title>

является единственным обязательным элементом заголовка и служит для того, чтобы дать документу название, которое будет отображаться в заголовке окна браузера.

Название документа представляет собой обычную текстовую строку, которая характеризует содержание документа в целом.

Тег <body> ... </body>

содержит тело документа, которое отображается браузером на экране. Этот тег имеет несколько параметров. Некоторые из них:

bgcolor – цвет фона документа background – указывает путь к графическому файлу, который будет использован в качестве фона документа.

Например,

```
<body bgcolor='yellow' background='fon.jpg'>
```

Примечание: если параметры цвета фона у данного тега не указаны, то фон документа будет белым. В HTML цвета можно задавать непосредственно названиями, а можно в шестнадцатеричной форме. Цветовая схема базируется на трех основных цветах RGB. Для каждого направления задается шестнадцатеричное значение от 00 до FF. Затем эти числа объединяются в одно число, перед которым ставится символ #.

Например, число #800080 обозначает фиолетовый цвет.

HTML предполагает, что автор документа ничего не знает о компьютере пользователя. Пользователь вправе установить любой размер окна и пользоваться любым из имеющихся у него шрифтов. Это означает, что место переноса в строке определяется только программой просмотра и установками конечного пользователя.

Для разбиения текста на абзацы используется тег

```
<p> ... </p>
```

Для начала абзаца следует разместить тег <p> в начало строки абзаца. Данный тег имеет большое значение при стилевом оформлении документа.

Изменяя значения шестнадцатеричного числа, можно подобрать требуемый цвет.

Тег абзаца имеет важный параметр

align

Данный параметр может иметь значения:

left –	выравнивание текста по левой границе окна	браузера; right –
	выравнивание текста по правой границе окна	браузера;
justify -	выравнивание по ширине; center -	выравнивание текста по центру окна
	браузера.	

По умолчанию действует параметр left.

```
<html>
<title>Пример с абзацами</title>
<body bgcolor="#ffff80">
<p align='center'>В нормативном документе Минобразования  
"Методика применения дистанционных образовательных  
технологий (дистанционного обучения) в образовательных  
учреждениях высшего, среднего и дополнительного  
профессионального образования Российской Федерации"  
дистанционное обучение формулируется  
следующим образом.</p>
<p align='right'>дистанционное обучение - совокупность  
образовательных технологий, при которых  
целенаправленное опосредованное или не полностью  
опосредованное взаимодействие обучающегося  
и преподавателя осуществляется независимо  
от места их нахождения и распределения во времени,  
на основе педагогически организованных информационных  
технологий, прежде всего с использованием средств  
телекоммуникаций и телевидения.</p>
</body>
</html>
```

Шрифты

Одной из важных возможностей HTML является полный контроль облика создаваемого документа, в том числе управление шрифтами. На практике тип выводимых шрифтов сильно зависит от программно-аппаратного обеспечения компьютера пользователя. Однако для HTML-документов проблема значительно облегчается тем, что на всех машинах сети должен быть установлен стандартный набор шрифтов. Обычно это Verdana, Arial, Times New Roman. Для работы со шрифтами используется тег

** ... ** Тег font имеет следующие параметры
size

Служит для указания размера шрифта в условных единицах от 1 до 7. Принято считать, что размер «нормального» шрифта соответствует числу 3.

Например, **** Размер шрифта равен 4 **** Параметр
color

Устанавливает цвет шрифта, который может быть указан как в формате RGB, так и стандартным именем.

Например, **** Цвет
желтый **** шрифта

```
<html>
<title>Пример работы со шрифтами</title>
<body>
<p><font color='red' size='1'>Размер равен 1</font></p>
<p><font color='blue' size='2'>Размер равен 2</font></p>
<p><font color='purple' size='3'>Размер равен 3</font></p>
<p><font color='green' size='4'>Размер равен 4</font></p>
<p><font color='brown' size='5'>Размер равен 5</font></p>
<p><font color='magenta' size='6'>Размер равен 6</font></p>
<p><font color='olive' size='7'>Размер равен 7</font></p>
</body>
</html>
```

Заголовки

Заголовки имеют важное значение в структурировании HTML-документа и являются одними из самых употребляемых в языке разметки текста. Их назначение – показать структуру документа. HTML располагает шестью уровнями заголовков.

<h1> ... </h1>
...
<h6> ... </h6>

Самый верхний уровень `<h1>`, нижний - `<h6>`. Все эти теги также имеют параметр align.

```
<html>
<title>Пример с заголовками</title>
<body bgcolor='#affffa'>
<h1>Заголовок первого уровня</h1>
<h2>Заголовок второго уровня</h2>
<h3>Заголовок третьего уровня</h3>
<h4>Заголовок четвертого уровня</h4>
<h5>Заголовок пятого уровня</h5>
<h6>Заголовок шестого уровня</h6>
</body>
</html>
```

`<i> ... </i>` - начало-конец оформления фрагмента текста курсивом; ` ... ` - начало-конец оформления фрагмента текста более жирным начертанием; `<u> ... </u>` - начало-конец оформления фрагмента

текста

```
<html>
<title>Пример с параметрами текста</title>
<body>
<p><i>Электронные учебные пособия</i> являются основой дистанционных образовательных технологий.</p>
<p>ЭУП создаются на высоком <b>научном и методическом уровне</b> и <u>должны полностью соответствовать</u> требованиям образовательного стандарта и рабочей программы.</p>
</body>
</html>
```

подчеркиванием.

Перевод строки

Для того, чтобы перейти на новую строку в любом месте текущей строки, в HTML существует тэг разрыва строки

`
`

Этот тег заставляет программу просмотра выводить стоящие после него символы с новой строки. В отличие от тэга абзаца, тэг `
` не добавляет пустую строку и не имеет закрывающего тэга.

Разделительные линии

Еще одним методом разделения документа является использование разделительных линий. Для проведения такой линии в теле документа используется тег

`<hr>`

Тег обладает следующими параметрами:

`align` – выравнивание линии; `width` – длина линии (может устанавливаться в %); `size` – ширина линии (в пикселях); `color` – цвет линии.

```
<html>
<title>Пример с линией</title>
<body bgcolor='#cdaa77'>
<hr width='10%' align='center' size='5px' color='#cdcccc'>
</body>
</html>
```

Вставка изображений в документ

Существует достаточно много форматов графических документов: GIF, BMP, JPEG (JPG), PCX, WMF и др. Однако при оформлении HTML-документа лучше ограничиться двумя: GIF, JPEG, поскольку их понимает большинство современных программ просмотра (Internet Explorer, Netscape Navigator, Opera и др.). Вставляется ссылка на файл с картинкой

Для вставки изображения в тело документа используется тег

``

Этот тег одинарный, он не имеет закрывающего тега. Его основной параметр:

```
<html>
<title>Пример с рисунком</title>
<body>
<img src='picture.jpg'>
<p>Это рисунок.
</html>
```

изображением.

Обтекание текстом изображения

По умолчанию после вставки изображения следующая строка оказывается после рисунка. Если есть необходимость, чтобы текст обтекал рисунок, можно воспользоваться параметром align для тега

```
<html>
<title>Пример с рисунком</title>
<body>
<img src='picture.jpg' align='left'>
<p>По умолчанию после вставки изображения  
следующая строка оказывается после рисунка.  
Если есть необходимость, чтобы текст обтекал  
рисунок, можно воспользоваться параметром align  
для тега img.
</body>
<img></html>
```

Принудительное выравнивание по центру В некоторых случаях возникает необходимость выровнять объект по центру документа, но либо объект не обладает свойством align, либо значение align='center' отображается некорректно. В этих случаях используют тег принудительного выравнивания

```
<html>
<head>
<title>Пример с рисунком</title>
</head>
<body>
<img src='kartinka.gif' align='center'>
<p align='center'>Рисунок 1</p>
<center><img src='kartinka.gif'></center>
<p align='center'>Рисунок 2</p>
</body>
</html>
```

по центру<center> ... </center>

Перевод текста в верхний / нижний регистр Для перевода фрагмента текста в верхний регистр используется тег

^{...}

Для перевода фрагмента текста в нижний регистр используется тег

```
<html>
<head>
<title>Пример работы с регистрами</title>
</head>
<body>
<p>Пример работы с верхним регистром:  
<br>x<sup>2</sup></p>
<p>Пример работы с нижним регистром:  
<br>x<sub>n</sub></p>
</body>
<sub> ... </sub></html>
```

Таблицы

Для оформления таблиц в документе используется несколько тегов:

<table> ... </table> - начало – конец таблицы,
 <tr> ... </tr> - начало – конец строки таблицы,
 <td> ... </td> - начало – конец ячейки строки таблицы.

Вместе с тегом table обычно используют параметры:

border – определение рамки таблицы;

align – выравнивание таблицы в окне браузера;
width – ширина таблицы (в пикселях или %).

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ
УНИВЕРСИТЕТ»**

Факультет среднего профессионального образования

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ
ЛАБОРАТОРНЫХ РАБОТ ПО ДИСЦИПЛИНЕ «ИНФОРМАТИКА»**

Специальность 21.02.05 Земельно-имущественные отношения

Оренбург 2022 г.

СОДЕРЖАНИЕ

<i>Введение</i>	<i>Error! Bookmark not defined.</i>
<i>Раздел 1. Информация и информационные процессы</i>	<i>Error! Bookmark not defined.</i>
Тема 1.1. Представление информации в компьютере.....	Error! Bookmark not defined.
Тема 1.2. Способы измерения информации.....	11
Тема 1.3. Алгоритмы. Алгоритмические языки. Языки программирования.	Error! Bookmark not defined.
Тема 1.4. Логические, физические и арифметические основы вычислительной техники	
Логические законы и правила преобразования логических выражений	26
<i>Раздел 2. Информационная деятельность человека</i>	Error! Bookmark not defined.
Тема 2.1. Информационная деятельность человека.....	Error! Bookmark not defined.
<i>Раздел 3. Средства информационных и коммуникационных технологий</i>	32
Тема 3.1. Архитектура компьютеров.....	32
<i>Раздел 4. Технологии создания и преобразования информационных объектов</i>	34
Тема 4.1. Текстовый редактор (текстовый процессор MS Word)	34
Тема 4.2. Электронные таблицы (Табличный процессор MS Excel).....	38
Тема 4.3. Информационные системы. Базы данных MS Access	47
Тема 4.4. Мастер презентаций MS Power Point	54
<i>Раздел 5. Телекоммуникационные технологии</i>	60
Тема 5.1. Компьютерные сети. Средства телекоммуникационных технологий.	60
Тема 5.2. Язык HTML. Основы WEB-дизайна	60

Лабораторная работа 1.1

Тема: Представление информации в компьютере (**Деловая игра**)

Цели занятия:

- **Обучающая:** повторить алгоритмы перевода из одной системы счисления в другие: перевод в десятичную систему счисления, из десятичной системы счисления, перевод дробных и смешанных чисел, перевод целых и смешанных чисел из двоичной в восьмеричную, шестнадцатеричную систему счисления и обратно; подготовить учащихся к итоговой контрольной работе по теме «Системы счисления»

- **Развивающая:** развивать навыки реализации теоретических знаний в практической деятельности; познавательный интерес, речь и внимание учащихся, творческое и логическое мышление (посредством выполнения занимательных задач); коммуникативные способности при работе в группе, формировать умение самооценки.
- **Воспитательная:** воспитать уровень информационной культуры, мотивации учащихся путем использования нестандартных заданий; творческий подход к решению задач, четкость и организованность, умение оценивать свою деятельность и деятельность своих одноклассников; дух здорового соперничества, дружелюбного отношения друг к другу.

Тип урока: урок систематизации и обобщения изученного материала

Форма урока: индивидуальная, групповая, фронтальная

Метод урока: деловая игра

Оборудование: ПК, мультимедийный проектор, экран; презентация к уроку, раздаточный материал: жетончики с двоичными числами, решение заданий «Системы счисления» №1 – 6 (Приложение №1), оценочные листы (Приложение №2).

План урока:

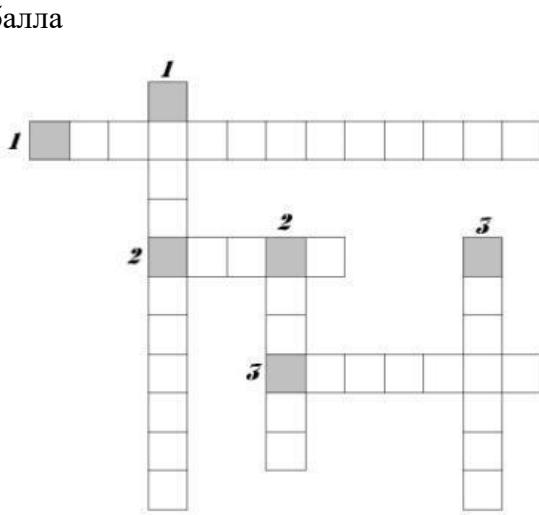
№	Этапы урока
I	Организационный момент
II	Повтор и закрепление изученного материала
III	Подведение итогов и рефлексия
IV	Домашнее задание

Ход урока

Этап урока	Деятельность учителя	Деятельность учащихся
I	<p>Здравствуйте ребята! Давайте проверим присутствующих.</p> <p>Эпиграф к уроку: «Все есть число» - говорили Пифагорийцы. Как вы думаете, почему я выбрала такой эпиграф к нашему уроку?</p>	<p>Приветствуют учителя и говорят, кто присутствует.</p> <p>Потому что мы изучаем тему «Системы счисления»; мы каждый</p>

	<p>день имеем дело с разными системами счисления: 60 – система счисления для измерения времени, 24 – количество часов в сутках, 7 – дни недели, 12 – месяцы, 2 – компьютерная система счисления, 10 – арабские цифры и т.д.; потому что нас окружает множество чисел...</p>
	<p>Цель урока: Сегодня на уроке мы повторим, обобщим и приведем в систему наши знания по теме «Системы счисления», но не в обычной форме, а в форме игры. Ваша задача – показать свои знания и умения по этой теме в ходе выполнения заданий. Прежде чем начать игру, предлагаю вам разбиться 2 команды по 6 человек, судьи – 2 человека, которые проверяют правильность выполнения заданий, начисляют баллы за конкурсы и делают записи в специальной таблице.</p> <p>При входе в кабинет каждый из вас получил номер в двоичной системе счисления, вам нужно, используя ваши знания, полученные на прошлых уроках, перевести числа в десятичную систему счисления и узнать, в какой группе вы будете работать. Учитель выводит слайд на экран с правильными ответами.</p>
II	<p>«Все есть число», говорили пифагорийцы, подчеркивая необычайно важную роль чисел в практической деятельности.</p> <p>А как умеют работать с числами участники сегодняшней игры? Сейчас нам это предстоит узнать.</p> <p>Итак, начнем работу. Задание 1. Задача 1.1. За правильный ответ 2 балла. Необходимо решить задачу: В бумагах одного чудака – математика,</p> <p>Учащиеся, переводят номера, используя алгоритм перевода в десятичное число. Учащиеся знакомятся с составом команд. Команды представляют свое название и девиз в течение 3 мин.</p> <p>Обучающие отвечают на вопрос задачи: недесятичная</p>

<p>найдена его биография. Она начиналась следующими удивительными словами: «Я окончил курс университета 44 лет от роду. Спустя год, 100-летним молодым человеком, я женился на 34-летней девушке. Незначительная разница в возрасте – всего 11 лет способствовала тому, что мы жили общими интересами и мечтами. Спустя немного лет у меня была уже и маленькая семья из 10 детей». Чем объяснить странные противоречия в числах этого отрывка? Восстановите их истинный смысл.</p>	<p>система счисления – вот единственная причина кажущейся противоречивости приведенных чисел. Основание этой системы определяется фразой: «спустя год (после 44 лет), 100-летним молодым человеком...». Если от прибавления одной единицы число 44 преображается в 100, то, значит, цифра 4 – наибольшая в этой системе (как 9 – в десятичной), а, следовательно, основанием системы является 5. Т. е. все числа в автобиографии записаны в пятеричной системе счисления.</p>
<p>44 -> 24, 100 -> 25, 34 -> 19, 11 -> 6, 10 -> 5</p>	
<p>Задача 1.2. За правильный ответ 2 балла. Решить задачу: Один мудрец писал «Мне 33 года, моей матери 124 года, а отцу 131 год. Вместе нам 343 года». Какую систему счисления использовал мудрец, и сколько ему лет?</p>	<p>Решение задачи 1.2:</p> $33_x + 124_x + 131_x = 343_x$ $3x + 3 + x^2 + 2x + 4 + x^2 + 3x + 1 =$ $3x^2 + 4x + 3$ $x^2 - 4x - 5 = 0 \quad x_1 = 5,$ $x_2 = -1 \quad (\text{не является решением})$
<p>Задача 1.3. За правильный ответ 2 балла. Решить задачу: Один человек имел 100 монет. Он поровну разделил их между двумя своими детьми. Каждому досталось по 11 монет и одна осталась лишней. Какая система счисления использовалась, и сколько было монет?</p>	<p>Ответ: $33_5 = 18$, $124_5 = 39$, $131_5 = 41$, $343_5 = 98$</p> <p>Решение:</p> $100_x = 11_x + 11_x + 1$ $x^2 - 2x - 3 = 0 \quad x_1 = 3, x_2 = -1 \quad (\text{не является решением})$ <p>Ответ: $100_3 = 9$, $11_3 = 4$</p>
	<p>По горизонтали:</p>

<p>балла</p>  <p>По горизонтали:</p> <ol style="list-style-type: none"> 1. Система, в которой количественное значение цифры не зависит от ее положения в числе (Непозиционная) 2. Символы, при помощи которых записывается число (цифра) 3. Самый яркий пример непозиционной системы счисления (римская) <p>По вертикали:</p> <ol style="list-style-type: none"> 1. Система, в которой количественное значение цифры зависит от ее положения в числе (позиционная) 2. Как называется позиция цифры в числе (разряд) 3. Совокупность различных цифр, используемых в позиционной системе счисления для записи чисел (алфавит) <p>Задание 3. Диктант. За все правильные ответы 7 баллов.</p> <p>Учитель: Если утверждение верно, ученики ставят 1, если неверно – 0</p> <ol style="list-style-type: none"> 1. Система счисления – это способ представления 	<p>1. Непозиционная 2. Цифра 3. Римская</p> <p>По вертикали:</p> <ol style="list-style-type: none"> 1. Позиционная 2. Разряд 3. Алфавит <p>Результат выполнения задания 3: 1001100</p>
--	---

	<p>чисел и соответствующие ему правила действий над числами.</p> <p>2. Информация, хранящаяся в компьютере, представлена в троичной системе счисления.</p> <p>3. В двоичной системе счисления $11 + 1 = 12$.</p> <p>4. Существует множество позиционных систем счисления, и они отличаются друг от друга алфавитами.</p> <p>5. В 16-ричной системе счисления символ F используется для обозначения числа 15.</p> <p>6. Римская система счисления – это позиционная система счисления.</p> <p>7. Непозиционной системы счисления – это система счисления, в которой значение цифры зависит от ее местоположения в числе.</p> <p>Задание 4. Решение неравенств. Задание оценивается 3-я баллами.</p> <p>Поставьте вместо знака ? знак <, > или =.</p> <p>$6C_{16} ? 101001_2$</p> <p>$55_{16} ? 125_8$</p> <p>$111111_2 ? 1111_8$</p> <p>Задание 5. Расположите числа, записанные в различных системах счисления, в порядке возрастания. Задание оценивается тремя баллами</p> <p>$35_{10}, 36_8, 100011, 01_2, 3C_{16}$</p> <p>Задание 6. Выполнить действия. Задание оценивается тремя баллами.</p> <p>$1011100 + 100101$</p> <p>$10011101 - 11110$</p> <p>$110101 \cdot 1011$</p>	<p>Учащиеся выполняют задание и получают результат. Ответ:</p> <p>$108_{10} > 41_{10}$</p> <p>Ответ: $85_{10} = 85_{10}$</p> <p>Ответ: $111111_2 < 1111_8$</p> <p>Задание 5. Ответ: $36_8, 35_{10}, 100011, 01_2, 3C_{16}$</p> <p>Задание 6. Ответ:=10000001 Ответ:=1111111 Ответ:=1111111</p>
III	Учащиеся, вам необходимо подсчитать количество	Учащиеся подсчитывают свои

<p>набранных вами баллов: максимальное количество баллов за урок – 25</p> <p>«5» - 24 – 25 баллов</p> <p>«4» - 18 – 23 баллов</p> <p>«3» -11 – 18 баллов</p> <p>Учитель: Пожалуйста, Оцените урок по 5-балльной системе (или свой вклад в результат команды), ответ запишите в двоичной системе счисления на обратной стороне листка с ответами. Оценка урока: (5 – 101, 4 – 100, 3 – 11, 2 – 10). Можно оценки за урок в двоичной системе счисления записать на доске (или в электронных таблицах), после выставления их учащимися подсчитать количество и дать среднюю оценку уроку.</p> <p>Вы сегодня работали хорошо, справились с поставленной перед вами задачей, а также показали хорошие знания по теме «Системы счисления». За работу на уроке вы получаете следующие оценки (объявляются оценки каждого ученика за работу на уроке).</p> <p>В конце урока хочу задать вам несколько вопросов. Итак, где же применяются системы счисления в жизни? Что нужно знать, чтобы перевести числа из одной системы счисления в другие?</p> <p>Рефлексия:</p> <ol style="list-style-type: none"> 1. Какое задание было самым интересным? 2. Какое задание, по вашему мнению, было самым сложным? 3. С какими трудностями вы столкнулись, выполняя задания? 4. Какие задания вы считаете самыми интересными, и какие задания можете 	<p>результаты и сравнивают со шкалой баллов.</p> <p>Оценивают урок по пятибалльной шкале и записывают в двоичной системе счисления.</p> <p>Обучающие слушают внимательно учителя, который объявляет результаты работы на уроке.</p> <p>Учащиеся внимательно слушают преподавателя и отвечают на поставленные вопросы и делятся своими впечатлениями от прошедшего урока.</p>
---	--

	предложить по данной теме?	
IV	<p>Для общения с компьютером нужна двоичная (восьмеричная, шестнадцатеричная) система счисления. В каких (кроме компьютера) приборах (и не только) применяется двоичная система счисления? Оправдано ли это применение (приведите аргументы в защиту).</p> <p>Решить задачу: Один шестиклассник о себе написал так: «Пальцев у меня 24, на каждой руке 5, а на ногах 12». Как это могло быть? (Ответ: восьмеричная система счисления)</p> <p>Выполнить арифметические действия:</p> <p>а) $1011100 + 100101$ $(= 10000001)$</p> <p>б) $10011101 - 11110$ $(= 1111111)$</p> <p>в) $110101 \cdot 1011$ $(= 1001000111)$</p>	Учащиеся записывают домашнее задание в тетрадь.

Приложение 1

Задание 1.1. (За правильный ответ **2 балла**).

Решить задачу: В бумагах одного чудака – математика найдена его биография. Она начиналась следующими удивительными словами: «Я окончил курс университета 44 лет от роду. Спустя год, 100-летним молодым человеком, я женился на 34-летней девушке. Незначительная разница в возрасте - всего 11 лет способствовала тому, что мы жили общими интересами и мечтами. Спустя немного лет у меня была уже и маленькая семья из 10 детей». Чем объяснить странные противоречия в числах этого отрывка? Восстановите их истинный смысл.

Задание 1.2. (За правильный ответ **2 балла**).

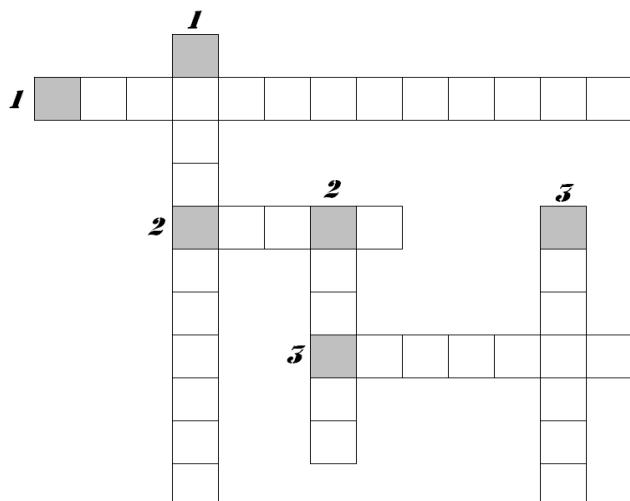
Решить задачу: Один мудрец писал «мне 33 года, моей матери 124 года, а отцу 131 год. Вместе нам 343 года». Какую систему счисления использовал мудрец, и сколько ему лет.

Задание 1.3. (За правильный ответ **2 балла**).

Решить задачу: Один человек имел 100 монет. Он поровну разделил их между двумя своими детьми. Каждому досталось по 11 монет и одна осталась лишней. Какая система счисления использовалась, и сколько было монет?

Задание 2. (За правильный ответ **3 балла**).

понятия



По

Кроссворд «Основные
системы счисления»

горизонтали:

1. Система, в которой количественное значение цифры не зависит от ее положения в числе
2. Символы, при помощи которых записывается число
3. Самый яркий пример непозиционной системы счисления

По вертикали:

1. Система, в которой количественное значение цифры зависит от ее положения в числе
2. Как называется позиция цифры в числе
3. Совокупность различных цифр, используемых в позиционной системе счисления для записи чисел

Задание 3. (За все правильные ответы **7 баллов**).

Если утверждение верно, ученики ставят 1, если неверно – 0 . Диктант:

1. Система счисления – это способ представления чисел и соответствующие ему правила действий над числами.
2. Информация, хранящаяся в компьютере, представлена в троичной системе счисления.
3. В двоичной системе счисления $11 + 1 = 12$.
4. Существует множество позиционных систем счисления, и они отличаются друг от друга алфавитами.
5. В 16-ричной системе счисления символ F используется для обозначения числа 15.
6. Римская система счисления – это позиционная система счисления.

7. Непозиционной системой счисления – это система счисления, в которой значение цифры зависит от ее местоположения в числе.

Задание 4. Решение неравенств. (Задание оценивается **3-мя баллами**)

Поставьте вместо знака ? знак $<$, $>$ или $=$.

1. $6C_{16}$? 101001_2
2. 55_{16} ? 125_8
3. 111111_2 ? 1111_8

Задание 5.

Расположите числа, записанные в различных системах счисления, в порядке возрастания.

(Задание оценивается **3-мя баллами**)

35_{10} , 36_8 , 100011_2 , 01_2 , $3C_{16}$

Задание 6.

Выполнить действия. (Задание оценивается **3-мя баллами**):

$$1011100 + 100101$$

$$10011101 - 11110$$

$$110101 * 1011$$

Приложение 2

Пожалуйста, Оцените урок по 5-балльной системе (или свой вклад в результат команды), ответ запишите в 2-й СС на обратной стороне листка с ответами.

Оценка урока:

--	--	--

Лабораторная работа 1.2.

Тема: Способы измерения информации

Цели занятия: научиться определять способ решения задачи и применять соответствующие формулы для ее решения.

Теоретическая часть.

1. **ЭНТРОПИЙНЫЙ** — общая мера неопределенности (энтропия) характеризуется некоторой математической зависимостью от между количеством информации и количеством возможных событий. Количество информации в сообщении определяется тем, насколько уменьшается эта мера неопределенности после получения сообщения. Существует формула, которая связывает между собой количество возможных событий N и количество информации I :

По этой формуле можно легко определить количество возможных событий, если известно количество информации.

$$N = 2^I$$

Задача №1: Мы получили 4 бита информации, тогда количество возможных событий составляет:

Дано:	$N = 2^I$
I= 4 бита	$N = 2^4 = 16$
Найти: N	

Задача №2: В игре в «Крестики — нолики» на поле 8x8 было получено сообщение о первом ходе. Какое количество информации несет это сообщение?

Дано:	$N = 2^I$
N = 8x8=64	$64 = 2^I$
Найти: I	I=6 бит

Задача №3 Сколько информации содержит сообщение о том, что из колоды карт достали король пик? В колоде 32 карты. В перемешанной колоде выпадение любой карты равновероятное событие

Дано:	$N = 2^I$
N=32	$32 = 2^I$
Найти: I	I=5 бит

Задача №3 Сколько информации содержит сообщение о выпадении грани с числом 3 на шестигранном игральном кубике?

Дано:	$N = 2^I$
N=6	$6 = 2^I$
Найти: I	$2^2 < 6 < 2^3$ I = 2,5 бит

Решите задачи:

1. Сколько информации содержит сообщение о том, что на поле 4x4 клетки одна из клеток закрашена?
2. В книге 512 страниц. Сколько информации несет сообщение о том, что закладка лежит на какой-либо странице?

Теоретическая часть.

2. **АЛФАВИТНЫЙ ПОДХОД** к определению количества информации заключается в том, что каждый символ сообщения несет некоторое количество битов информации. Количество информации в сообщении можно подсчитать, умножив количество информации, которое несет один символ, на количество символов в сообщении. Для определения используется следующая формула:

$$I_I = K * I, \text{ где}$$

I_I — информационный объем всего сообщения;

K — количество символов в сообщении;

I — информационный объем одного символа.

Задача №1: Считая, что каждый символ кодируется одним байтом, определите, чему равен информационный объем следующего высказывания Рене Декарта (знаки препинания и пробелы учитываются): «Я мыслю, следовательно, существую.»

$$\begin{array}{l} \text{Дано:} \\ I = 1 \text{ байт} = 8 \text{ бит} \\ K = 34 \\ \hline \text{Найти: } I_I \end{array}$$

$$\begin{array}{l} I_I = K * I \\ I_I = 34 * 8 = 272 \text{ бита} \end{array}$$

При алфавитном подходе также учитывается количество символов в алфавите, с помощью которого создано сообщение. Количество символов в алфавите (мощность алфавита) обозначается N и определяется по формуле:

$$N = 2^I \text{ где:}$$

N — мощность алфавита;
 I — информационный объем одного символа.

Задача №2: в 2-символьном алфавите каждый символ несет 1 бит информации ($2^x = 2$, откуда $x = 1$ бит).

Если $N=16$, то каждый символ несет 4 бита информации, т.к. $2^4 = 16$.

Если $N=32$, то один символ несет 5 бит информации.

При $N=64$, один символ «весит» 6 бит и т.д.

Задача №3: Племя “Обезьяны” пишет письма, пользуясь 32-символьным алфавитом. Племя “Слоны” пользуется 64-символьным алфавитом. Вожди племен обменялись письмами. Письмо племени “Обезьяны” содержало 90 символов, а письмо племени “Слоны” – 80 символов. Сравните объем информации, содержащейся в письмах.

$$\begin{array}{l} \text{Дано:} \\ N_0 = 32 \\ N_c = 64 \\ K_0 = 90 \\ K_c = 80 \\ \hline \text{Найти: } I_{I_0}, I_{I_c} \end{array}$$

$$\begin{array}{l} I_{I_0} = K * I \\ N_0 = 2^{I_0} \\ 32 = 2^{I_0} \\ I_0 = 5 \\ I_{I_0} = 90 * 5 = 450 \text{ бит} \end{array}$$

$$\begin{array}{l} I_{I_c} = K * I \\ N_c = 2^{I_c} \\ 64 = 2^{I_c} \\ I_c = 6 \\ I_{I_c} = 80 * 6 = 480 \text{ бит} \\ I_{I_c} > I_{I_0} \end{array}$$

ОТВЕТ: объем информации в письме вождя племени “Слоны” больше объема информации, которую передал в письме вождь племени “Обезьяны”.

Есть алфавит, который можно назвать достаточным. Это алфавит мощностью 256 символов. Алфавит из 256 символов используется для представления текстов в компьютере. В этом алфавите можно поместить практически все необходимые символы: латинские и русские буквы, цифры, знаки арифметических операций, скобки, знаки препинания, знаки псевдографики. Поскольку $256 = 2^8$, то один символ этого алфавита «весит» 8 бит.

8 бит информации присвоили свое название – **байт**.

Байт – поле из 8 последовательных бит. Байт широко используется как единица измерения количества информации.

1 байт = 8 бит

Компьютерные текстовые редакторы работают с алфавитом мощности 256 символов. Поскольку в настоящее время при подготовке книг используются текстовые редакторы, легко посчитать объем информации в тексте.

Если один символ алфавита несет 1 байт информации, то надо просто сосчитать число символов, полученное значение даст информационный объем текста в байтах.

В любой системе единиц измерения существуют основные единицы и производные от них.

Для измерения больших объемов информации используются производные от байта единицы:

$$1 \text{ килобайт} = 1 \text{ Кб} = 2^{10} \text{ байт} = 1024 \text{ байта}$$

$$1 \text{ мегабайт} = 1 \text{ Мб} = 2^{10} \text{ Кб} = 1024 \text{ Кб} = 1048576 \text{ байт}$$

$$1 \text{ гигабайт} = 1 \text{ Гб} = 2^{10} \text{ Мб} = 1024 \text{ Мб} = 1048576 \text{ Кб} = 1073741824 \text{ байт}$$

Задача №4: Объем сообщения — 7,5 Кбайт. Известно, что данное сообщение содержит 7680 символов. Какова мощность алфавита?

Дано:

$$I_l = 7,5 * 1024 * 8 = 61440 \text{ бит}$$

$$K = 7680$$

Найти: N

$$N = 2^I$$

$$I_l = K * I \Rightarrow I = I_l / K = 61440 / 7680 = 8$$

$$N = 2^8 = 256$$

Ответ: 256 символов

Задача №5: Книга, набранная с использованием текстового редактора, содержит 70 страниц, на каждой странице 38 строк, в каждой строке 56 символов. Определить объем информации, содержащейся в книге. Ответ представить в Мбайтах

Дано:

$$N = 256$$

$$K = 70 * 38 * 56 = 148960$$

Найти: I_l

$$I_l = K * I$$

$$N = 2^I \Rightarrow 256 = 2^I \Rightarrow I = 8$$

$$I_l = 148960 * 8 = 1191680 \text{ бита} = 1191680 / 8 / 1048576 = 0,142$$

Мбайт

Ответ: 0,142 Мб

Теоретическая часть.

3. ВЕРОЯТНОСТНЫЙ ПОДХОД

Следует различать два вида вероятностных подхода:

- Равновероятностный;
- Неравновероятностный.

Для частных, но широко распространенных случаев, когда события равновероятны, используются следующие формулы:

$$I = I_0 q_2 N \Rightarrow N = 2^I, \quad \text{где}$$

I — количество информации о событии;

N — количество возможных событий

$$P_I = K/N \quad \text{где}$$

P_I — вероятность равномерного события;

K — количество равномерных событий;

Для неравномерных событий, когда события неравновероятны, используются следующие формулы:

$$I = \log_2(1/P_1) \Rightarrow (1/P_1) = 2^I, \quad \text{где}$$

P₁ — вероятность неравномерного события;

$$P_1 = K/N \quad \text{где}$$

P₁ — вероятность равномерного события;

K — величина, показывающая сколько раз произошло интересующее нас событие;

Задача №1: В мешке находится 20 шаров, из них 15 белых и 5 красных. Какое количество информации несет сообщение о том, что достали

- A) белый шар;
- B) красный шар.

Дано:

N = 20

K_б = 15

K_к = 5

Найти: **I_б, I_к**,

$$P_{I_6} = K/N = 15/20 = 3/4$$

$$I_6 = \log_2(1/P_1) = \log_2(4/3)$$

$$I_6 = 0,477$$

$$P_{I_k} = K/N = 5/20 = 1/4$$

$$I_k = \log_2(1/P_1) = \log_2(4) \Rightarrow$$

$$4 = 2^{I_k} \Rightarrow I_k = 2$$

Дано:

N = 64

I = 4 бита

Найти: **K_б**

$$I_6 = \log_2(1/P_1) \Rightarrow 1/P_1 = 2^{I_6} = 2^4 = 16 \Rightarrow P_1 = 1/16$$

$$P_{I_6} = K/N \Rightarrow K_6 = P_{I_6} * N = 1/16 * 64 = 64/16 = 4$$

$$K_6 = 4$$

Ответ: 4 белых карандаша

Лабораторная работа 1.3.1.

Тема: Алгоритмы. Алгоритмические языки программирования

Цели занятия:

- познакомиться с понятием «алгоритм»;
- научиться создавать новый документ в среде построителя алгоритмов; создавать простейшую программу, выводящую одну или несколько строк, запускать программу и просматривать результат работы.

Теоретическая часть.

Алгоритм — это точное предписание, которое определяет процесс, ведущий от исходных данных к требуемому конечному результату.

Таким образом, **Программа для ЭВМ** представляет собой описание алгоритма и данных на некотором языке, понятном ЭВМ и полученным с помощью языка программирования. Программа при этом предназначена для последующего автоматического выполнения.

После решения задачи обработки информации результат должен быть выдан конечным пользователям в требуемом виде. Эта операция реализуется в ходе решения задачи выдачи информации. Выдача информации, как правило, производится с помощью внешних устройств ЭВМ в виде текстов, таблиц, графиков.

Существует несколько форм представления алгоритмов:

- На естественном языке (словесная форма).
- На языке блок-схем.
- На алгоритмическом языке – программа.

В повседневной жизни алгоритм часто записывается в виде предложений, расположенных в порядке выполнения. Запись алгоритма с помощью слов называется **словесным представлением алгоритма**.

Если вы внимательно оглянитесь вокруг, то обнаружите множество алгоритмов которые мы с вами постоянно выполняем. Мир алгоритмов очень разнообразен. Несмотря на это, удается выделить общие свойства, которыми обладает любой алгоритм.

Виды алгоритмов:

1. **Линейный алгоритм** (описание действий, которые выполняются однократно в заданном порядке);
2. **Циклический алгоритм** (описание действий, которые должны повторяться указанное число раз или пока не выполнено задание);
3. **Разветвляющий алгоритм** (алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий).

Для более наглядного представления алгоритма широко используется графическая форма – блок-схема, которая составляется из стандартных графических объектов.. Каждое графически обозначенное предложение алгоритма называется блоком. В блок записывается только одна команда. Блоки (шаги) алгоритма соединены стрелочками.

Вид стандартного графического объекта	Назначение
	Начало и конец алгоритма
	Ввод/вывод данных
	Выполняемое действие записывается внутри прямоугольника
	Условие выполнения действий записывается внутри ромба
	Счетчик кол-во повторов
	Последовательность выполнения действий

Примеры записи алгоритмов в виде блок-схемы:

Линейный алгоритм.

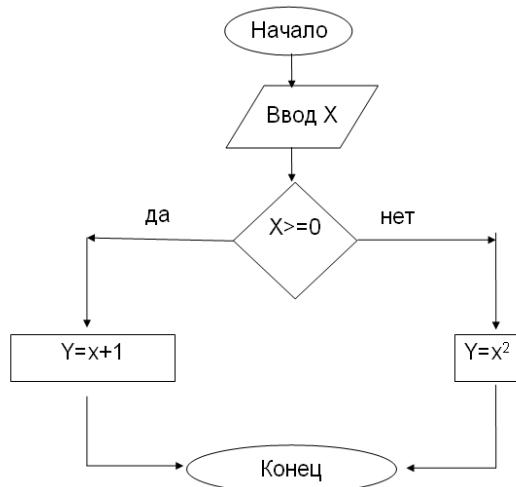
Вычислить площадь прямоугольника со сторонами A, B.



Разветвляющий алгоритм.

Вычислить:

$$Y = \begin{cases} X+1, & \text{если } X \geq 0 \\ X^2, & \text{если } X < 0 \end{cases}$$



Циклический алгоритм.

Стадии создания алгоритма:

1. Алгоритм должен быть представлен в форме, понятной человеку, который его разрабатывает.

2. Алгоритм должен быть представлен в форме, понятной тому объекту (в том числе и человеку), который будет выполнять описанные в алгоритме действия.

Объект, который будет выполнять алгоритм, обычно называют исполнителем.

Исполнитель – объект, который выполняет алгоритм.

Идеальными исполнителями являются машины, роботы, компьютеры...

Компьютер – автоматический исполнитель алгоритмов.

Алгоритм, записанный на “понятном” компьютеру языке программирования, называется **программой**.

В качестве примера использования ветвлений рассмотрим составление алгоритма для вычисления функции f в зависимости от конкретных значений x, a, b :

$$f = \begin{cases} x + a, & \text{при } x \leq 2.5; \\ x - a \cdot b, & \text{при } 2.5 < x < 10; \\ x - b, & \text{при } x \geq 10. \end{cases}$$

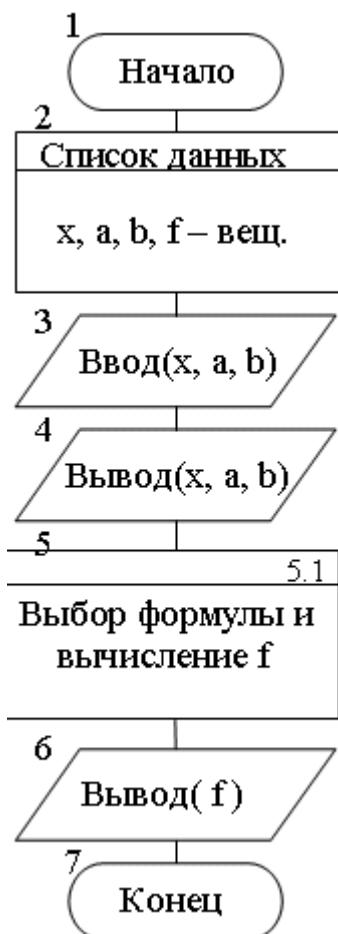
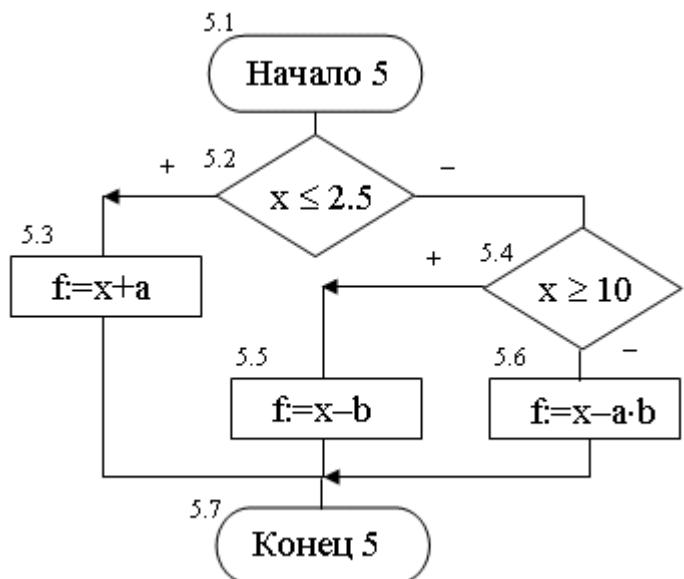


Рис. 10. Первый этап составления алгоритма.

Первое приближение алгоритма будет иметь вид, показанный на рис. 10. Анализ этого алгоритма показывает, что все его блоки, кроме блока 5, являются элементарными. Возможный вариант дальнейшей детализации блока 5 представлен на рис. 11. Все блоки второго этапа детализации являются элементарными, поэтому составление алгоритма практически закончено.

На заключительном этапе производится сборка алгоритма, т.е. содержимое рис. 11 помещается на рис. 10 вместо блока 5.



Лабораторная работа 1.3.2.

Тема: Знакомство с языком программирования Pascal

Цели и задачи занятия:

- познакомиться с алфавитом языка Pascal; правилами записи имен;
- научиться создавать новый документ в среде Pascal; создавать простейшую программу, выводящую одну или несколько строк, запускать программу и просматривать результат работы.

Теоретическая часть.

Окно интегрированной среды разработки состоит из трех частей:

- строка меню;
- строка состояния;
- рабочая область.

Структура программы:

Program имя программы;
uses раздел описания модулей;
const раздел описания констант;
var раздел описания переменных;
procedure (function) раздел описания подпрограмм;
begin
раздел операторов;

end.

Оператор – команда алгоритма, записанная на ЯП

Комментарий в программе можно записывать в { } на любом языке. Все, что записано в { } компилятором не воспринимается.

Имена заголовка программы, переменных составляются по следующим правилам:

- имя должно начинаться с буквы, может содержать латинские буквы, цифры и знак _
- длина имени должна быть меньше 255 символов, но значащиеся только первые 63 символа;
- прописные буквы (A...Z) тождественны строчным (a...z).

Например: book, KNIGA=kniga, nom_1, A2

Неверные имена: 2A, Vip*bum

Пример простой программы.

Составим программу, выводящую на экран строку приветствия.

Program pr1;	- имя программы
begin	- начало программы
writeln ('Привет!');	-оператор вывода на экран сообщения
end.	- конец программы.

writeln ('Привет!'); - оператор вывода на экран сообщения (то, что заключено в символах ' - апостроф, выводится на экран без изменений)

Запишите программу, запустите командой **Run - Run**

Откройте окно просмотра результатов командой **Debug - Output**

Просмотрите результат

Практическая часть.

Задание 1. Найди ошибки в записи программ:

programm 4A PROGRAM summa chisel
BEGIN ; начало
Вася {программа}
{Петя} конец.
end.

program 1
begin;
writeln(Мы создаем свою первую
программу!);
writeln("А вы нашли все ошибки?");
end

Задание 2. Определите, что будет выведено на экран в результате выполнения следующей программы?

```
program z1;
begin
writeln('Сегодня на улице хорошая погода');
writeln('Светит солнце!');
{И нет дождя!};
end.
```

Задание 3. Составьте программу, выводящую на экран сегодняшнюю дату в рамке:

* 15.09.2010 *

Вопросы для повторения

1. Из каких частей состоит окно разработки программы Pascal?
2. Какими командами меню можно создать новое окно, открыть имеющееся, сохранить программу?

3. Какими командами меню можно скопировать содержимое в буфер обмена, вставить из буфера?
4. Какой командой меню можно запустить программу на выполнение?
5. Какой командой меню можно открыть окно просмотра результатов?
6. Из каких блоков может состоять программа на Pascal?
7. По каким правилам составляются имена заголовков программы и переменных?
8. Какой существует оператор вывода на экран?

Лабораторная работа 1.3.3.

Тема: Операторы ввода и вывода данных. Оператор присваивания

Цели и задачи занятия:

- познакомиться с понятиями *оператор ввода, вывода, присваивания, переменная, типы данных*;
- научиться записывать операторы на языке Pascal, составлять простейшие программы на простой диалог и вычисления.

Теоретическая часть.

Представьте простую задачу. Подсчитать сумму двух чисел.

Если числа небольшие, то можно устно или на калькуляторе, а если числа очень большие, то требуется подумать. Но ЭВМ «думает» намного быстрее и безошибочно. Программа может применяться для любых чисел.

Возьмем две независимые переменные

a, b – исходные данные

c – результат

Например: 10+8=18 (целые)

-3+1000=997

9,3+8,1=17,4 (действительные числа)

Чтобы правильно работала программа, нужно выбрать тип переменных.

Типы переменных (числовых)

Основные типы данных:

- Целый integer - 32768 ... 32767
- Длинный целый longint -2147483648 ... 2147483647
- Вещественный real 2,9*10⁻³⁹ ... 1,7*10³⁸

Дополнительные типы данных:

Форматы целого типа:

Название типа	Длина, байт	Диапазон значений
byte	1	0..255
shortInt	1	-128..+127
word	2	0..65535
integer	2	-32768..32767
longint	4	-2147483648..+2147483647

Форматы вещественного типа:

Название типа	Длина, байт	Количество значащих цифр	Диапазон десятичного порядка
real	6	11..12	-39..+38
double	8	15..16	-324..+308

extended	10	19..20	-4951..+4932
comp	8	19..20	-263+1..+263-1

Описание переменных происходит в разделе описания переменных:

var переменная: тип;

Например: **var** x: integer;

var a, b, c: real;

var a, b: integer;

Чтобы компьютер производил вычисления, необходимо воспользоваться оператором присваивания.

Оператор присваивания

Имя переменной := выражение;

Например, $c:=a+b$

Сначала вычисляется выражение, затем полученной выражение присваивается переменной.

Чтобы компьютер знал над какими числами производить вычисления, необходимо ввести с клавиатуры значения переменных.

Оператор ввода

read () – оператор ввода значений переменных (читать)

readln () – оператор ввода значений переменных с переводом курсора на следующую строку (читать строку)

Например, **read (a)** или **read (a, b)**

read (a, b) – программа будет ждать, пока не введете значение переменных **a** и **b** через пробел

readln (a, b) – считывает значение **a** и **b**, затем переведет курсор на следующую строку.

Оператор вывода

write () – оператор вывода значений переменных или фразы (писать)

writeln () – оператор вывода значений переменных или фразы с переводом курсора на следующую строку (писать в строку)

Например, **write (c)** – выведет на экран значение переменной **c**

или **writeln (c)** – выведет на экран значение переменной и перейдет на следующую строку

write (a, b, c) – выведет значения переменных в строку без знаков препинания через позицию табуляции

write ('мама') – выведет слово, заключенное в апострофах

write ('сумма равна ', c) – выведет, например, **сумма равна 13**

writeln (a, b, c) – выведет значения переменных в строку без знаков препинания через позицию табуляции и переведет курсор на следующую строку.

Запишем теперь полностью программу на Pascal.

Program summa;

var a, b, c: integer;

begin

writeln ('введите слагаемые');

readln (a, b);

c:=a+b;

writeln ('сумма равна ', c);

end.

Арифметические выражения содержат числовые константы и переменные, знаки арифметических действий, скобки, стандартные функции.

Практическая часть.

Запустите программу

Исправьте программу следующим образом, запустите и найдите отличия:

```
Program summa;
var a, b, c: integer;
begin
  write ('a='); readln (a);
  write ('b='); readln (b);
  c:=a+b;
  writeln ('a+b= ',c);
end.
```

Исправьте в программе для переменных тип данных на вещественный, запустите и введите дробные значения:

Число **1.6500000000E+01** - это стандартная запись числа **16,5**, которую можно записать в виде **1,65*10¹**

Исправьте программу:

Запись **c:6:2** означает, что значение переменной с будет выведено в шесть позиций, из них в две после запятой.

Добавьте к программе операторы, позволяющие производить основные математические операции (вычитание, умножение, деление).

Задание 1. Определите количество строк, которое будет выведено на экран, в результате выполнения следующего фрагмента программы:

```
Writeln('Здесь первая строка!');
Write('А вот это, видимо - вторая');
Write('Третья строка - самая лучшая');
```

Задание 2. Определите, к каким типам данных относятся следующие числа:

23; 79; -128; -129; 1.99; -65899

Задание 3. Исправьте ошибки в приведенном фрагменте программного кода:

```
program z1;
var a,b:integer;
  c:real;
begin
  a:=10;
  b:=1,7;
  c:=a+b;
end.
```

Вопросы для повторения

1. Какие форматы целого типа данных существуют в Pascal?
2. Какие форматы вещественного типа данных существуют в Pascal?
3. В каком разделе описываются переменные?
4. Какие операторы ввода данных существуют в Pascal? Чем они отличаются?
5. Какие операторы вывода данных существуют в Pascal? Чем они отличаются?
6. Для чего необходим оператор присваивания?
7. Что означает запись **c:6:2** для переменной вещественного типа?

Лабораторная работа 1.3.4.

Тема: Арифметические выражения. Стандартные функции Pascal

Цели и задачи занятия:

- познакомиться со стандартными функциями Pascal, научиться их использовать при составлении программ,
- закрепить навыки составления линейных программ при решении задач.

Теоретическая часть.

Арифметические выражения содержат числовые константы и переменные, знаки арифметических действий, скобки, стандартные функции.

Стандартные функции

В Pascal	В математике	Тип результата
abs(x)	$ x $	целое, вещественное
sqr(x)	x^2	целое, вещественное
sgrt(x)	\sqrt{x}	вещественное
pi	$\pi=3.14$	вещественное
sin(x)	sin x, где x - в радианах	вещественное
cos(x)	cos x, где x - в радианах	вещественное
arctan(x)	arctn x, где x - в радианах	вещественное
exp(x)	e^x	вещественное
ln(x)	ln x	вещественное
int(x)	целая часть x	вещественное
frac(x)	дробная часть x	вещественное
trunc(x)	отсечение дробной части числа	целое
round(x)	округление до ближайшего целого	целое
A div B	деление A на B с отбрасыванием остатка	целое
A mod B	остаток от целого деления A на B	целое

Например,
int(-8,9) = -8,0
frac(3,6) = 0,6
round(-7,3) = -7
trunc(3,6) = 3
7 div 2 = 3
4 div 3 = 1
12 div 3 = 4
25 mod 7 = 4
17 mod 3 = 2
8 mod 2 = 0

Правила записи арифметического выражения:

- все символы пишутся в одну строку
- нельзя ставить 2 знака подряд
- обязательно указывать знак *

Приоритет операций (по убыванию старшинства)

- стандартные функции
- *, /, div, mod
- +, -

Например, запишем следующие математические выражения на языке Pascal:

$$x^2 - 7x + 6 \quad sgr(x) - 7 * x + 6$$

$\frac{ x - y }{1 + xy }$	(abs(x)-abs(y))/(1+abs(x*y))
------------------------------	------------------------------

Задача 1. Составим программу, вычисляющую площадь треугольника по формуле Герона.

a, b, c – стороны (исходные данные)

P – полупериметр (вспомогательная переменная)

S – площадь (результат)

Формулы:

$$P=1/2*(a+b+c)$$

$$S = \sqrt{p(p-a)(p-b)(p-c)}$$

```
Program treug;
var a, b, c : integer;
          P, S: real
begin
writeln ('введите стороны треугольника');
readln (a, b, c);
P=1/2*(a+b+c);
S:=sqrt(p*(p-a)*(p-b)*(p-c));
writeln ('площадь равна', S:6:2);
end.
```

Задача 2. Составить программу, которая преобразует введенное с клавиатуры дробное число в денежный формат.

Например, число 12,5 должно быть преобразовано к виду: 12 руб. 50 коп.

a – исходные данные

x, y – результаты

$$\text{trunc}(12,5)=12$$

$$\text{frac}(12,5)=0,5$$

$$0,5*100=50,0$$

$$\text{trunc}(50,0)=50$$

```
Program rub;
var a: real;
          x, y: integer;
begin
writeln ('введите дробное число');
readln (a);
x:=trunc(a);
y:=trunc(frac(a)*100);
writeln ('Ответ: ', x, ' руб. ', y, ' коп. ');
end.
```

Задача 3. Составить программу, вычисляющую сумму цифр трехзначного числа n.

n – трехзначное число (например, 762) – исходные данные

a, b, c – цифры этого числа (вспомогательные переменные)

S – сумма цифр этого числа (результат)

```
Program chislo;
var n, a, b, c, s : integer;
begin
```

```
writeln ('введите трехзначное число');
readln (n);
c:=n mod 10;
a:=n div 100;
b:=(n div 10) mod 10;
s:=a+b+c;
writeln ('сумма цифр числа равна ', s);
end.
```

Практическая часть.

Задание 1. Запишите математическую функцию в виде функции на Pascal:

- $\ln |(y - \sqrt{x})|$
- $x^y = e^{y \ln x}$
- $\left| x - \sqrt{\frac{y}{z + x^2/4}} \right|$

Задание 2. Составьте программу, вычисляющую расстояние между двумя точками с данными координатами.

$x_1, y_1; x_2, y_2$ – координаты точек (исходные данные)

d – расстояние (результат)

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Задание 3. Составьте программу, вычисляющую площадь треугольника, две стороны которого равны a и b , а угол между ними равен g (в градусах) ($0 < g < 180$)

a, b – стороны

g – угол в градусах (исходные данные)

S – площадь (результат)

x – угол в радианах (вспомогательная переменная)

$$S=1/2*a*b*sin x$$

$$x=p/180*g$$

Задание 4. Дано четырехзначное число n . Выписать число в обратном порядке.

n – четырехзначное число (например 1762) – исходные данные

a, b, c, d – цифры этого числа (вспомогательные переменные)

m – обратное число (результат).

Вопросы для повторения

1. Какие существуют стандартные функции Pascal?
2. Результат каких функций может быть только вещественное число? Целое число?
3. Перечислите правила записи арифметического выражения.
4. Перечислите приоритет основных операций.

Лабораторная работа 1.4

Тема: Логические и арифметические основы вычислительной техники

Цели занятия:

- познакомиться со стандартными логическими функциями;
- научиться их использовать при решении задач,

Теоретическая часть

Составные высказывания на естественном языке образуются с помощью союзов, которые в алгебре высказываний заменяются на логические операции.

Логическая операция КОНЬЮНКЦИЯ (логическое умножение):

- в естественном языке соответствует союзу **и**;
- в алгебре высказываний обозначение **&**;
- в языках программирования обозначение **And**.

Конъюнкция - это логическая операция, ставящая в соответствие каждым двум простым высказываниям составное высказывание, являющееся истинным тогда и только тогда, когда оба исходных высказывания истинны.

В алгебре множеств конъюнкции соответствует операция *пересечения множеств*, т.е. множеству получившемуся в результате умножения множеств A и B соответствует множество, состоящее из элементов, принадлежащих одновременно двум множествам.

Логическая операция ДИЗЬЮНКЦИЯ (логическое сложение):

- в естественном языке соответствует союзу **или**;
- обозначение **□** ;
- в языках программирования обозначение **Or**.

Дизьюнкция - это логическая операция, которая каждым двум простым высказываниям ставит в соответствие составное высказывание, являющееся ложным тогда и только тогда, когда оба исходных высказывания ложны и истинным, когда хотя бы одно из двух образующих его высказываний истинно.

Логическая операция ИНВЕРСИЯ (отрицание):

- в естественном языке соответствует словам **неверно, что...** и частице **не**;
- обозначение \bar{A} ;
- в языках программирования обозначение **Not**;

Отрицание - это логическая операция, которая каждому простому высказыванию ставит в соответствие составное высказывание, заключающееся в том, что исходное высказывание отрицается.

В алгебре множеств логическому отрицанию соответствует операция *дополнения до универсального множества*, т.е. множеству получившемуся в результате отрицания множества A соответствует множество \bar{A} , дополняющее его до универсального множества.

Пример 3.5. Получить таблицы истинности операций логического умножения, логического сложения и отрицания с использованием электронных таблиц StarOffice Calc.

Электронные таблицы StarCalc обладают встроенными логическими функциями. Функция логического умножения И(ЛогЗнач1;ЛогЗнач2;...) дает значение TRUE (1), только тогда, когда все логические аргументы имеют значение TRUE (1).

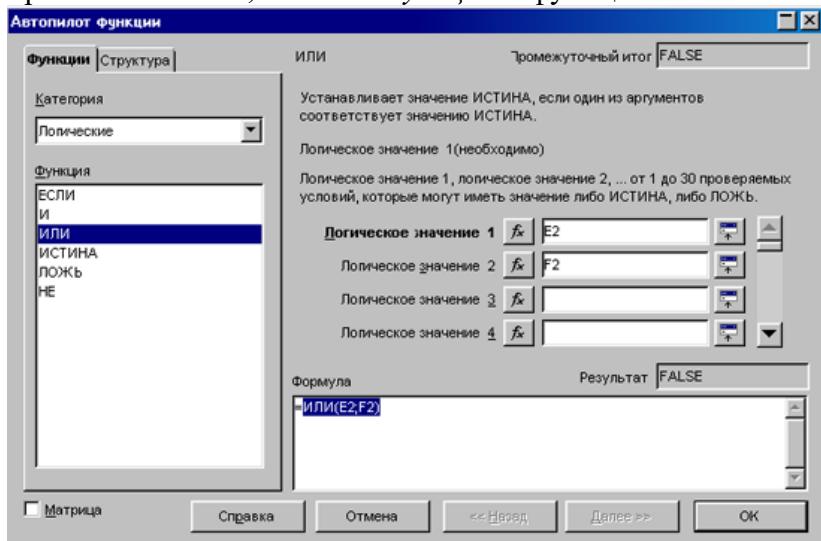
Функция логического сложения ИЛИ(ЛогЗнач1;ЛогЗнач2;...) дает значение TRUE (1), тогда, когда хотя бы один логический аргумент имеют значение TRUE (1).

Функция логического отрицания НЕ(ЛогЗнач) дает значение TRUE (1), когда логический аргумент имеют значение FALSE (0) и, наоборот, значение FALSE (0), когда логический аргумент имеют значение TRUE (1).

	Получение таблиц истинности операций логического умножения, логического сложения и отрицания с использованием электронных таблиц StarCalc.
1	Запустить интегрированное приложение StarOffice командой [Программы-StarOffice].
2	В появившемся окне приложения StarOffice запустить электронные таблицы StarCalc командой [Файл-Создать-Документ электронной таблицы].

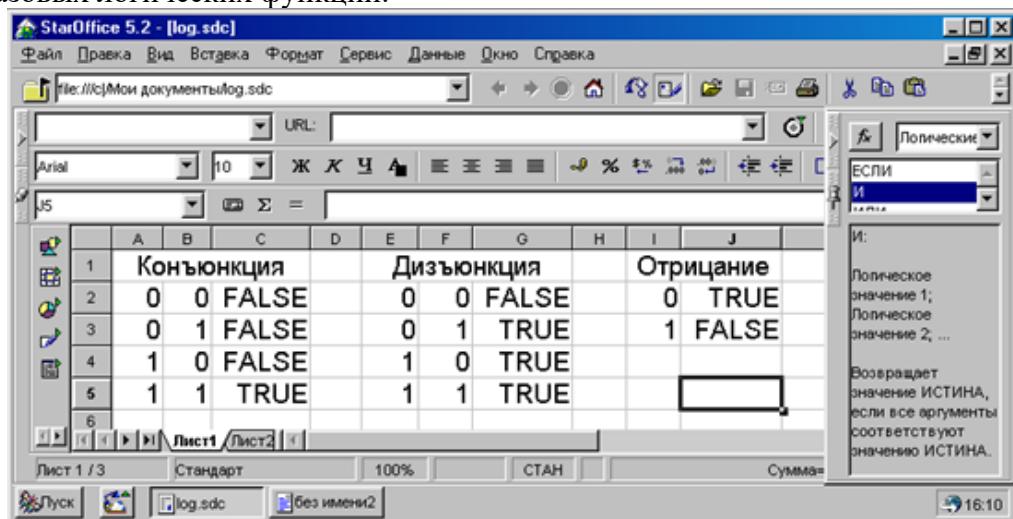
3 На листе *Лист1* создать таблицы истинности. Создать заголовки и ввести в столбцы *A* и *B*, *E* и *F*, *I* значения логических аргументов, а в столбцы *C*, *G*, *J* соответствующие логические функции.

4 Для ввода логических функций воспользоваться командой [Вставка-Функция]. На появившейся диалоговой панели *Автопилот функции* в раскрывающемся списке *Категории* выбрать *Логические*, а в окне *Функция* – функцию.



5 В окнах *Логическое значение 1* и т.д. можно видеть и выбирать местонахождение аргументов функции. В окне *Формула* можно видеть текущее состояние заданной функции, а в окне *Результат* – текущее состояние ее значения.

6 После ввода аргументов функций и их формул на листе появятся таблицы истинности трех базовых логических функций.



7 Переименовать лист *Лист1* в *Логические операции* и сохранить в файле [log.sdc](#).

Логическая операция ИМПЛИКАЦИЯ (логическое следование):

- в естественном языке соответствует обороту **если ..., то ...;**
- обозначение .

Импликация - это логическая операция, ставящая в соответствие каждым двум простым высказываниям составное высказывание, являющееся ложным тогда и только тогда, когда условие (первое высказывание) истинно, а следствие (второе высказывание) ложно.

A	B	A \square B
0	0	1
0	1	1
1	0	0

1	1	1
---	---	---

Логическая операция ЭКВИВАЛЕНЦИЯ (равнозначность):

- в естественном языке соответствует оборотам речи **тогда и только тогда; в том и только в том случае;**
- обозначения \square , \sim .

Эквиваленция – это логическая операция, ставящая в соответствие каждым двум простым высказываниям составное высказывание, являющееся истинным тогда и только тогда, когда оба исходных высказывания одновременно истинны или одновременно ложны.

Таблица истинности эквиваленции:

A	B	$A \square B$
0	0	1
0	1	0
1	0	0
1	1	1

Пример 3.6. Определите истинность составного высказывания: $(\bar{A} \& \bar{B}) \square (C \square D)$, состоящего из простых высказываний:

- $A = \{\text{Принтер} - \text{устройство вывода информации}\}$,
 $B = \{\text{Процессор} - \text{устройство хранения информации}\}$,
 $C = \{\text{Монитор} - \text{устройство вывода информации}\}$,
 $D = \{\text{Клавиатура} - \text{устройство обработки информации}\}$.

Сначала на основании знания устройства компьютера устанавливаем истинность простых высказываний: $A = 1, B = 0, C = 1, D = 0$.

Определим теперь истинность составного высказывания, используя таблицы истинности логических операций:

$$(\bar{1} \& \bar{0}) \square (1 \square 0) = (0 \& 1) \square (1 \square 0) = 0$$

Составное высказывание ложно.

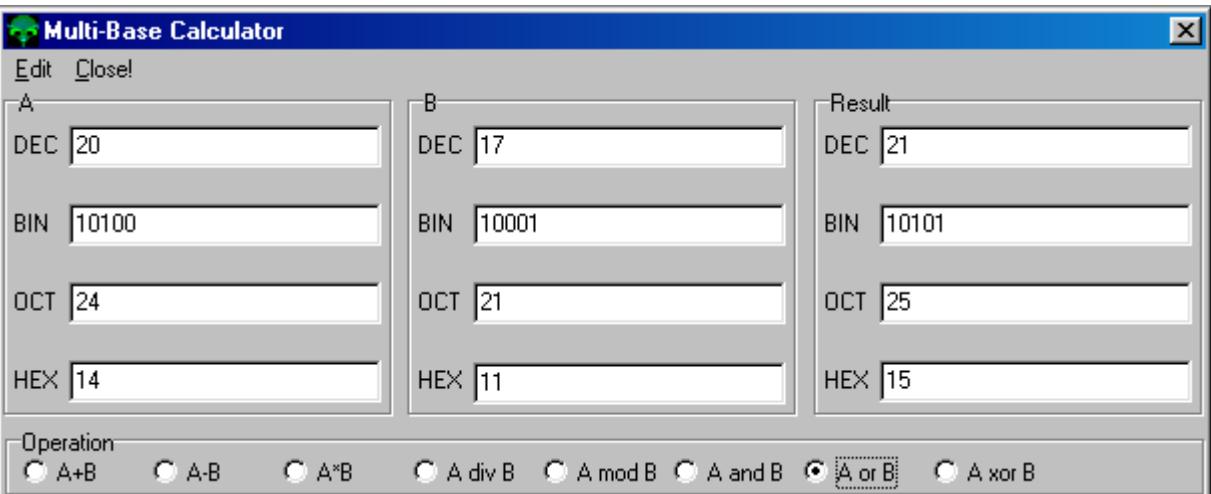
Проверим правильность полученного результата с использованием NumLock Calculator.

	Определение истинности составного логического высказывания с использованием NumLock Calculator.
1	Запустить программу NumLock Calculator.
2	Установить удобный вид калькулятора командой [Вид калькулятора-Универсальный].
3	Ввести составное логическое выражение, подставив в него значения простых логических выражений.
4	Нажать кнопку “=”, получим значение (0) составного логического выражения, которое совпадает с вычисленным алгебраически.

Пример 3.7. Даны три числа в различных системах счисления: $A = 20_{10}$, $B = 11_{16}$, $C = 30_8$. Переведите A , B и C в двоичную систему счисления и выполните поразрядно логические операции $(A \square B) \& C$. Ответ дайте в десятичной системе счисления.

	Арифметические и логические операции с использованием Wise Calculator.
1	Запустить программу Wise Calculator.
2	Запустить [Tools-Multi-Base Calculator].
3	Ввести значение логического аргумента A в текстовое поле <i>Dec</i> и аргумента B в текстовое поле <i>Hex</i> и с помощью группы переключателей <i>Operation</i> выбрать оператор

логического сложения $A \text{ or } B$.



- | | | |
|--|---|---|
| | 4 | В окне <i>Bin</i> столбца <i>Result</i> появится результат (10101) поразрядного логического сложения чисел $\hat{A} = 20_{10}$, $\hat{A} = 11_{16}$. |
| | 5 | Ввести значение промежуточного результата в текстовое поле <i>Bin</i> и аргумента <i>C</i> в текстовое поле <i>Oct</i> и с помощью группы переключателей <i>Operation</i> выбрать оператор логического умножения <i>A and B</i> . |
| | 6 | В окне <i>Bin</i> столбца <i>Result</i> появится результат (10000) поразрядного логического умножения чисел $A \square B = 10101_2$, $C = 30_8$, а в окне <i>Dec</i> (16), результат в десятичной системе счисления. |

Лабораторная работа 2.1

Тема: Информационная деятельность человека

Цели занятия:

- познакомить учащихся с понятиями архивации данных, с принципами архивации; архиваторами, характеристиками программ-архиваторов, функциями программ-архиваторов;
- познакомить с программой-архиватором WINRAR.

Теоретическая часть

Архивирование – это упаковка (сжатие) папки, файла или группы файлов для того, чтобы уменьшить место, занимаемое ими на диске, при обмене информацией по сети Интернет, а также при создании резервных копий, например на дискетах. Для архивирования используются специальные программы-архиваторы или диспетчеры архивов. Наибольшее распространение имеют способы сжатия данных в формате **arj**, **zip** и **rar**.

Для операционной системы MS DOS чаще всего используют программы **Arj.exe**, **PKZip.exe/PKUNZip.exe**, а для Windows95/98/NT/2000/XP – **WinArj**, **WinZip** или **WinRar**. Эти программы обеспечивают возможность использования и других архиваторов, поэтому, если на компьютере, куда перенесены сжатые в них файлы, отсутствуют указанные программы, архивы можно распаковать с помощью другого архиватора.

Основными характеристиками программ-архиваторов являются:

- Скорость работы;
- Сервис (набор функций архиватора);
- Степень сжатия – отношение размера исходного файла к размеру упакованного файла.

Степень сжатия файла характеризуются коэффициентом K_c , отражающим процентное отношение объёма сжатого файла V_c к объёму исходного файла V_o :

$$K_c = V_c/V_o * 100\%.$$

Программа **WinZip** не входит в комплект поставки Windows. Для использования этого архиватора его необходимо предварительно установить на компьютер. При установке программы-архиватора с операционной системой, в результате чего в контекстных меню объектов появляются дополнительные команды, рядом с которыми находятся характерные значки программы **WinZip**.

Практическая часть

Работа с программой архивации *WinRAR*

1. Создайте на рабочем столе папку *Архив-1*.
2. Скопируйте в папку *Архив-1* документы, рисунки из своей папки, которые вы собираетесь архивировать.
3. Запустите программу *WinRAR* с помощью ярлычка на рабочем столе. Найдите и откройте папку *Архив-1* в окне программы *WinRAR*.
4. Выделите файлы с расширением *.doc.
5. Выполните команду **Команды/Добавить файлы в архив**, либо щелкните на кнопке **Добавить**.
6. В открывшемся окне **Имя и параметры архива** установите основные параметры архивации:
 - Имя архива (DOC.rar);
 - Папку архива (Свою папку)
 - Метод сжатия (Обычный);
 - Формат архива (RAR);
7. Выделите файлы с расширениями *.xls.
8. Отправьте выделенные файлы в архив *DOC.rar*.
9. Создайте на вашем рабочем диске папку *Архив-2*.
10. Извлеките из архива *DOC.rar* файлы с расширением *.doc и поместите их в папку *Архив-2*
11. Откройте папку *Архив-1*.
12. Выделите все файлы.
13. Вызовите контекстное меню для выделенных файлов. Выберите команду **ДОБАВИТЬ В АРХИВ...**
7. В открывшемся окне **Имя и параметры архива** установите основные параметры архивации:
 - Имя архива (DOC1);
 - Папку архива (Рабочий стол);
 - Метод сжатия (Обычный);
 - Формат архива (RAR);
 - Параметры архивации («Создать SFX-архив» и «Удалить файлы после упаковки»)
8. У вас на рабочем столе создан самораспаковывающийся архив с расширением .exe.
9. Удалите созданные вами папки и файлы с рабочего стола.

ИСПОЛЬЗОВАНИЕ ПАРОЛИРОВАНИЯ И РЕЖИМА СОКРЫТИЯ ПАПОК И ФАЙЛОВ ДЛЯ ОБЕСПЕЧЕНИЯ ЗАЩИТЫ ИНФОРМАЦИИ

1. Установка режима сокрытия файлов и папок:
 - Выполнить команду: ПУСК/ПАНЕЛЬ УПРАВЛЕНИЯ;
 - В окне ПАНЕЛИ УПРАВЛЕНИЯ выбрать категорию СИСТЕМА И БЕЗОПАСНОСТЬ;

- В открывшемся окне СИСТЕМА И БЕЗОПАСНОСТЬ в левой панели выбрать пункт меню ОФОРМЛЕНИЕ И ПЕРСОНАЛИЗАЦИЯ, а в правой панели — ПАРАМЕТРЫ ПАПОК;
 - В открывшемся диалоговом окне ПАРАМЕТРЫ ПАПОК, перейти на вкладку ВИД, и в области дополнительных параметров активизировать переключатель НЕ ПОКАЗЫВАТЬ СКРЫТЫЕ ПАПКИ, ФАЙЛЫ И ДИСКИ (ПОКАЗЫВАТЬ СКРЫТЫЕ ПАПКИ, ФАЙЛЫ И ДИСКИ);
 - Щелкнуть на кнопке ОК.
2. **Установка атрибута СКРЫТЫЙ для файлов и папок:**
- Вызвать контекстное меню для папки или файла;
 - Выбрать команду СВОЙСТВА;
 - В открывшемся диалоговом окне СВОЙСТВА установить атрибут СКРЫТЫЙ;
 - Щелкнуть на кнопке ОК.
3. **Задание пароля на открытие, редактирование файла:**
- При сохранении файла в диалоговом окне СОХРАНЕНИЕ необходимо выполнить команду СЕРВИС/ПАРАМЕТРЫ БЕЗОПАСНОСТИ;
 - В открывшемся диалоговом окне БЕЗОПАСНОСТЬ ввести пароль для открытия файла и пароль разрешения записи файла;
 - Щелкнуть на кнопке ОК;
 - Продублировать введенные пароли;
 - Задать имя файла и папку для его хранения;
 - Щелкнуть на кнопке ОК;

Задания:

1. Установить атрибут СКРЫТЫЙ для своей личной папки.
2. Установить режим сокрытия папок и файлов;
3. Создать два документа (один в MS Word, другой — в MS Excel), сохранить их защитив паролем для открытия файла и разрешения записи файла.

Тема 3.1. Аппаратное обеспечение и программное обеспечение ЭВМ

Лабораторная работа №3.1.1.

Тема: Архитектура компьютеров

Цель работы:

- Отработка основных операций с окнами WINDOWS;
- Настройка конфигурации и запуск программ.

Практическая часть:

1. Расставить ярлыки программ и папки на рабочем столе в произвольном порядке.
2. Выполнить команду размещения значков на рабочем столе в определенном порядке.
3. Запустить программу WORD с помощью меню:

ПУСК-ПРОГРАММЫ-WORD

4. Свернуть окно программы WORD в кнопку на панели задач.

5. Запустить программу БЛОКНОТ с помощью меню:

ПУСК- ПРОГРАММЫ- СТАНДАРТНЫЕ

6. Свернуть окно программы БЛОКНОТ в кнопку на панели задач.

7. С рабочего стола WINDOWS запустить папку «Мой компьютер».

8. Развернуть окна программ БЛОКНОТ и WORD из кнопки на панели задач на рабочий стол.

9. Установить режимы отображения открытых окон на рабочем столе

- a) Каскадом

- b) Сверху вниз
- c) Слева направо

10. Вызвать контекстно-зависимое меню панели задач и выполнить команду:

СВЕРНУТЬ ВСЕ. (открытые окна).

1. Поменять фоновый узор и рисунок рабочего стола по своему усмотрению.

11. Установить на рабочем столе заставку: *ОБЪЕМНЫЙ ТЕКСТ*, изменить его содержание.

Установить интервал 15 минут.

12. Восстановить все свернутые окна на рабочий стол.

13. Показать задание преподавателю.

14. Просмотреть содержимое **КОРЗИНЫ**.

15. Получить системную информацию о вашем ПК с помощью программы **МОЙ КОМПЬЮТЕР**.

16. Показать задание преподавателю.

Лабораторная работа 3.1.2.

Тема: Операции с папками на рабочем столе (ОС Windows).

Цель занятия: *Формирование умений создавать, перемещать, копировать папки и текстовые документы на рабочем столе.*

Практическая часть:

1. На рабочем столе создать папку:

- ◆ Щелчок правой кнопкой мыши на рабочем столе;
- ◆ В контекстном меню выбрать СОЗДАТЬ – ПАПКА;
- ◆ Назвать ВЕСНА, нажать **enter**;

Создайте еще папку и назовите ЛЕТО.

2. Переименовать папку ЛЕТО на имя ОСЕНЬ:

- ◆ Щелчок по папке ЛЕТО правой кнопкой мыши;
- ◆ Выбрать в открывшемся меню ПЕРЕИМЕОВАТЬ;
- ◆ Удалить старое имя, ввести новое, нажать **enter**;

3. Отменить переименование:

- ◆ Щелкнуть правой кнопкой мыши рядом с папкой;
- ◆ Выбрать в контекстном меню ОТМЕНА ПЕРЕИМЕНОВАНИЯ.

4. В папке ЛЕТО создать документ ОТПУСК:

- ◆ Открыть созданную папку 2-мя щелчками мыши;
- ◆ В строке меню открыть ФАЙЛ-СОЗДАТЬ-ТЕКСТОВЫЙ ДОКУМЕНТ;
- ◆ Ввести имя, **enter**.

В папке ВЕСНА создать текстовый документ ГРАЧИ (аналогично)

5. Расположите окна сверху вниз:

- ◆ Открыть 2 папки ВЕСНА и ЛЕТО;
- ◆ На панели задач щелкнуть правой кнопкой мыши;
- ◆ В появившемся меню выбрать СВЕРХУ ВНИЗ;

6. В папке ЛЕТО сделайте две копии документа ОТПУСК:

1 способ:

- ◆ Выделить документ ОТПУСК, нажать кнопку КОПИРОВАТЬ на панели инструментов Стандартная;
- ◆ Нажать кнопку ВСТАВИТЬ;

2 способ:

- ◆ Правой кнопкой мыши перетащить ОТПУСК на свободное место в папке;
- ◆ В контекстном меню выбрать КОПИРОВАТЬ;

7. В папке ВЕСНА сделать две копии документа ГРАЧИ.

8. Перенести копии документа ГРАЧИ из папки ВЕСНА в папку ЛЕТО:

- ◆ Выделить копии (с помощью клавиши **Ctrl**);
 - ◆ Нажать кнопку **ВЫРЕЗАТЬ**;
 - ◆ Нажать кнопку **ВСТАВИТЬ** в папке **ЛЕТО**;
- Перенести копии документа **ОТПУСК** в папку **ВЕЧНА**:
- ◆ Выделить, перетащить левой кнопкой мыши.
9. Создать ярлык для документа **ГРАЧИ**:
- ◆ Выделить документ, щелкнуть правой кнопкой мыши;
 - ◆ Из контекстного меню выбрать **СОЗДАТЬ-ЯРЛЫК**;
10. Удалить ярлыки:
- ◆ Выделить ярлык;
 - ◆ Нажать кнопку «удалить» на панели инструментов **Стандартная**;
11. Удалить копии документов:
- 1 способ:** выделить документ, нажать **Delete**;
- 2 способ:** выделить документ, кнопка Удалить на панели инструментов;
- 3 способ:** выделить документ, щелкнуть по нему правой кнопкой мыши, из контекстного меню выбрать **УДАЛИТЬ**.
12. Закрыть обе папки.
13. Удалить папку **ВЕЧНА**, используя правую кнопку мыши.
14. Удалить папку **ЛЕТО** в корзину, перетащив левой кнопкой мыши.
15. Очистить корзину:
- ◆ Открыть корзину;
 - ◆ Выделить свои удаленные документы (используя **Ctrl**);
 - ◆ С помощью правой кнопки мыши удалить документы.

Лабораторная работа 4.1.1.

Тема: Текстовый редактор (текстовый процессор MS Word)

Цели занятия: освоить приемы работы по форматированию текста в виде колонок, буквицы.

Ход работы:

1. Открыть текстовый редактор Word и создать новый документ
2. Установить параметры страницы (левое – 3; правое – 1,5; верхнее – 2,5; нижнее – 2,5).
3. Ввести текст обычным способом с соблюдением абзаца:

КОМПОНЕНТ КРОВИ ДАЮЩИЙ ВТОРУЮ ЖИЗНЬ

Сравнительно недавно на станции переливания крови стали производить новый компонент – глюнат, который изготовлен из плазмы крови и способен творить чудеса. Он активно используется в терапевтической практике, акушерстве, гинекологии, для лечения хронического заболевания легких, воспаления желудка, неврологических, кожных и ЛОР заболеваний.

У глюната есть аналоги, но либо менее совершенные, как сыворотка Филатова, либо слишком дефицитные в связи с этим малодоступные для рядовых граждан, такие, как полебиолин, производящийся всего двумя учреждениями России.

Глюнат является недорогостоящим препаратом, его можно приобрести на станции переливания крови, а также в отдельных аптеках, но, естественно, по рецепту врача.

4. После ввода отбить пустой абзац в конце текста, и установить разрыв (п. М. **Вставка** – **Разрыв** (в появившемся окне опция новая страница), затем выделить основной текст без заголовка и разрыва, выбрать п. М. **Формат – Колонки** (в диалоговом окне

выбрать **2 колонки**, выбрать интервал между колонками и нажать «**OK**»). Текст поделится на 2 ровные колонки.

5. В первом абзаце отформатировать буквицу для этого: устанавливаем текстовый курсор в 1 абзаце, выбираем отступ первой строки 0 см п. М. **Формат – Буквица** (в открывшемся окне выбираем «**в тексте**» нажимаем «**OK**»)
6. Сохраняем текст в своей папке под именем «Колонки»

Созданный текст должен иметь следующий вид

КОМПОНЕНТ КРОВИ ДАЮЩИЙ ВТОРУЮ ЖИЗНЬ

равнительно недавно на станции переливания крови стали производить новый компонент – глюнат, который изготовлен из плазмы крови и способен творить чудеса. Он активно используется в терапевтической практике, акушерстве, гинекологии, для лечения хронического заболевания легких, воспаления желудка, неврологических, кожных и лор заболеваний.

У глюната есть аналоги, но либо менее совершенные, как сыворотка

Филатова, либо слишком дефицитные в связи с этим малодоступные для рядовых граждан, такие, как полебиолин, производящийся всего двумя учреждениями России.

Глюнат является недорогостоящим препаратом, его можно приобрести на станции переливания крови, а также в отдельных аптеках, но, естественно, по рецепту врача.

Лабораторная работа 4.1.2.

Тема: Создание графических информационных объектов

Цель работы: Получение практических навыков по созданию рисунков и схем с помощью графического редактора Word.

Запустите текстовый процессор **Word**. Создайте новый документ. Распахните окно документа.

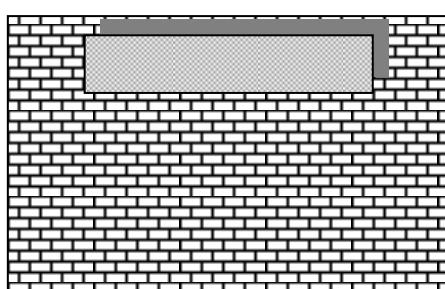
Проверьте и откорректируйте, если это нужно, настройку редактора в соответствии с параметрами, указанными в практической работе 3.

Выполните на экране **непечатаемые символы**.

Ведите текст:

Создание рисунков с помощью встроенного графического редактора Word.
Создание схемы **Параметры раздела**.

После первого абзаца текста вставьте несколько пустых абзацев и с помощью инструментов панели Рисование создайте многослойный рисунок (рис. 4.1.2.).



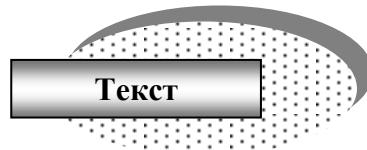


Рис. 4.1.2. Пример многослойного рисунка

Сохраните документ под именем **Рисование.doc**.

Для объединения всех элементов рисунка их следует выделить и сгруппировать.

Выделение нескольких элементов рисунка производится однократным мыши при одновременно нажатой клавише **Shift** или с помощью кнопки панели Рисование **Выбор объектов**

Сгруппировать рисунок можно с помощью кнопки панели Рисование **Действия**.

Сгруппируйте элементы рисунка.

Переместите рисунок к правой границе документа и убедитесь в том, что сгруппированы все элементы рисунка.

Привяжите рисунок к первому абзацу текста.

Для этого выделите рисунок, перетащите мышью символ привязки к нужному абзацу, введите команду **ФОРМА => Объект (Рисунок, Автофигура)** и на вкладке **Положение** установите соответствующий переключатель.

Создать в документе **Рисование.doc** объект **WordArt** и отредактируйте его с помощью панели **WordArt** и кнопок на Рисование.

Привяжите объект **WordArt** к первому абзацу текста.

Сохраните документ.

Создайте в документе **Рисование.doc** любую плоскую автофигуру.

Измените ее размер и сделайте копию рисунка с помощью мыши. (Копирование рисунков с помощью мыши производится при одновременно нажатой клавиши **Ctrl**)

С помощью кнопки **Тень** на панели Рисования настройте и откорректируйте тень для первой копии автофигуры.

Окрасьте рисунки, используя команды меню **ФОРМАт => Автофигура** и кнопки на панели Рисование.

Опробуйте различные способы заливки и обрамления рисунка.

Сделайте на первой автофигуре надпись **"Рисунок"**.

Удалите обрамление текстового поля.

Измените размер и цвет шрифта.

Используя кнопку **Действия => Порядок**, поместите надпись позади автофигуры.

Переместите надпись на передний план.

Удалите надпись.

С помощью кнопки **Объем** приайте второй автофигуре объемный вид.

Опробуйте различные варианты настройки объема.

Используя кнопку **Свободное вращение** (на панели Рисование), поверните рисунок.

С помощью кнопки **Действия** отразите рисунок сверху вниз.

Сохраните документ.

Лабораторная работа 4.1.3.

Тема: Создание и форматирование таблиц в MS WORD

Порядок работы

1. Откройте текстовый редактор Microsoft Word.

2. Установите параметры страницы (размер бумаги — А 4, ориентация — книжная, поля: левое — 3 см, правое — 1,5 см, верхнее — 2,5 см, нижнее — 2 см), используя команду **Файл/Параметры страницы**.

3. Установите формат абзаца (первая строка — отступ, межстрочный интервал — полуторный).

4. Создайте таблицу 9 x 6, используя команду *Таблица/Вставить/Таблица* (см. Рис.2.7) или кнопку панели инструментов *Добавить таблицу* (нажатием и продвижением по таблице с помощью левой кнопки мыши)

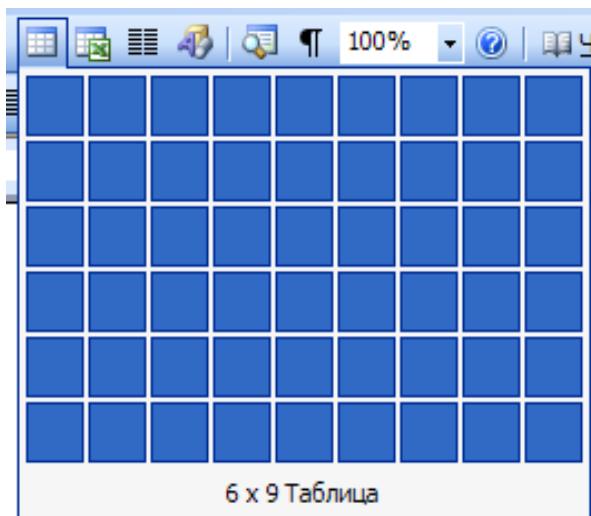


Рисунок 7 – Задание таблицы из панели инструментов

1. Измените ширину колонок: наведите стрелку мыши на вертикальный разделитель таблицы, при этом стрелка мыши примет вид разделителя (двусторонней стрелки); нажатием и продвижением разделителя левой кнопкой задайте нужную ширину столбцов таблицы (см. Табл. 2.1)

Таблица 1 -- Образец создания таблицы

Дата	Товарооборот		Выручка	Секции			Состав	Итого
	План	Факт		1	2	3		
2004	13552	13459	4598632	4562	1549	1249	25	1249
2005	15654	15486	5989642	9852	1255	2525	45	1554
2006	13658	14358	1259896	1554	1236	6459	96	15599
2007	56983	58962	125 584	2336	1255	2155	89	12544

6. Произведите объединение ячеек по вертикали и горизонтали как на образце. Для объединения или разбиения ячеек выделите группу ячеек и примените команду *Таблица/Объединить-разбить ячейки* или кнопками



7. Заполните таблицу данными, перемещаясь по ней мышью или с помощью клавиш [Tab], [Shift]-[Tab].

8. Добавьте в таблицу новую строку, для чего поместите курсор в правую ячейку нижней строки таблицы и нажмите клавишу [Tab] или воспользуйтесь командой *Таблица/Добавить/Строки выше/ниже*, предварительно установив курсор в любую ячейку нижней строки таблицы.

9. Выделите всю таблицу, для чего установите курсор в любую ячейку таблицы и выполните команду *Таблица/Выделить/Таблица* или щелкните левой кнопкой мыши по крестообразному указателю мыши в левом верхнем углу таблицы за ее контуром



10. Произведите обрамления таблицы по образцу, используя команду *Формат/Границы и заливка*.

11. Удалите добавленную пустую нижнюю строку таблицы и сохраните файл в вашей папке с именем «Таблица 1».

12. Скопируйте набранную таблицу в виде нового документа и произведите ее автоформатирование. Для этого установите курсор внутри таблицы, выберите в меню *Таблица* команду *Автоформат* и выберите формат таблицы — «Столбцы таблицы 3» (см. Рис. 2.8).

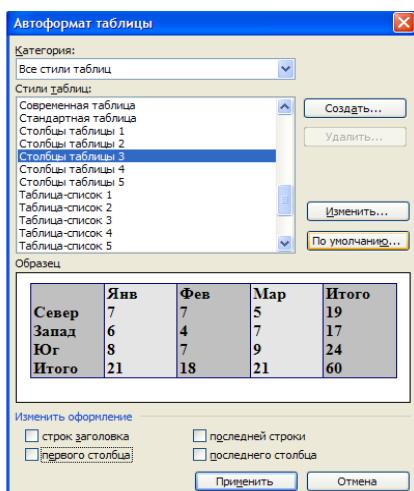


Рисунок 8 – Автоформат таблицы

13. Сохраните отформатированную таблицу в вашей папке с именем «Таблица 2» (*Файл/Сохранить как*).

Задание 2 Произвести вычисления успеваемости студентов в таблице.

Технология выполнения:

1. Создать таблицу по образцу (см. Табл. 2.2)

Таблица 2 – Образец таблицы

Сведения об успеваемости студентов								
Учебная лина	Группа	Средний балл	Всего авало	Отл	Хор	Удовл.	Неуд.	
<i>KCEa</i>								
1	МО11		<i>E4</i>	11	8	6	3	
2	МО12			12	7	6	2	
3	МО13			10	8	3	1	
ИТОГО								

2. Введите формулы для расчета числа студентов каждой группы, сдавших экзамен по дисциплине «Информатика». Для этого установите курсор в ячейку E4 и введите формулу: *SUM(RIGHT)*, предварительно убрав имеющуюся в ней запись. Ввести формулу надо, используя меню *Таблица\Формула....*

3. Произведите те же действия для ячеек E5 — E6.

Ведите формулу для расчетов среднего балла по дисциплине «Информатика» для группы МО11. Для этого установите курсор в ячейку D4 и введите формулу: $=(F4*5+G4*4+H4*3+I4*2)/E4$. Выберите формат числа 0,00.

4. Введите аналогичные формулы в ячейки D5 — D6.

Ведите формулу для расчета общего числа студентов, сдавших экзамен по каждой дисциплине на отлично, хорошо и т.д. Для этого установите курсор в ячейку F7 и введите формулу: *SUM(ABOVE)*, затем аналогично в ячейку G7 и т.д.

Лабораторная работа 4.2.1.

Тема: ОРГАНИЗАЦИЯ РАСЧЕТОВ В ТАБЛИЧНОМ ПРОЦЕССОРЕ MS EXCEL

Цель занятия: Изучение информационной технологии использования встроенных вычислительных функций Excel для финансового анализа.

Задание 1. Создать таблицу финансовой сводки за неделю, произвести расчеты, построить диаграмму изменения финансового результата, произвести фильтрацию данных.

Исходные данные представлены на рис. 1.

Порядок работы

1. Запустите редактор электронных таблиц Microsoft Excel и создайте новую электронную книгу (при стандартной установке MS Office выполните Пуск/Программы/Microsoft Excel).

The screenshot shows a Microsoft Excel spreadsheet titled "Финансовая сводка за неделю (тыс. руб)". The table has four columns: "Дни недели", "Доход", "Расход", and "Финансовый результат". The data rows are: понедельник (3245,20, 3628,50, ?), вторник (4572,50, 5320,50, ?), среда (6251,66, 5292,10, ?), четверг (2125,20, 3824,30, ?), пятница (3896,40, 3020,10, ?), суббота (5420,30, 4262,10, ?), воскресенье (6050,60, 4369,50, ?). Row 11 contains the formula =Ср.значение for the average of the last six days. Row 13 is a summary row with the formula =Сумм(Б4:Б11) for the total weekly income and =Сумм(В4:В11) for the total weekly expenses. The result cell for the weekly profit is also marked with a question mark.

Рис. 1. Исходные данные для Задания 8.1.

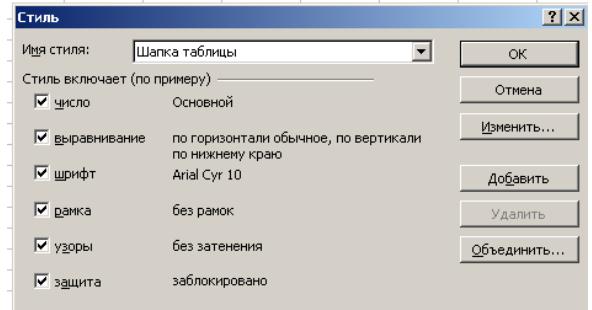
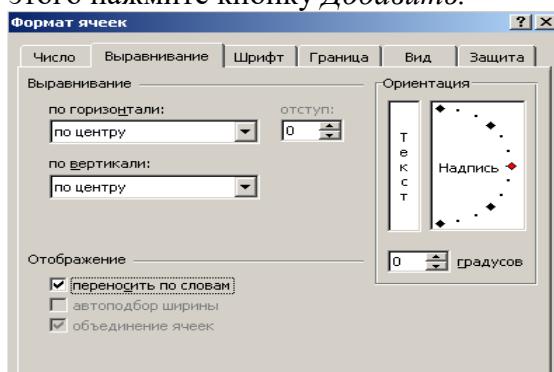


Рис. 2. Создание стиля оформления шапки таблицы

2. Введите заголовок таблицы «Финансовая сводка за неделю», начиная с ячейки A1.
3. Для оформления шапки таблицы выделите ячейки на третьей строке A3:D3 и создайте стиль для оформления. Для этого выполните команду *Формат/Стиль* и в открывшемся окне *Стиль* (рис. 2) наберите имя стиля «Шапка таблиц» и нажмите кнопку *Изменить*. В открывшемся окне на вкладке *Выравнивание* задайте *Переносить по словам* и выберите горизонтальное и вертикальное выравнивание — по центру (рис. 3), на вкладке *Число* укажите формат — *Текстовый*. После этого нажмите кнопку *Добавить*.



4. На третьей строке введите названия колонок таблицы — «Дни недели», «Доход», «Расход», «Финансовый результат», далее заполните таблицу исходными данными согласно Заданию 1.

Краткая справка. Для ввода дней недели наберите «Понедельник» и произведите автокопирование до «Воскресенья» (левой кнопкой мыши за маркер автозаполнения в правом нижнем углу ячейки).

5. Произведите расчеты в графе «Финансовый результат» по следующей формуле:

Финансовый результат = Доход - Расход,

для этого в ячейке D4 наберите формулу: = B4-C4.

Краткая справка. Введите расчетную формулу только для расчета по строке «Понедельник», далее произведите автокопирование формулы (так как в графе «Расход» нет незаполненных данными ячеек, можно производить автокопирование двойным щелчком мыши по маркеру автозаполнения в правом нижнем углу ячейки).

6. Для ячеек с результатом расчетов задайте формат — «Денежный» с выделением отрицательных чисел красным цветом (рис. 4) (*Формат/Ячейки/вкладка Число/формат — Денежный отрицательные числа — красные*. Число десятичных знаков задайте равное 2).

Обратите внимание, как изменился цвет отрицательных значений финансового результата на красный.

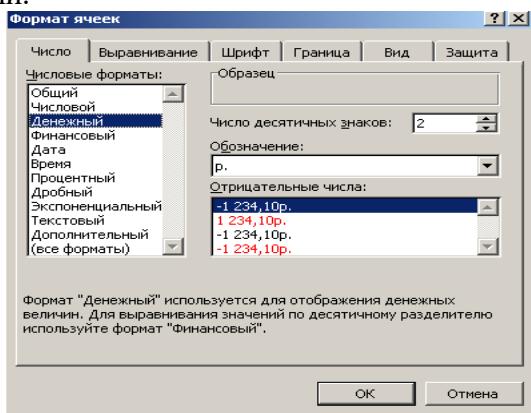


Рис. 4. Задание формата отрицательных чисел красным цветом

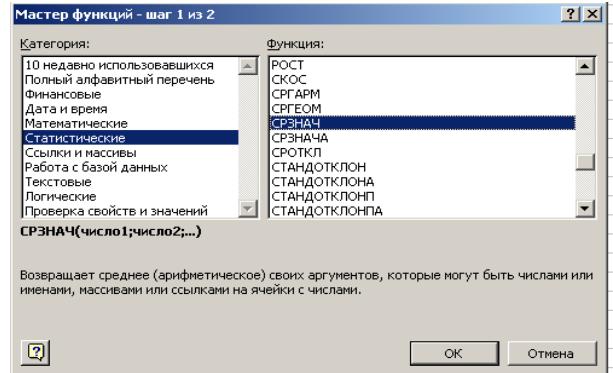


Рис. 5. Выбор функции расчета среднего значения

7. Рассчитайте средние значения Дохода и Расхода, пользуясь мастером функций (кнопка f_x). Функция «Среднее значение» (СРЗНАЧ) находится в разделе «Статистические». Для расчета функции СРЗНАЧ дохода установите курсор в соответствующей ячейке для расчета среднего значения (B11), запустите мастер функций (*Вставка/Функция/категория — Статистические/СРЗНАЧ*) (рис. 5). В качестве первого числа выделите группу ячеек с данными для расчета среднего значения — B4:B10.

Аналогично рассчитайте «Среднее значение» расхода.

8. В ячейке D13 выполните расчет общего финансового результата (сумма по столбцу «Финансовый результат»). Для выполнения автосуммы удобно пользоваться кнопкой *Автосуммирования* (Σ) на панели инструментов или функцией СУММ (рис. 6). В качестве первого числа выделите группу ячеек с данными для расчета суммы — D4:D10.

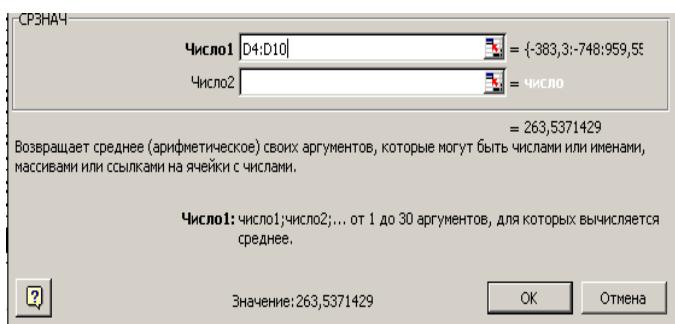


Рис. 6. Задание интервала ячеек при суммировании функцией СУММ

9. Проведите форматирование заголовка таблицы. Для этого выделите интервал ячеек от A1 до D1, объедините их кнопкой панели инструментов *Объединить и поместить в центр* или командой меню *Формат/Ячейки/вкладка Выравнивание/отображение — Объединение ячеек*. Задайте начертание шрифта — полужирное; цвет — по вашему усмотрению.

Конечный вид таблицы приведен на рис. 7.

Финансовая сводка за неделю (тыс. руб)			
Дни недели	Доход	Расход	Финансовый результат
понедельник	3245,20	3628,50	-383,30
вторник	4572,50	5320,50	-748,00
среда	6251,66	5292,10	959,56
четверг	2125,20	3824,30	-1699,10
пятница	3896,40	3020,10	876,30
суббота	5420,30	4262,10	1158,20
воскресенье	6050,60	4369,50	1681,10
Ср.значение	4508,84	4245,30	
Общий финансовый результат за неделю	1844,96		

Рис. 7. Таблица расчета финансового результата (Задание 1)

10. Постройте диаграмму (линейчатого типа) изменения финансовых результатов по дням недели с использованием мастера диаграмм.

Для этого выделите интервал ячеек с данными финансового результата и выберите команду *Вставка/Диаграмма*. На первом шаге работы с мастером диаграмм выберите тип диаграммы — линейчатая; на втором шаге на вкладке *Ряд* в окошке *Подписи оси X* укажите интервал ячеек с днями недели — A4:A10 (рис. 8).

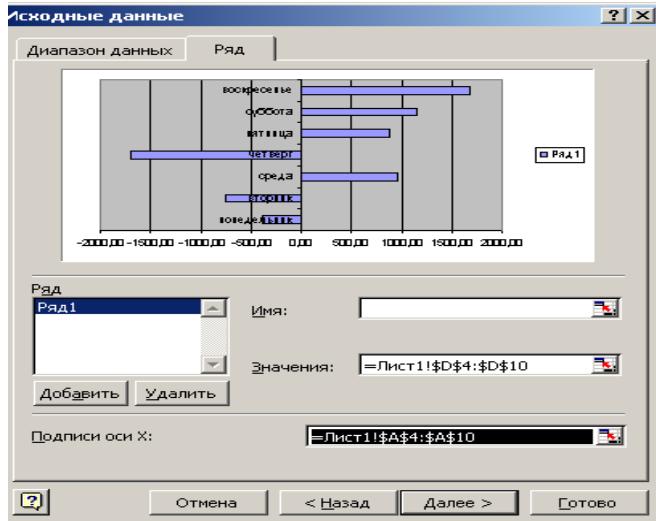


Рис. 8. Задание осей X при построении диаграммы

Далее введите название диаграммы и подписи осей; дальнейшие шаги построения диаграммы осуществляются автоматически по подсказкам мастера. Конечный вид диаграммы приведен на рис. 9.

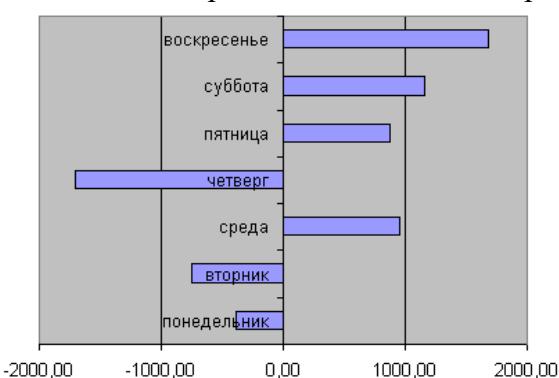


Рис. 9. Конечный вид диаграммы

Дополнительные задания

Задание 1. Заполнить таблицу, произвести расчеты, выделить минимальную и максимальную суммы покупки; по расчета построить круговую диаграмму суммы продаж.

	A	B	C	D
1				
2				
3	№	наименование	Цена руб.	Кол-во
4	1	Туфли	820,00	150
5	2	Сапоги	1530,00	60
6	3	Куртки	1500,00	25
7	4	Юбки	250,00	40
8	5	Шарфы	125,00	80
9	6	Зонты	80,00	50
10	7	Перчатки	120,00	120
11	8	Варежки	50,00	40
12				Всего
13				
14			Минимальная сумма покупки	
15			Максимальная сумма покупки	
16				

Задание 2

Заполнить ведомость учета брака, произвести расчеты, выделить максимальную, минимальную и среднюю суммы брака, а также средний процент брака; построить график значений изменения суммы брака по месяцам. (Сумма брака = Процент брака* Сумма зарплаты).

1	A	B	C	D	E	F
Ведомость учета брака						
2	Месяц	Ф.И.О.	Табельный номер	Процент брака	Сумма зарплаты	Сумма брака
3	Январь	Иванов	245	10%	3266	?
4	Февраль	Петров	289	8%	4568	?
5	Март	Сидоров	356	5%	4500	?
6	Апрель	Петров	786	11%	6804	?
7	Май	Паньчук	456	9%	6759	?
8	Июнь	Васин	987	12%	4673	?
9	Июль	Сорокин	9087	21%	5677	?
10	Август	Федорова	45	46%	6836	?
11	Сентябрь	светов	345	6%	3534	?
12	Октябрь	Меньшов	432	3%	5789	?
13	Ноябрь	Козлов	671	2%	4673	?
14	Декабрь	Титова	289	1%	6785	?
15						
16				Максимальная сумма брака	?	
17				Минимальная сумма брака	?	
18				Средняя сумма брака	?	
19				Средний процент брака	?	

Задание 3

Заполнить таблицу анализа продаж, произвести расчеты, выделить максимальную и минимальную продажи (количество и сумму); построить гистограмму значений изменения выручки по видам продукции.

Всего = Безналичные платежи + Наличные платежи

Выручка от продажи = Цена * Всего

1	A	B	C	D	E	F
АНАЛИЗ ПРОДАЖ продукции фирмы "ИНТЕРТРЕЙД" за текущий месяц						
2						
3						
4						
	Наименование продукции	Цена (руб)	Продажи		Выручка от продажи (руб)	
			Безналичные платежи (шт)	Наличные платежи (шт)	Всего (шт)	
5	Радиотелефон	4200	240	209	?	?
6	Телевизор	9500	103	104	?	?
7	Видеомагнитофон	6250	76	45	?	?
8	музыкальный центр	12750	10	17	?	?
9	Видеокамера	13790	57	45	?	?
10	Видеоплеер	4620	104	120	?	?
11	Аудиоплеер	450	72	55	?	?
12	Видеокассеты	120	516	247	?	?
13			Всего		?	
14						
15	Максимальные продажи	?	?	?		
16	Минимальные продажи	?	?	?		
17						
18						

Лабораторная работа №4.2.2.

Тема: СОЗДАНИЕ ЭЛЕКТРОННОЙ КНИГИ. ОТНОСИТЕЛЬНАЯ И АБСОЛЮТНАЯ АДРЕСАЦИИ В MS EXCEL

Цель занятия. Применение относительной и абсолютной адресаций для финансовых расчетов. Сортировка, условное форматирование и копирование созданных таблиц. Работа с листами электронной книги.

Задание 1. Создать таблицы ведомости начисления заработной платы за два месяца на разных листах электронной книги, произвести расчеты, форматирование, сортировку и защиту данных. Исходные данные представлены на рис. 1.

Порядок работы

1. Запустите редактор электронных таблиц Microsoft Excel и создайте новую электронную книгу.
2. Создайте таблицу расчета заработной платы по образцу (см. рис. 1). Введите исходные данные — Табельный номер, ФИО и Оклад, % Премии = 27%, % Удержания = 13 %.

	A	B	C	D	E	F	G
1	Ведомость начисления заработной платы						
2	за октябрь 2006 г.						
3	Табельный номер	Ф.И.О.	Оклад (руб)	Премия (руб)	Всего начислено (руб)	Удержания (руб)	К выдаче (руб)
4				27%		13%	
5	200	Иванов	3266,00	?	?	?	?
6	201	Петров	4568,00	?	?	?	?
7	202	Сидоров	4500,00	?	?	?	?
8	203	Петров	6804,00	?	?	?	?
9	204	Паньчук	6759,00	?	?	?	?
10	205	Васин	4673,00	?	?	?	?
11	206	Сорокин	5677,00	?	?	?	?
12	207	Федорова	6836,00	?	?	?	?
13	208	Светов	3534,00	?	?	?	?
14	209	Меньшов	5789,00	?	?	?	?
15	210	Козлов	4673,00	?	?	?	?
16	211	Титова	6785,00	?	?	?	?
17	212	Шашкин	8100,00	?	?	?	?
18	213	Жарова	6240,00	?	?	?	?
19		Всего	?	?	?	?	?
20							
21		Максимальный доход	?				
22		Минимальный доход	?				
23		Средний доход	?				

Рис. 1. Ведомость начисления заработной платы

При расчете Премии используется формула Премия = Оклад * % Премии, в ячейке D5 наберите формулу = \$D\$4 * C5 (ячейка D4 используется в виде абсолютной адресации) и скопируйте автозаполнением.

Рекомендации. Для удобства работы и формирования навыков работы с абсолютным видом адресации рекомендуется при оформлении констант окрашивать ячейку цветом, отличным от цвета расчетной таблицы. Тогда при вводе формул в расчетную окрашенная ячейка (т.е. ячейка с константой) будет вам напоминанием, что следует установить абсолютную адресацию (набором символов \$ с клавиатуры или нажатием клавиши [F4]).

Формула для расчета «Всего начислено»:

$$\text{Всего начислено} = \text{Оклад} + \text{Премия}.$$

При расчете Удержания используется формула

$$\text{Удержание} = \text{Всего начислено} * \% \text{ Удержания},$$

для этого в ячейке F5 наберите формулу = \$F\$4 * E5.

Формула для расчета столбца «К выдаче»:

$$\text{К выдаче} = \text{Всего начислено} - \text{Удержания}.$$

3. Рассчитайте итоги по столбцам, а также максимальный, минимальный и средний доходы по данным колонки «К выдаче» (*Вставка/Функция/категория — Статистические функции*).
4. Переименуйте ярлычок Листа 1, присвоив ему имя «Зарплата октябрь». Для этого дважды щелкните мышью по ярлычку и наберите новое имя. Можно воспользоваться командой *Переименовать* контекстного меню ярлычка, вызываемого правой кнопкой мыши. Результаты работы представлены на рис. 2.

Краткая справка. Каждая рабочая книга Excel может содержать до 255 рабочих листов. Это позволяет, используя несколько листов, создавать понятные и четко структурированные документы, вместо того, чтобы хранить большие последовательные наборы данных на одном листе.

5. Скопируйте содержимое листа «Зарплата октябрь» на новый лист (*Правка/Переместить/Скопировать лист*). Можно воспользоваться командой *Переместить/Скопировать* контекстного меню ярлычка. Не забудьте для копирования поставить галочку в окошке *Создавать копию* (рис. 3).

	A	B	C	D	E	F	G	H
1	Ведомость начисления заработной платы							
2	за октябрь 2006 г.							
3	Табельный номер	Ф.И.О.	Оклад (руб)	Премия (руб)	Всего начислено (руб)	Удержания (руб)	К выдаче (руб)	
4				27%		13%		
5	200	Иванов	3266,00	881,82	4147,82	539,22	3608,60	
6	201	Петров	4568,00	1233,36	5801,36	754,18	5047,18	
7	202	Сидоров	4500,00	1215,00	5715,00	742,95	4972,05	
8	203	Петров	6804,00	1837,08	8641,08	1123,34	7517,74	
9	204	Паньчук	6759,00	1824,93	8583,93	1115,91	7468,02	
10	205	Васин	4673,00	1261,71	5934,71	771,51	5163,20	
11	206	Сорокин	5677,00	1532,79	7209,79	937,27	6272,52	
12	207	Федорова	6836,00	1845,72	8681,72	1128,62	7553,10	
13	208	Светов	3534,00	954,18	4488,18	583,46	3904,72	
14	209	Меньшов	5789,00	1563,03	7352,03	965,76	6396,27	
15	210	Козлов	4673,00	1261,71	5934,71	771,51	5163,20	
16	211	Титова	6785,00	1831,95	8616,95	1120,20	7496,75	
17	212	Шашкин	8100,00	2187,00	10287,00	1337,31	8949,69	
18	213	Жарова	6240,00	1684,80	7924,80	1030,22	6894,58	
19		Всего	78204,00	21115,08	7924,80	12911,48	86407,60	
20								
21		Максимальный доход	8949,69					
22		Минимальный доход	3608,60					
23		Средний доход	6171,97					

Рис. 2. Результаты начисления заработной платы

Краткая справка. Перемещать и копировать листы можно, перетаскивая их корешки (для копирования удерживайте нажатой клавишу [Ctrl]).

6. Присвойте скопированному листу название «Зарплата ноябрь». Исправьте название месяца в названии таблицы. Измените значение премии на 32%. Убедитесь, что программа произвела пересчет формул.

7. Между колонками «Премия» и «Всего начислено» вставьте новую колонку «Доплата» (*Вставка/Столбец*) и рассчитайте значение доплаты по формуле Доплата = Оклад * % Доплаты. Значение доплаты примите равным 5 %.
8. Измените формулу для расчета значений колонки «Всего начислено»:
Всего начислено = Оклад + Премия + Доплата.

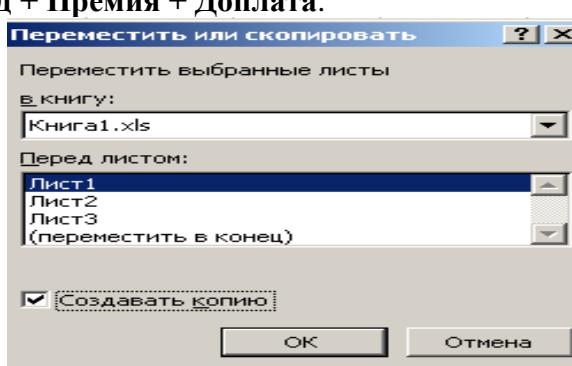


Рис. 3. Копирование листа электронной книги

9. Проведите условное форматирование значений колонки «К выдаче». Установите формат вывода значений между 7000 и 10 000 — зеленым цветом шрифта; меньше 7000 — красным; больше или равно 10 000 — синим цветом шрифта (*Формат/Условное форматирование*) (рис. 4).

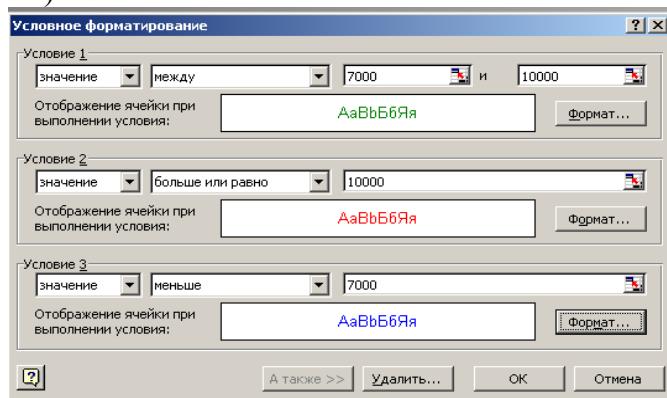


Рис. 4. Условное форматирование данных

10. Проведите сортировку по фамилиям в алфавитном порядке по возрастанию (выделите фрагмент с 5 по 18 строки)

Таблицы без итогов, выберите меню *Данные/Сортировка*, сортировать по — *Столбец В*) (рис. 5).

11. Поставьте к ячейке D3 комментарии «Премия пропорциональна окладу» (*Вставка/Примечание*), при этом в правом верхнем углу ячейки появится красная точка, которая свидетельствует о наличии примечания.
12. Защитите лист «Зарплата ноябрь» от изменений (*Сервис/Защита/Зашитить лист*). Задайте пароль на лист (рис. 6), сделайте подтверждение пароля (рис. 8).

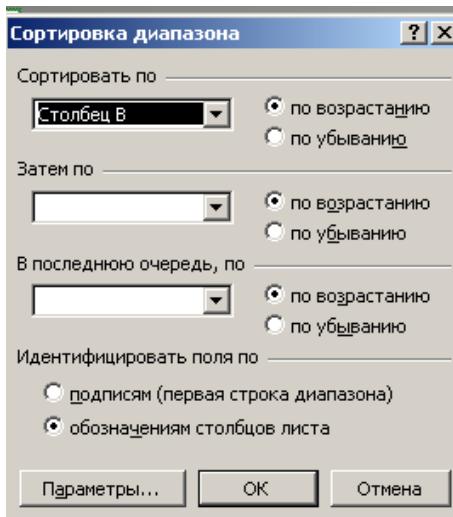


Рис. 5. Сортировка данных

Убедитесь, что лист защищен и невозможно удаление данных. Снимите защиту листа

{Сервис/Защита/Снять защиту листа}.

13. Сохраните созданную электронную книгу под именем «Зарплата» в своей папке.

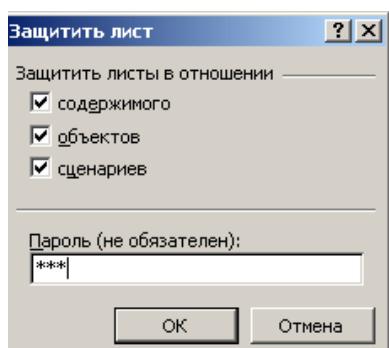


Рис. 6. Защита листа электронной книги

Задание 2. Сделать примечания к двум-трем ячейкам.

Задание 3. Выполнить условное форматирование оклада и

премии за ноябрь месяц:

до 2000 р. — желтым цветом заливки;

от 2000 до 10 000 р. — зеленым цветом шрифта;

свыше 10 000 р. — малиновым цветом заливки, белым цветом шрифта.

Задание 4. Защитить лист зарплаты за октябрь от изменений.

Проверьте защиту. Убедитесь в неизменяемости данных. Снимите защиту со всех листов электронной книги «Зарплата».

Задание 5. Построить круговую диаграмму начисленной суммы к выдаче всех сотрудников за ноябрь месяц.

Дополнительное задание

Задание 1. Рассчитать таблицу

Введите произвольно случайные числа от 0 до 10 в диапазон A2:A20. для чего в ячейку A2 введите формулу = ЦЕЛОЕ(СЛЧИС() * 10).

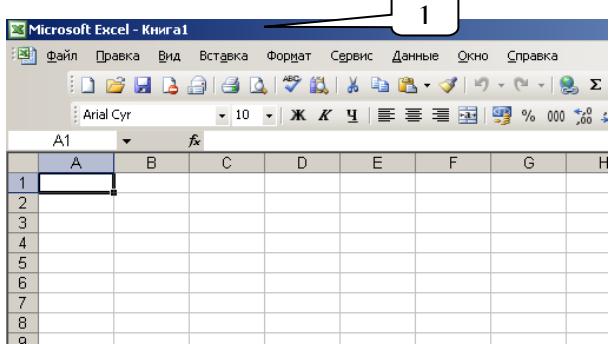
	A	B	C	D	E	F
1	Число	Квадрат числа	Куб последующего числа	Увеличение числа на 5%	Нарастающий итог	Разница между предыдущим и последующим числом
2	0	?	?	?	?	?
3	9	?	?	?	?	?
4	1	?	?	?	?	?
5	6	?	?	?	?	?
6	5	?	?	?	?	?
7	5	?	?	?	?	?
8	6	?	?	?	?	?
9	1	?	?	?	?	?
10	9	?	?	?	?	?
11	0	?	?	?	?	?
12	2	?	?	?	?	?
13	7	?	?	?	?	?
14	4	?	?	?	?	?
15	2	?	?	?	?	?
16	4	?	?	?	?	?
17	4	?	?	?	?	?
18	8	?	?	?	?	?
19	4	?	?	?	?	?
20	3	?	?	?	?	?
21						
22			Итоги			
23			Сумма чисел			?
24			Ср.знач квадратов			?
25			Мин кубов последующих чисел			?
26			Макс кубов последующих чисел			?
27			Кол-во элементов в нарастающем			?
28			Кол-во отриц элементов в разнице чисел			?

Самостоятельная работа по теме «Электронный табличный процессор»

1. Укажите назначение электронного табличного процессора.

2. Что понимают под условным названием КНИГА в работе программы MS EXCEL?

3. Назовите основные элементы окна программы



1

2

3

5

4

4. Укажите назначение следующих кнопок:



а

б

в

г

л

5. Чему будет равно значение ячейки Е5 после выполнения операции?

A	В	С	D	E
№ п/п	Ф.И.О.	тариф	отработано часов	начислено
2	1 Иванов И.П.	25	100	=C2*D2
3	2 Сидоров А.М.	23	150	
4	3 Орлова Л.Д.	24	150	
5	4 Михеева Е.В.	23	110	
6	5 Щукина И.А.	23	120	
7	6 Волович Т.П.	25	156	

а) 1200

б) 2400

в) 2760

г) 2500

5. В режиме предварительного просмотра заполненная таблица выглядит так:

Печать... Страница... Поля Разметка страницы Закрыть Справка					
№ п/п	Ф.И.О.	тариф	отработано часов	начислено	
1	Иванов И.П.	25	100		
2	Сидоров А.М.	23	150		
3	Орлова Л.Д.	24	150		
4	Михеева Е.В.	23	110		
5	Щукина И.А.	23	120		
6	Волович Т.П.	25	156		

Какую операцию необходимо выполнить, чтобы данные были представлены в форме таблицы?

6. В документе представлены ежемесячные сведения о доходах и расходах предприятия. В какой форме лучше представить эти данные, чтобы можно было максимально отследить эти изменения?
7. Чему будет равно значение ячейки D3 после выполнения операции?

A	B	C	D	E	F
1	12	45	9	=3*A1+C1)/B1	
2	11	25	6		
3	12	5	18		
4					

Лабораторная работа 4.3.1

Тема: Нормализация баз данных

Цель работы: Отработка способов создания БД, создания таблицы с помощью конструктора, определения форматов и размера полей .

ЗАДАНИЕ:

1. Создайте новую базу данных.
2. Создайте таблицу базы данных.
3. Определите поля таблицы в соответствии с таблицей:

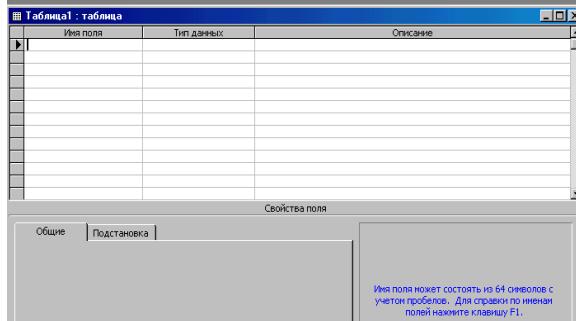
Имя поля	Тип данных	Размер поля
Код преподавателя	Счетчик	
Фамилия	Текстовый	15
Имя	Текстовый	15
Отчество	Текстовый	15
Дата рождения	Дата/время	Краткий
Должность	Текстовый	9
Дисциплина	Текстовый	11
Телефон	Текстовый	9
Зарплата	Денежный	

4. Сохраните созданную таблицу.

Ход работы:

1. Для создания новой базы данных:

- Загрузите ACCESS, в появившемся окне выберите пункт Новая база данных;
- В окне <<Файл новой базы данных>> задайте имя вашей базы и выберите папку, где ваша база данных будет находиться. По умолчанию ACCESS предлагает вам имя базы db1, а тип файла – Базы данных ACCESS. Имя задайте Преподаватели, а тип файла оставьте прежним, т.к. другие типы файлов нужны в специальных случаях;



- Щелкните по кнопке <Создать>.
2. Для создания таблицы базы данных:
- В окне базы данных выберите вкладку ТАБЛИЦЫ, а затем щелкните по кнопке <Создать>;
 - В окне <<Новая таблица>> выберите пункт Конструктор и щелкните по кнопке <Создать>;
3. Для определения полей таблицы:
- Введите в строку столбца <Имя поля> имя первого поля Код преподавателя;
 - В строке столбца <<Тип данных>> щелкните по кнопке списка и выберите тип данных Счетчик. Поля вкладки Общие оставьте такими, как предлагает ACCESS. Для определения всех остальных полей таблицы базы данных Преподаватели выполните действия, аналогичные указанным выше.

На вкладке **Общие** в нижней части экрана измените данные в пункте **Размер поля**, а остальные пункты оставьте по умолчанию.

4. Для сохранения таблицы:
- Выберите пункт меню Файл, Сохранить;
 - В диалоговом <<Сохранение>> введите имя таблицы Преподаватели;
 - Щелкните по кнопке <Ok>.

В результате щелчка Access предложит вам задать ключевое поле, т.е. поле, индентифицирующее каждую запись.

Лабораторная работа 4.3.2.

Тема: СОЗДАНИЕ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ.

Цель работы: Приобретение навыков работы с реляционной базой данных, создание связей между таблицами.

Задание

1. Создайте базу данных *Деканат*.
2. Создайте структуру таблицы *Студенты*.
3. Создайте структуру таблицы *Дисциплины*.
4. Измените структуру таблицы *Преподаватели*.
5. Создайте структуру таблицы *Оценки*.
6. Разработайте схему данных, т.е. создайте связи между таблицами.

Ход работы:

1. Создайте базу данных *Деканат*, выполнив следующие действия:
 - Загрузите Access, в появившемся окне выберите пункт *Новая база данных*, затем щелкните по кнопке <OK>;
 - В окне <Файл новой базы данных> задайте имя (пункт *Имя файла*) и выберите папку (пункт *Папка*), где ваша база будет находиться. По умолчанию Access предлагает имя базы *db1*, а тип файла – *Базы данных Access*. Имя задайте *Деканат*, а тип файла оставьте прежним, так как другие типы файлов нужны в специальных случаях;
 - Щелкните по кнопке <Создать>.
2. Создайте структуру таблицы *Студенты*. Для этого:

- в окне <<Новая таблица>> выберите пункт *Конструктор* и щелкните по кнопке <OK>. В результате проделанных операций открывается окно таблицы в режиме конструктора, в котором следует определить поля таблицы;
- определите поля таблицы в соответствии с таблицей №1

Имя поля	Тип данных	
код студента	Числовой	целое
фамилия	Текстовый	15
имя	Текстовый	12
отчество	Текстовый	15
номер группы	Числовой	целое
телефон	Текстовый	9
стипендия	Логический	да/ нет

- в качестве ключевого поля задайте <<Код студента>>. Для этого щелкните по полю <<Код студента>> и по кнопке на панели инструментов или выполните команду **Правка, Ключевое слово**;
- закройте таблицу, задав ей имя *Студенты*.

Примечание: заполнить таблицу данными пока не требуется, это будет сделано в режиме формы.

3. создайте структуру таблицу *Дисциплины* аналогично п.2 в соответствии с таблицей №2 (4.4).

Имя поля	Тип данных	
Код дисциплины	Числовой	целое
название дисциплины	Текстовый	30

В качестве ключевого поля задайте <<Код дисциплины>>. Заполняться эта таблица будет также в режиме формы.

4. Структура таблицы *Преподаватели* уже создана в работе 1 и заполнена данными, поэтому для работы используйте таблицу с одним лишь изменением – в соответствии с рисунком:



В структуру таблицы надо добавить поле <<Код дисциплины>> и заполнить его в соответствии с данными таблицы №2 (4.4).

5. Создайте структуру таблицы *Оценки* аналогично п. 2 в соответствии с таблицей №3

Имя поля	Тип данных	Размер поля
Код студента	Числовой	Целое
Код дисциплины	Числовой	Целое
Оценки	Числовой	Байт

В этой таблице задавать ключевое поле не надо, так как данные во всех полях могут повторяться. Эта таблица, аналогично предыдущим, будет заполняться в виде формы.

6. Разработайте схему данных, т. е. создайте связи между таблицами. Для этого:

- щелкните по кнопке на панели инструментов или выполните команду **Сервис, Схема данных**. На экране появится окно <<Схема данных>>;
- щелкните по кнопке на панели инструментов или выполните команду **Связи, добавить таблицу**;
- в появившемся окне будет выделено название одной таблицы. Щелкните по кнопке <Добавить>;

- переведите выделение на имя следующей таблицы и щелкните по кнопке <Добавить>. Аналогично добавьте оставшиеся две таблицы;
- закройте окно, щелкнув по кнопке <Закрыть>;
- Создайте связь между таблицами *Дисциплины* и *Оценки*. Для этого подведите курсор мыши к полю <<Код дисциплины>> в таблице *Дисциплины*, щелкните левой кнопкой мыши и, не отпуская ее, перетащите курсор на поле <<Код дисциплины>> в таблицу *Оценки*, а затем отпустите кнопку мыши. На экране откроется окно <<Связи>>;
- Установите флажок («галочку») в свойстве *обеспечение целостности данных*, щелкнув по нему;
- Установите флажок в свойстве *Каскадное обновление связанных полей* и *Каскадное удаление связанных записей*;

Примечание. Задание каскадного обновления связанных полей и каскадного удаления связанных записей позволит вам отредактировать записи только в таблице *Дисциплины*, а в таблице *Оценки* эти действия будут со связанными записями выполняться автоматически. Например, если вы удалите из таблицы *Дисциплины* один предмет, то в таблице *Оценки* удалятся все строки, связанные с этим предметом.

- Щелкните по кнопке <Сохранить> в окне <<Связи>>.
- Аналогичным образом создайте связь между таблицами *Студенты* и *Преподаватели*.

При этом в таблице *Студенты* в поле <<Код преподавателя>> будет отображаться значение поля <<Код студента>> из таблицы *Преподаватели*.

и
уфимского Университета >>
тавлен на

рисунке №2



- Закройте окно схемы данных, ответив *ДА* на вопрос о сохранении макета.

7. Показать задание преподавателю.

Лабораторная работа 4.3.3.

Тема: ФОРМИРОВАНИЕ СЛОЖНЫХ ЗАПРОСОВ

Цель работы: Приобретение навыков работы с мастером создания запросов, отработка навыков задания условий на выборку данных.

ЗАДАНИЕ:

- Разработайте запрос с параметрами о студентах заданной группы, в котором при вводе в окно параметров номера группы (в примере это 151 или 152) на экран должен выводится состав этой группы.
- Создайте запрос, в котором выводятся оценки студентов заданной группы по заданной дисциплине.
- Создайте перекрестный запрос, в результате которого создается выборка, отражающая средний балл по дисциплинам в группах.
- Разработайте запрос на увеличение на 10% заработной платы тех преподавателей, кто получает менее 500 руб.
- Создайте запрос на удаление отчисленных студентов.
- Разработайте запрос на создание базы данных отличников.
- Для всех созданных вами запросов разработайте формы.

Ход работы:

- Для создания запроса с параметрами о студентах заданной группы:
 - откройте вкладку *Запросы*;
 - щелкните по кнопке <Создать>;
 - в появившемся окне выберете *Простой запрос* и щелкните по кнопке <OK>;
 - в появившемся окне в строке *Таблицы/запросы* выберете из списка таблицу *Студенты*;

- перенести все поля из окна <<Доступные поля>> в окно <<Выбранные поля>>, щелкнув по кнопке 
 - щелкните по кнопке <Далее>. Выводить надо все поля, поэтому еще раз щелкните по кнопке <Далее>;
 - в появившемся окне введите имя запроса *Группа*;
 - щелкните по кнопке <Готово>. На экране появится таблица с данными запроса. Но вам надо, чтобы при выполнении запроса выяснялся номер группы. Для этого перейдите в режим конструктора;
 - в строке Условия отбора для поля <<Номер группы>> введите фразу (скобки тоже вводить): **[Введите номер группы]**;
 - выполните запрос, щелкнув по кнопке  на панели инструментов, или выполните команду **Запрос, Запуск**; !
 - в появившемся окне введите 151 и щелкните по кнопке <OK>. на экране появится таблица с данными о студентах 151 – ой группы;
 - сохраните запрос и закройте таблицу запроса.
2. Для создания запроса, в котором выводятся оценки студентов заданной группы поданной дисциплине:
- на вкладке *Запросы* щелкните по кнопке <Создать>;
 - выберите *Простой запрос* и щелкните по кнопке <OK>;
 - Выберите таблицу *Студенты* и перенесите поля <<Фамилия>>, <<Имя>>, <<Отчество>>, <<Номер группы>> в окно <<Выделенные поля>> (выделяя нужное поле и щёлкая по кнопке );
 - в таблице *Дисциплины* выберите поле <<Название дисциплины>>;
 - в таблице *Оценки* выберите поле <<Оценки>>. Вы сформировали шесть полей запроса – они связаны между собой посредством схемы данных;
 - щёлкните по кнопке <Далее>, затем в появившемся окне снова щёлкните по кнопке <Далее>;
 - в появившемся окне введите имя запроса *Оценки группы*, затем в появившемся окне снова щёлкните по ячейке *Изменение структуры запроса* (в ней должна появиться чёрная точка) – это позволит сразу перейти в режим конструктора;
 - щёлкните по кнопке <Готово>;
 - в строке Условия отбора для поля <<Номер группы>> введите фразу: **[Введите номер группы]**;
 - в строке Условия отбора для поля <<Название дисциплины>> введите фразу: **[Введите название дисциплины]**;
 - выполните запрос;
 - в первом появившемся окне введите 152, затем щёлкните по кнопке <OK>, во втором – введите *Информатика* и щёлкните по кнопке <OK>. На экране появится таблица со списком 152-й группы и оценками по информатике;
 - сохраните запрос и закройте таблицу запроса.
3. Создайте перекрестный запрос о среднем балле в группах по дисциплинам. Но такой запрос строится на основе одной таблице или одного запроса, в связи с чем надо сначала сформировать запрос, в котором были бы поля <<Номер группы>>, <<Название дисциплины>> и <<Оценки>>. Для этого:
- на вкладке *Запросы* щёлкните по кнопке <Создать>;
 - Выберите *Простой запрос* и щёлкните по кнопке <OK>;
 - выберите из таблицы *Студенты* поле <<Номер группы>>;
 - выберите из таблицы *Дисциплины* поле <<Название дисциплины>>;
 - выберите из таблицы *Оценки* поле <<Оценки>>;
 - щёлкните по кнопке <Далее>, затем в появившемся окне снова щёлкните по кнопке <Далее>;
 - в появившемся окне введите имя запроса *Дисциплины оценки группы*;
 - щёлкните по кнопке <Готово>;
 - сохраните запрос и закройте таблицу запроса.

Теперь можно создавать перекрёстный запрос. Для этого:

- на вкладке *Запросы* щёлкните по кнопке <Создать>;
- выберите *Перекрёстный запрос* и щёлкните по кнопке <OK>;
- щёлкните по ячейке *Запросы*, выберите *Дисциплины оценки группы* и щёлкните по ячейке *Запросы*, выберите *Дисциплины оценки группы* и щёлкните по кнопке <Далее>;
- выберите поле <<Название дисциплины>> и щёлкните по кнопке <Далее>;
- выберите поле <<Номер группы>> и щёлкните по кнопке <Далее>;
- выберите функцию **AVG**, т.е. среднее (она по умолчанию уже выделена), и щёлкните по кнопке <Далее>;
- введите название запроса *Средние оценки* и щёлкните по кнопке <Готово>. Обратите внимание на то, что Access создаёт ещё итогово значение средних оценок по дисциплинам;
- закройте таблицу запроса.

4. Для создания запроса на изменение заработной платы преподавателей:

- на вкладке *Запросы* щёлкните по кнопке <Создать>;
- выберите *Простой запрос*;
- в таблице *Преподаватели* выберите поле <Зарплата>;
- щёлкните по кнопке <Далее>, затем в появившемся окне снова щёлкните по кнопке <Далее>;
- в появившемся окне введите имя запроса *Изменение зарплаты*;
- щёлкните по ячейке *Изменение структуры запроса*;
- щёлкните по кнопке <Готово>;
- в строке *Условия отбора* введите <I>500;
- откройте пункт меню *Запрос* и выберите **Обновление**;
- в строке конструктора запроса **Обновление** в поле <<Зарплата>> введите: [Зарплата]*1,1;
- выполните запрос, подтвердив готовность на обновление данных;
- закройте запрос, подтвердив его сохранение;
- откройте форму *Преподаватели*;
- просмотрите изменение заработной платы у преподавателей, получающих меньше 500 р.;
- закройте форму.

5. Для создания запроса на отчисление студента гр. 152 *Перлова Кирилла Николаевича*:

- на вкладке *Запросы* щёлкните по кнопке <Создать>;
- выберите *Простой запрос*;
- в таблице *Студенты* выберите поля <<Фамилия>> <<Имя>>, <<Отчество>>, <<Номер группы>>;
- щёлкните по кнопке <Далее>, затем в появившемся окне снова щёлкните по кнопке <Далее>;
- в появившемся окне введите имя запроса *Отчисленные студенты*;
- щёлкните по ячейке *Изменение структуры запроса*;
- щёлкните по кнопке <Готово>;
- в строке *Условия отбора* введите: в поле <<Фамилия>> - *Перлов*, в поле <<Имя>> - *Кирилл*, в поле <<Отчество>> - *Николаевич*, в поле <<Номер группы>> - 152;
- откройте пункт меню *Запрос* и выберите **Удаление**;
- просмотрите удаляемую запись, щёлкните по кнопке  или выполнив команду **Вид, Режим таблицы**;
- если отчисляемый студент выбран правильно, то перейдите в режим конструктора и выполните запрос. Если условия отбора сделаны неправильно, измените их
- закройте запрос;
- откройте форму *Студенты* и удостоверьтесь в удалении записи о студенте *Перлове*;
- закройте форму.

6. Для создания запроса на создание базы данных отличников:

- на вкладке *Запросы* щёлкните по кнопке <Создать>;
- выберите *Простой запрос*;
- в таблице *Студенты* выберите поля <<Фамилия>>, <<Имя>>, <<Отчество>>, <<Номер группы>>, а в таблице *Оценки* – поле <<Оценки>>;
- щёлкните по кнопке <Далее>, а затем в появившемся окне вновь щёлкните по кнопке <Далее>;
- в появившемся окне введите имя запроса *Отличники*;

- щёлкните по ячейке *Изменение структуры запроса*;
 - щёлкните по кнопке <Готово>;

 - для выполнения групповых операций щёлкните на панели инструментов по кнопке или выполните команду **Вид, Групповые операции**;
 - в строке *Групповые операции* поля <<Оценки>> щёлкните по ячейке *Групповые операции*. Откройте раскрывающийся список и выберите функцию **SUM**;
 - в строке *Условия отбора* поля <<Оценки>> введите 20
 - просмотрите создаваемую базу, щёлкнув по кнопке  или выполните команду **Вид, Режим таблицы**;
 - перейдите в режим конструктора;
 - выполните команду **Запрос, Создание таблицы**;
 - введите имя таблицы *Студенты – отличники* щёлкните по кнопке <OK>;
 - подтвердите создание таблицы;
 - закройте с сохранением запрос;
 - откройте вкладку *Таблицы*;
 - откройте таблицу *Студенты – отличники*. Убедитесь в правильности создания таблицы. Закройте таблицу.
7. Для каждого из созданных запросов создайте форму (можно рекомендовать автоформу в столбец или ленточную автоформу) для удобного просмотра данных.
8. Покажите работу преподавателю.

Лабораторная работа 4.3.3.

Тема: СОЗДАНИЕ СЛОЖНЫХ ОТЧЁТОВ

Цель работы: Приобретение навыков работы с мастером создания отчетов на основе запросов, отработка навыков по использованию вычисляемых полей.

ЗАДАНИЕ:

1. Создайте запрос, на основе которого будет формироваться отчёт. В запросе должны присутствовать: из таблицы *Студенты* – поля <<Фамилия>>, <<Имя>>, <<Отчество>> и <<Номер группы>>, из таблицы *Дисциплин* - поле <<Название дисциплины>>. Из таблицы *Оценки* – поле <<Оценки>>.
2. Создайте отчёт по итогам сессии. В отчёте оценки студентов должны быть сгруппированы по номерам групп и дисциплинам. Для каждого студента должна вычисляться средняя оценка в сессию, а для каждой группы – среднее значение оценок по всем предметам.

Ход работы:

1. Для создания запроса:
 - на вкладке *Запросы* щелкните по кнопке <Создать>;
 - выберите *Простой запрос* и щёлкните по кнопке <OK>;
 - из таблицы *Студенты* выберите поля <<Фамилия>>, <<Имя>>, <<Отчество>> и <<Номер группы>>, из таблицы *Дисциплин* - поле <<Название дисциплины>>, из таблицы *Оценки* – поле <<Оценки>> и щелкните по кнопке <Далее>;
 - щелкните по кнопке <Далее>;
 - введите название запроса *Сессия* и щелкните по кнопке <Готово>;
 - закройте запрос;
2. Для создания итогового отчёта выполните следующее:
 - На вкладке *Отчёты* щелкните по кнопке <Создать>;
 - Выберите *Мастер отчётов*, из раскрывающего списка – запрос *Сессия* и щелкните по кнопке <OK>;
 - Выберите все поля запроса и щелкните по кнопке <Далее>;
 - Тип представления данных нас удовлетворяет, поэтому щелкните по кнопке <Далее>;
 - Добавьте уровень группировки по номеру группы, выбрав в левом окне *Номер группы* и перенеся его в правое окно, щелкните по кнопке 

- щелкните по кнопке <Далее>;
- щелкните по кнопке <Итоги>, так как надо вычислять средний балл;
- поставьте "галочку" в ячейке поля <<AVG>> (эта функция вычисляет среднее) и щелкните по кнопке <OK>;
- щелкните по кнопке <OK>, так как сортировка не требуется, потому что данными являются название дисциплины и оценки, порядок которых не столь важен;
- выберите макет отчёта. Рекомендуем ступенчатый, т.к. он занимает меньше места и в нём наглядно представлены данные (хотя это дело вкуса). Щелкните по кнопке <Далее>;
- выберите стиль отчёта и щелкните по кнопке <Далее>;
- введите название отчёта *Итоги сессии* и щелкните по кнопке <Готово>.

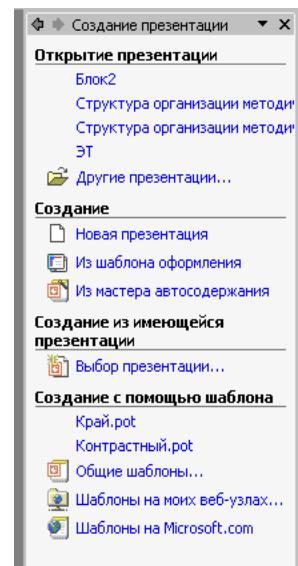
На экране появится отчёт. Его можно просмотреть, изменения масштаб (щёлкну по листу) и, перелистывая страницы (в нижней части экрана). Его можно также распечатать, выполнив команду **Файл, Печать**.

Лабораторная работа 4.4.1.

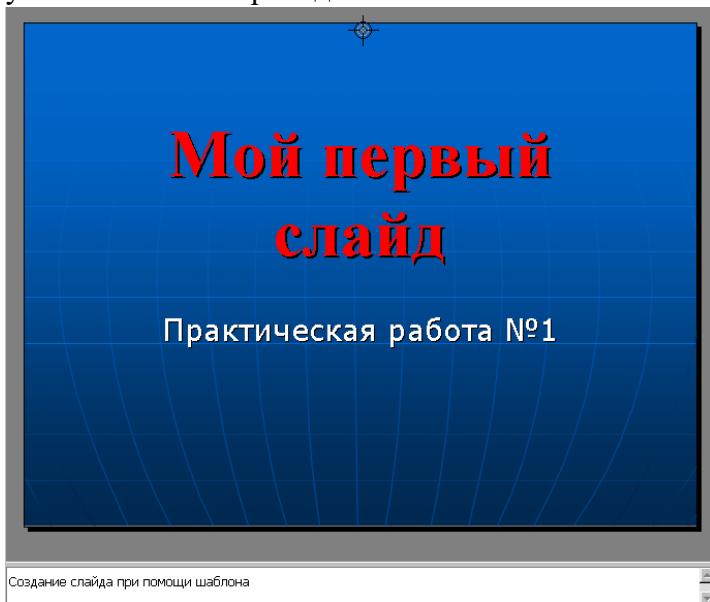
Тема: Создание слайда при помощи шаблона.

Порядок выполнения работы.

1. Запустите Microsoft Power Point (**Пуск → Программы → Microsoft Office → Power Point**)
2. Обратите внимание на появившийся диалог "Создание презентации" справа (Рис. 1). В этом диалоге в разделе "Создание" выберите пункт "Из шаблона оформления"
3. Из предложенных шаблонов выберите наиболее подходящий и щёлкните по нему левой кнопкой мыши.
4. Левой кнопкой мыши щёлкните по области, где написано "ЗАГОЛОВОК СЛАЙДА" и введите заголовок вашего слайда.
Например: "Мой первый слайд"
5. Установите для заголовка нужные параметры шрифта и цвета. (**Меню → Формат → Шрифт**)
6. Получите слайд такого вида как показано на рис.2 и **НЕ СОХРАНЯЯ** документ покажите преподавателю.



Rис. 1



Rис. 2

Практическая работа 4.2.2.

Тема: Добавление слайдов, режим просмотра слайдов, структура презентации.

Задание. Сформировать презентацию при помощи шаблона из 4-5 слайдов, войти в режим просмотра. В презентации режим авторазметки слайда.

Порядок выполнения работы.

1. В Power Point создайте заголовочный слайд такого вида, как показано на рисунке 3.
2. На панели инструментов найдите кнопку и щёлкните по ней левой кнопкой мыши. (Если такой кнопки там нет то войдите в меню **Вставка** → **Создать слайд**)



Рис. 3



Рис. 4

3. Обратите внимание на панель авторазметки слайда которая появилась справа (рис. 4). Выберите на ней нужный тип разметки слайда. (Если панель не появилась, то войдите в меню **Формат** → **Разметка слайда**).
4. Введите текст. Для получения следующего пункта списка нужно нажать **ENTER**. Вставьте рисунки из коллекции Microsoft. Для этого нужно щёлкнуть левой кнопкой мыши по иконке соответствующего объекта в окне разметки объектов на слайде.
5. Щёлкните правой кнопкой мыши по свободному месту слайда и выберите пункт **ФОН**. Выберите узор. В результате должен получиться слайд, как показано на рис 5.
6. Повторив порядок действий, самостоятельно получите слайд, как показано на рис. 6.
7. НЕ СОХРАНЯЯ ПРЕЗЕНТАЦИЮ, покажите преподавателю.
8. Запустите презентацию на просмотр, нажав клавишу **F5**

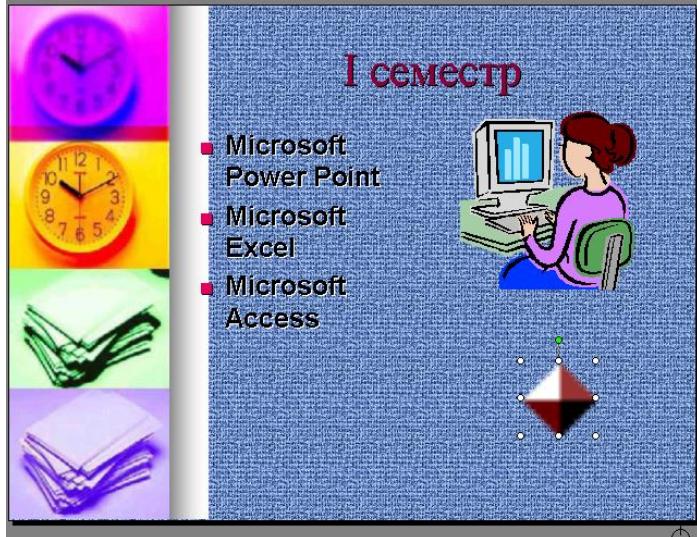


Рис. 5



Рис. 6

Практическая работа .4.2.3.

Тема: Перестановка слайдов. Копирование слайдов. Форматирование слайдов.

Задание:

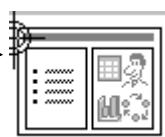
1. Создайте презентацию из 8 слайдов используя:

- ✓ *копирование слайдов*
- ✓ *изменение порядка слайдов*
- ✓ *создание итогового слайда*
- ✓ *вставка и размещение рисунков на слайде*

Порядок выполнения работы.

1. Создайте новую презентацию. Заголовочный слайд оформите как показано на рис. 7. Используйте шаблоны оформления презентации.

2. Создайте новый слайд. Выберите для него разметку, как показано на рисунке.



3. Выделите этот слайд и нажмите правую кнопку мыши. Выберите КОПИРОВАТЬ и создайте 5 копий этого слайда.

4. В созданных слайдах введите заголовки и вставьте соответствующие геометрические фигуры и тела (**Вставка** → **Рисунок** → **Из файла**). Создайте формулы площадей и объёмов, введите определения для фигур. Формулы удобно создавать в редакторе формул Microsoft Equation в Word.

5. На панели слева найдите кнопку "Структура" и щёлкните по ней.

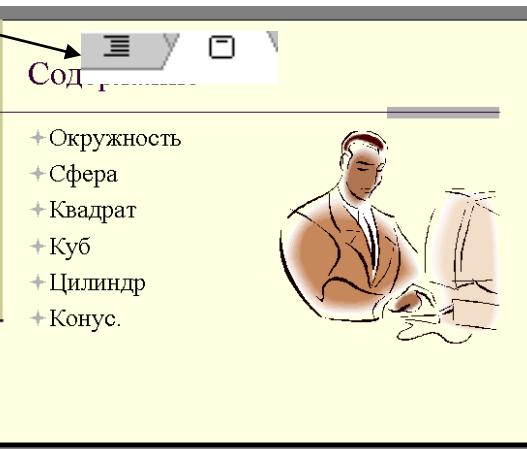
6. На панели инструментов "Структура" (если она не открыта, то войдите в меню **Сервис** → **Настройка** → **Панели инструментов** и включите её) — найдите кнопку итоговый слайд.

7. Выделите все слайды структуры (проведите по ним мышью, удерживая левую клавишу), после чего нажмите эту кнопку. Будет создан итоговый слайд.

8. Создав итоговый слайд, измените его заголовок на "Содержание". Проверьте, чтобы в маркированном

Геометрические фигуры и тела вращения.

Демонстрационная презентация



присутствовали заголовки всех созданных ранее вами слайдов. Например пункты: а) Окружность б) Сфера в) Квадрат г) Куб д) Конус е) Цилиндр.

9. Для форматирования маркированного списка выделите все его пункты, щёлкните по выделенному правой кнопкой мыши и выберите пункт **Список**. Далее выберите **Знак**. Выберите нужный знак.

10. Войдите в меню **Формат** → **Разметка слайда**. На панели разметки выберите соответствующую разметку и добавьте рисунок. Должен получиться слайд, как показано на рис. 8.

11. Переместите слайд с содержанием на вторую позицию, для этого в режиме "Слайды" щёлкните по нему мышью и, не отпуская, перетащите.

12. В режиме СЛАЙДЫ перетаскивая слайды левой кнопкой мыши расположите их в порядке:
1) Квадрат 2) Куб 3) Окружность 4) Сфера 5) Цилиндр 6) Конус.

13. Сохраните презентацию, покажите её преподавателю.

Практическая работа .4.2.4.

Тема: Форматирование слайдов. Добавление эффектов при показе слайдов. Оформление презентации

Задание: В презентации, созданной в предыдущей работе создать и настроить эффекты анимации.

Использовать:

- ✓ не менее двух эффектов анимации на каждом слайде.
- ✓ на одном из слайдов задать путь перемещения для объекта.
- ✓ в одном из эффектов использовать звуковой эффект
- ✓ использовать эффекты всех видов: входа, выхода, выделения

Порядок выполнения работы:

1. Откройте презентацию геометрические фигуры и тела вращения.
2. Откройте слайд **СОДЕРЖАНИЕ**. Выделите текстовую область. Щёлкните по ней правой кнопкой мыши и выберите пункт **ФОРМАТ ПРОТОТИПА**. На вкладке цвета и линии выберите цвет для заливки области. На вкладке **РАЗМЕР** укажите угол поворота.
3. Удалите заголовок "Содержание" (выделите его и нажмите клавишу <DELETE>). После этого войдите в меню **Вставка** → **Рисунок** → **Объект Word Art**. Введите заголовок СОДЕРЖАНИЕ. Настройте дополнительные параметры(цвет, поворот букв и.т.д), выделив рисунок, нажав правую кнопку мыши и выбрав пункт **СВОЙСТВА ОБЪЕКТА**. Слайд должен получиться такого вида, как показано на рис.9.

Настройка анимации

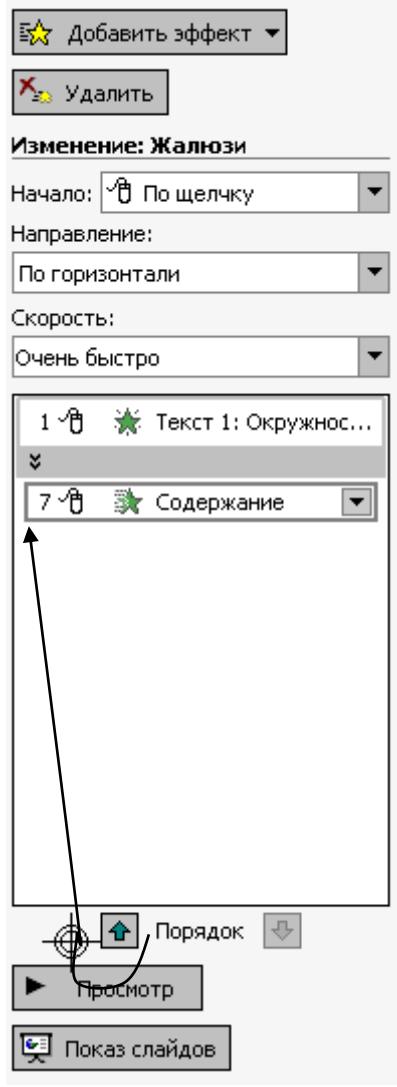


Рис. 11



Рис. 9

4. Войдите в меню **Показ слайдов ➔ Эффекты анимации**. (см рис 10) Выберите эффект, сделав просмотр. Эффект применится к тексту расположенному на слайде.

5. Выделите объект Word Art с заголовком "Содержание" и войдите в меню **Показ Слайдов ➔ Настройки Анимации**. (см рис 11). Нажмите кнопку ДОБАВИТЬ ЭФФЕКТ.

6. Откроется диалог, показанный на рис. 12. Здесь нужно указать эффекты которые будут применены при появлении объекта на слайде, эффект выделения объекта, эффект который будет применён при удалении объекта со слайда, если нет необходимости чтобы объект находился на слайде все время показа.

7. Настройте применённые эффекты. Например, сделайте так, чтобы они применялись автоматически друг за другом, а не по щелчку мыши. Сделать это можно либо в диалоге ИЗМЕНЕНИЕ, либо щёлкнув по эффекту правой кнопкой мыши. Подберите скорость эффекта.

8. Для изменения порядка выполнения эффектов на слайде в окошке эффектов (см рис. 13) выделите нужный эффект и перетащите его мышью или выделите и воспользуйтесь кнопками Порядок .

Если не удаётся определить эффект, который нужно изменить, то можно выделить объект к которому применяется эффект и этот эффект выделится в окне справа.

9. Чтобы настроить дополнительные параметры эффекта (звук, задержка и.т.д) выделите этот эффект в окне справа (рис. 13), нажмите правую кнопку мыши и выберите пункт "Параметры" или дважды щёлкните левой кнопкой мыши по эффекту. В открывшемся диалоге подберите звук на вкладке "Эффект", задайте время задержки на вкладке "Время".

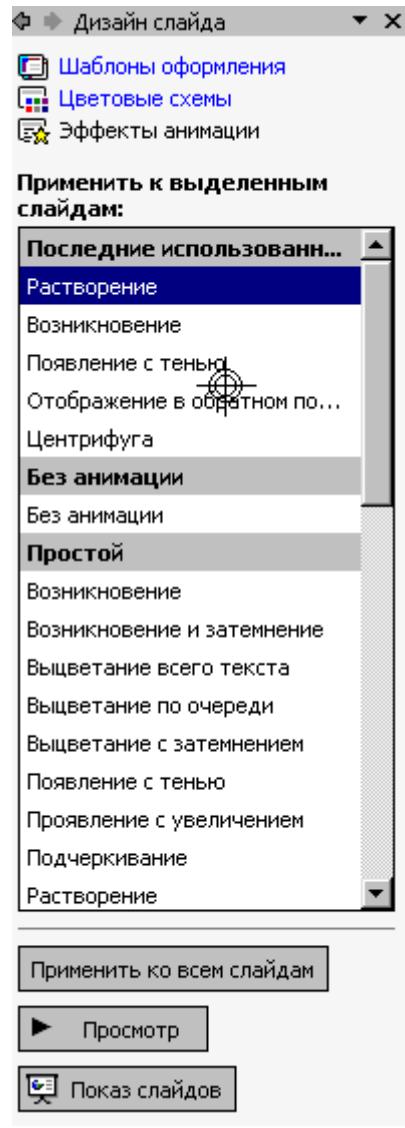


Рис. 10

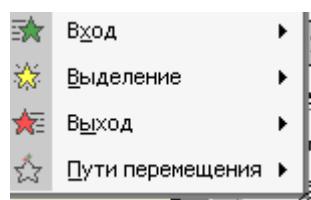


Рис.12

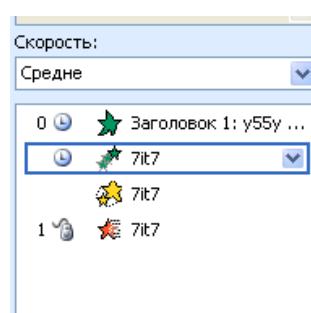
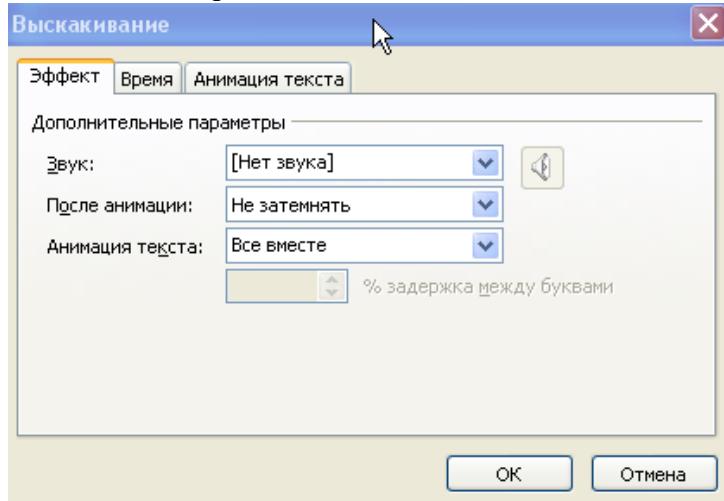


Рис.13

10. Выделите первый слайд и войдите в меню **Показ слайдов ➔ Смена слайдов**. В диалоге, открывшемся справа, выберите способ появления слайда. Например: шашки горизонтальные. Скорость перехода установите СРЕДНЕ. В разделе смена слайда установите флажок у пункта АВТОМАТИЧЕСКИ ПОСЛЕ и выберите время задержки слайда 5 секунд. Аналогичные действия проделайте со вторым слайдом, при этом время задержки должно быть больше, так как потребуется время на остальные эффекты добавленные ранее.



добавьте фон в виде градиентной заливки и примените по 1-2 эффекта. Готовую презентацию покажите преподавателю.

Практическая работа 4.4.5.

Тема: Добавление диаграмм, видеофрагментов и музыкального сопровождения к слайдам.
Использование гиперссылок.

Задание:

- 1. Создайте слайд с организационной диаграммой и оформите его.**
- 2. Создайте слайды, отвечающие основным блокам диаграммы.**
- 3. Добавьте на слайды видеофрагмент и звуковое сопровождение.**
- 4. Организуйте переходы между слайдами при помощи гиперссылок.**

Порядок выполнения работы:

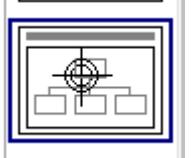
1. Создайте новую презентацию. Заголовочный слайд оформите произвольным образом и назовите "Руководство компании".
2. Добавьте новый слайд и войдите в режим разметки (**меню Формат ➔ Разметка слайда**). 
3. В появившемся диалоге выберите режим разметки, который указан на рис. 14.
4. Озаглавьте слайд "Структура подразделения".
5. Дважды щёлкните по месту где должна появиться организационная диаграмма и выберите пункт ОРГАНИЗАЦИОННАЯ ДИАГРАММА. На слайде появится форма для заполнения.
6. Введите текст и получите слайд примерно такого вида, как показано на рис. 15.

Рис.14

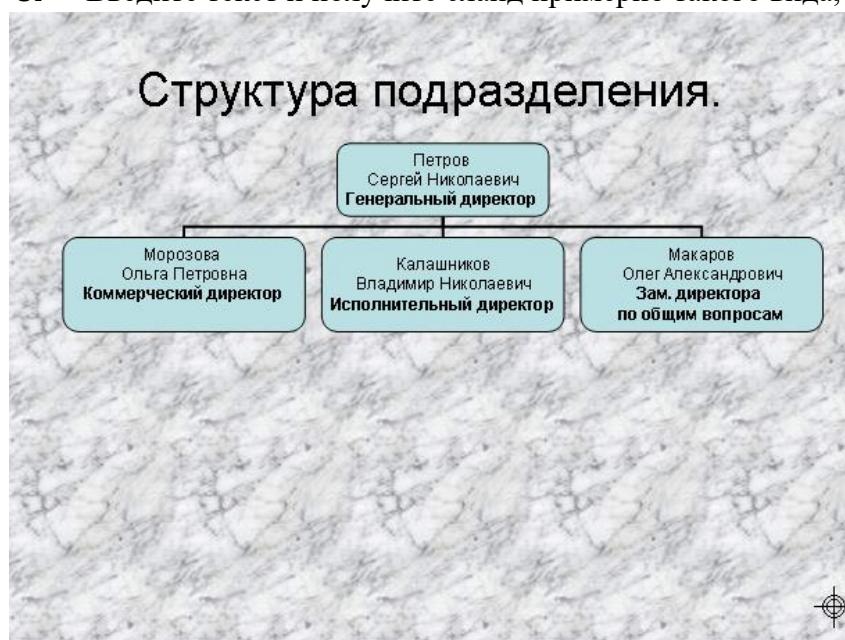


Рис. 15

7. Щёлкните по полученной модели, появится панель инструментов «Организационная диаграмма», как показано на рис. 16. Выделите ячейку «Генеральный директор» и на панели инструментов нажмите пункт «Добавить фигуру». Выберите Пункт «ПОМОЩНИК» и создайте окно помощника генерального директора, заполнив его текстом так как показано на рисунке 17.

8. Аналогичным образом задайте несколько подчинённых для каждого из подчинённых

генерального директора, заполнив их текстом произвольным образом.

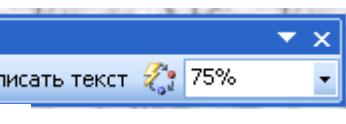
9. На панели инструментов найдите кнопку «Автоформат» и, щёлкнув по ней, в появившемся диалоге выберите пункт «Объёмный цвет».
10. В итоге слайд должен выглядеть так, как показано на рисунке 17.
11. Создайте новый слайд. Слайд оформите как слайд представляющий директора компании. На слайд поместите видеофрагмент и текст. Для этого выберите соответствующую разметку (см рис. 18). Для вставки видео фрагмента используйте файл из указанной папки. Текст слайда придумайте самостоятельно.
12. Создайте ещё 2-3 слайда представляющих помощников директора. Используйте текст и графические объекты.
13. На третьем слайде ("Генеральный директор") создайте автофигуру такого вида, как показано на рисунке 19. Выделите её и войдите в меню **Вставка** ➔

Гиперссылка. В открывшемся диалоге в колонке "СВЯЗАТЬ С..." (см рис 20), выберите пункт "С МЕСТОМ В ДОКУМЕНТЕ" и укажите слайд со структурой компании в окне справа. Таким образом будет получено 3 (4) гиперссылки со слайдов-представлений на слайд структуры.



Рис. 19

Рис. 17



16. Покажите презентацию преподавателю и сохраните на дискету.

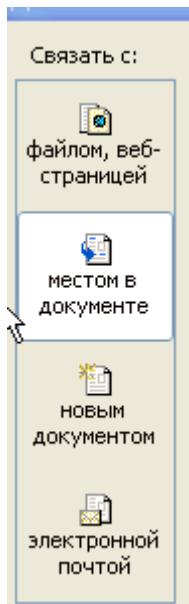


Рис. 20

Практическая работа 4.4.6.

Тема: Анимации графики средствами Power Point.

Задание:

Создать изображение по образцу из стандартных автофигур, которое будет анимировано появляться на слайде.

Порядок выполнения работы:

1. Создайте в Power Point чистый слайд (по желанию можно вставить заголовок).
2. Средствами рисования **Автофигуры** создайте рисунок как показано на рисунке 21.
3. Средствами анимации добейтесь чтобы все составляющие элементы появлялись по очереди: сначала точки, потом линии, потом окружность и собирались в единый рисунок.
4. Оформите слайд.

5. Покажите слайд преподавателю и сохраните в свою папку.

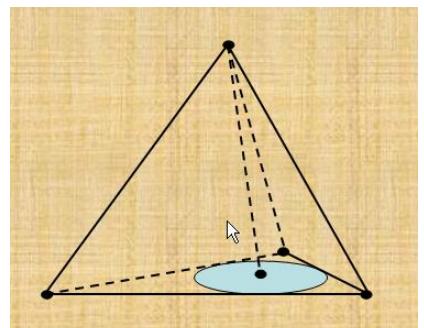


Рис. 21

Лабораторная работа 5.2.1.

Тема: Создание WEB-документа с использованием основных форматирующих тэгов HTML

Цель занятия: Получить практические навыки по созданию WEB—документов.

Структура Web-страницы. Большая часть тэгов образует контейнер, состоящий из открывающего и закрывающего тэгов. Тэги можно набирать как заглавными, так и строчными буквами.

Web-страница помещается в контейнер <HTML></HTML> и состоит из двух частей: заголовка и отображаемого в браузере содержания.

Заголовок страницы помещается в контейнер <HEAD></HEAD>. Заголовок содержит название страницы, которое помещается в контейнер <TITLE></TITLE> и при просмотре отображается в верхней строке окна браузера.

Также в заголовок помещаются не отображаемые при просмотре мета-тэги, задающие кодировку страницы для ее правильного отображения в браузере, а также содержащие описание и ключевые слова страницы, которые в первую очередь просматривают роботы поисковых систем.

Отображаемое в браузере содержание страницы помещается в контейнер <BODY></BODY>.

Создание Web-страницы «Основные тэги HTML»

- 1 Запустить текстовый редактор Блокнот командой [Пуск - Программы - Стандартные - Блокнот].

Ввести HTML-код, задающий структуру Web-страницы:

```
<HTML>
<HEAD>
2 <TITLE>Первое знакомство с тэгами HTML</TITLE>
</HEAD>
<BODY>

</BODY>
</HTML>
```

- 3 Ввести команду [Файл - Сохранить]. Файлу Web-страницы присвоить имя index.htm.
- 4 Запустить браузер и открыть созданный файл командой [Файл - Открыть]. В заголовке окна браузера высвечивается название Web-страницы *Первое знакомство с тэгамиHTML*.

Заголовки. Внести в текст страницы тэги заголовков различных уровней (размеров).

- 5 Заголовки различных уровней:

```
<H1>Заголовок первого уровня</H1>
<H2>Заголовок второго уровня</H2>
<H3>Заголовок третьего уровня</H3>
<H4>Заголовок четвертого уровня</H4>
<H5>Заголовок пятого уровня</H5>
<H6>Заголовок шестого уровня</H6>
```

Форматирование шрифта. Внести в текст страницы тэги, определяющие начертание шрифта и горизонтальных разделительных линий. Отделить этот фрагмент от остального текста с помощью горизонтальных разделительных линий.

- 6 Разделительная линия:

```
<HR>
```

Форматирование шрифта:

```
<B>Жирный</B>
<I>Курсив</I>
<U>Подчеркнутый</U>
<B><I><U>Жирный подчеркнутый курсив</B></I></U>
```

<TT>Равношириный</TT>

Выделение:

Выделение

Усиленное выделение

<HR>

Списки. Внести в текст страницы тэги, задающие списки нумерованные и ненумерованные, а также списки определений.

7 Нумерованный список:

```
<OL>
<LI>Первый элемент списка</LI>
<LI>Второй элемент списка</LI>
<LI>Третий элемент списка</LI>
</OL>
```

Ненумерованный список:

```
<UL>
<LI>Первый элемент списка</LI>
<LI>Второй элемент списка</LI>
<LI>Третий элемент списка</LI>
</UL>
```

Список определений:

```
<DL>
<DT>ТЕРМИН 1</DT>
<DD>Пояснение к термину 1</DD>
<DT>ТЕРМИН 2</DT>
<DD>Пояснение к термину 2</DD>
<DT>ТЕРМИН 3</DT>
<DD>Пояснение к термину 3</DD>
</DL>
```

Внесение изменений и дополнений в Web-страницу. В процессе создания Web-страницы приходится добавлять новые тэги и просматривать получаемый результат.

8 Активизировать Блокнот с открытой в нем редактируемой Web-страницей. Внести в содержимое страницы необходимые изменения и сохранить новый вариант страницы, выполнив команду меню [Файл - Сохранить].

9 Активизировать браузер с открытым в нем предыдущим вариантом страницы. Щелкнуть по кнопке *Обновить*. В окне браузера отобразится обновленная Web-страница.

Мета-тэги. В раздел заголовка Web-страницы могут быть добавлены информационные одиночные тэги <META>, имеющие атрибуты NAME, HTTP-EQUIV и CONTENT.

Мета тэг может информировать браузер о кодировке Web-страницы:

Мета тэги используются поисковыми системами для индексирования содержания, ключевых слов и автора Web-страницы:

```
<meta name="Description" content="">
<meta name="Keywords" content="">
<meta name="Author" content="">
```

Лабораторная работа 5.2.2.

Тема: Создание Web-сайта.

Цели занятия: научиться создавать мультимедийные и интерактивные сайты.

1. Подготовительный этап:

Ставиться общее задание для всей группы, определяются условия работы и ресурсы.

Задание:

Создание сайта по определенной теме.

2. Условия выполнения работы:

- Каждый выполняет только свою часть задания.
- Старший координирует работу группы

Ресурсы:

- Электронные копии учебников
- Электронные копии документации на ТСИ
- Конспекты лекций
- Информация в сети интернет

3. Организационный этап игры

1. Деление на группы по 5 человек (по желанию).

2. Выбор старшего в каждой группе.

3. Распределение ролей (обязанностей) для выполнения работы

Роли в группе (один из 4 вариантов):

Вариант 1.

Старший - координатор проекта

Остальные - равноправные члены рабочей группы.

Все этапы работы выполняются по мере возникновения каждым студентом. Предварительно старший распределяет подразделы темы между участниками. Коллективно выбирают стиль и форматы оформления проекта.

Вариант 2.

Старший - координатор проекта

Дизайнер - оформление проекта

Сборщик информации - поиск информации по теме

Сборщик проекта – объединение материалов

Аниматор

Все этапы работы выполняются последовательно, после завершения предыдущего этапа

Вариант 3.

Старший - координатор и редактор проекта

2 сборщика проекта

2 сборщика информации

Старший делит группу на 2 подгруппы, распределяет подразделы темы между ними. Все этапы работы выполняются по мере возникновения каждым студентом.

Вариант 4.

Старший - координатор

Редактор по графике дизайн и анимация

Сборщик информации - поиск информации

2 сборщика проекта

Старший распределяет подразделы темы между сборщиками проекта. Этапы работы выполняются последовательно, после завершения предыдущего этапа.

4. Получите задание и тему вашего проекта.

Таблица файла «результаты работы»

Критерий оценки	баллы	Само-оценка	группа 2	группа 3	группа 4	группа 5
Анимация	1-5					
Дизайн	1-5					
Содержание	1-5					
Читаемость информации	1-5					
Время выполнения	-1, 0, +1					
Всего	21	0	0	0	0	0
ОЦЕНКА	0	0	0	0	0	0

5. Итоги урока

1. Старший знакомит группу с оценкой за работу.

2. Подведение итогов совместно студентами и преподавателем:

- получили практический опыт работы в небольшом коллективе;
- освоили новый программный продукт;
- научились совместно обрабатывать информационный контент в сети в режиме онлайн;
- закрепили навыки, необходимые для развития профессиональных компетенций специальности 230701.