

**МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНО-СПЕЦИАЛЬНОГО  
ОБРАЗОВАНИЯ РЕСПУБЛИКИ УЗБЕКИСТАН**

**Самаркандский государственный университет**

**Учебно-методический комплекс по предмету  
“Техническое и программное  
обеспечение компьютеров”  
для студентов 1-курса по специальности  
“Методика преподавание информатики”  
Бустанов Х.А.**



**Самарканд-2019**

**Министерство высшего и среднего специального образования  
республики Узбекистан  
Самаркандский государственный университет  
Факультет прикладной математики и информатики  
Кафедра «Информационных технологий»**

**Утверждаю:**

**Проректор по учебной части**

\_\_\_\_\_ **проф. А. Солеев**

**«\_\_» \_\_\_\_\_ 2019 г.**

**Учебно-методический комплекс по предмету  
“Техническое и программное  
обеспечение компьютеров”  
для студентов 1-курса по специальности  
“Методика преподавание информатики”  
*Бустанов Х.А.***

**Область знаний:** 100000 – Гуманитарная сфера  
**Направление образования:** Методика преподавание информатики

**Самарканд - 2019**

*Учебно-методический комплекс предмета “Техническое и программное обеспечения компьютеров” разработана согласно с учебно-рабочей программой.*

**Составили :**

Доцент кафедры «Информационных технологий» СамГУ доц. И.Б. Аминов и старший преподаватель Х.А.Бустанов.

**Рецензенты:**

Доцент кафедры «Информатика» Самаркандского филиала УТИТ А.Каршиев.  
Доцент кафедры «Информационных технологий» СамГУ Абдуллаев А.

*Учебно-методический комплекс предмета обсуждена и утверждена на собрании кафедры «Информационных технологий» « \_\_\_\_ » августа 2019 года №1.*

**Заведующей кафедрой:**

**проф. И. И. Жуманов**

*Учебно-методический комплекс предмета обсуждена и утверждена на учебно-методическом Совете факультета прикладной математики и информатики « \_\_\_\_ » \_\_\_\_\_ 2019 года №1.*

**Председатель методического совета факультета**

**Ш.Маматов**

**Председатель Совета факультета:**

**доц. А. Бобоеров**

**Согласовал:**

**Начальник учебно-методического  
управлений :**

**Аликулов Б.**

**Календарный рабочий план темы лекционных занятий по предмету  
“Техническое и программное обеспечение компьютеров”**

<b>№</b>	<b>Темы</b>	<b>Часы</b>
1.	Основы предмета “Техническое и программное обеспечение компьютеров”. Истории развития ВТ и их поколений.	2
2.	Основные устройства компьютеров. Диски. Принтеры и другие устройства компьютеров.	2
3.	Характеристики основных и дополнительных устройств компьютеров а также их назначений.	2
4.	Програмное обеспечения компьютеров. Типы и характеристики микропроцессоров, Intel, Intel Celeron, Intel Core i7.	2
5.	Параллельные процессоры в вычислительных системах. SPMD и другие.	2
6.	Память компьютеров и их типы. Носители информации, диски и их разновидности.	2
7.	Системные программное обеспечение и их основные функции.	2
8.	Операционные системы современных компьютеров, примеры.	2
9.	Служебные программы для файлов и дисков компьютера.	2
10.	Программы архиваторов и антивирусные программы, примеры.	2
11.	Прикладные программы и их назначение, примеры Word, Excel, PowerPoint, Access, Outlook Express и другие.	2
12.	Офисные программы Microsoft Office 2007-10, Microsoft Office 2016.	2
13.	Возможности и принципы работы Microsoft Word, Microsoft Excel, Microsoft PowerPoint, Microsoft Access.	2
14.	Графические возможности компьютеров, графические редакторы и примеры Microsoft PowerPoint и другие.	2
15.	Основы работы с графическим редактором Corel Draw.	2
16.	Применение прикладных программ в различные направлениях.	2
17.	Прикладные программы для решения математических задач.	2
18.	Технических и программных средств для ИТ, основы работы в интернет.	2
19.	Основы программирования и языки программирования, типы.	2
20.	Основы языка программирования C++, Visual Basic и ее возможности.	2
21.	Основы языка программирования Delphi и C++, возможности.	2
22.	Массивы и графические возможности языка программирования C++, Delphi.	2
	Всего	44

## Оглавление

<b>№:</b>	<b>Тема</b>	<b>Стр.</b>
1	<b>Введение. Основы предмета ТПОК. Устройства ИТ и их назначения. Формирование и единицы измерения информации.</b>	6
2	<b>Основные устройства компьютеров. Память компьютеров и их виды. Диски. Принтеры и другие устройства компьютеров.</b>	16
3	<b>Программное обеспечение. Операционные системы. Понятие файла, папки и дисков. Функции операционной системы Windows.</b>	20
4	<b>Основные элементы ОС Windows и рабочий стол. Возможности Windows. Стандартные программы ОС Windows.</b>	36
5	<b>Панель инструментов Microsoft Word. Работа с графическими объектами и с таблицами в Microsoft Word</b>	46
6	<b>Работа с гиперссылками в Microsoft Word</b>	54
7	<b>Создание презентаций в Power Point. Меню Дизайн</b>	81
8	<b>Меню «Анимация» в PowerPoint.</b>	88
9	<b>Технологии демонстрации документов в компьютерах.</b>	92
10	<b>Электронная таблица Excel. Окно Excel.</b>	96
11	<b>Работа с формулами в Excel.</b>	105
12	<b>Создание графиков функций в Excel. Работа с диаграммами.</b>	116
13	<b>Компьютерная графика. Программа AdobePhtotoshop.</b>	124
14	<b>Инструменты AdobePhtotoshop. Меню «Фильтр»</b>	131
15	<b>Создание новой кисти в AdobePhtotoshop</b>	138
16	<b>Основы работы в Интернет - www (wordwideweb). Браузеры и провайдеры Интернет. СайтыИнтернет.</b>	142
17	<b>Электронная почта Интернет (E-mail).</b>	149
18	<b>Основы работы в электронных почтах сайтов Интернет.</b>	155
19	<b>Вирусы и их разновидности.</b>	159
20	<b>Защита компьютера от вирусов. Антивирусные программы.</b>	165
21	<b>Программы архиваторы и архивация файлов и папок. Язык C++.</b>	171
22	<b>Список литературы и сайтов</b>	180

## Тема-1: Введение. Основы предмета ТПОК. Понятие информация. Современные компьютеры и их обеспечение.

### План:

1. Основы предмета информатики.
2. Понятие информация, создание, обработки, прием и передача информации.
3. История развития вычислительной техники
4. Устройства информационных технологий и их назначения.

**Ключевые слова:** информационная технология, технология, язык пользователя, единицы информации, свойства информации, информационные ресурсы, информационная технология

**Ключевые слова:** Информатика, Информация, Вычислительная техника, компьютер.

### Введение.

Общим положительным фактором стало распространение высокоскоростного доступа к Интернету, развитие мобильной связи, реализация соответствующих государственных программ и принятие новых правовых актов в области ИКТ. Развитию информационного общества способствовало также постепенное дерегулирование рынка телекоммуникаций. Особое значение имела разработка и реализация правительственных программ развития ИКТ и укрепление информационной инфраструктуры.

Что такое информатика?

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

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

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

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

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

Информатика — комплексная научная дисциплина с широчайшим диапазоном применения. Её **приоритетные направления**:

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

Академик А.А. Дородницын выделяет в информатике три неразрывно и существенно связанные части — **технические средства, программные и алгоритмические.**

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

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

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

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

### **Что такое информация?**

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. В настоящее время наука пытается найти общие свойства и закономерности, присущие многогранному понятию *информация*, но пока это понятие во многом остается интуитивным и 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.**

Приведем еще несколько определений:

- *Информация — это сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые уменьшают имеющуюся о них степень неопределенности, неполноты знаний* (Н.В. Макарова);

- *Информация — это отрицание энтропии* (Леон Бриллюэн);

- *Информация — это мера сложности структур* (Моль);

- *Информация — это отраженное разнообразие* (Урсул);

- *Информация — это содержание процесса отражения* (Тузов);

- *Информация — это вероятность выбора* (Яглом).

Современное научное представление об информации очень точно сформулировал **Норберт Винер**, "отец" кибернетики. А именно:

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

**Информация есть характеристика не сообщения, а соотношения между сообщением и его потребителем. Без наличия потребителя, хотя бы потенциального, говорить об информации бессмысленно.**

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 символических 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.**

В каком виде существует информация?

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, посредством которых передаются по наследству признаки и свойства организмов и т.д.

Предметы, процессы, явления материального или нематериального свойства, рассматриваемые с точки зрения их информационных свойств, называются информационными объектами.

**Как передаётся информация?**

Информация передаётся в форме **сообщений** от некоторого **источника** информации к её **приёмнику** посредством **канала связи** между ними. Источник посылает **передаваемое сообщение**, которое **кодируется в передаваемый сигнал**. Этот сигнал посылается по **каналу связи**. В результате в приёмнике появляется **принимаемый сигнал**, который **декодируется** и становится **принимаемым сообщением**.

Примеры:

1. *Сообщение, содержащее информацию о прогнозе погоды, передаётся приёмнику (телезрителю) от источника — специалиста-метеоролога посредством канала связи — телевизионной передающей аппаратуры и телевизора.*

2. *Живое существо своими органами чувств (глаз, ухо, кожа, язык и т.д.) воспринимает информацию из внешнего мира, перерабатывает её в определенную последовательность нервных импульсов, передает импульсы по нервным волокнам, хранит в памяти в виде состояния нейронных структур мозга, воспроизводит в виде звуковых сигналов, движений и т.п., использует в процессе своей жизнедеятельности.*

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

**История развития вычислительной техники**

Первым устройством, предназначенным для облегчения счета, были счеты. С помощью костяшек счетов можно было совершать операции сложения и вычитания и несложные умножения.

1642 г. — французский математик Блез Паскаль сконструировал первую механическую счетную машину «Паскалина», которая могла механически выполнять сложение чисел.

1673 г. — Готфрид Вильгельм Лейбниц сконструировал арифмометр, позволяющий механически выполнять четыре арифметических действия.

1941 г. — немецкий инженер Конрад Цузе построил небольшой компьютер на основе нескольких электромеханических реле.

1943 г. — в США на одном из предприятий фирмы IBM Говард Эйкен создал компьютер под названием «Марк-1».

1943 г. — в США группа специалистов под руководством Джона Мочли и Проспера Экерта начала конструировать компьютер ENIAC на основе электронных ламп.

1945 г. — к работе над ENIAC был привлечен математик Джон фон Нейман, который подготовил доклад об этом компьютере. В своем докладе фон Нейман сформулировал общие принципы функционирования компьютеров, т. е. универсальных вычислительных устройств. До сих пор подавляющее большинство компьютеров сделано в соответствии с теми принципами, которые изложил Джон фон Нейман.

1947 г. — Экертом и Мочли начата разработка первой электронной серийной машины UNIVAC (Universal Automatic Computer). Первый образец машины (UNIVAC-1) был построен для бюро переписи США и пущен в эксплуатацию весной 1951 г. Синхронная, последовательного действия вычислительная машина UNIVAC-1 была создана на базе ЭВМ ENIAC и EDVAC. Работала она с тактовой частотой 2,25 МГц и содержала около 5000 электронных ламп. Внутреннее запоминающее устройство емкостью 1000 12-разрядных десятичных чисел было выполнено на 100 ртутных линиях задержки.

1949 г. — английским исследователем Морнсом Уилксом построен первый компьютер, в котором были воплощены принципы фон Неймана.

1951 г. — Дж. Форрестер опубликовал статью о применении магнитных сердечников для хранения цифровой информации, в машине «Whirlwind-1» впервые была применена память на магнитных сердечниках. Она представляла собой 2 куба с 32-32-17 сердечниками, которые обеспечивали хранение 2048 слов для 16-разрядных двоичных чисел с одним разрядом контроля на четность.

1952 г. — фирма IBM выпустила свой первый промышленный электронный компьютер IBM 701, который представлял собой синхронную ЭВМ параллельного действия, содержащую 4000 электронных ламп и 12 000 диодов. Усовершенствованный вариант машины IBM 704 отличался высокой скоростью работы, в нем использовались индексные регистры и данные представлялись в форме с плавающей запятой.

После ЭВМ IBM 704 была выпущена машина IBM 709, которая в архитектурном плане приближалась к машинам второго и третьего поколений. В этой машине впервые была применена косвенная адресация и впервые появились каналы ввода — вывода.

1952 г. — фирма Remington Rand выпустила ЭВМ UNIVAC-t 103, в которой впервые были применены программные прерывания. Сотрудники фирмы Remington Rand использовали алгебраическую форму записи алгоритмов под названием «Short Code» (первый интерпретатор, созданный в 1949 г. Джоном Мочли).

1956 г. — фирмой IBM были разработаны плавающие магнитные головки на воздушной подушке. Изобретение их позволило создать новый тип памяти — дисковые запоминающие устройства (ЗУ), значимость которых была в полной мере оценена в последующие десятилетия развития вычислительной техники. Первые ЗУ на дисках появились в машинах IBM 305 и RAMAC. Последняя имела пакет, состоявший из 50 металлических дисков с магнитным покрытием, которые вращались со скоростью 12000 об./мин. На поверхности диска размещалось 100 дорожек для записи данных, по 10 000 знаков каждая.

1956 г. — фирма Ferranti выпустила ЭВМ «Pegasus», в которой впервые нашла воплощение концепция регистров общего назначения (РОН). С появлением РОН устранено различие между индексными регистрами и аккумуляторами, и в распоряжении программиста оказался не один, а несколько регистров-аккумуляторов.

1957 г. — группа под руководством Д. Бэкуса завершила работу над первым языком программирования высокого уровня, получившим название ФОРТРАН. Язык, реализованный впервые на ЭВМ IBM 704, способствовал расширению сферы применения компьютеров.

1960-е гг. — 2-е поколение ЭВМ, логические элементы ЭВМ реализовываются на базе полупроводниковых приборов-транзисторов, развиваются алгоритмические языки программирования, такие как Алгол, Паскаль и другие.

1970-е гг. — 3-е поколение ЭВМ, интегральные микросхемы, содержащие на одной полупроводниковой пластине тысячи транзисторов. Начали создаваться ОС, языки структурного программирования.

1974 г. — несколько фирм объявили о создании на основе микропроцессора Intel-8008 персонального компьютера — устройства, выполняющего те же функции, что и большой компьютер, но рассчитанного на одного пользователя.

1975 г. — появился первый коммерчески распространяемый персональный компьютер Альтаир-8800 на основе микропроцессора Intel-8080. Этот компьютер имел оперативную память всего 256 байт, клавиатура и экран отсутствовали.

Конец 1975 г. — Пол Аллен и Билл Гейтс (будущие основатели фирмы Microsoft) создали для компьютера «Альтаир» интерпретатор языка Basic, позволивший пользователям просто общаться с компьютером и легко писать для него программы.

Август 1981 г. — компания IBM представила персональный компьютер IBM PC. В качестве основного микропроцессора компьютера использовался 16-разрядный микропроцессор Intel-8088, который позволял работать с 1 мегабайтом памяти.

1980-е гг. -1990-е гг. — 4-е поколение ЭВМ, построенное на больших интегральных схемах. Микропроцессоры реализовываются в виде единой микросхемы, Массовое производство персональных компьютеров.

2000-е гг. — поколение ЭВМ. Интеграция ЭВМ и бытовой техники, встраиваемые компьютеры, развитие сетевых вычислений.

### ***Компьютеры и их виды. Периферийное оборудование.***

В связи с этим под компьютерным аппаратным обеспечением, являющимся, по определению, неотъемлемой частью средств ИКТ, используемых в общем среднем образовании, целесообразно понимать персональные компьютеры и другие аппаратные устройства, работающие во взаимодействии с ними.

### **Понятие информационной технологии.**

#### **Что такое информационная технология?**

**Технология** — это комплекс научных и инженерных знаний, реализованных в приемах труда, наборах материальных, технических, энергетических, трудовых факторов производства, способах их соединения для создания продукта или услуги, отвечающих определенным требованиям. Поэтому технология неразрывно связана с механизацией производственного или непромышленного, прежде всего управленческого процесса. Управленческие технологии основываются на применении компьютеров и телекоммуникационной техники.

### **Этапы развития информационных технологий**

Существует несколько точек зрения на развитие информационных технологий с использованием компьютеров, которые определяются различными признаками деления.

Общим для всех изложенных ниже подходов является то, что с появлением персонального компьютера начался новый этап развития информационной технологии. Основной целью становится удовлетворение персональных информационных потребностей человека как для профессиональной сферы, так и для бытовой.

Признак деления - вид задач и процессов обработки информации

1-й этап (60 - 70-е гг.) — обработка данных в вычислительных центрах в режиме коллективного пользования. Основным направлением развития информационной технологии являлась автоматизация операционных рутинных действий человека.

2-й этап (с 80-х гг.) — создание информационных технологий, направленных на решение стратегических задач.

Признак деления — проблемы, стоящие на пути информатизации общества

1-й этап (до конца 60-х гг.) характеризуется проблемой обработки больших объемов данных в условиях ограниченных возможностей аппаратных средств.

2-й этап (до конца 70-х гг.) связывается с распространением ЭВМ серии IBM/360. Проблема этого этапа - отставание программного обеспечения от уровня развития аппаратных средств.

3-й этап (с начала 80-х гг.) - компьютер становится инструментом непрофессионального пользователя, а информационные системы – средством поддержки принятия его решений. Проблемы- максимальное удовлетворение потребностей пользователя и создание соответствующего интерфейса работы в компьютерной среде.

4-й этап (с начала 90-х гг.) - создание современной технологии меж организационных связей и информационных систем. Проблемы этого этапа весьма многочисленны. Наиболее существенными из них являются:

- выработка соглашений и установление стандартов, протоколов для компьютерной связи;
- организация доступа к стратегической информации;
- организация защиты и безопасности информации.

Начинают широко использоваться в различных областях глобальные и локальные компьютерные сети.

### **Формирование и единицы измерения информации.**

В качестве единицы информации Клод Шеннон предложил принять один **бит** (англ. *bit* — *binary digit* — двоичная цифра).

**Бит в теории информации** — количество информации, необходимое для различения двух равновероятных сообщений (типа "орел" — "решка", "чет" — "нечет" и т.п.).

**В вычислительной технике** битом называют наименьшую "порцию" памяти компьютера, необходимую для хранения одного из двух знаков "0" и "1", используемых для внутри машинного представления данных и команд.

Бит — слишком мелкая единица измерения. На практике чаще применяется более крупная единица — **байт**, равная **восемью битам**. Именно восемь битов требуется для того, чтобы закодировать любой из 256 символов алфавита клавиатуры компьютера ( $256=2^8$ ).

Широко используются также ещё более крупные производные единицы информации:

- **1 Килобайт (Кбайт) = 1024 байт =  $2^{10}$  байт,**
- **1 Мегабайт (Мбайт) = 1024 Кбайт =  $2^{20}$  байт,**
- **1 Гигабайт (Гбайт) = 1024 Мбайт =  $2^{30}$  байт.**

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

- **1 Терабайт (Тбайт) = 1024 Гбайт =  $2^{40}$  байт,**
- **1 Петабайт (Пбайт) = 1024 Тбайт =  $2^{50}$  байт.**

За единицу информации можно было бы выбрать количество информации, необходимое для различения, например, десяти равновероятных сообщений. Это будет не двоичная (бит), а десятичная (дит) единица информации.

### **Что можно делать с информацией?**

Информацию можно:

- |                 |                    |              |
|-----------------|--------------------|--------------|
| • создавать;    | • формализовать;   | • собирать;  |
| • передавать;   | • распространять;  | • хранить;   |
| • воспринимать; | • преобразовывать; | • искать;    |
| • использовать; | • комбинировать;   | • измерять;  |
| • запоминать;   | • обрабатывать;    | • разрушать; |
| • принимать;    | • делить на части; | • и др.      |
| • копировать;   | • упрощать;        | •            |

Все эти процессы, связанные с определенными операциями над информацией, называются **информационными процессами**.

### **Какими свойствами обладает информация?**

Свойства информации:

- достоверность;
- полнота;
- ценность;
- своевременность;
- понятность;
- доступность;
- краткость;
- 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.** Одинаково 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.

**Информация должна преподноситься в доступной** (по уровню восприятия) форме. Поэтому одни и те же вопросы по разному излагаются в школьных учебниках и научных изданиях.

**Информацию по одному и тому же вопросу можно изложить кратко** (сжато, без несущественных деталей) **или пространно** (подробно, многословно). Краткость информации необходима в справочниках, энциклопедиях, учебниках, всевозможных инструкциях.

Что такое обработка информации?

**Обработка информации** — получение одних информационных объектов из других информационных объектов путем выполнения некоторых алгоритмов

Обработка является одной из основных операций, выполняемых над информацией, и главным средством увеличения объёма и разнообразия информации.

**Средства обработки информации** — это всевозможные устройства и системы, созданные человечеством, и в первую очередь, компьютер — универсальная машина для обработки информации.

**Компьютеры обрабатывают информацию** путем выполнения некоторых алгоритмов.

**Живые организмы и растения обрабатывают информацию** с помощью своих органов и систем.

## Что такое информационные ресурсы и информационные технологии?

**Информационные ресурсы** — это идеи человечества и указания по их реализации, накопленные в форме, позволяющей их воспроизводство.

Это книги, статьи, патенты, диссертации, научно-исследовательская и опытно-конструкторская документация, технические переводы, данные о передовом производственном опыте и др.

**Информационные ресурсы** (в отличие от всех других видов ресурсов — трудовых, энергетических, минеральных и т.д.) **тем быстрее растут, чем больше их расходуют.**

**Информационная технология** — это совокупность методов и устройств, используемых людьми для обработки информации.

Человечество занималось обработкой информации тысячи лет. Первые информационные технологии основывались на использовании счётов и письменности. Около пятидесяти лет назад началось исключительно быстрое развитие этих технологий, что в первую очередь связано с появлением компьютеров.

В настоящее время термин "**информационная технология**" употребляется в связи с использованием компьютеров для обработки информации. Информационные технологии охватывают всю вычислительную технику и технику связи и, отчасти, — бытовую электронику, телевидение и радиовещание.

Они находят применение в промышленности, торговле, управлении, банковской системе, образовании, здравоохранении, медицине и науке, транспорте и связи, сельском хозяйстве, системе социального обеспечения, служат подспорьем людям различных профессий и домохозяйкам.

Народы развитых стран осознают, что **совершенствование информационных технологий представляет самую важную, хотя дорогостоящую и трудную задачу.**

В настоящее время создание крупномасштабных информационно-технологических систем является экономически возможным, и это обуславливает появление национальных исследовательских и образовательных программ, призванных стимулировать их разработку.

## Что понимают под информатизацией общества?

**Информатизация общества** — организованный социально-экономический и научно-технический процесс создания оптимальных условий для удовлетворения информационных потребностей и реализации прав граждан, органов государственной власти, органов местного самоуправления организаций, общественных объединений на основе формирования и использования информационных ресурсов

**Цель информатизации** — улучшение качества жизни людей за счет увеличения производительности и облегчения условий их труда.

Информатизация — это сложный социальный процесс, связанный со значительными изменениями в образе жизни населения. Он требует серьёзных усилий на многих направлениях, включая ликвидацию компьютерной неграмотности, формирование культуры использования новых информационных технологий и др.

## **Тема-2: Основные устройства компьютеров. Память компьютеров и их виды. Диски. Принтеры и другие устройства компьютеров.**

### **План:**

1. Основные устройства компьютеров. Память компьютеров и их виды. Диски.
2. Принтеры и другие устройства компьютеров.

**Ключевые слова:** корпус, блок питания, материнская плата, процессор, оперативная память, жесткий диск или винчестер, видеокарта, звуковая карта, сетевая плата, принтер, плоттер, сканер.

### **Основные устройства компьютера и их функции.**

Процессор является главным устройством компьютера, в котором собственно и происходит обработка всех видов информации. Другой важной функцией процессора является обеспечение согласованного действия всех узлов, входящих в состав компьютера. Соответственно наиболее важными частями процессора являются арифметико-логическое устройство АЛУ и устройство управления УУ.

Теперь, когда мы знаем основные устройства компьютера и их функции, осталось выяснить, как они взаимодействуют между собой. Для этого обратимся к функциональной схеме современного компьютера, приведенной на рисунке.

Для связи основных устройств компьютера между собой используется специальная информационная магистраль, обычно называемая инженерами шиной. Шина состоит из трех частей:

- шина адреса, на которой устанавливается адрес требуемой ячейки памяти или устройства, с которым будет происходить обмен информацией;
- шина данных, по которой собственно и будет передана необходимая информация; и, наконец,
- шина управления, регулирующей этот процесс (например, один из сигналов на этой шине позволяет компьютеру различать между собой адреса памяти и устройств ввода/вывода).

### **Внутреннее устройство:**

После рассмотрения устройств ввода вывода, пришло время изучить внутреннее устройство компьютера, а именно составляющие элементы системного блока, комплектующие.

**Корпус.** Внешняя оболочка системного блока. В нем находятся все комплектующие ПК. Корпус снабжен системой охлаждения и вентиляции. Довольно важный элемент в устройстве компьютера. От него зависит качество установки и сборки компьютера. А от толщины стенок и системы охлаждения, зависит уровень шума и вибрации. Это надо учитывать, при выборе корпуса. Важно, чтобы все внутреннее устройство компьютера хорошо охлаждалось, и было надежно закреплено. **Блок питания.** Этот представитель внутреннего устройства компьютера, обычно идет в комплекте с корпусом. Задача, блока питания, обеспечить все внутренние элементы электричеством, нужного тока и напряжения. Чем, мощнее ваш компьютер, тем мощнее необходим блок питания, иначе ПК может даже не включиться.

**Материнская плата.** Или системная плата, элемент внутреннего устройства компьютера, который объединяет все комплектующие между собой. Плата представляет собой большое количество разъемов, к которым подключаются все остальные комплектующие. Именно от материнской платы, зависит совместимость тех или иных комплектующих.

**Процессор.** CPU (центральный процессор), обеспечивает главную обработку всей информации. Производительность во многом зависит от процессора. Именно этот элемент внутреннего устройства компьютера, отвечает за команды другим комплектующим ПК.

**Оперативная память.** Она относится к ОЗУ (оперативно запоминающие устройства), то есть эта память хранит информацию, пока она нужна компьютеру. Информация все время обновляется. От объема оперативной памяти зависит, то, какое количество информации компьютер способен держать в “уме”, а значит и быстродействие компьютера.

**Жесткий диск или винчестер.** Относится к ПЗУ (постоянно запоминающие устройства). Этот элемент внутреннего устройства ПК, отвечает за постоянное хранение информации. Винчестер работает по принципу магнитных дисков. На него можно записывать файлы для постоянного хранения.

**Видеокарта.** Также составляет внутреннее устройство компьютера. Эта плата отвечает за вывод графической информации на устройство вывода, монитор, проектор и так далее. Важный элемент для современных видео игр. Так как они используют большое количество видео ресурсов, для обеспечения красивой и реалистичной графики.

Это все обязательные элементы, которые составляют внутреннее устройство компьютера. Они есть в каждом ПК.

**Звуковая карта.** Является устройством вывода звукового сигнала на периферийные устройства вывода, акустические системы, наушники и так далее. Чаще всего, используется звуковая карта, встроенная в материнскую плату, но можно установить и отдельную.

**Сетевая плата.** Используется для соединения компьютеров в локальную сеть. Чаще, интегрирована в материнскую плату.

Дисковод и оптический привод. Необходим для взаимодействия компьютера с внешней информацией. По средствам дискет или дисков, можно заносить или выносить информацию.

**Внутренний модем.** Является устройством ввода вывода, используется для соединения с Интернетом. Другие платы расширения. Сюда можно отнести ТВ-тюнер, который обеспечивает прием телевизионных каналов, различные контроллеры и тому подобное.

В общем, все те внутренние устройства, которые расширяют функциональность компьютера. Также не нужно забывать и о системе охлаждения, она присутствует практически у каждого элемента внутреннего устройства компьютера. Система охлаждения может быть воздушной, в виде вентилятора, может быть в виде радиатора или тепловых трубок, также существует водяное охлаждение.



**ПРИНТЕР.** Это периферийное устройство предназначено для вывода на печать различной текстовой и графической

информации: фотографий, рисунков, набранных текстов, графиков, чертежей и т. п.

По возможности отображения цвета подразделяются на:

Цветные с возможностью печати не только цветных картинок, но и полноценных фотографий (фотопринтеры);

Чёрно-белые, печатающие простые тексты и графику с оттенками серого цвета, но без возможности вывода цвета.

Конструктивно, существующие на сегодняшний день печатающие устройства можно разделить на:

Матричные, где печатающая головка содержит «иголки» удар которых, через красящую ленту по бумаге и формирует знаки и изображение. Разрешение таких принтеров может достигать 300 точек на дюйм (dpi), а количество иголок в головке 9 и 24 штуки. Недостаток – невысокая скорость печати и повышенный шум, но при этом у таких устройств самая низкая стоимость отпечатка и возможность печати через копирку;

- струйные, само название которых, говорит само за себя – печать осуществляется за счёт управляемой струйки чернил при помощи сопел в печатающей головке. струйные принтеры выпускаются как чёрно-белые, так и цветные, причём цветные струйные принтеры отлично подходят для печати фотографий благодаря высокому разрешению (600-900 dpi) и отличным характеристикам цветопередачи. скорость печати достигает 10-20 страниц в минуту.

- лазерные где формирование изображение на бумаге происходит методом ксерографии. изображение переносится на бумагу при помощи лазерного луча и специального порошка, который засыпается в специальный картридж. первый в мире матричный принтер был выпущен в 1964 году. но использовался он вначале не в компьютерных системах, а для распечатки показаний точного времени ⌚ в конструкции часов компании seiko.

## ПЛОТТЕР



Это устройство хотя практически и не распространено у домашних пользователей ПК, но оказывается весьма ценным для тех, кто постоянно работает с чертежами или рисунками больших форматов. Подобно принтеру, плоттер способен выводить на бумагу выполненные на компьютере изображения – чертежи и рисунки очень больших форматов (вплоть до формата «A0»). Такими устройствами снабжаются конструкторские и архитектурные бюро, они незаменимы при печати больших, демонстрационных диаграмм, схем и рисунков для различных презентаций.

Первые модели плоттера отличались от принтера тем, что они не смешивали цвета – конструктивно они чертили заправленными каждый своим цветом «рапидографами». Можно было подбирать толщину линий, но вот цвет всегда оставался одним из установленного в плоттере набора. Сегодня, последние модели полностью лишены этого недостатка.

## СКАНЕР



Как и принтер, сканер является на сегодняшний день одним из самых популярных устройств, практически у всех пользователей ПК, как офисных и производственных, так и домашних. Предназначен он для сканирования (снятия визуальной копии) любых изображений и сохранения их в цифровом виде на каком-либо носителе для хранения данных. Сегодняшние сканеры, пройдя путь от медленных и примитивных устройств, сегодня являются образцом производственных технологий и научных достижений.

Большинство моделей имеют впечатляющие технические характеристики, которые определяют качество получаемых копий:

Конструктивно, все сканеры делятся на:

Ручные. Изображение сканируется при проведении по нему сканером;

Планшетные. Отлично подходят для домашнего и офисного использования при сканировании как фотографий, так и любых, других типов изображений;

Листопротяжные. Сканирование осуществляется при протягивании листа через специальное считывающее устройство. По параметрам схожи со сканерами планшетного типа;

Барабанные, применяются исключительно в полиграфических целях, так как обладают высокопрофессиональными параметрами и очень дороги.



## УСТРОЙСТВА ЗВУКОВОСПРОИЗВЕДЕНИЯ

Уже много лет все производители материнских плат включают в их состав звуковую карту. То есть практически все ПК способны воспроизводить звуковые и музыкальные файлы, прослушивать музыку и смотреть видеофильмы со звуком. Кроме этого, как правило, на материнской плате имеются и звуковые входы для подключения внешних устройств – микрофона, магнитофона, внешнего CD и DVD проигрывателя, других звуковоспроизводящих устройств.



Всегда обращайте внимание на материал амбушюр – они непосредственно прилегают к ушам, и от этого будет зависеть насколько комфортно вам будет при прослушивании, особенно в течении длительного времени.

Микрофоны для записи или передачи голоса также доступны для приобретения в отдельном порядке. Моделей существует большое множество – от настольных, до имеющих крепление на монитор.



Если использование микрофона предполагается только для проведения конференций и разговоров по сети, то имеет смысл приобрести веб-камеру, которая имеет в своём составе кроме самой камеры, ещё и встроенный микрофон. В этом случае можно сэкономить не только на рабочем пространстве возле монитора, но и денежные средства.

Подключается веб-камера к ПК через USB-порт, при этом для получения качественного изображения, версия USB-порта должны быть не ниже 2.0. ИГРОВЫЕ МАНИПУЛЯТОРЫК устройствам компьютерной периферии относятся и игровые манипуляторы – джойстики, игровые комплекты – «руль-педали», и другие приспособления. Подключаются они к порту USB ПК и



предназначены для имитации реальных приборов управления в игровых приложениях.



### «СВЕТОВОЕ ПЕРО»

Устройство выполняется в виде обычного карандаша или ручки, которая удобно ложится в руку и связана с ПК проводным или беспроводным способом. Она в точности копирует все движения пользователя по специальному коврику или экрану монитора, позволяя, таким образом, выполнять рисунок привычным способом – рисуя. Кроме этого, «световое перо» может с успехом играть и роль манипулятора наподобие компьютерной «мыши».



### ДИГИТАЙЗЕР

Представляет собой графический планшет, подключаемый к персональному компьютеру и предназначенный для ввода графических данных, которые пользователь пишет или рисует на его поверхности.

## **Тема-3: Программное обеспечение. Операционные системы. Примеры к операционным системам. Понятие файла, папки и дисков. Функции операционной системы Windows.**

### **План:**

1. Программное обеспечение. Операционные системы. Примеры к операционным системам
2. Понятие файла, папки и дисков.
3. Среда операционной системы Windows.
4. Функции операционной системы Windows.

### **Программное обеспечение и операционная система ПК**

Все многообразие программ, используемых на современном компьютере, называется программным обеспечением (ПО).

Программное обеспечение компьютера можно разделить на системное ПО, системы программирования и прикладное программное обеспечение.

Две первые группы иногда называют базовым программным обеспечением. Ядром системного ПО является операционная система (ОС).

ОС - это часть программного обеспечения компьютера, наиболее тесно связанная с его техническими средствами и управляющая всеми его аппаратными компонентами.



### **Основные функции ОС:**

- 1) Управление ресурсами компьютера: процессорным временем, распределением внутренней памяти, файлами, внешними устройствами;
- 2) Организация диалога с пользователем.

А теперь подробнее. Вот некоторые причины, по которым ПК снабжаются операционными системами:

- Многие программы нуждаются в выполнении одних и тех же рутинных действий (например, чтения ввода с клавиатуры, вывода на экран, записи на диск). Чтобы не программировать их каждый раз заново, ОС предоставляет системные библиотеки часто используемых процедур.

- Одни и те же данные зачастую нужно использовать с помощью разных программ. Например, текст или изображение сначала может редактироваться, а потом отправляться по электронной почте. Следовательно, необходим единый, независимый от конкретных программ способ хранения данных. Большинство ОС отвечают на эту потребность реализацией файловых систем. Современные ОС, кроме того, непосредственно «связывают» вывод одной программы с вводом другой, минуя медленные дисковые операции.

- Один и тот же ПК может использоваться многими людьми. Владельцу компьютера нужна возможность определять полномочия каждого пользователя, например, чтобы севший вечером поиграть сынишка не «снёс» по ошибке свежее подготовленную диссертацию отца семейства или не залез в любовную переписку старшей сестры. Иногда каким-то данным желательно предоставить удалённый доступ по сети, но, конечно же, не ко всем. Современные ОС – многопользовательские. Они предоставляют систему разграничения полномочий локальных и удалённых пользователей.

- Программы, увы, слишком сложны, чтобы быть безошибочными, и люди за компьютером также не свободны от ошибок. Необходимо ограничить ущерб, который может нанести программа с ошибкой или неверное действие пользователя. Система разграничения полномочий служит для этой цели.

- Иногда даже одному пользователю желательно запускать несколько программ одновременно, а так же есть программы, которые требуется запускать автоматически по расписанию или при наступлении определённого события. Современные ОС – это многозадачные ОС с разделением времени. Их специальный компонент – планировщик – делит процессорное время на короткие (сотые или тысячные доли секунды) промежутки и поочередно передаёт управление разным процессам (запущенным программам), так что возникает иллюзия «одновременной» работы нескольких программ.

@ Файл – это совокупность данных, записанных на внешний носитель, которая имеет определенное имя.

Английское слово file переводится буквально как «шпилька для накалывания бумаг». Действительно, данные собираются в один информационный блок и оформляются в виде именного файла.

В существующий файл может быть записана дополнительная информация либо из него могут быть удалены ненужные данные.

С файлами можно выполнять и другие операции, связанные с различными информационными процессами: поиск конкретных данных в файлах, копирование данных и т.д.

Файл является универсальным информационным объектом. Любая работа, выполненная на компьютере (текст, рисунки, аудио- и видеозапись, база данных, программа), оформляется в виде файлов.

@ Имя файла состоит из двух частей, разделенных точкой: собственно имя файла и расширение, определяющее его тип (программа, данные и т. д.). Эти две части разделены между собой точкой. Собственно имя файлу дает пользователь, а тип файла обычно задается программой автоматически при его создании.

В различных операционных системах существуют различные форматы имен файлов. В операционной системе MS-DOS собственно имя файла должно содержать не более восьми букв латинского алфавита и цифр, а расширение состоит из трех латинских букв, например: proba.txt

В ОС Windows имя файла может иметь до 255 символов, причем допускается использование русского алфавита, например: Единицы измерения информации.doc.

Имя файла должно содержать не менее одного символа.

**Запрещается использовать в имени файла следующие символы / : \* ? " < > | .**

*Тип файла*

*Расширение*

*Исполняемые программы - exe, com*

*Текстовые файлы - txt, rtf, doc*

*Графические файлы - Bmp, gif, jpg, png, pds и др.*

*Web-страницы - htm, html*

*Звуковые файлы - wav, mp3, midi, kar, ogg*

*Видеофайлы - avi, mpg*

*Код (текст) программы на языках программирования - bas, pas, cpp и др*

@ Каталог – это раздел на диске, в котором регистрируются другие файлы, а также каталоги, вложенные в данный каталог.

В каталог ОС MS-DOS входят файлы с одинаковым расширением. Роль каталога в ОС Windows выполняют папки.

Для работы с файлами и папками в Windows предусмотрена программа Проводник, которая устанавливается на компьютер вместе с системой Windows. Интерфейс Проводника напоминает интерфейс окна Мой компьютер

### **Файловая система.**

папки и файлы операционной системы На каждом носителе информации (гибком, жестком или лазерном диске) может храниться большое количество файлов. Порядок хранения файлов на диске определяется установленной файловой системой.

Файловая система - это система хранения файлов и организации каталогов. Для дисков с небольшим количеством файлов (до нескольких десятков) удобно применять одноуровневую файловую систему, когда каталог (оглавление диска) представляет собой линейную последовательность имен файлов. Для отыскания файла на диске достаточно указать лишь имя файла.

Если на диске хранятся сотни и тысячи файлов, то для удобства поиска файлы организуются в многоуровневую иерархическую файловую систему, которая имеет «древовидную» структуру (имеет вид перевернутого дерева).

Путь к файлу. Для того чтобы найти файл в иерархической файловой структуре необходимо указать путь к файлу. В путь к файлу входят записываемые через разделитель "\" логическое имя диска и последовательность имен, вложенных друг в друга каталогов, в последнем из которых находится данный нужный файл. объекты Рабочего стола Windows

В Windows на вершине иерархии папок находится папка Рабочий стол. (Следующий уровень представлен папками Мой компьютер, Корзина и Сетевое окружение (если компьютер подключен к локальной сети).

Структура папок выводится на панель папок, то есть в левую часть окна Проводник. Правая часть показывает содержимое папки, раскрытой в левой части.

С помощью значков легко ориентироваться в составе папок:

1. Если папка имеет значок «+», то внутри нее имеются вложенные папки
2. Значок «-» означает, что папка уже развернута до следующего уровня
3. Отсутствует «+» или «-» рядом с пиктограммой папки означает, что в данной папке нет вложенных папок

Просмотр свойств объектов

Чтобы просмотреть свойства объекта, достаточно щелкнуть по пиктограмме правой кнопкой мыши и выбрать из контекстного меню опцию Свойства. В открывшемся диалоговом окне можно узнать:

- Тип файла
- Имя охватывающей папки
- Размер файла в байтах
- Даты создания, изменения, открытия файла

Создание новых папок и ярлыков

Выберите в левой части окна Проводник диск, на котором создается новая папка. Последовательно раскрывая папки, доберитесь до папки, внутри которой будет находиться создаваемая папка. Откройте эту папку и вызовите из меню Файл команду Создать – Папка. Эту же команду можно вызвать из контекстного меню при щелчке правой кнопки мыши в правой части окна Проводник. Аналогично создается ярлык файла.

**Копирование файлов и папок**

Копировать файлы из папки в папки (или, к примеру, с жесткого диска на дискету) можно несколькими способами:

- Вызвать Контекстное меню файла и выбрать пункт Копировать. Теперь перейдите в папку или на диск, куда вы хотите поместить копию ваш файла, вновь вызовите Контекстное меню и выберите пункт Вставить;

- Эту же операцию можно проделать с помощью кнопок Копировать и Вставить на Панели инструментов окна Мой компьютер.

- Наконец, файл можно просто перетянуть мышкой в новую папку, держа при этом нажатой клавишу [Ctrl] клавиатуре (или перетянуть правой кнопкой мыши и выбрать в появившемся меню команду копировать).

- Выберите команду Правка – Копировать, откройте папку, в которую собираетесь скопировать объект, и затем выполните команду Вставить из меню Правка ( либо нажмите кнопку Вставить из буфера).

@ Буфер обмена – это область памяти компьютера, зарезервированная для временного хранения каких-либо данных. Буфер обмена позволяет выполнять операции копирования, вырезания и вставки.

Для переименования файла или папки можно выбрать пункт Переименовать Контекстного меню или просто нажать клавишу F2 на вашей клавиатуре. Обратите внимание — если ваш Проводник (Мой компьютер) умеет показывать не только имя файла, но и его расширение, будьте особо внимательны при переименовании. Если имя файла можно менять, то его расширение обязательно должно остаться в неизменным.

**Для удаления не нужных файлов можно воспользоваться одним из нескольких способов:**

- Вызвать Контекстное меню файла и выбрать пункт Удалить;
- Дать команду Файл?Удалить;
- Нажать кнопку на клавиатуре [Delete];
- Перетащить файл или папку в корзину.

#### 1) Фронтальный опрос

1. Как запустить программу Проводник?
2. Как выполнить просмотр свойств файлов и папок?
3. Как создать новую папку или ярлык?
4. Какие способы копирования Вы знаете?
5. Что нужно сделать для перемещения файла?
6. Как восстановить из Корзины документ?
7. На жестком диске "С:" в каталоге "Robot" хранится документ "resumeis.doc". Каков полный путь к файлу?
  - Robot esumeis.doc;
  - c:... esumeis.doc;
  - c: obot esumeis.doc.

ОС является как средой для организации работы пользователя, так и средой исполнения и взаимодействия различных программ.

#### **Основные функции:**

- Исполнение запросов программ (ввод и вывод данных, запуск и остановка других программ, выделение и освобождение дополнительной памяти и др.).
- Загрузка программ в оперативную память и их выполнение.
- Стандартизованный доступ к периферийным устройствам (устройства ввода-вывода).

- Управление оперативной памятью (распределение между процессами, организация виртуальной памяти).

- Управление доступом к данным на энергонезависимых носителях (таких как жёсткий диск, оптические диски и др.), организованным в той или иной файловой системе.

- Обеспечение пользовательского интерфейса.

- Сохранение информации об ошибках системы.

OS/360 использовалась на большинстве компьютеров IBM начиная с 1966, включая те компьютеры, которые помогли NASA отправить человека на Луну.

#### **Дополнительные функции:**

- Параллельное или псевдопараллельное выполнение задач (многозадачность).

- Эффективное распределение ресурсов вычислительной системы между процессами.

- Разграничение доступа различных процессов к ресурсам.

- Организация надёжных вычислений (невозможности одного вычислительного процесса намеренно или по ошибке повлиять на вычисления в другом процессе), основана на разграничении доступа к ресурсам.

- Взаимодействие между процессами: обмен данными, взаимная синхронизация.

- Защита самой системы, а также пользовательских данных и программ от действий пользователей (злонамеренных или по незнанию) или приложений.

- Многопользовательский режим работы и разграничение прав доступа

Компоненты операционной системы

- Загрузчик

- Ядро

- Командный процессор

- Драйверы устройств

- Встроенное программное обеспечение

**В функции операционной системы входит:**

- — осуществление диалога с пользователем;

- — ввод-вывод и управление данными;

- — планирование и организация процесса обработки программ;

- — распределение ресурсов (оперативной памяти и кэша, процессора, внешних устройств);

- — запуск программ на выполнение;

- — всевозможные вспомогательные операции обслуживания;

- — передача информации между различными внутренними устройствами;

- — программная поддержка работы периферийных устройств (дисплея, клавиатуры, дисковых накопителей, принтера и др.).

- — организация среды взаимодействия и обмена информацией между работающими программами.

## Тема-4: Основные элементы ОС Windows и рабочий стол. Возможности Windows. Стандартные программы ОС Windows.

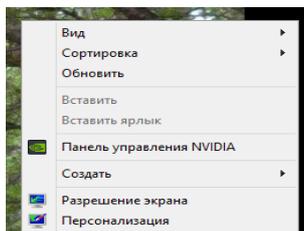
### План:

1. Основные элементы ОС Windows и рабочий стол.
  2. Работа с файлами, папками и дисками в среде Windows.
  3. Программа «Проводник».
  4. Специальные возможности Windows. Работа в стандартных программах ОС Windows.
  5. Работа в программе MSPaint
- Ключевые слова: Графический интерфейс. Рабочий стол. Окна Windows. Мой компьютер. Папка мои документы. Папка Корзина.

### Основные элементы ОС Windows и рабочий стол. Графический интерфейс.

Большая часть манипуляций с объектами в графической оболочке системы Windows производится с помощью мыши.

Поместив курсор мыши на какой-либо объект (значок, ярлык или папку) и нажав левую или правую кнопку, можно, не отпуская ее, перетащить объект в нужное место, например, другое окно, в другую папку, в корзину "для мусора" и т. д. Теперь, стоит отпустить кнопку и объект попадет в указанное ему место.



### Контекстно-зависимое меню

Видно, что объект можно открыть, т. е. запустить программу, если это программа или приложение, которое обрабатывает этот файл, или попросту *открыть документ*. Объект можно *печатать*, если у вас есть принтер, *отправить* по электронной почте, на дискету, его можно *удалить, переместить, скопировать, переименовать* и получить доступ к его *свойствам*.

Поместив курсор мыши на какой-либо объект (значок, ярлык или папку) и нажав дважды на левую кнопку мыши, мы заставим систему запустить программу или приложение, обрабатывающее документ.

Кроме мыши, при работе с системой можно пользоваться сочетаниями клавиш, иногда их называют *горячими*.

### Рабочий стол

После загрузки системы перед пользователем появляется так называемый рабочий стол, который имеет прямую аналогию с реальным письменным столом.

У рабочего стола Windows есть поверхность, покрытая *обоями* (аналог сукна), на столе стоят папки специальные папки, Корзина, меню **Пуск**, которое представляет собой своеобразный выдвижной ящик с множеством отделений, есть также *часы, индикатор раскладки клавиатуры* и многое другое в зависимости от настроек.

На рабочем столе сразу после установки системы появляется несколько значков. Как правило, это:

- **Мой компьютер.** С помощью этого значка пользователь получает полный доступ к содержимому компьютера — файлам, дискам, управлению ресурсами и всем настройкам.

- **Мои документы.** Этот значок открывает папку, в которой система по умолчанию сохраняет документы, созданные пользователем.

- **Корзина.** Аналог "мусорной" корзины. В корзину перемещаются файлы и документы при их удалении.

- **Сетевое окружение.** С помощью этого значка пользователь получает доступ к локальной компьютерной сети. Значок будет отсутствовать, если компьютер не подключен к сети.

Ярлыки создаются командой **Создать | Ярлык**, входящей в меню **Файл** различных окон папок, а также в контекстном меню, создаваемое нажатием правой кнопки мыши.

## Окна Windows

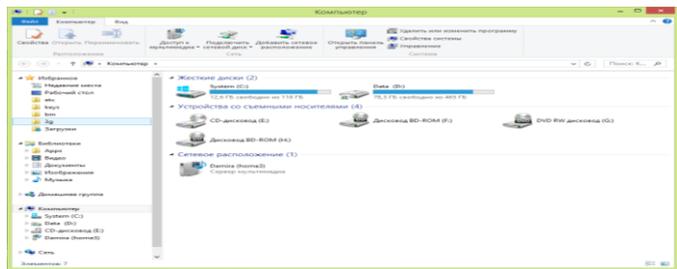
**Windows** с английского переводится как "Окна". Именно благодаря такой "оконной" концепции построения пользовательского интерфейса работа в системе становится быстрой и удобной.

## Мой компьютер

Основным значком на рабочем столе является значок **Мой компьютер** с изображением компьютера. Этот значок запускает приложение, которое является своеобразным менеджером всего компьютера — дает доступ к файловой системе вашего компьютера, позволяет запустить программу, настроить систему, получить доступ к сети, выбрать принтер. Окно **Мой компьютер**

Итак, в окне **Мой компьютер** содержатся следующие элементы

- **Ярлыки дисков.** Их может быть несколько — в виде значков отображены дисковод гибких дисков, жесткие диски (может быть один или несколько), привод компакт-дисков.



- **Панель управления.** Это служебная программа, запустив которую, вы получаете доступ к настройкам ресурсов компьютера.

- **Принтеры.** Папка содержит информацию об установленных принтерах. С ее помощью можно установить новый драйвер принтера

- **Удаленный доступ к сети.** позволяет настроить соединение с удаленной сетью или компьютером с помощью модема.

В зависимости от конфигурации системы папка **Мой компьютер** может содержать и другие служебные папки.

## Папка Мои документы

Один важный значок рабочего стола — **Мои документы**. По сути — это ярлык, который осуществляет переход к папке с документами, с которыми вы работаете.

**Папка Корзина.** Отдельного разговора заслуживает специальная папка **Корзина**, предназначенная для удаления документов.

## Работа с файлами, папками и дисками в среде Windows.

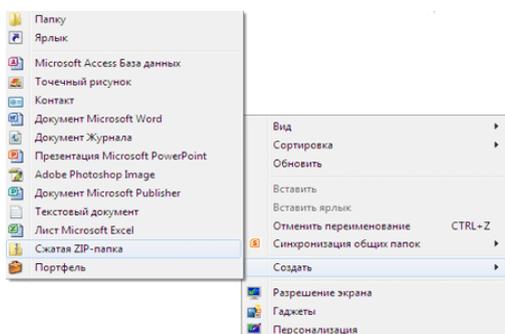


**Создание папок и файлы.** Окно папки **мой компьютер** можно производить создание файлов и папок, при этом

существует несколько способов создания. Чтобы создать папку или файл требуется, прежде всего, путем перемещения по папкам или дискам указать, где вы будете создавать новый объект.

Первый способ Первый способ заключается в использовании контекстного меню, подведите указатель к свободной области панели и нажмите правую кнопку мыши. Далее произведите щелчок на строке создать данного меню, появиться дополнительное меню с командами создания объектов, после вызова команды отображается значок объекта, и под ним строка ввода для названия, после того как название введено, нажмите кнопку Enter на клавиатуре и объект будет создан.

Второй способ Второй способ создания - это использование верхнего меню окна. Откройте Пункт меню файл, из него раскройте подменю создать, далее выберите команду Создание папки и произведите на ней щелчок левой кнопки мыши, появиться созданный элемент, для которого необходимо произвести ввод названия.



Для создания файла с помощью меню, необходимо выбрать одну из программы в списке создать, после чего создать файл с помощью выбранного приложения. После того, как файлы или папки созданы, с ними можно производить различные операции.

В папках также могут храниться другие папки. Папку внутри папки обычно называют **вложенной папкой**. Можно создать любое число вложенных папок, и каждая из них может содержать любое число файлов и дополнительных вложенных папок.

Создание папки равнозначно созданию каталога для хранения своих файлов. Папку можно создать непосредственно на Рабочем столе или на любом уровне внутри структуры каталогов.

Создание вложенной папки.

Создание вложенной папки необходимо выполнить следующие действия:

1. Открыть окно папки (или диска).
2. Выполнить одно из ниже приведенных действий.
  - ◆ В контекстном меню окно папки выбрать команду **Создать** →

**Папку;**

- ◆ В строке меню дать команду **Файл** → **Создать** → **Папку**;

- ◆ В списке типичных задач в группе задачи для файлов и папок щелкнуть ссылку создать новую папку появиться значок новой папки. Вместо имени по умолчанию Новая папка можно ввести любое название, соответствующее правилам Windows и нажать клавишу «Enter» (или щелкнуть мышью вне имени папки);

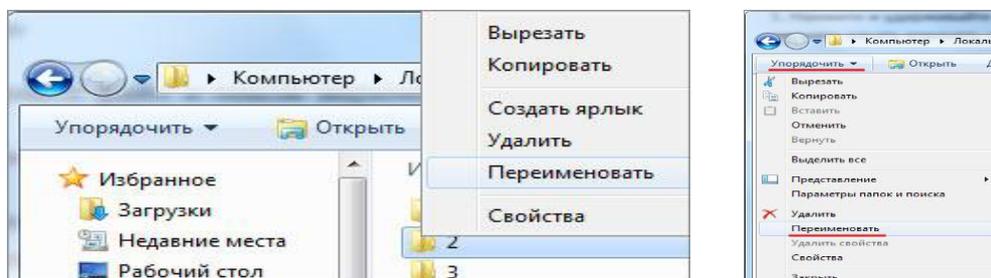
3. папка останется выделенной (подсвеченной). Повторное нажатие клавиши «Enter» откроет папку. Щелчок мышью вне папки – снимает выделение.

Переименование папки, файлов. (5 способов переименовать файл или папку)

Если Вы тот человек, который любит знать много способов для выполнения одной и той же задачи в Windows 7, то вот Вам 5 способов для переименования файлов, папок и ярлыков в проводнике Windows (в любой папке). Многие из них работают на рабочем столе и в окнах «Открыть» и «Сохранить».

1. Нажмите и удерживайте левую кнопку мыши на файле 1-2 секунды. Когда отпустите, Вы сможете начать изменять имя файла.

2. Выделите нужный файл для переименования и нажмите клавишу F2 на клавиатуре.

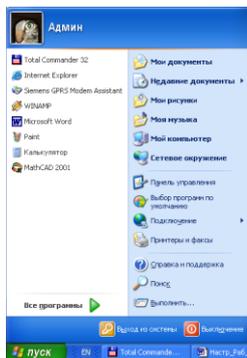


3. Щелкните правой кнопкой мыши по файлу и во всплывающем меню выберите «Переименовать».

4. Выделите файл, и левом верхнем углу проводника выберите «Упорядочить/ Переименовать».

5. Выделите файл для переименования, затем нажмите клавишу контекстного меню на клавиатуре, в появившемся меню выберите «Переименовать».

Панель инструментов и Панель задач Управление работой Windows осуществляется из трех источников: Панели задач, Панели управления и панели инструментов. Подробно рассмотрим каждый из них. *Панель задач* служит для запуска задач и для их переключения (рис.3). *Задача* — это любое приложение, выполняемое на компьютере в данный момент времени.



По своей сути меню **Пуск** — это папка и ей присущи практически все свойства стандартного объекта Windows (за исключением некоторых, например, удаление, перенос, копирование) Внутри же этой папки можно перемешать, копировать, удалять ярлыки и другие папки. В структуре меню **Пуск** по умолчанию есть несколько разделов, несколько отличающихся в зависимости от версии ОС Windows

- В разделе **Программы** находится список установленного на вашем ПК программного обеспечения, представленный в виде папок и значков. При установке новой программы значок для ее запуска автоматически добавляется в этот раздел.

- Раздел **Недавние Документы** содержит список недавно использованных вами документов

- Раздел **Панель управления** содержит все необходимые инструменты для настройки конфигурации вашего ПК

- Раздел **Поиск** поможет вам отыскать файл или папку на компьютере или в Internet.

- Пункт **Справка** и поддержка дублирует нажатие клавиши <F1> в операционной системе;

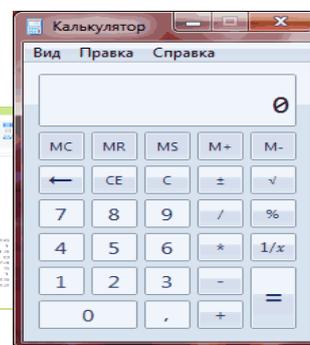
- пункт **Выполнить** предоставляет некое подобие командной строки на старой ОС и дает возможность специалисту быстрее выполнить некоторые действия,

- пункт **Выключение** производит выход из системы или ее перезагрузку.

### Программа Проводник

Для обзора и управления содержимым файловой системы была разработана программа Проводник. Запустить это приложение можно следующими способами. Самый простой — нажать комбинацию клавиш <Win>+<E>. Другой способ — нажать кнопку **Пуск**, а затем выбрать **Программы | Проводник**. Еще один вариант: щелкнуть правой кнопкой мыши на кнопке **Пуск** и выбрать соответственно пункт **Проводник** из контекстно-зависимого меню. Интерфейс Проводника оформлен в виде двух вертикально разделенных окон, в левом из которых отображается древовидная структура файловой системы, системы управления (служебные папки) и сеть, а в правом окне отражается содержимое выбранной в левом окне папки — файлы и вложенные папки.

Объектов, расположенных в разных местах списка, выделяют щелчком левой кнопки мыши при нажатой клавише <Ctrl>. Теперь выделенный объект или группу выделенных объектов



можно копировать, удалять и перемещать. Для приложений и документов, помимо перечисленных операций, можно создавать ярлыки. Если вы перетаскиваете выделенные объекты в Корзину, то объекты (файлы и папки) удаляются. Проводник позволяет изменять *свойства* тех или иных объектов, например, устанавливать для файлов и папок различные *атрибуты*, такие как "только чтение", "скрытый", "архивный", "системный". Эти атрибуты используются системой для того, чтобы определить, насколько свободно пользователь может работать с объектом.

### Стандартные программы ОС Windows.

Как и все программы, установленные на компьютере, значки стандартных программ находятся в меню **Пуск**. Заходим в меню **Пуск**, переходим во **Все программы** и ищем в списке папку **Стандартные**. Если нажать на этой папке левой кнопкой мыши, то она раскроется, и мы увидим список программ, которые по умолчанию установлены в Windows. Некоторые из этих программ могут быть нам полезны, поэтому давайте рассмотрим их. Мы поговорим о программах Калькулятор, Звукозапись, Экранная клавиатура, Экранная лупа.

#### Калькулятор.

С помощью программы Калькулятор можно производить не только арифметические вычисления, но и кое-что еще. Давайте об этом поговорим.

Запустим программу Калькулятор из папки Стандартная в меню Пуск.

Среди них есть специальная программа для рисования Paint.

Paint – это бесплатная программа для рисования на компьютере, которая изначально встроена в систему Windows.

Возможности Paint весьма ограничены, но, несмотря на это, в ней можно создавать чуть ли не шедевры. Все зависит от желания и таланта.

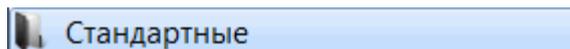
Как открыть программу Paint

Чтобы открыть на компьютере программу для рисования, нажмите на кнопку «Пуск» в левом нижнем углу экрана.

В открывшемся списке нажмите на пункт «Все программы» (Программы).



Появится довольно большой список. Выберите пункт «Стандартные».



И, наконец, откройте программу Paint.



Также на многих компьютерах вместо Paint установлена похожая программа под названием Paint.net. Выглядит она вот так:

#### **Тема-4: Современная обработка документов. Текстовый процессор Word.**

**План:**

1. Современные обработки документов.
2. Текстовый процессор Word.

Ключевые слова: шрифт, формат файла, гипертекст, символы, слово, абзац, фрагмент.

#### **Основные понятия**

**Текстовый редактор** - это простая программа для работы с текстами.

Текстовый редактор удобен для создания небольших сообщений и текстов. Текст состоит из букв, цифр, знаков препинания и специальных символов, которые можно ввести с помощью клавиатуры компьютера. Фотографии, чертежи, рисунки, схемы и таблицы текстовый редактор обрабатывать не может.

**Текстовый процессор** - это более мощная программа для работы с текстами. В текстовом процессоре можно писать письма, рассказы, стихи, доклады, книги и т.д.

Любой текст, созданный с помощью текстового редактора, а так же включённые в него нетекстовые материалы (графика, звуковые фрагменты) называют документом. Документ может быть статьёй, докладом, приглашением и т.д. При работе в сети части одного документа могут храниться на разных компьютерах, расположенных далеко друг от друга.

**Форматирование документа** - это оформление текста. Кроме текстовых символов форматированный текст содержит специальные невидимые коды, которые сообщают программе, как надо его отображать на экране и печатать на принтере.

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

**Шрифт** - это полный набор символов определенного начертания. Каждый шрифт имеет своё название, например Times New Roman, Arial, Comic Sans MS. Единицей измерения шрифта является пункт (1 пт. = 0,367 мм). Размеры шрифтов можно изменять в больших пределах. Кроме нормального (обычного) начертания символов обычно применяют полужирное, курсивное, полужирное курсивное.

Для проверки орфографии и синтаксиса используются специальные программные модули, которые обычно включаются в состав текстовых процессоров и издательских систем. Такие системы содержат словари и грамматические правила для нескольких языков, что позволяет исправлять ошибки в многоязычных документах.

Формат файла определяет способ хранения текста в файле. Простейший формат текстового файла (TXT) содержит только символы (числовые коды символов), другие же форматы (DOC, RTF) содержат дополнительные управляющие числовые коды, которые обеспечивают форматирование текста.

### **Классификация текстовых редакторов и процессоров**

Для обработки текстовой информации на компьютере используются приложения общего назначения - текстовые редакторы.

Многооконный редактор «Лексикон» - один из простейших текстовых редакторов отечественной разработки для несложных документов на русском или английском языке.

Простой текстовый редактор «Блокнот» позволяет редактировать текст и осуществлять простейшее форматирование шрифта.

Текстовый редактор Microsoft DOS (англ. Disk Operating System - дисковая операционная система) Editor используют для обработки информации состоящей из букв и цифр, без изображений.

Классическим примером текстового процессора является Microsoft Word из офисного пакета Microsoft Office, или Writer из пакета Star Office.

По сравнению с редакторами общего назначения специализированные редакторы позволяют легче готовить и редактировать соответствующие тексты: например, редакторы научных текстов удобнее для набора математических или химических формул (ChiWriter).

Для подготовки к публикации в Интернете web-страниц используют специализированные приложения (например, Microsoft Front Page или Macromedia Dreamweaver).

### **Характеристика текстовых редакторов и процессоров. Блокнот**

Чаще всего блокнот используют для проверки, и в некоторых случаях написания, HTML кода для web - страниц. Это связано с тем что, блокнот записывает в своё файл только чистый код символов без форматирования, что

является очень удобным для программистов. И иногда это просмотр.dll.inf.cfg и.bat файлов, которые небольшие по размерам, но важные по содержанию.

### **Среда текстового редактора**

**Microsoft Word** - основа любого офиса и, пожалуй, самая нужная и популярная программа во всем Microsoft Office. Эта программа установлена практически на каждом ПК и де-факто стала стандартом в обработке текста. На примере Word очень удобно изучать интерфейс всех остальных программ Microsoft Office.

Общими элементами среды любого текстового редактора являются:

- рабочее поле - пространство на экране, на котором отображается текст;
- курсор - место активного воздействия на рабочее поле;
- строка состояния - содержит информацию о текущем состоянии TP;
- строка меню - содержит команды переключения режимов работы TP и др. действий;
- полосы прокрутки - горизонтальная и вертикальные полосы справа и внизу окна документа, появляющиеся в случае не вмещающегося в пределах окна содержимого;
- панель инструментов и др.

### **Текстовый процессор**

Любой текст, созданный с помощью текстового редактора, а так же включённые в него нетекстовые материалы (графика, звуковые фрагменты) называют документом. Документ может быть статьёй, докладом, приглашением и т.д. При работе в сети части одного документа могут храниться на разных компьютерах, расположенных далеко друг от друга.

Гипертекст - это способ организации документа, позволяющий быстро находить нужную информацию. Он часто используется при построении систем оперативной подсказки и компьютерных версий больших справочников и энциклопедий. Основными объектами текстового документа являются: символ, слово, строка, абзац, страница, фрагмент.

Символы - это буквы, цифры, пробелы, знаки пунктуации, специальные символы. Символы можно форматировать (изменять их внешний вид). Среди основных свойств символов можно выделить следующие: шрифт, размер, начертание и цвет.

Слово - произвольная последовательность символов (букв, цифр), ограниченная с двух сторон служебными символами.

Абзац - произвольная последовательность символов, завершающаяся специальным символом конца абзаца. Допускаются пустые абзацы.

Фрагмент - это некоторое количество рядом стоящих символов, которые можно рассматривать как единое целое. Фрагментом может быть отдельное слово, строка, абзац, страница и даже весь вводимый текст.

Набор (ввод) текста, как правило, осуществляется с помощью клавиатуры. Роль бумаги при этом играет экран компьютера. Место для ввода очередного

символа текста указывается на экране с помощью мерцающего прямоугольника - курсора.

При наборе текста на компьютере человек не следит за концом строки: как только он достигнут, курсор автоматически переходит на начало следующей строки. Для того чтобы перейти к вводу нового абзаца, нажимается клавиша

<Enter>. С помощью стрелок вверх, вниз, влево, вправо курсор можно перемещать по всему экрану, подводить его к любому символу. Кроме того существует режим прокрутки, позволяющий быстро вывести на экран части текста, находящиеся за пределами документа. При перемещении курсора по экрану документ остаётся неподвижным, а при прокрутке неподвижным остаётся курсор.

### **Редактирование**

При редактировании текста мы просматриваем его, чтобы убедиться, что всё правильно, исправляем все обнаруженные ошибки и вносим необходимые исправления.

Редакторы текстов программ рассчитаны на редактирование программ на том или ином языке программирования. Часто они встроены в систему программирования на некотором языке программирования.

Редакторы текста, и рассчитаны на тексты программ, и выполняют следующие функции:

- диалоговый просмотр текста;
- редактирование строк программы;
- копирование и перенос блоков текста из одного места в другое;
- копирование одной программы или её части в указанное место другой программы;
- контекстный поиск и замену подстрок текста;
- автоматический поиск строки, содержащей ошибку;
- распечатку программы или её необходимой её части;

Редакторы документов - программы для обработки документов, ориентированные на работу с текстами, имеющие структуру документа, т.е. состоящими из разделов, страниц, абзацев, предложений, слов. Следовательно, редакторы для обработки документов обеспечивают функции, ориентированные на структуру документа, а именно:

- возможность использования различных шрифтов символов;
- задание произвольных межстрочных промежутков;
- автоматический перенос слов на следующую строку;
- автоматическую нумерацию страниц;
- обработку и нумерацию строк;
- печать верхних и нижних заголовков страниц (колонтитулов);
- выравнивание краев абзаца;
- набор текста в несколько столбцов;

- создание таблиц и построение диаграмм;
- проверку правописания и подбор символов.

Существует большое количество редакторов текста - от простых до сложных. Среди наиболее распространенных в мире редакторов выделяется Microsoft Word, Word Perfect, WordStar. Среди простых редакторов текста в Мире в определенный период времени был распространен ЛЕКСИКОН.

### **Форматирование документа**

Любой документ состоит из страниц, поэтому в начале работы над документом необходимо задать значения параметров страницы: формат, ориентацию, поля и др. Стандартным является формат страницы А4 (21x29,7 см), который соответствует размеру стандартного листа бумаги для принтера. Для объявлений и плакатов подходит формат А3, размер которого в два раза больше стандартного листа. Для писем можно выбрать формат А5, который в два раза меньше стандартного листа.

Существуют две возможные ориентации страницы - книжная и альбомная. Для обычных текстов чаще используется книжная ориентация, а для таблиц с большим количеством столбцов - альбомная.

Абзац является одним из основных объектов текстового документа. Абзац с литературной точки зрения - это часть текста, представляющая собой законченный по смыслу фрагмент произведения, окончание которого служит естественной паузой для перехода к новой мысли.

Списки применяются для размещения в документе различных перечней.

Существуют списки различных типов:

- Нумерованные списки, списка сопровождаются арабскими или римскими числами и буквами
- Маркированные списки, когда элементы списка отмечаются с помощью специальных символов-маркеров и др.
- Многоуровневый

По способу представления в компьютере различаются шрифты растровые и векторные. Для представления растровых шрифтов служат методы растровой графики, символы шрифта - это группы пикселей. Растровые шрифты допускают масштабирование только с определенными коэффициентами. В векторных шрифтах символы описываются математическими формулами и возможно произвольное их масштабирование. Среди векторных шрифтов наибольшее распространение получили шрифты типа TrueType.

Можно также установить дополнительные параметры форматирования символов: подчеркивание символов различными типами линий, изменение вида символов (верхний и нижний индекс, зачеркнутый), изменение расстояний между символами. Если планируется цветная печать документа, то можно задать различные цвета для различных групп символов.

### **Проверка орфографии и синтаксиса. Автотекст. Автозамена.**

Подбор синонимов / антонимов предназначен для стилистической обработки текста, что в конечном итоге служит лучшему восприятию информации, заключенной в документе. Для замены слова синонимом нужно

выделить это слово в тексте. Из поля - списка Значения выбрать значение, в котором используется это слово.

Автозамена используется для автоматической замены неправильно набираемых слов в тексте.

Для проверки орфографии и синтаксиса используются специальные программные модули, которые обычно включаются в состав текстовых процессоров и издательских систем. Такие системы содержат словари и грамматические правила для нескольких языков, что позволяет исправлять ошибки в многоязычных документах.

Word производит автоматическую проверку орфографии и синтаксиса во время ввода текста. Слово, отсутствующее в словаре или введенные с ошибкой, подчеркиваются на экране красной волнистой чертой. При неправильно расставленных знаках пунктуации или неправильно построенной фразе фрагменты текста подчеркиваются зеленой волнистой чертой.

**Гипертекст.** Гипертекст - текст, содержащий в себе связи с другими текстами, графической, видео- или звуковой информацией. Гипертекст позволяет структурировать документ путем выделения в нем слов-ссылок (гиперссылок). При активизации гиперссылки происходит переход на фрагмент текста, заданный в ссылке.

Гиперссылка состоит из двух частей: указатель ссылки и адресной части ссылки. Указатель ссылки - это объект (фрагмент текста или рисунок), который выделяется в документе (обычно синим цветом и подчеркиванием). Адресная часть гиперссылки представляет собой название закладки в документе, на который указывает ссылка. Закладка - это элемент документа, которому присвоено уникальное имя.

В качестве указателей ссылок и закладок могут использоваться не только фрагменты текста, но и графические изображения, поэтому такие структуры иногда называют гипермедиа.

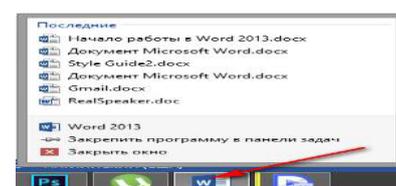
## Начало работы, дизайн и первые настройки в Word 2013-16

Человек, который только начинает разбирать Word 2013, постоянно забывает куда сохранял файлы. Поэтому самым первым шагом будет, это запоминать где вы что сохраняли. Это умеет Word, да ещё как. Если вы перетяните иконку (если у вас Windows 7 или Windows 8) на панель задач или откроете любой документ в Word. То при нажатии на иконку правой клавишей мыши вы увидите последние открытые документы.



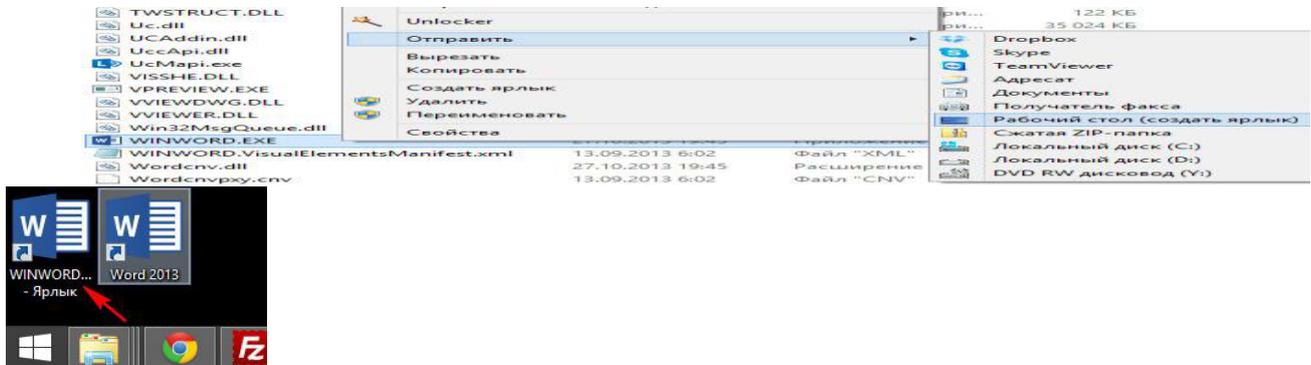
Кстати изменить параметры последних открытых документов, их количество и прочее, вы можете в Файл – Параметры – Дополнительно

Если нет на рабочем столе ярлыка Word и как его искать. Итак, второе,



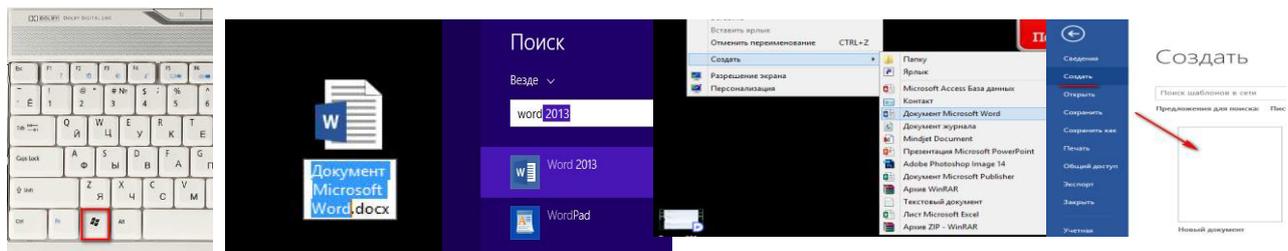
которое можно назвать первым, как вы ищите Word вообще. Иногда бывает, что пользователь не может найти Word на компьютере. Когда он у кого-то в друзьях или на новом рабочем месте. Я вам скажу, как быстро найти его.

Первое дело найти в папке C:\Program Files\Microsoft Office\Office 15 и ищем файл WINWORD.EXE. Запустили Word открылся. Мы также можем нажать правой клавишей мыши и отправить ярлык на рабочий стол, чтобы на рабочем столе появился ярлык Word



Также вы можете нажав на клавишу Windows, начав набирать на клавиатуре Word и также найти ярлык Word.

Ещё вы можете на рабочем столе кликнуть правой клавишей мыши – Создать – Документ Word и на рабочем столе появится новый документ Word

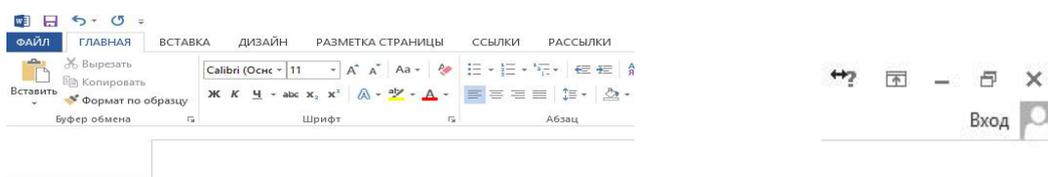


## Интерфейс Word 2013

При открытии Word 2013 мы видим перед собой Последние открытые документы, предложение создать новый документ и кучу шаблонов.

Дизайн выполнен в синих тонах, как и предшественники Word 2010, Word 2007 и Word 2003.

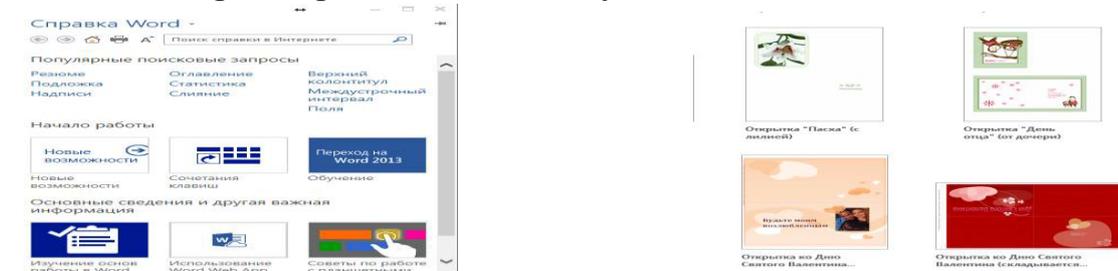
сверху находиться 3 клавиши (как и везде подобным) свернуть Word в трей, максимизировать или минимизировать и закрыть.



есть ещё справочный центр, его можно вызвать нажав на клавиатуре клавишу F1

Давайте попробуем какие-то функции и создадим новый документ. Перейдите в Файл – Создать – Новый документ

Слева, при открытии нового документа появляется Навигация.



Навигация Word 2013 это быстрый поиск по заголовкам, тексту который вы помните и постраничное разбиение. Если у вас нет навигации, кликните на левый нижний угол (Страница X из X) и у вас она появится



Тут вы можете найти нужную часть документа, открыть быстро любую его страницу

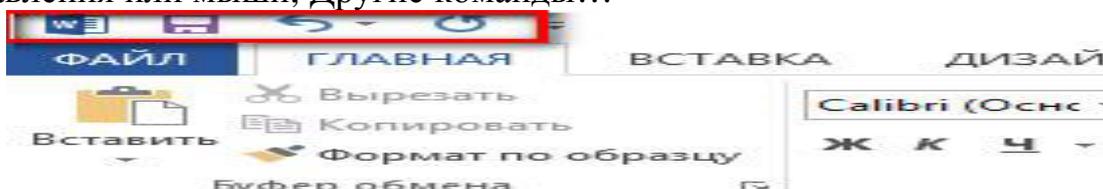
## Тема-5: Панель инструментов Microsoft Word. Работа с графическими объектами и с таблицами в Microsoft Word

### План:

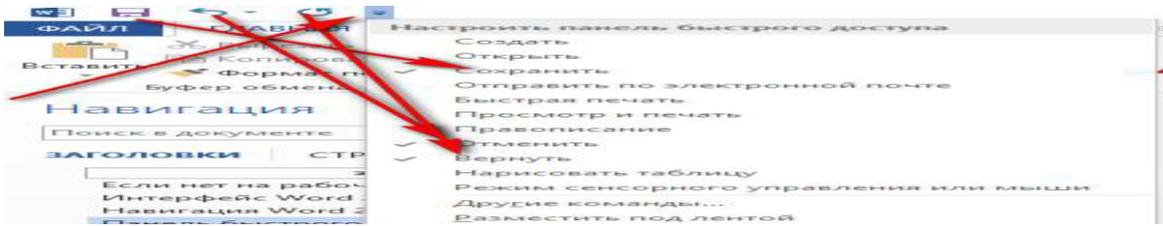
1. Панель быстрого доступа (самая верхняя). Вкладка "Главная"
2. Пример простого форматирования текста word. Выделение текста. Изменение шрифта word. Меню Вид
3. Вставка изображения. Вставка изображения с веб-страницы. Замена одного рисунка другим
4. Обтекание текстом. Замещающие рамки рисунков
5. Вставка таблицы. Рисование таблицы. Преобразование текста в таблицу.
6. Добавление в таблицу ячейки, строки и столбца. Удаление ячеек, строк или столбцов из таблицы. Объединение и разбиение ячеек в таблице

Панели инструментов Word – это основа рабочей среды данного текстового редактора, поэтому познакомиться с интерфейсом Word 2010 (2007) крайне важно и полезно любому пользователю, все зависимости от его совокупных знаний и навыков, поскольку достаточно контрастно отличается от внешнего вида Word прошлых версий программы.

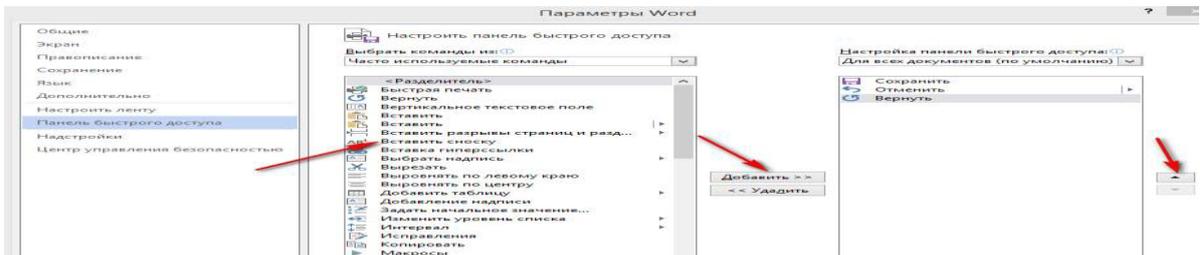
В самой верхней панели (с самого верху, левее от названия), вы можете выставить быстрые инструменты, такие как: Создать, Открыть, Сохранить, Отправить по электронной почте, Быстрая печать, Просмотр и печать, Правописание, Отменить, Вернуть, Нарисовать таблицу, Режим сенсорного управления или мыши, Другие команды...



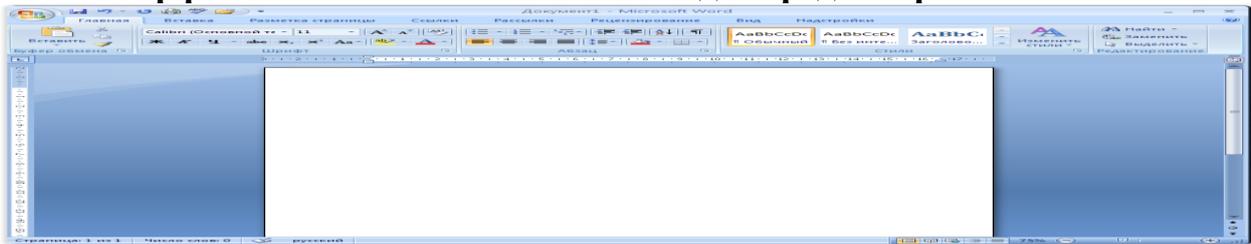
Для того, чтобы редактировать список команд сверху, вам надо нажать на чёрную стрелочку и галочками выбрать те команды, которые вы хотите видеть сверху



В другие команды, которые вы также можете вывести, вы можете добавить если перейдёте в другие команды. Там же вы можете регулировать ихние расположение, поменять местами как вам удобно.

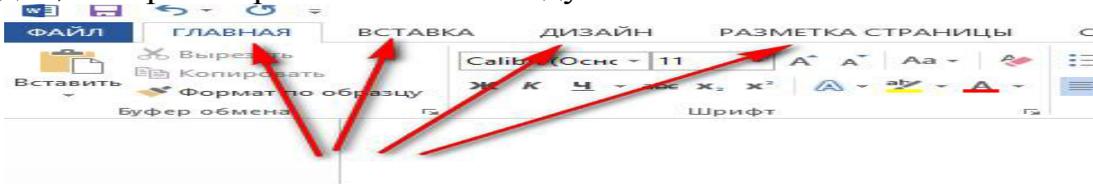


## Интерфейс или внешний вид редактора MS Word



### Главное меню Word

Главное меню состоит из меню и подменю. Главное меню выглядит в виде вкладок, которые переключаются между собой.



Если мы перейдём в меню Главная, то увидим перед собой подменю в удобном виде. Тоесть каждое подменю разделено для удобства, по стилям, шрифтам, то есть по подкатегориям что очень удобно

### Вкладка

### "Главная"



MS Word Вкладка главная

Основные инструменты этой вкладки - это:

Шрифт - иначе говоря внешний вид буквы. Каждый шрифт в программе Word имеет свое имя, сейчас выбран "Calibri"

Размер - тут понятно, размер букв, сейчас выбран 11-й размер

Ж К Ч - Жирный (буквы такие же как все, но толще), Курсив (буквы с наклоном), Ч - подчеркнутые буквы

- выравнивание текста по левому, по центру, по правому и по обеим сторонам

И так. На этом теория заканчивается и пора нам перейти к практике (к работе в MS Word), что бы хорошо понять и запомнить какой инструмент и для чего. Для этого нам необходимо выполнить следующее задание, которое сводится к тому, что надо напечатать текст и с помощью инструментов предать им требуемый внешний вид.

Вот это нам надо сделать:

#### **Заголовок текста.**

**Рассмотрим первым делом** выравнивание текста посередине, левому, правому краю и по обеим сторонам, т.е. по ширине. Для этого нам надо немного набрать текста, что бы на нем экспериментировать и применять те инструменты, которые мы рассматриваем.

*Рассмотрим первым делом* выравнивание текста посередине, левому, правому краю и по обеим сторонам, т.е. по ширине. Для этого нам надо немного набрать текста, что бы на нем экспериментировать и применять те инструменты, которые мы рассматриваем.

Рассмотрим первым делом выравнивание текста посередине, левому, правому краю и по обеим сторонам, т.е. по ширине. Для этого нам надо немного набрать текста, что бы на нем экспериментировать и применять те инструменты, которые мы рассматриваем.

**Рассмотрим первым делом** выравнивание текста посередине, левому, правому краю и по обеим сторонам, т.е. по ширине. Для этого нам надо немного набрать текста, что бы на нем экспериментировать и применять те инструменты, которые мы рассматриваем.

#### **Пример простого форматирования текста word**

Первым делом нам необходимо открыть программу MS Word и набрать там текст изображенный ниже на картинке (сначала изображение клавиатуры, а потом самого текста). Имейте ввиду, что сейчас Вы набираете просто текст, используя только буквы, запятую и точку. Клавишу Enter не нажимаем, при наборе текста в программе Word, текст автоматически переходит на новую строку, поэтому клавишу Enter нажимают тогда, когда надо сделать НОВЫЙ АБЗАЦ!

Чтобы напечатать большую букву, надо нажать клавишу Shift и не отпуская ее, саму букву (подробнее см. сочетание клавиш). Клавиши Shift обычно 2 штуки. Справа выделены 2 клавиши Shift и клавиша которой можно ввести точку - если просто на нее нажать и запятую - если нажать Shift и ее (так же как и ввод большой буквы)

Клавиатура Shift запятая



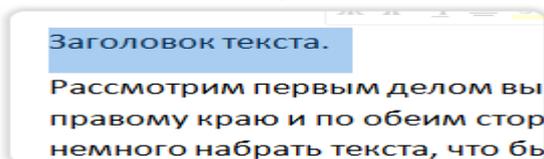
И так, набираем следующий текст.

Заголовок текста. Рассмотрим первым делом выравнивание текста посередине, левому, правому краю и по обеим сторонам, т.е. по ширине. Для этого нам надо немного набрать текста, что бы на нем экспериментировать и применять те инструменты, которые мы рассматриваем.

Теперь начинаем его форматировать - т.е. придавать ему внешний вид

Первым делом перенесем на новый абзац все, что идет после фразы "Заголовок текста." Для этого поставим каретку - это та самая мигающая палочка на месте которой появляется текст, прям перед словом "Рассмотрим", чтобы это сделать надо кликнуть левой кнопкой мыши наведя курсор туда, где должна быть каретка. После того как там появилась мигающая каретка нажмите клавишу Enter. эти мы все, что стояло после нее перевели на новый абзац.

Для дальнейших операций нам надо научиться выделять отрывки текста. Нам надо выделить фразу "Заголовок текста." Сначала поставьте каретку перед буквой "Заголовок" далее наведите туда же курсор мыши и нажмите левую кнопку мыши и не отпуская ее тяните курсор до точки этой фразы после чего отпустите. Должно получиться вот так:

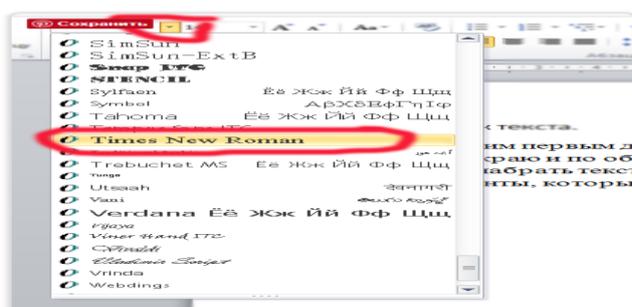


### Выделение текста

И пока текст выделен нажимаем выравнивание по середине для этого находим инструмент выравнивание и выбираем второй элемент из четырех. Этот элемент выравнивает текст по центру, об этом уже говорилось выше.

Если все сделано правильно то выделенный текст должен стать по середине страницы.

Далее выделим фразу "Рассмотрим первым делом"



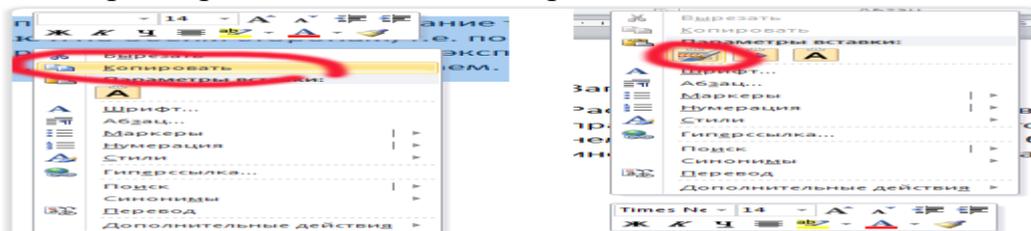
и текст станет жирным. Далее выделяем все кроме заголовка и меняем внешний вид букв выбирая в списке шрифтов "Times New Roman"

### Изменение шрифта Word

Ну вот и изменился внешний вид букв, далее так же выделяя

заголовок, рядом с выбором шрифта выбираем из списка размер шрифта 14-й.

Заметьте, что на моем рисунке, текст повторяется 4 раза, что бы его не набирать, его нужно скопировать и вставить. Для этого выделяем весь текст, по выделенному тексту ждем правой кнопкой мыши и выбираем пункт "копировать". После сего ставим нашу каретку в самый конец текста, делаем новый абзац (нажимая клавишу Enter) и еще раз кликаем правой кнопкой мыши и ждем "параметры вставки." И еще 2 раза вставляем.

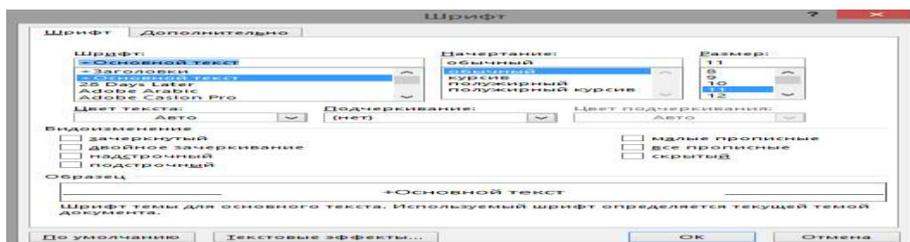


## Копировать и вставить в MS WORD

Прежде чем что-то с чем-то сделать, это что-то надо выделить, а потом нажать на инструмент, который с ним что-то сделает.

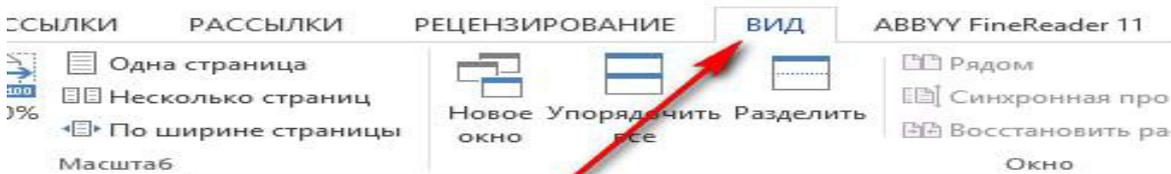


Если, мы нажмём Ctrl + D, к примеру, то вылезет настройка Шрифта, одним словом, удобно!

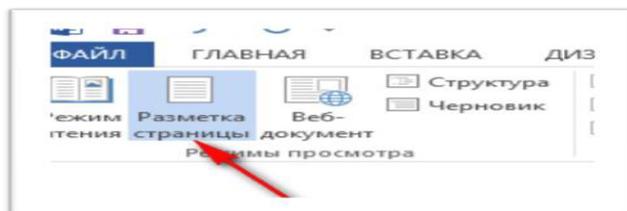


## Меню Вид

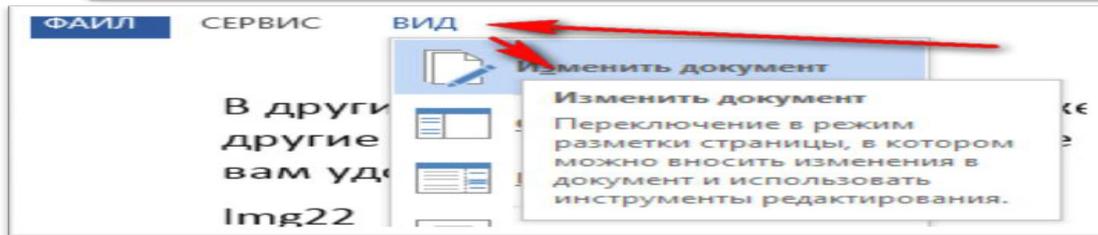
Меню вид – это меню просмотра документа Word. Есть режим чтения, режим веб-документа. По нажатии по пункту вебпросмотр, мы увидим, как будет выглядеть наш документ при печати или вебпросмотре



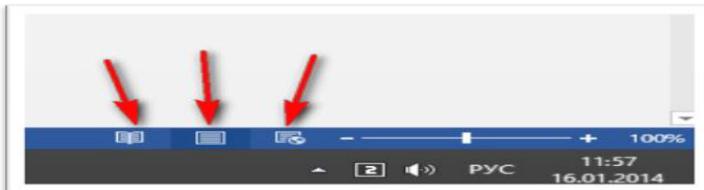
Чтобы перейти к обычному редактированию, надо перейти в “Разметку страницы”. Также очень удобен “режим чтения”, но чтобы выйти из него надо нажать



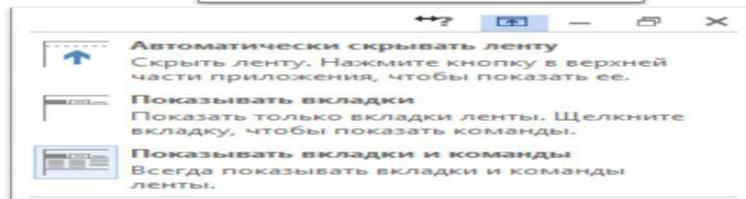
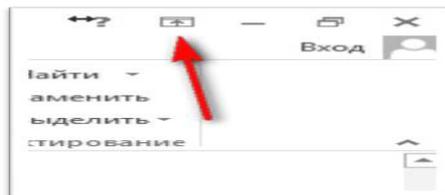
клавишу “Esc” или перейти в пункт Вид – Изменить документ.



Эти же 3 режима также есть снизу справа, рядом с процентным увеличением масштаба в виде полоски



Кстати ещё один маленький момент. Если у вас огромный экран, то сверху рядом с закрыть, есть панель в которой вы можете выбрать скрывать панель. Может кому-то да пригодится!

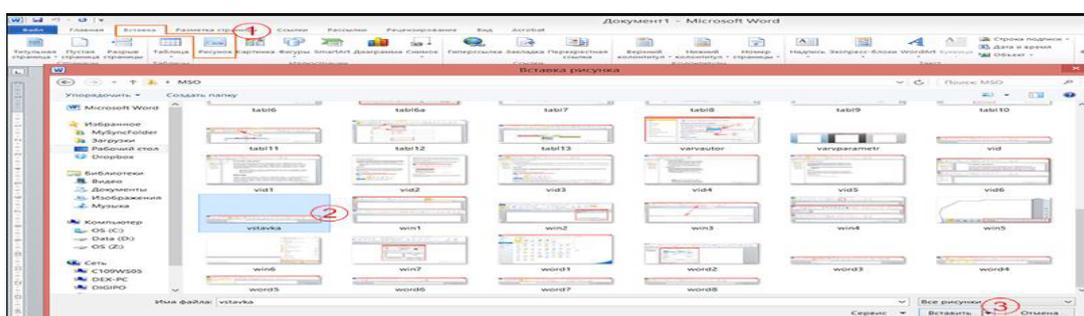


### Вставка изображения

Изображения, в частности, рисунки и фотографии, можно добавлять в документ Word 2010 из различных источников: лазерных носителей, флэш-дисков, жесткого диска или различных веб-ресурсов. Для того чтобы вставить в документ изображение, необходимо щелкнуть в том месте куда будет вставляться изображение с жесткого диска, после чего:

Перейдите на вкладку «Вставка» - «Рисунок»

В открывшемся окне «Вставка рисунка» найдите файл изображения, щелкните по нему; Нажмите кнопку «Вставить»



Обратите внимание на то, что вставка изображения может быть выполнена тремя различными способами.

«Вставить» - в этом случае изображение будет помещено в документ без связи с исходным файлом. Изображение помещается в документ без изменений, т.е. «как есть». В то же время, изменения, вносимые в исходный файл, никак не влияют (изменяют) изображение, вставленное в документ.

«Связать с файлом» - в данном случае в документ помещается лишь эскиз изображения, а сам рисунок не размещается в теле документа. Если вы продолжаете работать над изображением (постоянно его изменяете), то изменения отражаются на эскизе рисунка (вы их увидите при последующем открытии документа). Если вы перенесете документ на другой носитель, не поместив на него использованное изображение, то вы его просто потеряете. Вывод – связанные изображения должны либо находиться в теле документа, либо размещаться в папке документа и быть связаны (процедуру вставки изображения придется проделать повторно).

«Вставить и связать» - гибрид первого и второго метода. В данном случае изображение вставляется полностью, но остается связь с исходным файлом. То есть изменения, вносимые в оригинал, так же будут отражены во вставленном изображении. Его имеет смысл использовать, в случае если вам необходимо переносить документ на другие носители или размещать в Интернете одним файлом, но при создании документа вы хотите видеть изменения в изображениях. Учтите, что при переносе на носитель документа с эскизом изображения, связь с оригиналом теряется (если он не размещен на том же носителе и путь к нему не обновлен). То есть данный подход целесообразно использовать в процессе создания документа, изображения которого подвергаются изменениям.

При работе с окном «Вставка рисунка» вы можете быстро вставлять изображения способом «Вставить» (файл изображения добавляется как внедренный рисунок без связей с оригиналом) - для этого дважды щелкните по файлу изображения. Данный способ вставки используется в Word 2010 по умолчанию.

Вставка изображения с веб-страницы

Первый способ:

Откройте документ Word 2010, а затем веб-страницу с изображением;

Щелкните правой кнопкой мыши на рисунке (на веб-странице) и в контекстном меню выберите пункт «Копировать»;

Щелкните правой кнопкой мыши в нужном месте страницы документа Word 2010 и выберите пункт «Вставить».

Второй способ:

Откройте документ Word 2010 и веб-страницу с изображением.

Щелкните изображение и перетащите его в документ Word 2010.

Учтите, что данный способ подходит только в случаях, когда уменьшенная копия изображения содержит в себе ссылку на полноразмерное изображений. Ссылка может быть привязана к чему угодно, но чаще всего это копия

изображения или тематический текст. Некоторые сайты не поддерживают открытие изображений в отдельных окнах.

Если же вы попытаетесь перетащить неразвернутое изображение (в составе веб-страницы), то вы переместите только ссылку на него. Она привяжется к находящемуся в буфере обмена тексту или изображению, который автоматически будет вставлен с данной гиперссылкой. В отдельном случае, вы можете увидеть надпись «Ошибка! Недопустимый объект гиперссылки». Она сообщает о том, что изображение не может быть перемещено. Гиперссылка на него не может быть привязана к информации, помещенной в буфер обмена.

Для вставки изображений с веб-страницы вы также можете их предварительно сохранить на жестком диске компьютера, а затем вставить в документ описанными выше способами:

Откройте изображение на веб-странице, а затем щелкните на нем правой кнопкой мыши;

В контекстном меню выберите пункт «Сохранить рисунок как».

Замена одного рисунка другим

Если изображение в документе надо заменить на другое, при неизменных настройках размера и положения рисунка в документе, сделайте следующее:

Первый способ. Выделите в документе Word 2010 нужное изображение

Откройте вкладку «Формат» и в группе «Изменить» щелкните по кнопке «Изменить рисунок»

В открывшемся окне «Вставка изображения» найдите нужное изображение и щелкните «Вставить».

Второй способ. Щелкните по изображению правой кнопкой мыши в окне документа Word 2010

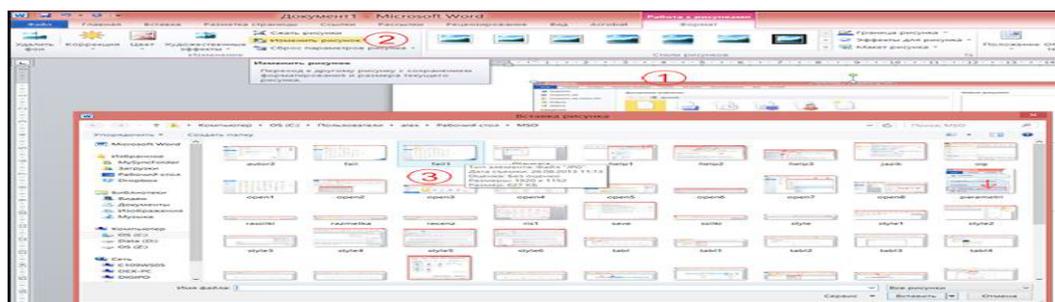
В контекстном меню выберите пункт «Изменить рисунок»

В окне «Вставка изображения» выберите нужное изображение.

Обтекание текстом

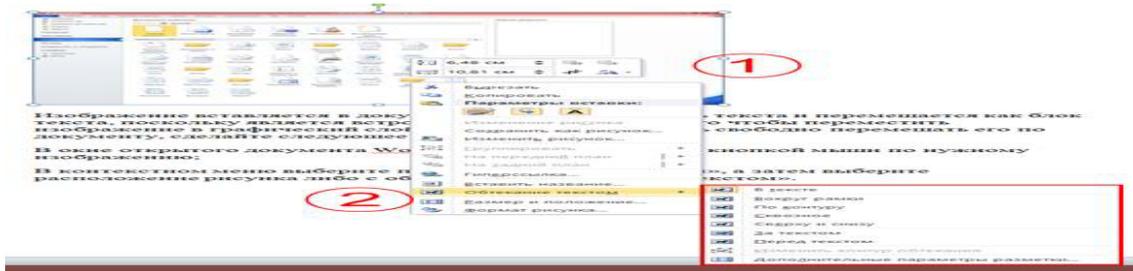
Изображение вставляется в документ Word 2010 как часть текста и перемещается как блок текста, поскольку является встроенным объектом. Для того чтобы переместить изображение в графический слой и получить возможность свободно перемещать его по документу, сделайте следующее:

В  
окне



открытого документа Word 2010 щелкните правой кнопкой мыши по нужному изображению;

В контекстном меню выберите пункт «Обтекание текстом», а затем выберите расположение рисунка либо с обтеканием, либо «Перед текстом».



### Замещающие рамки рисунков

По мере вставки в документ изображений увеличивается размер документа, и нагрузка на компьютер возрастает. На маломощных компьютерах документ Word 2010 с большим количеством изображений (особенно цветных и высокого качества) может замедлять просмотр документа. В данном случае можно ускорить работу, отображая вместо картинок только их рамки. Тем более это удобно, если вам необходимо просмотреть (редактировать) только текст.

Щелкните вкладку «Файл» - «Параметры»

Выберите закладку «Дополнительно»

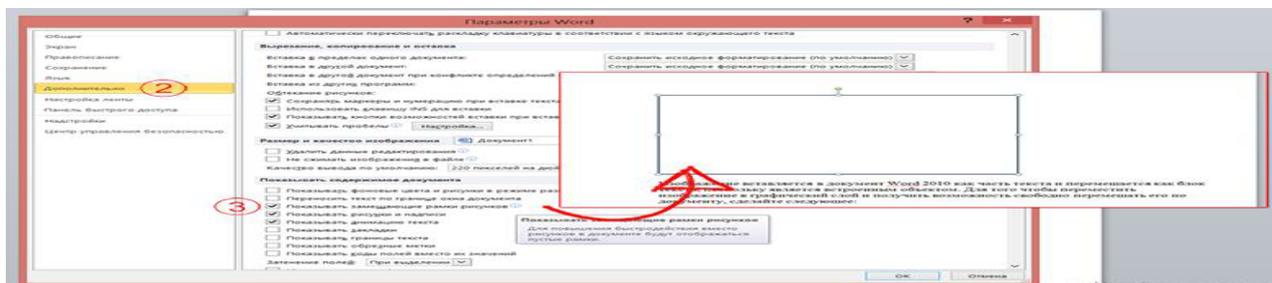
В группе «Показывать содержимое документа» поставьте флажок «Показывать замещающие рамки рисунков» и щелкните «ОК».

### Вставка таблицы

В Microsoft Office Word 2007 можно вставить таблицу, выбрав одну из предварительно отформатированных таблиц, заполненных примерными данными, или указав нужное количество строк и столбцов в таблице. Таблица может быть вставлена непосредственно в документ или вложена в другую таблицу, что позволяет создавать сложные таблицы.

### Использование шаблонов таблиц

Чтобы вставить в документ таблицу из коллекции предварительно отформатированных таблиц, можно воспользоваться шаблонами таблиц.



Шаблоны таблиц содержат примерные данные, которые помогают оценить вид таблицы после того, как в нее будут добавлены данные.

Щелкните место, куда требуется вставить новую таблицу.

На вкладке Вставка в группе Таблицы нажмите кнопку Таблица, выделите пункт Экспресс-таблицы и выберите нужный шаблон.

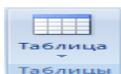


Замените данные шаблона своими данными.

Использование меню Таблица

Щелкните в документе место, куда требуется вставить новую таблицу.

На вкладке Вставка в группе Таблицы нажмите кнопку Таблица, а затем в области Вставить таблицу путем перетаскивания выберите нужное число строк и столбцов.

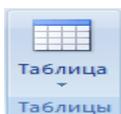


Использование команды Вставить таблицу

С помощью команды Вставить таблицу можно перед вставкой таблицы в документ указать ее размеры и формат.

Щелкните в документе место, куда требуется вставить новую таблицу.

На вкладке Вставка в группе Таблицы щелкните Таблица и затем выберите Вставить таблицу.



В области Размер таблицы введите количество столбцов и строк.

В области Авто подбор ширины столбцов выберите параметры подстройки размера таблицы.

Создание таблицы

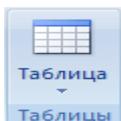
Таблицу можно создать, нарисовав ее или преобразовав текст в таблицу.

Рисование таблицы

Сложную таблицу, например таблицу с ячейками разной высоты или с меняющимся числом столбцов в строке, можно нарисовать.

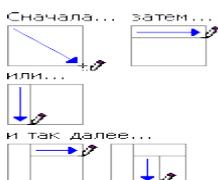
Щелкните в документе место, где требуется создать таблицу.

На вкладке Вставка в группе Таблицы щелкните Таблица и затем выберите Нарисовать таблицу.



Указатель мыши примет вид карандаша.

Чтобы определить внешние границы таблицы, нарисуйте прямоугольник. Затем внутри этого прямоугольника нарисуйте линии столбцов и строк.



Чтобы стереть линию или блок линий, на вкладке Конструктор контекстных инструментов Работа с таблицами в группе Нарисовать границы нажмите кнопку Ластик.

Щелкните линию, которую требуется стереть. Чтобы стереть таблицу целиком, см. раздел Удаление таблицы или ее содержимого.

Нарисовав таблицу, щелкните одну из ее ячеек, чтобы начать ввод текста или вставить изображение.

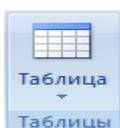
#### Преобразование текста в таблицу

Вставьте знаки разделителей, такие как запятая или знак табуляции, в местах, где текст должен быть разбит по столбцам. Используя знак абзаца, укажите, где должны начинаться новые строки.

Например, в списке с двумя словами в строке вставьте запятые или знаки табуляции после первых слов, чтобы создать таблицу с двумя столбцами.

Выделите текст, который требуется преобразовать.

На вкладке Вставка в группе Таблицы выберите пункт Таблица, а затем выберите команду Преобразовать в таблицу.



В диалоговом окне Преобразовать в таблицу в области Разделитель укажите знак разделителя, использованный в тексте.

Задайте другие необходимые параметры.

#### Вложение таблицы в другую таблицу

Таблицы, находящиеся внутри других таблиц, называются вложенными. Они часто используются при создании веб-страниц. Если представить веб-страницу в виде большой таблицы, содержащей другие таблицы, в ячейках которой помещены текст и изображения, можно создать разметку разных частей веб-страницы.

Чтобы вставить вложенную таблицу, щелкните ячейку, а затем вставьте таблицу любым из способов вставки или нарисуйте ее.

#### Рисование таблицы

##### ☐ Скрыть все

Можно нарисовать таблицу любой сложности — например, содержащую ячейки (Ячейка. Место для ввода данных, образованное пересечением строки и столбца на листе или в таблице.) разной высоты с разным количеством столбцов в строке.

Щелкните мышью в том месте, где необходимо нарисовать таблицу.

На вкладке Вставка в группе Таблицы щелкните Таблица.

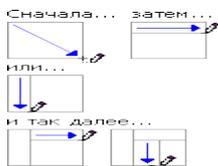


Выберите команду Нарисовать таблицу.

Указатель мыши примет вид карандаша.

Нарисуйте прямоугольник, обозначающий внешние границы таблицы.

В прямоугольнике нарисуйте линии строк и столбцов.



Чтобы удалить линию или блок линий, на вкладке Конструктор контекстных инструментов Работа с таблицами в группе Нарисовать границы нажмите кнопку Ластик. 

Щелкните линию, которую необходимо стереть. По окончании работы снова выберите команду Нарисовать таблицу, чтобы продолжить рисование таблицы.

Нарисовав таблицу, щелкните одну из ее ячеек, чтобы начать ввод текста или вставить изображение.

Добавление в таблицу ячейки, строки и столбца

Добавление ячейки

Щелкните ячейку справа или сверху от того места, куда требуется вставить новую ячейку.

В группе Средства таблицы щелкните вкладку Макет, а затем нажмите кнопку запуска диалогового окна Строки и столбцы.

Выберите один из следующих параметров.

Параметр	Действие
со сдвигом вправо	Вставка ячейки с перемещением остальных ячеек строки вправо. Примечание. Это действие не приводит к вставке нового столбца. В результате количество ячеек в строке может оказаться больше, чем в других строках.
со сдвигом вниз	Вставка ячейки с перемещением ячеек на одну строку вниз. В таблицу снизу добавляется новая строка.
вставить целую строку	Вставка строки над выделенной ячейкой
вставить целый столбец	Вставка столбца слева от выделенной ячейки

Добавление строки сверху или снизу

Щелкните ячейку, в строке, над или под которой требуется добавить новую строку.

На вкладке Положение контекстных инструментов Работа с таблицами выполните одно из следующих действий.

Чтобы добавить строку над выделенной ячейкой, в группе строки и столбцы нажмите кнопку Вставить выше.

Чтобы добавить строку под выделенной ячейкой, в группе строки и столбцы нажмите кнопку Вставить ниже.

Добавление столбца слева или справа

Щелкните ячейку в столбце, расположенном справа или слева от того места, куда требуется вставить столбец.

На вкладке Макет контекстных инструментов Работа с таблицами выполните одно из следующих действий.

Чтобы добавить столбец слева от выделенной ячейки, в группе строки и столбцы нажмите кнопку Вставить слева.

Чтобы добавить столбец справа от выделенной ячейки, в группе строки и столбцы нажмите кнопку Вставить справа.

Удаление ячеек, строк или столбцов из таблицы

Удаление ячейки. Выделите ячейку, которую необходимо удалить, щелкнув ее левую границу.



В группе Средства таблицы щелкните вкладку Макет.

В группе Строки и столбцы нажмите кнопку Удалить и выберите команду Удалить ячейки.

Выберите один из следующих параметров.

Параметр	Действие
Со сдвигом влево	Удаление ячейки с перемещением <i>остальных ячеек строки влево</i> . Примечание. Это действие не приводит к удалению нового столбца. В результате количество ячеек в строке может оказаться меньше, чем в других строках.
Со сдвигом вверх	Удаление ячейки с перемещением остальных строк в этом столбце на одну строку вверх. В столбец снизу добавляется пустая ячейка.
Удалить всю строку	Удаление всей строки, содержащей выделенную ячейку.
Удалить весь столбец	Удаление всего столбца, содержащего выделенную ячейку.

Удаление строки

Выделите строку, которую необходимо удалить, щелкнув ее левую границу.

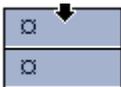


В группе Средства таблицы щелкните вкладку Макет.

В группе Строки и столбцы нажмите кнопку Удалить и выберите команду Удалить строки.

Удаление столбца

Выделите столбец, который необходимо удалить, щелкнув верхнюю линию сетки столбца или его верхнюю границу.



В группе Средства таблицы щелкните вкладку Макет.

В группе Строки и столбцы нажмите кнопку Удалить и выберите команду Удалить столбцы.

Объединение и разбиение ячеек в таблице

Объединение ячеек

Две и более ячеек, расположенных в одной строке или в одном столбце, можно объединить в одну. Например, можно объединить несколько ячеек в одной строке, чтобы создать заголовок таблицы, общий для нескольких столбцов.

Чтобы выделить ячейки, которые необходимо объединить, щелкните левую границу одной из них и перетащите указатель, удерживая нажатой кнопку мыши, по нужным ячейкам.

В группе Средства таблицы щелкните вкладку Макет, а затем в группе Слияние выберите команду Объединить ячейки.

Разбиение ячеек

Выделите одну или несколько ячеек, которые требуется разбить.

В группе Средства таблицы щелкните вкладку Макет, а затем в группе Слияние выберите команду Разбить ячейки...

Введите число столбцов и строк, на которое следует разбить выделенные ячейки.

Перемещение и копирование таблицы

Перемещение таблицы

В режиме разметки установите указатель на таблице и удерживайте его, пока не отобразится значок перемещения таблицы .

Удерживайте указатель на значке перемещения таблицы, пока он не примет вид крестообразной стрелки, а затем щелкните значок перемещения таблицы.

Перетащите таблицу в новое место.

Копирование таблицы с последующей вставкой в новом месте

Чтобы вставить таблицу в новом месте, ее можно предварительно скопировать или вырезать. При копировании таблицы исходная таблица остается на своем месте. Если таблица вырезана, исходная таблица удаляется.

В режиме разметки установите указатель на таблице и удерживайте его, пока не отобразится значок перемещения таблицы .

Щелкните значок перемещения таблицы, чтобы выбрать ее.

Выполните одно из следующих действий.

Чтобы скопировать таблицу, нажмите клавиши CTRL+C.

Чтобы вырезать таблицу, нажмите клавиши CTRL+X.

Поместите указатель мыши туда, где требуется вставить таблицу.

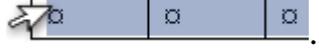
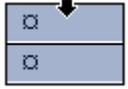
Нажмите клавиши CTRL+V, чтобы вставить таблицу в новое место.

Перемещение и копирование элементов таблицы

На вкладке Начальная страница в группе Абзац выберите команду Скрыть/Отобразить.



Выделите элемент, который требуется переместить или скопировать.

Выделяемый объект	Действие
Ячейка	Щелкните левый край ячейки. 
Строка	Щелкните слева от строки 
Столбец	Щелкните верхнюю линию сетки (т. е. верхнюю границу) столбца 

Выполните одно из следующих действий:

Чтобы переместить выделенный фрагмент, перетащите его на новое место.

Чтобы скопировать выделенный фрагмент, нажмите клавишу CTRL и, удерживая ее, перетащите выделенный фрагмент на новое место.

## Тема-6: Работа с гиперссылками в Microsoft Word

### План:

1. Гиперссылка в Word: создание и настройка
2. Гиперссылка на элемент текущего документа
3. Гиперссылка на другой документ
4. Гиперссылка на элемент другого документа
5. Гиперссылка на новый документ
6. Гиперссылка на адрес электронной почты
7. Удаление гиперссылок в Word
8. Гиперссылка в Word: создание и настройка

При написании дипломной работы или подготовке годового отчета возникает проблема ориентирования в документе из-за большого количества страниц. Уходит много времени, чтобы найти необходимый раздел и внести в него коррективы. Но есть простое решение – гиперссылка в Word:



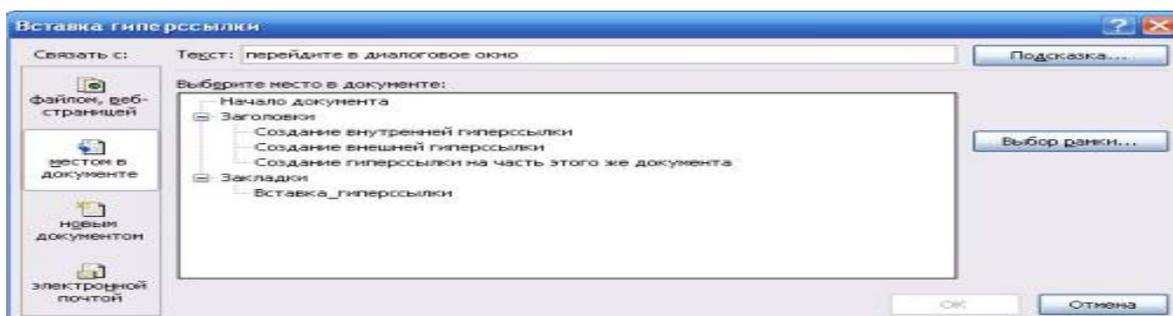
Гиперссылка – это объект, который, при нажатии на него левой кнопкой мыши, ссылается на другой элемент, расположенный в этом документе или на дисках компьютера, или является ссылкой на

ресурс в интернете. Объектом для гиперссылки может выступать единичный символ, слово, предложение, графическое изображение. Далее речь пойдет о том, как делать гиперссылки.

Гиперссылка на элемент текущего документа

Для начала нужно выбрать объект, который будет являться ссылкой. Если гиперссылка будет указывать на какой-то элемент текущего документа, то необходимо его выделить и создать в этом месте закладку. В Microsoft Word 2007 это сделать нетрудно. Нужно найти вкладку «Вставка», которая находится вверху на панели инструментов, открыть ее, найти группу «Ссылки», а в ней кликнуть по кнопке «Закладка». Откроется диалоговое окно, в котором необходимо заполнить поле «Имя», и для завершения кликнуть на кнопку «Добавить». Закладка готова.

Следующий этап в создании гиперссылки можно начать одним из двух способов: либо правой кнопкой мыши кликнуть по выделенному объекту, который был выбран для создания ссылки, либо снова открыть вкладку «Вставка» и в группе «Ссылки» выбрать пункт «Гиперссылка». Далее откроется окно, и в разделе «Связать с» необходимо выбрать элемент с надписью «местом в документе»:



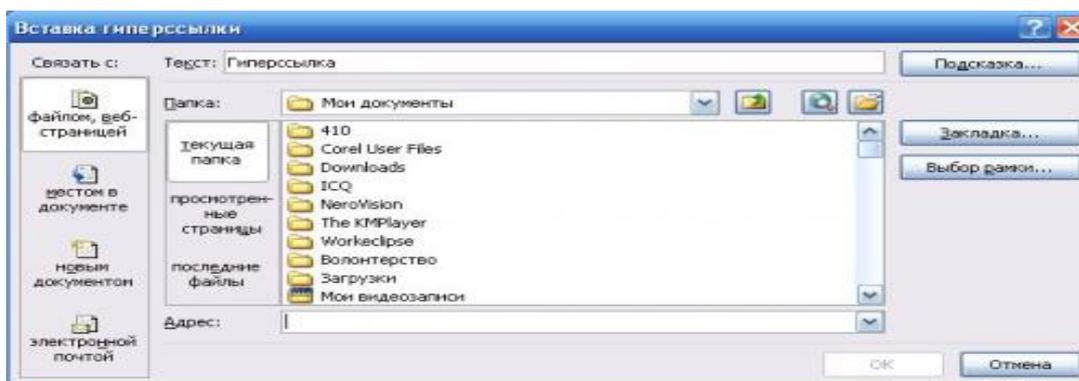
После этого будет представлен список, в котором отобразится созданная ранее закладка. Ее нужно выбрать, кликнув по ней, и нажать кнопку «ОК». Весь процесс займет менее минуты, потому как создать гиперссылку в Word очень легко.

Гиперссылка на другой документ

Если необходимо сделать ссылку на другой документ, то следует выполнить следующие манипуляции:

- В текущем документе выделить объект, который будет являться гиперссылкой;
- Открыть окно «Вставка гиперссылки», кликнув по выделенному объекту правой кнопкой мыши, и выбрать пункт «Гиперссылка». Или же найти этот пункт в «Ссылках» на вкладке «Вставка»;
- Далее в колонке под названием «Связать с» выбрать первый пункт с надписью «файлом, веб-страницей»;
- В поле «Папка» в стандартном окне проводника выбрать нужный элемент файловой системы;
- В случае если необходимо сослаться на интернет-ресурс, нужно скопировать из строки адреса URL веб-страницы или сайта и вставить его в поле «Адрес»;

- Для завершения кликнуть «Ок»:



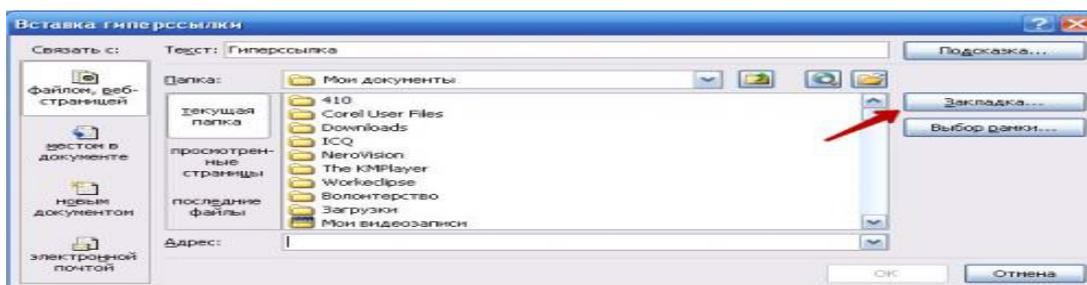
Гиперссылка на элемент другого документа

Чтобы сформировать гиперссылку на какой-то раздел или элемент другого документа, нужно, для начала, его открыть и создать в нем закладку в том месте, на которое должна вести гиперссылка.

После чего выполнить следующее:

- Открыть документ, где будет находиться гиперссылка, и выбрать для нее объект;
- Открыть окно «Вставка гиперссылки» (следуя по шагам, описанным в предыдущем пункте);
- Связать с необходимым документом, выбрав «Файлом, веб-страницей»;
- Кликнуть по кнопке «Закладка...». В открывшемся окне выбрать созданную ранее закладку.
- Нажать кнопку «Ок».

После выполнения этих действий, будет создана гиперссылка на определенный раздел другого документа:

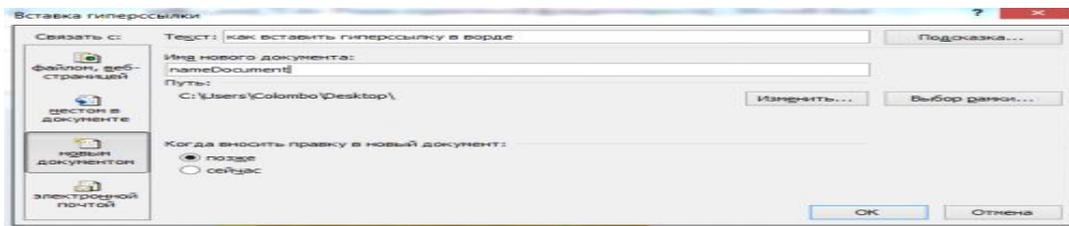


Гиперссылка на новый документ

Как сделать гиперссылку в Word на новый документ? Для этого необходимо выполнить все действия, описанные в предыдущем пункте до момента, когда открывается диалоговое окно «Вставка гиперссылки».

После чего необходимо сделать следующее:

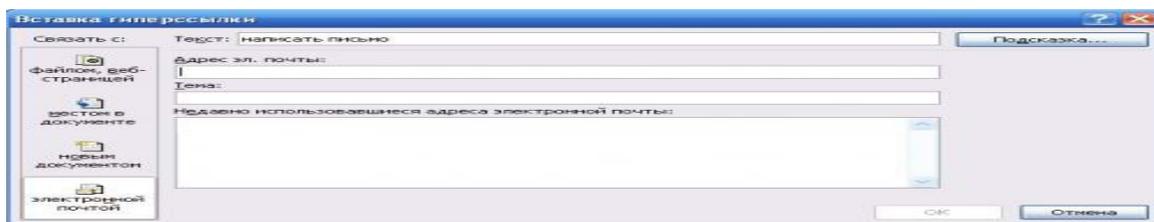
- В колонке «Связать с» выбрать пункт «Новым документом»;
- В поле «Имя нового документа» ввести имя, которое будет присвоено документу после активирования гиперссылки;
- Указать место на диске, где будет храниться новый документ. Для этого нужно кликнуть по кнопке «Изменить» и выбрать нужный раздел;
- В поле «Когда вносить правку в новый документ» выбрать «Позже»:



После нажатия по гиперссылке откроется новый документ Word, находящийся в указанном при создании гиперссылки месте.

Гиперссылка на адрес электронной почты

Так же, как и ранее, необходимо открыть диалоговое окно «*Вставка гиперссылки*». Меняется лишь пункт в разделе «*Связать с*», в котором необходимо выбрать элемент с надписью «*Электронной почтой*». Выполнив это, нужно в поле «*Адрес*» ввести адрес электронной почты и нажать «*Ок*»:



Теперь после клика по гиперссылке стандартный почтовый клиент будет создавать шаблон письма на указанный ранее адрес.

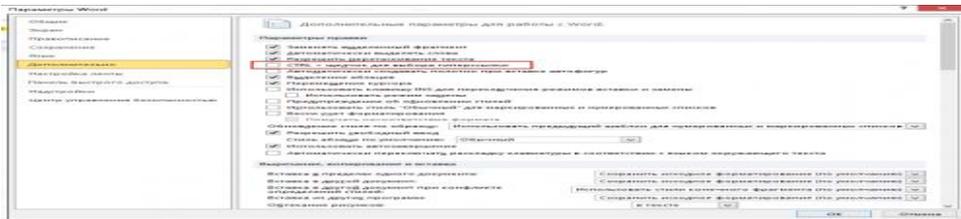
Есть еще один способ, как можно вставить в текст документа гиперссылку – необходимо просто набрать его с помощью клавиатуры и нажать Enter. Ссылка будет создана автоматически. Это очень удобно, если гиперссылка не должна быть скрытой.

То же самое можно проделать и с URL адресом веб-страницы – его нужно скопировать из адресной строки браузера и вставить в документ. Получается готовая ссылка, при нажатии на которую откроется нужная страница в интернете.

Для того чтобы гиперссылками было удобнее пользоваться, при их создании в окне «*Вставка гиперссылки*» можно добавлять подсказку, кликнув по кнопке «*Подсказка*». Откроется окно под названием «*Подсказка для гиперссылки*». В поле «*Текст подсказки*» необходимо ввести словосочетание, по которому можно легко понять, куда приведет ссылка. Для завершения нужно нажать на кнопку «*Ок*». Теперь при наведении курсора на ссылку будет отображаться созданная подсказка.

По умолчанию гиперссылка в Word активируется при помощи кнопки «*Ctrl*» и последующего клика по ссылке. Чтобы не нажимать клавишу «*Ctrl*», необходимо зайти в настройки программы, кликнув по вкладке «*Файл*» сверху панели инструментов, и выбрать пункт «*Параметры*».

После этого откроется окно «*Параметры Word*», в котором необходимо выбрать пункт «*Дополнительно*». Справа, в подразделе «*Параметры правки*» нужно снять галочку с пункта «*Ctrl + щелчок для выбора гиперссылки*». В результате этих манипуляций ссылка будет открываться после клика по ней:



## Удаление гиперссылок в Word

Как удалить гиперссылку в Word? Чтобы удалить гиперссылку, достаточно просто навести на нее курсор, нажать правую кнопку мыши и выбрать из меню пункт «Удалить гиперссылку». При этом текст или изображение, которое было основой для ссылки, останется на своем месте без каких-либо изменений:



Есть еще один способ, как убрать гиперссылку – установить курсор у крайнего справа символа ссылки, нажать клавишу «Backspace» один раз. После этого исчезнет выделение цветом и подчеркивание. Это означает, что ссылка удалена. Чтобы полностью удалить всю ссылку вместе с объектом (символы, изображение, текст) нужно полностью выделить объект и нажать клавишу «Delete». В этой статье было описано, как вставить гиперссылку в Word, которая ссылалась бы на раздел текущего документа или создавала новый на адрес электронной почты или интернет ресурс. Также были рассмотрены способы удаления гиперссылок. Процесс создания и удаления гиперссылок в версиях Microsoft Word 2007, 2010, 2013 не отличается, поэтому данная статья будет актуальна для всех перечисленных версий продукта.

## Тема-7: Создание презентаций в Power Point. Меню Дизайн

### План:

1. Меню «Главная» в программе Power Point
2. Вставка текста на слайд
3. Вставка таблиц графики, диаграмм и гиперссылка
4. Оформление слайда

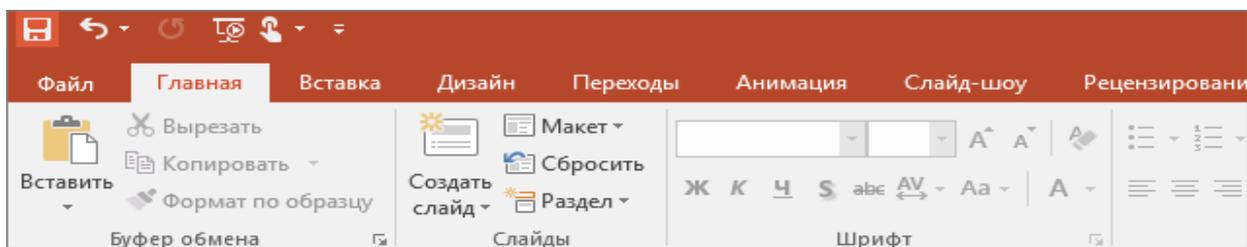
### Меню «Главная» в программе PowerPoint

В программе PowerPoint имеются все функции нужные для создание профессиональных презентации.

#### Запуск PowerPoint.

Для запуска PowerPoint нажмите кнопку *Пуск* и выберите в главном меню команду **Программы→MicrosoftOffice→MicrosoftPowerPoint**.

Вкладка **Главная** содержит команды **Вырезать** и **Вставить** группы параметров **Шрифт** и **Абзац** а также все нужно для добавления и упорядочение слайдов.



Основные компоненты презентации, создаваемой в программе PowerPoint:

- **Слайды.** Слайды называют чёрно-белые и прозрачные плёнки, предназначенные для презентации на видеоаппаратуре, а также изображения, представляемые на экране компьютера или с помощью проектора, управляемого компьютером.

- **Заметки.** К каждому слайду прилагается страница заметок, на которой находится уменьшенная копия слайда и отведено место для примечаний

- **Раздаточные материалы.** Это краткое содержание презентации, напечатанное на одной странице, которое помогает следить за ходом презентации.

### О слайдах

Слайды содержат, которые представляет собой области слайда, могут содержать много различных значений, включая текст, картинки, графику и т.д.

Также в них могут быть представляющие определенные команды, такие как Картина Добавление диаграммы, вставка видео.

В PowerPoint достаточно навести курсор мыши на эти иконки, чтобы узнать какой тип содержимого можно добавить.

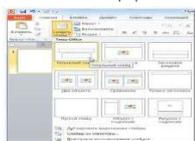
### Макеты слайдов.

Заполнители расположены на различных макетах, которые могут быть применены к существующим слайдам или выбраны при создании нового слайда. Это набор заполнителей для текста, картинок, таблиц или мультимедиа.



### Создание презентаций.

Каждая презентация PowerPoint состоит из набора слайдов. Создание презентации заключается в первую очередь в размещении текста и объектов на слайдах.



При создании каждой презентации в PowerPoint выполняется одно и та же последовательность действий:

1. Проектирование презентации с помощью мастера или на основе шаблона
2. Корректировка схема презентации и форматирование слайдов
3. Добавление в слайд звуковых эффектов, музыкальных файлов и видеозаписей

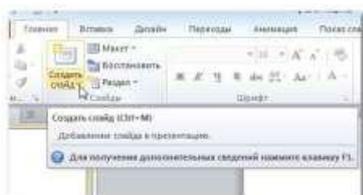
4. Настойка времени показа слайдов и создание анимационных эффектов
5. Подготовка раздаточных материалов и заметок докладчика
6. Демонстрация презентации.

Чтобы вставить новый слайд:

На вкладке Главная команды Создать слайд, чтобы открыть меню.

Выберете слайд который хотите вставить

Чтобы сразу добавить слайд с таким же образом, как выделенной сейчас слайд, просто кликните по верхней части команды Создать слайд

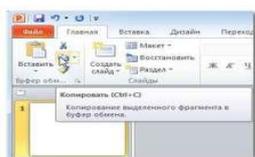


Чтобы скопировать и вставить слайд:

На вкладке Слайды в левой части выберите нужный слайд.

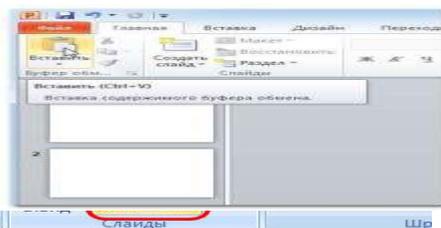
По команде, **Копировать** на вкладке, **Главная**. Вы также можете кликнуть по выбранному слайду правой мыши и выбрать, **Копировать**.

Кликните по команде, **Вставить** на вкладке, **Главная** или кликните правой кнопкой мыши и выберите, **Вставить**. Появится скопированный слайд.



Если мы хотим удалить слайд отмечаем левой кнопкой мыши слайд подлежащий удалению, затем,

можно в меню кнопке **Удалить**.



**Главная** щелкаем по

А можно еще нажмите клавишу **Delete** или **Backspace**

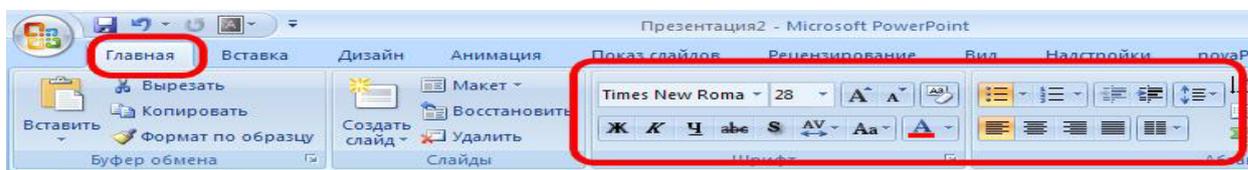
## Форматирование текста

Иногда требуется изменить стиль текста на каком либо заполнителе. Для изменение стиля текста в каком либо из заполнителей, нужно выделить текст. При выделении текста появляется всплывающее меню форматирования.



Например, на вкладке, **Главная** в группе ” **Шрифт** “ отображены основные кнопки форматирования: тип, размер шрифта, выделение полужирный или курсивом цвет и другие.

А также можно выделить текст зайти в меню Главная .Там также находится инструменты форматирования текста.



. Вкладке **Главная** упорядочивать объектов на слайде путем изменение их порядке положения и угла поворота. Кроме того, можно сгруппировать несколько объектов таким образом, чтобы в дальнейшем их можно было обрабатывать как один объект.

Используйте команду **Выделить объект** для выделения объектов расположенных позади текста.

Заменить { Ctrl+N }

Замена текста в документе.

В контекстном меню или в меню **Формат** выбрать команду **Разметка слайда** - в **Области задач** появится панель **Разметка слайдов**, содержащая стандартные макеты размещения объектов. Чтобы применить один из макетов к выбранному слайду, достаточно щелкнуть на нем.

Вставка текста на слайд

Текст на слайд PowerPoint вводится в специальную область, называемую рамкой.

Введенный текст может быть отформатирован с помощью соответствующих команд меню **Формат** или кнопок панели инструментов **Форматирование**.

В надпись можно вводить и текст, скопированный из документа Word. При копировании и вставке текста можно использовать параметры кнопки **Параметры вставки**, которая появляется под вставленным текстом. Эти параметры позволяют сохранить исходное форматирование или отказаться от него.

Если кнопка **Параметры вставки** не отображается, эту функцию можно включить, выбрав в меню **Сервис** команду **Параметры**, а затем на вкладке **Правка** установить соответствующий флажок.

Вставка таблиц, графики, диаграмм и гиперссылки

В PowerPoint создать таблицу можно командой **Таблица** меню **Вставка**. Вся дальнейшая работа с таблицей производится с помощью кнопок появившейся панели инструментов **Таблицы и границы** и очень похожа на аналогичную работу в приложении Word.

Вставка графики

Для вставки изображения используется команда **Рисунок** меню **Вставка** или кнопка на панели инструментов **Рисование**. Используя параметры команды **Формат рисунка**, которая вызывается из меню **Формат** или контекстного меню, можно изменить размер вставленного рисунка, при необходимости обрезать его и/или сжать.

Чтобы создать фотоальбом, выберите в меню **Вставка** команду **Рисунок**, а затем **Создать фотоальбом**.

Вставка диаграмм

Для создания и редактирования диаграмм в программе PowerPoint используется встроенный модуль Microsoft Graph. Он управляет окном, в котором во время работы с диаграммой открываются его собственные меню и панели инструментов. Для вставки диаграммы щелкните на кнопке **Добавление диаграммы** панели инструментов **Стандартная** или в меню **Вставка** выберите команду **Диаграмма**. Обратите внимание, что изменилось меню и панели инструментов, теперь они напоминают Excel.

Добавление на слайд объектов, созданных в других приложениях

1-й способ. Выбрать в меню Вставка команду Объект.

2-й способ. Скопировать в исходном приложении необходимый фрагмент, перейти в PowerPoint и выбрать команду Специальная вставка в меню Правка.

Вставка гиперссылок

Для создания гиперссылки выделите объект и выберите команду Гиперссылка в меню Вставка или воспользуйтесь кнопкой панели инструментов Стандартная.

Оформление слайда

Для создания образца слайда выполните команду меню Вид Образец Образец слайдов. Появится новая панель инструментов Образец, которая содержит кнопки, позволяющие вставлять, удалять, переименовывать и сохранять созданный образец.

Добавление эффектов анимации к объекту на слайде.

Чтобы назначить анимационный эффект некоторому объекту на слайде, перейдите в режим Обычный и выполните следующие действия.

1. Откройте тот слайд презентации PowerPoint, на который планируется добавить анимационный эффект. Выделите объект, к которому будет применена анимация.

2. Выберите команду Настройка анимации в меню Показ слайдов или в контекстном меню.

В любом случае в области задач откроется панель Настройка анимации.

1. Щелкните на кнопке Добавить эффект. (Если кнопка Добавить эффект неактивна, значит, на слайде нет выделенного объекта). Выберите в раскрывшемся меню один из четырех типов анимационного эффекта, который необходимо применить к выделенному объекту, а затем в подменю и сам эффект.
2. Если требуемый эффект анимации отсутствует в открытом подменю, выберите команду Другие эффекты. Выбранный анимационный эффект будет применен к объекту и название этого эффекта появится в списке Изменение на панели Настройка анимации области задач. Порядковый номер эффекта также появится на слайде возле объекта, к которому данный анимационный эффект был применен, а его название и номер помещены в список эффектов на панели Настройка анимации области задач. Этот номер отражается только в обычном режиме при открытой на экране панели Настройка анимации области задач. Используя кнопки Порядок, можно изменить последовательность эффектов. Чтобы просмотреть добавленные эффекты анимации, щелкните на кнопке Просмотр панели Настройка анимации. Анимационные эффекты, примененные к объектам, можно дополнительно настроить, отредактировав их параметры. Чтобы настроить дополнительные параметры эффектов анимации, выполните следующие действия.

1. Щелкните правой кнопкой мыши на кнопке со стрелкой, расположенной возле названия эффекта в списке на панели Настройка анимации области задач.

2. В раскрывшемся меню выберите команду Параметры эффектов. Откроется диалоговое окно, название и содержание которого зависит от выбранного эффекта и от объекта, к которому он применяется.

3. Определите нужные значения и щелкните на кнопке ОК. Настройки диалогового окна позволяют добавить к эффекту звук, изменить цвет объекта, задать необходимые временные характеристики и другие индивидуальные параметры.

Анимация диаграмм, таблиц и векторных изображений  
Возможна настройка анимации не только для диаграмм построенных в PowerPoint, но и импортированных из Excel в виде диаграммы. Делается это следующим образом:

- выделив диаграмму, выберите из контекстного меню или из меню Показ слайдов команду Настройка анимации;

- в области задач с помощью кнопки Добавить эффект откройте список эффектов анимации и выберите понравившийся эффект. Эффект появится на панели Настройка анимации и будет распространяться на всю диаграмму целиком;

Если Вы хотите отдельно анимировать элементы диаграммы, надо выполнить еще несколько действий.

1. Щелкните правой кнопкой мыши на кнопке со стрелкой, расположенной возле названия эффекта в списке на панели Настройка анимации области задач.

2. В раскрывшемся меню выберите команду Параметры эффектов. Откроется диалоговое окно, содержащее вкладку Анимация диаграммы.

3. Определите нужные значения и щелкните на кнопке ОК

Создание произвольного показа

1. В меню Показ слайдов выберите команду Произвольный показ и нажмите кнопку Создать.

2. В области Слайды презентации выберите слайды, которые следует включить в произвольный показ и нажмите кнопку Добавить. Чтобы выделить несколько слайдов, нажмите клавишу CTRL и, удерживая ее, по очереди выберите требуемые слайды. С помощью стрелок можно изменить порядок показа.

3. Введите имя произвольного показа и нажмите кнопку ОК.

Можно оформить каждый раздел презентации в виде произвольного показа, затем автоматически создать оглавление и связать его пункты с произвольными показами.

1. Откройте презентацию, для которой требуется создать слайд с оглавлением.

2. Создайте произвольный показ для каждого раздела презентации.

3. В режиме Сортировщика слайдов выделите первый слайд в каждом новом произвольном показе. (Чтобы выделить несколько слайдов, последовательно выберите их, удерживая нажатой клавишу CTRL.)

4. На панели инструментов Сортировщик слайдов нажмите кнопку Итоговый слайд.

5. Дважды щелкните новый слайд, который появится перед остальными слайдами. Он будет содержать заголовки первых слайдов каждого раздела.

6. Чтобы в каждом маркированном элементе списка создать гиперссылку на соответствующий произвольный показ, выделите маркированный элемент, а затем в меню Показ слайдов выберите команду Настройка действия.

7. Установите флажок Перейти по гиперссылке, а затем в списке выберите

пункт Произвольный показ. Выберите произвольный показ, к которому требуется перейти по гиперссылке, а затем установите флажок Показать и вернуться.

8. Повторите пункты 6-7 для каждого маркированного элемента слайда с оглавлением.

Наиболее часто задаваемый вопрос как сделать, чтобы музыка проигрывалась непрерывно.

1. Щелкните правой кнопкой мыши на пиктограмме звукового файла и выберите команду Настройка анимации.

2. Щелкните на стрелке "вниз" рядом с пиктограммой звука и выберите из появившегося меню команду Параметры эффектов. Откроется диалоговое окно Воспроизвести звук.

3. Щелкните на переключателе После (в разделе Закончить) и установите в нем количество слайдов, на протяжении которых должна играть музыка.

При подсчете слайдов начните со слайда, на котором включается звук. Например, представьте, что в презентации десять слайдов. Вы помещаете звук на третий слайд и хотите, чтобы он воспроизводился на слайдах 3, 4, 5 и 6 и выключался на слайде 7. В этом случае в поле После следует установить цифру.

4. Щелкните на кнопке ОК.

Вставить слайды из Microsoft PowerPoint в Prezi достаточно просто.

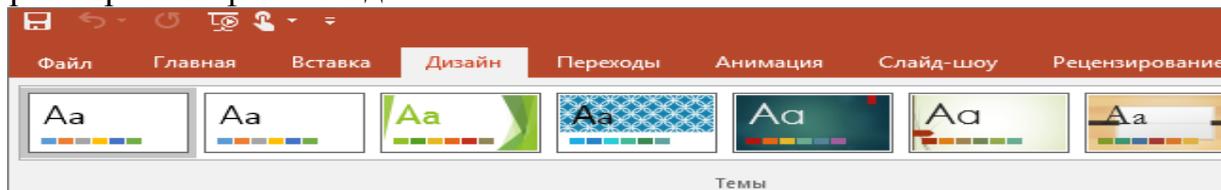
Особенностью вставки является то, что из слайдов PowerPoint импортируется текст, который можно быстро изменить в Prezi. Иллюстрации из PowerPoint могут быть изменены непосредственно в веб-редакторе Prezi.

Для вставки слайдов из PowerPoint откройте Prezi-презентацию в режиме редактирования и нажмите кнопку Powerpoint в верхнем центральном меню.

### Работа с меню «дизайн» в powerpoint

Работа с меню Дизайн предназначена для оформления слайдов и редактирования презентации. Рассмотрим вкладки меню Дизайна;

На вкладке Дизайн можно добавить тему или цветовую схему, или отформатировать фон слайда.

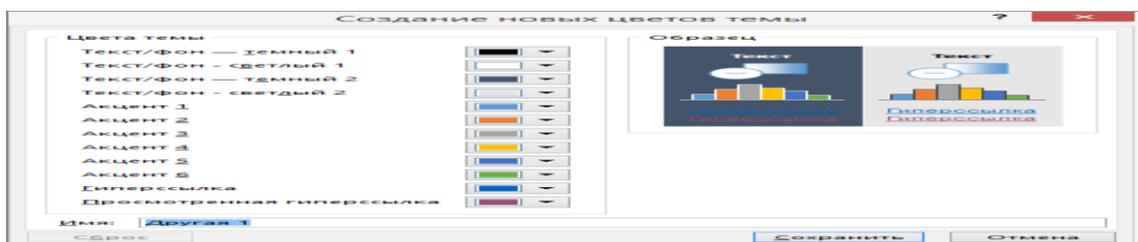


Изменение цветов темы

Цвета темы включают четыре цвета для текста и фона, шесть цветов для выделения и два цвета для гиперссылок. В разделе **Образец** можно просмотреть, как будут выглядеть стили и цвета шрифтов текста, перед выбором сочетания цветов.

1. На вкладке **Вид** выберите **Образец слайдов**, нажмите кнопку **Цвета**  и выберите команду **Настроить цвета**. Цвета, показанные на кнопке **Цвета** , представляют тему, примененную к презентации.

2. В диалоговом окне **Создание новых цветов темы** в разделе **Цвета темы** выполните одно из указанных ниже действий.



3. Нажмите кнопку напротив элемента цвета темы (например, **Акцент 1** или **Гиперссылка**), который хотите изменить, а затем выберите цвет в меню **Цвета темы**.

ИЛИ

Нажмите кнопку **Другие цвета** и выполните одно из указанных ниже действий.

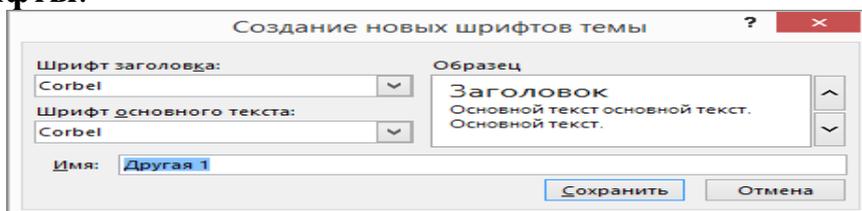
- ❖ На вкладке **Обычные** выберите цвет.
- ❖ На вкладке **Другие** введите номер формулы необходимого цвета.
- ❖ Повторите действие для всех элементов цвета темы, которые хотите изменить.

4. В поле **Имя** укажите название для нового набора цветов темы, а затем нажмите кнопку **Сохранить**.

Чтобы вернуть исходные цвета темы для всех элементов, перед нажатием кнопки **Сохранить** выберите команду **Сброс**.

Изменение шрифтов темы

1. На вкладке **Вид** нажмите кнопку **Образец слайдов**. На вкладке **Образец слайдов** откройте меню **Шрифты**  и выберите **Настроить шрифты**.



2. Выберите шрифты, которые следует использовать, в полях **Шрифт заголовка** и **Шрифт основного текста** диалогового окна **Создание новых шрифтов темы**.

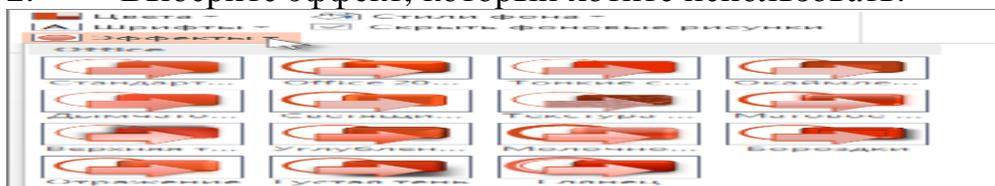
3. В поле **Имя** укажите название для нового набора шрифтов темы и нажмите кнопку **Сохранить**.

Выбор набора эффектов темы

Хотя пользователь не может создать свой набор эффектов темы, но может выбрать набор эффектов для презентации.

1. На вкладке **Вид** нажмите кнопку **Образец слайдов**. На вкладке **Образец слайдов** откройте меню **Эффекты** .

2. Выберите эффект, который хотите использовать.



Сохранение настроенной темы

Чтобы применять изменения цветов, шрифтов и эффектов к другим презентациям, их можно сохранить как тему (ТНМХ-файл).

1. На вкладке **Вид** нажмите кнопку **Образец слайдов**. На вкладке **Образец слайдов** выберите **Темы**.
2. Выберите команду **Сохранить текущую тему**.
3. В поле **Имя файла** укажите название новой темы и нажмите кнопку **Сохранить**.



## **Тема-8: Меню «Анимация» в PowerPoint.**

План:

1. **Работа с текстом**
2. **Работа с графикой**
3. **Работа с фигурами**

Если в нашей презентации только текст, то люди быстро устают.

Когда внимание рассеяно, донести основные мысли доклада невозможно в принципе.

Поэтому, с помощью анимации мы будем регулировать скорость чтения и управлять вниманием.

### **ОПРЕДЕЛЕНИЕ**

Компьютерная анимация — последовательный показ (слайд-шоу) заранее подготовленных графических файлов, а также компьютерная имитация движения с помощью изменения (и перерисовки) формы объектов или показа последовательных изображений с фазами движения.

Итак, из рассмотренного нами определения видно, что под анимацией понимаются движения на картинке.

**Однако, мы будем понимать анимацию, как те эффекты Power Point которые применяются к объектам и заставляют их двигаться, показываться и скрываться.**

**Как применить теорию на практике**

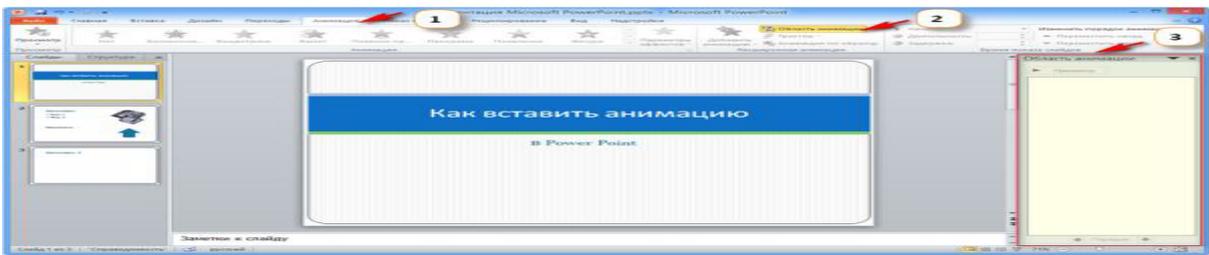
В Power Point есть четыре типа анимации:

- А) Входа;
- Б) Выделения;
- С) Выхода.
- Д) пути перемещения

Все вышеописанные типы характеризуются появлением, движением и исчезновением таких объектов как текст, графика, фигуры.

### **I. Работа с текстом**

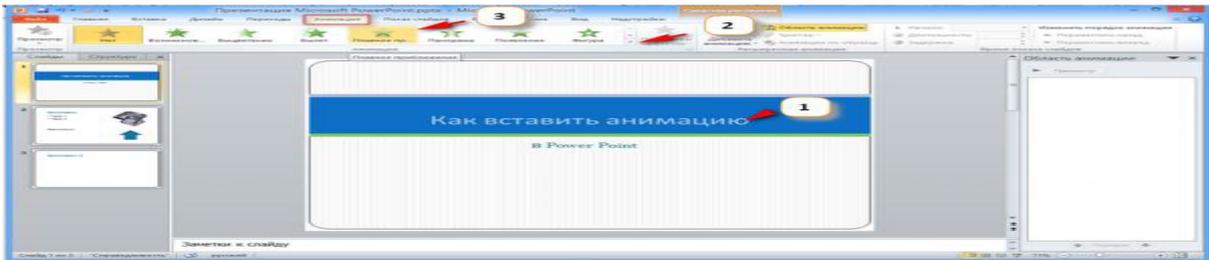
1. В начале откроем «пульт управления анимацией», нажав на закладке «Анимация» и затем на кнопке «Область анимации».



(Рисунок 1)

2. Затем выделив слайд начинаем применять анимацию ко всем его элементам.

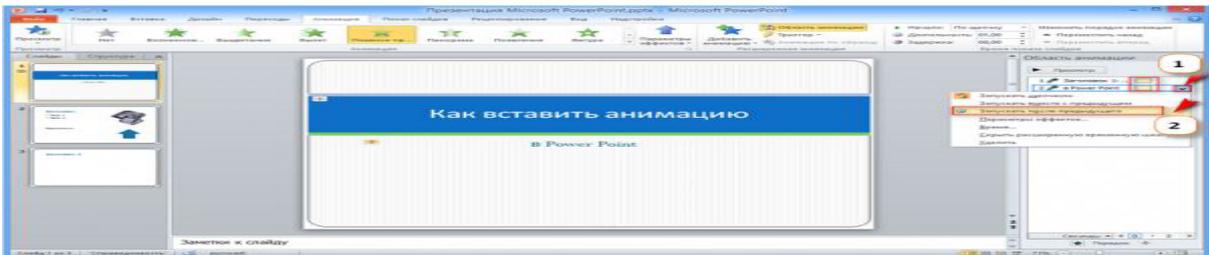
2.1. Выделяем заголовок презентации и на закладке «Анимация» выбираем тип анимации «Плавное приближение».



(Рисунок 2)

2.2. Выделяем подзаголовок и повторяем действия из пункта 2.1.

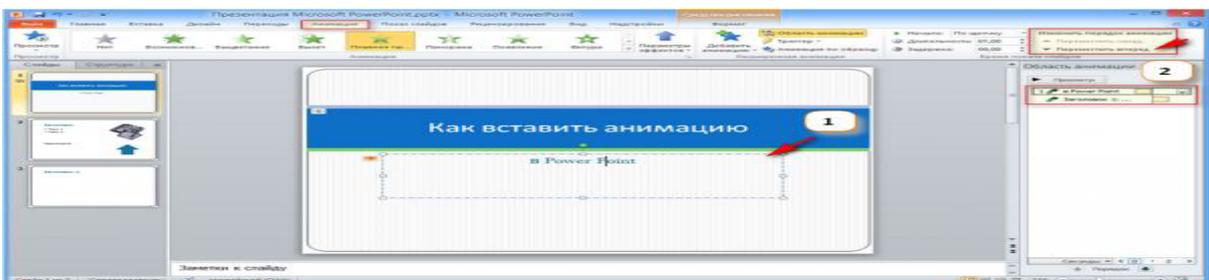
3. В области анимации кликаем на раскрывающееся меню и выбираем для подзаголовка время запуска – «После предыдущего».



(Рисунок 3)

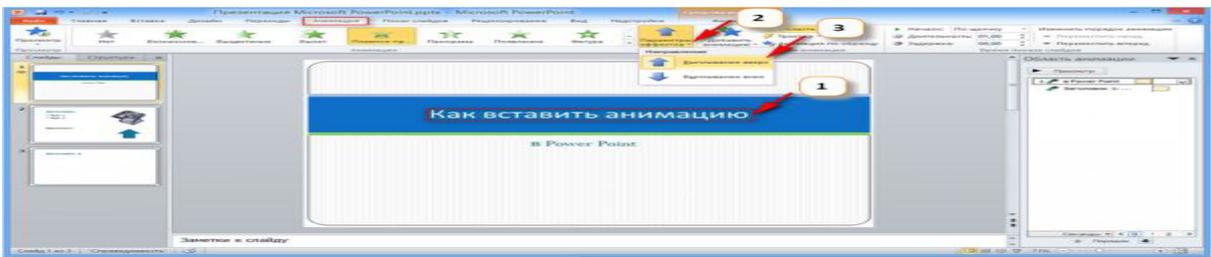
Информация:

- в окне «Область анимации» все элементы нумеруются;
- последовательность эффектов можно изменить, нажав во фрейме «Изменить порядок презентации» - «Переместить вперед».



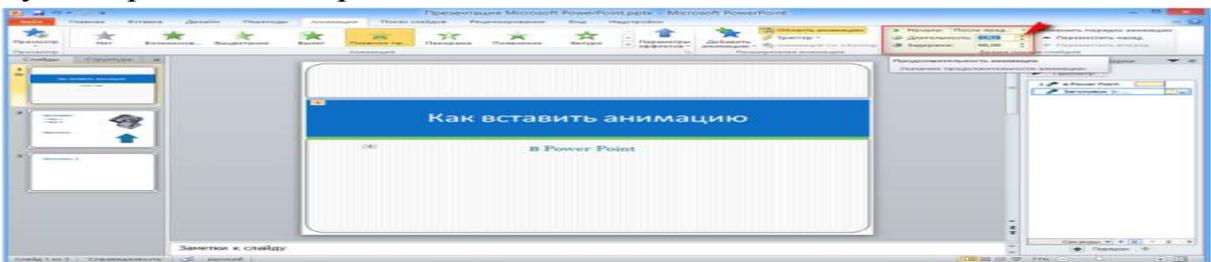
(Рисунок 4)

4. Изменяем направление движения текста, выделив текст и нажав на закладке «Анимация» кнопку «Параметры эффектов» - «Вверх».



(Рисунок 5)

5. Увеличиваем / уменьшаем **время воспроизведения анимации**, выделив подзаголовок и во фрейме «Время показа слайдов» увеличив время нажав на кнопку со стрелочкой вверх.

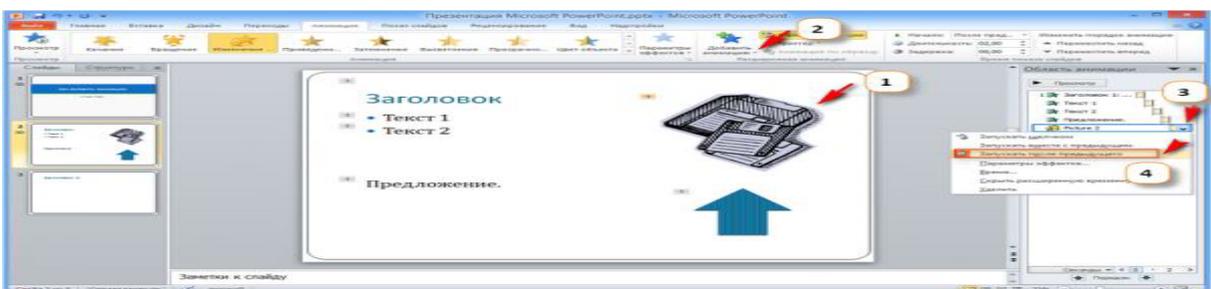


(Рисунок 6)

## II. Работа с графикой

6. Выделив картинку нажимаем на кнопку «Добавить анимацию» и выбираем «Покачивание» (Рисунок 7, пункт 1,2).

7. Так как анимация картинки будет показываться после текста, то мы нажимаем на кнопку выпадающего меню и выбираем «Запускать после предыдущего» (Рисунок 7, пункт 3,4).



(Рисунок 7)

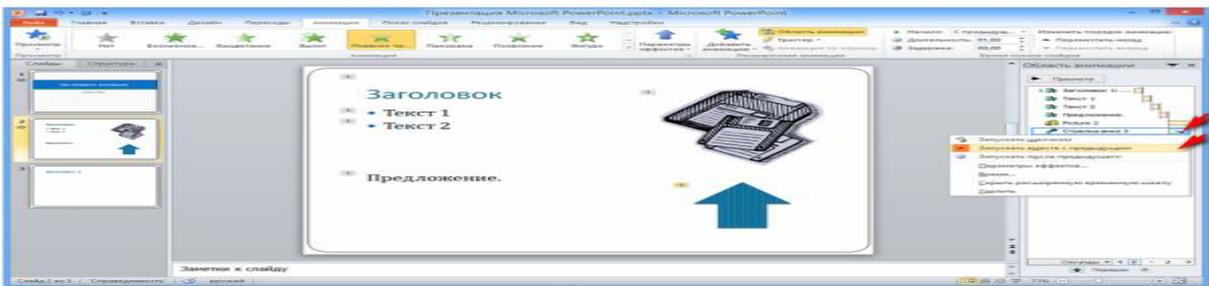
## III. Работа с фигурами

8. Выделив фигуру стрелочки нажимаем на кнопку «Добавить анимацию» и выбираем «Плавное приближение» .



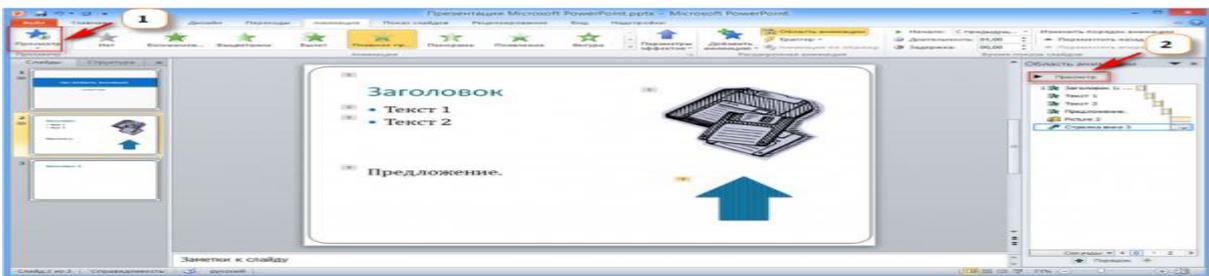
(Рисунок 8)

9. Так как стрелочка будет указывать на рисунок, то мы запустим ее по времени вместе с картинкой, а не с текстом. Вызываем меню настройки анимации фигуры «Стрелочка» и выбираем «Вместе с предыдущим».



(Рисунок 9)

10. Теперь можно нажать на кнопку «Просмотр», чтобы увидеть какие изменения мы внесли на втором слайде.



(Рисунок 10)

Профессиональный уровень

Если Вы хотите сделать сложные передвижения в презентации применив анимацию, то для этого:

- выделим объект и нажав «добавить анимацию», выберем «Пользовательский путь»;



(Рисунок 11)

- нарисовав появившимся карандашом путь, мы получаем очень сложное движение.



(Рисунок 12)

## Тема-9: Технологии демонстрации документов в компьютерах.

### План:

1. Изменение цветов темы
2. Изменение шрифтов темы

3. Выбор набора эффектов темы
4. Представление информации на экране
5. Работа с образцами.
6. Ввод и оформление текста
7. Текстовые эффекты в power point
8. Художественное оформление презентаций.
9. Шаблоны дизайна.

### **Представление информации на экране.**

PowerPoint предоставляет пользователю возможность работать и просматривать информацию в различных видах. В зависимости от того, что делаем: вводим текст и хотим рассмотреть его структуру, создаем заметки или вставляем в слайд графику — можно установить соответствующий вид и тем самым повысить удобство своей работы. Таких видов пять, и установить их можно, нажав одну из кнопок внизу главного окна программы.

**1. Вид слайдов** наиболее удобен, если мы постепенно формируем каждый слайд, выбираем для него оформление, вставляем текст или графику.

**2. Вид структуры** следует устанавливать для работы над текстом презентации. В этом случае возможно просмотреть заголовки всех слайдов, весь текст и структуру презентации.

**3. Вид сортировщика** слайдов наиболее удобен для добавления переходов и установки длительности пребывания слайда на экране. Кроме того, в этом режиме можно переставлять слайды.

**4. Вид заметок** предназначен для создания заметок к докладу.

**5. Демонстрация** используется для того, чтобы увидеть результаты работы. В этом режиме слайды по очереди выводятся на экран. Установить нужный вид можно и с помощью команд из меню **Вид**.

### **Работа с образцами.**

Вид презентации будет лучше, если мы оформим все ее слайды в одном стиле. Кроме того, часто возникает необходимость размещения на всех слайдах одного и того же элемента дизайна. Поэтому в PowerPoint существует возможность задания для всех слайдов и страниц одинаковое оформление. Это делается в режиме работы с образцами.

Чтобы войти в этот режим, нужно выбрать в меню **Вид** команду **Образец**, а в открывшемся подменю — элемент презентации, образец которого мы хотим исправить по-своему усмотрению.

Для слайдов в меню предназначены две команды — **Образец слайдов** и **Образец заголовков**. Вторая команда применяется для определения образца титульных слайдов, вид всех остальных слайдов презентации определяется образцом слайдов

При выборе команды **Образец слайдов** видно, что в каждой области слайда содержится подсказка о том, что нужно делать для внесения тех или иных изменений в образец.

Мы можем установить тип, начертание и размер шрифта, задать параметры абзацев, изменить размеры областей образца, вставить в него рисунок или нарисовать какой-либо графический элемент.

Все помещенные в образец элементы появятся на каждом слайде презентации, а внесенные изменения сразу же отразятся на всех остальных слайдах.

Таким образом, в PowerPoint можно создать индивидуальный дизайн и определить элементы, которые должны быть одинаковыми для всей презентации.

### **Создание презентации “с нуля”.**

Если мы закрыли диалоговое окно, которое открывается при вызове PowerPoint, или закрыли файл презентации, с которым работали, то для создания новой презентации нам придется вызвать команду **Создать** из меню **Файл**. После этого на экране появится окно **Создать презентацию** с активным разделом **Дизайны презентаций**.

В этом диалоговом окне можно задать шаблон дизайна презентации. После щелчка на одном из шаблонов в окне **Просмотр** можно увидеть его изображение. Выбрав шаблон, выполним на нем двойной щелчок. В результате откроется диалоговое окно **Создать слайд**.

В области **Выберите авторазметку** можно определить авторазметку для вновь создаваемого слайда. В правом нижнем углу окна содержится ее краткая характеристика. При двойном щелчке на образце авторазметки на экране появится новый слайд, содержащий заглушки.

Окно для создания нового слайда можно вызвать, выбрав команду **Создать слайд** из меню **Вставить** или активизировав комбинацию клавиш (**Ctrl+M**).

Вернемся опять к окну **Создать презентацию**. Если активизировать панель **Презентации**, то можно запустить Мастер автосодержания или выбрать шаблон презентации.

Теперь активизируем раздел **Общие**. Мы увидим список шаблонов PowerPoint, которые содержатся в папке MS OFFICE / ШАБЛОНЫ. Сейчас здесь находится только шаблон **Новая презентация**.

Выполним щелчок в этом шаблоне, в результате на экране появится окно **Создать слайд**. Выберем двойным щелчком нужный вид слайда, после чего слайд появится на экране. Однако он не будет иметь художественного оформления, и мы можем самостоятельно полностью разработать его дизайн.

### **Ввод и оформление текста.**

#### *Вид структуры.*

Создавая презентацию с помощью Мастера автоформы, мы уже вводили текст непосредственно в слайд. Однако во многих случаях работать с текстом гораздо удобнее, если презентация представлена в виде структуры. В этом случае легче увидеть структуру презентации и удобнее править ее текст, поскольку работа идет в одном окне. Для замены текста, который содержится, например, в шаблоне, нужно выделить его, а затем ввести новой текст.

Отметим только следующее:

— Для перемещения слайда вместе с содержимым нужно перетащить значок слайда (он расположен слева от заголовка) в нужное место документа.

— Для создания нового элемента списка необходимо установить курсор в конец существующего списка и нажать кнопку (Enter).

— Для перемещения элемента списка в другую позицию нужно установить курсор слева от него (курсор приобретает вид крестика) и перетащить данный элемент в нужное место.

— Для создания нового слайда сначала нужно создать элемент списка, а затем нажать кнопку **Повысить уровень** до тех пор, пока маркер не превратится в значок слайда.

— Если мы хотим перейти из вида структуры в вид слайдов, нужно выполнить двойной щелчок на значке соответствующего слайда.

### **Текстовые эффекты в PowerPoint.**

Удачно выбранный шрифт и различные текстовые эффекты, которые предоставляет в распоряжение пользователя PowerPoint, улучшат внешний вид нашей презентации и сделают ее более удобной для восприятия.

Чтобы задать для каких-либо текстовых элементов всей презентации одинаковый шрифт, нужно выбрать в меню **Вид** команду **Образец слайда**. Затем выполнить щелчок в области, для которой мы хотим провести изменения (вокруг нее появится рамка), и выбрать в панели инструментов соответствующую

кнопку для задания типа, размера и начертания шрифта. Можно также задать для символов тень (кнопка **Тень текста**) или рельефное начертание (меню **Формат** команда **Шрифт**). Если потребуется изменить регистр выделенного текста, можно воспользоваться командой **Регистр** меню **Формат**.

Мы можем создать и фигурный текст, например, расположить символы вдоль кривой или окружности или поместить внутри какой-либо фигуры. Для этого из меню **Объект** необходимо выбрать команду **Вставка**, а из списка **Тип объекта** — элемент **Microsoft WordArt**. После этого на слайде появится область, в которой будет размещен фигурный текст, и откроется окно для ввода самого текста. Введем текст. Если выполнить щелчок на кнопке **Обновить экран**, то он появится в области слайда.

Панель инструментов включает в себя списки для выбора формы фигурного текста, типа и размера шрифта, а также кнопки, которые предназначены для задания различных эффектов.

### **Художественное оформление презентаций.**

Художественное оформление является очень важным этапом разработки презентаций, т.к. , во-первых, представленные в графическом виде данные часто выглядят лучше текстовых, во-вторых, использование графики позволяет выделить наиболее важные моменты презентации или облегчить понимание трудных положений доклада. И вообще, чем эффектней презентация, тем лучшее впечатление на слушателей произведет выступление.

### **Шаблоны дизайна.**

PowerPoint предлагает пользователю большое количество шаблонов дизайна, разработанных профессионалами. Задать для презентации шаблон дизайна можно при помощи команды **Применить шаблон дизайна** меню **Формат**. После вызова команды откроется одноименное диалоговое окно. Выполнив щелчок на имени шаблона дизайна, мы видим его изображение в окне предварительного просмотра. Для выбора того или иного шаблона достаточно произвести на его имени двойной щелчок.

### **Рисование графических объектов.**

В PowerPoint можно самим нарисовать графический объект практически любой степени сложности. Для рисования предназначена панель инструментов **Рисование**, которая появляется на экране, если мы находимся в виде слайдов или в виде заметок. Дополнительные средства рисования размещаются на

панели инструментов Рисование+, для отображения которой требуется установить указатель мыши на любую панель инструментов, нажать правую кнопку мыши и в появившемся контекстном меню выбрать команду Рисование+.

А сейчас рассмотрим основные приемы рисования в PowerPoint :

— Чтобы начать рисовать какую-либо фигуру, надлежит выбрать инструмент рисования, нажав соответствующую кнопку на панели инструментов.

— Чтобы нарисовать несколько объектов одного типа или выполнить над готовыми объектами одинаковые действия, не вызывая каждый раз один и тот же инструмент, следует выполнить на его кнопке двойной щелчок.

— Чтобы выполнить над созданной фигурой какие-либо действия, необходимо сначала выделить ее, выполнив щелчок мышью.

— Чтобы выделить объект, который полностью или частично скрыт другими объектами, следует выделить объект, расположенный на первом плане, а потом нажимать клавишу (Tab) до тех пор, пока не будет выделен нужный объект.

— Чтобы инструментом **Эллипс** нарисовать окружность, инструментом **Прямоугольник**

- квадрат, а инструментом **Дуга** — дугу окружности, следет во время рисования удерживать нажатой клавишу (Shift).

— Чтобы линия, которая рисуется инструментом **Линия**, была горизонтальной или вертикальной, следует во время рисования удерживать нажатой клавишу (Shift).

— Чтобы выделить несколько объектов, необходимо последовательно выполнять на них щелчки мышью, удерживая нажатой клавишу (Shift). Все объекты можно выделить, нажав комбинацию клавиш (Ctrl+A).

## **Тема-10: Электронная таблица Excel. Окно Excel.**

### **План:**

1. Интерфейс и управление
2. Ввод и редактирование данных
3. Форматирование данных
4. Формат данных ячеек

**Ключевые слова:** ячейка, формула, функция, аргумент, диапазон, диаграмма, таблица, график.

Самым популярным приложением, входящим в пакет Microsoft Office, является тестовый редактор (процессор) Word. Однако есть и еще одна программа, без которой редко обходятся любой офисный работник. Microsoft Excel (Эксель) относится к программным продуктам, которые называются электронными таблицами. С помощью Excel, в наглядной форме, можно посчитать и автоматизировать расчеты практически всего что угодно, начиная с личного месячного бюджета и заканчивая сложными математическими и экономико-статистическими выкладками, содержащими большие объемы массивов данных.

Теперь, после окончания вводной части, пора переходить к делу. В первой части цикла, для лучшего усвоения материала, в качестве примера мы создадим

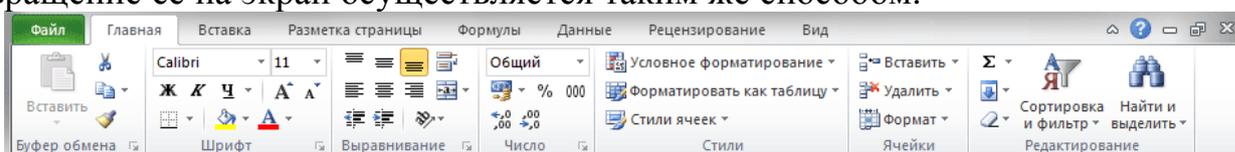
обычную таблицу, отражающую личные бюджетные расходы за полгода вот такого вида:

	Январь	Февраль	Март	Аперль	Май
Продукты питания	18 250р.	15 720р.	17 980р.	16 540р.	15 230р.
Коммунальные платежи	5 100р.	4 980р.	5 130р.	5 250р.	4 830р.
Покупка вещей	6 250р.	0р.	4 800р.	1 380р.	6 900р.
Обслуживание автомобиля	2 500р.	3 700р.	1 250р.	4 620р.	2 940р.
Выплата кредитов	3 300р.				

Но перед тем как начинать ее создание, давайте сначала рассмотрим основные элементы интерфейса и управления Excel, а так же поговорим о некоторых базовых понятиях этой программы.

## Интерфейс и управление

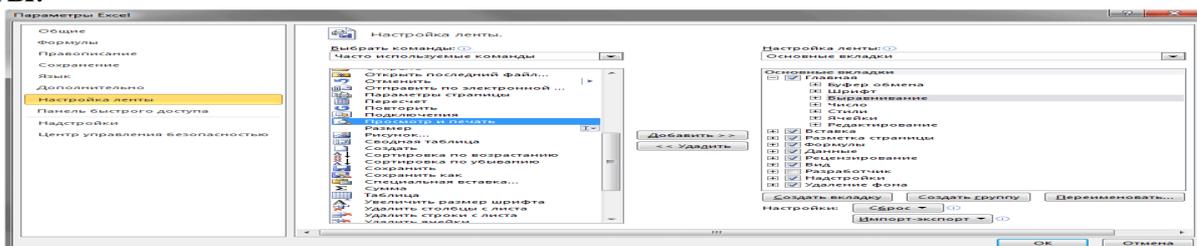
Если вы уже знакомы с редактором Word, то разобраться в интерфейсе Excel не составит труда. Ведь в его основе лежит та же **Лента**, но только с другим набором вкладок, групп и команд. При этом чтобы расширить рабочую область, некоторые группы вкладок выводятся на дисплей только в случае необходимости. Так же ленту можно свернуть вовсе, щелкнув по активной вкладке два раза левой кнопкой мыши или нажав сочетание клавиш Ctrl+F1. Возвращение ее на экран осуществляется таким же способом.



Стоит отметить, что в Excel для одной и той же команды может быть предусмотрено сразу несколько способов ее вызова: через ленту, из контекстного меню или с помощью сочетания горячих клавиш. Знание и использование последних может сильно ускорить работу в программе.

Контекстное меню является контекстно-зависимым, то есть его содержание зависит от того, что пользователь делает в данный момент. Контекстное меню вызывается нажатием правой кнопки мыши почти на любом объекте в MS Excel. Это позволяет экономить время, потому что в нем отображаются наиболее часто используемые команды к выбранному объекту.

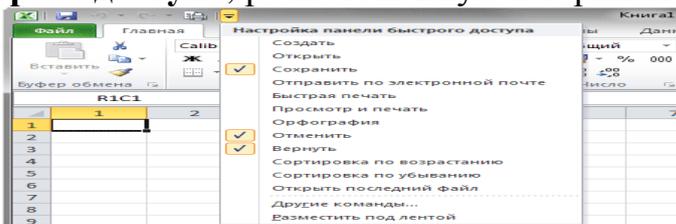
Не смотря на такое разнообразие управления, разработчики пошли дальше и предоставили пользователям в Excel 2010 возможность вносить изменения во встроенные вкладки и даже создавать собственные с теми группами и командами, которые используются наиболее часто. Для этого необходимо кликнуть правой кнопкой мыши на любой вкладке и выбрать пункт **Настройка ленты**.



В открывшемся окне, в меню справа, выберите нужную вкладку и щелкните по кнопке **Создать вкладку** или **Создать группу**, а в левом меню нужную команду, после чего кликните кнопку **Добавить**. В этом же окне можно переименовать существующие вкладки и удалять их. Для отмены

ошибочных действий существует кнопка **Сброс**, возвращающая настройки вкладок к начальным.

Так же наиболее часто используемые команды можно добавить на **Панель быстрого доступа**, расположенную в верхнем левом углу окна программы.

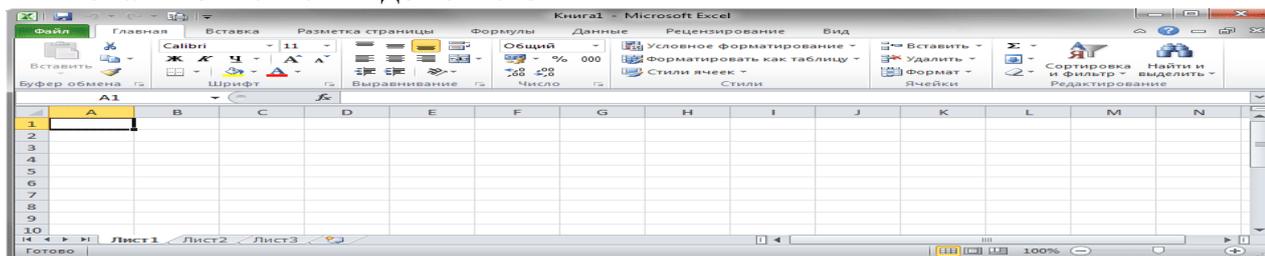


Сделать это можно нажав на кнопку **Настройка панели быстрого доступа**, где достаточно выбрать нужную команду из списка, а в случае отсутствия в нем необходимой, щелкнуть на пункт **Другие команды**.

## Ввод и редактирование данных

Создаваемые в Excel файлы называются рабочими книгами и имеют расширение «xls» или «xlsx». В свою очередь рабочая книга состоит из нескольких рабочих листов. Каждый рабочий лист – это отдельная электронная таблица, которые в случае необходимости могут быть взаимосвязаны. Активной рабочей книгой является та, с которой в данный момент вы работаете, например, в которую вводите данные.

После запуска приложения автоматически создается новая книга с именем «Книга1». По умолчанию рабочая книга состоит из трех рабочих листов с именами от «Лист1» до «Лист3».

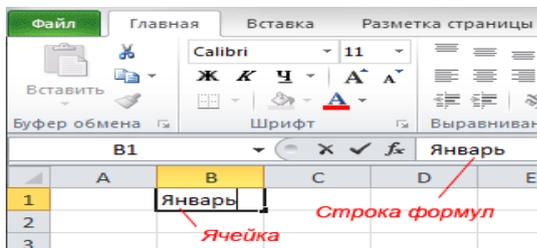


Рабочее поле листа Excel поделено на множество прямоугольных ячеек. Объединённые по горизонтали ячейки составляют строки, а по вертикали – столбцы. Для возможности исследования большого объема данных каждый рабочий лист программы имеет 1 048 576 строк пронумерованных числами и 16 384 столбцов обозначенных буквами латинского алфавита.

Таким образом, каждая ячейка – это место пересечения различных столбцов и строк на листе, формирующей ее собственный уникальный адрес, состоящий из буквы столбца и номера строки, которым она принадлежит. Например, имя первой ячейки – A1, так как она находится на пересечении столбца «А» и строки «1».

Если в приложении включена **Строка формул**, которая расположена сразу же под **Лентой**, то слева от нее находится **Поле имени**, где отображается имя текущей ячейки. Сюда же вы всегда сможете ввести имя искомой ячейки, для быстрого перехода к ней. Особенно эта возможность полезна в больших документах, содержащих тысячи строк и столбцов.

Так же для просмотра разных областей листа, внизу и справа располагаются полосы прокрутки. Помимо этого перемещаться по рабочей области Excel можно с помощью клавиш со стрелками.



Чтобы начать вводить данные в нужную ячейку, ее необходимо выделить. Для перехода в нужную ячейку щелкните на ней левой кнопкой мыши, после чего она будет окружена черной рамкой, так называемым индикатором активной ячейки. Теперь просто начинайте печатать на клавиатуре, и вся вводимая информация будет оказываться в выбранной ячейке.

При вводе данных в ячейку можно так же использовать и строку формул. Для этого выделите нужную клетку, а затем щелкните мышью на поле строки формул и начните печатать. При этом вводимая информация будет автоматически отображаться в выбранной ячейке.

После окончания ввода данных нажмите:

- Клавишу «Enter» - следующей активной ячейкой станет клетка снизу.
- Клавишу «Tab» - следующей активной ячейкой станет клетка справа.
- Щелкните мышью по любой другой ячейке, и она станет активной.

Чтобы изменить или удалить содержимое любой ячейки, щелкните на ней два раза левой кнопкой мыши. Перемещайте мигающий курсор в нужное место для внесения необходимых правок. Как и во многих других приложениях, для удаления и внесения исправлений используются клавиши стрелок, «Del» и «Backspace». При желании, все необходимые правки можно производить и в строке формул.

Объем данных, который вы будете вводить в ячейку, не ограничивается ее видимой частью. То есть клетки рабочего поля программы могут содержать как одну цифру, так и несколько абзацев текста. Каждая ячейка Excel способна вместить до 32 767 числовых или текстовых символов.

## Форматирование данных ячеек

После ввода названий строк и столбцов получаем таблицу такого вида:

	A	B	C	D	E
1		Январь	Февраль	Март	Аперл
2	Продукты питания				
3	Коммунальные платежи				
4	Покупка вещей				
5	Обслуживание автомобиля				
6	Выплата кредитов				
7					
8					
9					

Как видно из нашего примера, несколько названий статей расходов «вышли» за рамки границ ячейки и если соседняя клетка (клетки) тоже будет содержать какую-то информацию, то введенный текст частично перекрывается ей и становится невидим. Да и сама таблица выглядит довольно некрасивой и непрезентабельной. При этом если напечатать такой документ, то сложившаяся ситуация сохранится – разобрать в такой таблице что к чему будет довольно сложно, в чем вы можете сами убедиться из рисунка ниже.

	Январь	Февраль	Март	Аперль	Май	Июнь
Продукты	18250	15720	17980	16540	15230	16100
Коммунал	5100	4980	5130	5250	4830	5180
Покупка ве	6250	0	4800	1380	6900	0
Обслужива	2500	3700	1250	4620	2940	5730
Выплата кр	3300	3300	3300	3300	3300	3300

Чтобы сделать табличный документ более аккуратным и красивым, часто приходится изменять размеры строк и столбцов, шрифт содержимого ячейки, ее фон, осуществлять выравнивание текста, добавлять границы и прочее.

Для начала давайте приведем в порядок левый столбец. Переведите курсор мыши на границу столбцов «А» и «В» в строку, где отображаются их названия. При изменении курсора мыши на характерный символ с двумя разнонаправленными стрелками, нажмите и удерживая левую клавишу, тащите появившуюся пунктирную линию в нужном направлении для расширения столбца до тех пор пока все названия не уместятся в рамках одной ячейки.

	А	В	С	Д
1		Январь	Февраль	Март
2	Продукты питания	18250	15720	17980
3	Коммунальные платежи	5100	4980	5130
4	Покупка вещей	6250	0	4800
5	Обслуживание автомобиля	2500	3700	1250
6	Выплата кредитов	3300	3300	3300
7				
8				

Те же самые действия можно проделать и со строкой. Это является одним из самых легких способов изменения размера высоты и ширины ячеек.

Если необходимо задать точные размеры строк и столбцов, то для этого на вкладке **Главная** в группе **Ячейки** выберите пункт **Формат**. В открывшемся меню с помощью команд **Высота строки** и **Ширина столбца** можно задать эти параметры вручную.

Очень часто необходимо изменить параметры сразу нескольких ячеек и даже целого столбца или строки. Для того что бы выделить целый столбец или строку, щелкните на его названии сверху или на ее номере слева соответственно.

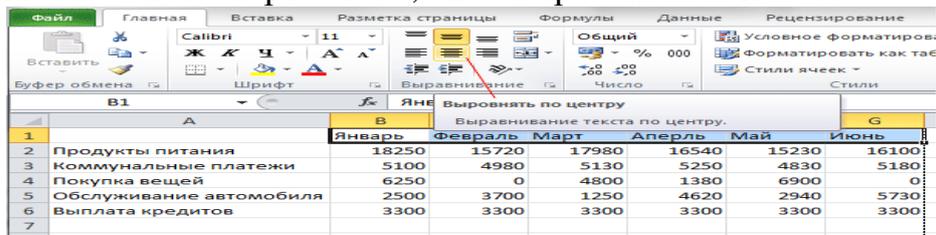
	А	В	С
1		Январь	Февраль
2	Продукты питания	18250	15720
3	Коммунальные платежи	5100	4980
4	Покупка вещей	6250	0
5	Обслуживание автомобиля	2500	3700
6	Выплата кредитов	3300	3300
7			

Для выделения группы соседних ячеек, обведите их курсором, удерживайте левую кнопку мыши. Если же необходимо выделить разрозненные поля таблицы, то нажмите и удерживайте клавишу «Ctrl», после чего щелкайте мышью по необходимым ячейкам.

	А	В	С
1		Январь	Февраль
2	Продукты питания	18250	15720
3	Коммунальные платежи	5100	4980
4	Покупка вещей	6250	0
5	Обслуживание автомобиля	2500	3700
6	Выплата кредитов	3300	3300
7			

Теперь, когда вы знаете, как выделять и форматировать сразу несколько ячеек, давайте выровняем название месяцев в нашей таблице по центру. Различные команды выравнивания содержимого внутри клеток находятся на вкладке **Главная** в группе с говорящим названием **Выравнивание**. При этом

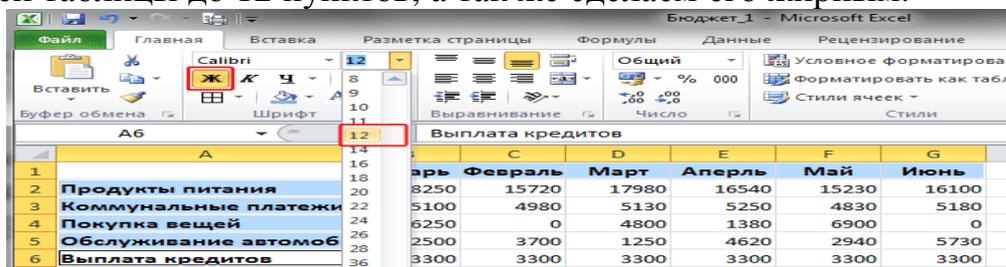
для табличной ячейки это действие можно производить как относительно горизонтального направления, так и вертикального.



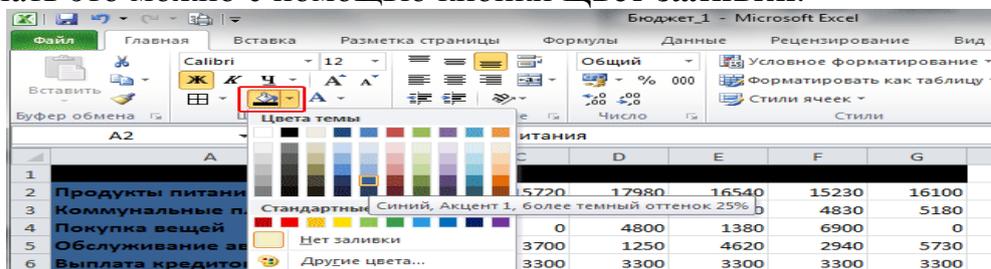
Обведите ячейки с названием месяцев в шапке таблицы и щёлкните на кнопке **Выровнять по центру**.

В группе **Шрифт** на вкладке **Главная** можно изменить тип шрифта, его размер, цвет и начертание: жирный, курсивный, подчеркнутый и так далее. Так же здесь размещены кнопки изменения границ ячейки и цвета ее заливки. Все эти функции пригодятся нам для дальнейшего изменения внешнего вида таблицы.

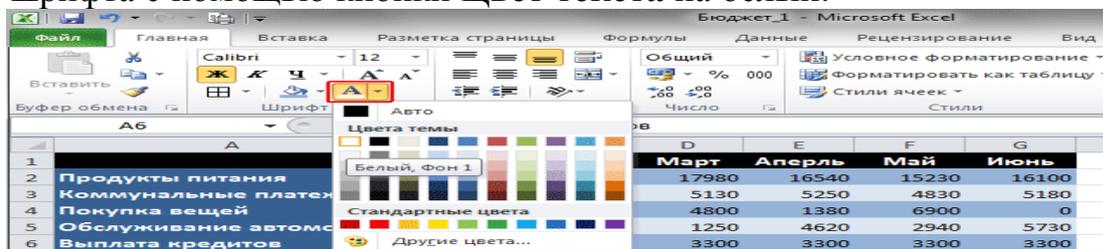
Итак, для начала давайте увеличим шрифт названия колонок и столбцов нашей таблицы до 12 пунктов, а так же сделаем его жирным.



Теперь выделяем сначала верхнюю строку таблицы и устанавливаем ей черный фон, а затем в левом столбце ячейкам с A2 по A6 – темно-синий. Сделать это можно с помощью кнопки **Цвет заливки**.

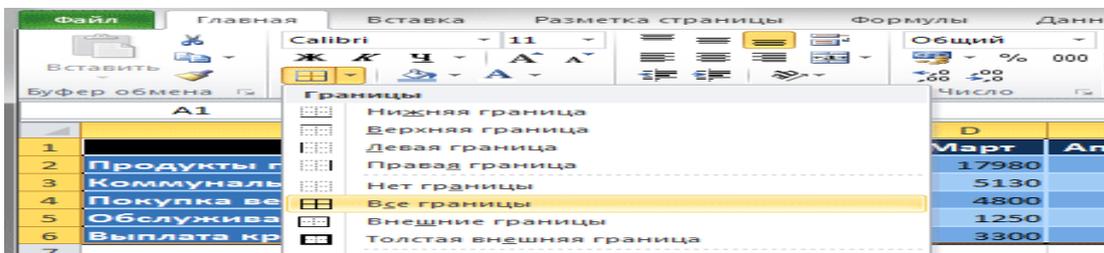


Наверняка вы заметили, что цвет текста в верхней строке слился с цветом фона, да и в левом столбце названия читаются плохо. Исправим это, изменив цвет шрифта с помощью кнопки **Цвет текста** на белый.

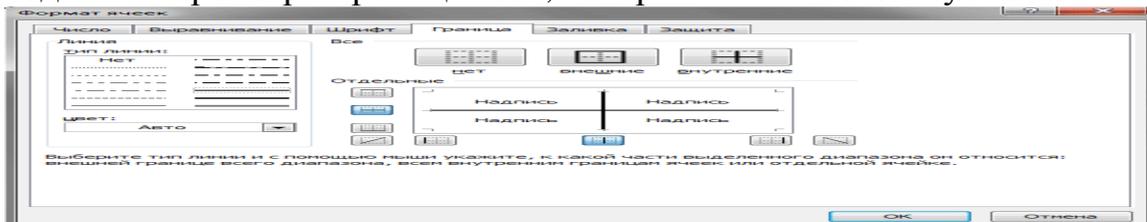


Так же с помощью уже знакомой команды **Цвет заливки** мы придали фону четных и нечетных строк с числами различный синий оттенок.

Чтобы ячейки не сливались, давайте определим им границы. Определение границ происходит только для выделенной области документа, и может быть сделано как для одной ячейки, так и для всей таблицы. В нашем случае выделите всю таблицу, после чего щелкните на стрелочку рядом с кнопкой **Другие границы** все в той же группе **Шрифт**.



В открывшемся меню выводится список быстрых команд, с помощью которых можно выбрать отображение нужных границ выделенной области: нижней, верхней, левой, правой, внешних, всех и прочее. Так же здесь содержатся команды для рисования границ вручную. В самом низу списка находится пункт **Другие границы** позволяющий более детально задать необходимые параметры границ ячеек, которым мы и воспользуемся.



В открывшемся окне сначала выберите тип линии границы (в нашем случае тонкая сплошная), затем ее цвет (выберем белый, так как фон таблицы темный) и наконец, те границы, которые должны будут отображаться (мы выбрали внутренние).

	А	В	С	Д	Е	Ф	Г
		Январь	Февраль	Март	Аперль	Май	Июнь
1							
2	Продукты питания	18250	15720	17980	16540	15230	16100
3	Коммунальные платежи	5100	4980	5130	5250	4830	5180
4	Покупка вещей	6250	0	4800	1380	6900	0
5	Обслуживание автомобиля	2500	3700	1250	4620	2940	5730
6	Выплата кредитов	3300	3300	3300	3300	3300	3300

В итоге с помощью набора команд всего одной группы **Шрифт** мы преобразовали неказистый внешний вид таблицы во вполне презентабельный, и теперь зная как они работают, вы самостоятельно сможете придумывать свои уникальные стили для оформления электронных таблиц.

## Формат данных ячейки

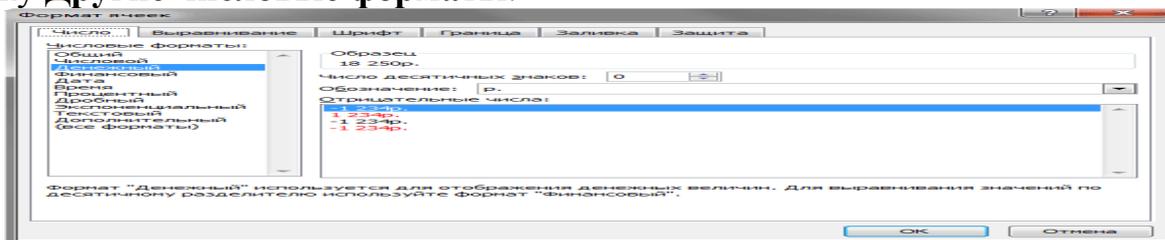
Теперь, что бы завершить нашу таблицу, необходимо надлежащим образом оформить те данные, которые мы туда вводим. Напомним, что в нашем случае это денежные расходы.

В каждую из ячеек электронной таблицы можно вводить разные типы данных: текст, числа и даже графические изображения. Именно поэтому в Excel существует такое понятие, как «формат данных ячейки», служащий для корректной обработки вводимой вами информации.

Изначально, все ячейки имеют **Общий формат**, позволяющий содержать им как текстовые, так и цифровые данные. Но вы вправе изменить это и выбрать: числовой, денежный, финансовый, процентный, дробный, экспоненциальный и форматы. Помимо этого существуют форматы даты, времени почтовых индексов, номеров телефонов и табельных номеров.

Для ячеек нашей таблицы, содержащих названия ее строк и столбцов вполне подойдет общий формат (который задается по умолчанию), так как они содержат текстовые данные. А вот для ячеек, в которые вводятся бюджетные расходы больше подойдет денежный формат.

Выделите в таблице ячейки содержащие информацию по ежемесячным расходам. На ленте во вкладке **Главная** в группе **Число** щелкните по стрелке рядом с полем **Числовой Формат**, после чего откроется меню со списком основных доступных форматов. Можно выбрать пункт **Денежный** прямо здесь, но мы для более полного ознакомления выберем самую нижнюю строку **Другие числовые форматы**.



В открывшемся окне в левой колонке будут отображены название всех числовых форматов, включая дополнительные, а в центре, различные настройки их отображения.

Выбрав денежный формат, сверху окна можно увидеть, то, как будет выглядеть значение в ячейках таблицы. Чуть ниже можно установить число отображения десятичных знаков. Чтобы копейки не загромождали нам поля таблицы, выставим здесь значение равным нулю. Далее можно выбрать валюту и отображение отрицательных чисел.

Теперь наша учебная таблица, наконец, приняла законченный вид:

	Январь	Февраль	Март	Аперль	Май
<b>Продукты питания</b>	18 250p.	15 720p.	17 980p.	16 540p.	15 230p.
<b>Коммунальные платежи</b>	5 100p.	4 980p.	5 130p.	5 250p.	4 830p.
<b>Покупка вещей</b>	6 250p.	0p.	4 800p.	1 380p.	6 900p.
<b>Обслуживание автомобиля</b>	2 500p.	3 700p.	1 250p.	4 620p.	2 940p.
<b>Выплата кредитов</b>	3 300p.				

Кстати, все манипуляции, которые мы проделывали с таблицей выше, то есть форматирование ячеек и их данных можно выполнить с помощью контекстного меню, щелкнув правой кнопкой мыши на выделенной области и выбрав пункт **Формат ячеек**. В одноименном открывшемся окне для всех рассмотренных нами операций существуют вкладки: **Число**, **Выравнивание**, **Шрифт**, **Граница** и **Заливка**.

Теперь по окончании работы в программе вы можете сохранить или распечатать полученный результат. Все эти команды находятся во вкладке **Файл**.

### Список литературы:

1. Симонич С.В. и другие. Специальная информатика. Учебное пособие. –М.: Издательство «АСТПресс», 2001
2. Куртис Фрай "Microsoft Excel 2013. Шаг За Шагом"

## Тема-11: Работа с формулами в Excel.

### План:

1. Формулы
2. Функции
3. Вложенные функции.
4. О ссылках в формулах
5. Об операторах в формулах
6. Порядок выполнения действий в формулах

**Ключевые слова: ячейка, формула, функция, аргумент, диапазон, диаграмма, таблица, график.**

### **Формулы**

Возможность использования формул и функций является одним из важнейших свойств программы обработки электронных таблиц. Это, в частности, позволяет проводить статистический анализ числовых значений в таблице.

Текст формулы, которая вводится в ячейку таблицы, должен начинаться со знака равенства (=), чтобы программа Excel могла отличить формулу от текста. После знака равенства в ячейку записывается математическое выражение, содержащее аргументы, арифметические операции и функции.

#### **Элементы формулы**

**Функции.** Функция ПИ() возвращает значение числа Пи: 3,142...

**Ссылки (или имена).** A2 возвращает значение ячейки A2.

**Константы.** Числа или текстовые значения, введенные непосредственно в формулу, например 2.

**Операторы.** Оператор ^ возводит число в степень, а звездочка (\*) выполняет умножение.

#### **О функциях в формулах**

**Функции** - заранее определенные формулы, которые выполняют вычисления по заданным величинам, называемым аргументами, и в указанном порядке. Эти функции позволяют выполнять как простые, так и сложные вычисления. Например, функция ОКРУГЛ округляет число в ячейке A10.

#### **Структура функции**

Структура функции начинается со знака равенства (=), за ним следует имя функции, открывающая скобка, список аргументов, разделенных запятыми, закрывающая скобка.

**Аргументы.** Существуют различные типы аргументов: число, текст, логическое значение (ИСТИНА и ЛОЖЬ), массивы, значение ошибки (например #Н/Д), или ссылки на ячейку. В качестве аргументов используются константы (постоянное (не вычисляемое) значение) формулы, или функции. В каждом конкретном случае необходимо использовать соответствующий тип аргумента.

#### **Вложенные функции.**

В некоторых случаях может потребоваться использование функции как одного из аргументов другой функции. Например, в следующей формуле функция СРЗНАЧ вложена в функцию ЕСЛИ для сравнения среднего значения нескольких значений с числом 50.

#### **О ссылках в формулах**

Ссылка указывает на ячейку или диапазон ячеек листа и передает в Microsoft Excel сведения о расположении значений или данных, которые требуется использовать в формуле. При помощи ссылок можно использовать в одной формуле данные, находящиеся в разных частях листа, а также использовать в нескольких формулах значение одной ячейки. Кроме того, можно задавать ссылки на ячейки других листов той же книги и на другие книги. Ссылки на ячейки других книг называются связями.

#### **Стиль ссылок A1**

По умолчанию Microsoft Excel использует стиль ссылок A1, определяющий столбцы буквами (от A до IV, всего не более 256 столбцов), а строки номерами (от 1 до 65536). Эти буквы и номера называются заголовками строк и столбцов. Для ссылки на ячейку введите букву столбца, а следом номер строки.

**Относительные ссылки.** Относительная ссылка в формуле, например A1, основана на относительной позиции ячейки, содержащей формулу, и ячейку, на которую указывает ссылка. При изменении позиции ячейки, содержащей формулу, изменяется и ссылка. При копировании формулы вдоль строк и вдоль столбцов ссылка автоматически корректируется. По умолчанию в новых формулах используются относительные ссылки.

**Абсолютные ссылки.** Абсолютная ссылка ячейки в формуле, например \$A\$1, всегда ссылается на ячейку, расположенную в определенном месте. При изменении позиции ячейки, содержащей формулу, абсолютная ссылка не изменяется. При копировании формулы вдоль строк и вдоль столбцов абсолютная ссылка не корректируется. По умолчанию в новых формулах используются относительные ссылки, и для использования абсолютных ссылок надо выбрать соответствующий параметр.

**Смешанные ссылки.** Смешанная ссылка содержит либо абсолютный столбец и относительную строку, либо абсолютную строку и относительный столбец. Абсолютная ссылка столбцов приобретает вид \$A1, \$B1 и т.д. Абсолютная ссылка строки приобретает вид A\$1, B\$1 и т.д. При изменении позиции ячейки, содержащей формулу, относительная ссылка изменяется, а абсолютная ссылка не изменяется. При копировании формулы вдоль строк и вдоль столбцов относительная ссылка автоматически корректируется, а абсолютная ссылка не корректируется.

### **Стиль ссылок R1C1**

Также можно использовать стиль ссылок, в котором нумеруются как строки, так и столбцы. Стиль ссылок R1C1 полезен при вычислении положения столбцов и строк в макросах. В стиле ссылок R1C1 Microsoft Excel указывает положение ячейки буквой «R», за которой идет номер строки, и буквой «C», за которой идет номер столбца.

При записи макроса Microsoft Excel записывает некоторые команды с использованием стиля ссылок R1C1. Например, если записывается такая команда как нажатие кнопки **Автосумма** для вставки формулы, суммирующей диапазон ячеек, Microsoft Excel использует при записи формулы стиль ссылок R1C1, а не A1.

### **Об операторах в формулах**

Операторами обозначаются операции, которые следует выполнить над операндами формулы. В Microsoft Excel включено четыре вида операторов: арифметические, текстовые, операторы сравнения и операторы ссылок.

#### Типы операторов

**Арифметические операторы.** Служат для выполнения арифметических операций, таких как сложение, вычитание, умножение. Операции выполняются над числами. Используются следующие арифметические операторы.

<b>Арифметический оператор</b>	<b>Значение (пример)</b>	
--------------------------------	------------------------------	--

+ (знак плюс)	Сложение (3+3)	
- (знак минус)	Вычитание (3-1) Отрицание (-1)	
* (звездочка)	Умножение (3*3)	
/ (косая черта)	Деление (3/3)	
% (знак процента)	Процент (20%)	
^ (крышка)	Возведение в степень (3^2)	

**Операторы сравнения.** Используются для сравнения двух значений. Результатом сравнения является логическое значение: либо ИСТИНА, либо ЛОЖЬ.

Оператор сравнения	Значение (пример)	
= (знак равенства)	Равно (A1=B1)	
> (знак больше)	Больше (A1>B1)	
< (знак меньше)	Меньше (A1<B1)	
>= (знак больше или равно)	Больше или равно (A1>=B1)	
<= (знак меньше или равно)	Меньше или равно (A1<=B1)	
<> (знак не равно)	Не равно (A1<>B1)	

**Оператор ссылки.** Для описания ссылок на диапазоны ячеек используются следующие операторы.

Оператор ссылки	Значение (пример)	
:	Ставится между ссылками на первую и последнюю ячейки диапазона. Такое сочетание является ссылкой на диапазон (B5:B15)	
;(точка с запятой)	Оператор объединения. Объединяет несколько ссылок в одну ссылку (СУММ (B5:B15; D5:D15))	
(пробел)	Оператор пересечения множеств, служит для ссылки на общие ячейки двух диапазонов (B7:D7 C6:C8)	

### Порядок выполнения действий в формулах

Формулы вычисляют значения в определенном порядке. Формула в Microsoft Excel всегда начинается со знака равенства (=). Знак равенства

свидетельствует о том, что последующие знаки составляют формулу. Элементы, следующие за знаком равенства, являются операндами, разделяемыми операторами вычислений. Формула вычисляется слева направо, в соответствии с определенным порядком для каждого оператора в формуле.

### Приоритет оператора

Если в одной формуле используется несколько операторов, Microsoft Excel выполняет операции в порядке, показанном в следующей таблице. Если формула содержит операторы с одинаковым приоритетом - например операторы деления и умножения - они выполняются слева направо.

Оператор	Описание
: (двоеточие) (один пробел) (запятая)	Операторы ссылок.
-	Знак «минус»
%	Процент
^	Возведение в степень
* и /	Умножение и деление
+ и -	Сложение и вычитание
&	Объединение двух текстовых строк в одну
= <> <= >= <>	Сравнение

Для того чтобы изменить порядок выполнения, заключите часть формулы, которая должна выполняться первой, в скобки.

### Создание формулы

Формулы представляют собой выражения, по которым выполняются вычисления на странице. Формула начинается со знака равенства (=).

Microsoft Excel - средство для работы с электронными таблицами, намного превышающее по своим возможностям существующие редакторы таблиц, первая версия данного продукта была разработана фирмой Microsoft в 1985 году. Табличные процессоры - удобный инструмент для экономистов, бухгалтеров, инженеров, научных работников - всех тех, кому приходится работать с большими массивами числовой информации. Microsoft Excel – это программа управления электронными таблицами общего назначения, которая используется для вычислений, организации и анализа деловых данных.

Excel, как видно из всего сказанного выше очень мощный инструмент для решения задач с использованием различных функций.

Функции – один из основных функционалов MS-Excel. По сути, функция – это заранее подготовленный кусок кода, выполняющий какую-то конкретную задачу. Функции в Excel используются для выполнения стандартных вычислений в рабочих книгах.

Основными задачами выполнения данной контрольной работы являются:

1) изучение использования функции в электронных таблицах Microsoft Excel;

2) способность определять, какую функцию использовать в конкретной ситуации;

3) научиться избегать ряда типичных ошибок при использовании функций.

Итак, статистика – это цифры живые, красноречивые. Однако это определение исходит из итогов статистической работы, результатом которой являются статистические цифры.

## Статические и динамические информационные модели

### Статистические функции в Excel

1. Среднее значение: =СРЗНАЧ(диапазон)
2. Квадратическое отклонение: =КВАДРОТКЛ(диапазон)
3. Дисперсия: =ДИСП(диапазон)
4. Дисперсия для генеральной совокупности: =ДИСПР(диапазон)
5. Среднеквадратическое отклонение: =СТАНДОТКЛОН(диапазон)
6. Коэффициент корреляции: =КОРРЕЛ(диапазон 1;диапазон 2)
7. Коэффициент корреляции Пирсона: =ПИРСОН(диапазон 1;диапазон 2)
- 2) 8. Коэффициент детерминации: =КВПИРСОН(изв\_значение\_y; изв\_значение\_x)
9. Стандартная ошибка S: =СТОШУХ(изв\_значение\_y; изв\_значение\_x)
10. Граничные точки: =СТЮДРАСПОБР( $\alpha$  ; n-m-1)
11. Прогноз на следующий период: =ТЕНДЕНЦИЯ(диапазон Y;диапазон X; новое значение X)
12. Ковариация: =КОВАР(массив1; массив2).
13. Уравнение регрессии  $y = bx+a$ : =ЛИНЕЙН(диапазон Y;диапазон X;1;1).

	X	Y
1	119	298,10
2	200,3	481,03
3	281	599,13
4	305	653,57
5	381	987,66
6	363	1252,85
7	389	1276,88
8	387	1396,7
9	315	952,03
10	217	619,96
11	149	384,4
12	192	516,59

○ Выделите блок ячеек размером 2 столбца и 5 строк.

	X	Y
1	119	298,10
2	200,3	481,03
3	281	599,13
4	305	653,57
5	381	987,66

○ Перейти в режим редактирования (клавиша F2);

	X	Y
1	119	298,10
2	200,3	481,03
3	281	599,13
4	305	653,57
5	381	987,66

○ Нажать клавиши **Ctrl+Shift+Enter**.

	A15	B	C	D	E
1	1	1	1	1	1
2	1	1	1	1	1
3	1	1	1	1	1
4	1	1	1	1	1
5	1	1	1	1	1
6	1	1	1	1	1
7	1	1	1	1	1
8	1	1	1	1	1
9	1	1	1	1	1
10	1	1	1	1	1
11	1	1	1	1	1
12	1	1	1	1	1
13	1	1	1	1	1
14	1	1	1	1	1
15	1	1	1	1	1
16	1	1	1	1	1
17	1	1	1	1	1
18	1	1	1	1	1
19	1	1	1	1	1
20	1	1	1	1	1
21	1	1	1	1	1
22	1	1	1	1	1
23	1	1	1	1	1
24	1	1	1	1	1
25	1	1	1	1	1
26	1	1	1	1	1
27	1	1	1	1	1
28	1	1	1	1	1
29	1	1	1	1	1
30	1	1	1	1	1
31	1	1	1	1	1
32	1	1	1	1	1
33	1	1	1	1	1
34	1	1	1	1	1
35	1	1	1	1	1
36	1	1	1	1	1
37	1	1	1	1	1
38	1	1	1	1	1
39	1	1	1	1	1
40	1	1	1	1	1
41	1	1	1	1	1
42	1	1	1	1	1
43	1	1	1	1	1
44	1	1	1	1	1
45	1	1	1	1	1
46	1	1	1	1	1
47	1	1	1	1	1
48	1	1	1	1	1
49	1	1	1	1	1
50	1	1	1	1	1
51	1	1	1	1	1
52	1	1	1	1	1
53	1	1	1	1	1
54	1	1	1	1	1
55	1	1	1	1	1
56	1	1	1	1	1
57	1	1	1	1	1
58	1	1	1	1	1
59	1	1	1	1	1
60	1	1	1	1	1
61	1	1	1	1	1
62	1	1	1	1	1
63	1	1	1	1	1
64	1	1	1	1	1
65	1	1	1	1	1
66	1	1	1	1	1
67	1	1	1	1	1
68	1	1	1	1	1
69	1	1	1	1	1
70	1	1	1	1	1
71	1	1	1	1	1
72	1	1	1	1	1
73	1	1	1	1	1
74	1	1	1	1	1
75	1	1	1	1	1
76	1	1	1	1	1
77	1	1	1	1	1
78	1	1	1	1	1
79	1	1	1	1	1
80	1	1	1	1	1
81	1	1	1	1	1
82	1	1	1	1	1
83	1	1	1	1	1
84	1	1	1	1	1
85	1	1	1	1	1
86	1	1	1	1	1
87	1	1	1	1	1
88	1	1	1	1	1
89	1	1	1	1	1
90	1	1	1	1	1
91	1	1	1	1	1
92	1	1	1	1	1
93	1	1	1	1	1
94	1	1	1	1	1
95	1	1	1	1	1
96	1	1	1	1	1
97	1	1	1	1	1
98	1	1	1	1	1
99	1	1	1	1	1
100	1	1	1	1	1

Методы сглаживания и выравнивания динамических рядов 2:

### Общее представление о функциях MS Excel

Существует несколько способов ввода функции:

- ввод функций вручную;
- ввод функции с помощью кнопки "сигма";
- работа с мастером функций;

Для набора простейших формул, содержащих функции, можно не пользоваться специальными средствами, а просто писать их. Однако этот способ плохо подходит для набора длинных формул.

Слишком большая вероятность допустить ошибку, набирая вручную сложные и длинные формулы, и на это уходит много времени.

Одно из средств облегчить и ускорить работы с функциями — кнопка на панели инструментов «Стандартная». В ней разработчики Microsoft «спрятали» пять часто используемых функций:

- СУММ(минимум один, максимум 30 аргументов). Суммирует свои аргументы.
- СРЗНАЧ(минимум один, максимум 30 аргументов). Находит среднее арифметическое аргументов;
- СЧЁТ(минимум один, максимум 30 аргументов). Подсчитывает количество чисел в списке аргументов (используется для подсчета количества ячеек с числами, пустые ячейки и текст игнорируются);
- МАКС(минимум один, максимум 30 аргументов). Возвращает максимальный аргумент;
- МИН(минимум один, максимум 30 аргументов). Возвращает минимальный аргумент.

Существует 3 способа запуска мастера функций:

- с помощью кнопки в строке формул;
- с помощью команды "Другие функции..." кнопки ;
- с помощью пункта меню "Вставка" "Функция";

После выполнения одного из этих действий откроется окно мастера функций. Мы выбираем нужную нам функцию, пользуясь поиском или фильтром категорий. После выбора нужной функции нажимаем "ОК" и выделяем необходимый диапазон ячеек

В одну функцию можно вставить другую функцию. Конечно, функцию можно записать вручную (писать название вложенной функции, открывать скобки, ставить точки с запятой). Однако это противоречит самой идеологии мастера функций, который должен облегчать написание формул, защищать пользователя от ошибок и свести к минимуму ручную работу. Существует более удобный способ вложить функцию - специальная кнопка на панели "Строка формул".

## Тема-12: Создание графиков функций в Excel. Работа с диаграммами.

### План:

1. Работа с диаграммами в Excel
2. Создание графиков функций в Excel.

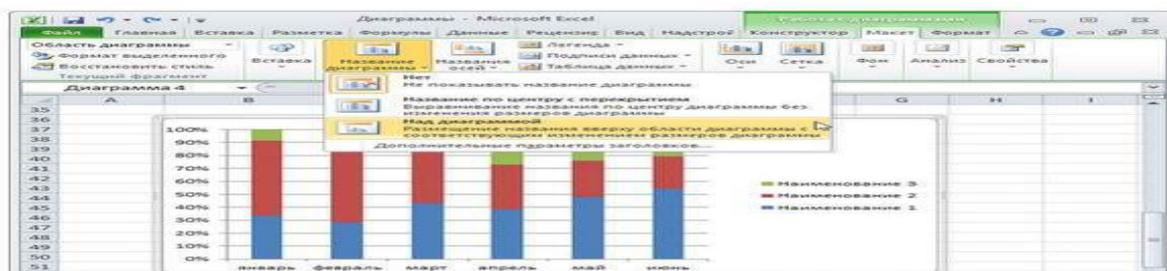
Пользователь может изменить вид диаграммы, выполнив ее настройку. В процессе настройки можно задать заголовок диаграммы, подписи осей, изменить цвет и стиль элементов диаграммы (столбиков диаграммы, линий графика), цвет фона, вид линий сетки и т.д.

Чтобы приступить к настройке диаграммы, ее следует выделить, т.е. сделать щелчок в поле диаграммы (в результате диаграмма будет выделена широкой рамкой) и открыть вкладку Работа с диаграммами -> Макет.

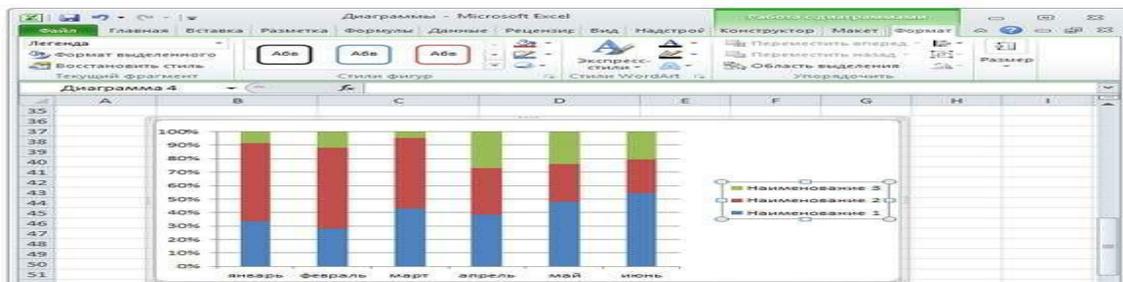


Настройка диаграммы выполняется путем настройки ее элементов (названия диаграммы, названия осей, легенды, подписей рядов данных и др.).

Чтобы настроить элемент диаграммы, его требуется выбрать, т.е. сделать щелчок кнопкой мыши на элементе диаграммы. Следует обратить внимание, что не все элементы диаграммы, например название, отображаются на диаграмме по умолчанию. Поэтому, если нужный элемент не отображается, необходимо раскрыть вкладку Работа с диаграммами -> Макет, сделать щелчок по соответствующей кнопке и в появившемся списке выбрать способ его отображения. Например, чтобы над диаграммой появилось ее название, следует сделать щелчок на кнопке Название диаграммы и в появившемся списке выбрать Над диаграммой.



Можно настроить элементы диаграммы и по-другому: сначала выделите любой элемент диаграммы, появится вкладка Работа с диаграммами, в ней выберите дополнительную вкладку Формат. В зависимости от выбранного элемента диаграммы будет меняться панель настроек.



### Название диаграммы

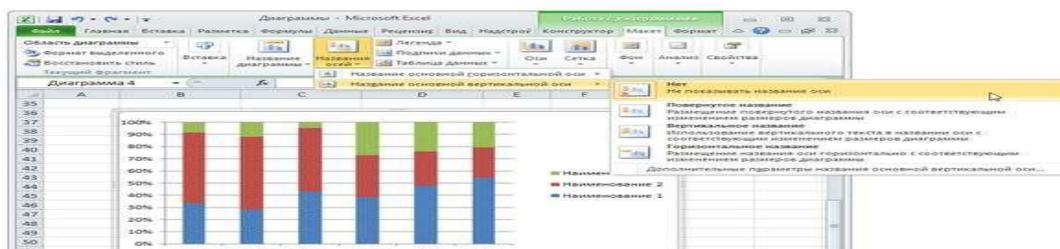
Если на диаграмме больше одного ряда данных, то заголовок не отображается. Чтобы заголовок появился, надо на вкладке Работа с диаграммами -> Макет -> Подписи раскрыть список Название диаграммы и выбрать способ размещения названия диаграммы: Над диаграммой или По центру с перекрытием. Первый способ предполагает автоматическое уменьшение высоты области построения диаграммы (масштабирование по высоте) так, чтобы заголовок не перекрывал диаграмму. При выборе второго способа заголовок будет помещен внутрь области построения диаграммы.



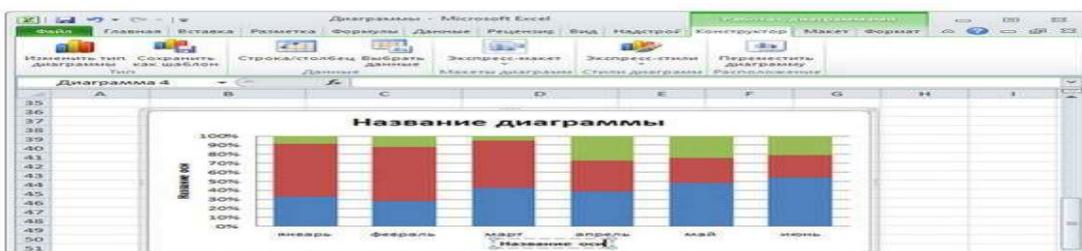
### Названия осей

По умолчанию на диаграмме названия вертикальной и горизонтальной осей не отображаются.

Для того чтобы рядом с вертикальной осью появилось ее название, необходимо на вкладке Работа с диаграммами -> Макет -> Подписи раскрыть список Названия осей -> Название основной вертикальной оси и выбрать способ размещения текста в области отображения названия.



В результате рядом с осью появится соответствующим образом ориентированное поле названия. Чтобы ввести в поле текст (название оси), требуется установить указатель мыши на поле и сделать щелчок левой кнопкой мыши. В поле появится курсор, после чего можно ввести текст - название оси.



Название вертикальной оси задается аналогичным образом.



## Сетка

По умолчанию на диаграмме отображаются только основные горизонтальные линии сетки. Пользователь может сделать видимой сетку (для этого требуется сделать видимыми вертикальные линии сетки) или скрыть ее. Настройка линий сетки выполняется путем выбора на вкладке Работа с диаграммами -> Макет -> Оси в списке Сетка соответствующей команды.



## Легенда

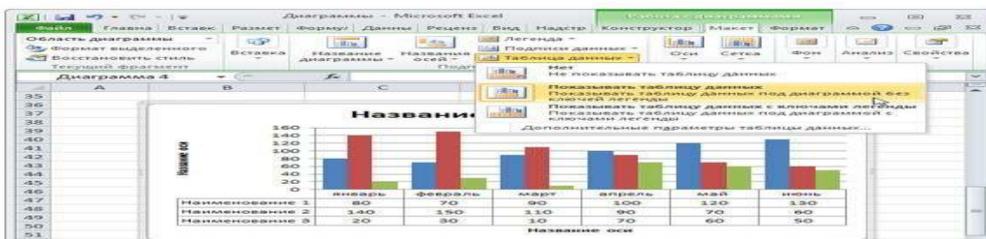
Легенда позволяет идентифицировать данные, которые отображаются на диаграмме. По умолчанию легенда размещается справа от области построения диаграммы. Чтобы изменить положение легенды или скрыть ее, следует на вкладке Работа с диаграммами -> Макет -> Подписи раскрыть список Легенда и выбрать требуемое положение легенды относительно области построения диаграммы.



## Таблица данных

В некоторых случаях бывает удобно, когда рядом с диаграммой находятся данные, которые представлены на диаграмме.

Для того чтобы под диаграммой появилась таблица данных, нужно на вкладке Работа с диаграммами -> Макет -> Подписи раскрыть список Таблица данных и выбрать вид таблицы данных.



Построение графиков функций — одна из возможностей Excel.

Функция, это множество точек  $(x, y)$ , удовлетворяющее выражению  $y=f(x)$ . Поэтому, нам необходимо заполнить массив таких точек, а Excel построит нам на их основе график функции.

1) Рассмотрим пример построения графика линейной функции:  $y=5x-2$

Графиком линейной функции является прямая, которую можно построить по двум точкам. Создадим табличку

	D	E	F
	x	y	

В нашем случае  $y=5x-2$ . В ячейку с первым значением у введем формулу:  $=5*D4-2$ . В другую ячейку формулу можно ввести аналогично (изменив D4 на D5) или использовать маркер автозаполнения.

В итоге мы получим табличку:

1	A	B	C	D	E
2					
3				x	y
4				-5	-27
5				5	23
6					

Теперь можно приступить к созданию графика.

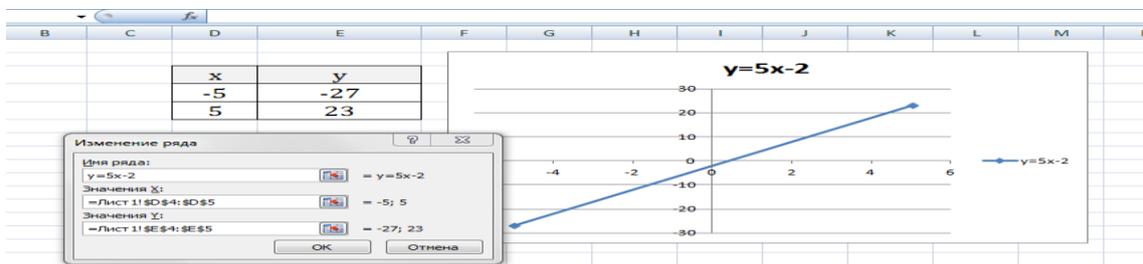
Выбираем: ВСТАВКА —> ТОЧЕЧНАЯ -> ТОЧЕЧНАЯ С ГЛАДКИМИ КРИВЫМИ И МАРКЕРАМИ (рекомендую использовать именно этот тип диаграммы)



Появится пустая область диаграмм. Нажимаем кнопку ВЫБРАТЬ ДАННЫЕ



Выберем данные: диапазон ячеек оси абсцисс (x) и оси ординат (y). В качестве имени ряда можем ввести саму функцию в кавычках « $y=5x-2$ » или что-то другое. Вот что получилось:



Нажимаем ОК. Перед нами график линейной функции.

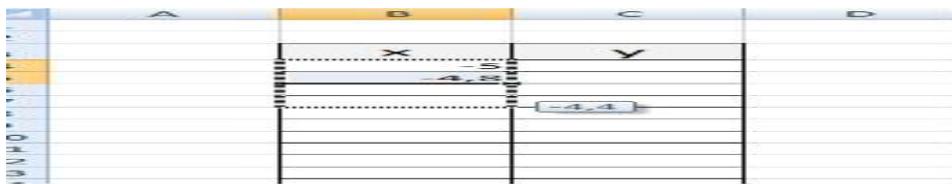
2) Рассмотрим процесс построения графика квадратичной функции — параболы  $y=2x^2-2$

Параболу по двум точкам уже не построить, в отличие от прямой.

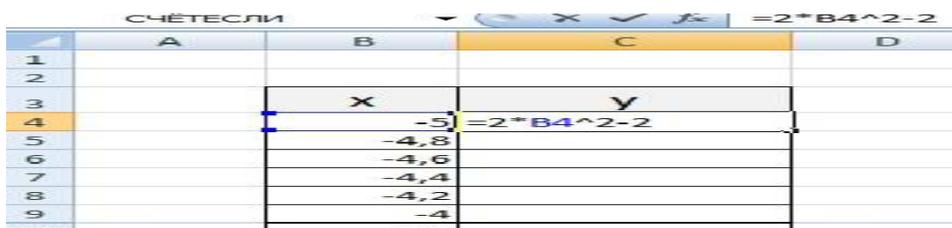
Зададим интервал на оси x, на котором будет строиться наша парабола. Выберу [-5; 5].

Задам шаг. Чем меньше шаг, тем точнее будет построенный график. Выберу 0,2.

Заполняю столбец со значениями x, используя маркер автозаполнения до значения x=5.

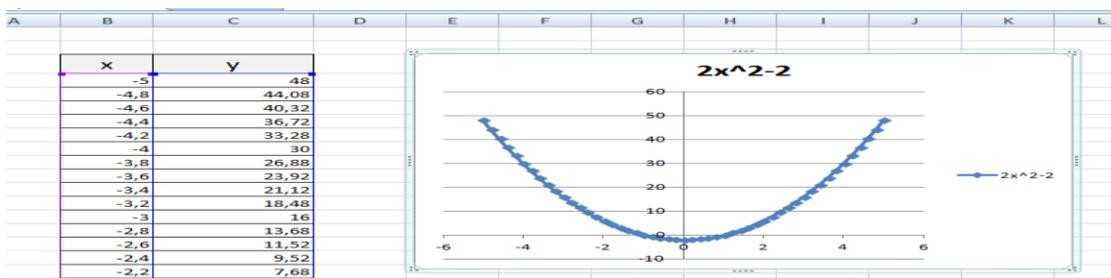


Столбец значений y рассчитывается по формуле:  $=2*B4^2-2$ . Используя маркер автозаполнения, рассчитываем значения y для остальных x.



Выбираем: ВСТАВКА — > ТОЧЕЧНАЯ -> ТОЧЕЧНАЯ С ГЛАДКИМИ КРИВЫМИ И МАРКЕРАМИ и действуем аналогично построению графика линейной функции.

Получим:



Чтобы не было точек на графике, поменяйте тип диаграммы на ТОЧЕЧНАЯ С ГЛАДКИМИ КРИВЫМИ.

Любые другие графики непрерывных функций строятся аналогично.

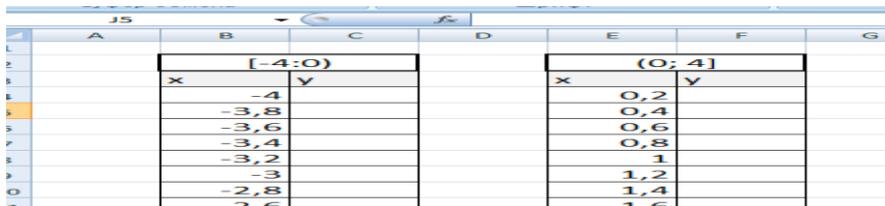
3) Если функция кусочная, то необходимо каждый «кусочек» графика объединить в одной области диаграмм.

Рассмотрим это на примере функции  $y=1/x$ .

Функция определена на интервалах  $(-\infty; 0)$  и  $(0; +\infty)$

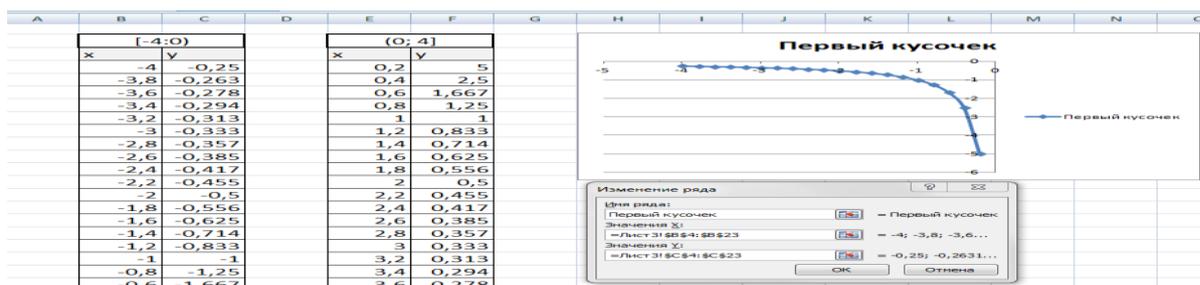
Создадим график функции на интервалах:  $[-4; 0)$  и  $(0; 4]$ .

Подготовим две таблички, где x изменяется с шагом 0,2:

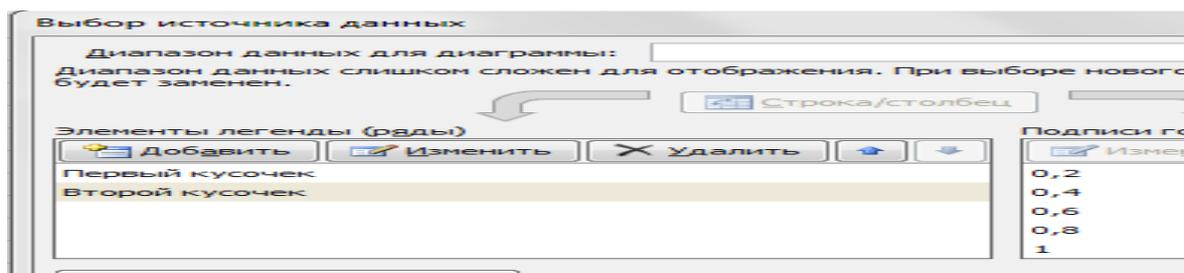


Находим значения функции от каждого аргумента x аналогично примерам выше.

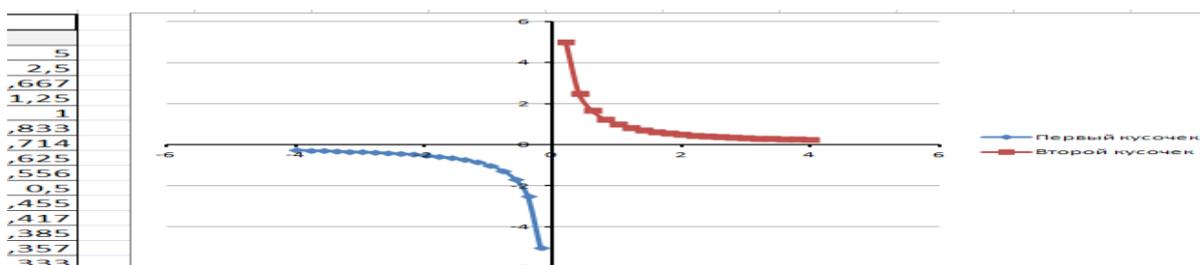
На диаграмму вы должны добавить два ряда — для первой и второй таблички соответственно



Далее нажимаем кнопку ДОБАВИТЬ и заполняем табличку ИЗМЕНЕНИЕ РЯДА значениями из второй таблички



Получаем график функции  $y=1/x$



### ТЕМА-13: Компьютерная графика. Программа AdobePhtotoshop.

План:

1. Компьютерная графика.
2. Программа AdobePhtotoshop.

**Компьютерная графика** — раздел информатики, предметом которого является создание и обработка на компьютере с графических изображений (рисунков, чертежей, фотографий и пр.)

#### *История компьютерной графики*

О компьютерной графике заговорили после опытов Джей У. Форрестера (инженер компьютерной лаборатории Массачусетского технологического института) в 1951 году.

К предшественникам компьютерных рисунков можно отнести первые незатейливые картинки из точек и букв, получаемые на телетайпах телеграфа, а позже — на печатающих устройствах, подключенных к ЭВМ.

Итак, в начале были точки и простые линии. Этот набор стремительно обогащался. 1970-е годы стали временем широкого использования машинной графики. Одно из важнейших отличий современных ПК состоит в возможности вывода на экран графического изображения.

В доступный для многих инструмент компьютерная графика превратилась благодаря Айвену Сазерленду, автору одной из первых графических систем.

### *Направления компьютерной графики*

<b>Направление</b>	<b>Назначение</b>	<b>Программное обеспечение</b>
<b>Научная</b>	Визуализация объектов научных исследований, графическая обработка результатов расчетов, проведение вычислительных экспериментов с наглядным представлением их результатов.	
<b>Деловая</b>	Создание иллюстраций, используемых составления иллюстрации статистических отчетов и пр. Используется в работе учреждений.	Электронные таблицы
<b>Конструкторская</b>	Создание плоских и трехмерных изображений. Используется в работе инженеров-конструкторов.	Системы автоматизированного проектирования (САПР)
<b>Иллюстративная</b>	Создание произвольных рисунков и чертежей.	Графические редакторы
<b>Рекламная</b>	Создание реалистических изображений. Используется для создания рекламных роликов, мультфильмов, компьютерных игр, видеоуроков, видеопрезентаций и пр.	Графические редакторы (со сложным математическим аппаратом)
<b>Компьютерная анимация</b>	Создание движущихся изображений на экране монитора. Слово «анимация» означает «оживление».	

### **Аналоговый и дискретный способы представления ГРАФИЧЕСКИХ ИЗОБРАЖЕНИЙ**

Человек способен воспринимать и хранить информацию в форме образов (*зрительных, звуковых, осязательных, вкусовых, обонятельных*).

Зрительные образы могут быть сохранены в виде изображений (рисунков, фотографий, ...)

При аналоговом представлении физическая величина принимает бесконечное множество значений, причем ее значения изменяются *непрерывно*.

При дискретном представлении физическая величина принимает конечное множество значений, причем ее величина изменяется *скачкообразно*.

**Примером аналогового представления графической информации**



может служить живописное полотно, цвет которого изменяется непрерывно.

**Примером дискретного представления графической информации**



напечатанное с помощью струйного принтера и состоящее из отдельных точек разного цвета.

Все органы чувств человека имеют дело с аналоговыми сигналами.

Любая информация, используемая в технических системах, также начинается и заканчивается аналоговым сигналом.

**Таким образом, представление об аналоговом способе следует рассматривать в качестве необходимой предпосылки перехода к цифровым технологиям.**



**Пространственная дискретизация** – перевод графического изображения из аналоговой формы в цифровую форму путем разбиения изображения на отдельные фрагменты, причем каждому фрагменту присваивается код цвета (*красный, синий, ...*).

**Растровая графика**

Растровое изображение представляет собой мозаику из очень мелких элементов — **пикселей**.

Оно похоже на лист клетчатой бумаги, на котором каждая клеточка (пиксель) закрашена определенным цветом, и в результате такой раскраски формируется изображение.



**Качество кодирования изображения зависит от:**

- **размера точки** - чем меньше её размер, тем больше количество точек в изображении

- **количества цветов (палитры)** - чем большее количество возможных состояний точки, тем качественнее изображение

**Достоинства растровой графики:**

1. Каждому видеопикселю можно придать любой из миллионов цветовых оттенков. Если размеры пикселей приближаются к размерам видеопикселей, то

растровое изображение выглядит не хуже фотографии. Таким образом, *растровая графика эффективно представляет изображения фотографического качества.*

2. Компьютер легко управляет устройствами вывода, которые используют точки для представления отдельных пикселей. *Поэтому растровые изображения могут быть легко распечатаны на принтере.*

### **Недостатки растровой графики:**

1. В файле растрового изображения запоминается информация о цвете каждого видеопикселя в виде комбинации битов. Простые растровые картинки занимают небольшой объем памяти (несколько десятков или сотен килобайтов). Изображения фотографического качества часто требуют нескольких мегабайтов. Таким образом, *для хранения растровых изображений требуется большой объем памяти.*

Самым простым решением проблемы хранения растровых изображений является увеличение емкости запоминающих устройств компьютера. Современные жесткие и оптические диски предоставляют значительные объемы памяти для данных. Обратной стороной этого решения является стоимость, хотя цены на эти запоминающие устройства в последнее время заметно снижаются.

Другой способ решения проблемы заключается в **сжатии графических файлов**, т. е. использовании программ, уменьшающих размеры файлов растровой графики за счет изменения способа организации данных. Существует несколько методов сжатия графических данных.

2. Проблемой растровых файлов является **масштабирование:**

- при существенном увеличении изображения появляется зернистость, ступенчатость

- при большом уменьшении существенно снижается количество точек, поэтому исчезают наиболее мелкие детали, происходит потеря четкости. Для обработки растровых файлов используют редакторы: **MS Paint, Adobe Photoshop**

### **Векторная графика**



**Векторные изображения** являются оптимальным средством хранения высокоточных графических объектов (рисунки, чертежи, схемы...), для которых имеет значение сохранение четких и ясных контуров.

Векторные изображения формируются из объектов (точка, линия, окружность, прямоугольник ...), которые хранятся в памяти компьютера в виде графических примитивов и описывающих их математических формул.

### **Достоинства векторной графики**

1. При кодировании векторного изображения хранится не само изображение объекта, а координаты точек, используя которые программа каждый раз воссоздает изображение заново.

Поэтому **объем памяти векторных изображений очень мал по сравнению с растровой графикой.**

### **RECTANGLE 1, 1, 200, 200, Red, Green**

Несжатое растровое описание квадрата требует примерно в 1333 раза большей памяти, чем векторное.

2. Векторные изображения могут быть легко масштабированы без потери качества.

*Это возможно, так как масштабирование изображений производится с помощью простых математических операций (умножения параметров графических примитивов на коэффициент масштабирования).*

### **Недостатки векторной графики**

1. Векторная графика не предназначена для создания изображений фотографического качества. В векторном формате изображение всегда будет выглядеть, как рисунок.

В последних версиях векторных программ внедряется все больше элементов "живописности" (падающие тени, прозрачности и другие эффекты, ранее свойственные исключительно программам точечной графики).

2. Векторные изображения иногда не выводятся на печать или выглядят на бумаге не так, как хотелось бы.

Это происходит оттого, что векторные изображения описываются тысячами команд.

В процессе печати эти команды передаются принтеру, а он может, не распознав какой-либо примитив, заменить его другим – похожим, понятным принтеру.

Информация о векторном изображении кодируется как обычная буквенно-цифровая и обрабатывается специальными программами: **CoreIDRAW**, **Adobe Illustrator**.

### **Фрактальная графика**

Изображение строится по формуле. В памяти компьютера хранится не изображение, а только формула, с помощью которой можно получить бесконечное количество различных изображений.

**Фракталы**- это геометрические объекты с удивительными свойствами: любая часть фрактала содержит его уменьшенное изображение.

То есть, сколько фрактал не увеличивай, из любой его части на вас будет смотреть его уменьшенная копия.

Фракталы замечательны тем, что многие из них удивительно похожи на то, что мы встречаем в природе. Снежинку, морского конька, ветви деревьев, разряд молнии и горные массивы можно нарисовать, используя фракталы. Поэтому многие современные учёные говорят о том, что природа имеет свойство фрактальности.



Программные средства для работы с фрактальной графикой предназначены для автоматической генерации изображений путем математических расчетов.

**Создание фрактальной художественной композиции состоит не в рисовании или оформлении, а в программировании.** Ее часто используют в развлекательных программах.



Adobe Photoshop – это один из самых мощных и гибких графических редакторов, удовлетворяющих требованиям любого пользователя. Photoshop, самый популярный среди графических редакторов. Adobe Photoshop это профессиональный пакет для полномасштабной обработки изображений, который поможет вам работать более эффективно, познать новые грани и возможности создания высококачественных изображений, а также является одним из самых точных и удобных средств для хранения, печати и передачи изображения. Вы сможете создавать уникальные изображения, используя быстрый и удобный способ доступа к вашим данным, современный Web-дизайн и работа с графикой, позволяет быстро и с профессиональным качеством редактировать фотографии и многое другое, потому данная тема имеет **актуальное значение**.

**Тема-16: Основы работы в Интернет - www (wordwideweb). Браузеры и провайдеры Интернет (Internet Explorer, Firefox Mozilla, Safari). Сайты Интернет (Rambler, Aport, Yahoo, Yandex, Google, Google.uz)**

#### **План:**

### **1. Основные характеристики современных компьютерных сетей**

Среды передачи данных, их характеристики

Протокол IP, принципы маршрутизации пакетов, DHCP

### **2. История создания сети интернет**

### **3. Характеристика интернет и причины "бегства" в него**

Проблема «интернет на рабочем месте»

Что такое интернет провайдер

### **4. Схема использования сети Интернет**

### **5. Браузер**

Ключевые слова: MILNET, ARPANET, провайдер, Сервер, Wi-Fi

### **Основные характеристики современных компьютерных сетей**

Качество работы сети характеризуют следующие свойства: производительность, надежность, совместимость, управляемость, защищенность, расширяемость и масштабируемость.

К основным характеристикам производительности сети относятся:

- время реакции – характеристика, которая определяется как время между возникновением запроса к какому-либо сетевому сервису и получением ответа на него;

- пропускная способность – характеристика, которая отражает объем данных, переданных сетью в единицу времени;

- задержка передачи – интервал между моментом поступления пакета на вход какого-либо сетевого устройства и моментом его появления на выходе этого устройства.

Для оценки надежности сетей используются различные характеристики, в том числе:

- коэффициент готовности, означающий долю времени, в течение которого система может быть использована;

- безопасность, т.е. способность системы защитить данные от несанкционированного доступа;

- отказоустойчивость – способность системы работать в условиях отказа некоторых ее элементов.

Расширяемость означает возможность сравнительно легкого добавления отдельных элементов сети (пользователей, компьютеров, приложений, сервисов), наращивания длины сегментов сети и замены существующей аппаратуры более мощной.

Масштабируемость означает, что сеть позволяет наращивать количество узлов и протяженность связей в очень широких пределах, при этом производительность сети не ухудшается.

Прозрачность – свойство сети скрывать от пользователя детали своего внутреннего устройства, упрощая тем самым его работу в сети.

Управляемость сети подразумевает возможность централизованно контролировать состояние основных элементов сети, выявлять и разрешать проблемы, возникающие при работе сети, выполнять анализ производительности и планировать развитие сети.

Совместимость означает, что сеть способна включать в себя самое разнообразное программное и аппаратное обеспечение.

**Компьютерные сети можно классифицировать по различным признакам.**

По принципам управления:

1. Одноранговые - не имеющие выделенного сервера. В которой функции управления поочередно передаются от одной рабочей станции к другой;

2. Многоранговые - это сеть, в состав которой входят один или несколько выделенных серверов. Остальные компьютеры такой сети (рабочие станции) выступают в роли клиентов.

По способу соединения:

1. "**Прямое соединение**"- два персональных компьютера соединяются отрезком кабеля. Это позволяет одному компьютеров (ведущему) получить доступ к ресурсам другого (ведомого);

2. "**Общая шина**" - подключение компьютеров к одному кабелю;

3. "**Звезда**" - соединение через центральный узел;

4. "**Кольцо**" - последовательное соединение ПК по двум направлениям.

По охвату территории:

1. **Локальная сеть** (сеть, в которой компьютеры расположены на расстоянии до километра и обычно соединены при помощи скоростных линий связи.) - 0,1 - 1,0 км; Узлы ЛВС находятся в пределах одной комнаты, этажа, здания.

2. **Корпоративная сеть** (в пределах находятся в пределах одной организации, фирмы, завода). Количество узлов в КВС может достигать нескольких сотен. При этом в состав корпоративной сети обычно входят не только персональные компьютеры, но и мощные ЭВМ, а также различное технологическое оборудование (роботы, сборочные линии и т.п.).

Корпоративная сеть позволяет облегчить руководство предприятием и управление технологическим процессом, установить четкий контроль за информационными и производственными ресурсами.

3. **Глобальная сеть** (сеть, элементы которой удалены друг от друга на значительное расстояние) - до 1000 км.

В качестве линий связи в глобальных сетях используются как специально проложенные (например, трансатлантический оптоволоконный кабель), так и существующие линии связи (например, телефонные сети). Количество узлов в ГВС может достигать десятков миллионов. В состав глобальной сети входят отдельные локальные и корпоративные сети.

4. **Всемирная сеть** - объединение глобальных сетей (Internet).

## **Среды передачи данных, их характеристики**

### **Коаксиальный кабель**

Коаксиальный кабель был первым типом кабеля, использованным для соединения компьютеров в сеть.

### **"Витая пара"**

Кабель типа "витая пара" (twisted pair), является одним из наиболее распространенных типов кабеля в настоящее время. Он состоит из нескольких пар медных проводов, покрытых пластиковой оболочкой.

### **Оптоволоконный кабель**

Оптоволоконные кабели представляют собой наиболее современную кабельную технологию, обеспечивающую высокую скорость передачи данных на большие расстояния, устойчивую к интерференции и прослушиванию. Оптоволоконный кабель состоит из центрального стеклянного или пластикового проводника, окруженного слоем стеклянного или пластикового покрытия и внешней защитной оболочкой.

## **Протокол IP, принципы маршрутизации пакетов, DNS**

Протокол IP является самым главным во всей иерархии протоколов семейства TCP/IP. Именно он используется для управления рассылкой TCP/IP пакетов по сети Internet. Среди различных функций, возложенных на IP обычно выделяют следующие:

- определение пакета, который является базовым понятием и единицей передачи данных в сети Internet. Многие зарубежные авторы называют такой IP-пакет датаграммой;

- определение адресной схемы, которая используется в сети Internet;
- передача данных между канальным уровнем (уровнем доступа к сети) и транспортным уровнем (другими словами мультиплексирование транспортных датаграмм во фреймы канального уровня);
- маршрутизация пакетов по сети, т.е. передача пакетов от одного шлюза к другому с целью передачи пакета машине-получателю;
- "нарезка" и сборка из фрагментов пакетов транспортного уровня.

Принцип маршрутизации является одним из тех факторов, который обеспечил гибкость сети Internet и ее победу в соревновании с другими сетевыми технологиями. При неустойчивой работе сети пакеты могут пересылаться по различным маршрутам и затем собираться в единое сообщение. При коммутации путь придется каждый раз вычислять заново для каждого пакета, а в этом случае коммутация потребует больше накладных затрат, чем маршрутизация.

**DHCP**(DynamicHostConfigurationProtocol) предлагается достаточно простой и естественный способ решения проблемы для ситуации, когда число физических подключений ограничено, или реально все пользователи не работают в сети одновременно.

## **Протокол TCP**

Протокол управления передачей (TCP - TransmissionControlProtocol) предназначен для использования в качестве надежного протокола общения между хост-компьютерами в коммуникационных компьютерных сетях с коммутацией пакетов, а также в системах, объединяющих такие сети.

## **Система DNS**

Система имен доменов (DNS - DomainNameSystem) это распределенная база данных, которая используется приложениями TCP/IP, для установления соответствия между именами хостов и IP адресами. DNS также используется для маршрутизации электронной почты.

## **История создания сети Интернет**

В 1961 году Defence Advanced Research Agency (DARPA) по заданию министерства обороны США приступило к проекту по созданию экспериментальной сети передачи пакетов. Эта сеть, названная ARPANET, предназначалась первоначально для изучения методов обеспечения надежной связи между компьютерами различных типов. Многие методы передачи данных через модемы были разработаны в ARPANET. Тогда же были разработаны и протоколы передачи данных в сети - TCP/IP. TCP/IP - это множество коммуникационных протоколов, которые определяют, как компьютеры различных типов могут общаться между собой.

Эксперимент с ARPANET был настолько успешен, что многие организации захотели войти в нее, с целью использования для ежедневной передачи данных.

И в 1975 году ARPANET превратилась из экспериментальной сети в рабочую сеть. Ответственность за администрирование сети взяло на себя

Defence Communication Agency (DCA), в настоящее время называемое Defence Information Systems Agency (DISA). Но развитие ARPANET на этом не остановилось; Протоколы TCP/IP продолжали развиваться и совершенствоваться.

В 1983 году вышел первый стандарт для протоколов TCP/IP, вошедший в Military Standards (MIL STD), т.е. в военные стандарты, и все, кто работал в сети, обязаны были перейти к этим новым протоколам. Для облегчения этого перехода DARPA обратилась с предложением к руководителям фирмы Berkley Software Design - внедрить протоколы TCP/IP в Berkeley(BSD) UNIX. С этого и начался союз UNIX и TCP/IP.

Спустя некоторое время TCP/IP был адаптирован в обычный, то есть в общедоступный стандарт, и термин Internet вошел во всеобщее употребление.

В 1983 году из ARPANET выделилась MILNET, которая стала относиться к Defence Data Network (DDN) министерства обороны США. Термин Internet стал использоваться для обозначения единой сети: MILNET плюс ARPANET. И хотя в 1991 году ARPANET прекратила свое существование, сеть Internet существует, ее размеры намного превышают первоначальные, так как она объединила множество сетей во всем мире. Число хостов, подключенных к сети Internet, выросло с 4 компьютеров в 1969 году до 14 миллионов в 1997 году.

Число пользователей ежегодно удваивается. Если в 1980 г. Интернет объединял лишь 25 сетей, то через 15 лет - уже свыше 44 тысяч университетских, государственных и корпоративных сетевых систем 160 стран мира, связанных высокоскоростными частными и общедоступными средствами связи. За один 1995 год к Интернет подключились от 40 до 50 млн. компьютеров и по оценкам число их ежемесячно увеличивается на 10-20 %.

Согласно исследованию, которое Nielsen называет первым в истории анализом глобального распространения Интернета, население всемирной Сети в 20-ти странах приближается до 300 млн. (2000год) В Северной Америке людей с домашним доступом к Интернет наиболее (близко 150 млн.), ее догоняют европейские страны, где в совокупности таких людей - 82 млн. Лидируют Великобритания, Германия и Италия, частица которых в европейском населении Интернета превышает половину. Исследование также показало, что большинство пользователей имеют доступ к Интернету из дома, а не на работе. Исключения составляет Швейцария, где тех и тех поровну.

### **Характеристика Интернет и причины "бегства" в него**

Все причины можно разделить на 10 больших разделов. Рассмотрим их поподробнее:

#### *1. Внешние причины:*

- Травматическая ситуация, стресс, породившие отчуждение от реального мира и людей. У человека возникает потребность отделиться от других, спрятаться от них, никого не видеть, но ему нужны поддержка и общение с людьми и они обращаются к Сети. Так, например, девушка, которая пережила

изнасилование, не может общаться с "живыми" людьми. Интернет дает ей возможность жить дальше.

- Деадаптация к реальной жизни.

- Общение со знакомыми, которые находятся на расстоянии - в других странах, городах, в таком случае Интернет просто быстрое и значительно более дешевое средство связи.

- Нахождение друзей, любимых, знакомых, людей, с которыми можно провести время.

- Престижность, модность Интернета.

- Поиск необходимой информации.

- Социальная изоляция - вынужденная или нарочно созданная. Человек оказывается в изоляции и пытается искать пути выхода из этой изоляции. Так, парень, который уехал из Украины в США, оказался в другой среде - среде людей с другой культурой, взглядами, поведением, один среди чужих. Он стал ночами бывать на украинском чате, удовлетворяя свою потребность в общении со "своими".

"В тюрьму прибыла инспекция и видит - нет замков, двери открыты, нет дежурных, но все преступники сидят и не убегают.

- Как вы сделали так, что они не убегают? - спросили инспектора у начальника тюрьмы. Начальник ответил:

- Подключил их к Интернету". (Анекдот)

## *2. Расстройства психики:*

- Агорафобия - страх открытого пространства. Пример хорошо проиллюстрирован в фильме "Имитатор". Главная героиня, больная агорафобией, сидит дома и общается с людьми лишь через сеть.

- Ксенофобия - страх чужих. Человек боится быть среди незнакомых людей, боится, чтобы с ним никто не заговорил, не сделал что-то. Сеть, для них, способ поговорить с тем, кто не является "своим".

- Дистиморфобия - человек постоянно переживает по поводу собственной внешности. Он не хочет, чтобы его увидели с "таким ужасным носом". Скрывается, таким образом, от чужих глаз.

- Тревожность и прочие социальные страхи - страх общения с другими людьми, страх отрицательной оценки и т.д.

- Комплекс неполноценности - в Интернете никто не знает, кто ты есть. И ты можешь быть другим.

- Депрессии - депрессивные больные, которые устали от реала, ищут смысл жизни в виртуальной реальности.

- Разные акцентуации и патологии, которые толкают человека на удовлетворение своих, противозаконных в реале, но не в Сети, фантазий. Садисты, мазохисты, педофилы и прочие нашли себе место удовлетворения любых потребностей.

### *3. Качества личности:*

- Стеснительность, нерешительность - есть люди, для которых заговорить, познакомиться с другим человеком целая проблема.

- Непривлекательность, внешние недостатки - в Сети это не имеет значения.

- Индивидуализм, эгоизм, желание выделиться и проявить себя.

- Интровертизм.

*4. Удовлетворение потребности* в общении, одобрении, поддержке, безопасности, защищенности - главные потребности человека. Если он не может удовлетворить их в реальной жизни, он удовлетворяет их в виртуальной реальности.

*5. Одиночество, отсутствие друзей, знакомых* - нет значения, вследствие чего человек ощущает себя одиноким, лишь факт наличия одиночества толкает человека искать что-то, что может помочь.

"...Очень много людей отвергли действительность, заменив ее виртуальностью. Так ведь не от хорошей жизни люди идут. Не потому ли человек ночь сидит за монитором, что ему просто некому позвонить и не с кем встретиться. Вы хотя бы раз ощущали одиночество? Если хочется кричать? Если болезненно хочется поговорить. Просто вылить все, что есть на душе. А еще лучше - и найти понимание..." (Из разговора).

*6. Удовлетворение сексуальных желаний, стремлений, склонностей.* Даже, если для получения удовлетворения в реальности не нужно совершать преступления, все же в Сети это легче сделать, легче достичь цели. Без трудностей и без отрицательных последствий - нежелательной беременности (как и желательной, кстати), СПИДа, венерических болезней.

*7. Возможность создания собственного образа, персоны.* Этот образ будет собственным, индивидуальным.

- Возможность создания своего образа, вызывая предусмотренную и желательную реакцию окружающих, получение признания окружающих. Как правило, эти люди имеют комплекс неполноценности или заикленные на своей внешности, имеют низкую самооценку, и т.п. При помощи создания красивого образа, который нравится другим, они получают необходимое одобрение, признание и окружение. "В виртуальной среде вы вообще можете

быть кем хотите, выглядеть как угодно, быть существом любого пола на выбор, словом, у вас нет ограничений, характерных для материального мира". "В Интернете никто не знает, что вы - собака".

Проблема «Интернет на рабочем месте»

Проблема "Интернета на рабочем месте" или, точнее, неограниченного доступа в глобальную Сеть с типичного рабочего места в офисе, без преувеличения входит в список самых болезненных для тех, кто по роду своей деятельности связан с вычислительной техникой. Отношение к ней, сложившееся в обществе в последние годы, однозначное и никак не зависит ни от специальности работников, ни от их географического положения, ни от каких-то других факторов: спросите любую поисковую машину на тему "Интернет влияние продуктивность" - и вы убедитесь, что Интернет на работе считается несомненным злом, самым ужасным образом сказывающимся на производительности работников умственного труда, на их взаимоотношениях с коллегами и начальством, на благополучии самого работодателя. А причина всех бед - злоупотребления, допускаемые пользователями: вместо того, чтобы заниматься выполнением основных обязанностей, они качают из Сети музыку, смотрят фильмы, общаются в чатах и т.д., и т.п. Разве не логично просто закрыть выход в Сеть, решив проблему на корню? Так зачастую и делают - оставляя в качестве канала для связи лишь электронную почту. Но давайте не будем торопиться. Буквально на днях в западной прессе появились данные, говорящие, что не всё так уж однозначно плохо.

Замахнуться на святое позволили себе авторы исследования под названием National Technology Readiness Survey (NTRS). По своей сути, это телефонный опрос, проводимый среди американцев от 18 лет и старше на протяжении вот уже четырёх с лишним лет. Назначение его - изучение отношения населения США к электронным сервисам, выявление и анализ основных способов их использования. Тема это чрезвычайно обширная, но нам из неё интересна только одна категория: влияние Сети на продуктивность офисных работников. В отчёте о последнем исследовании (за 2002 год) приведена любопытная статистика: оказывается, часы, убитые на похождения в Сети для удовлетворения личных нужд и интересов во время рабочего дня, обычно с лихвой компенсируются в нерабочее время.

Исследователи предлагают обратить внимание на работников, имеющих выход в Сеть, как со своего основного рабочего места, так и из дома. Такие пользователи тратят на личные нужды в Сети в рабочее время примерно 3.7 часа каждую неделю. Однако, 47% из них, вернувшись домой, проводят по 5.9 часа в Сети, выполняя свои основные рабочие обязанности. Ещё 25% таких пользователей отдадут своей работе дома примерно то же время, какое они потеряли, блуждая по Сети в офисе. Таким образом, большинство офисных работников, имеющих возможность выйти в Интернет, как из дома, так и со своего основного рабочего места, в целом оказываются более продуктивны, нежели их коллеги, лишённые доступа в Сеть на работе или дома. Последние (могущие попасть в Сеть только из дома) представляют собой наиболее опасную категорию: они активней других тратят рабочее время на web-сёрфинг в личных целях, естественно, никак его не компенсируя.

Выводы исследователей просты. Говорить о полной свободе Интернет-доступа на рабочем месте не следует, но работодателям всё же стоит пересмотреть свои принципы, позволив подчинённым с некоторыми ограничениями пользоваться Сетью в личных интересах во время рабочего дня. Хорошим выходом может стать разработка внутренней политики сетевого доступа, пресекающей особо радикальные выходки вроде просмотра порнографии или скачки музыкальных файлов, но разрешающей общение и web-сёрфинг. Впрочем, на пути к этому светлому будущему ещё множество преград - уж очень глубоко засела в умах идея о вредоносности Сети. Реально ли вообще заставить работодателей решиться на открытие Интернет-доступа с рабочих мест своих подчинённых? Какие ещё отрицательные стороны, помимо избыточного трафика, может скрывать в себе такое решение? Какую именно сетевую активность в рабочее время стоит разрешить, а какую - поставить вне закона? Пока эти вопросы остаются на совести работодателя.



Каждому пользователю для подключения к Интернету требуется компьютер, браузер (программа для просмотра веб-страниц) и провайдер услуг Интернет.

Смею предположить, что компьютер у Вас уже есть (ноутбук, настольный ПК, планшет и т.п.) В состав операционной системы Windows входит браузер Internet Explorer (его еще называют «встроенный браузер»).

### **Что такое Интернет провайдер**

Если компьютер и браузер у пользователя имеются в наличии, остается определиться с Интернет провайдером.

Провайдер – это компания, которая предоставляет доступ к Интернету.

Сокращенно ее можно назвать ISP, что означает Internet Service Provider. Это можно перевести как Интернет-провайдер или лучше по-русски перевести как поставщик услуг Интернет.

Можно дать и такое определение:

Поставщик услуг Интернет (Интернет провайдер) – это компания, которая обладает постоянно действующим каналом связи с Сетью.

Чтобы воспользоваться возможностями Интернет провайдера, необходимо получить доступ к одному из его серверов, через которые организовывается использование канала связи с Сетью.

Пользователей Интернета много. Компаний, предоставляющих доступ к Интернету, намного меньше. Поэтому каждому клиенту предоставляется лишь небольшая часть «общего пирога», то есть, некоторая часть общей пропускной способности канала связи Интернет провайдера. Какая именно часть – это зависит от типа подключения к Сети, а также от тарифного плана – чем больше денег Вы платите, тем более быстрый доступ получаете. Соответственно самый

быстрый доступ – это доступ без ограничений или так называемый безлимитный доступ, но он и самый дорогой.

Идентификация каждого пользователя Интернет провайдера осуществляется с помощью имени пользователя (login - логин) и пароля (password).

Схема использования сети Интернет

Рассмотрим, как взаимодействует Ваш компьютер и сервер Интернет провайдера. Для этого допустим, что Вы организовали устойчивую связь между своим компьютером и сервером Интернет провайдера. Иными словами, Вы уже пользуетесь услугами провайдера. Используя специальное программное обеспечение (как правило, браузер или, его еще называют, web-браузер), Вы посылаете со своего компьютера удаленному серверу провайдера какой-либо запрос на получение информации (например, запрос на загрузку этой веб-страницы, которую Вы сейчас читаете).

Сервер Интернет провайдера обрабатывает Ваш запрос и передает его необходимому серверу сети Интернет (тому серверу, на котором хранится запрашиваемая страница). Именно на таких серверах хранятся все сайты, картинки, музыка, видео и другие данные, которые Вы можете просматривать или зачислять из Глобальной сети.

Сервер, находящийся в Глобальной сети (обозначен цифрой 3 на рис. 1), принимает запрос от сервера вашего поставщика услуг Интернет (цифра 2 на рис. 1) и, если он может предоставить запрашиваемую информацию, то передает ее серверу вашего провайдера. Сервер Интернет провайдера, в свою очередь, переадресовывает полученные результаты запроса вашему компьютеру (цифра 1 на рис. 1).

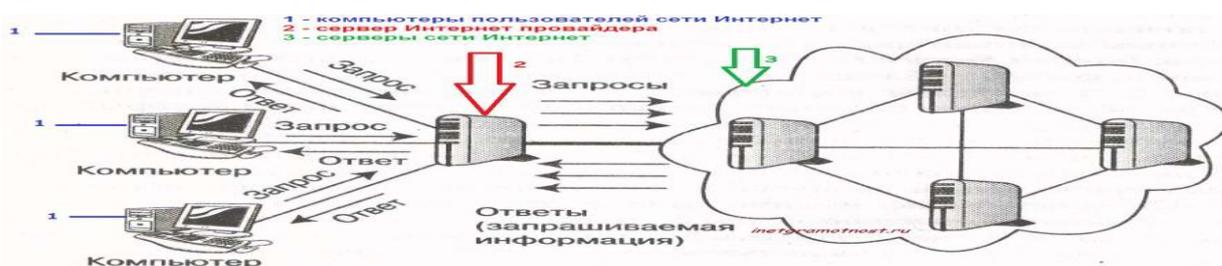


Рис. 1 Схема использования сети Интернет

В результате выполнения такой цепочки действий Вы либо получите у себя на компьютере ответ на Ваш запрос, то есть, нужную Web-страницу, либо увидите уведомление о невозможности (по разным причинам) найти требуемый сервер сети Интернет или информацию на нем.

Как видим, Интернет сложно устроен, но достаточно прост в использовании.

Как найти провайдера

Разные провайдеры обслуживают разные области и географические регионы. Лучше всего узнать у своих знакомых, соседей и друзей, как они подключены к сети и что думают о качестве связи, предоставляемой

конкретным провайдером. Узнайте также о стоимости услуг каждого из провайдеров и выберите того из них, кто больше всего Вам подходит (если будет из кого выбирать).

Также можно найти в Интернете тематические форумы с отзывами пользователей о работе того или иного Интернет провайдера.

Обычно услуги предоставляются на месячной основе. Если Вам что-то не понравится, то через месяц можно перейти к другому провайдеру.

4 нюанса при подключении к Интернету

1) Интернет провайдер предоставляет доступ в Интернет на платной основе. Но далеко не всегда за подключение к Интернет платит пользователь ПК. Иногда можно получить **доступ в Интернет бесплатно**, через Wi-Fi. Современные ноутбуки и планшеты поддерживают подключение к Wi-Fi. Поэтому при нахождении в зоне действия Wi-Fi доступ к Интернету можно получить бесплатно без пароля, либо по паролю, который Вам предоставляют, как клиенту. Такое возможно на некоторых станциях метро, иногда в кафе, в гостиницах на ресепшен, в ВУЗах и т.п., где оплату услуг провайдеру осуществляет организация, предоставившая Вам Wi-Fi подключение.

2) Провайдеры могут предоставлять доступ в Интернет с поминутной тарификацией, по объему скачиваемого трафика (входящего и исходящего), либо могут предоставить безлимитный доступ. Желательно узнать о таких «нюансах», точнее, **о тарифах** прежде, чем Вы выйдете в Интернет. Иначе в конце месяца Вы можете быть неприятно удивлены, увидев представленный счет за пользование услугами доступа в Интернет.

Когда-то у меня был Интернет на месяц с ограничением по объему трафика. Один раз получилось так, что этот предельный объем был немного превышен. За это пришлось внести сумму, превышающую оплату Интернета за месяц в 1,5 раза. С тех пор мне ничего не остается, как быть очень любознательной и внимательной по отношению к тарифам Интернет-провайдеров.

3) Подключиться к Интернету можно разными способами. Самый медленный из них – подключение с помощью телефонного модема и обычной домашней телефонной линии (коммутируемый способ). Лучше пользоваться высокоскоростными видами подключения, например,

- ADSL-модем, значительно увеличивающий скорость обмена данными по телефонной линии,

- подключение по выделенной линии LAN (для этого провайдер должен иметь техническую возможность «дотянуть» эту линию непосредственно до Вашего ПК),

- WLAN-подключение к провайдеру услуг мобильной телефонии с помощью СИМ-карты (для этого Ваш ПК должен обладать такими возможностями, или надо к нему подключить дополнительные устройства)

- и др.

Однако высокоскоростные подключения к Интернету доступны далеко не везде. К сожалению, выбор способа подключения обычно определяется не личными предпочтениями пользователя, а той местностью, в которой Вы живете, и предоставляемыми Вам возможностями.

Правда, все меняется быстро, поэтому через некоторое время после подключения к Интернету можно сильно удивиться, как изменились тарифы и, соответственно, выбрать для себя более подходящий тариф, или сменить провайдера услуг на другого, более дешевого или на более качественного.

4) Помните, что **Вы платите провайдеру за одно подключение к Интернету**. После настройки домашней сети можно использовать одно и то же подключение для разных компьютеров, мобильных телефонов, телевизоров и других устройств, требующих подключения к Интернету.

Если у Вас есть «проводной Интернет», то можно подключенный к Интернету ноутбук настроить так, что он будет раздавать Wi-Fi для других Ваших устройств, например, для планшета, для смартфона.

Провайдер – это компания, которая предоставляет доступ к Интернету.

Сокращенно ее можно назвать ISP, что означает Internet Service Provider. Это можно перевести как Интернет-провайдер или лучше по-русски перевести как поставщик услуг Интернет.

Можно дать и такое определение:

Поставщик услуг Интернет (Интернет провайдер) – это компания, которая обладает постоянно действующим каналом связи с Сетью.

Чтобы воспользоваться возможностями Интернет провайдера, необходимо получить доступ к одному из его серверов, через которые организовывается использование канала связи с Сетью.

Пользователей Интернета много. Компаний, предоставляющих доступ к Интернету, намного меньше. Поэтому каждому клиенту предоставляется лишь небольшая часть «общего пирога», то есть, некоторая часть общей пропускной способности канала связи Интернет провайдера. Какая именно часть – это зависит от типа подключения к Сети, а также от тарифного плана – чем больше денег Вы платите, тем более быстрый доступ получаете. Соответственно самый быстрый доступ – это доступ без ограничений или так называемый безлимитный доступ, но он и самый дорогой.

Идентификация каждого пользователя Интернет провайдера осуществляется с помощью имени пользователя (login — логин) и пароля (password).

Схема использования сети Интернет

Рассмотрим, как взаимодействует Ваш компьютер и сервер Интернет провайдера. Для этого допустим, что Вы организовали устойчивую связь между своим компьютером и сервером Интернет провайдера. Иными словами, Вы уже пользуетесь услугами провайдера. Используя специальное программное

обеспечение (как правило, браузер или, его еще называют, web-браузер), Вы посылаете со своего компьютера удаленному серверу провайдера какой-либо запрос на получение информации (например, запрос на загрузку этой веб-страницы, которую Вы сейчас читаете).

Сервер Интернет провайдера обрабатывает Ваш запрос и передает его необходимому серверу сети Интернет (тому серверу, на котором хранится запрашиваемая страница). Именно на таких серверах хранятся все сайты, картинки, музыка, видео и другие данные, которые Вы можете просматривать или зачислять из Глобальной сети.

Сервер, находящийся в Глобальной сети (обозначен цифрой 3 на рис. 1), принимает запрос от сервера вашего поставщика услуг Интернет (цифра 2 на рис. 1) и, если он может предоставить запрашиваемую информацию, то передает ее серверу вашего провайдера. Сервер Интернет провайдера, в свою очередь, переадресовывает полученные результаты запроса вашему компьютеру (цифра 1 на рис. 1).



## Схема использования сети Интернет

Рис. 1 Схема использования сети Интернет

В результате выполнения такой цепочки действий Вы либо получите у себя на компьютере ответ на Ваш запрос, то есть, нужную Web-страницу, либо увидите уведомление о невозможности (по разным причинам) найти требуемый сервер сети Интернет или информацию на нем.

Как видим, Интернет сложно устроен, но достаточно прост в использовании.

## Как найти провайдера

Разные провайдеры обслуживают разные области и географические регионы. Лучше всего узнать у своих знакомых, соседей и друзей, как они подключены к сети и что думают о качестве связи, предоставляемой конкретным провайдером. Узнайте также о стоимости услуг каждого из провайдеров и выберите того из них, кто больше всего Вам подходит (если будет из кого выбирать).

Также можно найти в Интернете тематические форумы с отзывами пользователей о работе того или иного Интернет провайдера.

Обычно услуги предоставляются на месячной основе. Если Вам что-то не понравится, то через месяц можно перейти к другому провайдеру.

## 5 нюансов при подключении к Интернету

## Первый нюанс

Последовательность этапов подключения и настройки домашней сети с использованием роутера (или без него) состоит из следующих 4-х пунктов:

1) Выберите «своего провайдера», с которым Вы собираетесь работать и который предоставит Вам доступ в Интернет. Практически у всех Интернет-провайдеров есть свои сайты.

На сайте выбранного Вами интернет-провайдера ознакомьтесь:

с требованиями к оборудованию, которое должно быть с Вашей стороны, с возможностями (лимитный или безлимитный, скорость, зона покрытия, другие моменты),

со стоимостью,

с условиями предоставления (бесплатно или за деньги, ежемесячная оплата или же сразу после исчерпания лимита предоставленного интернет-трафика).

После чего можно обратиться к выбранному интернет-провайдеру, уточнить у них какие-то вопросы, если они у Вас есть, и заключить договор с провайдером на подключение к интернету и дальнейшему техническому обслуживанию.

Когда договор с провайдером у Вас подписан, дальше уже все сделает его техподдержка: придет мастер в согласованное с Вами время, который занимается установкой и настройкой Интернета.

Если же Вы хотите самостоятельно получить больше информации о том, как происходит подключение к сети, тогда прошу читать дальше.

2) На сайте провайдера или по телефону горячей линии (либо по телефону их техподдержки), указанному на сайте, ознакомьтесь с правилами Настройки оборудования, устанавливаемого с Вашей стороны.

3) Там же ознакомьтесь с правилами Настройки Ваших компьютеров и гаджетов, чтобы они без проблем имели доступ в Интернет.

Как правило, современные роутеры таковы, что специальные Настройки компьютеров и гаджетов не требуются, но могут быть исключения.

4) Не забывайте регулярно оплачивать услуги провайдера, и все будет хорошо работать. Возможные инциденты будут устраняться силами техподдержки интернет-провайдера.

Например, если в качестве провайдера выбрать Yota, то понадобится модем Yota, который подключается через USB для работы только на одном компьютере. А для выхода в Интернет одновременно нескольких компьютеров и гаджетов еще понадобится приобрести роутер с возможностями раздачи Wi-Fi. Модем и роутер Yota нужно будет настроить, следуя инструкциям, прилагаемым к модему и роутеру.

Со стороны компьютеров и гаджетов Настройки не требуются. Операционная система, которая стоит на Вашем компьютере, как правило, не имеет значения для подключения к Интернету.

## Второй нюанс

Интернет провайдер предоставляет доступ в Интернет на платной основе. Но далеко не всегда за подключение к Интернет платит пользователь ПК.

Иногда можно получить доступ в Интернет бесплатно, через Wi-Fi. Современные ноутбуки и планшеты поддерживают подключение к Wi-Fi. Поэтому при нахождении в зоне действия Wi-Fi доступ к Интернету можно получить бесплатно без пароля, либо по паролю, который Вам предоставляют, как клиенту. Такое возможно на некоторых станциях метро, иногда в кафе, в гостиницах на ресепшен, в ВУЗах и т.п., где оплату услуг провайдеру осуществляет организация, предоставившая Вам Wi-Fi подключение.

#### Третий нюанс

Провайдеры могут предоставлять доступ в Интернет с поминутной тарификацией, по объему скачиваемого трафика (входящего и исходящего), либо могут предоставить безлимитный доступ. Желательно узнать о таких «нюансах», точнее, о тарифах прежде, чем Вы выйдете в Интернет. Иначе в конце месяца Вы можете быть неприятно удивлены, увидев представленный счет за пользование услугами доступа в Интернет.

Когда-то у меня был Интернет на месяц с ограничением по объему трафика. Один раз получилось так, что этот предельный объем был немного превышен. За это пришлось внести сумму, превышающую оплату Интернета за месяц в 1,5 раза. С тех пор мне ничего не остается, как быть очень любознательной и внимательной по отношению к тарифам Интернет-провайдеров.

#### Четвертый нюанс

Подключиться к Интернету можно разными способами. Самый медленный из них – подключение с помощью телефонного модема и обычной домашней телефонной линии (коммутируемый способ). Лучше пользоваться высокоскоростными видами подключения, например,

ADSL-модем, значительно увеличивающий скорость обмена данными по телефонной линии,

подключение по выделенной линии LAN (для этого провайдер должен иметь техническую возможность «дотянуть» эту линию непосредственно до Вашего ПК),

WLAN-подключение к провайдеру услуг мобильной телефонии с помощью СИМ-карты (для этого Ваш ПК должен обладать такими возможностями, или надо к нему подключить дополнительные устройства)

и др.

Однако высокоскоростные подключения к Интернету доступны далеко не везде. К сожалению, выбор способа подключения обычно определяется не личными предпочтениями пользователя, а той местностью, в которой Вы живете, и предоставляемыми Вам возможностями.

Правда, все меняется быстро, поэтому через некоторое время после подключения к Интернету можно сильно удивиться, как изменились тарифы и, соответственно, выбрать для себя более подходящий тариф, или сменить провайдера услуг на другого, более дешевого или на более качественного.

## Пятый нюанс

Помните, что Вы платите провайдеру за одно подключение к Интернету. После настройки домашней сети можно использовать одно и то же подключение для разных компьютеров, мобильных телефонов, телевизоров и других устройств, требующих подключения к Интернету.

Если у Вас есть «проводной Интернет», то можно подключенный к Интернету ноутбук настроить так, что он будет раздавать Wi-Fi для других Ваших устройств, например, для планшета, для смартфона.

Программа для просмотра сайтов (браузер) является стандартной программой ОС Windows. Она называется Internet Explorer. В настоящее время существуют различные версии этой программы. Обновление версий можно осуществить в Интернет.

Браузер Internet Explorer не является единственной программой для работы в Интернет, в настоящее время используются Firefox, Opera, Google Chrome и другие, причем на одном компьютере могут быть одновременно несколько браузеров. Структура всех браузеров одинакова и состоит из следующих элементов:

- Адресная строка – в ней указывается URL-адрес необходимого сайта.
- Меню поиска – отправка запросов одному из поисковиков Интернета.
- Окно – часть экрана для просмотра сайта.
- Навигационные кнопки – для перемещения по страницам.
- Меню – для управления браузером.
- Контекстное меню – дополнительное меню пользователя.
- Статусная строка – информация о работе браузера.

Организация, предоставляющая подключение к своему узлу называется поставщиком услуг Интернета - провайдер. Она оказывает подобную услугу на договорной основе.

Физическое подключение может быть выделенным, коммутируемым или беспроводным. Для выделенного соединения необходимо проложить новую или арендовать готовую физическую линию связи (кабельную, оптоволоконную, радиоканал, спутниковый канал и т. п.). Такое подключение используют организации и предприятия, нуждающиеся в передаче больших объемов данных. От типа линии связи зависит ее пропускная способность (измеряется в единицах бит в секунду).

Для подключения к компьютеру провайдера Интернета надо правильно настроить программу «Удаленный доступ к сети». При настройке программы необходимы данные, которые должен сообщить поставщик услуг:

- имя пользователя (login);
- пароль (password);

· IP-адрес сервера DNS (на всякий случай вводят два адреса – основной и дополнительный, используемый, если основной сервер DNS по каким-то причинам временно не работает).

Этих данных достаточно для подключения к Интернету, хотя при заключении договора с провайдером можно получить и дополнительную информацию, например номера телефонов службы поддержки. Вводить собственный IP-адрес для настройки программы не надо. Сервер поставщика услуг выделит его автоматически на время проведения сеанса работы.

В настоящее время все большее распространение получают провайдеры, использующие сотовую связь для подключения Интернет с помощью специальных беспроводных модемов. В этом случае настройки соединения с интернет минимальны, так как уже заложены в эти устройства. Интернет – всемирная информационная компьютерная сеть, представляющая собой объединение множества региональных компьютерных сетей и компьютеров, обменивающихся друг с другом информацией по каналам общественных телекоммуникаций (выделенным телефонным аналоговым и цифровым линиям, оптическим каналам связи и радиоканалам, в том числе спутниковым линиям связи).

Информация в Интернет хранится на серверах. Серверы имеют свои адреса и управляются специализированными программами. Они позволяют пересылать почту и файлы, производить поиск в базах данных и выполнять другие задачи.

Обмен информацией между серверами сети выполняется по высокоскоростным каналам связи (выделенным телефонным линиям, оптоволоконным и спутниковым каналам связи). Доступ отдельных пользователей к информационным ресурсам Интернет обычно осуществляется через провайдера или корпоративную сеть.

Провайдер - поставщик сетевых услуг – лицо или организация предоставляющие услуги по подключению к компьютерным сетям. В качестве провайдера выступает некоторая организация, имеющая модемный пул для соединения с клиентами и выхода во всемирную сеть.

Основными ячейками глобальной сети являются локальные вычислительные сети. Если некоторая локальная сеть непосредственно подключена к глобальной, то и каждая рабочая станция этой сети может быть подключена к ней.

Существуют также компьютеры, которые непосредственно подключены к глобальной сети. Они называются хост - компьютерами (host - хозяин). Хост – это любой компьютер, являющийся постоянной частью Интернет, т.е. соединенный по Интернет – протоколу с другим хостом, который в свою очередь, соединен с другим, и так далее.

Службы (сервисы) – это виды услуг, которые оказываются серверами сети Internet.

World Wide Web – всемирная паутина – служба поиска и просмотра гипертекстовых документов, включающих в себя графику, звук и видео.

E-mail – электронная почта – служба передачи электронных сообщений.

Telnet - предназначен для управления удаленными компьютерами в терминальном режиме через командную строку.

Usenet, News – телеконференции, группы новостей – разновидность сетевой газеты или доски объявлений.

WAIS – система хранения и поиска документов в тематических базах данных.

FTP – служба передачи файлов.

ICQ – служба для общения в реальном времени с помощью клавиатуры.

Telnet – служба удаленного доступа к компьютерам.

Gopher – служба доступа к информации с помощью иерархических каталогов.

Среди этих служб можно выделить службы, предназначенные для коммуникации, то есть для общения, передачи информации (E-mail, ICQ), а также службы, назначение которых – это хранение информации и обеспечение доступа к этой информации пользователей, лидирующее место по объему хранимой информации занимает служба WWW, поскольку данная служба наиболее удобна для работы пользователей и наиболее прогрессивна в техническом плане. На втором месте находится служба FTP, поскольку какие бы интерфейсы и удобства не разрабатывали для пользователя, информация все равно хранится в файлах, доступ к которым и обеспечивает эта служба.

Гипертекст (hypertext) — информационная структура, позволяющая устанавливать смысловые связи между элементами текста на экране компьютера таким образом, чтобы можно было легко осуществлять переходы от одного элемента к другому. Обычно гипертекст представляется имеющим свойства интерфейса набором текстов, содержащих узлы перехода между ними, которые позволяют избирать читаемые сведения или последовательность чтения. Общеизвестным и ярко выраженным примером гипертекста служат веб-страницы — документы HTML (язык разметки гипертекста), размещённые в Сети.

Гиперссылка (hyperlink) — часть гипертекстового документа, ссылающаяся на другой элемент (команда, текст, заголовок, примечание, изображение) в самом документе, на другой объект (файл, каталог, приложение), расположенный на локальном диске или в компьютерной сети, либо на элементы этого объекта.

Гиперссылка может быть добавлена к любому элементу гипертекстового документа и обычно выделяется графически. В HTML-документах текстовые ссылки по умолчанию выделяются синим цветом, при наведении на них курсором мыши в окне браузера изменяются, например, меняют цвет или выделяются подчеркиванием. При навигации в браузере с помощью клавиатуры текстовые и графические ссылки выделяются прямоугольной пунктирной рамкой. Посещенная ранее ссылка обычно выделяется цветом, отличным от цвета непосещенной ссылки.

Браузер -прикладное программное обеспечение для просмотра веб-страниц; содержания веб-документов, компьютерных файлов и их каталогов; управления веб-приложениями; а также для решения других задач. В глобальной сети браузеры используют для запроса, обработки, манипулирования и отображения содержания веб-сайтов. Многие современные

браузеры также могут использоваться для обмена файлами с серверами ftp, а также для непосредственного просмотра содержания файлов многих графических форматов (gif, jpeg, png, svg), аудио-видео форматов (mp3, mpeg), текстовых форматов (pdf, djvu) и других файлов.

Функциональные возможности браузеров постоянно расширяются и улучшаются благодаря конкуренции между их разработчиками и высоким темпом развития и внедрения информационных технологий. Несмотря на то, что браузеры разных изготовителей базируются на разных технологических решениях, большинство современных браузеров придерживается международных стандартов и рекомендаций W3C в области обработки и отображения данных.

Opera — мощная бесплатная программа предоставляющая множество удобных функций для навигации в Интернете.

Google Chrome - веб-браузер производства Google, сочетающий поддержку новейших веб-приложения и удобный поиск с высокой скоростью и стабильностью.

Safari — бесплатный веб-браузер изначально разработанный компанией Apple Inc. для Mac OS X. С 2007 года существует также, предоставленная на этой странице, версия для Windows.

Новый Internet Explorer претерпел значительные перемены как внутри, так и снаружи. В программе реализован полностью новый интерфейс, добавлена поддержка всех востребованных веб-стандартов, повышена производительность.

С каждым годом объемы Интернета увеличиваются в разы, поэтому вероятность найти необходимую информацию резко возрастает. Все чаще при обращении к Интернет основной проблемой оказывается не отсутствие искомой информации, а возможность ее найти. Как правило, обычный человек в силу разных обстоятельств не может или не хочет тратить на поиск нужного ему ответа больше 15-20 минут. Поэтому особенно актуально правильно и грамотно научиться, казалось бы, простой вещи – где и как искать, чтобы получать желаемые ответы.

Поисковая система — это программное обеспечение, состоящее из базы данных документов, снабженной пользовательским интерфейсом, которое разработано для поиска информации во всемирной паутине. Пользуясь поисковой системой, пользователи составляют свои собственные поисковые запросы с ключевыми словами, а не используют фиксированные системой пути поиска.

Результаты поиска обычно представлены в серии результатов, часто называемых страницами результатов поисковой системы. Информация может быть совокупностью веб-страниц, изображений и других типов файлов. Основная задача поисковой системы заключается в выборе наилучшего возможного подмножества в ответ на конкретный запрос, то есть множества документов, которые наиболее соответствуют тому, что ищет пользователь (обычно в порядке убывания релевантности).

Наиболее распространенные поисковые системы: Yandex, Rambler, Google, Yahoo!, Tut.by, Gogo.by и др.

Идеальный процесс поиска выглядит следующим образом:

1. Делаем общий запрос, вводя в поле поиска ключевые слова, и нажимаем кнопку Поиск.
2. Получив ответ с результатами, выделяем в нем описания наиболее подходящих ссылок.
3. Добавляем к прежнему запросу уточняющие ключевые слова, которые есть в описании нужных ссылок, и повторяем процесс.

Помимо описанного варианта поиска на многих сайтах можно воспользоваться поиском через тематические каталоги, в которых ссылки на сайты разделены по темам.

Если вы все делаете правильно, то каждый запрос приближает вас к нужной информации. Таким образом, с каждым шагом уменьшается несоответствие между нужной информацией и тем, что выдает поисковый сервер.

Большинство пользователей именно так и поступают, однако для того, чтобы быстро и эффективно находить нужную информацию, просто скорректировать запрос бывает недостаточно. Необходимо еще применять операторы языка запросов поискового сервера. Эти операторы помогают находить такую информацию, которую бывает сложно найти с помощью простых запросов. Далее рассмотрены операторы, общие для большинства поисковых серверов.

Один из таких операторов — оператор строгого соответствия; как правило, в современных поисковых системах это кавычки. Сочетание слов, которое вы укажете в кавычках, будет учитываться сервером как единое целое, т.е. вы задаете порядок следования слов друг за другом.

Например, по запросу компьютерные журналы (без кавычек) можно получить в результатах поиска веб-страницы со словами, упоминающимися обособленно, т.е. на одной странице может быть слово «компьютерные», на другой — «журналы» и т.д. Конструкция же «компьютерные журналы» (в кавычках) заставляет поисковую систему отбросить все лишние, страницы и показывать лишь те, на которых эти два слова идут друг за другом.

Еще два оператора — обязательного наличия слова «+» и обязательного отсутствия слова «-». Например, если вы хотите найти сайт журнала «Компьютерра», следует ввести «компьютерные журналы» + Компьютерра. Если же нужно найти все журналы, кроме «Компьютерра», следует ввести «компьютерные журналы» - Компьютерра.

Достаточно часто бывает необходимо, чтобы искомые ключевые слова содержались в пределах одного документа. Для этого следует использовать логический оператор И (AND). Однако во всех поисковых системах этого же результата можно достичь, поставив между словами обычный пробел. К примеру, запросы компьютерные журналы и компьютерные AND журналы, как правило, дадут один и тот же результат. Логический оператор ИЛИ (OR) позволяет найти страницы, содержащие хотя бы одно слово из запроса.

Например, с помощью запроса компьютерные OR журналы можно найти документы, в которых встречается или слово «компьютерные», или слово «журналы».

## **Тема-17: Электронная почта Интернет (E-mail).**

### **План:**

1. Электронная почта в Интернет
2. Адрес электронной почты
3. Протокол Simple Mail Transfer Protocol-SMTP

Технология компьютерного способа пересылки и обработки информации, позволяющая поддерживать оперативную связь между сотрудниками, руководителями, учеными и всеми желающими, получила название электронной почты. Электронная почта (e-mail) - это специальный пакет программ для хранения и пересылки сообщений между пользователями ЭВМ. Посредством электронной почты реализуется служба безбумажных почтовых отношений.

Она является системой сбора, регистрации, обработки и передачи любой информации (текстовых документов, изображений, цифровых данных, звукозаписи и т.д.) по сетям ЭВМ. Выполняет такие функции как редактирование документов перед передачей, их хранение в специальном банке, пересылка корреспонденции, проверка и исправление ошибок, возникающих при передаче, выдача подтверждения о получении корреспонденции адресатом, получение и хранение информации в собственном «почтовом ящике», просмотр полученной корреспонденции.

Я выбрала эту тему, так как я сама пользуюсь данным видом связи и считаю, что интересно было бы узнать об электронной почте больше. С помощью электронной почты я имею возможность регулярно общаться с теми людьми, которые живут очень далеко, и я не могу использовать другие виды связи, т.е. я не могу им позвонить, написать смс, а обычное письмо будет идти туда очень долго. Так я поддерживаю связь со своими друзьями, родными мне людьми и просто знакомыми.

### **Электронная почта в Интернет**

Электронная почта является чрезвычайно важным информационным ресурсом Интернет. Помимо того, что она представляет собой самое массовое средство электронных коммуникаций, через нее можно принять или послать сообщения еще в два десятка международных компьютерных сетей, часть из которых вовсе не имеют on-line сервиса, т.е. прямого подключения к Интернет.

Электронная почта во многом похожа на обычную почтовую службу. Корреспонденция подготавливается пользователем на своем рабочем месте либо программой подготовки почты, либо обычным текстовым редактором. Затем пользователь должен вызвать систему отправки почты (программа

подготовки почты вызывает программу отправки автоматически), которая посылает сообщение на почтовый сервер адресата, где специальная программа занимается сортировкой почты и рассылкой ее по ящикам конечных пользователей. После запуска программы получения почты адресат устанавливает соединение со своим почтовым сервером и организует пересылку всех, полученных на свое имя сообщений. Отметим, что почтовые серверы постоянно подключены к Сети, тогда как компьютеры участников переписки могут устанавливать соединение с ними по мере необходимости. Кроме того, получить и отправить почту можно через разные серверы Интернет. При настройке программы работы с электронной почтой независимо от ее интерфейса необходима следующая информация от провайдера: имя сервера исходящей почты, имя сервера входящей почты, имя пользователя и пароль, а также типы протоколов, используемые при почтовом обмене.

Итак, электронная почта зависит от Интернета и не может без него функционировать, а так же каждый пользователь электронной почты должен иметь свой личный адрес.

### **Адрес электронной почты**

Адрес электронной почты имеет формат: имя пользователя@имя домена, например, Tumanova@mail.ru Часть слева от значка «@» - это имя почтового ящика (E-mail Account Name) на сервере, из которого владелец адреса забирает письма (в данном примере - Tumanova).

Электронная почта построена по принципу клиент-серверной архитектуры. Пользователь общается с клиентской программой, которая в свою очередь связывается с почтовым сервером. Для отправления писем используются протокол SMTP (Simple Mail Transfer Protocol - простой протокол пересылки почты) и, соответственно, SMTP-серверы.

### **Протокол Simple Mail Transfer Protocol-SMTP**

Главной целью протокола Simple Mail Transfer Protocol (SMTP, RFC-821, - 822) является надежная и эффективная доставка электронных почтовых сообщений. SMTP - это довольно независимая подсистема, требующая только надежного канала связи. Средой для SMTP может служить отдельная локальная сеть, система сетей или вся сеть Internet.

## **Тема-18: Основы работы в электронных почтах сайтов Интернет.**

План:

Знакомство с электронной почтой

1. *Почтовый адрес*
2. *Почтовые сообщения*

Почтовый клиент OutlookExpress

## Параметры работы OutlookExpress

Что такое геолокация?

Стандарт геолокации W3C

Знакомство с электронной почтой

История систем электронной почты насчитывает более 15 лет. За это время электронная почта успела занять ведущее место в структуре средств обмена информацией между людьми. В отличие от таких традиционных способов передачи информации, как обычная «бумажная» почта, телефонные переговоры или факс-передача, электронная почта имеет ряд неоспоримых преимуществ.

В последнее время пристальное внимание привлекает все, что имеет отношение к глобальной сети Интернет. На базе сети Интернет функционирует электронная почта *E-mail*.

E-mail – это комплекс приложений, работающих в архитектуре клиент-сервер. *Сервер* это ядро системы обмена информацией. Он осуществляет все основные функции хранения и передачи информации, разграничения полномочий пользователей, предотвращения несанкционированного доступа к данным, защиты сохраняемых и передаваемых сообщений, контроля работоспособности и обеспечения безотказной работы системы. *Клиент* отвечает за подключение к серверу, формирование, отправку и получение почты. В качестве клиентского почтового приложения могут быть использованы Outlook Express, MicrosoftExchange, Microsoft Outlook и т.д. Схема взаимодействия *клиента* и *серверов* приведена на рис. 10.1.

Как видно из схемы в электронной почте E:mail работают два сервера:

*сервер SMTP*, который предназначен для отправки сообщений по протоколу SMTP (Simple Mail Transfer Protocol, RFC821). Сервер работает через 25-й порт;

*сервер POP3*, который предназначен для приема сообщений по протоколу POP3 (Post Office Protocol – Version 3, RFC1939). Сервер работает через 110-й порт.

*Почтовый адрес* в электронной почте E:mail генерируется в формате SMTP и имеет следующий вид: электронный почта сообщение письмо

*xxxxxxx@dddd.ddd*,

где xxxxxxx – идентификатор пользователя. Присваивается пользователю на этапе регистрации;

dddd.ddd – имя почтового сервера.

*Почтовые сообщения* в простейшем случае представляют собой текстовый фрагмент определенной структуры. Сообщение условно можно разделить на две основные части: *заголовок сообщения* и *тело сообщения*. Заголовок включает в себя поля:

- адрес получателя, т.е. кому адресовано сообщение;
  - адрес отправителя;
  - тема сообщения, т.е. краткий идентификатор сообщения.
- Тело сообщения содержит текст сообщения и файлы-вложения.

Почтовый клиент OutlookExpress

Почтовый клиент OutlookExpress является стандартным приложением, предназначенным для работы с электронной почтой. Основными функциями OutlookExpress являются получение входящих сообщений, подготовка и отправка сообщений.

Пользовательский интерфейс OutlookExpress

**Строка заголовка** содержит стандартные элементы окна Windows-приложения (кнопки *Свернуть*, *Восстановить* и *Заккрыть*). В этой строке также указано название приложения (Outlook Express).

**Строка меню** содержит заголовки меню, предоставляющие доступ ко всем функциям, необходимым для работы с Outlook Express. С помощью меню можно создавать учетные записи, формировать новые сообщения, отправлять и получать почту, пересылать сообщения другим пользователям, настраивать интерфейс Outlook Express и т.д.

**Панель инструментов** предназначена для быстрого доступа к некоторым наиболее часто используемым командам Outlook Express.

**Панель «Папки»** позволяет вывести на экран списки сообщений и их содержимое, хранящихся либо в одной из стандартных папок Outlook Express, либо в личных папках пользователя.

**Область просмотра Outlook Express** при работе с электронной почтой или новостями разделена на две части: вверху Вы видите список сообщений электронной почты из текущей папки, а в нижней части окна показывается содержимое отмеченного письма. Содержимое письма можно посмотреть и в отдельном окне, если сделать двойной щелчок по строке с выбранным письмом. Область просмотра может быть разделена или по горизонтали, или по вертикали. Все письма в папках располагаются в соответствии с алфавитным порядком имен отправителей, но их можно отсортировать и иначе – например, по дате поступления письма.

Несколько слов о том, как интерпретируются значки с различными изображениями конверта в области просмотра:

- *открытый конверт* отмечает прочитанное письмо;
- *закрытый конверт плюс жирный шрифт* отмечает непрочитанное письмо;
- *скрепка у письма* указывает на наличие в сообщении файлов-вложений (например, документ в формате Word, графический файл и т.д.). Если выделить такое письмо, и щелкнуть по изображению скрепки в правом углу нижней части окна, то будет показано имя файла. Двойной щелчок по имени вложенного файла позволит просмотреть его содержимое с помощью соответствующей программы.

**В строке состояния** OutlookExpress указывает:

- общее число сообщений в текущей (выделенной) папке;
- число непрочитанных сообщений в текущей папке;
- уведомление о получении нового сообщения или, наоборот, отсутствии новых писем;
- режим работы клиента (автономная работа, локальная сеть, удаленное соединение).

Почтовые папки OutlookExpress

Outlook Express предоставляет пользователю следующие стандартные папки: *Входящие*, *Исходящие*, *Отправленные*, *Удаленные* и *Черновики*.

***Входящие.*** Предназначена для приема и хранения всех поступающих сообщений.

***Исходящие.*** Предназначена для временного хранения отправляемых сообщений. Зачем это нужно? Представьте, что Вы подготавливаете несколько писем друг за другом. Чтобы не соединяться каждый раз с Интернетом для отправки очередного письма, письма временно накапливаются в этой папке. Затем при нажатии на кнопку *Доставить почту* они разом уходят на почтовый сервер и далее – к своим адресатам. Именно такой режим отправки писем устанавливается в Outlook Express по умолчанию.

***Отправленные.*** Предназначена для хранения копий отправленных сообщений.

***Удаленные.*** Предназначена для временного хранения удаленных сообщений. Если вы решите окончательно удалить сообщения из этой папки, сделайте правый щелчок по значку папки и из выпавшего меню выберите пункт *Очистить папку*.

***Черновики.*** Предназначена для хранения рабочих материалов. В эту папку могут быть перемещены как входящие сообщения, так и новые сообщения для дальнейшей обработки.

Outlook Express позволяет создавать новые дополнительные папки пользователя, которые позволят хранить сообщения в систематизированном виде. По мере поступления почты можно помещать сообщения в соответствующие папки для последующей обработки.

Настройка OutlookExpress

Для работы с почтовой программой ее необходимо настроить на конкретный почтовый ящик, т.е. создать учетную запись.

Для создания учетной записи необходимо выполнить команду *Сервис / Учетные записи*. В появившемся диалоговом окне выбрать закладку *Почта* и нажать кнопку *Добавить*.

Далее последовательно в диалоге вы должны ввести:

имя, которое будет отображаться в заголовке в поле «От» для всех отправляемых сообщений;

свой почтовый адрес;

имя сервера POP3;

имя сервера SMTP;

идентификатор пользователя и пароль, которые были присвоены на этапе регистрации на почтовом сервере.

Имя сервера POP3 и сервера SMTP обычно совпадают с именем почтового сервера, на котором вы регистрировались. Например, если вы регистрировались на mail.ru, поля «имя сервера POP3» и «имя сервера SMTP» можно заполнить «mail.ru».

OutlookExpress позволяет создать несколько учетных записей, т.е. один почтовый клиент может читать разные почтовые ящики. Причем, почтовые ящики необязательно должны находиться на одном почтовом сервере.

Соединение с почтовым сервером

Для получения входящей почты и отправки исходящей необходимо установить соединение с почтовым сервером. Как это сделать?

Опять выполняем команду *Сервис / Учетные записи*. В появившемся диалоговом окне выбрать закладку *Почта*, выделить учетную запись и нажать кнопку *Свойства*.

В этом диалоговом окне вы можете изменить учетные характеристики, которые вы вводили при создании учетной записи (закладки *Общие, Серверы*). А для настройки соединения предназначена закладка *Подключение*.

Если вы работаете с почтовым сервером через локальную сеть, необходимо выбрать «Локальная сеть».

Если почтовый сервер находится в сети Интернет, необходимо выбрать ранее созданное удаленное соединение, через которое вы выходите в сеть. При соединении с почтовым сервером для удаленного соединения появляется дополнительное окно, где необходимо вводить ваше имя пользователя и пароль.

Когда соединиться с почтовым сервером? OutlookExpress позволяет устанавливать следующие параметры соединения:

*чтение сообщений*: проверять новые сообщения через определенные промежутки времени, т.е. через установленный промежуток времени программа будет соединяться с сервером и проверять наличие новых сообщений;

*отправка сообщений*: отправлять созданные сообщения немедленно, т.е. при нажатии кнопки *Отправить* после создания нового сообщения будет немедленное соединение с почтовым сервером для его отправки;

чтение и отправка сообщений через кнопку *Доставить* (Рис. 10.5), которая позволяет как одновременное чтение и отправку сообщений (*Доставить все*), так и необходимое (*Получить все, Отправить все*), и выбрать учетную запись через которую будет происходить отправка сообщений.

**Геолокация** — определение географического местоположения интернет-пользователя.

**Геолокация** — неразрушающее обнаружение и исследование подповерхностных объектов грунтовых сред методом радиолокационного зондирования.

GPS (англ. Global Positioning System — система глобального позиционирования, читается Джи Пи Эс) — спутниковая система навигации, обеспечивающая измерение расстояния, времени и определяющая местоположение во всемирной системе координат WGS 84.

Основной принцип использования системы — определение местоположения путём измерения моментов времени приёма синхронизированного сигнала от навигационных спутников антенной потребителя. Для определения трёхмерных координат GPS-приёмнику нужно иметь четыре уравнения: «расстояние равно произведению скорости света на разность моментов приёма сигнала потребителем и момента его синхронного излучения от спутников»:  $|x - a_{\{j\}}| = c(t_{\{j\}} - \tau)$ . Здесь:  $a_{\{j\}}$  — местоположение  $\{j\}$ -го спутника,  $t_{\{j\}}$  — момент времени приёма сигнала от  $\{j\}$ -го спутника по часам потребителя,  $\tau$  — неизвестный момент времени

синхронного излучения сигнала всеми спутниками по часам потребителя,  $c$  — скорость света,  $x$  — неизвестное трёхмерное положение потребителя.

### **Что такое геолокация?**

Геолокация – определение географического положения пользователя или вычислительного устройства с помощью разнообразных механизмов сбора данных. Как правило, для определения местоположения большинство служб геолокации используют адреса сетевой маршрутизации или внутренние устройства GPS. Геолокация является API, зависящим от устройства. Это означает, что браузеры или устройства должны поддерживать геолокацию, чтобы использовать её в веб-приложениях.

### **Стандарт геолокации W3C**

Приложения, которые будут выполнять геолокацию, должны поддерживать стандарт геолокации W3C. Обратите внимание, что код в примере выше определяет местоположение пользователя с помощью свойства `W3C navigator.geolocation`.

Некоторые браузеры для определения местоположения пользователя используют IP-адреса. Однако такой способ может лишь приблизительно определить положение пользователя. Метод W3C является самым простым и широко поддерживаемым, и потому именно ему следует отдавать приоритет перед другими методами геолокации.

## **Тема-19: Вирусы и их разновидности.**

План:

Компьютерные вирусы

Классификация и разновидности компьютерных вирусов

### **1. Проявление наличия вируса в работе на компьютере**



#### **Компьютерные вирусы**

Персональный компьютер играет в жизни современного человека важную роль, поскольку он помогает ему почти во всех областях его деятельности. Современное общество все больше вовлекается в виртуальный мир Интернета. Но с активным развитием глобальных сетей актуальным является вопрос информационной безопасности, так как проникающие их сети вирусы могут нарушить целостность и сохранность вашей информации. Защита компьютера от вирусов – это та задача, решать которую приходится всем пользователям, и особенно тем, кто активно пользуется Интернетом или работает в локальной сети.

**Первая «эпидемия»** компьютерного вируса произошла в 1986 году, когда вирус по имени Brain (англ. «мозг») «заражал» дискеты персональных компьютеров. В настоящее время известно несколько десятков тысяч вирусов, заражающих компьютеры и распространяющихся по компьютерным сетям.

## Что же такое вирус? И чем биологический вирус отличается от компьютерного?

**Вирус** – мельчайшая неклеточная частица, размножающаяся в живых клетках, возбудитель инфекционного заболевания.



**Компьютерный вирус** – специально созданная небольшая программа, способная к саморазмножению, засорению компьютера и выполнению других нежелательных действий.

Что же общего между биологическим и компьютерным вирусами?

- Способность к размножению.
- Вред для здоровья человека и нежелательные действия для компьютера.
- Скрытность, т.к. вирусы имеют инкубационный период

## ИСТОРИЯ КОМПЬЮТЕРНЫХ ВИРУСОВ

Первый прототип вируса появился еще в 1971г.. Программист Боб Томас, пытаясь решить задачу передачи информации с одного компьютера на другой, создал программу Creeper, самопроизвольно «перепрыгивавшую» с одной машины на другую в сети компьютерного центра. Правда эта программа не саморазмножилась, не наносила ущерба.

Первые исследования саморазмножающихся искусственных конструкций проводилась в середине прошлого столетия учеными фон Нейманом и Винером.



Джон фон Нейман



Норберт Винер

## ЧЕМ ОПАСЕН КОМПЬЮТЕРНЫЙ ВИРУС?

После заражения компьютера вирус может активизироваться и начать выполнять вредные действия по уничтожению программ и данных.

Активизация вируса может быть связана с различными **событиями**:

- наступлением определённой даты или дня недели
- запуском программы
- открытием документа

### Признаки заражения

- общее замедление работы
- компьютера и уменьшение
- размера свободной оперативной
- памяти;

- некоторые программы перестают
- работать или появляются различ-
- ные ошибки в программах;
- на экран выводятся посторонние символы и сообщения, появляются различные звуковые и видеоэффекты;
- размер некоторых исполнимых файлов и время их создания изменяются;
- некоторые файлы и диски оказываются испорченными;
- компьютер перестает загружаться с жесткого диска.

## Классификация компьютерных вирусов

### ПРИЗНАКИ КЛАССИФИКАЦИИ



## ФАЙЛОВЫЕ ВИРУСЫ

Внедряются в программы и активизируются при их запуске. После запуска заражённой программой могут заражать другие файлы до момента выключения компьютера или перезагрузки операционной системы.



По способу заражения файловые вирусы разделяются на:

1. Перезаписывающие вирусы. Записывают свое тело вместо кода программы, не изменяя название исполняемого файла, вследствие чего программа перестает запускаться.
2. Вирусы-компаньоны. Создают свою копию на месте заражаемой программы, но не уничтожают оригинальный файл, а переименовывают его или перемещают. При запуске программы вначале выполняется код вируса, а затем управление передается оригинальной программе.
3. Файловые черви создают собственные копии с привлекательными для пользователя названиями в надежде, что он их запустит.
4. Вирусы-звенья не изменяют код программы, а заставляют ОС выполнить свой код, изменяя адрес местоположения на диске зараженной программы, на собственный адрес.

5. Паразитические вирусы изменяют содержимое файла, добавляя в него свой код. При этом зараженная программа сохраняет полную или частичную работоспособность. Код может внедряться в начало, середину или конец программы.

6. Вирусы, поражающие исходный код программы. Вирусы данного типа поражают исходный код программы или ее компоненты (.OBJ, .LIB, .DCU). После компиляции программы оказываются встроенными в неё.

### МАКРОВИРУСЫ

Заражают файлы документов, например текстовых. После загрузки заражённого документа в текстовый редактор макровирус постоянно присутствует в оперативной памяти компьютера и может заражать другие документы. Угроза заражения прекращается только после закрытия текстового редактора.

### СЕТЕВЫЕ ВИРУСЫ

Могут передавать по компьютерным сетям свой программный код и запускать его на компьютерах, подключённых к этой сети. Заражение сетевым вирусом может произойти при работе с электронной почтой или при «путешествиях» по Всемирной паутине.

#### Сетевые вирусы

Сетевые черви – программы, распространяющие свои копии по локальным или глобальным сетям с целью:

- проникновения на удаленные компьютеры;
- запуска своей копии на удаленном компьютере;
- дальнейшего распространения на другие

Троянские программы. «Троянский конь» употребляется в значении: тайный, коварный замысел. Эти программы осуществляют различные несанкционированные пользователем действия:

- сбор информации и ее передача злоумышленникам;
- разрушение информации или злонамерная модификация;
- нарушение работоспособности компьютера;
- использование ресурсов компьютера в неблагоприятных целях.

#### Хакерские утилиты и прочие вредоносные программы.

К данной категории относятся:

- утилиты автоматизации создания вирусов, червей и троянских программ;

- программные библиотеки, разработанные для создания вредоносного ПО;
- хакерские утилиты скрытия кода зараженных файлов от антивирусной проверки;
- программы, сообщающие пользователю заведомо ложную информацию о своих действиях в системе;
- прочие программы, тем или иным способом намеренно наносящие прямой или косвенный ущерб данному или удаленным компьютерам.

## **ДЕСТРУКТИВНЫЕ ВОЗМОЖНОСТИ**

### **По деструктивным особенностям вирусы можно разделить на:**

- **безвредные**, т.е. никак не влияющие на работу компьютера (кроме уменьшения свободной памяти на диске в результате своего распространения);
- **неопасные**, влияние которых ограничивается уменьшением свободной памяти на диске и графическими, звуковыми и пр. эффектами;
- **опасные вирусы**, которые могут привести к серьезным сбоям в работе компьютера;
- **очень опасные**, в алгоритмах работы которых заведомо заложены процедуры, которые могут привести к потере программ, уничтожить данные, стереть необходимую для работы компьютера информацию, записанную в системных областях памяти, и даже, как гласит одна из непроверенных компьютерных легенд, способствовать быстрому износу движущихся частей механизмов - вводить в резонанс и разрушать головки некоторых типов винчестеров.

### **Пути проникновения вирусов**

- Глобальная сеть Internet
- Электронная почта
- Локальная сеть
- Компьютеры «Общего назначения»
- Пиратское программное обеспечение
- Ремонтные службы
- Съёмные накопители

### **Пути проникновения вирусов**

#### **Глобальная сеть Интернет**

Основным источником вирусов на сегодняшний день является глобальная сеть Internet. Возможно заражение через страницы Интернет ввиду наличия на страницах всемирной паутины различного «активного» содержимого: скриптов, ActiveX-компоненты, Java-апплетов. В этом случае используются уязвимости программного обеспечения, установленного на компьютере пользователя, либо уязвимости в ПО владельца сайта, а ничего не подозревающие пользователи зайдя на такой сайт рискуют заразить свой компьютер.

### **Электронная почта**

Сейчас один из основных каналов распространения вирусов. Обычно вирусы в письмах электронной почты маскируются под безобидные вложения: картинки, документы, музыку, ссылки на сайты. В некоторых письмах могут содержаться действительно только ссылки, то есть в самих письмах может и не быть вредоносного кода, но если открыть такую ссылку, то можно попасть на специально созданный веб-сайт, содержащий вирусный код. Многие почтовые вирусы, попав на компьютер пользователя, затем используют адресную книгу из установленных почтовых клиентов типа Outlook для рассылки самого себя дальше.

### **Локальные сети**

Третий путь «быстрого заражения» — локальные сети. Если не принимать необходимых мер защиты, то зараженная рабочая станция при входе в сеть заражает один или несколько служебных файлов на сервере. На следующий день пользователи при входе в сеть запускают зараженные файлы с сервера, и вирус, таким образом, получает доступ на компьютеры пользователей.

### **Персональные компьютеры «общего пользования»**

Опасность представляют также компьютеры, установленные в учебных заведениях. Если один из учащихся принес на своих носителях вирус и заразил какой-либо учебный компьютер, то очередную «заразу» получают и носители всех остальных учащихся, работающих на этом компьютере. То же относится и к домашним компьютерам, если на них работает более одного человека.

### **Пиратское программное обеспечение**

Нелегальные копии программного обеспечения, как это было всегда, являются одной из основных «зон риска». Часто пиратские копии на дисках содержат файлы, зараженные самыми разнообразными типами вирусов.

### **Ремонтные службы**

Достаточно редко, но до сих пор вполне реально заражение компьютера вирусом при его ремонте или профилактическом осмотре. Ремонтники — тоже люди, и некоторым из них свойственно наплевательское отношение к элементарным правилам компьютерной безопасности.

## Съемные накопители

В настоящее время большое количество вирусов распространяется через съёмные накопители, включая цифровые фотоаппараты, цифровые видеокамеры, цифровые плееры (MP3-плееры), сотовые телефоны.

### **Тема-20: Защита компьютера от вирусов. Антивирусные программы.**

#### **План:**

- 1. Методы защиты от компьютерных вирусов**
- 2. Действия при заражении вирусом**
- 3. История антивирусов.**
- 4. Классификация антивирусных программ.**
- 5. Сравнение антивирусных программ.**

#### Методы защиты

- ◆ *Защита локальных сетей*
- ◆ *Использование дистрибутивного ПО*
- ◆ *Резервное копирование информации*
- ◆ *Использование антивирусных программ*
- ◆ *Не запускать непроверенные файлы*

История антивирусов. Существуют конкурирующие заявления по поводу первого разработчика антивирусной программы. Вероятно, первое публичное уничтожение сильного вируса на ПК было сделано в начале 1987 года европейцем Бернтом Фиксом (также Бернд). Фикс нейтрализовал заражение венского вируса. В 1987 году вышла первая польская антивирусная программа `mks_vir`. В этой программе была доступна версия только на польском языке. Осенью 1988 года британец Алан Соломон выпустил антивирусную программу «Dr. Solomon's Anti-Virus Toolkit». К 1990 году в продаже на рынке появилось 19 отдельных антивирусных программ, включая «Norton AntiVirus» and «VirusScan» из McAfee.

Антивирусные программы.

Антивирусное Программное Обеспечение — это термин, означающий компьютерную программу, которая пытается определить, нейтрализовать или уничтожить вредоносные программы. Этот тип Программного Обеспечения носит такое название потому, что самые первые антивирусные программы были созданы специально для борьбы с компьютерными вирусами. Однако, большинство современных антивирусных программ создаются для борьбы с

широким спектром угроз, включая вирусов-червей, фишинг-атаки, руткитов, троянских программ и других вредоносных программ

Для антивирусных программ характерно использование двух методов для борьбы с вредоносными программами:

· проверка (сканирование) файлов на наличие известных вирусов, сопоставляя определения со словарем вирусов определение подозрительной работы любой компьютерной программы, которая может говорить о заражении. Такой анализ может включать сбор данных, сканирование порта и другие методы. Большинство коммерческих антивирусных программ используют обе эти технологии, но делая при этом акцент на методе сопоставления со словарем вирусов.

2. Классификация антивирусных программ. Данные программы можно классифицировать по пяти основным группам: фильтры, детекторы, ревизоры, доктора и вакцинаторы. Антивирусы-фильтры — это резидентные программы, которые оповещают пользователя обо всех попытках какой-либо программы записаться на диск, а уж тем более отформатировать его, а также о других подозрительных действиях. Антивирусы-детекторы рассчитаны на конкретные вирусы и основаны на сравнении последовательности кодов содержащихся в теле вируса с кодами проверяемых программ. Ревизоры — это программы, которые анализируют текущее состояние файлов и системных областей диска и сравнивают его с информацией, сохранённой ранее в одном из файлов данных ревизора.

Методы работы антивирусных программ.

В методе сопоставления со словарем, антивирусная программа проверяет файл и ссылается на словарь известных вирусов, которые определили разработчики этой антивирусной программы. Если часть кода в файле совпадает с каким-либо вирусом в словаре, антивирусная программа предпринимает следующие действия:

1. пытается вылечить файл, удаляя из него вирус
2. заносит файл в карантин (в этом случае файл становится недоступным для других программ, а вирус, находящийся в нем, не может распространяться).
3. удаляет зараженный файл

В составе антивируса идет антивирусный монитор, который также проверяет все файлы что называется «на лету» а также электронную почту.

Итак, теперь можно спокойно выбрать из предложенных вариантов антивирус, который вам лучше подойдет, сравнив их по возможностям выявления вирусов и скорости работы.

**Тема-21: Защита информации.**

## Компьютерные преступления и средства защиты информации

Проблемы информационной безопасности постоянно усугубляются процессами проникновения практически во все сферы деятельности общества технических средств обработки и передачи данных и, прежде всего, компьютерных сетей. Это дает основание поставить задачу компьютерного права, одним из основных аспектов которого являются так называемые компьютерные посягательства.

Объектами посягательств могут быть сами технические средства (компьютеры и периферия) как материальные объекты, программное обеспечение и базы данных, для которых технические средства являются окружением.

На сегодняшний день сформулированы базовые принципы информационной безопасности, которая должна обеспечивать:

- целостность данных - защиту от сбоев, ведущих к потере информации, а также от неавторизованного создания или уничтожения данных.
- конфиденциальность информации и, одновременно, ее доступность для всех авторизованных пользователей.

Следует также отметить, что отдельные сферы деятельности (банковские и финансовые институты, информационные сети, системы государственного управления, оборонные и специальные структуры) требуют специальных мер безопасности данных и предъявляют повышенные требования к надежности функционирования информационных систем, в соответствии с характером и важностью решаемых ими задач.

### **Тема-22: Программы архиваторы и архивация файлов и папок.**

#### **План:**

1. Основные виды программ-архиваторов
2. Сжатие файлов при архивации
3. Показатель степени сжатия файлов
4. Оценка функциональности самых популярных архиваторов
5. WinZip
6. WinRAR
7. WinAce
8. 7-Zip

Архивация - это сжатие, уплотнение, упаковка информации с целью ее более рационального размещения на внешнем носителе (диске или дискете).  
Архиваторы - это программы, реализующие процесс архивации, позволяющие создавать и распаковывать архивы.

Необходимость архивации связана с резервным копированием информации на диски и дискеты с целью сохранения программного обеспечения компьютера и защиты его от порчи и уничтожения (умышленного,

случайного или под действием компьютерного вируса). Чтобы уменьшить потери информации, следует иметь резервные копии всех программ и файлов.

Программы-упаковщики (архиваторы) позволяют за счет специальных методов сжатия информации создавать копии файлов меньшего размера и объединять копии нескольких файлов в один архивный файл. Это даёт возможность на дисках или дискетах разместить больше информации, то есть повысить плотность хранения информации на единицу объёма носителя (дискеты или диска).

Кроме того, архивные файлы широко используются для передачи информации в Интернете и по электронной почте, причем благодаря сжатию информации повышается скорость её передачи. Это особенно важно, если учесть, что быстродействие модема и канала связи (телефонной линии) намного меньше, чем процессора и жесткого диска.

Работа архиваторов основана на том, что они находят в файлах повторяющиеся участки и пробелы, помечают их в архивном файле и затем при распаковке восстанавливают по этим отметкам исходные файлы.

Программы-упаковщики (или архиваторы) позволяют помещать копии файлов в архив и извлекать файлы из архива, просматривать оглавление архива и тестировать его целостность, удалять файлы, находящиеся в архиве, и обновлять их, устанавливать пароль при извлечении файлов из архива и др. Разные программы архивации отличаются форматом архивных файлов, скоростью работы, степенью сжатия, набором услуг (полнотой меню для пользователя), удобством пользования (интерфейсом), наличием помощи, собственным размером.

Ряд архиваторов позволяют создавать многотомные архивы, самоизвлекающиеся архивы, архивы, содержащие каталоги. Наиболее популярны и широко используются следующие архиваторы: ARJ, PKZIP/PKUNZIP, RAR, ACE, LHA, ICE, PAK, PKARC/PKXARC, ZOO, HYPER, AIN.

Наиболее высокоэффективными являются архиваторы RAR, ACE, AIN, ARJ.

### **Основные виды программ-архиваторов**

Различными разработчиками были созданы специальные программы для архивации файлов. Как правило, программы для архивации файлов позволяют помещать копии файлов на диске в сжатом виде в архивный файл, извлекать файлы из архива, просматривать оглавление архива и т.д. Разные программы отличаются форматом архивных файлов, скоростью работы, степенью сжатия файлов при помещении в архив, удобством использования.

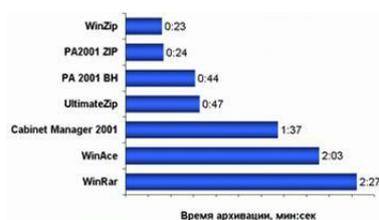
В настоящее время применяется несколько десятков программ - архиваторов, которые отличаются перечнем функций и параметрами работы, однако лучшие из них имеют примерно одинаковые характеристики. Из числа наиболее популярных программ можно выделить: PKPAK, LHA, ICE, HYPER,

ZIP, PAK, ZOO, EXPAND, разработанные за рубежом, а также AIN и RAR. Обычно упаковка и распаковка файлов выполняются одной и той же программой, но в некоторых случаях это осуществляется разными программами, например, программа PKZIP производит упаковку файлов, а PKUNZIP - распаковку файлов.

Программы-архиваторы позволяют создавать и такие архивы, для извлечения из которых содержащихся в них файлов не требуются какие - либо программы, так как сами архивные файлы могут содержать программу распаковки. Такие архивные файлы называются самораспаковывающимися.

Самораспаковывающийся архивный файл - это загрузочный, исполняемый модуль, который способен к самостоятельной разархивации находящихся в нем файлов без использования программы - архиватора.

Самораспаковывающийся архив получил название SFX - архив (Self - eXtracting).



## Сжатие файлов при архивации

Все алгоритмы сжатия оперируют входным потоком информации с целью получения более компактного выходного потока при помощи некоторого преобразования. Основными техническими характеристиками процессов сжатия и результатов их работы являются:

- степень сжатия - отношение объемов исходного и результирующего потоков;

- скорость сжатия - время, затрачиваемое на сжатие некоторого объема информации входного потока, до получения из него эквивалентного выходного потока;

- качество сжатия - величина, показывающая, на сколько сильно упакован выходной поток при применении к нему повторного сжатия по тому же или другому алгоритму.

Алгоритмы, которые устраняют избыточность записи данных, называются алгоритмами сжатия данных, или алгоритмами архивации. В настоящее время существует огромное множество программ для сжатия данных, основанных на нескольких основных способах.

Все алгоритмы сжатия данных делятся на:

- ) алгоритмы сжатия без потерь, при использовании которых данные на приемной восстанавливаются без малейших изменений;

) алгоритмы сжатия с потерями, которые удаляют из потока данных информацию, незначительно влияющую на суть данных, либо вообще невоспринимаемую человеком.

Существует два основных метода архивации без потерь:

алгоритм Хаффмана (англ. Huffman), ориентированный на сжатие последовательностей байт, не связанных между собой,

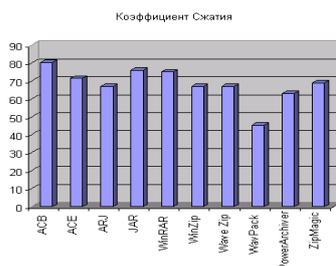
алгоритм Лемпеля-Зива (англ. Lempel, Ziv), ориентированный на сжатие любых видов текстов, то есть использующий факт неоднократного повторения "слов" - последовательностей байт.

Практически все популярные программы архивации без потерь (ARJ, RAR, ZIP и т.п.) используют объединение этих двух методов - алгоритм LZH.

Алгоритм Хаффмана.

Алгоритм основан на том факте, что некоторые символы из стандартного 256-символьного набора в произвольном тексте могут встречаться чаще среднего периода повтора, а другие, соответственно, - реже. Следовательно, если \$+o записи распространенных символов использовать короткие последовательности бит, длиной меньше 8, а для записи редких символов - длинные, то суммарный объем файла уменьшится.

Алгоритм Лемпеля-Зива. Классический алгоритм Лемпеля-Зива -LZ77, названный так по году своего опубликования, предельно прост. Он формулируется следующим образом: если в прошедшем ранее выходном потоке уже встречалась подобная последовательность байт, причем запись о ее длине и смещении от текущей позиции короче чем сама эта последовательность, то в выходной файл записывается ссылка (смещение, длина), а не сама последовательность.



## Показатель степени сжатия файлов

Сжатие информации в архивных файлах производится за счет устранения избыточности различными способами, например за счет упрощения кодов, исключения из них постоянных битов или представления повторяющихся символов или повторяющейся последовательности символов в виде коэффициента повторения и соответствующих символов. Алгоритмы подобного сжатия информации реализованы в специальных программах-архиваторах (наиболее известные из которых arj/arjfolder, pzip/pkzip/winzip, rar/winrar) применяются определенные Сжиматься могут как один, так и

несколько файлов, которые в сжатом виде помещаются в так называемый архивный файл или архив.

## Оценка функциональности самых популярных архиваторов WinZip



- один из старейших (представлен на рынке с 1991 года) архиваторов для Windows, имеющих собственный графический интерфейс. Столь солидный возраст, по всей видимости, и обеспечивает данному архиватору столь внушительную аудиторию пользователей. сжимает файлы, преобразуя их в такие форматы, как: .zip, LHA (.lzh и .lha). Кроме того, WinZip имеет и собственный формат сжатия .zipx, использование которого позволяет добиться максимальной компрессии данных.

WinZip имеет поддержку практически всех известных на сегодня форматов, используемых для сжатия файлов, это: .rar, .7z, .bz2, .cab, .gzip, .tar, .cab и многие другие, кроме того WinZip располагает функционалом, позволяющим преобразовывать сжатые файлы этих форматов в Zip. С помощью WinZip очень удобно сжимать цифровые фото без потери качества изображения, что может понадобиться, к примеру, для отправки большого количества файлов по электронной почте или FTP, а также и для хранения на всевозможных съемных носителях. В винзип имеется возможность просмотра и редактирования файлов содержащихся в архивах других типовых форматов, например таких как: 7z или RAR

1.

### Тема. Основы программирования и языки программирования, типы.

#### Введение в C++. Язык программирования C++

Язык программирования C++ представляет высокоуровневый компилируемый язык программирования общего назначения со статической типизацией, который подходит для создания самых различных приложений. На сегодняшний день C++ является одним из самых популярных и распространенных языков.

Своими корнями он уходит в язык Си, который был разработан в 1969—1973 годах в компании Bell Labs программистом Деннисом Ритчи (Dennis Ritchie). В начале 1980-х годов датский программист Бьерн Страуструп (Bjarne Stroustrup), который в то время работал в компании Bell Labs, разработал C++ как расширение к языку Си. Фактически вначале C++ просто дополнял язык Си некоторыми возможностями объектно-ориентированного программирования. И поэтому сам Страуструп вначале называл его как "C with classes" ("Си с классами").

## Основные этапы развития

В 1979-80 годах Бьерн Страуструп разработал расширение к языку Си - "Си с классами". В 1983 язык был переименован в C++.

В 1985 году была выпущена первая коммерческая версия языка C++, а также первое издание книги "Языка программирования C++", которая представляла первое описание этого языка при отсутствии официального стандарта.

В 1989 была выпущена новая версия языка C++ 2.0, которая включала ряд новых возможностей. После этого язык развивался относительно медленно вплоть до 2011 года. Но при этом в 1998 году была предпринята первая попытка по стандартизации языка организацией ISO (International Organization for Standardization). Первый стандарт получил название ISO/IEC 14882:1998 или сокращенно C++98. В дальнейшем в 2003 была издана новая версия стандарта C++03.

В 2011 году был издан новый стандарт C++11, который содержал множество добавлений и обогащал язык C++ большим числом новых функциональных возможностей. После этого в 2014 году было выпущено небольшое добавление к стандарту, известное также как C++14. И еще один ключевой релиз языка намечен на 2017.

## Компиляторы и среды разработки

Для разработки программ на C++ необходим компилятор - он транслирует исходный код на языке C++ в исполняемый файл, который затем можно запускать. Но в настоящий момент есть очень много различных компиляторов. Они могут отличаться по различным аспектам, в частности, по реализации стандартов. Базовый список компиляторов для C++ можно посмотреть в [википедии](#). Рекомендуется для разработки выбирать те компиляторы, которые развиваются и реализуют все последние стандарты. Так, на протяжении всего руководства преимущественно будет использоваться свободно распространяемый компилятор g++, разработанный в рамках проекта GNU.

Также для создания программ можно использовать интегрированные среды разработки IDE, такие как Visual Studio, Netbeans, Eclipse, Qt и т.д.

## Основы языка Си и элементы C++

### Создание проекта в Microsoft Visual Studio

Для разработки программ в среде Microsoft Visual Studio и Microsoft Visual Studio Express следует создать так называемый *проект* или *решение*. Рассмотрим, как это сделать.

**Шаг 1.** Выберите пункт меню **File->New->Project**.

**Шаг 2.** Выберите тип проекта **Visual C++ Win32 Console Application**(консольное приложение).

**Шаг 3.** Укажите имя проекта и нажмите «**ОК**».

**Шаг 4.** Нажмите кнопку «**Finish**».

**Шаг 5.** В окне «**Solution Explorer**» откройте файл с расширением `cpp`.

### **Первая программа**

Существует старая добрая традиция, согласно которой изучать любой язык программирования начинают с написания программы «Hello, World!» или «Здравствуй, Мир!». Эта программа выводит на экран приветствие миру и представляет собой наиболее простой и быстрый способ заставить компьютер выполнить нашу команду. Для написания этой программы наберите в открытом редакторе следующий программный код:

```
//Первая программа на Си
/* Данная программа выводит
на экран приветствие */
#include "stdafx.h"
#include
using namespace std;
int main() {
cout << "Hello, World!" << endl;
getchar();
return 0;
}
```

### **Вариации функции main()**

Существует несколько разновидностей функции `main()`. Первый вариант – это то, что предлагает нам Visual Studio по умолчанию. В скобках указываются параметры, которые передаются в программу, если соответствующий ей исполняемый `exe`-файл вызывается через командную строку. Аргумент `argc` – это число параметров, а массив `argv[]` – это список параметров.

```
int _tmain(int argc, _TCHAR* argv[]) {
...
return 0;
}
```

Стандартный вид функции main() следующий:

```
int main(int argc, _TCHAR* argv[]) {
...
return 0;
}
```

Для многих программ передаваемые параметры не требуются, поэтому их можно опустить:

```
int main() {
...
return 0;
}
```

Некоторые компиляторы, в частности, тот, который задан по умолчанию в Microsoft Visual Studio, поддерживают функцию main() без указания кода возврата:

```
void main() {
...
}
```

Данный способ удобен, поскольку здесь нет ничего лишнего, однако некоторые компиляторы, например, тот, который по умолчанию встроен в Code::Blocks, не поддерживают такой вид функции main().

### Переменные и типы данных

Строго говоря, *переменная* – это именованный участок памяти, размер которого определяется типом этой переменной. Узнать размер типа можно при помощи функции **sizeof()**. Ниже будет рассмотрен пример ее использования, а сейчас перечислим основные типы данных языка Си.

Тип	Описание	Размер типа (байт)
-----	----------	--------------------

		Диапазон значений
char	Символ или целое число со знаком	1
		-128..127
unsigned char	Символ или целое число без знака	1
		0..255
short	Целое число со знаком	2
		-32768..32767
unsigned short	Целое число без знака	2
		0..65535
int	Целое число со знаком	4
		-2147483647..2147483647
unsigned int	Целое число без знака	4
		0..4294967295
float	Вещественное Число	4
		6 знаков после запятой
double	Вещественное Число	8
		10 знаков после запятой
bool	Логический Тип	1
		true и false (0 и 1)
void	Пустой тип	-
		-

Перед работой с переменной ее необходимо **объявить; объявление переменной** – это указание ее имени и типа. Переменные могут быть объявлены по одной или сразу несколько, при этом указываются тип переменной и ее имя через пробел, а завершается объявление точкой с запятой. Если сразу объявляется несколько переменных, то они перечисляются через запятую:

```
int number, age, weight;
```

```
float length, width;
```

```
bool isEven;
```

```
char symbol;
```

**Присвоение значения** объявленной переменной – это занесение значения в выделенную под переменную ячейку памяти. Осуществляется присвоение при помощи оператора =. Указывается имя переменной и ее новое значение, а между ними ставится оператор присваивания:

```
number = 10;
```

```
length = 2.5;
```

```
isEven = true;
```

```
symbol = 'A';
```

Присвоить значение переменной также можно при ее объявлении, в этом случае говорят, что происходит **инициализация** переменной или переменная **инициализируется**:

```
int number = 10, x=7;
```

```
float length = 2.5;
```

```
bool isEven = true;
```

```
char symbol = '*', letter = 'A';
```

Имя переменной нужно выбирать так, чтобы оно отражало назначение этой переменной. Конечно, можно добавить комментарий:

```
//Возраст студента
```

```
int x;
```

однако предпочтительнее использовать следующее имя:

```
int student_age;
```

В таком случае комментарий вообще не понадобится.

Язык Си, как и большинство современных языков программирования, допускает объявление переменной в любом месте программы. В отличие, например, от языка Паскаль, в котором все переменные объявляются в специальном блоке под названием Var. Более того, существует рекомендация объявлять переменную максимально близко к тому месту, где она используется. Например, если переменная используется только внутри некоторой функции, то

объявлять ее нужно внутри функции, а если переменная используется только внутри тела цикла, то объявлять ее нужно внутри тела цикла.

## Ключевые слова и идентификаторы

Все слова, не считая знаков препинания, операций и чисел, в языке Си можно разделить на две группы: **ключевые слова и идентификаторы**. **Ключевые слова** – это зарезервированные компилятором слова для работы; к ним относятся названия типов, условные операторы, циклы и ряд других конструкций языка, которые подробно рассматриваются далее в этой следующих темах. Как правило, современные среды разработки подсвечивают ключевые слова специальным цветом. В языке Си существует 32 ключевых слова:

auto	break	case	char
const	continue	default	do
double	else	enum	extern
float	for	goto	if
Int	long	register	return
short	signed	sizeof	static
struct	switch	typedef	union
unsigned	void	volatile	while

**Идентификаторы** – это названия переменных, функций и других объектов, создаваемых программистом или уже созданных в заголовочных файлах. Следует заметить, что активно используемые функции **printf()**, **scanf()**, а также название функции **main()** являются идентификаторами, а не ключевыми словами.

При выборе идентификаторов необходимо придерживаться трех правил:

- в идентификаторах можно использовать буквы нижнего и верхнего регистра, цифры и знак подчеркивания «\_»;
- первым символом в идентификаторе может быть буква или знак подчеркивания;
- язык Си чувствителен к регистру букв, например, `int` и `Int` – это два разных слова.

Вот примеры корректных и некорректных идентификаторов:

Корректные	Некорректные
<code>_123</code>	<code>1number</code>

my_variable	int
BIG_LETTERS	t^
FLoat	float
a1b2	_\$var

## Основные арифметические операции

Для работы с переменными используются операции, наиболее часто – арифметические.

Операция	Описание	Пример
+	Сложение	$z = x + y$
-	Вычитание	$z = x - y$
-	Изменение знака	$z = -x$
*	Умножение	$z = x * y$
/	Деление нацело (если операнды - целочисленные)	$z = x / y$
/	Деление (если хотя бы один операнд вещественный)	$z = x / y$
%	Остаток от деления (применим только для целочисленных операндов)	$z = x \% y$

## Вывод информации на экран(оператор вывода в C++)

Для вывода информации на экран на первых порах мы будем использовать C++-ый оператор `cout`. В дальнейшем мы познакомимся ещё с C-шными функциями вывода, используемыми для форматного вывода. Как правило, этот оператор используется в следующей форме:

```
cout << <выводимое выражение> << endl;
```

Например,

```
cout << "Hello, World!" << endl;
```

```
cout << 14 << endl;
```

```
float my_variable = 10;
```

```
cout << my_variable << endl;

int x=5, y=10;

cout << (x+y) << endl; //Выведет 15
```

Обратите внимание на то, что для вывода значения переменной нужно указать имя этой переменной. Таким образом, команда

```
float my_variable = 10;

cout << my_variable << endl;

выведет число 10, а команда

cout << "my_variable" << endl;

выведет строку my_variable.
```

Одной командой можно вывести несколько выражений:

```
cout << "my_variable=" << my_variable << endl;
```

Эта команда выведет **my\_variable=10**. Для перевода курсора на новую строку используется endl.

Итак, оператор cout может использоваться для вывода на экран переменных и выражений любых типов.

### **Ввод информации с клавиатуры(оператор ввода в C++)**

Для ввода информации с клавиатуры используйте оператор cin. Необходимо только указать имя переменной, куда будет сохранено введённое значение:

```
//Ввод символа

char symbol;

cin >> symbol;

//Ввод целого числа

int number;

cin >> number;

//Ввод строки

char str[100];
```

```
cin >> str;
```

```
//Ввод вещественного числа
```

```
float x;
```

```
cin >> x;
```

После ввода указанные переменные будут хранить значения, полученные с клавиатуры. Обратите внимание на то, что при использовании **cout** ставится значок <<, а при использовании **cin** – >>.

### Работа с символами и строками

При работе с символами и строками существуют некоторые тонкости, поэтому рассмотрим некоторые из них. Более детально различные аспекты работы со строками рассматриваются в теме «Строки».

Несколько символов, идущих подряд, образуют строку. **Строка** – это массив символов. Для объявления строки необходимо указать её имя и максимальный размер:

```
char my_string[100];
```

```
char name[50];
```

Задать значение строки можно при её объявлении:

```
char str[] = "Математика – царица наук.";
```

Вывести строку на экран можно следующими способами:

```
cout << str << endl;
```

```
cout << "Заседание продолжается!" << endl;
```

```
cout << 'H'<<'e'<<'l'<<'l'<<'o'<<'!' <<endl;< p=""></endl;<>
```

Обратите внимание на то, что строки заключаются в двойные кавычки, а символы – в апострофы. Это нужно учитывать при работе с ними. Ниже приведены примеры.

### //Корректные записи

```
//Пример 1.
```

```
char ch = 'A';
```

```
//Пример 2.
```

```
char str[] = "Кашу маслом не испортишь";
```

//Пример 3.

```
cout << "Назвался груздем — полезай в кузов";
```

//Пример 4.

```
cout << "С"; cout << "ы"; cout << "р" << endl;
```

//Пример 5.

```
cout << 'С'; cout << 'ы'; cout << 'р' << endl;
```

### **//Некорректные записи**

//Пример 1. (Символ заключён в кавычки)

```
char ch = "А";
```

//Пример 2. (Запись строки в char)

```
char ch = "Россия";
```

//Пример 3. (Запись строки в char)

```
char ch = 'Россия';
```

//Пример 4 (Строка заключена в апострофы).

```
char str[] = 'Кашу маслом не испортишь';
```

//Пример 5 (Строка заключена в апострофы).

```
cout << 'Кашу маслом не испортишь';
```

## **Первая программа на C++**

### **Код первой программы**

Наберите следующий код:

```
#include <iostream>
```

```
#include <cstdlib> // для system
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    cout << "Hello, world!" << endl;
```

```
system("pause"); // Только для тех, у кого MS Visual Studio
return 0;
}
```

## Описание синтаксиса(основные функции компилятора C++)

Директива **#include** используется для подключения других файлов в код. Строка **#include <iostream>**, будет заменена содержимым файла «iostream.h», который находится в стандартной библиотеке языка и отвечает за ввод и вывод данных на экран.

**#include <cstdlib>** подключает стандартную библиотеку языка C. Это подключение необходимо для работы функции **system**.

Содержимое третьей строки — **using namespace std;** указывает на то, что мы используем по умолчанию пространство имен с названием «std». Все то, что находится внутри фигурных скобок функции **int main() {}** будет автоматически выполняться после запуска программы.

Строка **cout << "Hello, world!" << endl;** говорит программе выводить сообщение с текстом «**Hello, world**» на экран.

Оператор **cout** предназначен для вывода текста на экран командной строки. После него ставятся две угловые кавычки (<<). Далее идет текст, который должен выводиться. Он помещается в двойные кавычки. Оператор **endl** переводит строку на уровень ниже.

Если в процессе выполнения произойдет какой-либо сбой, то будет сгенерирован код ошибки, отличный от нуля. Если же работа программы завершилась без сбоев, то код ошибки будет равен нулю. Команда **return 0** необходима для того, чтобы передать операционной системе сообщение об удачном завершении программы.

— В конце каждой команды ставится **точка с запятой**.

## Компиляция и запуск

Теперь скомпилируйте и запустите программу. Тем, кто пользуется MS Visual Studio, нужно нажать сочетание клавиш «Ctrl+F5». Пользователям GCC нужно выполнить следующие команды:

```
с++ имя_файла.cpp -o имя_выходного_бинарника # компиляция кода
./имя_выходного_бинарника # запуск программы
```

Если программа собралась с первого раза, то хорошо. Если компилятор говорит о наличии ошибок, значит вы что-то сделали неправильно.

Прочитайте текст ошибки и попробуйте ее исправить своими силами. Если не получится, напишите о вашей проблеме в комментариях.

В качестве домашнего задания, переделайте эту программу так, чтобы вместо сообщения «Hello, World» выводилось сообщение «Hello, User».

## Переменные и типы данных в C++ →.

### Переменные и типы данных в C++

**Переменная** — это «ячейка» оперативной памяти компьютера, в которой может храниться какая-либо информация.

В программировании переменная, как и в математике, может иметь название, состоящее из одной латинской буквы, но также может состоять из нескольких символов, целого слова или нескольких слов.

- [Типы данных](#)
- [Основные типы данных в C++](#)
- [Объявление переменной](#)
- [Пример объявления переменных](#)
- [Простой калькулятор на C++](#)
- [Разбор кода](#)
- [Домашнее задание](#)

### Типы данных

В языке C++ *все переменные* имеют определенный тип данных. Например, переменная, имеющая целочисленный тип не может содержать ничего кроме целых чисел, а переменная с плавающей точкой — только дробные числа.

**Тип данных** присваивается переменной при ее объявлении или инициализации. Ниже приведены основные типы данных языка C++, которые нам понадобятся.

### Основные типы данных в C++

- **int** — целочисленный тип данных.
- **float** — тип данных с плавающей запятой.
- **double** — тип данных с плавающей запятой двойной точности.
- **char** — символьный тип данных.
- **bool** — логический тип данных.

### Объявление переменной

Объявление переменной в C++ происходит таким образом: сначала указывается тип данных для этой переменной а затем название этой переменной.

### Пример объявления переменных

```
int a; // объявление переменной a целого типа.
```

```
float b; // объявление переменной b типа данных с плавающей запятой.
```

```
double c = 14.2; // инициализация переменной типа double.
```

```
char d = 's'; // инициализация переменной типа char.
```

```
bool k = true; // инициализация логической переменной k.
```

- Заметьте, что в C++ **оператор присваивания (=)** — не является знаком равенства и не может использоваться для сравнения значений. Оператор равенства записывается как «двойное равно» — **==**.
- Присваивание используется для сохранения определенного значения в переменной. Например, запись вида **a = 10** задает переменной **a** значение числа 10.

## Простой калькулятор на C++

Сейчас мы напишем простую программу-калькулятор, которая будет принимать от пользователя два целых числа, а затем определять их сумму:

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    setlocale(0, "");
```

```
    /*7*/ int a, b; // объявление двух переменных a и b целого типа данных.
```

```
    cout << "Введите первое число: ";
```

```
    cin >> a; // пользователь присваивает переменной a какое-либо значение.
```

```
    cout << "Введите второе число: ";
```

```
    cin >> b;
```

```
    /*12*/ int c = a + b; // новой переменной c присваиваем значение суммы введенных пользователем данных.
```

```
    cout << "Сумма чисел = " << c << endl; // вывод ответа.
```

```
    return 0;
```

```
}
```

## Разбор кода

В 7-й строке кода программы мы объявляем переменные «a» и «b» целого типа **int**. В следующей строке кода выводится сообщение пользователю, чтобы он ввел с клавиатуры первое число.

В 9-й строке стоит еще незнакомая вам конструкция — **cin >>**. С помощью нее у пользователя запрашивается ввод значения переменной «a» с клавиатуры. Аналогичным образом задается значение переменной «b».

В 12-й строке мы производим инициализацию переменной «c» суммой переменных «a» и «b». Далее находится уже знакомый вам оператор **cout**, который выводит на экран строку и значение переменной «c».

- При выводе переменных, они не заключаются в кавычки, в отличие от строк.

## Домашнее задание

Попробуйте провести несколько экспериментов с программой — сделайте аналогичный пример с умножением или вычитанием переменных. Не бойтесь издеваться над программным кодом, потому что ошибки — неотъемлемая часть обучения любому делу. И не забываем про точки с запятой.

## Конструкция ветвления в C++ →. Конструкция ветвления в C++ (операторы переходов в C++)

Встречаются ситуации, когда программе нужно выбрать, какую операцию ей выполнить, в зависимости от определенного условия.

К примеру, мы вводим с клавиатуры целое число. Если это число больше десяти, то программа должна выполнить одно действие, иначе — другое. Реализуем этот алгоритм на C++ с помощью **конструкции ветвления**.

- [Оператор if](#)
- [Пример конструкции ветвления](#)

```
#include <iostream>
using namespace std;
int main()
{
    setlocale(0, "");
    double num;
    cout << "Введите произвольное число: ";
    cin >> num;
    if (num < 10) { // Если введенное число меньше 10.
        cout << "Это число меньше 10." << endl;
    } else { // иначе
        cout << "Это число больше либо равно 10." << endl;
    }
    return 0;
}
```

Если вы запустите эту программу, то при вводе числа, меньшего десяти, будет выводиться соответствующее сообщение.

Если введенное число окажется большим, либо равным десяти — отобразится другое сообщение.

## Оператор if

**Оператор if** служит для того, чтобы выполнить какую-либо операцию в том случае, когда условие является верным. *Условная конструкция в C++* всегда записывается в круглых скобках после оператора **if**.

Внутри фигурных скобок указывается тело условия. Если условие выполнится, то начнется выполнение всех команд, которые находятся между фигурными скобками.

### Пример конструкции ветвления

```
if (num < 10) { // Если введенное число меньше 10.  
    cout << "Это число меньше 10." << endl;  
} else { // иначе  
    cout << "Это число больше либо равно 10." << endl;  
}
```

Здесь говорится: «Если переменная `num` меньше 10 — вывести соответствующее сообщение. Иначе, вывести другое сообщение».

Усовершенствуем программу так, чтобы она выводила сообщение, о том, что переменная `num` равна десяти:

```
if (num < 10) { // Если введенное число меньше 10.  
    cout << "Это число меньше 10." << endl;  
} else if (num == 10) {  
    cout << "Это число равно 10." << endl;  
} else { // иначе  
    cout << "Это число больше 10." << endl;  
}
```

Здесь мы проверяем три условия:

- Первое — когда введенное число меньше 10-ти
- Второе — когда число равно 10-ти
- И третье — когда число больше десяти

Заметьте, что во втором условии, при проверке равенства, мы используем оператор равенства `==`, а не оператор присваивания, потому что мы не изменяем значение переменной при проверке, а сравниваем ее текущее значение с числом 10.

- Если поставить оператор присваивания в условии, то при проверке условия, значение переменной изменится, после чего это условие выполнится.

Каждому оператору `if` соответствует только один оператор `else`. Совокупность этих операторов — `else if` означает, что если не выполнилось предыдущее условие, то проверить данное. Если ни одно из условий не верно, то выполняется тело оператора `else`.

Если после оператора `if`, `else` или их связки `else if` должна выполняться только одна команда, то фигурные скобки можно не ставить. Предыдущую программу можно записать следующим образом:

```

#include <iostream>
using namespace std;
int main()
{
    setlocale(0, "");
    double num;

    cout << "Введите произвольное число: ";
    cin >> num;
    if (num < 10) // Если введенное число меньше 10.
        cout << "Это число меньше 10." << endl;
    else if (num == 10)
        cout << "Это число равно 10." << endl;
    else // иначе
        cout << "Это число больше 10." << endl;
    return 0;
}

```

Такой метод записи выглядит более компактно. Если при выполнении условия нам требуется выполнить более одной команды, то фигурные скобки необходимы. Например:

```

#include <iostream>
using namespace std;
int main()
{
    setlocale(0, "");
    double num;
    int k;
    cout << "Введите произвольное число: ";
    cin >> num;
    if (num < 10) { // Если введенное число меньше 10.
        cout << "Это число меньше 10." << endl;
        k = 1;
    } else if (num == 10) {
        cout << "Это число равно 10." << endl;
        k = 2;
    } else { // иначе
        cout << "Это число больше 10." << endl;
    }
}

```

```

    k = 3;
}
cout << "k = " << k << endl;
return 0;
}

```

Данная программа проверяет значение переменной **num**. Если она меньше 10, то присваивает переменной **k** значение единицы. Если переменная **num** равна десяти, то присваивает переменной **k** значение двойки. В противном случае — значение тройки. После выполнения ветвления, значение переменной **k** выводится на экран.

Хорошенько потренируйтесь, попробуйте придумать свой пример с ветвлением. Когда поймете, что поняли эту тему, можете смело приступать к следующему уроку.

## Циклы в C++ →. Циклы в C++ (Циклические операторы в C++)

Иногда необходимо повторять одно и то же действие несколько раз подряд. Для этого используют циклы. В этом уроке мы научимся программировать **циклы на C++**, после чего посчитаем сумму всех чисел от 1 до 1000.

- [Цикл for](#)
  - [Описание синтаксиса](#)
  - [Пример кода](#)
- [Цикл while](#)
- [Цикл do while](#)

### Цикл for

Если мы знаем точное количество действий (итераций) цикла, то можем использовать **цикл for**. Синтаксис его выглядит примерно так:

```

for (действие до начала цикла;
    условие продолжения цикла;
    действия в конце каждой итерации цикла)
{
    инструкция цикла;
    инструкция цикла 2;
    инструкция цикла N;
}

```

Итерацией цикла называется один проход этого цикла

Существует частный случай этой записи, который мы сегодня и разберем:

```

for (счетчик = значение; счетчик < значение; шаг цикла)

```

```
{  
    тело цикла;  
}
```

**Счетчик цикла** — это переменная, в которой хранится количество проходов данного цикла.

### Описание синтаксиса

1. Сначала присваивается первоначальное значение счетчику, после чего ставится точка с запятой.
2. Затем задается конечное значение счетчика цикла. После того, как значение счетчика достигнет указанного предела, цикл завершится. Снова ставим точку с запятой.
3. Задаем шаг цикла. **Шаг цикла** — это значение, на которое будет увеличиваться или уменьшаться счетчик цикла при каждом проходе.

### Пример кода

Напишем программу, которая будет считать сумму всех чисел от 1 до 1000.

```
#include <iostream>  
using namespace std;  
int main()  
{  
    int i; // счетчик цикла  
    int sum = 0; // сумма чисел от 1 до 1000.  
    setlocale(0, "");  
    for (i = 1; i <= 1000; i++) // задаем начальное значение 1, конечное 1000 и  
        // задаем шаг цикла - 1.  
    {  
        sum = sum + i;  
    }  
    cout << "Сумма чисел от 1 до 1000 = " << sum << endl;  
    return 0;  
}
```

### Цикл while

Когда мы не знаем, сколько итераций должен произвести цикл, нам понадобится цикл **while** или **do...while**. Синтаксис цикла **while** в C++ выглядит следующим образом.

**while** (Условие)

```
{  
    Тело цикла;  
}
```

Данный цикл будет выполняться, пока условие, указанное в круглых скобках является истиной. Решим ту же задачу с помощью цикла **while**. Хотя здесь мы точно знаем, сколько итераций должен выполнить цикл, очень часто бывают ситуации, когда это значение неизвестно.

Ниже приведен исходный код программы, считающей сумму всех целых чисел от 1 до 1000.

```
#include <iostream>  
using namespace std;  
int main()  
{  
    setlocale(0, "");  
    int i = 0; // инициализируем счетчик цикла.  
    int sum = 0; // инициализируем счетчик суммы.  
    while (i < 1000)  
    {  
        i++;  
        sum += i;  
    }  
    cout << "Сумма чисел от 1 до 1000 = " << sum << endl;  
    return 0;  
}
```

## Цикл do while

Цикл **do while** очень похож на цикл **while**. Единственное их различие в том, что при выполнении цикла **do while** один проход цикла будет выполнен независимо от условия. Решение задачи на поиск суммы чисел от 1 до 1000, с применением цикла *do while*.

```
#include <iostream>  
using namespace std;  
int main ()  
{  
    setlocale(0, "");  
    int i = 0; // инициализируем счетчик цикла.  
    int sum = 0; // инициализируем счетчик суммы.  
    do { // выполняем цикл.
```

```
    i++;
    sum += i;
} while (i < 1000); // пока выполняется условие.
cout << "Сумма чисел от 1 до 1000 = " << sum << endl;
return 0;
}
```

## Массивы в C++ →. Массивы в C++

Сегодня мы с поговорим о массивах. Вы уже знаете, что **переменная** — это ячейка в памяти компьютера, где может храниться одно единственное значение. **Массив** — это область памяти, где могут последовательно храниться несколько значений.

Возьмем группу студентов из десяти человек. У каждого из них есть фамилия. Создавать отдельную переменную для каждого студента — не рационально. Создадим массив, в котором будут храниться фамилии всех студентов.

- [Пример инициализации массива](#)
- [Описание синтаксиса](#)
- [Вывод элементов массива через цикл](#)
- [Объявление массива без инициализации](#)
- [Заполнение массива с клавиатуры](#)

### Пример инициализации массива

```
string students[10] = {
    "Иванов", "Петров", "Сидоров",
    "Ахмедов", "Ерошкин", "Выхин",
    "Андреев", "Вин Дизель", "Картошкин", "Чубайс"
};
```

### Описание синтаксиса

Массив создается почти так же, как и обычная переменная. Для хранения десяти фамилий нам нужен массив, состоящий из 10 элементов. Количество элементов массива задается при его объявлении и заключается в квадратные скобки.

Чтобы описать элементы массива сразу при его создании, можно использовать фигурные скобки. В фигурных скобках значения элементов массива перечисляются через запятую. В конце закрывающей фигурной скобки ставится точка с запятой.

Попробуем вывести наш массив на экран с помощью оператора **cout**.

```
#include <iostream>
```

```

#include <string>
int main()
{
    std::string students[10] = {
        "Иванов", "Петров", "Сидоров",
        "Ахмедов", "Ерошкин", "Выхин",
        "Андреев", "Вин Дизель", "Картошкин", "Чубайс"
    };
    std::cout << students << std::endl; // Пытаемся вывести весь массив
    непосредственно
    return 0;
}

```

Скомпилируйте этот код и посмотрите, на результат работы программы. Готово? А теперь запустите программу еще раз и сравните с предыдущим результатом. В моей операционной системе вывод был следующим:

- Первый вывод: 0x7ffff8b85820
- Второй вывод: 0x7fff7a335f90
- Третий вывод: 0x7ffff847eb40

Мы видим, что выводится адрес этого массива в оперативной памяти, а никакие не «Иванов» и «Петров».

Дело в том, что при создании переменной, ей выделяется определенное место в памяти. Если мы объявляем переменную типа **int**, то на машинном уровне она описывается двумя параметрами — ее адресом и размером хранимых данных.

Массивы в памяти хранятся таким же образом. Массив типа **int** из 10 элементов описывается с помощью адреса его первого элемента и количества байт, которое может вместить этот массив. Если для хранения одного целого числа выделяется 4 байта, то для массива из десяти целых чисел будет выделено 40 байт.

Так почему же, при повторном запуске программы, адреса различаются? Это сделано для защиты от атак **переполнения буфера**. Такая технология называется **рандомизацией адресного пространства** и реализована в большинстве популярных ОС.

Попробуем вывести первый элемент массива — фамилию студента Иванова.

```

#include <iostream>
#include <string>
int main()
{
    std::string students[10] = {

```

```

    "Иванов", "Петров", "Сидоров",
    "Ахмедов", "Ерошкин", "Выхин",
    "Андреев", "Вин Дизель", "Картошкин", "Чубайс"
};
std::cout << students[0] << std::endl;
return 0;
}

```

Смотрим, компилируем, запускаем. Убедились, что вывелся именно «Иванов». Заметьте, что нумерация элементов массива в С++ начинается с нуля. Следовательно, фамилия первого студента находится в `students[0]`, а фамилия последнего — в `students[9]`.

В большинстве языков программирования нумерация элементов массива также начинается с нуля.

Попробуем вывести список всех студентов. Но сначала подумаем, а что если бы вместо группы из десяти студентов, была бы кафедра их ста, факультет из тысячи, или даже весь университет? Ну не будем же мы писать десятки тысяч строк с `cout`?

Конечно же нет! Мы возьмем на вооружение циклы, о которых был написан [предыдущий урок](#).

### Вывод элементов массива через цикл

```

#include <iostream>
#include <string>
int main()
{
    std::string students[10] = {
        "Иванов", "Петров", "Сидоров",
        "Ахмедов", "Ерошкин", "Выхин",
        "Андреев", "Вин Дизель", "Картошкин", "Чубайс"
    };
    for (int i = 0; i < 10; i++) {
        std::cout << students[i] << std::endl;
    }
    return 0;
}

```

Если бы нам пришлось выводить массив из нескольких тысяч фамилий, то мы бы просто увеличили конечное значение счетчика цикла — строку `for (...; i < 10; ...)` заменили на `for (...; i < 10000; ...)`.

Заметьте что счетчик нашего цикла начинается с нуля, а заканчивается девяткой. Если вместо оператора строгого неравенства — `i < 10` использовать оператор «меньше, либо равно» — `i <= 10`, то на последней итерации программа обратится к несуществующему элементу массива — `students[10]`. Это может привести к **ошибкам сегментации** и аварийному завершению программы. Будьте внимательны — подобные ошибки бывает сложно отловить.

Массив, как и любую переменную можно не заполнять значениями при объявлении.

### Объявление массива без инициализации

```
string students[10];
```

```
// или
```

```
string teachers[5];
```

Элементы такого массива обычно содержат в себе «мусор» из выделенной, но еще не инициализированной, памяти. Некоторые компиляторы, такие как GCC, заполняют все элементы массива нулями при его создании.

При создании статического массива, для указания его размера может использоваться только константа. Размер выделяемой памяти определяется на этапе компиляции и не может изменяться в процессе выполнения.

```
int n;
```

```
cin >> n;
```

```
string students[n]; /* Неверно */
```

Выделение памяти в процессе выполнения возможно при работе с **динамическими массивами**. Но о них немного позже.

Заполним с клавиатуры пустой массив из 10 элементов.

### Заполнение массива с клавиатуры(ввод одномерного массива в C++)

```
#include <iostream>
```

```
#include <string>
```

```
using std::cout;
```

```
using std::cin;
```

```
using std::endl;
```

```
int main()
```

```
{
```

```
    int arr[10];
```

```
    // Заполняем массив с клавиатуры
```

```
    for (int i = 0; i < 10; i++) {
```

```
        cout << "[" << i + 1 << "]" << ": ";
```

```
        cin >> arr[i];
```

```

}
// И выводим заполненный массив.
cout << "\nВаш массив: ";
for (int i = 0; i < 10; ++i) {
    cout << arr[i] << " ";
}
cout << endl;
return 0;
}

```

Скомпилируем эту программу и проверим ее работу.

```

ssh
Терминал — zsh — 80x24
selevit ~ % ./foo
[1]: 4
[2]: 5
[3]: 3
[4]: 3
[5]: 5
[6]: 2
[7]: 8
[8]: 4
[9]: 4
[10]: 9
Ваш массив: 4 5 3 3 5 2 8 4 4 9
selevit ~ %

```

Если у вас возникают проблемы при компиляции исходников из уроков — внимательно прочитайте ошибку компилятора, попробуйте проанализировать и исправить ее. Если вы нашли ошибку в коде — [напишите об этом в комментариях](#) к уроку.

Массивы — очень важная вещь в программировании. Автор советует вам хорошо попрактиковаться в работе с ними.

## Функции в C++ →. Функции в C++(описание функции в C++)

Сегодня мы поговорим о функциях в C++. Очень часто в программировании необходимо выполнять одни и те же действия. Например, мы хотим выводить пользователю сообщения об ошибке в разных местах программы, если он ввел неверное значение. без функций это выглядело бы так:

```

#include <iostream>
#include <string>
using namespace std;
int main()
{
    string valid_pass = "qwerty123";

```

```

string user_pass;
cout << "Введите пароль: ";
getline(cin, user_pass);
if (user_pass == valid_pass) {
    cout << "Доступ разрешен." << endl;
} else {
    cout << "Неверный пароль!" << endl;
}
return 0;
}

```

А вот аналогичный пример с функцией:

```

#include <iostream>
#include <string>
using namespace std;
void check_pass (string password)
{
    string valid_pass = "qwerty123";
    if (password == valid_pass) {
        cout << "Доступ разрешен." << endl;
    } else {
        cout << "Неверный пароль!" << endl;
    }
}
int main()
{
    string user_pass;
    cout << "Введите пароль: ";
    getline (cin, user_pass);
    check_pass (user_pass);
    return 0;
}

```

По сути, после компиляции не будет никакой разницы для процессора, как для первого кода, так и для второго. Но ведь такую проверку пароля мы можем делать в нашей программе довольно много раз. И тогда получается копия паста и код становится нечитаемым. Функции — один из самых важных компонентов языка C++.

- Любая функция имеет тип, также, как и любая переменная.

- Функция может возвращать значение, тип которого в большинстве случаев аналогично типу самой функции.
- Если функция не возвращает никакого значения, то она должна иметь тип **void** (такие функции иногда называют процедурами)
- При объявлении функции, после ее типа должно находиться имя функции и две круглые скобки — открывающая и закрывающая, внутри которых могут находиться один или несколько аргументов функции, которых также может не быть вообще.
- после списка аргументов функции ставится открывающая фигурная скобка, после которой находится само тело функции.
- В конце тела функции обязательно ставится закрывающая фигурная скобка.

### Пример построения функции

```
#include <iostream>
using namespace std;
```

```
void function_name ()
{
    cout << "Hello, world" << endl;
}
```

```
int main()
{
    function_name(); // Вызов функции
    return 0;
}
```

Перед вами тривиальная программа, **Hello, world**, только реализованная с использованием функций.

Если мы хотим вывести «Hello, world» где-то еще, нам просто нужно вызвать соответствующую функцию. В данном случае это делается так: `function_name();`. Вызов функции имеет вид имени функции с последующими круглыми скобками. Эти скобки могут быть пустыми, если функция не имеет аргументов. Если же аргументы в самой функции есть, их необходимо указать в круглых скобках.

Также существует такое понятие, как параметры функции по умолчанию. Такие параметры можно не указывать при вызове функции, т.к. они примут значение по умолчанию, указанно после знака присваивания после данного параметра и списке всех параметров функции.

В предыдущих примерах мы использовали функции типа **void**, которые не возвращают никакого значения. Как многие уже догадались, оператор **return** используется для возвращения вычисляемого функцией значения.

Рассмотрим пример функции, возвращающей значение на примере проверки пароля.

```
#include <iostream>
#include <string>
using namespace std;

string check_pass (string password)
{
    string valid_pass = "qwerty123";
    string error_message;
    if (password == valid_pass) {
        error_message = "Доступ разрешен.";
    } else {
        error_message = "Неверный пароль!";
    }
    return error_message;
}

int main()
{
    string user_pass;
    cout << "Введите пароль: ";
    getline (cin, user_pass);
    string error_msg = check_pass (user_pass);
    cout << error_msg << endl;
    return 0;
}
```

В данном случае функция **check\_pass** имеет тип **string**, следовательно она будет возвращать только значение типа **string**, иными словами говоря строку. Давайте рассмотрим алгоритм работы этой программы.

Самой первой выполняется функция **main()**, которая должна присутствовать в каждой программе. Теперь мы объявляем переменную **user\_pass** типа **string**, затем выводим пользователю сообщение «Введите пароль», который после ввода попадает в строку **user\_pass**. А вот дальше начинает работать наша собственная функция **check\_pass()**.

В качестве аргумента этой функции передается строка, введенная пользователем.

Аргумент функции — это, если сказать простым языком переменные или константы вызывающей функции, которые будет использовать вызываемая функция.

При объявлении функций создается **формальный параметр**, имя которого может отличаться от параметра, передаваемого при вызове этой функции. Но типы формальных параметров и передаваемых функции аргументов в большинстве случаев должны быть аналогичны.

После того, как произошел вызов функции `check_pass()`, начинает работать данная функция. Если функцию нигде не вызвать, то этот код будет проигнорирован программой. Итак, мы передали в качестве аргумента строку, которую ввел пользователь.

Теперь эта строка в полном распоряжении функции (хочу обратить Ваше внимание на то, что переменные и константы, объявленные в разных функциях независимы друг от друга, они даже могут иметь одинаковые имена. В следующих уроках я расскажу о том, что такое область видимости, локальные и глобальные переменные).

Теперь мы проверяем, правильный ли пароль ввел пользователь или нет. Если пользователь ввел правильный пароль, присваиваем переменной `error_message` соответствующее значение. Если нет, то сообщение об ошибке.

После этой проверки мы **возвращаем** переменную `error_message`. На этом работа нашей функции закончена. А теперь, в функции `main()`, то значение, которое возвратила наша функция мы присваиваем переменной `error_msg` и выводим это значение (строку) на экран терминала.

Также, можно организовать повторный ввод пароля с помощью **рекурсии** (о ней мы еще поговорим). Если объяснить вкратце, рекурсия — это когда функция вызывает сама себя. Смотрите еще один пример:

```
#include <iostream>
#include <string>
using namespace std;
bool password_is_valid (string password)
{
    string valid_pass = "qwerty123";
    if (valid_pass == password)
        return true;
    else
        return false;
}
```

```

void get_pass ()
{
    string user_pass;
    cout << "Введите пароль: ";
    getline(cin, user_pass);
    if (!password_is_valid(user_pass)) {
        cout << "Неверный пароль!" << endl;
        get_pass (); // Здесь делаем рекурсию
    } else {
        cout << "Доступ разрешен." << endl;
    }
}

int main()
{
    get_pass ();
    return 0;
}

```

Функции очень сильно облегчают работу программисту и намного повышают читаемость и понятность кода, в том числе и для самого разработчика (не удивляйтесь этому, т. к. если вы откроете код, написанный вами полгода назад, не сразу поймете соль, поверьте на слово).

## Указатели в C++ →. Указатели в C++

При выполнении любой программы, все необходимые для ее работы данные должны быть загружены в оперативную память компьютера. Для обращения к переменным, находящимся в памяти, используются специальные адреса, которые записываются в шестнадцатеричном виде, например **0x100** или **0x200**.

Если переменных в памяти потребуется слишком большое количество, которое не сможет вместить в себя сама аппаратная часть, произойдет перегрузка системы или её зависание.

Если мы объявляем переменные статично, так как мы делали в предыдущих уроках, они остаются в памяти до того момента, как программа завершит свою работу, а после чего уничтожаются.

Такой подход может быть приемлем в простых примерах и несложных программах, которые не требуют большого количества ресурсов. Если же наш проект является огромным программным комплексом с высоким функционалом, объявлять таким образом переменные, естественно, было бы довольно не умно.

Можете себе представить, если бы неизвестная **Battlefield 3** использовала такой метод работы с данными? В таком случае, самым злым геймерам пришлось бы перезагружать свои высоконагруженные системы кнопкой reset после нескольких секунд работы игры.

Дело в том, что играя в тот же Battlefield, геймер в каждый новый момент времени видит различные объекты на экране монитора, например сейчас я стреляю во врага, а через долю секунды он уже падает убитым, создавая вокруг себя множество спецэффектов, таких как пыль, тени, и т.п.

Естественно, все это занимает какое-то место в оперативной памяти компьютера. Если не уничтожать неиспользуемые объекты, очень скоро они заполнят весь объем ресурсов ПК.

По этим причинам, в большинстве языков, в том числе и C/C++, имеется понятие указателя. Указатель — это переменная, хранящая в себе адрес ячейки оперативной памяти, например **0x100**.

Мы можем обращаться, например к массиву данных через указатель, который будет содержать адрес начала диапазона ячеек памяти, хранящих этот массив.

После того, как этот массив станет не нужен для выполнения остальной части программы, мы просто освободим память по адресу этого указателя, и она вновь станет доступно для других переменных.

Ниже приведен конкретный пример обращения к переменным через указатель и напрямую.

### Пример использования статических переменных в C++

```
#include <iostream>
using namespace std;
int main()
{
    int a; // Объявление статической переменной
    int b = 5; // Инициализация статической переменной b

    a = 10;
    b = a + b;
    cout << "b is " << b << endl;
    return 0;
}
```

### Пример использования динамических переменных в C++

```
#include <iostream>
using namespace std;
```

```

int main()
{
    int *a = new int; // Объявление указателя для переменной типа int
    int *b = new int(5); // Инициализация указателя
    *a = 10;
    *b = *a + *b;
    cout << "b is " << *b << endl;
    delete b;
    delete a;
    return 0;
}

```

Синтаксис первого примера вам уже должен быть знаком. Мы объявляем/инициализируем статические переменные **a** и **b**, после чего выполняем различные операции напрямую с ними.

Во втором примере мы оперируем динамическими переменными посредством указателей. Рассмотрим общий синтаксис указателей в C++.

**Выделение памяти** осуществляется с помощью оператора **new** и имеет вид: **тип\_данных \*имя\_указателя = new тип\_данных;**, например **int \*a = new int;**. После удачного выполнения такой операции, в оперативной памяти компьютера происходит выделение диапазона ячеек, необходимого для хранения переменной типа **int**.

**Инициализация значения**, находящегося по адресу указателя выполняется схожим образом, только в конце ставятся круглые скобки с нужным значением: **тип\_данных \*имя\_указателя = new тип\_данных(значение)**. В нашем примере это **int \*b = new int(5)**.

Для того, чтобы получить **адрес** в памяти, на который ссылается указатель, используется имя переменной-указателя с префиксом **&** перед ним (*не путать со знаком ссылки в C++*).

Например, чтобы вывести на экран адрес ячейки памяти, на который ссылается указатель **b** во втором примере, мы пишем **cout << "Address of b is " << &b << endl;**. В моей системе, я получил значение **0x1aba030**. У вас оно может быть другим, потому что адреса в оперативной памяти распределяются таким образом, чтобы максимально уменьшить фрагментацию. Поскольку, в любой системе список запущенных процессов, а также объем и разрядность памяти могут отличаться, система сама распределяет данные для обеспечения минимальной фрагментации.

Для того, чтобы получить **значение**, которое находится **по адресу**, на который ссылается указатель, используется **префикс \***. Данная операция называется **разыменованием указателя**.

Во втором примере мы выводим на экран значение, которое находится в ячейке памяти (у меня это **0x1aba030**): `cout << "b is " << *b << endl;`. В этом случае необходимо использовать знак\*.

Чтобы изменить значение, находящееся по адресу, на который ссылается указатель, нужно также использовать звездочку, например, как во втором примере — `*b = *a + *b;`.

- Когда мы оперируем **данными**, то используем знак **\***
- Когда мы оперируем **адресами**, то используем знак **&**

В этих вещах очень часто возникают недопонимания, и кстати, не только у новичков. Многие из тех, кто начинал программировать с того же **php**, также часто испытывают подобную путаницу при работе с памятью.

Для того, чтобы освободить память, выделенную оператором **new**, используется оператор **delete**.

### Пример освобождения памяти

```
#include <iostream>
using namespace std;
int main()
{
    // Выделение памяти
    int *a = new int;
    int *b = new int;
    float *c = new float;
    // ... Любые действия программы
    // Освобождение выделенной памяти
    delete c;
    delete b;
    delete a;
    return 0;
}
```

При использовании оператора **delete** для указателя, знак **\*** не используется.

### Динамические массивы в C++ →. Динамические массивы в C++

В пятом уроке мы разобрали понятие массива. При объявлении, мы задавали массиву определенный постоянный размер. Возможно, кто-то из читателей пробовал делать так:

```
int n = 10;
```

```
int arr[n];
```

Но, как уже было сказано — при объявлении статического массива, его размером должна являться числовая константа, а не переменная. В большинстве случаев, целесообразно выделять определенное количество памяти для массива, значение которого изначально неизвестно.

Например, необходимо создать динамический массив из  $N$  элементов, где значение  $N$  задается пользователем. В [предыдущем уроке](#) мы учились выделять память для переменных, используя указатели. Выделение памяти для динамического массива имеет аналогичный принцип.

### Создание динамического массива

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int num; // размер массива
```

```
    cout << "Enter integer value: ";
```

```
    cin >> num; // получение от пользователя размера массива
```

```
    int *p_darr = new int[num]; // Выделение памяти для массива
```

```
    for (int i = 0; i < num; i++) {
```

```
        // Заполнение массива и вывод значений его элементов
```

```
        p_darr[i] = i;
```

```
        cout << "Value of " << i << " element is " << p_darr[i] << endl;
```

```
    }
```

```
    delete [] p_darr; // очистка памяти
```

```
    return 0;
```

```
}
```

Синтаксис выделения памяти для массива имеет вид **указатель = new тип[размер]**. В качестве размера массива может выступать любое целое положительное значение.

### Параметры командной строки в C++ →. Параметры командной строки в C++

При запуске программы из командной строки, ей можно передавать дополнительные параметры в текстовом виде. Например, следующая команда

```
ping -t 5 google.com
```

Будет отправлять пакеты на адрес google.com с интервалом в 5 секунд. Здесь мы передали программе ping три параметра: «-t», «5» и «google.com», которые программа интерпретирует как задержку между запросами и адрес хоста для обмена пакетами.

В программе эти параметры из командной строки можно получить через аргументы функции `main` при использовании функции `main` в следующей форме:

```
int main(int argc, char* argv[]) { /* ... */ }
```

Первый аргумент содержит количество параметров командной строки. Вторым аргументом — это массив строк, содержащий параметры командной строки. Т.е. первый аргумент указывает количество элементов массива во втором аргументе.

Первый элемент массива строк (`argv[0]`) всегда содержит строку, использованную для запуска программы (либо пустую строку). Следующие элементы (от 1 до `argc - 1`) содержат параметры командной строки, если они есть. Элемент массива строк `argv[argc]` всегда должен содержать 0.

### Пример 1.1

```
#include <iostream>
```

```
using namespace std;
```

```
int main(int argc, char *argv[])
```

```
{
```

```
    for (int i = 0; i < argc; i++) {
```

```
        // Выводим список аргументов в цикле
```

```
        cout << "Argument " << i << " : " << argv[i] << endl;
```

```
    }
```

```
    return 0;
```

```
}
```

Откройте командную строку и запустите оттуда скомпилированную программу.

```
Терминал — zsh — 101x25
vim zsh zsh
selevit Documents % ./foo
Argument 0 : ./foo
selevit Documents % ./foo 10 44 77
Argument 0 : ./foo
Argument 1 : 10
Argument 2 : 44
Argument 3 : 77
selevit Documents % ./foo 100 200 300 400
Argument 0 : ./foo
Argument 1 : 100
Argument 2 : 200
Argument 3 : 300
Argument 4 : 400
selevit Documents % ./foo abc def ghi klm
Argument 0 : ./foo
Argument 1 : abc
Argument 2 : def
Argument 3 : ghi
Argument 4 : klm
selevit Documents %
```

Для получения числовых данных из входных параметров, можно использовать функции `atoi` и `atof`.

## Классы в C++ →. Классы в C++

Весь реальный мир состоит из объектов. Города состоят из районов, в каждом районе есть свои названия улиц, на каждой улице находятся жилые дома, которые также состоят из объектов.

Практически любой материальный предмет можно представить в виде совокупности объектов, из которых он состоит. Допустим, что нам нужно написать программу для учета успеваемости студентов. Можно представить группу студентов, как класс языка C++. Назовем его **Students**.

- Основные понятия
- Модификаторы доступа `public` и `private`
- Программа учета успеваемости студентов
- Отделение данных от логики
- Создание объекта через указатель
- Конструктор и деструктор класса

```
class Students {
    // Имя студента
    std::string name;
    // Фамилия
    std::string last_name;
    // Пять промежуточных оценок студента
    int scores[5];
    // Итоговая оценка за семестр
    float average_ball;
};
```

## Основные понятия

Классы в программировании состоят из свойств и методов. Свойства — это любые данные, которыми можно характеризовать объект класса. В нашем случае, объектом класса является студент, а его свойствами — имя, фамилия, оценки и средний балл.

У каждого студента есть имя — `name` и фамилия `last_name`. Также, у него есть промежуточные оценки за весь семестр. Эти оценки мы будем записывать в целочисленный массив из пяти элементов. После того, как все пять оценок будут проставлены, определим средний балл успеваемости студента за весь семестр — свойство `average_ball`.

Методы — это функции, которые могут выполнять какие-либо действия над данными (свойствами) класса. Добавим в наш класс функцию `calculate_average_ball()`, которая будет определять средний балл успеваемости ученика.

- **Методы** класса — это его функции.
- **Свойства** класса — его переменные.

```
class Students {
    public:
        // Функция, считающая средний балл
        void calculate_average_ball()
        {
            int sum = 0; // Сумма всех оценок
            for (int i = 0; i < 5; ++i) {
                sum += scores[i];
            }
            // считаем среднее арифметическое
            average_ball = sum / 5.0;
        }

        // Имя студента
        std::string name;
        // Фамилия
        std::string last_name;
        // Пять промежуточных оценок студента
        int scores[5];

    private:
```

*// Итоговая оценка за семестр*

```
float average_ball;  
};
```

Функция `calculate_average_ball()` просто делит сумму всех промежуточных оценок на их количество.

## Модификаторы доступа `public` и `private`

Все свойства и методы классов имеют права доступа. По умолчанию, все содержимое класса является доступным для чтения и записи только для него самого. Для того, чтобы разрешить доступ к данным класса извне, используют модификатор доступа `public`. Все функции и переменные, которые находятся после модификатора `public`, становятся доступными из всех частей программы.

Закрытые данные класса размещаются после модификатора доступа `private`. Если отсутствует модификатор `public`, то все функции и переменные, по умолчанию являются закрытыми (как в первом примере).

Обычно, приватными делают все свойства класса, а публичными — его методы. Все действия с закрытыми свойствами класса реализуются через его методы. Рассмотрим следующий код.

```
class Students {  
    public:  
        // Установка среднего балла  
        void set_average_ball(float ball)  
        {  
            average_ball = ball;  
        }  
        // Получение среднего балла  
        float get_average_ball()  
        {  
            return average_ball;  
        }  
        std::string name;  
        std::string last_name;  
        int scores[5];  
  
    private:  
        float average_ball;  
};
```

Мы не можем напрямую обращаться к закрытым данным класса. Работать с этими данными можно только посредством методов этого класса. В примере выше, мы используем функцию `get_average_ball()` для получения средней оценки студента, и `set_average_ball()` для выставления этой оценки.

Функция `set_average_ball()` принимает средний балл в качестве параметра и присваивает его значение закрытой переменной `average_ball`. Функция `get_average_ball()` просто возвращает значение этой переменной.

## Программа учета успеваемости студентов

Создадим программу, которая будет заниматься учетом успеваемости студентов в группе. Создайте заголовочный файл `students.h`, в котором будет находиться класс `Students`.

```
/* students.h */
```

```
#include <string>
```

```
class Students {
```

```
    public:
```

```
        // Установка имени студента
```

```
        void set_name(std::string student_name)
```

```
        {
```

```
            name = student_name;
```

```
        }
```

```
        // Получение имени студента
```

```
        std::string get_name()
```

```
        {
```

```
            return name;
```

```
        }
```

```
        // Установка фамилии студента
```

```
        void set_last_name(std::string student_last_name)
```

```
        {
```

```
            last_name = student_last_name;
```

```
        }
```

```
        // Получение фамилии студента
```

```
        std::string get_last_name()
```

```
        {
```

```
            return last_name;
```

```
        }
```

```
        // Установка промежуточных оценок
```

```

void set_scores(int student_scores[])
{
    for (int i = 0; i < 5; ++i) {
        scores[i] = student_scores[i];
    }
}
// Установка среднего балла
void set_average_ball(float ball)
{
    average_ball = ball;
}
// Получение среднего балла
float get_average_ball()
{
    return average_ball;
}

```

**private:**

```

// Промежуточные оценки
int scores[5];
// Средний балл
float average_ball;
// Имя
std::string name;
// Фамилия
std::string last_name;
};

```

Мы добавили в наш класс новые методы, а также сделали приватными все его свойства. Функция `set_name()` сохраняет имя студента в переменной `name`, а `get_name()` возвращает значение этой переменной. Принцип работы функций `set_last_name()` и `get_last_name()` аналогичен.

Функция `set_scores()` принимает массив с промежуточными оценками и сохраняет их в приватную переменную `int scores[5]`.

Теперь создайте файл `main.cpp` со следующим содержимым.

```

/* main.cpp */
#include <iostream>
#include "students.h"

```

```
int main()
{
    // Создание объекта класса Student
    Students student;

    std::string name;
    std::string last_name;

    // Ввод имени с клавиатуры
    std::cout << "Name: ";
    getline(std::cin, name);

    // Ввод фамилии
    std::cout << "Last name: ";
    getline(std::cin, last_name);

    // Сохранение имени и фамилии в объект класса Students
    student.set_name(name);
    student.set_last_name(last_name);

    // Оценки
    int scores[5];
    // Сумма всех оценок
    int sum = 0;

    // Ввод промежуточных оценок
    for (int i = 0; i < 5; ++i) {
        std::cout << "Score " << i+1 << ": ";
        std::cin >> scores[i];
        // суммирование
        sum += scores[i];
    }

    // Сохраняем промежуточные оценки в объект класса Student
    student.set_scores(scores);
    // Считаем средний балл
```

```

float average_ball = sum / 5.0;
// Сохраняем средний балл в объект класса Students
student.set_average_ball(average_ball);
// Выводим данные по студенту
std::cout << "Average ball for " << student.get_name() << " "
    << student.get_last_name() << " is "
    << student.get_average_ball() << std::endl;

return 0;
}

```

В самом начале программы создается объект класса **Students**. Дело в том, что сам класс является только описанием его объекта. Класс **Students** является описанием любого из студентов, у которого есть имя, фамилия и возможность получения оценок.

Объект класса **Students** характеризует конкретного студента. Если мы захотим выставить оценки всем ученикам в группе, то будем создавать новый объект для каждого из них. Использование классов очень хорошо подходит для описания объектов реального мира.

После создания объекта **student**, мы вводим с клавиатуры фамилию, имя и промежуточные оценки для конкретного ученика. Пусть это будет Вася Пупкин, у которого есть пять оценок за семестр — две тройки, две четверки и одна пятерка.

Введенные данные мы передаем **set**-функциям, которые присваивают их закрытым переменным класса. После того, как были введены промежуточные оценки, мы высчитываем средний балл на основе этих оценок, а затем сохраняем это значение в закрытом свойстве **average\_ball**, с помощью функции **set\_average\_ball()**.

Скомпилируйте и запустите программу.

```

selevit@gentoo students-cpp % c++ main.cpp -o students
selevit@gentoo students-cpp % ./students
Name: Вася
Last name: Пупкин
Score 1: 3
Score 2: 3
Score 3: 4
Score 4: 4
Score 5: 5
Average ball for Вася Пупкин is 3.8
selevit@gentoo students-cpp %

```

## Отделение данных от логики

Вынесем реализацию всех методов класса в отдельный файл **students.cpp**.

```
/* students.cpp */
#include <string>
#include "students.h"

// Установка имени студента
void Students::set_name(std::string student_name)
{
    Students::name = student_name;
}

// Получение имени студента
std::string Students::get_name()
{
    return Students::name;
}

// Установка фамилии студента
void Students::set_last_name(std::string student_last_name)
{
    Students::last_name = student_last_name;
}

// Получение фамилии студента
std::string Students::get_last_name()
{
    return Students::last_name;
}

// Установка промежуточных оценок
void Students::set_scores(int scores[])
{
    for (int i = 0; i < 5; ++i) {
        Students::scores[i] = scores[i];
    }
}

// Установка среднего балла
```

```
void Students::set_average_ball(float ball)
{
    Students::average_ball = ball;
}
```

*// Получение среднего балла*

```
float Students::get_average_ball()
{
    return Students::average_ball;
}
```

А в заголовочном файле **students.h** оставим только прототипы этих методов.

*/\* students.h \*/*

**#pragma once** */\* Защита от двойного подключения заголовочного файла \*/*

**#include** **<string>**

```
class Students {
    public:
        // Установка имени студента
        void set_name(std::string);
        // Получение имени студента
        std::string get_name();
        // Установка фамилии студента
        void set_last_name(std::string);
        // Получение фамилии студента
        std::string get_last_name();
        // Установка промежуточных оценок
        void set_scores(int []);
        // Установка среднего балла
        void set_average_ball(float);
        // Получение среднего балла
        float get_average_ball();

    private:
        // Промежуточные оценки
        int scores[5];
        // Средний балл
        float average_ball;
```

```
// Имя
std::string name;
// Фамилия
std::string last_name;
};
```

Такой подход называется **абстракцией данных** — одного из **фундаментальных принципов** объектно-ориентированного программирования. К примеру, если кто-то другой захочет использовать наш класс в своем коде, ему не обязательно знать, как именно высчитывается средний балл. Он просто будет использовать функцию `calculate_average_ball()` из второго примера, не вникая в алгоритм ее работы.

## Создание объекта через указатель

При создании объекта, лучше не копировать память для него, а выделять ее в куче с помощью **указателя**. И освободить ее после того, как мы закончили работу с объектом. Реализуем это в нашей программе, немного изменив содержимое файла `main.cpp`.

```
/* main.cpp */
#include <iostream>
#include "students.h"

int main()
{
    // Выделение памяти для объекта Students
    Students *student = new Students;

    std::string name;
    std::string last_name;

    // Ввод имени с клавиатуры
    std::cout << "Name: ";
    getline(std::cin, name);

    // Ввод фамилии
    std::cout << "Last name: ";
    getline(std::cin, last_name);

    // Сохранение имени и фамилии в объект класса Students
    student->set_name(name);
```

```

student->set_last_name(last_name);

// Оценки
int scores[5];
// Сумма всех оценок
int sum = 0;

// Ввод промежуточных оценок
for (int i = 0; i < 5; ++i) {
    std::cout << "Score " << i+1 << ": ";
    std::cin >> scores[i];
    // суммирование
    sum += scores[i];
}
// Сохраняем промежуточные оценки в объект класса Student
student->set_scores(scores);

// Считаем средний балл
float average_ball = sum / 5.0;
// Сохраняем средний балл в объект класса Students
student->set_average_ball(average_ball);
// Выводим данные по студенту
std::cout << "Average ball for " << student->get_name() << " "
    << student->get_last_name() << " is "
    << student->get_average_ball() << std::endl;
// Удаление объекта student из памяти
delete student;
return 0;
}

```

При создании статического объекта, для доступа к его методам и свойствам, используют операция прямого обращения — «.» (символ точки). Если же память для объекта выделяется посредством указателя, то для доступа к его методам и свойствам используется оператор косвенного обращения — «->».

## Список литературы

1. Информатика в понятиях и терминах: Кн. для учащихся ст. классов сред. шк./ Г.А. Бордовский, В.А. Извозчиков, Ю.В. Исаев, В.В. Морозов; Под ред. В.А. Извозчикова. - М.: Просвещение, 1991. - 208 с.
2. Основы информатики и вычислительной техники: Проб. учеб. пособие для сред. учеб. заведений. В 2-х ч. Ч.1/ А.П. Ершов, В.М. Монахов, С.А. Бешенков и др.; Под ред. А.П. Ершова, В.М. Монахова. - М.: Просвещение, 1985. - 96 с.
3. Основы информатики и вычислительной техники: Проб. учеб. пособие для сред. учеб. заведений. В 2-х ч. Ч.2/ А.П. Ершов, В.М. Монахов, А.А. Кузнецов и др.; Под ред. А.П. Ершова, В.М. Монахова. - М.: Просвещение, 1986. - 143 с.
4. Гук М. Аппаратные средства IBM PC. Энциклопедия. СПб.: Издательство "Питер", 2000. - 816 с. Средств и.т.д.
5. Гордеев А. В. Операционные системы: Учебник для вузов. — 2-е изд. — СПб.: Питер, 2007. — 416 с.
6. Деннинг П. Дж., Браун Р. Л. Операционные системы // Современный компьютер. — М., 1986.
7. Иртегов Д. В. Введение в операционные системы. — 2-е изд. — СПб.: ВHV-СПб, 2007.
8. Керниган Б. У., Пайк Р. У. UNIX — универсальная среда программирования = The UNIX Programming Environment. — М., 1992.
9. Олифер В. Г., Олифер Н. А. Сетевые операционные системы. — СПб.: Питер, 2002. — 544 с.
10. СтоллингсУ. Операционные системы = Operating Systems: Internals and Design Principles. — М.: Вильямс, 2004. — 848 с
11. ТаненбаумЭ. С. Многоуровневая организация ЭВМ = Structured Computer Organization. — М.: Мир, 1979. — 547 с.
12. Таненбаум Э. С. Современные операционные системы = Modern Operating Systems. — 2-е изд. — СПб.: Питер, 2005. — 1038 с.

### Список веб сайтов:

1. <http://pc-azbuka.ru/standartnye-programmy-windows/>
2. <http://windowstips.ru/notes/5941>
3. <http://pc-azbuka.ru/kalkulyator-v-windows/>
4. <http://volboris.ru/standartniye-programmyi-windows-ekrannaya-klaviatura>