

Кафедра «Математика и информатика»

Текст лекций по предмету «Информатика»

Составитель: Кадирова Н.

Ташкент – 2010

## Оглавление

[Лекция 1.](#) Информация и предмет информатики.

[Лекция 2.](#) Аппаратные средства персонального компьютера.

[Лекция 3.](#) Программное обеспечение ПК и основные функции операционной системы.

Служебные программы операционной системы Windows XP.

[Лекция 4.](#) Компьютерные сети. Протоколы. Адресация в Интернете.

[Лекция 5.](#) Создание Web-страниц. Язык HTML. Структура документа.

[Лекция 6.](#) Теги форматирования документов.

[Лекция 7.](#) Таблицы. Графические объекты.

[Лекция 8.](#) Основные понятия базы данных. Назначение и возможности Ms Access.

[Лекция 9.](#) Создание таблиц в программе Access.

[Лекция 10.](#) Организация Запросов.

[Лекция 11.](#) Создание форм. Панель управления. Создание отчетов.

[Лекция 12.](#) Назначение Ms Excel и основные возможности. Стандартные операции.

[Лекция 13.](#) Формулы и функции.

[Лекция 14.](#) Работа с диаграммами в Excel.

[Лекция 15.](#) Создание макросов.

## Введение

Способность *принимать, передавать, запоминать* и некоторым образом *обрабатывать* разнообразную *информацию* является, по-видимому, одним из самых главных признаков живого мира. В той или иной степени это свойство присуще животным, растениям и даже бактериям. Информация не только поступает в организм извне, но и передается живым организмом во внешнюю среду. Так, с помощью звука или характерных поз животные дают понять о своих намерениях. Обмениваясь различного рода сигналами, пчелы передают друг другу информацию об опасности или о найденном месте для сбора нектара. Наиболее важные сведения живые организмы запоминают, и эта «хранящаяся» в организме информация активно используется в процессе обработки вновь поступающей информации. Например, найдя дорогу к водопою, животные ее запоминают и затем неоднократно по ней проходят. Следовательно, с помощью приема, передачи, хранения и обработки различного рода информационных потоков из внешней среды любой отдельно взятый организм обеспечивает свою жизнедеятельность. В связи с этим необходимо отметить, что информация для живого организма не менее важна, чем пища, с помощью которой в организм поступает необходимая для его жизнедеятельности энергия.

## Лекция 1

### Информация и предмет информатики.

**Цель лекции:** ознакомить с понятием информатики, с составными этого предмета, с понятием информации и ее свойств, с видом представления данных и их хранения в компьютере, с понятиями информационных систем и информационных технологий.

**Ключевые слова:** Предмет информатики, информация, свойства информации, кодирование информации, структуры данных (линейная, табличная, иерархическая), хранение данных(файл, формат файла), информационная система, информационные технологии.

#### Содержание

Введение.

Предмет информатики

Информация. Свойства информации. Кодирование информации.

Структура данных. Хранение данных.

Понятие информационной системы

Понятие информационных технологий

Предмет информатики. Информатика - это комплексная, техническая наука, которая систематизирует приемы создания, сохранения, воспроизведения, обработки и передачи данных средствами вычислительной техники, а также принципы функционирования этих средств и методы управления ними. Термин "информатика" происходит от французского слова *Informatique* и образован из двух слов: информация и автоматика. Этот термин введен во Франции в середине 60-х лет XX ст., когда началось широкое использование вычислительной техники. Тогда в англоязычных странах вошел в употребление термин "*Computer Science*" для обозначения науки о преобразовании информации, которая базируется на использовании вычислительной техники. Теперь эти термины являются синонимами. Появление информатики обусловлено возникновением и распространением новой технологии сбора, обработки и передачи информации, связанной с фиксацией данных на машинных носителях.

Предмет информатики как науки составляют:

- аппаратное обеспечение средств вычислительной техники;
- программное обеспечение средств вычислительной техники;
- средства взаимодействия аппаратного и программного обеспечения;
- средства взаимодействия человека с аппаратными и программными средствами.
- Средства взаимодействия в информатике принято называть интерфейсом.

Поэтому средства взаимодействия аппаратного и программного обеспечения иногда называют также программно-аппаратным интерфейсом, а средства взаимодействия человека с аппаратными и программными средствами - интерфейсом пользователя.

Основная задача информатики как науки - это систематизация приемов и методов работы с аппаратными и программными средствами вычислительной техники. Цель систематизации состоит в том, чтобы выделять, внедрять и развивать передовые, более эффективные технологии автоматизации этапов работы с данными, а также методически обеспечивать новые технологические исследования.

Информатика - практическая наука. Ее достижения должны проходить проверку на практике и приниматься в тех случаях, если они отвечают критерию повышения эффективности. В составе основной задачи сегодня можно выделить такие основные направления информатики для практического применения :

- архитектура вычислительных систем (приемы и методы построения систем, предназначенных для автоматической обработки данных);
- интерфейсы вычислительных систем (приемы и методы управления аппаратным и программным обеспечением);
- программирование (приемы, методы и средства разработки комплексных задач); преобразование данных (приемы и методы преобразования структур данных);
- защита информации (обобщение приемов, разработка методов и средств защиты данных);
- автоматизация (функционирование программно-аппаратных средств без участия человека);
- стандартизация (обеспечение совместимости между аппаратными и программными средствами, между форматами представления данных, относящихся к разным типам вычислительных систем).

На всех этапах технического обеспечения информационных процессов для информатики ключевым вопросом является эффективность. Для аппаратных средств под эффективностью понимают соотношение производительности оснащение к его стоимости. Для программного обеспечения под эффективностью принято понимать производительность работающих с ним пользователей. В программировании под эффективностью понимают объем программного кода, созданного программистами за единицу времени. В информатике всю жестко ориентированное на эффективность. Вопрос как осуществить ту или другую операцию, для информатики важный, но не основной. Основным есть вопрос как совершить данную операцию эффективно.

В рамках информатики, как технической науки можно сформулировать понятия информации, информационной системы и информационной технологии.

## Информация. Свойства информации

Информация - это совокупность сведений (данных), которая воспринимается из окружающей среды (входная информация), выдается в окружающую среду (исходная информация) или сохраняется внутри определенной системы. Информация существует в виде документов, чертежей, рисунков, текстов, звуковых и световых сигналов, электрических и нервных импульсов и т.п.. Важнейшие свойства информации:

- объективность и субъективность;

*Дуализм информации* характеризует ее двойственность. С одной стороны, информация *объективна* в силу объективности данных, с другой — *субъективна*, в силу субъективности применяемых методов. Иными словами, методы могут вносить в большей или меньшей степени субъективный фактор и таким образом влиять на информацию в целом. Например, два человека читают одну и ту же книгу и получают подчас весьма разную информацию, хотя прочитанный текст, т.е. данные, были одинаковы. Более объективная информация применяет методы с меньшим субъективным элементом.

- полнота (достаточность для принятия решения);

*Полнота информации* характеризует степень *достаточности* данных для принятия решения или создания новых данных на основе имеющихся. Неполный набор данных оставляет большую долю неопределенности, т.е. большое число вариантов выбора, а это требует применения дополнительных методов, например, экспертных оценок, бросание жребия и т.п. Избыточный набор данных затрудняет доступ к нужным данным, создает повышенный информационный шум, что также вызывает необходимость дополнительных методов, например, фильтрацию, сортировку. И неполный и избыточный наборы затрудняют получение информации и принятие адекватного решения.

- достоверность( соответствие информации объективной реальности);  
*Достоверность информации* — это свойство, характеризующее степень соответствия информации реальному объекту с необходимой *точностью*. При работе с неполным набором данных достоверность

информации может характеризоваться вероятностью, например, можно сказать, что при бросании монеты с вероятностью 50 % выпадет герб.

- адекватность(степень соответствия информации, полученной потребителем тому, что вложено в нее автором);

*Адекватность информации* выражает степень соответствия создаваемого с помощью информации образа реальному объекту, процессу, явлению. Полная адекватность достигается редко, так как обычно приходится работать с не самым полным набором данных, т.е. присутствует неопределенность, затрудняющая принятие адекватного решения. Получение адекватной информации также затрудняется при недоступности адекватных методов.

- доступность(мера возможности получить ту или иную информацию);

*Доступность информации* — это возможность получения информации при необходимости. Доступность складывается из двух составляющих: из доступности данных и доступности методов. Отсутствие хотя бы одного дает неадекватную информацию.

- актуальность(степень соответствия информации текущему моменту времени).

*Актуальность информации.* Информация существует во времени, так как существуют во времени все информационные процессы.

Информация, актуальная сегодня, может стать совершенно ненужной по истечении некоторого времени. Например, программа телепередач на нынешнюю неделю будет неактуальна для многих телезрителей на следующей неделе.

Данные являются составной частью информации, представляющие собой зарегистрированные сигналы. Единица измерения информации называется бит –binary digit. В компьютерной технике бит соответствует физическому состоянию носителя информации: намагничено – не намагничено( 01 ). Последовательностью битов можно закодировать текст, изображение, звук или какую-нибудь другую информацию. Такой метод представления называется двоичным кодированием. В информатике часто используется величина, называемая байтом и равная 8 битам. Каждому символу соответствует своя последовательность из 8 нулей и единиц. Соответствие байтов и символов задается с помощью таблицы, в которой для каждого кода указывается свой символ. Например, буква М имеет код 11101101, буква И- код 11101001, а пробел- код 00100000.

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

- 1 Килобайт (Кбайт) = 1024 байт =  $2^{10}$  байт,
- 1 Мегабайт (Мбайт) = 1024 Кбайт =  $2^{20}$  байт,
- 1 Гигабайт (Гбайт) = 1024 Мбайт =  $2^{30}$  байт,
- 1 Терабайт (Тбайт) = 1024 Гбайт =  $2^{40}$  байт.

Кодирование информации. В процессе преобразования информации из одной формы представления в другую осуществляется кодирование. Средством кодирования служит таблица соответствия, которая устанавливает взаимно однозначное соответствие между знаками или группами знаков двух различных знаковых систем. В процессе обмена информацией часто приходится производить операции кодирования и декодирования. При вводе знака алфавита в компьютер путем нажатия соответствующей клавиши выполняется его кодирование, т.е. преобразование в компьютерный код. При выводе знака на экран монитора или принтер происходит обратный процесс - декодирование, когда из компьютерного кода знак преобразуется в графическое изображение.

Кодирование изображений и звука. Информация, в том числе графическая и звуковая, может быть представлена в аналоговой и дискретной форме. В аналоговом представлении физическая величина принимает бесконечное множество значений. Примером аналогового представления графической информации может служить живописное полотно, цвет которого изменяется непрерывно, а дискретной - изображение, напечатанное с помощью струйного принтера и состоящее из отдельных точек разного цвета. Пример аналогового хранения звуковой информации — виниловая пластинка (звуковая дорожка изменяет форму непрерывно), а дискретного — аудиокомпакт-диск (звуковая дорожка содержит участки с различной отражающей способностью). Компьютер может обрабатывать только числовую информацию. Вся другая информация (звуки, изображения, показания приборов и т.д.) для обработки на компьютере должна быть преобразована в числовую. Например, для музыкального звука измеряется интенсивность звука на определенных частотах через небольшие промежутки времени и результаты измерений представляются в числовой форме. Для изображений задаются координаты точек и числовые коды цветов, в которые выкрашены точки.

## Структуры данных

Работа с большим количеством данных автоматизируется проще, когда данные упорядочены. Для упорядочивания данных применяются следующие структуры: *линейные (списки), табличные, иерархические (дерево)*.

**Линейная структура.** *Линейная структура данных* (или список) — это упорядоченная структура, в которой адрес данного однозначно определяется его номером (индексом). Примером линейной структуры может быть список учебной группы или дома, стоящие на одной улице.

В списках, как правило, новый элемент начинается с новой строки. Если элементы располагаются в строчку, нужно внести разделительный знак между элементами. Поиск осуществляется по разделителям (чтобы найти, например, десятый элемент, надо отсчитать девять разделителей).

Если элементы списка одной длины, структура называется *вектором данных*, разделители не требуются. При длине одного элемента —  $d$ , зная номер элемента —  $n$ , его начало определяется соотношением  $d(n-1)$ .

**Табличная структура данных.** *Табличная структура данных* — это упорядоченная структура, в которой адрес данного однозначно определяется двумя числами — номером строки и номером столбца, на пересечении которых находится ячейка с искомым элементом.

**Иерархическая структура.** Нерегулярные данные, которые трудно представляются в виде списка или таблицы, могут быть представлены в *иерархической структуре*, в которой адрес каждого элемента определяется *путем (маршрутом доступа)*, идущим от вершины структуры к данному элементу.

## Хранение данных

Для устройств обработки данных, к которым относится и компьютер, большое значение имеет организация метода хранения информации на внешних носителях, позволяющих сохранять данные энергонезависимо. Способ хранения данных на таких носителях должен обеспечивать их целостность, доступность и защищенность. В настоящее время наиболее популярными внешними носителями являются диски. На одном диске помещается информация, объем которой может измеряться триллионами байтов. В этом случае эффективный способ хранения особенно важен. Разработчики программного обеспечения предложили оригинальный способ организации хранения информации: в виде файлов.

Под *файлом* понимается именованная область носителя, содержащая данные произвольной длины и воспринимаемая компьютерной системой как единое целое. *Имя файла* имеет особое значение, оно сопоставлено адресу размещения файла на носителе. Носитель имеет служебную таблицу, в каждой строке которой записано имя файла и адрес его местонахождения на носителе. Эта таблица используется специальной программой, которая называется *файловой системой*. Для доступа к данным она получает имя файла, находит по таблице его местоположение на носителе и возвращает содержимое файла. Как правило, процесс обработки информации сопровождается ее последующим сохранением. Для этого компьютерная программа объединяет какой-либо блок обрабатываемых данных в единое целое, снабжает его именем и передает файловой системе для записи на внешний носитель.

Кроме имени, файловая система, создавая файл, снабжает его дополнительной информацией: датой и временем создания (или модификации), размером сохраненных данных, правами доступа к информации, хранящейся в нем. Эта информация называется *атрибутами файла* и предоставляет возможности файловой системе оперативно работать с файлами.

Файл в числовом виде хранит информацию разных типов, например, текстовую, звуковую, графическую и т.д. Программа, создающая файл, размещает информацию в нем таким образом, чтобы при дальнейшей работе с файлом записанные данные можно было распознать и правильно извлечь. Способ представления данных в файле называется *форматом файла*. Формат определяет внутреннюю организацию информации, хранимой в файле. Открывая файл, прикладная программа проверяет его формат. Если он соответствует распознаваемым ею форматам, информация, хранящаяся в файле, извлекается в удобном для работы виде. Современные операционные системы автоматически распознают формат файла и самостоятельно запускают работающую с ним прикладную программу. Имеется возможность определять формат файла, не исследуя его структуры. Для

Как было уже сказано, задачу централизованного управления данными решает файловая система. Она выполняет функции распределения внешней памяти, отображения имен файлов в соответствующие адреса и обеспечения доступа к данным.

Во время информационного процесса данные преобразовываются из одного вида в другого с помощью методов. Обработка данных включает в себя множество разных операций. Основными операциями являются:

- сбор данных - накопление информации с целью обеспечения достаточной полноты для принятия решения;
- формализация данных - приведение данных, которые поступают из разных источников к единой форме;
- фильтрация данных - устранение лишних данных, которые не нужны для принятия решений;
- сортировка данных - приведение в порядок данных за заданным признаком с целью удобства использования;
- архивация данных - сохранение данных в удобной и доступной форме;
- защита данных - комплекс мер, направленных на предотвращение потерь, воспроизведения и модификации данных;
- транспортирование данных - прием и передача данных между отдаленными пользователями информационного процесса. Источник данных принято называть сервером, а потребителя - клиентом;
- преобразование данных - преобразование данных с одной формы в другую, или с одной структуры в другую, или изменение типа носителя.

**Информационная система.** В информатике понятие "система" чаще используют относительно набора технических средств и программ. Системой называют также аппаратную часть компьютера. Дополнение понятия "система" словом "информационная" отображает цель ее создания и функционирования.

Информационная система - взаимосвязанная совокупность средств, методов и персонала, используемая для сохранения, обработки и выдачи информации с целью решения конкретной задачи. Современное понимание информационной системы предусматривает использование компьютера как основного технического средства обработки информации. Компьютеры, оснащенные специализированными программными средствами, являются технической базой и инструментом информационной системы.

В работе информационной системы можно выделить следующие этапы:

1. Зарождение данных - формирование первичных сообщений, которые фиксируют результаты определенных операций, свойства объектов и субъектов управления, параметры процессов, содержание нормативных и юридических актов и т.п..
2. Накопление и систематизация данных - организация такого их размещения, которое обеспечивало бы быстрый поиск и отбор нужных сведений, методическое обновление данных, защита их от искажений, потери, деформирования целостности и др.
3. Обработка данных - процессы, вследствие которых на основании прежде накопленных данных формируются новые виды данных: обобщающие, аналитические, рекомендательные, прогнозные. Производные данные тоже можно обрабатывать, получая более обобщенные сведения.
4. Отображение данных - представление их в форме, пригодной для восприятия человеком. Прежде всего - это вывод на печать, то есть создание документов на так называемых твердых (бумажных) носителях. Широко используют построение графических иллюстративных материалов (графиков, диаграмм) и формирование звуковых сигналов.

**Информационные технологии.** В широком смысле слово технология - это способ освоения человеком материального мира с помощью социально организованной деятельности, которая включает три компоненты: информационную (научные принципы и обоснование), материальную (орудие работы) и социальную (специалисты, имеющие профессиональные навыки). Эта триада составляет сущность современного понимания понятия технологии. Понятие информационной технологии появилось с возникновением информационного общества, основой социальной динамики в котором являются не традиционные материальные, а информационные ресурсы: знания, наука, организационные факторы, интеллектуальные способности, инициатива, творчество и т.д. К сожалению, это понятие настолько общее и всеохватывающее, что до сих пор специалисты не пришли к четкой, формализованной формулировке.

Наиболее удачным определением понятия информационной технологии дано академиком Глушковым В.М., который трактовал ее как человеко-машинную технологию сбора, обработки и передачи информации, которая базируется на использовании вычислительной техники. Эта технология быстро развивается, охватывая все виды общественной деятельности: производство, управление, науку, образование, финансово-банковские операции, медицину, быт и др.

### ***Контрольные вопросы***

1. Что изучает наука информатика?
2. Что понимают под интерфейсом пользователя?
3. Что такое информация? Какими свойствами она владеет?
4. В чем заключается процесс кодирования информации
5. Из каких операций складывается процесс обработки данных?
6. Что такое информационная система?
7. Из каких этапов складывается работа информационных систем?
8. Какой смысл вкладывается в понятия "информационная технология"?

### ***Темы самостоятельных работ***

1. Представление текстовых, звуковых и графических и данных в двоичном коде.
2. Представление звуковых в двоичном коде.
3. Представление графических в двоичном коде.
4. Понятие алгоритма и его свойства.
5. Структуры данных.

### ***Список рекомендованной литературы***

1. Степанов А.Н. Информатика. Учебник для вузов. СПб.: Питер, 2006.
2. Информатика. Компьютерная техника. Компьютерные технологии. / Пособие под ред. О.И.Пушкаря. - Издательский центр "Академия", Киев, - 2001 г.
3. Коцюбинский А.О., Грошев С.В. Современный самоучитель профессиональной работы на компьютере. - Г.: Триумф, 1999 г.
4. Информатика. Базовый курс. / Под ред. С.В.Симоновича. – СПб:Питер,2000.
5. Могилев А.В. , Пак Н.И., Хеннер Е.К. Информатика. М.:ACADEMIA, 1999.
6. Шафрин Ю.А. Азбука компьютерных технологий. М.: Изд-во института психотерапии, 2000.
7. Соболев Б.В., Галин А.Б., Панов Ю.В. и др. Информатика. Учебник. Ростов-на-Дону:Феникс, 2007.
8. Волков В.В. Работа на персональном компьютере. Практический курс. Киев: ЮНИОР,1999.

[На оглавление](#)

## **Лекция 2**

### **Аппаратные средства персонального компьютера**

**Цель лекции:** ознакомить с основными аппаратными средствами персонального компьютера.

**Ключевые слова:** логическая структура ПК, процессор, тактовая частота, внутренняя память, винчестер, оптические диски, флэш-память, клавиатура, манипулятор «мышь», монитор, разрешающая способность, принтеры, сканеры, модемы

## Содержание

Понятие архитектуры компьютера.

Логическая структура компьютера.

Классификация компьютеров.

Процессор. Виды памяти.

Стандартные устройства ввода-вывода.

Периферийные устройства ввода-вывода.

Под **архитектурой компьютера** понимается совокупность сведений об основных устройствах компьютера и их назначении, о способах представления программ и данных в машине, об особенностях ее организации и функционирования.

**Электронная вычислительная машина (ЭВМ), или компьютер** — это электронное устройство, используемое для автоматизации процессов приема, хранения, обработки и передачи информации, которые осуществляются по заранее разработанным человеком алгоритмам (программам).

Еще раз обращаем внимание на важнейшие моменты этого определения:

- ✎ компьютер представляет собой электронное устройство;
- ✎ компьютер выполняет действия без вмешательства человека — автоматически;
- ✎ компьютеру должен быть заранее задан разработанный человеком и записанный в специальной форме план действий — программа.

Логическая структура компьютера.

- устройство ввода,
- центральный процессор,
- запоминающее устройство,
- устройство вывода.

Все эти блоки состоят из отдельных меньших устройств. В частности, в центральный процессор могут входить арифметико-логическое устройство (АЛУ), внутреннее запоминающее устройство в виде регистров процессора и внутренней кэш-памяти, управляющее устройство (УУ).

Устройство ввода, как правило, тоже не является одной конструктивной единицей. Поскольку виды входной информации разнообразны, источников ввода данных может быть несколько. Это касается и устройств вывода.

источников ввода данных может быть несколько. Это касается и устройств вывода.

**Запоминающее устройство** - это блок ЭВМ, предназначенный для временного (оперативная память) и продолжительного (постоянная память) хранения программ, входных и результирующих данных, а также промежуточных результатов. Информация в оперативной памяти сохраняется временно лишь при включенном питании, но оперативная память имеет

большее быстродействие. В постоянной памяти данные могут сохраняться даже при отключенном компьютере, но скорость обмена данными между постоянной памятью и центральным процессором, в подавляющем большинстве случаев, значительно меньше.

**Арифметико-логическое устройство** - это блок ЭВМ, в котором происходит преобразование данных по командам программы: арифметические действия над числами, преобразование кодов и др.

**Управляющее устройство** координирует работу всех блоков компьютера. В определенной последовательности он выбирает из оперативной памяти команду за командой. Каждая команда декодируется, по потребности элементы данных из указанных в команде ячеек оперативной памяти передаются в АЛУ; АЛУ настраивается на выполнение действия, указанной текущей командой (в этом действии могут принимать участие также устройства ввода-вывода); дается команда на выполнение этого действия. Этот процесс будет продолжаться до тех пор, пока не возникнет одна из следующих ситуаций: исчерпаны входные данные, от одного из устройств поступила команда на прекращение работы, выключено питание компьютера.

Реальная структура компьютера значительно сложнее, чем рассмотренная выше

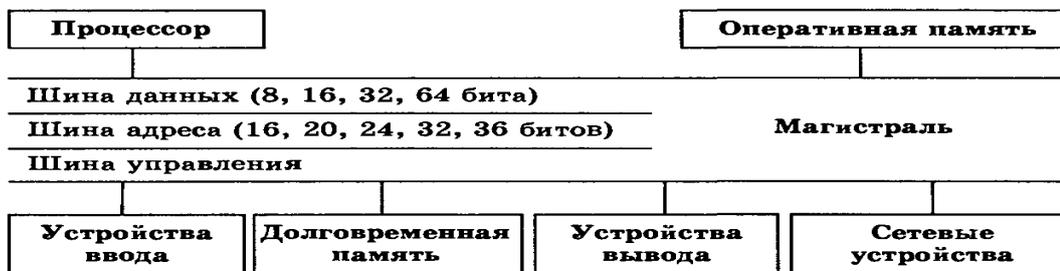


Рис. 1.1. Магистрально-модульное устройство компьютера

Различные устройства компьютера подсоединяют друг к другу с помощью стандартизированных и унифицированных аппаратных средств – кабелей, разъемов и т. д. При этом устройства обмениваются друг с другом информацией и управляющими сигналами, которые также приводятся к некоторым стандартным формам. Совокупность этих стандартных средств и форм образует конкретный **интерфейс** того или иного устройства или компьютера в целом.

Заметим, что для обозначения совокупности устройств, которые могут быть включены в состав компьютера той или иной модели, а также средств их соединения используется термин **аппаратное обеспечение**.

### Классификация компьютеров

Номенклатура видов компьютеров сегодня огромная: машины различаются по назначению, мощности, размерам, элементной базе и т.д. Поэтому классифицируют ЭВМ по разным признакам. Следует заметить, что любая классификация является в некоторой мере условной, поскольку развитие компьютерной науки и техники настолько бурное, что, например, сегодняшняя микроЭВМ не уступает по мощности миниЭВМ пятилетней давности и даже суперкомпьютерам недавнего прошлого. Кроме того, зачисление компьютеров к определенному классу довольно условно через нечеткость разделения групп, так и вследствие внедрения в практику заказной сборки компьютеров, где номенклатуру узлов и конкретные модели

адаптируют к требованиям заказчика. Рассмотрим распространенные критерии классификации компьютеров.

#### Классификация по назначению

- большие электронно-вычислительные машины (ЭВМ);
- миниЭВМ;
- микроЭВМ;
- персональные компьютеры.

**Большие ЭВМ (Main Frame)** . Применяют для обслуживания крупных областей народного хозяйства. Они характеризуются 64-разрядными параллельно работающими процессорами (количество которых достигает до 100), интегральным быстродействием до десятков миллиардов операций в секунду, многопользовательским режимом работы. Доминирующее положение в выпуске компьютеров такого класса занимает фирма IBM (США). Наиболее известными моделями суперЭВМ являются: IBM 360, IBM 370, IBM ES/9000, Cray 3, Cray 4, VAX-100, Hitachi, Fujitsu VP2000. На базе больших ЭВМ создают вычислительный центр, который содержит несколько отделов или групп . Штат обслуживания - десятки людей.

**МиниЭВМ.** Похожа на большие ЭВМ, но меньших размеров. Используют на крупных предприятиях, научных учреждениях и организациях. Часто используют для управления производственными процессами. Характеризуются мультипроцессорной архитектурой, подключением до 200 терминалов, дисковыми запоминающими устройствами, которые наращиваются до сотен гигабайт, разветвленной периферией. Для организации работы с миниЭВМ, нужен вычислительный центр, но меньший чем для больших ЭВМ.

**МикроЭВМ.** Доступны многим учреждениям. Для обслуживания достаточно вычислительной лаборатории в составе нескольких человек, с наличием прикладных программистов. Необходимые системные программы покупаются вместе с микроЭВМ, разработку прикладных программ заказывают в больших вычислительных центрах или специализированных организациях. Программисты вычислительной лаборатории занимаются внедрением приобретенного или заказанного программного обеспечения, выполняют его настройку и согласовывают его работу с другими программами и устройствами компьютера. Могут вносить изменения в отдельные фрагменты программного и системного обеспечения.

**Персональные компьютеры.** Бурное развитие приобрели в последние 20 лет. Персональный компьютер (ПК) предназначен для обслуживания одного рабочего места и способен удовлетворить потребности малых предприятий и отдельных лиц. С появлением Интернета популярность ПК значительно возросла, поскольку с помощью персонального компьютера можно пользоваться научной, справочной, учебной и развлекательной информацией. Персональные компьютеры условно можно поделить на профессиональные и бытовые, но в связи с удешевлением аппаратного обеспечения, грань между ними размывается. С 1999 года введен международный сертификационный стандарт - спецификация PC99:

- массовый персональный компьютер (Consumer PC)
- деловой персональный компьютер (Office PC)
- портативный персональный компьютер (Mobile PC)
- рабочая станция (WorkStation)
- развлекательный персональный компьютер (Entertainment PC)

Большинство персональных компьютеров на рынке подпадают до категории массовых ПК. Деловые ПК - имеют минимум средств воспроизведения графики и звука. Портативные ПК

отличаются наличием средств коммуникации отдаленного доступа (компьютерная связь). Рабочие станции - увеличенные требования к устройствам хранения данных. Развлекательные ПК - основной акцент на средствах воспроизведения графики и звука.

### Классификация по размеру

- настольные (desktop);
- портативные (notebook);
- карманные (palmtop).

Наиболее распространенными являются настольные ПК, которые позволяют легко изменять конфигурацию. Портативные удобны для пользования, имеют средства компьютерной связи. Карманные модели можно назвать "интеллектуальными" записными книжками, разрешают хранить оперативные данные и получать к ним быстрый доступ.

**Системный блок** - основная составляющая ПК, в середине которой находятся важнейшие компоненты. Устройства, находящиеся в середине системного блока называют внутренними, а устройства, подсоединенные извне называют внешними. Внешние дополнительные устройства, предназначенные для ввода и вывода информации называются также периферийными.

**Основной платой ПК является материнская плата (MotherBoard). На ней расположены:**

- **процессор** - основная микросхема, выполняющая математические и логические операции;
- **чипсет** (микропроцессорный комплект) - набор микросхем, которые руководят работой внутренних устройств ПК и определяют основные функциональные возможности материнской платы;
- **шины** - набор проводников, по которым происходит обмен сигналами между внутренними устройствами компьютера;
- **оперативное запоминающее устройство (ОЗУ)** - набор микросхем, предназначенных для временного сохранения данных, пока включен компьютер;
- **постоянное запоминающее устройство (ПЗУ)** - микросхема, предназначенная для долговременного хранения данных, даже при отключенном компьютере;
- **разъемы** для подсоединения дополнительных устройств (слоты).

### Процессор

**Процессор** - это главная микросхема компьютера, его "мозг". Он разрешает выполнять программный код, находящийся в памяти и руководит работой всех устройств компьютера. Скорость его работы определяет быстродействие компьютера. Конструктивно, процессор - это кристалл кремния очень маленьких размеров. Процессор имеет специальные ячейки, которые называются регистрами. Именно в регистрах помещаются команды, которые выполняются процессором, а также данные, которыми оперируют команды. Работа процессора состоит в выборе из памяти в определенной последовательности команд и данных и их выполнении. На этом и базируется выполнение программ.

В ПК обязательно должен присутствовать центральный процессор (Central Rprocessing Unit - CPU), который выполняет все основные операции. Часто ПК оснащен дополнительными сопроцессорами, ориентированными на эффективное выполнение специфических функций, такие как, математический сопроцессор для обработки числовых данных в формате с плавающей точкой, графический сопроцессор для обработки графических изображений, сопроцессор ввода/вывода для выполнения операции взаимодействия с периферийными устройствами.

## Основными параметрами процессоров являются:

- тактовая частота,
- разрядность,
- рабочее напряжение,
- коэффициент внутреннего умножения тактовой частоты,
- размер кеш памяти.

**Тактовая частота** определяет количество элементарных операций (тактов), выполняемые процессором за единицу времени. Тактовая частота современных процессоров измеряется в МГц (1 Гц соответствует выполнению одной операции за одну секунду, 1 МГц=106 Гц). Чем больше тактовая частота, тем больше команд может выполнить процессор, и тем больше его производительность. Первые процессоры, которые использовались в ПК работали на частоте 4,77 МГц, сегодня рабочие частоты современных процессоров достигают отметки в 2 ГГц (1 ГГц = 103 МГц).

**Разрядность** процессора показывает, сколько бит данных он может принять и обработать в своих регистрах за один такт. Разрядность процессора определяется раз

**Шины.** С другими устройствами, и в первую очередь с оперативной памятью, процессор связан группами проводников, которые называются шинами. Основных шин три:

- шина данных,
- адресная шина,
- командная шина.

**Адресная шина.** Данные, которые передаются по этой шине трактуются как адреса ячеек оперативной памяти. Именно из этой шины процессор считывает адреса команд, которые необходимо выполнить, а также данные, с которыми оперируют команды. В современных процессорах адресная шина 32-разрядная, то есть она состоит из 32 параллельных проводников.

**Шина данных.** По этой шине происходит копирование данных из оперативной памяти в регистры процессора и наоборот. В ПК на базе процессоров Intel Pentium шина данных 64-разрядная. Это означает, что за один такт на обработку поступает сразу 8 байт данных.

**Командная шина.** По этой шине из оперативной памяти поступают команды, выполняемые процессором. Команды представлены в виде байтов. Простые команды вкладываются в один байт, но есть и такие команды, для которых нужно два, три и больше байта. Большинство современных процессоров имеют 32-разрядную командную шину, хотя существуют 64-разрядные процессоры с командной шиной.

Шины на материнской плате используются не только для связи с процессором. Все другие внутренние устройства материнской платы, а также устройства, которые подключаются к ней, взаимодействуют между собой с помощью шин. От архитектуры этих элементов во многом зависит производительность ПК в целом.

## Виды памяти

Оперативная память

Процессор компьютера *имеет непосредственный доступ* ко всей информации, которая находится в оперативной памяти и именно поэтому программы, находящиеся в оперативной памяти, могут быть выполнены процессором, а данные, находящиеся в оперативной памяти, могут быть по этим программам обработаны.

Из определения следует, что в оперативной памяти на *стадии выполнения* могут *одновременно* находиться несколько программ. Кроме того, в оперативной памяти могут находиться как обрабатываемые, так и уже обработанные программой данные.

Можно считать, что оперативная память представляет собой последовательность пронумерованных байтов. Каждый байт имеет свой собственный номер, который по аналогии с номерами домов на улице принято называть **адресом**. Содержимое любого байта памяти может обрабатываться независимым от остальных байтов образом. Указав адрес байта, можно прочесть код, который в нем записан, или занести, записать в этот байт какой-либо другой код. Поэтому оперативную память называют еще **прямоадресуемой** памятью, **памятью с прямым доступом**, и обозначают **RAM** (Random Access Memory — память произвольного доступа).

## Кэш-память

Кроме оперативной памяти в состав персонального компьютера входит родственная ей **кэш-память**, или просто **кэш** (cache — запас, тайный склад или наличные, карманные деньги, то есть деньги, которые всегда «под рукой»). Это сверхбыстрая память относительно небольшого объема — 128–512 Кбайт (в последних по времени разработках до 1–2 Мбайт). Иногда ее называют **сверхоперативной** памятью. По структуре и принципу работы кэш ничем не отличается от оперативной памяти. Однако скорость передачи данных при обмене с кэшем значительно выше, чем при обмене с оперативной памятью, но и стоит она дороже. Кэш используется как промежуточное звено между процессором и оперативной памятью, которое обеспечивает *повышение скорости вычислений*. Дело в том, что процессор обрабатывает данные с очень большой скоростью, которая намного превышает скорость считывания и записи в оперативную память. Поэтому при совместной работе процессор будет простаивать, подстраиваясь под скорость оперативной памяти. Чтобы избежать этого, как раз и вводится промежуточный, скоростной уровень памяти — кэш, который обеспечивает значительное сглаживание разницы в скоростях работы процессора и оперативной памяти. В тот момент, когда процессору потребуются данные для обработки, он обращается не в «далекую» оперативную память, а в кэш-память, которая находится «под рукой», поближе. В современных машинах предусматривается до трех уровней кэш-памяти.

## Постоянная память

Можно упомянуть и еще один вид памяти компьютера — **постоянную память**, **ПЗУ** (постоянное запоминающее устройство), или **ROM** (Read Only Memory — память только для чтения). Эта память отличается от оперативной тем, что запись информации в ПЗУ осуществляется только один раз на заводе-изготовителе. И в дальнейшем из этой памяти возможно только чтение. Кроме того, при отключении электропитания данные, записанные в ПЗУ, сохраняются. Постоянная память используется для хранения наиболее важных и часто используемых *служебных* программ, которые осуществляют проверку работы отдельных устройств компьютера (тестирование), а также выполняют постоянно используемые операции по обмену данными между клавиатурой, монитором и памятью компьютера. Этот комплекс программ образует **базовую систему ввода-вывода** или сокращенно **BIOS** (Base Input Output System — базовая ввода-вывода система).

## Энергонезависимая память CMOS.

Работа таких стандартных устройств, как клавиатура, может обслуживаться программами BIOS, но такими средствами невозможно обеспечить работу со всеми возможными устройствами (в связи с их огромным разнообразием и наличием большого количества разных параметров). Но

для своей работы программы BIOS требуют всю информацию о текущей конфигурации системы. По очевидной причине эту информацию нельзя сохранять ни в оперативной памяти, ни в постоянной. Специально для этих целей на материнской плате есть микросхема энергонезависимой памяти, которая называется CMOS. От оперативной памяти она отличается тем, что ее содержимое не исчезает при отключении компьютера, а от постоянной памяти она отличается тем, что данные можно заносить туда и изменять самостоятельно, в соответствии с тем, какое оборудование входит в состав системы.

Микросхема памяти CMOS постоянно питается от небольшой батарейки, расположенной на материнской плате. В этой памяти сохраняются данные про гибкие и жесткие диски, процессоры и т.д. Тот факт, что компьютер четко отслеживает дату и время, также связан с тем, что эта информация постоянно хранится (и обновляется) в памяти CMOS. Таким образом, программы BIOS считывают данные о составе компьютерной системы из микросхемы CMOS, после чего они могут осуществлять обращение к жесткому диску и другим устройствам.

## Внешняя память

---

**Внешней памятью** называется группа устройств, которые предназначены для долговременного хранения больших массивов информации — программ и данных.

Внешнюю память компьютера, или **ВЗУ (внешние запоминающие устройства)**, можно представлять себе как значительный по объему *информационный склад*, где программы и данные могут храниться годами до тех пор, пока они не потребуются. Несмотря на то что фактически эти устройства находятся внутри корпуса персонального компьютера, для их обозначения используется термин «внешняя память», так как это сложилось исторически.

Процессор, то есть устройство, обеспечивающее задаваемую программой обработку данных, *не имеет непосредственного доступа* к внешней памяти. Поэтому программа, находящаяся во внешней памяти, не может в ней выполняться, а данные не могут быть каким-либо образом обработаны. В этом и состоит самое главное функциональное отличие внешней памяти от оперативной. Во внешней памяти программы и данные хранятся в «нерабочем состоянии», а в оперативной программы и данные хранятся во время выполнения (и только во время выполнения) программ. Для того чтобы выполнить какую бы то ни было программу, ее сначала нужно «взять со склада» — найти на внешнем устройстве и перенести в оперативную память, где она и сможет выполняться. Аналогичным образом, чтобы обработать данные, физически находящиеся во внешней памяти, их нужно сначала перенести в оперативную память.

Самыми распространенными являются накопители на магнитных дисках, которые делятся на **накопители на жестких магнитных дисках (НЖМД)** и **накопители на оптических дисках**, такие как накопители CD-ROM, CD-R, CD-RW и DVD-ROM.

**Накопители на жестких магнитных дисках (НЖМД)**. НЖМД - это основное устройство для долговременного хранения больших объемов данных и программ. Другие названия: жесткий диск, винчестер, HDD (Hard Disk Drive). Внешне, винчестер представляет собой плоскую, герметически закрытую коробку, внутри которой на общей оси находятся несколько жестких алюминиевых или стеклянных пластинок круглой формы. Поверхность любого из дисков покрыта тонким ферромагнитным слоем (вещество, которое реагирует на внешнее магнитное поле), собственно на нем хранятся записанные данные. При этом запись проводится

на обе поверхности каждой пластины (кроме крайних) с помощью блока специальных магнитных головок. Каждая головка находится над рабочей поверхностью диска на расстоянии 0,5-0,13 мкм. Пакет дисков вращается непрерывно и с большой частотой (4500-10000 об/мин), поэтому механический контакт головок и дисков недопустим. производительность современных жестких дисков.

**Среди других параметров, которые влияют на быстродействие HDD следует отметить следующие:**

- **скорость обращения дисков** - в наше время выпускаются накопители EIDE с частотой обращения 4500-7200 об/мин, и накопители SCSI - 7500-10000 об/мин;
- **емкость кэш-памяти** - во всех современных дисковых накопителях устанавливается кэш-буфер, ускоряющий обмен данными; чем больше его емкость, тем выше вероятность того, что в кэш-памяти будет необходимая информация, которую не надо считывать с диска (этот процесс в тысячи раз медленней); емкость кэш-буфера в разных устройствах может изменяться в границах от 64 Кбайт до 2Мбайт;
- **среднее время доступа** - время (в миллисекундах), на протяжении которого блок головок смещается с одного цилиндра на другой. Зависит от конструкции привода головок и составляет приблизительно 10-13 миллисекунд;
- **время задержки** - это время от момента позиционирования блока головок на нужный цилиндр до позиционирования конкретной головки на конкретный сектор, другими словами, это время поиска нужного сектора;

## **Оптические диски**

**Основные характеристики CD-ROM:**

- скорость передачи данных - измеряется в кратных долях скорости проигрывателя аудио компакт-дисков (150 Кбайт/сек) и характеризует максимальную скорость с которой накопитель пересылает данные в оперативную память компьютера, например, 2-скоростной CD-ROM (2x CD-ROM) будет считывать данные с скоростью 300 Кбайт/сек., 50-скоростной (50x) - 7500 Кбайт/сек.;
- время доступа - время, нужное для поиска информации на диске, измеряется в миллисекундах.

Основной недостаток стандартных CD-ROM - невозможность записывания данных, но существуют устройства однократной записи CD-R и многократной записи CD-RW.

**Накопитель CD-R (CD-Recordable).** Внешне похожи на накопители CD-ROM и совместимые с ними по размерам дисков и форматам записи. Позволяют выполнить однократную запись и неограниченное количество считываний. Запись данных осуществляется с помощью специального программного обеспечения. Скорость записи современных накопителей CD-R составляет 4x-8x.

**Накопитель CD-RW (CD-ReWritable).** Используются для многократной записи данных, причем можно как просто дописать новую информацию на свободное пространство, так и полностью перезаписать диск новой информацией (предыдущие данные уничтожаются). Как и в случае с накопителями CD-R, для записи данных необходимо установить в системе специальные программы, причем формат записи совместимый с обычным CD-ROM. Скорость записи современных накопителей CD-RW составляет 2x-4x.

**Накопитель DVD (Digital Video Disk).** Устройство для чтения цифровых видеозаписей. Внешне DVD-диск похож на обычный CD-ROM (диаметр - 120 мм, толщина 1,2 мм), однако отличается от него тем, что на одной стороне DVD-диска может быть записано до 4,7 Гбайт, а на двух - до 9,4 Гбайт. В случае использования двухслойной схемы записи на одной стороне можно разместить уже до 8,5 Гбайт информации, соответственно на двух сторонах - около 17 Гбайт. DVD-диски допускают перезапись информации.

Важнейшим фактором, сдерживающим широкое применение накопителей CD-R, CD-RW и DVD, является высокая стоимость как их самих, так и сменных носителей.

Процесс взаимодействия пользователя с персональным компьютером (ПК) непременно включает процедуры ввода входных данных и получение результатов обработки этих данных. Поэтому, обязательными составляющими типичной конфигурации ПК являются разнообразные устройства ввода-вывода, среди которых можно выделить стандартные устройства, без которых современный процесс диалога вообще невозможен, и периферийные, т.е. дополнительные. К стандартным устройствам ввода-вывода относятся монитор, клавиатура и манипулятор "мышка".

монитор, клавиатура и манипулятор "мышка".

**Flash-память.** Flash-память — это энергонезависимый тип памяти, позволяющий записывать и хранить данные в микросхемах. Карты flash-памяти (рис. 1.8) не имеют в своем составе движущихся частей, что обеспечивает высокую сохранность данных при их использовании в мобильных устройствах (портативных компьютерах, цифровых камерах и др.).



Рис. 1.8. Карты flash-памяти

## Стандартные устройства ввода-вывода

**Мониторы.** Первые компьютеры мониторов не имели, был лишь набор мигающих светодиодов и распечатка результатов на принтере. С развитием компьютерной техники появились мониторы и сейчас они являются необходимой частью базовой конфигурации персонального компьютера.

**Монитор** (дисплей) - это стандартное устройство вывода, предназначенное для визуального отображения текстовых и графических данных. В зависимости от принципа действия, мониторы делятся на:

- мониторы с электронно-лучевой трубкой;
- дисплеи на жидких кристаллах.

**Основные параметры мониторов.** С точки зрения пользователя, основными характеристиками монитора являются размер по диагонали, разрешающая способность, частота регенерации (обновление) и класс защиты.

**Размер монитора.** Экран монитора измеряется по диагонали в дюймах. Размеры колеблются от 9 дюймов (23 см) до 42 дюймов (106 см). Чем больше экран, тем дороже монитор. Распространенными являются размеры 14, 15, 17, 19 и 21 дюйма. Мониторы большого размера лучше использовать для настольных издательских систем и графических работ, в которых нужно видеть все детали изображения. Оптимальными для массового использования являются 15- и 17-дюймовые мониторы.

**Разрешающая способность.** В графическом режиме работы изображение на экране монитора состоит из точек (пикселей). Количество точек по горизонтали и вертикали, которые монитор способен воссоздать четко и отдельно называется его разрешающей способностью. Выражение "разрешающая способность 800x600" означает, что монитор может выводить 600 горизонтальных строк по 800 точек в каждой. Стандартными являются такие режимы разрешающей способности: 800x600, 1024x768, 1152x864 и выше. Это свойство монитора определяется размером точки (зерна) экрана. Размер зерна экрана современных мониторов не превышает 0,28 мм. Чем больше разрешающая способность, тем лучше качество изображения. Качество изображения также связано с размером экрана. Так, для удовлетворительного качества изображения в режиме 800x600 на 15-дюймовом мониторе можно ограничиться размером зерна 0,28 мм,

**Видеоадаптер.** Работой монитора руководит специальная плата, которую называют видеоадаптером (видеокартой). Вместе с монитором видеокарта создает видеоподсистему персонального компьютера. В первых компьютерах видеокарты не было. В оперативной памяти существовал экранный участок памяти, куда процессор заносил данные об изображении. Контроллер экрана считывал данные об яркости отдельных точек экрана из ячеек памяти и руководил разверткой горизонтального луча электронной пушки монитора.

При переходе от монохромных мониторов к цветным и с увеличением разрешающей способности экрана, участка видеопамати стало недостаточно для хранения графических данных, а процессор не успевал обрабатывать изображения. Все операции, связанные с управлением экрана были отведены в отдельный блок - видеоадаптер.

Видеоадаптер имеет вид отдельной платы расширения, которую вставляют в определенный слот материнской платы (в современных ПК это слот AGP). Видеоадаптер выполняет функции видеоконтроллера, видеопроцессора и видеопамати.

**Клавиатура.** Клавиатура - это стандартное клавишное устройство ввода, предназначенное для ввода алфавитно-цифровых данных и команд управления. Комбинация монитора и клавиатуры обеспечивает простейший интерфейс пользователя: с помощью клавиатуры руководят компьютерной системой, а с помощью монитора получают результат.

Клавиатура относится к стандартным средствам ПК, поэтому для реализации ее основных функций не требуется наличие специальных системных программ (драйверов). Необходимое программное обеспечение для работы с клавиатурой находится в микросхеме постоянной памяти в составе базовой системы ввода-вывода BIOS. Именно поэтому, ПК реагирует на нажатие клавиш на клавиатуре сразу после включения. Клавиатуры имеют по 101-104 клавише, размещенных по стандарту QWERTY (в верхнем левом углу алфавитной части клавиатуры находятся клавиши Q, W, E, R, T, Y). Отличаются они лишь незначительными вариантами расположения и формой служебных клавиш, а также особенностями, обусловленными используемым языком.

Набор клавиш клавиатуры разбит на несколько функциональных групп:

- алфавитно-цифровые;
- функциональные;
- управления курсором;
- служебные;

- клавиши дополнительной панели.

Основное назначение **алфавитно-цифровых клавиш** - ввод знаковой информации и команд, которые набираются по буквам. Каждая клавиша может работать в двух режимах (регистрах) и, соответственно, может использоваться для ввода нескольких символов. Переключение между нижним регистром (ввод маленьких символов) и верхним регистром (ввод больших символов) осуществляется при нажатии клавиши <SHIFT> (нефиксированное переключение) или с помощью клавиши <CAPS LOCK> (фиксированное переключение).

Группа **функциональных клавиш** включает двенадцать клавиш, обозначенных от F1 к F12, и расположена в верхней части клавиатуры. Функции этих клавиш зависят от конкретной, работающей в данный момент времени программы, а в некоторых случаях и от операционной системы. Жесткого закрепленного значения клавиш нет.

**Клавиши управления курсором** подают команды на передвижение курсора по экрану монитора относительно текущего изображения. Курсором называется экранный элемент, указывающий на место ввода знаковой информации. Эти клавиши разрешают руководить позицией ввода данных. Конкретное значение клавиш управления курсором может зависеть от программы. Тем не менее, чаще всего клавиши с стрелками служат для перемещения курсора в направлении указанном стрелкой или прокручивании текста по экрану, клавиши <Page Up> и <Page Down> прокручивают текст сразу на страницу вверх или вниз, соответственно, клавиша <Home> устанавливает курсор на начало строки, а клавиша <End> - на конец.

**Служебные клавиши** используются для разных вспомогательных целей, таких как, изменение регистра, режимов вставки, образование комбинаций "горячих" клавиш и т.д. К этой группе относятся такие клавиши, как <SHIFT>, <CAPS LOCK>, <Enter>, <Ctrl>, <Alt>, <Esc>, <Del>, <Insert>, <Tab>, <BackSpace> и прочие.

Группа **клавиш дополнительной панели** дублирует действие цифровых клавиш, клавиш управления курсором и некоторых служебных клавиш. Основное назначение - ввод чисел, поэтому клавиши размещены в порядке, удобном для такой работы. Переход в режим дублирования клавиш управления курсором и, наоборот, осуществляется нажатием на клавишу <Num Lock>. Кроме этого, клавиши дополнительной панели используются для ввода символов, имеющих расширенный код ASCII, но не имеющих соответствующей клавиши на клавиатуре.

**Манипулятор "мышка"**. Мышка - это устройство управления манипуляторного типа. Она имеет вид небольшой пластмассовой коробочки с двумя (или тремя) клавишами. Перемещение мышки по поверхности синхронизировано с перемещением графического объекта, который называется курсор мышки, по экрану монитора. В отличие от клавиатуры, мышка не является стандартным устройством управления, поэтому для работы с ней требуется наличие специальной системной программы - драйвера мышки. Драйвер мышки предназначен для интерпретации сигналов, поступающих от мышки, а также для обеспечения механизма передачи информации о положении и состоянии мышки операционной системе и другим прикладным программам. Драйвер мышки устанавливается при первом подключении мышки или при загрузке операционной

К числу параметров мышки, которыми может настроить пользователь, относят: чувствительность (характеризует величину перемещения курсора мышки по экрану при заданном перемещении мышки), функции левой и правой клавиш, а также чувствительность к двойному клику (определяет максимальный промежуток времени, на протяжении которого два отдельных клика клавиши рассматриваются как один двойной клик).

## Периферийные устройства ввода-вывода

Периферийными или внешними устройствами называют устройства, размещенные вне системного блока и задействованные на определенном этапе обработки информации. Прежде всего - это устройства фиксации выходных результатов: принтеры, плоттеры, модемы, сканеры и т.д. Понятие "периферийные устройства" довольно условное. К их числу можно отнести, например, накопитель на компакт-дисках, если он выполнен в виде самостоятельного блока и соединен специальным кабелем к внешнему разъему системного блока. И наоборот, модем может быть внутренним, то есть конструктивно выполненным как плата расширения, и тогда нет оснований относить его к периферийным устройствам.

**Принтеры.** Принтеры предназначены для вывода информации на твердые носители, большей частью на бумагу. Существует большое количество разнообразных моделей принтеров, которые различаются по принципу действия, интерфейсу, производительности и функциональным возможностям. По принципу действия различают: матричные, струйные и лазерные принтеры.

### Основные характеристики лазерных принтеров:

- **Скорость печатания.** Определяется скоростью механического протягивания листа и скоростью обработки данных, поступающих с компьютера. Средняя скорость печати 4-16 страниц за минуту.
- **Разрешающая способность.** В современных лазерных принтерах достигает 2400 dpi. Стандартным считается значение в 300 dpi.
- **Память.** Работа лазерного принтера связана с огромными вычислениями. Например, при разрешающей способности 300 dpi, на странице формата А4 будет почти 9 млн. точек, и нужно рассчитать координаты каждой из них. Скорость обработки информации зависит от тактовой частоты процессора и объема оперативной памяти принтера. Объем оперативной памяти черно-белого лазерного принтера составляет не меньше 1 Мбайт, в цветных лазерных принтерах значительно больше.

**Сканеры.** Сканер - это устройство, позволяющее вводить в компьютер черно-белое или цветное изображение, считывать графическую и текстовую информацию. Сканер используют в случае, когда возникает потребность ввести в компьютер из имеющегося оригинала текст и/или графическое изображение для его дальнейшей обработки (редактирование и т.д.). Ввод такой информации с помощью стандартных устройств ввода требует много времени. Сканированная информация после обрабатывается с помощью специального программного обеспечения (например, программой FineReader) и сохраняется в виде текстового или графического файла.

### Основные характеристики сканеров.

**Разрешающая способность.** Сканер рассматривает любой объект как набор отдельных точек (пикселей). Плотность пикселей (количество на единицу площади) называется разрешающей способностью сканера и измеряется в dpi (dots per inch - точек на дюйм). Пиксели располагаются строками, образуя изображение. Процесс сканирования происходит по строкам, вся строка сканируется одновременно. Обычная разрешающая способность сканера составляет 200-720 dpi.

**Глубина представления цветов.** При преобразовании оригинала в цифровую форму, сохраняются данные о любом пикселе изображения. Простые сканеры определяют наличие или отсутствие цвета, результирующее изображение будет черно-белым. Для представления пикселей достаточно одного разряда (0 или 1). Для передачи оттенков серого между черным и белым цветом необходимо как минимум 4 разряда (16 оттенков) или 8 разрядов (256 оттенков). Чем больше разрядов, тем качественней передаются цвета.

Большинство современных цветных сканеров поддерживает глубину цвета 24 разряда. Соответственно, сканер разрешает распознавать около 16 млн. цветов и можно качественно сканировать фотографии. На рынке сканеров есть модели, которые имеют глубину представления цвета 30 и 34 разряда.

**Область сканирования.** Максимальный размер сканируемого изображения. Ручные сканеры - до 105 мм, барабанные, планшетные сканеры - от формата А4 до Full Legar (8.5'x14').

**Модемы. Модем** - это устройство, предназначенное для подсоединения компьютера к обычной телефонной линии. Название происходит от сокращения двух слов - Модуляция и Демодуляция.

Компьютер вырабатывает дискретные электрические сигналы (последовательности двоичных нулей и единиц), а по телефонным линиям информация передается в аналоговой форме (то есть в виде сигнала, уровень которого изменяется непрерывно, а не дискретно). Модемы выполняют цифро-аналоговое и аналого-цифровое преобразования. При передаче данных, модемы накладывают цифровые сигналы компьютера на непрерывную частоту телефонной линии (модулируют ее), а при их приеме демодулируют информацию и передают ее в цифровой форме в компьютер. Модемы передают данные по обычным, то есть комутированным, телефонным каналам со скоростью от 300 до 56 000 бит в секунду, а по арендованным (выделенным) каналам скорость может быть и выше. Кроме того, современные модемы осуществляют сжатие данных перед отправлением, и соответственно, реальная скорость может превышать максимальную скорость модема.

По конструктивному выполнению модемы бывают встроенными (вставляются в системный блок компьютера в один из слотов расширения) и внешними (подключаются через один из коммуникационных портов, имеют отдельный корпус и собственный блок питания). Однако, без соответствующего коммуникационного программного обеспечения, важнейшей составляющей которого является протокол, модемы не могут работать. Наиболее распространенными протоколами модемов являются v.32 bis, v.34, v.42 bis и прочие.

Современные модемы для широкого круга пользователей имеют встроенные возможности отправления и получения факсимильных сообщений. Такие устройства называются факсами-модемами. Также, есть возможность поддержки языковых функций, с помощью звукового адаптера.

На выбор типа модема влияют следующие факторы:

- цена: внешние модемы стоят дороже, поскольку в цену входит стоимость корпуса и источника питания;
- наличие свободных портов/слотов: внешний модем подсоединяется к последовательному порту. Внутренний модем к слоту на материнской плате. Если порты или слоты заняты, нужно выбрать одно из устройств;
- удобство пользования: на корпусе внешнего модема имеются индикаторы, отображающие его состояние, а также выключатель источника питания. Для установки внешнего модема не нужно разбирать корпус компьютера.

### ***Контрольные вопросы***

1. Что такое архитектура компьютера? Общая структура компьютера?
2. Принципы современной архитектуры компьютера?
3. Методы классификации компьютеров

4. Что такое материнская плата? Какие компоненты персонального компьютера на ней находятся?
5. В чем состоит выполнение программ центральным процессором?
6. Что такое кэш-память?
7. Чем отличается оперативная память от постоянной памяти?
8. В какой памяти сохраняются программы BIOS?
9. Что такое внешняя память? Какие разновидности внешней памяти вы знаете?
10. Что такое жесткий диск? Для чего он предназначен? Какую емкость имеют современные винчестеры?
11. Какие параметры влияют на быстродействие винчестера?
12. Какие вы знаете разновидности накопителей на оптических дисках? Чем они различаются между собою?
13. В чем измеряется скорость передачи данных в накопителях на оптических носителях?

### ***Темы самостоятельных работ***

1. Классификация компьютеров.
- 2 Работа с компакт-дисками. Дисковые утилиты. Драйверы.
- 3 Энергонезависимая память CMOS.
- 4 Форматирование магнитных дисков.
- 5 Порты персонального компьютера.
- 6 Перспективы развития технических средств обработки информации.

### ***Список рекомендованной литературы***

1. Информатика. Базовый курс. / Под ред. С.В.Симоновича.- СПб., 2000 г.
2. А.П.Микляев, Настольная книга пользователя IBM PC 3-издание М.: "Солон-Р", 2000, 720 с.
3. Симонович С.В., Евсеев Г.А., Мураховский В.И. Вы купили компьютер: Полное руководство для начинающих в вопросах и ответах. - М.: АСТ-ПРЕСС КНИГА; Инфорком-Пресс, 2001.- 544 с.: ил. (1000 советов).
4. Ковтанюк Ю.С., Соловьян С.В. Самоучитель работы на персональном компьютере - К.:Юниор, 2001.- 560с., ил.

[На оглавление](#)

## Лекция 3

### Программное обеспечение ПК. Операционные системы и их функции. Служебные программы.

**Цель лекции:** ознакомить с уровнями программного обеспечения ПК, дать характеристику программам прикладного уровня.

**Ключевые слова:** уровни программного обеспечения, электронные словари, инструментальные языки и системы программирования, системы видеомонтажа, экспертные системы, бухгалтерские системы, интегрированные системы делопроизводства, системы автоматизированного перевода, браузеры, настольные издательские системы, системы автоматизированного проектирования; операционная система, служебные программы

#### Содержание

Программное обеспечение ПК.

Уровни программного обеспечения.

Классификация прикладного уровня.

Операционная система. Функции операционной системы.

Программы служебного уровня.

В основу работы компьютеров положен программный принцип управления, состоящий в том, что компьютер выполняет действия по заранее заданной программе. Этот принцип обеспечивает универсальность использования компьютера: в определенный момент времени решается задача соответственно выбранной программе. После ее завершения в память загружается другая программа и т.д.

**Программа** - это запись алгоритма решения задачи в виде последовательности команд или операторов языком, который понимает компьютер. Конечной целью любой компьютерной программы является управление аппаратными средствами.

Для нормального решения задач на компьютере нужно, чтобы программа была отлажена, не требовала доработок и имела соответствующую документацию. Поэтому, относительно работы на компьютере часто используют термин программное обеспечение (software), под которым понимают совокупность программ, процедур и правил, а также документации, касающихся функционирования системы обработки данных.

Программное и аппаратное обеспечение в компьютере работают в неразрывной связи и взаимодействии. Состав программного обеспечения вычислительной системы называется программной конфигурацией. Между программами существует взаимосвязь, то есть работа множества программ базируется на программах низшего уровня.

Междупрограммный интерфейс - это распределение программного обеспечения на несколько связанных между собою уровней. Уровни программного обеспечения представляют собой пирамиду, где каждый высший уровень базируется на программном обеспечении предшествующих уровней.

Прикладной уровень

Служебный уровень

## Системный уровень

### Базовый уровень

**Базовый уровень.** Базовый уровень является низшим уровнем программного обеспечения. Отвечает за взаимодействие с базовыми аппаратными средствами. Базовое программное обеспечение содержится в составе базового аппаратного обеспечения и сохраняется в специальных микросхемах постоянного запоминающего устройства (ПЗУ), образуя базовую систему ввода-вывода BIOS. Программы и данные записываются в ПЗУ на этапе производства и не могут быть изменены во время эксплуатации.

**Системный уровень.** Системный уровень - является переходным. Программы этого уровня обеспечивают взаимодействие других программ компьютера с программами базового уровня и непосредственно с аппаратным обеспечением. От программ этого уровня зависят эксплуатационные показатели всей вычислительной системы. При подсоединении к компьютеру нового оборудования, на системном уровне должна быть установлена программа, обеспечивающая для остальных программ взаимосвязь с устройством. Конкретные программы, предназначенные для взаимодействия с конкретными устройствами, называют драйверами.

Другой класс программ системного уровня отвечает за взаимодействие с пользователем. Благодаря ему, можно вводить данные в вычислительную систему, руководить ее работой и получать результат в удобной форме. Это средства обеспечения пользовательского интерфейса, от них зависит удобство и производительность работы с компьютером.

Совокупность программного обеспечения системного уровня образует ядро операционной системы компьютера. Наличие ядра операционной системы - это первое условие для возможности практической работы пользователя с вычислительной системой. Ядро операционной системы выполняет такие функции: управление памятью, процессами ввода-вывода, файловой системой, организация взаимодействия и диспетчеризация процессов, учет использования ресурсов, обработка команд и т.д.

**Служебный уровень.** Программы этого уровня взаимодействуют как с программами базового уровня, так и с программами системного уровня. Назначение служебных программ (утилит) состоит в автоматизации работ по проверке и настройке компьютерной системы, а также для улучшения функций системных программ. Некоторые служебные программы (программы обслуживания) сразу входят в состав операционной системы, дополняя ее ядро, но большинство являются внешними программами и расширяют функции операционной системы. То есть, в разработке служебных программ отслеживаются два направления: интеграция с операционной системой и автономное функционирование.

**Прикладной уровень.** Программное обеспечение этого уровня представляет собой комплекс прикладных программ, с помощью которых выполняются конкретные задачи (производственных, творческих, развлекательных и учебных). Между прикладным и системным программным обеспечением существует тесная взаимосвязь. Универсальность вычислительной системы, доступность прикладных программ и широта функциональных возможностей компьютера непосредственно зависят от типа имеющейся операционной системы, системных средств, помещенных в ее ядро и взаимодействии комплекса человек-программа-оборудование.

## **Классификация прикладного программного обеспечения**

**1. Текстовые редакторы.** Основные функции - это ввод и редактирование текстовых данных. Для операций ввода, вывода и хранения данных текстовые редакторы используют системное программное обеспечение. С этого класса прикладных программ начинают знакомство с программным обеспечением и на нем приобретают первые привычки работы с компьютером.

**2. Текстовые процессоры.** Разрешают форматировать, то есть оформлять текст. Основными средствами текстовых процессоров являются средства обеспечения взаимодействия текста, графики, таблиц и других объектов, составляющих готовый документ, а также средства автоматизации процессов редактирования и форматирования. Современный стиль работы с документами имеет два подхода: работа с бумажными документами и работа с электронными документами. Приемы и методы форматирования таких документов различаются между собой, но текстовые процессоры способны эффективно обрабатывать оба вида документов.

**3. Графические редакторы.** Широкий класс программ, предназначенных для создания и обработки графических изображений. Различают три категории:

- растровые редакторы;
- векторные редакторы;
- 3-D редакторы (трехмерная графика).

В **растровых редакторах** графический объект представлен в виде комбинации точек (растров), которые имеют свою яркость и цвет. Такой подход эффективный, когда графическое изображение имеет много цветов и информация про цвет элементов намного важнее, чем информация про их форму. Это характерно для фотографических и полиграфических изображений. Применяют для обработки изображений, создания фотоэффектов и художественных композиций.

**Векторные редакторы** отличаются способом представления данных изображения. Объектом является не точка, а линия. Каждая линия рассматривается, как математическая кривая  $n$  порядка и представлена формулой. Такое представление компактнее, чем растровое, данные занимают меньше места, но построение объекта сопровождается пересчетом параметров кривой в координаты экранного изображения, и соответственно, требует более мощных вычислительных систем. Широко применяются в рекламе, оформлении обложек полиграфических изданий.

**Редакторы трехмерной графики** используют для создания объемных композиций. Имеют две особенности: разрешают руководить свойствами поверхности в зависимости от свойств освещения, а также разрешают создавать объемную анимацию.

**4. Системы управления базами данных (СУБД).** Базой данных называют большие массивы данных, организованные в табличные структуры. Основные функции СУБД:

- создание пустой структуры базы данных;
- наличие средств ее заполнения или импорта данных из таблиц другой базы;
- возможность доступа к данным, наличие средств поиска и фильтрации.

В связи с распространением сетевых технологий, от современных СУБД требуется возможность работы с удаленными и распределенными ресурсами, которые находятся на серверах Интернета.

**5. Электронные таблицы.** Предоставляют комплексные средства для хранения разных типов данных и их обработки. Основной акцент смещен на преобразование данных, предоставлен широкий спектр методов для работы с числовыми данными. Основная особенность электронных таблиц состоит в автоматическом изменении содержимого всех ячеек при изменении отношений, заданных математическими или логическими формулами.

Широкое применение находят в бухгалтерском учете, анализе финансовых и торговых рынков, средствах обработки результатов экспериментов, то есть в автоматизации регулярно повторяемых вычислений больших объемов числовых данных.

**6. Системы автоматизированного проектирования (САД-системы).** Предназначены для автоматизации проектно-конструкторских работ. Применяются в машиностроении, приборостроении, архитектуре. Кроме графических работ, разрешают проводить простые расчеты и выбор готовых конструктивных элементов из существующей базы данных.

Особенность САД-систем состоит в автоматическом обеспечении на всех этапах проектирования технических условий, норм и правил. САПР являются необходимым компонентом для гибких производственных систем (ГВС) и автоматизированных систем управления технологическими процессами (АСУ ТП).

**7. Настольные издательские системы.** Автоматизируют процесс верстки полиграфических изданий. Издательские системы отличаются расширенными средствами управления взаимодействия текста с параметрами страницы и графическими объектами, но имеют более слабые возможности по автоматизации ввода и редактирования текста. Их целесообразно применять к документам, которые предварительно обработаны в текстовых процессорах и графических редакторах.

**8. Редакторы HTML (Web-редакторы).** Особый класс редакторов, объединяющих в себе возможности текстовых и графических редакторов. Предназначены для создания и редактирования Web-страниц Интернета. Программы этого класса можно использовать при подготовке **Редакторы HTML** овке электронных документов и мультимедийных изданий.

**9. Браузеры** (средства просмотра Web-документов). Программные средства предназначены для просмотра электронных документов, созданных в формате HTML. Воспроизводят, кроме текста и графики, музыку, человеческий язык, радиопередачи, видеоконференции и разрешают работать с электронной почтой.

**10. Системы автоматизированного перевода.** Различают электронные словари и программы перевода языка.

**Электронные словари** - это средства для перевода отдельных слов в документе. Используются профессиональными переводчиками, которые самостоятельно переводят текст.

**Программы автоматического перевода** используют текст на одном языке и выдают текст на другом, то есть автоматизируют перевод. При автоматизированном переводе невозможно получить качественный исходный текст, поскольку все сводится к переводу отдельных лексических единиц. Но, для технического текста, этот барьер снижен.

**Программы автоматического перевода целесообразно использовать:**

- при абсолютном незнании иностранного языка;
- при необходимости быстрого ознакомления с документом;

- для перевода на иностранный язык;
- для создания черновика, который потом будет подправлен полноценным переводом.

**13. Интегрированные системы делопроизводства.** Средства для автоматизации рабочего места руководителя. В частности, это функции создания, редактирования и форматирования документов, централизация функций электронной почты, факсимильной и телефонной связи, диспетчеризация и мониторинг документооборота предприятия, координация работы подразделов, оптимизация административно-хозяйственной деятельности и поставка оперативной и справочной информации.

**14. Бухгалтерские системы.** Имеют функции текстовых, табличных редакторов и СУБД. Предназначены для автоматизации подготовки начальных бухгалтерских документов предприятия и их учета, регулярных отчетов по итогам производственной, хозяйственной и финансовой деятельности в форме, приемлемой для налоговых органов, внебюджетных фондов и органов статистического учета.

**15. Финансовые аналитические системы.** Используют в банковских и биржевых структурах. Разрешают контролировать и прогнозировать ситуацию на финансовых, торговых рынках и рынках сырья, выполнять анализ текущих событий, готовить отчеты.

**16. Экспертные системы.** Предназначены для анализа данных, содержащихся в базах знаний и выдачи результатов, при запросе пользователя. Такие системы используются, когда для принятия решения нужны широкие специальные знания. Используются в медицине, фармакологии, химии, юриспруденции. С использованием экспертных систем связана область науки, которая носит название инженерии знаний.

Инженеры знаний - это специалисты, являющиеся промежуточным звеном между разработчиками экспертных систем (программистами) и ведущими специалистами в конкретных областях науки и техники (экспертами).

**17. Геоинформационные системы (ГИС).** Предназначены для автоматизации картографических и геодезических работ на основе информации, полученной топографическим или аэрографическими методами.

**18. Системы видеомонтажа.** Предназначены для цифровой обработки видеоматериалов, монтажа, создания видеоэффектов, исправления дефектов, добавления звука, титров и субтитров. Отдельные категории представляют учебные, справочные и развлекательные системы и программы. Характерной особенностью являются повышенные требования к мультимедийной составляющей.

**19. Инструментальные языки и системы программирования.** Эти средства служат для разработки новых программ. Компьютер "понимает" и может выполнять программы в машинном коде. Каждая команда при этом имеет вид последовательности нулей и единиц. Писать программы на машинном языке крайне неудобно. Поэтому программы разрабатываются на языке, понятном человеку (инструментальный язык или алгоритмический язык программирования), после чего, специальной программой, которая называется транслятором, текст программы переводится (транслируется) на машинный код.

**Трансляторы бывают двух типов:**

- интерпретаторы,
- компиляторы.

**Интерпретатор** читает один оператор программы, анализирует его и сразу выполняет, после чего переходит к обработке следующего оператора.

**Компилятор** сначала читает, анализирует и переводит на машинный код всю программу и только после завершения всей трансляции эта программа выполняется.

Инструментальные языки делятся на языки низкого уровня (близкие к машинному языку) и языки высокого уровня (близкие к человеческим языкам). К языкам низкого уровня принадлежат ассемблеры, а высокого - Pascal, Basic, C/C++, языки баз данных и т.д Basic, Visual C++, Visual FoxPro и др.

### Операционные системы

Операционные системы (ОС) являются основой системного программного обеспечения. Без операционной системы доступ к аппаратуре и программам современного компьютера совершенно невозможен. Все аппаратные, а также и программные средства предоставляются пользователю только через посредника — операционную систему (рис. 3.1).



**Рис. 3.1.** Взаимодействие пользователя с аппаратными и программными средствами компьютера

**Операционной системой** называется комплекс программ, которые обеспечивают автоматизацию доступа к аппаратным и программным ресурсам компьютера.

Основными функциями операционных систем являются:

- загрузка программ в оперативную память и управление ходом их выполнения;
- обеспечение операций по обмену данными между выполняющейся программой и внешними устройствами;
- обслуживание нестандартных ситуаций в ходе выполнения программы;
- удаление выполненной программы из оперативной памяти и освобождение места для загрузки новой программы;
- организация хранения и поиска программ и данных на внешних носителях;
- организация взаимодействия пользователя и операционной системы — прием и выполнение команд пользователя;
- выполнение различных вспомогательных (сервисных) функций, таких как форматирование дисковых устройств, копирование информации с одного дискового устройства на другое и некоторые другие.

Операционная система должна находиться в рабочем состоянии в течение всего времени функционирования компьютера. Работа операционной системы начинается в момент включения компьютера и заканчивается в момент его выключения.

Для каждой модели компьютера разрабатываются индивидуальные операционные системы. Более того, для одной и той же модели, как правило, имеется несколько различных операционных систем с разным назначением и разными возможностями и свойствами. Так, существуют операционные системы, которые могут управлять одновременным выполнением нескольких программ — **многопрограммные** — или только одной — **однопрограммные ОС**. Есть системы, которые могут обслуживать только одного — **однопользовательские** — или одновременно нескольких человек — **многопользовательские ОС**. Для обеспечения работы локальных и глобальных сетей разработаны **сетевые** операционные системы.

Для IBM-совместимых персональных компьютеров разработано несколько различных семейств операционных систем: MS-DOS, Windows, Linux (разновидность популярной системы Unix) и некоторые другие. Одной из наиболее простых операционных систем для персональных компьютеров является устаревшая *однопользовательская* и *однопрограммная* операционная система MS-DOS (Microsoft Disk Operation System). Ее первая версия была разработана еще в 1981–1982 годах. Операционные системы семейства Windows 9x являются *многопрограммными*, но *однопользовательскими*, а системы семейств Windows NT/2000/XP и Linux относятся к *многопрограммным, многопользовательским и сетевым*.

#### Служебные программы ОС WINDOWS (утилиты)

В состав операционных систем обычно входят программы настройки и обслуживания компьютера, а также программы, позволяющие получать информацию о работе компьютера

1. Дефрагментация диска.
2. Очистка диска.
3. Сведения о системе.
4. Архивация данных.
5. Восстановление системы

#### **Контрольные вопросы**

1. На какие уровни делится программное обеспечение?
2. Какие функции выполняет ядро операционной системы?
3. Для чего предназначены программы базового уровня?
4. Какие классы программ служебного уровня вы знаете?
5. С помощью программ какого класса можно осуществлять ввод, редактирование и оформление текстовых данных?
6. Какие вы знаете категории графических редакторов?
7. В каких случаях целесообразно использовать системы автоматизированного перевода?
8. Что такое интерпретатор и компилятор? Какая между ними разница?

#### **Темы самостоятельных работ**

1. Общая структура и возможности справочной системы Windows XP.
2. Архивация файловых объектов. Создание архива в Windows XP.
3. Вирусы и борьба с ними. Классификация вирусных и антивирусных программ.
4. Программное обеспечение. Переводчики.

5. Служебные программы: дефрагментация и очистка диска.
6. Служебные программы: восстановление системы.

### *Список рекомендованной литературы*

1. Информатика. Базовый курс. / Под ред. С.В.Симоновича. - СПб., 2000 г.
2. А.П.Микляев, Настольная книга пользователя IBM PC 3-издание М.:, "Солон-Р", 2000, 720с.
3. Симонович С.В., Евсеев Г.А., Мураховский В.И. Вы купили компьютер: Полное руководство для начинающих в вопросах и ответах. - М.: АСТ-ПРЕСС КНИГА; Инфорком-Пресс, 2001.- 544 с.: ил. (1000 советов).
4. Ковтанюк Ю.С., Соловьян С.В. Самоучитель работы на персональном компьютере - К.:Юниор, 2001.- 560с., ил.

[На оглавление](#)

## Лекция 4

### Компьютерные сети. Протоколы. Адресация в Интернете

**Цель лекции:** ознакомить с понятием вычислительных сетей, с их основным назначением и классификацией, охарактеризовать сеть Интернет и основные принципы работы в сети.

**Ключевые слова:** вычислительные сети, локальные сети, региональные и глобальные сети, топология сетей, протоколы TCP и IP, цифровая и доменная адресация, домены высшего уровня, служба WWW, электронная почта, поисковые системы, служба FTP, служба ICQ, служба IRC.

#### Содержание

Вычислительные сети. Общая характеристика.

Компьютерные сети. Их классификация.

Топология сетей.

Сеть Интернет. Протоколы и адресация в сети Интернет.

Информационные службы в сети Интернет.

**Вычислительные сети. Общая характеристика.** Вычислительные сети, как результат развития современной технологии, возникли как системы для передачи информации на большие расстояния и расширения возможностей ЭВМ, однако, подобно железнодорожной сети, они меняют свое назначение и начинают служить для связи между людьми.

В ходе своего развития средства обработки информации последовательно дополнялись введением телеграфа и телефона, передачей факсимильных сообщений и ксерографий.

Современная техника обработки информации все больше согласуется с естественным языком, представленными в письменной, печатной или устной форме.

Вычислительные сети являются одним из конкретных воплощений общей тенденции к распределению информации, которая проявляется в создании мультипроцессорных систем, распределенных баз данных и в использовании интеллектуальных терминалов.

Вычислительная сеть объединяет терминалы, компьютеры, базы данных и другие компоненты, удаленные друг от друга.

Техническая база вычислительных сетей продолжает быстро развиваться. В целом ряде сетей уже используется оптоволоконные системы связи и спутниковые системы передачи данных. Для создания глобальной системы передачи цифровой информации были дополнительно проложены подводные кабели цифровой связи (благодаря развитию оптоволоконной технике). Как только какого рода системы начали эффективно эксплуатироваться, была создана многофункциональная глобальная система передачи цифровой информации.

В сети начали использовать для передачи речи метод коммутации пакетов. Это уже сделано в спутниковой сети фирмы Satellite Business Systems, где совместно с передачей речи, данных передается и факсимильная информация с использованием методов адаптивного мультиплексирования.

Типичная современная сеть обеспечивает множество различных функций в отличие от ранних сетей, где задачи непосредственной обработки и передачи данных смешивались, и специфика конструкции системы стирала границу между этими функциями. Один запрос пользователя может вызвать в сети целую серию обменов сообщениями.

Современные сети имеют тенденцию к неоднородности, используя компьютеры и терминалы, выпускаемые различными фирмами. К сожалению, не существует никакого единого стандарта, которому подчинялись бы интерфейс, форматы и процедуры.

Сеть включает ЭВМ которая выполняет различные функции (вычислительные, поиск информации, деловые операции, хранение данных), а также терминалы пользователей сети и концентраторы, связывающие группы терминалов с подсистемой передачи данных.

Компьютеры, подсоединенные к базовой сети передачи данных, называются главными – host computer , т.е. эти компьютеры действуют как «хозяин» по отношению к установленной там же сети передачи данных. Главный компьютер обычно обеспечивает обслуживание остальной сети, но может получать обслуживание от других компьютеров.

Все основные функции по передаче данных возлагаются на базовую сеть. Базовая сеть должна обеспечить защиту от ошибок в линиях связи с помощью повторной передачи или обходов вышедших из строя участников сети изменением маршрута трафика. Возложение некоторых функций по передаче данных на главные компьютеры позволяет значительно повысить надежность передачи данных.

Внутри главных компьютеров также имеется дополнительный «интерфейс связи», показанный на рисунке штриховой линией.

Итак, компьютерная сеть (Computer Network) – это совокупность компьютеров, соединенных с помощью каналов связи и средств коммутации в единую систему для обмена сообщения и доступа пользователей к программным, техническим, информационным и организационным ресурсам сети. Компьютерную сеть представляют как совокупность узлов (компьютеров и сетевого оборудования) и соединяющих их ветвей (каналов связи). Ветвь сети – это путь, соединяющий два смежных узла. Различают оконечные, расположенные в конце только одной ветви, промежуточные, расположенные на концах более чем одной ветви, и смежные – такие узлы соединены по крайней мере одним путем, не содержащим никаких других узлов.

**Классификация компьютерных сетей.** Компьютеры могут соединяться в сеть разными способами. Логический и физический способы соединения компьютеров, кабелей и др. компонентов сети, называется ее топологией. Топология характеризует свойства сетей, не зависящие от их размеров. При этом не учитываются производительность и принцип работы этих объектов, их типы, длины каналов (хотя при проектировании эти факторы очень важны). По географическому размещению сети делятся на локальные, региональные, корпоративные, глобальные .

**Локальная сеть** (ЛВС или LAN-Local Area Network) – сеть, связывающая абонентов, расположенных в пределах небольшой территории. Обычно такая сеть привязана к конкретному месту(комната, здание, предприятие, фирма, офис, банк). Протяженность локальной сети можно ограничить пределами 2–2,5 км.

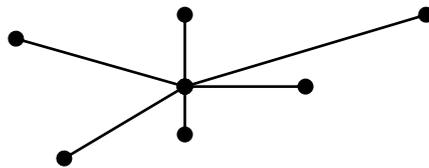
**Региональная сеть** (РВС или MAN-Metropolitan Area Network) – связывает абонентов, расположенных на значительном расстоянии друг от друга. Она может включать абонентов внутри большого города, экономического региона, отдельной страны. Обычно расстояние между абонентами региональной вычислительной сети составляет от 10 до 100 км.

**Глобальная сеть** (ГВС или WAN-Wide Area NetWork) – сеть, соединяющая абонентов (компьютеры), расположенных в различных странах на различных континентах. Взаимодействие между абонентами такой сети может осуществляться на базе телефонных линий связи, радиосвязи и систем спутниковой связи. ГВС позволяют решить проблему объединения информационных ресурсов всего человечества и организации доступа к этим ресурсам. Глобальная сеть объединяет локальные сети. Объединение ГВС, РВС И ЛВС позволяет создавать многосетевые иерархии. Они обеспечивают мощные, экономически целесообразные средства обработки огромных информационных массивов и доступ к неограниченным информационным ресурсам.

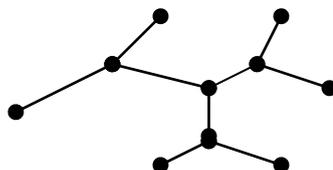
### **Топология ВС. Формы базовой сети передачи данных.**

Наиболее очевидным образом вычислительные сети могут отличаться топологией, т.е. по форме объединения терминалов и узлов с помощью линий связи.

- 1) **Звездообразная топология.** Здесь центральный узел соединяется с каждым из терминалов и обычно помещен приблизительно в «центре тяжести» сети. Вся информация между рабочими местами проходит через этот центральный узел. Производительность такой вычислительной сети в первую очередь зависит от мощности центральной ЭВМ. Она может быть узким местом вычислительной сети. В случае выхода из строя центрального узла нарушается работа всей сети.



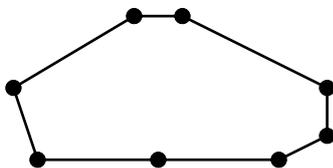
- 2) **Древовидная топология.** Позволяет линиям сливаться в промежуточных точках. Для любого заданного множества точек существует «минимальное стягивающее дерево», которое сводит общую длину линий к наименьшему возможному значению.



- 3) **Многосвязная топология.** Наиболее часто используется в сетях передачи данных общего пользования и в некоторых современных частных сетях. Здесь линии обеспечивают несколько возможных путей между узлами.



- 4) **Кольцевая топология.** Эта топология менее экономична, чем древовидная, однако лучше, чем звездообразная. Кроме того, кольцевая топология более надежна при передаче данных в любом из направлений по кольцу. Кольцевая топология чаще применяется в локальных сетях.



Сообщения циркулируют по кругу. Основная проблема - каждая рабочая станция должна активно участвовать в пересылке информации, в случае выхода из строя одной станции парализуется вся сеть.

### **Сеть Интернет.**

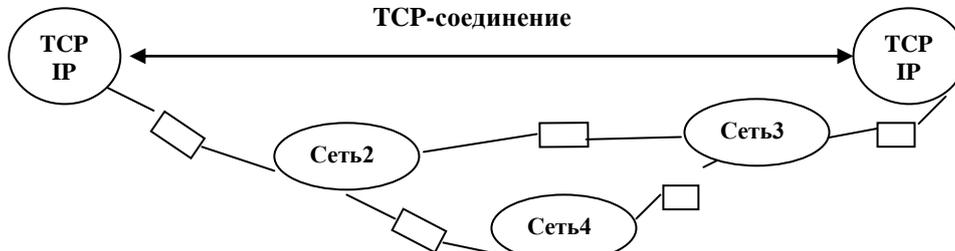
Интернет, ставший символом нового этапа развития информационных технологий на рубеже веков, оказывает значительное влияние на работу специалистов в самых разных областях человеческой деятельности, в том числе и на работу специалистов-гуманитариев. По-видимому, роль Интернета в их профессиональной деятельности будет только возрастать. В связи с этим большое значение для специалистов приобретают знание основных возможностей сети и практические навыки работы в Интернете.

Интернет в настоящее время является самой крупной и наиболее популярной в мире глобальной сетью. Считается, что Интернет — это прямой потомок сети ARPANET (ARPA — Advanced Research Projects Agency, US Defense Department, Агентство перспективных исследований министерства обороны США), которая была создана в 1969 году в США в результате соединения линиями связи четырех крупных компьютеров, расположенных в различных местах страны. Исходная цель создания сети ARPANET — исследование возможности «выживания», то есть сохранения работоспособности информационной сети при вероятном возникновении военных действий и выводе из строя некоторых ее участков. В этих исследованиях принимали участие научные работники университетских и академических центров, которые быстро оценили преимущества работы в информационных сетях. Очень скоро локальные и глобальные сети стали использоваться в научных исследованиях, образовании и экономике. Особую популярность приобрели такие информационные услуги в сетях, как электронная почта и электронные доски объявлений. Затем появились доступ к ресурсам других компьютеров, электронные журналы, телеконференции, дистанционное образование и т. д.

### **Протоколы и адресация в сети Интернет.**

Протокол TCP (Transmission Control Protocol) — протокол управления передачей данных, использующий автоматическую повторную передачу пакетов, содержащих ошибку. Этот протокол отвечает за разбиение передаваемой информации на пакеты и правильное восстановление информации из пакетов получателя. Кроме того, этот протокол решает задачу по

гарантированию надежности передачи данных, т.е. обеспечение надежного канала обмена данными между процессами в сети. Надёжность достигается за счёт того, что TCP основан на установлении логических соединений между взаимодействующими процессами. Ошибки в передаче данных на уровне протокола IP не будут влиять на правильное получение данных.



Между конечными узлами TCP создаёт надёжный канал связи.

Схема передачи информации по протоколу TCP/IP такова: протокол TCP/IP разбивает информацию на пакеты и нумерует их; далее с помощью протокола IP все пакеты передаются получателю, где с помощью протокола TCP проверяется все ли пакеты получены; после получения всех пакетов протокол TCP располагает их в нужном порядке и собирает в единое целое.

Протокол IP (Internet Protocol) – протокол межсетевого взаимодействия, отвечающий за адресацию и позволяющий пакетам на пути к конечному пункту назначения проходить по нескольким сетям.

### Структура IP

	Заголовок	
	Идентификатор пакета	
Время жизни	Протокол верхнего уровня	Контрольная сумма
	IP Адрес источника	
	IP Адрес назначения	

Каждому компьютеру назначается адрес, который известен по всему Интернету. Каждая дейтаграмма содержит адрес отправителя и получателя. Промежуточные маршрутизаторы используют адрес получателя для того, чтобы принимать решения о дальнейшем маршруте дейтаграммы.

IP-протокол должен передавать пакеты между сетями, в каждой очередной сети, лежащей на пути перемещения пакета, IP вызывает транспортные средства, принятые в этой сети, чтобы с их помощью передать этот пакет на маршрутизатор, ведущий к следующей сети.

Протоколы TCP/IP позволяют любой паре компьютеров, присоединенных к нему, взаимодействовать друг с другом

Протоколы TCP/IP обеспечивают подтверждения между отправителем и получателем, а не между отправителем и промежуточными машинами на пути, даже когда две машины не связаны общей физической сетью.

Помимо базовых средств транспортного уровня, протоколы TCP/IP включают стандарты для наиболее часто используемых приложений, таких как электронная почта, передача файлов и удалённый доступ.

Поэтому при разработке прикладных программ, использующих TCP/IP, программисты часто могут обнаружить, что существующее программное обеспечение уже обеспечивает средства, которые им нужны.

**Адресация.** Каждый компьютер, подключённый к сети Интернет имеет два равноценных уникальных адреса: цифровой IP–адрес и символический доменный адрес, т. е. адрес имеет такой формат, который позволяет вести его обработку автоматически и несёт некоторую информацию о своём владельце.

Цифровой IP–адрес.

IP–адрес компьютера состоит из 4 байт, разбитые на блоки по 8 бит. Два блока определяют адрес сети, а два другие – адрес компьютера внутри этой сети.

1 байт	2 байт	3 байт	4 байт
адрес сети		адрес подсети	адрес компьютера

Для удобства IP–адрес записывают в виде 4 чисел со значениями от 0 до 255, разделённых точками, напр.: 192.168.11.7.

192.168 – адрес сети

11 – адрес подсети

7 – адрес компьютера в подсети.

IP–адрес назначается администратором во время конфигурирования компьютеров и маршрутизаторов. Номер сети выбирается произвольно, но если сеть находится в Интернете номер назначается по рекомендации специального подразделения Internet(Internet Network Information Center, InterNIC).

Поставщики услуг Internet получают диапазоны адресов у подразделений InterNIC, а затем распределяют их между своими абонентами.

Символьные доменные адреса. Символьные имена строятся по иерархическому признаку. Через точку в имени указываются сначала имя узла, затем имя группы узлов( напр.,организация), затем имя более крупной группы (поддомена) и далее имя домена самого высокого уровня (например домены, объединяющие по географическому принципу– RU, UK, US, CA...).

Пример адреса: Base 2.sales.zil.ru

Между доменным именем и IP–адресом узла нет никакого алгоритмического соответствия, поэтому приходится использовать дополнительные таблицы или службы, чтобы узел сети однозначно определялся как по доменному имени, так и по IP–адресу. В сетях TCP/IP используется специальная распределённая служба Domain Name System(DNS), которая устанавливает эти соответствия на основании создаваемых администраторами сети таблиц соответствия. Поэтому доменные имена называют также DNS–именами.

Доменный адрес в отличие от цифрового является символическим и легче запоминается человеком. Доменный адрес определяет область, представляющую ряд хост–компьютеров. В отличие от цифрового адреса он читается в обратном порядке. В начале стоит имя компьютера, затем имя сети, в которой он находится. Пример, Perdca.silk.glas.apc.org

Имя реального компьютера–Perdca silk–имя группы, присвоившей имя компьютеру, glas–имя более крупной группы, присвоившей имя домену silk и т.д.

В процессе передачи данных доменный адрес преобразуется в IP–адрес.

В системе адресов Internet приняты домены, представленные географическими регионами (они представлены двумя буквами).

Напр, Франция – fr

Канада – Ca

США – US

Россия – ru

Германия – de

Швеция – se

Узбекистан – uz

(Компьютер на территории Канады может иметь имя – hockey.guelph.ca)

Общее число кодов стран – 300, компьютерные сети существуют принципиально в 170 из них.

К доменам первого или высшего уровня также относятся шесть организационных доменов:

- Com – коммерческие организации
- Edu – учебные заведения (университеты, колледжи, ...)
- Org – некоммерческие организации
- gov – правительственные организации
- mil – военные организации (флот, армия)
- int – международные организации
- net – организации, предоставляющие сетевые ресурсы.

Например, [WWW.virginiya.edu](http://WWW.virginiya.edu)

Имя имя  
университета зоны

Имя hosta [virginiya.edu](http://virginiya.edu) присвоено университету.

Например, 128.10.2.30.

10000000    00001010    00000010    00011110

## Информационные службы в сети Интернет

Единство протоколов TCP/IP в Интернете обеспечивает возможность связи между миллионами компьютеров, распределенных по всему миру. Для плодотворного взаимодействия компьютеров должна существовать сетевая служба. В Интернете существует немало сетевых служб, среди них самая распространенная - WWW (World Wide Web). Для ее работы необходима пара программ: Web - сервер и Web - клиент. Web - серверы устанавливают организации, поставляющие информацию. Web - клиенты работают на компьютерах пользователей. Web - клиенты еще называют **браузерами**. Для компьютеров IBM PC наиболее распространен браузер Microsoft Internet Explorer, входящий в стандартный комплект Windows. То есть для работы со службой WORLD WIDE WEB никакие специальные клиентские программы устанавливать не нужно. В операционной системе уже есть всё необходимое. Служба WWW предназначена для доступа к электронным документам особого рода, которые называются Web-страницами. Web-страница - это документ, в котором кроме текста содержатся специальные команды форматирования, а также встроенные объекты (рисунки, аудио - и видеоклипы и др.) Просматривают Web-страницы браузерами. Каждый такой Web-документ в Интернете имеет свой уникальный адрес - **URL (Uniform Resource Locator)**. В первой части адреса указывается имя сетевого протокола **HTTP (Hypertext Transfer Protocol)**, по которому осуществляется доступ к данному ресурсу. Второй элемент **URL-адреса** - доменное имя компьютера, на котором хранится данный документ. Последний элемент адреса - путь доступа к файлу, содержащему **Web-документ**. Например, <http://www.pogoda.ru/Photo.html>.

## *Электронная почта*

Старейшая служба Интернета - **электронная почта (E-mail)**. Она предназначена для обмена сообщениями между удаленными партнерами. Как и любая служба, она основана на паре программ: почтовый сервер плюс почтовый клиент. Почтовые серверы установлены на узловых компьютерах Интернета, а почтовые клиенты устанавливают пользователи на своих компьютерах. Стандартный почтовый клиент в системе Windows - программа **Microsoft Outlook Express**. В электронной почте используются два протокола. По одному протоколу происходит отправка почты, а по другому - ее прием. Необходимость в двух протоколах связано с требованиями безопасности. Для отправки на сервер используют протокол **SMTP (Simple Mail**

**Transfer Protocol**), который не требует идентификации личности. Для получения поступившей почты используют протокол **POP3 (Post Office Protocol 3)**. Он предусматривает идентификацию личности, т.е. проверку регистрационного имени (Login) и пароля (Password).

### *Служба передачи файлов FTP.*

В тех случаях, когда надо отправить или получить файл произвольного формата, например, программу, используется служба **FTP**, основанная на одноименном протоколе (**File Transfer Protocol- Протокол передачи файлов**). На многих узловых компьютерах Интернета работают **FTP- сервер**, а на пользовательском компьютере должен быть установлен **FTP- клиент**. В системе Windows в **FTP-клиент** встроен в **Internet Explorer**. Поэтому когда в ходе работы со службой **WWW** пользователь сталкивается с архивом или файлом, передаваемым по протоколу **FTP**, **Internet Explorer** автоматически обеспечивает его прием и сохранение.

### *Служба телеконференций или группы новостей*

Эта служба позволяет отправлять сообщения, указав имя телеконференции. Сначала сообщение поступает на сервер поставщика услуг Интернета, потом передается на все серверы, которые с ним связаны, потом на серверы, связанные с ними и т.д. Сообщения телеконференций хранятся на серверах от 3 до 7 дней. В течение этого времени любой пользователь может прочитать это сообщение на сервере своего поставщика услуг. Для работы с группами новостей служит программа **Outlook Express**. При первом соединении с сервером новостей программа принимает полный список всех телеконференций открытых на данном сервере.

### *Поисковые службы*

В мире созданы многие миллионы **Web-страниц**. Пользуясь ссылками на страницах можно бесконечно долго путешествовать в информационном пространстве Сети, переходя от одной страницы к другой. На помощь приходят специальные поисковые машины (поисковые сервера). По принципу действия поисковые сервера делятся на две группы: поисковые каталоги и поисковые индексы.

Поисковые каталоги служат для тематического поиска. Характерная особенность каталогов связана с тем, что их готовят люди. Отсюда основное достоинство - высокая релевантность. В результате строгого отсева в поисковые каталоги включаются только те ресурсы, которые действительно представляют значимость для потребителей.

Услугами поисковых каталогов целесообразно пользоваться при проведении наиболее общего поиска, если тема поиска очерчена широко. Так часто бывает перед подготовкой рефератов, статей, учебно-методических пособий. Наиболее популярный зарубежный каталог Yahoo. Среди русскоязычных каталогов наиболее популярны Yandex, Rambler, Aport, Mail.

Поисковые индексы работают как алфавитные указатели и в отличие от каталогов составляются автоматически. Специальные программы (роботы) в автоматическом режиме круглосуточно просматривают пространство WWW и копируют обнаруженные документы на сервер системы. Далее документы индексируются, и в итоге образуется огромный справочник (указатель), по которому система очень быстро может указать, в каких документах встречаются те или иные слова. Крупнейшим поисковым указателем мира является система Google, также известен сервер AltaVista, alltheweb, Lycos, а также из российских - Апорт, Яндекс.

Недостатком таких поисковых указателей является недостаточная релевантность результатов поиска, т.е. несмотря на огромную базу не каждый найденный документ соответствует реальным ожиданиям пользователя. Услугами указателей целесообразно пользоваться при проведении глубокого или специализированного поиска, когда тема поиска хорошо известна, и можно четко сформировать набор ключевых слов, характеризующих документ.

Имеются сайты, посвященные поисковым системам: searchenginewatch.com, searchengineshowdown.com, freepint.com, webmasterworld.com и т.д.

### *Служба ICQ*

В 1996 году появилась коммуникационная служба, которая с необыкновенной скоростью распространилась по всему “Интернет - свету”. Это служба **ICQ (I Seek You)**. Систему ICQ поддерживают серверы Mirabilis, обеспечивающие соединение по особому эффективному протоколу. Подключение к конкретному серверу обеспечивает программа-клиент, устанавливаемая на пользовательском компьютере. В ICQ введена номерная система идентификации пользователей. Всем включающимся в ICQ присваивается уникальный регистрационный номер UIN. На каждый UIN заводится автоматически персональный **web – сайт**, где приводится информация о данном ICQ-пользователе. Популярность этой службы объясняется тем, что, выйдя в Интернет, можно обмениваться сообщениями с партнерами, можно переслать файл, собраться в кружок на **Chat** (чат), послать почтовое сообщение, переслать URL - ссылку на интересный WEB- сайт и т.д.

### *Служба IRC (Internet Relay Chat)*

Это служба была изначально разработана для непосредственного общения по Интернету. Общение в этой среде поддерживается **IRC-серверами**, связанными в **IRC-сети (IRCnet, EFnet, RUSnet, и т.д.)**. Серверы обеспечивают взаимодействие многих участников, объединяющихся для общения в специальные группы – каналы. Каналы формируются по какому-либо определенному признаку – по тематике, языку, проблеме. Каждый канал обслуживается конкретной **IRC-сетью**. После выбора **IRC-сервера** на экране появляется список каналов, можно определить направленность бесед на них. Все, кто подключился к каналу, видят одинаковый текст на своих компьютерах и список участников. Служба **IRC** также позволяет обмениваться файлами, ссылками, информировать о появлении на канале желанного **Nickname**, группировать собеседников частных бесед и т.д.

### *Контрольные вопросы*

1. Вычислительные сети. Причины их создания.
2. Опишите типы сетей.
3. Какие формы объединения компьютеров с помощью линий связи вы знаете.
4. Что такое протокол? Опишите функции протоколов TCP и IP в сети Интернет.
5. Виды адресации в сети Интернет. Связь между адресами.
6. Служба WWW. Клиентская программа службы.
7. Электронная почта. Основные протоколы.
8. Поисковые каталоги и поисковые роботы.
9. Коммуникационная служба **ICQ**.
10. Служба IRC (Internet Relay Chat)

### *Темы самостоятельных работ*

1. История возникновения и развития сети Интернет.
2. Мультимедиа технологии в Интернете.
3. Интерактивное общение в Интернете. Интернет-пейджеры (IRC, ICQ, NetMeeting).
4. Электронная коммерция в Интернете. Интернете.
5. Организация телеконференций в Интернете. Электронные доски объявлений.

## 6. Перспективы развития сети Интернет.

### *Список рекомендованной литературы*

1. Н. Н. Жеретинцева. Курс лекций по компьютерным сетям. Владивосток: ДВГМА, 2000. – 158 с.
2. В.А. КАЙМИН. ИНФОРМАТИКА. Москва: ИНФРА-М, 2001.
3. В.Г.Олифер, Н.А. Олифер. Компьютерные сети. Учебник для ВУЗов. Питер, 2006.

[На оглавление](#)

## Лекция 5

### Создание Web-страниц. Язык HTML. Структура документа.

**Цель лекции:** ознакомить с базовыми понятиями языка HTML, описать структуру HTML-документа и теги.

**Ключевые слова:** язык разметки гипертекста, теги, атрибуты тегов, логическое и физическое форматирование текста.

#### Содержание

Язык HTML.

Понятие тегов. Параметры или атрибуты тегов.

Структура HTML-документа.

Тэг <BODY> и его атрибуты.

Тэги форматирования текста.

**Язык HTML.** HTML -Hyper Text Markup Language – представляет собой язык разметки, т.е код, записанный на этом языке и включающий специальные символы – отметки. Такие символы лишь управляют отображением документа, но сами не отображаются. В языке HTML эти символы принято называть тэгами (tag – ярлык, признак).

Например,

**<B>** Этот текст будет выведен жирным шрифтом **</B>** ,

т.е. текст между тэгами будет выглядеть следующим образом

**Этот текст будет выведен жирным шрифтом.**

Все тэги в HTML принято выделять символами-ограничителями (< , >), между которыми записывается идентификатор (имя) тэга (в примере это **B**), и возможно, его атрибуты. Единственным исключением являются тэги комментария с более сложными ограничителями (< ! -- и -- >).

Большинство тегов используются попарно, т.е. для открывающего тэга имеется его пара – закрывающий тэг. Оба тэга в паре записываются одинаково, за исключением символа “/” (слеш) в закрывающем тэге. Принципиальным отличием парных тегов является то, что закрывающие тэги не используют параметры. Тэги, которые нуждаются в завершающих тэгах, называются также тэгами – контейнерами. Все, что включается между открывающим и закрывающим тэгом, называется содержимым

тэга-контейнера.

Существует ряд тэгов, для которых не нужно вставлять закрывающий тэг. Если опущены некоторые завершающие тэги, то современные браузеры во многих случаях правильно форматирует документы, однако такая практика не рекомендуется. Например, тэг вставки изображения `<IMG>`, перевода строки `<BR>` и др. могут быть записаны без соответствующих пар `</IMG>`, `</BR>` и т.п. Неверно записанный тэг или его параметр просто игнорируются браузерами (сюда относятся также тэги, не распознанные браузерами).

Параметры или атрибуты тэгов. Наборы параметров индивидуальны для каждого тэга. Параметры записываются по следующим правилам:

- после имени тэга могут следовать параметры, отделенные пробелами;
- порядок следования параметров произволен;
- параметры могут иметь значения, которые задаются после названия параметра через знак `' = '`;
- обычно значения параметров задаются в кавычках;

**Структура документа.** Первым тэгом, с которого следует начинать описание HTML-документа, является

тэг `<HTML>`.

Он должен всегда начинать описание документа, а завершать описание должен

тэг `</HTML>`.

Эти тэги означают, что все находящееся между ними, представляет единый HTML-документ. Сам по себе документ является обычным текстовым ASCII-файлом. Без этих тэгов браузер возможно будет не в состоянии идентифицировать формат документа и правильно его интерпретировать.

Между `<HTML>` и `</HTML>` располагается сам документ, который может состоять из двух разделов – раздела заголовка (`<HEAD>` и `</HEAD>`) и раздела содержательной части документа (`<BODY>` и `</BODY>`).

**Раздел HEAD HTML –документа.** Раздел **HEAD** является заголовком и не является обязательным тэгом. Однако хорошо составленный заголовок может быть весьма полезным. Цель введения заголовка - представление сопутствующей информации для программы, интерпретирующей данный документ. Тэги этого раздела, кроме названия документа – тэг `<TITLE>`- на экране не отображаются.

Обычно тэг `<HEAD>` следует сразу же за тэгом `<HTML>`.

Тэг `<TITLE>` является единственным и обязательным тэгом заголовка, и служит для задания названия документу. Название документа представляет строку текста, заключенную между `<TITLE>` и `</TITLE>`. Это название появляется в заголовке окна браузера (рекомендуется не более 60 символов). По умолчанию этот текст используется при задании закладки (bookmark) для документа. Название документа должно кратко характеризовать его содержимое, необходимо избегать близких названий (например, Noimage, Index и т.д.). Учитывая, что при загрузке документа первым делом отображается его название и далее выполняется загрузка основного содержимого документа, которое производится с форматированием (это требует определенного времени и может затянуться), необходимо задавать название документа, чтобы пользователь хотя бы лицезрел эту информативную строчку.

Часто HTML – документы связаны между собой, т. е. имеют ссылки друг на друга. Ссылки могут быть как абсолютные, так и относительные.

Абсолютные ссылки могут быть слишком громоздкими и перестают работать, если перемещен младший по иерархии документ.

Относительные ссылки легче вводить и обновлять, но и эта связь обрывается, если перемещен старший по иерархии документ. Оба вида связи могут нарушиться, если документ перенесен с одного компьютера на другой.

**Раздел BODY документа и его параметры.** Этот раздел включает в себя содержательную часть документа. Раздел должен начинаться с тэга `<BODY>` и заканчиваться тэгом `</BODY>`.

Строго говоря, наличие этих тэгов необязательно, т.к. браузеры могут определить начало содержательной части документа по тексту. Тэг **<BODY>** имеет ряд параметров, ни один из которых не является обязательным.

Параметры тэга **<BODY>** :

**BACKGROUND** – указывает на URL-адрес изображения, которое используется в качестве фонового;

**BGCOLOR** – определяет цвет фона документа;

**TEXT** – определяет цвет текста;

Цвета в языке HTML могут быть заданы с помощью шестнадцатеричного кода или с помощью цветовых имен. Цветовая база основывается на трех цветах – красном, зеленом и синем – и обозначается RGB. Для каждого цвета задается шестнадцатеричное значение в пределах от 00 до FF, что соответствует диапазону от 0 до 255 в десятичном исчислении, затем эти значения объединяются вместе в одно число и перед ними ставится знак #.

Например, # 800080 означает фиолетовый цвет.

Примеры

**<BODY TEXT = “# 000000”>** или **<BODY TEXT = black>**

**< BODY BGCOLOR = “# ffffff”>** или **< BODY BGCOLOR = WHITE >**

**< BODY LINK = “ # ff0000”>** или **<BODY LINK = RED>**

**< BODY LINK = “# 00FFFF” ALINK = “ # 80080”>** или

**< BODY VLINK = Aqua ALINK = PURPLE>** .

Все браузеры распознают 16 стандартных цветов. Это цвета

Black = # 000000	Maroon = # 800000
Silver = # C0C0C0	Red = # FF0000
Gray = # 808080	Purple = # 800080
White = # FFFFFFFF	Fuchsia = # FF00FF
Green = # 008000	Navy = # 000080
Lime = # 00FF00	Blue = # 0000FF
Olive = # 808000	Teal = # 008080
Yellow = # FFFF00	Aqua = # 00FFFF

Пример

```
<BODY
  BGCOLOR = AQUA
  TEXT = “#848484”>
```

Если параметр **BGCOLOR** задается для вывода фонового цвета с помощью задания названия цвета или его составляющих в шестнадцатеричном коде, то параметр **BACKGROUND** используется для задания фона странички с помощью изображения. В качестве изображения должен использоваться графический файл формата GIF или JPG. Фоновое изображение для HTML документа всегда заполняет все окно просмотра. Если размер изображения меньше размеров окна, то оно будет размножено по принципу мозаики. В качестве фонового изображения обычно берется небольшое изображение, для загрузки которого по сети не требуется значительного времени. Либо в качестве фона используется изображение в виде бледного рельефного логотипа.

Пример:

```
<BODY BACKGROUND = texture.gif BGCOLOR = gray>.
```

При создании странички рекомендуется всегда задавать цвет фона. Если задается также и фоновое изображение, то желательно подобрать цвета фона и изображения близкими друг к другу.

Пример:

```
<BODY
  TEXT =BLUE
  BGCOLOR = AQUA
BACKGROUND="http://www.foo.com/jkorpela/HTML3.2/wave.gif">
```

Пример с заданием заголовка и основной части документа:

```
<HTML>
  <HEAD>
    <TITLE> пример с заданием фона страницы </TITLE>
  </HEAD>
  <BODY BGCOLOR = YELLOW
    TEXT = BLACK>
  </BODY>
</HTML>
```

**Тэги форматирование текста.** Некоторые HTML-тэги, которые могут появляться в разделе BODY, называются тэгами уровня блока (block level), в то время как другие называются тэгами уровня текста (text level) или последовательными (inline).

Тэги уровня блока могут содержать в себе тэги уровня текста и другие тэги уровня блока. Тэги уровня текста могут содержать только данные или другие тэги уровня текста. Блочные тэги описывают структуру документа.

**Логическое и физическое форматирование.** Для форматирования текста в HTML-документах предусмотрены тэги, которые условно можно разделить на тэги логического и физического форматирования.

Тэги логического форматирования выполняют структурную разметку, которая не влияет на экранное представление фрагмента браузером. Поэтому такая разметка называется логической. Фрагменты с логическим форматированием браузеры отображают на экране определенным образом, заданным по умолчанию.

Тэг <B> - отображает текст полужирным шрифтом.

Например:

<B> Мультимедиа </B> включает в себя текст, графику, видео и звук.

На экране это будет выглядеть следующим образом:

**Мультимедиа** включает в себя текст, графику, видео и звук.

Тэг <I> - отображает текст курсивом.

Например,

<I> Мультимедиа </I> включает в себя интерактивный интерфейс.

*Мультимедиа* включает в себя интерактивный интерфейс.

Тэг <U> - отображает текст подчеркнутым .

Например,

Протокол в WWW, называется <U> Hyper Text Transfer Protocol </U>.

Протокол в WWW, называется Hyper Text Transfer Protocol .

Тэги <STRIKE> и <S> отображают текст, перечеркнутый по горизонтали (в версии 4.0. отмененный тэг).

Например,

Это пример <STRIKE> зачеркнутого </STRIKE> текста.

Это пример ~~зачеркнутого~~ текста

Тэг **<SUB>** - сдвигает текст ниже уровня строки и выводит его шрифтом меньшего размера. Удобно использовать для математических индексов.

Например,

элементами массива А являются: А **<SUB>** 11 **</SUB>** А**<SUB>**12 **</SUB>**

элементами массива А являются: А<sub>11</sub> А<sub>12</sub>

Тэг **<SUP>** - сдвигает текст выше уровня строки и выводит его шрифтом меньшего размера. Удобно использовать для задания степени чисел в математике или сносок.

Например:

А**<SUP>**2**</SUP>**+В**<SUP>**2**</SUP>**

А<sup>2</sup>+В<sup>2</sup>

Тэги форматирования могут быть вложенными друг в друга, при этом необходимо, чтобы один контейнер находился внутри другого целиком. Например,

Это **<B>**полужирный **</B>**шрифт

Это **<I>**курсив **</I>**шрифт

Этот текст **<B>****<I>**и полужирный и курсивный **</I>****</B>**.

Тэг **<FONT>**-указывает параметры шрифта.

Для тэга **<FONT>** могут быть заданы следующие параметры:

**FACE**

**SIZE**

**COLOR.**

Параметр **FACE** - служит для указания типа шрифта. Значением параметра является название шрифта, которое в точности должно совпадать с названием шрифта, имеющегося у пользователя на компьютере. Если такого шрифта нет, то будет использован шрифт, установленный по умолчанию. Можно указать несколько названий шрифтов, разделив их запятыми. Такой список просматривается слева направо, и если на компьютере нет текущего шрифта, выбирается из списка следующий и т.д.

Пример

**<HTML>**

**<HEAD>**

**<TITLE>** Назначение шрифтов **</TITLE>**

**</HEAD>**

**<BODY>**

**<I>** Это текст, записанный шрифтом по умолчанию **</I>** **<BR>**

**<FONT FACE = "Verdana", "Arial", "Helvetica">**

**<B>** А это пример задания шрифта **</B>**

**</FONT >**

**</BODY>**

**</HTML>**.

Параметр **SIZE** - служит для указания размеров шрифта в условных единицах от 1 до 7. Конкретный размер шрифта зависит от используемой программы просмотра. Принято считать, что размер "нормального" шрифта составляет 3 единицы. Размер шрифта задается как абсолютная величина (**SIZE=2**), так и относительная (**SIZE=+1**).

Пример

**<HTML>**

**<HEAD>**

**<TITLE>**Назначение размеров шрифтов**</TITLE>**

**</HEAD>**

**<BODY>**

**<FONT SIZE=1>** Шрифт размер 1 **</FONT>** **<BR>**

**<FONT SIZE=-1 >** Шрифт размер 2 **</FONT>** **<BR>**

**<FONT SIZE=3>** Шрифт размер 3 **</FONT>** **<BR>**

**<FONT SIZE=4>** Шрифт размер 4 **</FONT>** **<BR>**

```
<FONT SIZE=5> Шрифт размер 5 </FONT> <BR>
<FONT SIZE=+3> Шрифт размер 6 </FONT> <BR>
<FONT SIZE=7> Шрифт размер 7 </FONT> <BR>
</BODY>
</HTML>
```

Параметр COLOR – устанавливает цвет шрифта, который может задаваться с помощью названия цветов или в формате #RRGGBB

Пример:

```
<HTML>
  <HEAD>
    <TITLE> Назначение цвета шрифтам </TITLE>
  </HEAD>
  <BODY>
<FONT COLOR=green> Текст зеленого цвета </FONT> <BR>
<FONT COLOR=#FF0000> Текст красного цвета </FONT> <BR>
  </BODY>
</HTML>
```

Пример:

```
<HTML>
  <HEAD>
    <TITLE> Назначение шрифта, размера, цвета </TITLE>
  </HEAD>
  <BODY>
<FONT SIZE=+1 COLOR = YELLOW FACE = COURIER>
Цвет текста желтый, размер большой, тип шрифта – courier </FONT>
  </BODY>
</HTML>.
```

### ***Контрольные вопросы***

1. Что означает “разметить текст”.
2. Разделы HTML- документа.
3. Раздел заголовка HTML- документа. Зачем нужна информация из этого раздела.
4. Тег и атрибуты основного раздела HTML- документа.
5. Теги форматирования текста.
6. Тег <font> и его атрибуты

### ***Темы самостоятельных работ***

1. Создайте сайт, посвященный 2200-летию города Ташкента. Задайте фон в виде рисунка с изображением какой-либо достопримечательности нашего города. Разместите сайт в Интернете
2. Создайте сайт , посвященный истории нашего родного края и ее знаменитым людям. Представьте все виды обтекания текста вокруг снимков, размещенных на странице. Разместите сайт в Интернете.
3. Подготовьте электронный вариант реферата на тему ”Мой родной институт”.
- 4.Создайте титульный лист к реферату в программе MS Publisher. Свяжите текст или картинку на обложке с оглавлением вашего документа. Пункты оглавления свяжите внутренними ссылками с соответствующими разделами документа.

5. Создайте сайт , посвященный стране, язык и традиции , которой вы изучаете. Разместите сайт в Интернете.
6. Создайте сайт, посвященный жизни вашего института( факультета, кафедры). Разбейте страницу на окна так, чтобы в одном из них был о фото с изображением вашего института, в другом – список факультетов, в третьем – информация о деятельности каждого факультете.

### *Список рекомендованной литературы*

1. А.Матросов, А. Сергеев А., М. Чаунин. HTML 4.0 в подлиннике. БХВ-Петербург, 2004.
2. Николай Чувахин. Практическое руководство по HTML.  
<http://www.yi.com/home/ChuvakhinNikolai>

[На оглавление](#)

## Лекция 6

### Теги форматирования документов

**Цель лекции:** ознакомить с тегами , формирующими структуру документа, а именно: заголовками, списками, абзацами текста, ссылками.

**Ключевые слова:** форматирование документа, теги формирования абзаца, теги заголовков, предварительное форматирование текста, организация отступов, ссылка, внутренняя и внешняя ссылки, понятие списка, нумерованные и маркированные списки.

#### Содержание

Теги форматирования документа  
Организация ссылок в HTML – документах  
Теги построения списков на страницах.

**Форматирование HTML – документа.** Любые тексты имеют определенную структуру. Обычно элементами такой структуры являются заголовки, подзаголовки, списки, таблицы, абзацы и т.д.

Обычно абзацы выражают в тексте законченную мысль. В обычных текстовых редакторах абзацы создаются вводам символа перевода строки (нажатие клавиши <ENTER>). Однако при создании HTML – документов символы перевода строки не приводят к образованию абзаца. Так как символы перевода строки оригинального документа игнорируются, то текст, отлично смотревшейся в окне автора документа, может превратиться в сплошной неудобочитаемый текст в окне программы просмотра, поэтому в языке HTML для разделения текста на абзацы введен тэг <P>. Этот тэг необходимо помещать перед началом каждого абзаца. Закрывающий </P> тэг не обязателен. Браузеры интерпретируют несколько подряд расположенных тэгов <P> как один тэг и обычно отделяют абзацы друг от друга одной пустой строкой.

Параметр тэга <P>:

**ALIGN**

Возможные значения параметра горизонтального выравнивания:

**LEFT, RIGHT, CENTER, JUSTIFY.**

Пример:

<HTML>

```
<HEAD>
<TITLE> Использование тэга абзаца </TITLE>
</HEAD>
<BODY>
```

<P> Это обычный текстовый абзац, который содержит так много символов, что текст занимает несколько строчек. </P>

<P ALIGN=CENTER> Этот текстовый абзац будет выровнен по центру окна, так как содержит параметр горизонтального выравнивания по центру </P>

```
</BODY>
</HTML>
```

Тэг <BR> - служит для задания принудительного перевода строки (не имеет закрывающего тэга). Включение в текст тэга <BR> обеспечит размещение последующего текста с начала новой строки.

Например, такой подход можно использовать для создания структур типа списков без использования специальных тэгов разметки списка или при записи стихотворения и т.д.

С помощью тэга <BR> можно смоделировать вложенный абзац, т.к. тэг <P> не допускает вложенности.

Например,

```
<HTML>
<HEAD>
<TITLE> Использование перевода строки </TITLE>
</HEAD>
<BODY>
<P>
```

```
    Усердный в службе <BR>
    Не должен бояться <BR>
    Своего незнания, <BR>
    Ибо всякое новое дело <BR>
    Он прочтет.
```

```
</P>
```

```
<P align=right> Козьма Прутков </P>
```

```
</BODY>
</HTML>
```

Тэги заголовков внутри HTML – документа.

```
<H1> <H2> ... <H6>.
```

Для задания заголовков отдельным частям документа можно использовать шесть различных уровней тэгов <H1>, <H2>, <H3>, <H4>, <H5>, <H5>, <H6> и соответственно, </H1>, </H2>, </H3>, </H4>, </H5>, </H6>.

Самым крупным является заголовок под номером 1 (заголовок верхнего уровня), самым мелким – заголовок с номером 6. Тэги заголовка являются тэгами уровня блока, поэтому с помощью них нельзя размечать отдельные слова текста для увеличения их размера. При использовании тэгов заголовков, до и после заголовка вставляется пустая строка, поэтому тэги абзаца или перевода строки здесь не требуются.

Параметр тэгов заголовков:

**ALIGN**

Значения параметра:

**LEFT, RIGHT, CENTER, JUSTIFY** (по умолчанию LEFT)

Пример:

```
<HTML>
<HEAD>
<TITLE> Примеры заголовков</TITLE>
```

```

</HEAD>
<BODY>
<H1> Заголовок размера 1 </H1>
<H2> Заголовок размера 2 </H2>
<H3> ALIGN=CENTER> Заголовок размера 3 </H3 >
<H4 ALIGN=RIGHT> Заголовок размера 4 </H4>
<H5> Заголовок размера 5 </H5>
<H6> Заголовок размера 6 </H6>
    ...Здесь основной текст документа
</BODY>
</HTML>.

```

Тэг **<HR>** - горизонтальная линия.

Используется для визуального подчёркивания законченности той или иной области страницы. Этот тэг не требует закрывающего тэга. До и после линии автоматически вставляется пустая строка.

Параметры тэга **<HR>**:

**ALIGN** - LEFT, RIGHT, CENTER (по умолч.);  
**WIDTH** - устанавливает длину линии в пикселях или в процентах по отношению к ширине окна;  
**SIZE** - устанавливает толщину линии в пикселях  
**NOSHADA** - отменяет рельефность линии  
**COLOR** - указывает цвет линии

Пример:

```

<HTML>
<HEAD> <TITLE> Горизонтальная черта </TITLE>
</HEAD>
<BODY>
<P> Текст, помещенный до горизонтальной черты </P>
<HR ALIGN= CENTER WIDTH=70% NOSHADE>
<P> Далее размещается сам текст</P>
</BODY>
</HTML>.

```

Тэг **<PRE>** . Использование предварительно отформатированного текста. Для включения отформатированного текста, выполненного традиционным способом при помощи символов перевода строки, необходимого количества пробелов, символов табуляции и т.д., предусмотрен специальный тэг- контейнер **<PRE>** . Текст, размеченный тэгом **<PRE>** , будет отображаться в таком виде, как он выглядит в обычном текстовом редакторе. Для отображения всегда будет использоваться моноширинный шрифт.

Одним из вариантов использования этого тэга являются таблицы, построенные без применения специальных тэгов разметки таблиц. Другим является вывод больших блоков.

Пример.

```

<HTML>
<HEAD> <TITLE> Преформатированный текст </TITLE>
</HEAD>
<BODY>
<H3> Таблица чисел </H3>
<PRE>
0.5138707      0.1757256      0.3086337
0.858954      0.3896298      0.2777221
0.8229621     0.1519211      0.6254769
</PRE>
</BODY>

```

</HTML>

Параметр тэга <PRE>:

**WIDTH**

Значение параметра может быть задано в пикселях или в процентах и указывает максимальную длину строки преформатированного текста.

Тэг <BLOCKQUOTE>. Используется для выделения цитат из основного текста. Этот тэг является контейнером и может содержать любые тэги форматирования. Текст, размеченный с помощью <BLOCKQUOTE>, при отображении отделяется от основного текста пустыми строчками и выводится с небольшим отступом справа.

Пример

```
<HTML>
<HEAD>
<TITLE>Блок цитирования использование перевода строки </TITLE>
</HEAD>
<BODY>
<P>
    В рукописном варианте
    <I> <B> Медного всадника </B> </I> сказано
```

<BLOCKQUOTE>

```
Над омраченным Петроградом <br>
Дышал ноябрь осенним хладом. <br>
Плеская шумною волной,<br>
Нева металась...
< P ALIGN=RIGHT > А.С.Пушкин </P>
</BLOCKQUOTE>
</BODY>
</HTML>.
```

Тэг <ADDRESS>. Применяется для идентификации автора документа и для указания адреса автора. Обычно этот тэг размещается в начале или в конце самого документа. Часто в этом тэге указывают дату создания и последнего обновления документа.

Тэг <ADDRESS> является контейнером.

Пример

```
<HTML>
<HEAD>
<TITLE> Адресный блок </TITLE>
</HEAD>
<BODY>
<P> Информация для контакта </P>
<ADDRESS>
<P>
    Jukka Korpela, m.s. (Math) <BR>
    Helsinki University of Technology Computing Centre <BR>
    FIN – 02150 Espoo <BR>
    FINLAND
    </P>
<P>
    Telephone International + 35894514319 </P>
<P> Electronic Mail: J. Korpela@hut.fi </P>
</ADDRESS>
</BODY>
```

</HTML>.

Тэг **комментариев в документ** <! - - ...- -> Комментарии могут состоять из произвольного числа строк и должны начинаться тэгом <! - - и заканчивается тэгом - ->. Все, что заключено внутри тэгов не будет отображаться при просмотре странички. Обычно комментарии используются авторами для заметок, предназначенных только для собственного использования. Существует ещё один тэг- контейнер для записи комментариев - <COMMENT>.

**Организация ссылок в HTML – документах.** На Web – страницах могут быть размещены различные типы мультимедийных объектов В гипертекстовых документах в дополнение к самому тексту используются видео – и аудиообъекты, а в качестве ссылок к ним применяют не только тексты, но и разнообразную графику.

Ссылки состоят из двух частей. Первая – это то, что вы видите на Web – страничке, она называется указателем ссылки (anchor). Вторая – дает инструкцию браузеру и называется адресной частью ссылки (URL - адрес).

При щелкании мышкой по указателю ссылки браузер загружает документ, месторасположение которого задается URL – адресом. Указатели ссылки бывают двух типов: текстовые и графические. Это может быть слово, группа слов или изображение. Цвет текстового указателя может регулироваться автором странички или установками программы – просмотра.

Например.

```
<A HREF="example.htm">Этот текст является указателем ссылки </A>
```

или

```
<A HREF="example.htm"> <IMG SRC="picture gif"> </A>
```

Первый является примером использования текстового указателя в ссылке, второй – графического указателя. По принципу действия обе ссылки ничем не отличаются.

Второй частью ссылки является URL - адрес, т.е. адрес Web – страницы, которая будет загружена при щелчке мышью по указателю. Указание адреса может быть относительным или абсолютным.

Для указания браузеру указателя ссылки, а также адреса документа, на который вы ссылаетесь, используется тэг <A>.

Тэг <A> имеет обязательный параметр – HREF, значением которого является URL - адрес.

Например,

```
<A HREF=http://www.server.com/home/index.htm> указатель ссылки </A>
```

Ссылки можно организовывать как на другие документы, так и на разные части текущего документа. Например, большой документ читается лучше, если он имеет оглавление со ссылками на соответствующие разделы.

Для построение **внутренней** ссылки сначала нужно создать указатель, определяющий место назначения. При размещении указателя ему дается имя при помощи параметра NAME тэга <A>. При этом параметр HREF не используется.

Например,

```
<A NAME = "glava_5"> </A>
```

После определения можно создать ссылку на него. Для этого в параметре HREF нужно поместить имя ссылки с префиксом "#", говорящим о том, что это внутренняя ссылка.

Например,

```
<A HREF = "#glava_5"> глава_5</A>
```

Если ссылки указывают на другую Web – страницу, то она выводится в окне браузера. Если же ссылка указывает на документ иного типа, программа просмотра принимает документ и затем решает, что с ним делать дальше. Следующими действиями браузерами могут быть:

- Браузеры знает этот тип документа и умеет с ним обращаться. Например, если ссылки выполняется на графический файл формата GIF, то программа просмотра очистит окно и загрузит, указанное изображение.

- Браузер не распознает тип принятого документа. В этом случае браузер обращается к вспомогательным программам, имеющимся на компьютере пользователя. Если такая программа найдется, браузер запустит ее.

Например, если ссылка выполняется на видеофайл формата AVI, браузер загрузит файл, найдет программу для демонстрации AVI-файлов и запустит ее.

На Web – страничке также можно создать ссылку на электронный адрес почты, добавив его словом “mailto:”.

Например,

```
<A HREF="mailto:somebody@mail.info.ru"> присылайте ваши отзывы </A>
```

Рассмотрим пример, где использованы рассмотренные выше тэги.

```
<HTML>
  <HEAD>
    <TITLE> Персональная страничка </TITLE>
  </HEAD>
  <BODY>
<H1 ALIGN=CENTER> Добро пожаловать! </H1>
<HR>
    основное содержание страницы
  <HR>
  <ADDRESS>
    Последнее обновление выполнено 1.11.2005
    </ADDRESS>
  <A HREF="mailto:mne@mail.ru"> присылайте ваши отзывы </A>
  </BODY>
</HTML>
```

В качестве ссылок могут использоваться также графические изображения, т.е. иллюстрации могут выполнять роль указателей гипертекстовых связей. Для обеспечения работы изображения в качестве ссылки достаточно включить изображение внутрь контейнера тега <A>.

**Списки.** В языке HTML предусмотрен специальный набор тэгов для представления информации в виде списков. Использование списков удобно в следующих случаях:

- описание сложных пошаговых процессов;
- расположение информации в стиле оглавления;
- объединение фрагментов информации в единую структуру.

В языке HTML предусмотрены следующие основные типы списков:

- маркированный;
- нумерованный;
- список определений.

Маркированный список (неупорядоченный – Unordered List (**UL**))

В маркированном списке для выделения элементов используются специальные символы, называемые маркерами или буллетами.

Для создания маркированного списка необходимо использовать парный тэг- контейнер <UL>, внутри которого располагаются все элементы списка. Эти тэги автоматически отделяют список от основного содержимого документа пустыми строками.

Каждый элемент списка должен начинаться тэгом <LI>(List Item).

Пример

```
<HTML>
  <HEAD>
<TITLE>Маркированный список</TITLE>
```

```

</ HEAD>
<BODY>
  <UL>
    <B>Знаки Зодиака</ B>
  <LI>Овен <! -- закрывающий тэг не обязателен-->
  <LI>Телец
  <LI>Рак
  <LI>Лев
  <LI>Дева
  </UL>
</ BODY>
</ HTML>

```

Параметры тэга <UL>:

#### КОМПАКТ TYPE

Параметр **КОМПАКТ** не имеет значений и указывает на то, что данный список следует выводить в компактном виде, т.е. может быть уменьшен шрифт или расстояние между строками списка.

Параметр **TYPE** может иметь следующие значения:

Disc – маркеры отображаются закрашенными кружками (по умолчанию);

Circle – маркеры отображаются незакрашенными кружками;

Square – маркеры отображаются закрашенными квадратиками (в 4.0 , незакрашенный квадратик).

Значение параметра **TYPE** можно задавать также для отдельного элемента списка.

Пример.

```

<HTML>
  <HEAD>
    <TITLE>Маркированный список</ TITLE>
  <BODY>
    <H3>Remember to buy</H3>
    <UL TYPE = CIRCLE КОМПАКТ>
    <LI>milk
    <LI>bread
    <LI>apples
    </ BODY>
  </ HTML>

```

Нумерованный список (упорядоченный – Ordered List (**OL**)).

Другим типом списка в языке HTML является нумерованный список. В отличие от маркированного списка, перед элементами нумерованного проставляются порядковые номера. Вид нумерации зависит от браузера и может задаваться параметрами тэга списка.

Для создания нумерованного списка используется парный тэг-контейнер <OL>, внутри которого располагаются все элементы списка.

Тэги <OL> и </OL> обеспечивают перевод строки до и после списка, отделяя список от основного содержания документа. Каждый элемент списка должен начинаться с тэга <LI>.

Параметры тэга <OL>:

**КОМПАКТ** ( то же, что и для маркированных списков)

#### TYPE

#### START

TYPE используется для задания вида нумерации списка и может принимать значения:

TYPE = A – задает номера в виде прописных букв латинского алфавита;

TYPE = a – задает номера в виде строчных латинских букв;

TYPE = I - задает номера в виде больших римских цифр;  
TYPE = i - задает номера в виде маленьких римских цифр;  
TYPE = 1 - задает номера в виде арабских цифр (используется по умолчанию).

После номера элемента списка всегда дополнительно выводится точка.

Для указания типа нумерации для отдельного элемента списка можно указать параметр TYPE со значением в самом тэге <LI>.

Параметр START тэга <OL> позволяет начать нумерацию не с единицы. В качестве значения параметра START всегда должно указываться натуральное число, вне зависимости от вида нумерации.

Например,

```
< TYPE = A START = 5 >
```

Такая запись определяет нумерацию списка, начиная с 5.

Тэг <LI> разрешает для нумерованных списков использовать параметры TYPE и VALUE . Значение параметра VALUE позволяет изменить номер данного элемента списка, при этом изменяется и нумерация всех последующих элементов списка. Типичным является список с пропуском некоторых элементов.

Пример:

```
<HTML>
  <HEAD>
<TITLE> Нумерованный список </TITLE>
  </HEAD>
  <BODY>
<B> Самые яркие звезды, видимые с Земли: </B>
<OL>
<LI>Сириус
<LI>Канопус
<LI>Арктур
<LI>Альфа Центавра
<LI>Вега
<LI>Капелла
<LI>Ригель
<LI>Порцион
<LI>Ахернар
<LI>Бета Центавра
<LI>Бетельгейзе
<LI>Альдебаран
<BR>...
<LI VALUE = 58 > Мицар <BR>...
<LI VALUE = 75 > Полярная
</OL>
  </BODY>
</HTML>
```

### *Контрольные вопросы*

1. Что означает “отформатировать документ”.
2. Какие теги организации абзаца вы знаете.
3. Какие теги используются для создания заголовков текста
4. Как можно связать документы между собой.
5. Какой атрибут является обязательным в тэге ссылки.
6. Как можно установить закладку на страничку
7. Можно ли создать вложенные списки.

### *Темы самостоятельных работ*

1. Создайте сайт, посвященный 2200-летию города Ташкента. Задайте фон в виде рисунка с изображением какой-либо достопримечательности нашего города. Разместите сайт в Интернете
2. Создайте сайт , посвященный истории нашего родного края и ее знаменитым людям. Представьте все виды обтекания текста вокруг снимков, размещенных на странице. Разместите сайт в Интернете.
3. Подготовьте электронный вариант реферата на тему ”Мой родной институт”.
- 4.Создайте титульный лист к реферату в программе MS Publisher. Свяжите текст или картинку на обложке с оглавлением вашего документа. Пункты оглавления свяжите внутренними ссылками с соответствующими разделами документа.
- 5.Создайте сайт , посвященный стране, язык и традиции , которой вы изучаете. Разместите сайт в Интернете.
- 6.Создайте сайт, посвященный жизни вашего института( факультета, кафедры). Разбейте страницу на окна так, чтобы в одном из них был о фото с изображением вашего института, в другом – список факультетов, в третьем – информация о деятельности каждого факультете.

### *Список рекомендованной литературы*

1. А.Матросов, А. Сергеев А., М. Чаунин. HTML 4.0 в подлиннике. БХВ-Петербург, 2004.
2. Николай Чувахин. Практическое руководство по HTML.  
<http://www.yi.com/home/ChuvakhinNikolai>

[На оглавление](#)

## Лекция 7

### **Таблицы. Графические объекты.**

**Цель лекции :** ознакомить с тегами построения и оформления таблиц, с возможностями вставки графических объектов на страницу и применением соответствующих ссылок.

**Ключевые слова:** теги создания таблицы, строки таблицы, ячейки в строке, создание названия таблицы, вставка изображения, атрибуты тега изображения.

#### **Содержание**

Создание таблиц.

Включение заголовка к таблице.

Вставка изображений на страничку.

Установка ссылок на изображения.

**Создание таблиц в языке HTML.** Традиционно в HTML таблицы используются как метод представления данных. Но, кроме такого использования, в HTML таблицы служат средством форматирования Web-страницы.

Например, ниже представлен типичный пример использования таблицы для представления числовых данных.

Таблица 1.

Валюта	Макс. покупка	Мин. продажа	Сред. покупка	Сред. продажа
USD	25.10	25.40	24.87	25.55
DEM	13.00	13.59	12.32	13.76
FIM	4.35	4.47	4.08	4.52

Таблица разбита на 4 строки и 5 столбцов, т.е. на этой таблице представлены явно ряды строк и столбцов.

Использование таблиц не ограничивается только заданием данных, но состоит и в организации расположения разнообразных элементов на странице, которые могут состоять из простого текста, изображений, других таблиц и т.д., т.е. использования таблицы лишь в целях форматирования документа. При просмотре такого документа обычно сразу не видно, что для построения была использована таблица, так как рамки вокруг ячеек не прорисовываются.

Рассмотрим сначала минимальный набор тэгов, необходимый для организации простейшей таблицы. Описание таблицы должно располагаться внутри раздела <BODY>. Документ может содержать произвольное количество таблиц, причем допускается вложенность таблиц друг в друга. Каждая таблица должна начинаться с тэга <TABLE> и заканчиваться тэгом </TABLE>. Обычно таблица состоит из одной или нескольких строк, в каждой из которых задаются данные для отдельных ячеек.

Каждая строка начинается с тэга <TR>(Table Row) и завершается тэгом </TR>. Отдельная ячейка в строке обрамляется парой тэгов <TD> (Table Data) и </TD> или <TH>(Table Header) и </TH>. Тэг <TH> обычно используется для ячеек - заголовков таблицы, а <TD> --- для ячеек - данных. Различие заключается в типе шрифта и расположении данных внутри ячейки. Содержимое ячеек типа <TH> отображается полужирным шрифтом и располагается по центру. Содержимое ячеек типа <TD> отображается выровненным влево и посередине вертикального направления.

Количество строк в таблице определяется числом тэгов <TR>, а количество столбцов – максимальным количеством <TD> или <TH> среди всех строк.

Часть ячеек могут не содержать данные, такие ячейки описываются парой тэгов <TD> и </TD>.

Построение таблиц, в которых в разных строчках располагается различное количество столбцов одного и того же размера не разрешается.

Пример простейшей таблицы:

```
<HTML>
  <HEAD>
    <TITLE> Простейшая таблица </TITLE>
  </HEAD>
<BODY>
  <TABLE>
    <TR>
    <TD> Ячейка 1 строка 1 </TD>
    <TD> Ячейка 2 строка 1 </TD>
    </TR>
    <TR>
    <TD> Ячейка 1 строка 2 </TD>
```

```

<TD> Ячейка 2 строка 2 </TD>
</TR>
</TABLE>
</BODY>
</HTML>

```

Ячейка 1 строка 1	Ячейка 2 строка 1
Ячейка 1 строка 2	Ячейка 2 строка 2

Таблица может иметь заголовок. Для задания заголовка всей таблице используется пара тэгов <CAPTION> и </CAPTION>. Описание заголовка должно располагаться внутри тэгов <TABLE> и </TABLE> сразу же после <TABLE> и до первого <TR>.

Параметры тэга <CAPTION> :

VALIGN

Значения параметра вертикального выравнивания:

TOP - заголовок над таблицей (по умолчанию);

BOTTOM - подпись под таблицей.

Заголовок таблицы всегда располагается по центру горизонтального направления.

ALIGN

Значения параметра горизонтального выравнивания:

LEFT - выравнивание заголовка влево;

RIGHT - выравнивание заголовка вправо;

CENTER - выравнивание заголовка по центру (по умолчанию).

Обычно в заголовке можно давать простой текст, однако между тэгами <CAPTION> и </CAPTION> можно использовать все тэги, употребляемые в <BODY>.

Например,

```

<CAPTION VALIGN = BOTTOM>
    <B> <FONT COLOR=ORANGE FACE=FORTE SIZE=3>
    Заголовок внизу таблицы
    </FONT></B>
</CAPTION>.

```

Параметры тэга <TABLE>:

BORDER

BORDERCOLOR

CELLSPACING

CELLPADDING

WIDTH

HEIGHT

ALIGN

BGCOLOR

BORDER - управляет изображением рамки вокруг каждой ячейки и вокруг всей таблицы. По умолчанию рамки не рисуются. Чтобы задать рамки в тэге <TABLE> необходимо задать параметр BORDER. При этом для рамки вокруг таблицы можно назначать толщину рамки в пикселях, а для рамок вокруг ячеек всегда принимается минимальная толщина, равная 1.

Например,

```

<TABLE BORDER>

```

<TABLE BORDER=10>

Общий размер таблицы при отсутствии или наличии параметра BORDER не изменяется (исключением составляет BORDER=0).

Чтобы расположить ячейки как можно ближе друг к другу, необходимо задать параметр BORDER=0, что означает отсутствие рамок. Для браузера Internet Explorer параметр BORDER=0 является значением по умолчанию, т.е. <TABLE> и <TABLE BORDER=0> являются идентичными.

BORDERCOLOR- задает цвет рамок таблицы.

<CELLSPACING > - определяет расстояние между смежными ячейками (т.е. между рамками ячеек) как по горизонтали, так и по вертикали. Форма записи этого параметра CELLSPACING = num, где num - численное значение в пикселях (по умолчанию num =2). При задании CELLSPACING = 0 рамки смежных ячеек сливаются и создают единую линию сетки таблицы.

Например,

<BODY>

<TABLE CELLSPACING=0 BORDER><CAPTION VALIGN = TOP>

<B> <FONT COLOR=ORANGE FACE=FORTE SIZE=3>

Заголовок таблицы

</FONT></B>

</CAPTION>

<TR>

<TD> Ячейка 1 строка 1 </TD>

<TD> Ячейка 2 строка 1 </TD>

</TR>

<TR>

<TD> Ячейка 1 строка 2 </TD>

<TD> Ячейка 2 строка 2 </TD>

</TR>

</TABLE>

</BODY>

### Заголовок таблицы

Ячейка 1 строка 1	Ячейка 2 строка 1
Ячейка 1 строка 2	Ячейка 2 строка 2

Еще один пример с параметром CELLSPACING, отличным от нуля.

<BODY>

<TABLE CELLSPACING=4 BORDER><CAPTION VALIGN = TOP>

<B> <FONT COLOR=ORANGE FACE=FORTE SIZE=3>

Заголовок внизу таблицы

</FONT></B>

</CAPTION>

<TR>

<TD> Ячейка 1 строка 1 </TD>

<TD> Ячейка 2 строка 1 </TD>

</TR>

<TR>

<TD> Ячейка 1 строка 2 </TD>

<TD> Ячейка 2 строка 2 </TD>

</TR>

</TABLE>

</BODY>

Ячейка 1 строка 1	Ячейка 2 строка 1
Ячейка 1 строка 2	Ячейка 2 строка 2

### Заголовок внизу таблицы

CELLPADDING - определяет расстояние между рамкой ячейки и данными внутри ячейки. Форма записи CELLPADDING = num, где num – численное значение в пикселях. По умолчанию num =1, значения num =0 приводит к касанию части текста ячейки с рамкой.

Например, при включении параметра CELLPADDING в предыдущий пример со значением 0 и со значением, отличным от 0, мы получим таблицу в следующих видах:

<TABLE CELLSPACING=4 CELLPADDING=0 BORDER>

Ячейка 1 строка 1	Ячейка 2 строка 1
Ячейка 1 строка 2	Ячейка 2 строка 2

<TABLE CELLSPACING=4 CELLPADDING=7 BORDER>

Ячейка 1 строка 1	Ячейка 2 строка 1
Ячейка 1 строка 2	Ячейка 2 строка 2

Параметры CELLSPACING и CELLPADDING влияют на отступы одновременно как по горизонтали, так и по вертикали.

Наиболее компактная таблица будет получена при задании следующего описания

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0>

Ячейка 1 строка 1	Ячейка 2 строка 1
Ячейка 1 строка 2	Ячейка 2 строка 2

WIDTH – используется для принудительного задания ширины таблицы. Форма записи WIDTH= num или WIDTH= num %, где num – численное значение ширины всей таблицы в пикселях или в % от всего размера окна. Допустимо num > 100%.

HEIGHT – используется для принудительного задания высоты таблицы. Форма записи HEIGHT = num или num % и определяется аналогично заданию num в WIDTH .

В большинстве случаев динамическое определение размеров таблицы дает в результате эстетически соразмерное изображение с эффективным использованием реальной площади окна просмотра.

ALIGN – определяет горизонтальное размещение таблицы в области просмотра.

Допустимые значения:

LEFT

RIGHT

CENTER

По умолчанию таблица выровнена влево. Присутствие этого параметра в тэге <TABLE> определяет не только месторасположение таблицы, но и задает обтекание текстом с противоположной стороны.

## Пример

```
<HTML>
  <HEAD>
<TITLE> Таблица с обтеканием текста</TITLE>
  </HEAD>
<BODY>
<TABLE ALIGN =LEFT WIDTH= 70 % BORDER CELLSPACING=0>
<CAPTION><H3>Популярные имена</H3></ CAPTION >
<TR><TD VALIGN=TOP>
Абрам <BR> Александр <BR> Алексей <BR> Альберт <BR> Анатолий <BR>
Андрей <BR> Аркадий <BR> Борис <BR> Вадим <BR> Валентин <BR>
</TD>
<TD VALIGN=TOP>
Евгений <BR> Ефим <BR> Иван <BR> Игорь <BR> Илья <BR> Иосиф <BR>
Константин <BR>Лев<BR>Леонид<BR>Михаил<BR>
</TD>
</TR>
</TABLE>
<BR><BR><BR><BR>
Приведенные данные получены <BR>
на основе анализа выборки,<BR>
содержащей сведения о 5000 мужчин<BR>
в возрасте старше 18 лет.
</BODY>
</HTML>.
```

Приведем еще один пример с использованием вышеописанных параметров.

```
<HTML>
  <HEAD>
    <TITLE> Северные страны </TITLE>
  </HEAD>
<BODY>
<TABLE ALIGN =RIGHT WIDTH= 40%
BORDER=10 CELLSPACNG=10 CELLPADDING=10>
<CAPTION><B><I>Территория северных стран</I></B></ CAPTION>
<TR>
<TH>Страна</TH>
<TH>Общая площадь</TH>
<TH>Площадь суши</TH>
</TR>
<TR>
<TD>Дания</TD>
<TD>43000</TD>
<TD>42370</TD>
</TR>
<TR>
<TD>Финляндия</TD>
<TD>337030</TD>
<TD>305470</TD>
</TR>
```

```

<TR>
<TD>Норвегия</TD>
<TD>324220</TD>
<TD>307860</TD>
</TR>
<TR>
<TD>Швеция</TD>
<TD>449964</TD>
<TD>410928</TD>
</TR>
</TABLE>
</BODY>
</HTML> .

```

Ниже приводится результат выполнения этого HTML- кода.

### *Территория северных стран*

Страна	Общая площадь	Площадь суши
Дания	43000	42370
Финляндия	337030	305470
Норвегия	324220	307860
Швеция	449964	410928

### **Вставка изображений**

Тэг <IMG> - используется для включения в документ изображения. Его обязательным параметром является SRC , определяющий URL – адрес файла с изображением.

Например,

```
<IMG SRC = picture.gif>.
```

При включении изображения в документ можно указывать его расположение относительно текста или других элементов страницы. Для этих целей используется параметр ALIGN тэга <IMG>. Этот параметр может принимать следующие значения:

TOP- верхняя граница изображения выравнивается по самому высокому элементу строки;

BOTTOM – выравнивание нижней границы изображения по базовой линии текущей строки;

MIDDLE – выравнивание середины изображения по базовой линии текущей строки;

LEFT – изображение прижимается к левому полю окна. Текст обтекает изображение справа;

RIGHT – изображение прижимается к правому полю окна. Текст обтекает изображение с левой стороны.

Пример

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> Выравнивание изображений </TITLE>
```

```

</HEAD>
<BODY>
Выравнивание <IMG SRC = tagle.gif ALIGH =top> по верхнему краю
<P>
Выравнивание <IMG SRC = tagle.gif ALIGH = bottom> по базовой линии </P>
<P> Выравнивание <IMG SRC = tagle.gif ALIGH = middle> по средней линии </P>
</BODY>
</HTML>

```

Выравнивание



по верхнему краю

Выравнивание



базовой линии

Выравнивание

по средней линии



Теперь рассмотрим использование изображения в качестве ссылки. Например:

```
<A HREF = My_doc.html><IMG SRC = map.gif></A>
```

Любая часть такого изображения будет работать как указатель ссылки на документ MY\_doc.html.

Тэг встраивания изображения <IMG> имеет ряд необязательных параметров:

HEIGHT – указывает размер изображения по вертикали при отображении;

WIDTH - указывает размер изображения по горизонтали при отображении.

Значения параметров могут указываться как в пикселях так и в процентах от размеров окна просмотра. Значения параметров могут не совпадать с истинными размерами изображения. В этом случае браузер автоматически при загрузке выполняет его перемасштабирование. Заметим, параметры изображения необходимо задавать аккуратно, чтобы сохранить пропорции рисунка. При задании размеров один из параметров может быть опущен. В этом случае при загрузке рисунка второй параметр автоматически вычисляется исходя из условий сохранения пропорций.

Пример

```
<HEAD>
```

```
<TITLE>Задание размеров изображений</TITLE>
</HEAD>
<BODY>
<IMG SRC=spb.gif WIDTH=300>
<IMG SRC=spb.gif WIDTH=150 HEIGHT=175>
<IMG SRC=spb.gif WIDTH=75>
</BODY>
</HEAD>
```

Для тэга <IMG> можно задавать параметры, значения которых определяют отступы от изображения, оставляемые пустыми.

HSPACE- отступ по горизонтали;

VSPACE- отступ по вертикали.

Значения этих параметров могут указываться также в пикселях или в процентах.

Пример

```
<HEAD>
<TITLE>Задание отступов вокруг изображений</TITLE>
</HEAD>
<BODY>
<IMG SRC=spb.gif ALIGN=left HSPACE=20 VSPACE=20>
</BODY>
</HEAD>
```

Изображение, встраиваемое в страницу, можно поместить в рамку различной толщины. Для этого используется параметр BORDER тэга <IMG>. В качестве значения параметра указывается число в пикселях.

Пример

```
<HEAD>
<TITLE>Задание рамки вокруг изображения</TITLE>
</HEAD>
<BODY>
<IMG SRC=spb.gif>
<IMG SRC=spb.gif ALIGN=left HSPACE=20 VSPACE=20 BORDER=1>
<IMG SRC=spb.gif BORDER=5>
<IMG SRC=spb.gif BORDER=10 HEIGHT=175>
</BODY>
```

Параметр ALT тэга <IMG>.

Для пользователей, работающих в режиме отключения загрузки изображений, параметр ALT дает возможности получить текстовую информацию об изображении. Современные браузеры этот текст будут отображать в качестве подсказки (tooltip) при помещении курсора мыши в область изображения.

Например,

```
<IMG SRC=spb.gif ALT="герб города" WIDTH=150 HEIGHT=174>
```

При отключенной загрузке изображения будет показана рамка с альтернативным текстом "герб города".

### ***Контрольные вопросы***

1. Создание таблиц для представления данных на странице. Атрибуты таблицы.
2. Создание таблиц для форматирования документа.
3. Тег <CAPTION> и его атрибуты.

4. Обязательный атрибут при вставке изображения на страницу.
5. Чем вставка изображения в теге <IMG> отличается от вставки картинки в теге <BODY>.
6. Как превратить изображение на странице в ссылку.

### ***Темы самостоятельных работ***

1. Фреймы. Создание фреймов.
2. Организация взаимодействия между фреймами.
3. Создайте сайт с перечнем литературы по какому-либо предмету и с возможностью просмотра каждой выбранной

### ***Список рекомендованной литературы***

1. А.Матросов, А. Сергеев А., М. Чаунин. HTML 4.0 в подлиннике. БХВ-Петербург, 2004.
2. Николай Чувахин. Практическое руководство по HTML.  
<http://www.yi.com/home/ChuvakhinNikolai>, 2007.

[На оглавление](#)

## **Лекция 8**

### **Основные понятия базы данных. Назначение и возможности Ms Access.**

**Цель лекции:** ознакомить с понятиями баз данных и системой управления базами данных, с понятиями ключа и организации связи между объектами БД.

**Ключевые слова:** база данных, СУБД, таблицы базы данных, связи между таблицами, первичный ключ, запросы, формы, отчеты.

#### **Содержание**

Основы проектирования.

Понятие базы данных. СУБД.

Объекты БД.

Главный объект-таблица. Операции над данными в таблицах.

Связи между таблицами.

Определение ключевого поля.

#### **Основы проектирования.**

Жизненный цикл информационных систем можно разбить на три стадии:

- проектирование (бумажное)
- программная реализация
- эксплуатация.

На стадии проектирования :

- собирается информация о предметной области, о ее структуре, анализируются и выделяются объекты исходной системы и связи между ними;
- для каждого объекта выясняются свойства и характеристики, которым назначаются поля, составляются исходные таблицы;
- для каждого объекта назначаются первичные ключи и проводится нормализация.

На стадии реализации разрабатывается приложение на компьютере, связанное

- с описанием таблиц средствами СУБД и введением их в компьютер;
- с разработкой интерфейсов работы с БД, т.е. экранных форм для ввода и отображения данных, отчетов для печати данных на бумагу, запросов для отбора данных и т.д.;
- с порядком ведения и поддержания базы в рабочем состоянии.

На стадии эксплуатации таблица

- заполняется реальными исходными данными;
- после чего происходит непосредственное ее использование.

Пример проектирования .

**Постановка задачи:** проектируется база, в одной из таблиц которой необходимо хранить информацию о преподавателях и о предметах, которые они читают с указанием факультета.

Кодпреп	ФИО	Телефон	Кодпред	Название	Факультет
1	Акмалов Н.С.	534-00-97	1	Экономика	МЭО
2	Мебуке М.Т.	534-95-00	1	Экономика	МО
3	Карба А.О.	490-00-12	2	Математика	Филология
4	Акмалов Н.С.	534-00-97	3	Информатика	История

Разобьем таблицу так, чтобы получить «хорошую» структуру данных, которая характеризуется непротиворечивостью, избыточностью информации и очевидностью операций удаления, добавления и изменения. В итоге получим 3 таблицы:

## Преподаватель

Кодпреп	ФИО	Телефон
1	Акмалов Н.Е.	534-00-97
2	Мебуке М.Т.	534-95-00
3	Карба А.О.	490-00-12

## Предмет

Кодпред	Название
1	Экономика
1	Экономика
2	Математика
3	Информатика

## Читает

В MS Access

создание связей между  
позволяет использовать  
нескольких таблиц

Это экономит память,

скорость обработки данных, позволяет избежать ненужного дублирования и лишних ошибок.

Кодпреп	Кодпред	факультет
1	1	МЭО
2	1	МО
3	2	Филология
1	3	Информатика

возможно

таблицами, что

данные из

одновременно.

увеличивает

## Понятие базы данных.

Базы данных — это совокупность структур, предназначенных для хранения больших объемов информации и программных модулей, осуществляющих управление данными, их выборку, сортировку и другие подобные действия. Информация базы данных хранится в одной или нескольких **таблицах**. Любая таблица с данными состоит из набора однотипных **записей**, расположенных друг за другом. Они представляют собой строки таблицы, которые можно добавлять, удалять или изменять. Каждая запись является набором именованных полей, или ячеек, которые могут хранить самую разнообразную информацию, начиная от даты рождения и заканчивая подробным описанием кулинарного рецепта. Однотипные поля разных записей образуют столбец таблицы.

Для создания БД используются специальные программные инструментальные системы - системы управления базами данных(СУБД).

Основное назначение СУБД в следующем:

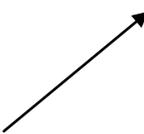
1. Описание таблиц БД и связей между ними, операций над данными в таблицах.
2. Контроль целостности и непротиворечивости данных в БД.
3. Автоматическое отображение информационной модели в физическую БД на носителях компьютера.

Microsoft Access является профессиональной программой управления **базами данных**. С ее помощью можно накапливать и систематизировать разнообразную информацию, искать и сортировать объекты согласно выбранным критериям, конструировать удобные формы для ввода данных и генерировать на основании имеющихся записей прекрасно оформленные отчеты. Программа Access обеспечивает одновременный доступ к данным десяткам пользователей. Кроме этой программы, существуют различные мощные пакеты для работы с базами данных - dBase, FoxPro, Oracle и др.

Примером табличного представления данных ( набор строк и столбцов) может быть телефонный справочник.

ФИОАбонента	НомерТелефона	Адрес
Иванов И.П.	2678522	Ул. Мира, 34
Петров А.С.	2688503	Ул. Королева, д.2 кв.45
Иванов И.П.	2759336	Ул. Мира, 56

При всей, на первый взгляд, простоте таблицы, обработка данных сталкивается с проблемами и трудностями при **поиске** информации и при **изменении** данных. Первая проблема. Быстрый поиск по таблице требует сортировки по ФИО в алфавитном порядке, а поиск по телефону - сортировки в порядке возрастания номеров. Т.е. приходится хранить две таблицы, что крайне неэкономно и затруднит изменения данных параллельно в двух таблицах. Вторая проблема, как различать записи с одинаковыми ФИО, например, Иванов И.П. 2678522 и Иванов И.П. 2759336. Обычно такие вопросы решаются введением дополнительного поля - первичного ключа или индекса(primary key). Первичный ключ должен иметь уникальные (неповторяющиеся) значения, однозначно определяющие записи в таблице. Введением в таблицу

	КодАбонента	ФИОАбонента	НомерТелефона	Адрес
 Первич- ный ключ	900	Иванов И.П.	2678522	Ул.Мира, 34
	901	Петров А.С.	2688503	Ул. Королева, д.2 кв.45
	902	Иванов И.П.	2759336	Ул.Мира,56

дополнительного поля с кодом абонента мы добьемся уникальности каждой записи.

Быстрый поиск можно обеспечить двумя способами – записи либо сортируют, либо индексируют по полю, по которому ведется поиск.

При сортировке происходит физическая перестановка записей в таблице.

При индексировании – к полю добавляется дополнительное невидимое числовое поле (вторичный ключ), упорядочивающее поле в порядке возрастания или убывания значений, при этом не происходит физическая перестановка записей.

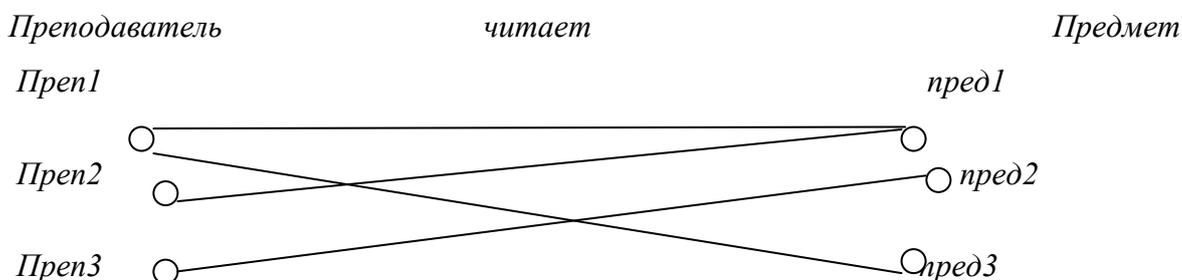
Затруднения изменения данных в таблицах связаны с дублированием, т.е. избыточностью информации. В нашем примере ул. Мира встречается 2 раза: в случае переименования потребуется внести изменения в 2 местах. При большом количестве записей любое изменение записей крайне трудоемко и связано с внесением дополнительных ошибок. Эта проблема решается разделением таблицы на части (нормализация). Например, данные об улицах можно вынести в отдельную таблицу

КодАбонента	ФИОАбонента	НомерТелефона	КодУлицы	ДомКв
900	Иванов И.П.	2678522	11	34
901	Петров А.С.	2688503	14	2 -45
902	Иванов И.П.	2759336	11	56

КодУлицы	ДомКв
11	Мира
14	Королева

В таблице появилось новое поле КодУлицы: оно есть внешний ключ, т.к. является первичным ключом другой таблицы. Здесь же видна связь между таблицами: один-ко-многим. Существуют также связи один-к-одному, многие-ко-многим, многие-к-одному.

Так в примере с преподавателями и предметами связь может быть представлена следующим образом



Связь **читает** обеспечивает между сущностями отношение многие-ко-многим, т.е. преподаватель может читать несколько предметов, а предмет может читаться несколькими преподавателями.

#### Типы данных

Формат	Тип данных	Размер
Text	Текст или числа, не требующие расчетов	<255 знаков
Поле Мемо	Очерь длинный текст	<65535
Number	Числа для расчетов	До 8 байт
Date/Time	Дата и время	8 байт
AutoNumber	Счетчик	4 байт
Currency	Денежные значения	8 байт
Boolean	Значения Да и Нет	1 бит
OLE-object	Объект, внедренный в таблицу	До 1Гб
Hyperlink	Адрес гипессылки	4 байта

#### Объекты базы данных.

**Таблицы.** В БД данные хранятся в виде двумерных таблиц . Можно импортировать и связывать таблицы из других СУБД. Одновременно можно открыть до 24 таблиц.

**Запросы.** При помощи запросов можно произвести выборку данных по какому-нибудь критерию из разных таблиц. В запрос можно включать до 255 полей.

**Формы.** Формы отображают данные из таблиц и запросов в более удобном для восприятия виде. С помощью форм можно добавлять и изменять данные, содержащиеся в таблице.

**Отчеты.** Отчеты предназначены для печати данных , содержащихся в таблицах и запросах в красиво оформленном виде.

**Страницы.** Специальный тип веб-страниц, предназначенный для работы с данными, хранящимися в БД через Интернет или интрасеть.

**Макросы.** Автоматическое выполнение последовательности команд, обозначающих часто выполняемые действия.

Окно базы данных позволяет получить доступ ко всем объектам базы данных и выбрать режим работы с объектом. Панель объектов находится в левой части окна. Она содержит ярлыки для каждого из объектов Access — Таблицы (Tables), Запросы (Queries), Формы (Forms), Отчеты (Reports), Страницы (Pages), Макросы (Macros) и Модули (Modules).

### ***Контрольные вопросы***

1. Дайте определение базе данных.
2. Опишите стадии проектирования баз данных.
3. Какие СУБД вы знаете. Каковы основные функции СУБД.
4. Перечислите типы связей между таблицами в базе данных.
5. Зачем нужен первичный ключ и как он может быть введен в базу данных.
6. Объекты БД : запрос, форма и отчет.

### ***Темы самостоятельных работ***

*Этап проектирования базы данных.*

1. Разработать информационную систему ДЕТСКАЯ\_ПОЛИКЛИНИКА.  
В базу данных по поликлинике ввести списки детей и списки сделанных им прививок, при этом надо иметь возможность просматривать как перечень прививок по каждому ребенку, так и списки детей по каждой прививке.
2. Разработать информационную систему НАЛОГОВАЯ ИНСПЕКЦИЯ. Налоговая инспекция создает базу данных юридических лиц по видам деятельности. Юридическое лицо может иметь несколько видов деятельности. Надо иметь возможность просматривать

как список юридических лиц по каждому виду, так и перечень видов деятельности по каждому юридическому лицу.

3. Разработать информационную систему КАФЕДРА. Кафедре ВУЗа требуется вести списки преподавателей и читаемых ими предметов. Каждый преподаватель может читать несколько предметов, а один предмет может читаться несколькими преподавателями. Надо иметь возможность просматривать как список преподавателей по каждому предмету, так и перечень предметов по каждому преподавателю. Требуется также распечатывать анкетные данные о преподавателях.

4. Разработать информационную систему ДЕКАНАТ. Следует вести перечни групп, списки студентов по группам, перечни предметов по группам. А также требуется хранить и распечатывать анкетные данные о студентах.

5. Разработать информационную систему АВТОБАЗА. Необходимо вести учет заказов с перечнем дополнительно устанавливаемых деталей (доукомплектовка), расчет суммы, печать заказа и суммы продаж за определенный период времени.

6. Разработать информационную систему ОКНА&ДВЕРИ. Фирма принимает заказы на изготовление пластиковых окон. Каждый заказ содержит данные о клиенте, размерах окна(высота и ширина). Погонный метр пластика стоит 4900 сум., квадратный метр стекла стоит 3000сум. Кроме пластика окно может комплектоваться ручками, петлями, подоконниками. Необходимо обеспечить ведение заказов от клиентов со сроком изготовления и расчетом стоимости заказа. Обеспечить просмотр списка заказов(размеры окна, клиент, стоимость) на любой вводимый срок изготовления .

### ***Список рекомендованной литературы***

1. Михеева В., Харитонов И. Microsoft Access 2003 в подлиннике. Наиболее полное руководство. БХВ-Петербург, 2003.
2. Кен Блюттман. Access трюки. Питер, 2006.
3. Лори Фуллер, Кен Кук, Джон Кауфельд. Access 2007 для чайников. М:Диалектика, 2007.

[На оглавление](#)

## **Лекция 9**

### **Создание таблиц в программе Access**

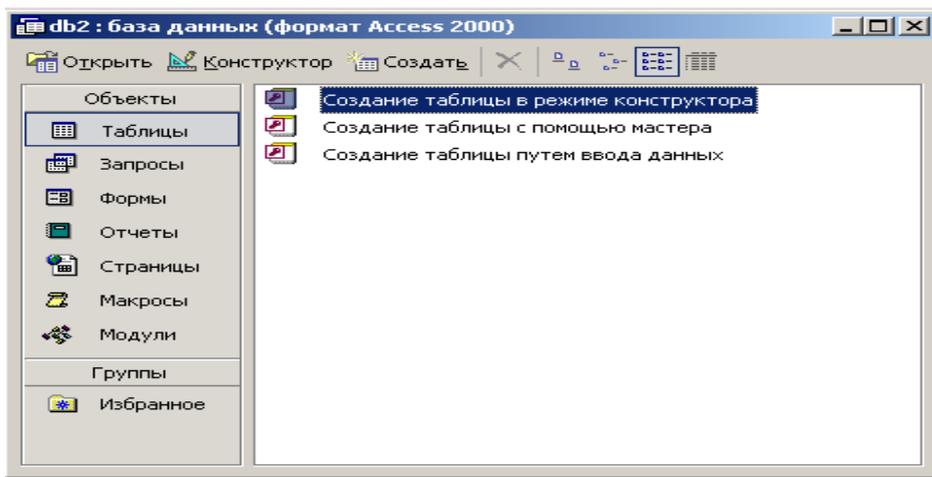
**Цель лекции:** ознакомить со способами создания объекта-таблица, описать связи и операции, выполняемые над данными в таблицах.

**Ключевые слова:** конструктор таблиц, мастер шаблонов, ввод данных, связи, операции сортировки, фильтрации и поиска данных.

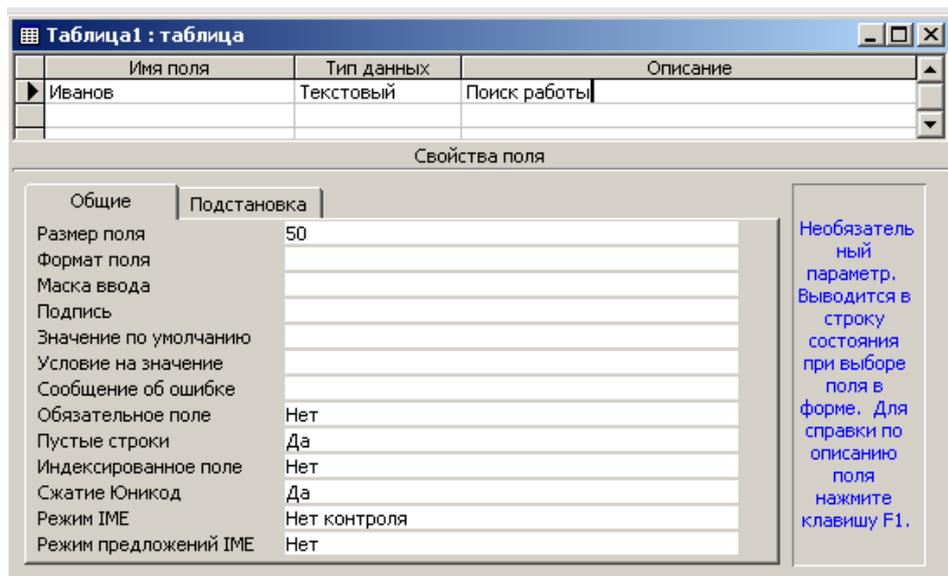
### **Содержание**

Способы создания таблиц  
Типы связей между таблицами  
Операции над данными

Таблицы – основные объекты любой БД. В таблицах хранятся все данные, а также структура базы (поля, их типы и свойства). Существует несколько способов создания таблиц.



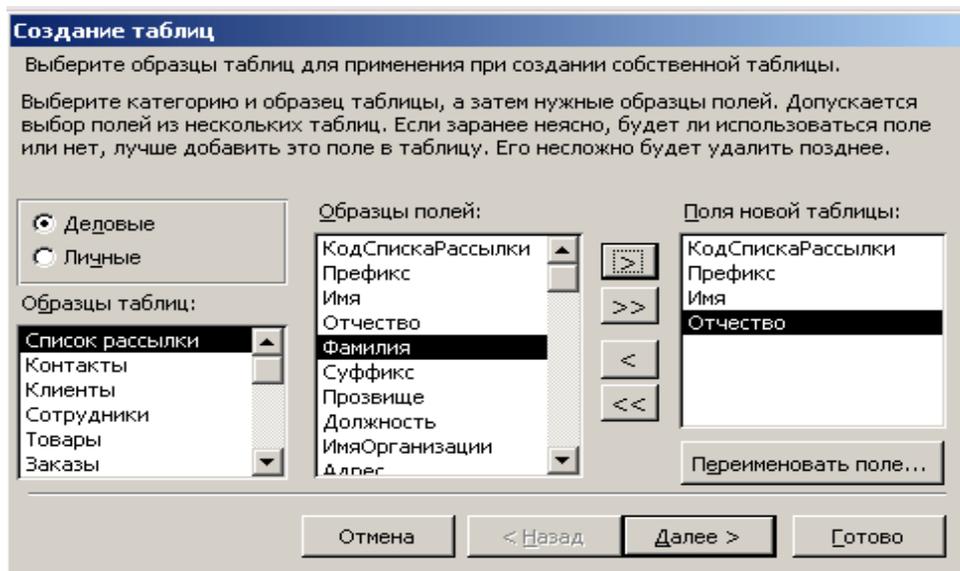
Первый способ- создание таблицы в режиме конструктора. Конструктор таблиц предназначен для задания и изменения структуры таблицы. С помощью конструктора можно формировать сколь угодно сложные таблицы с полями любого типа.



Конструктор таблиц

Второй способ-использование шаблонов. Мастер шаблонов позволяет быстро сформировать поля таблицы, содержащие наиболее типичные данные. При выборе этого способа в диалоговом окне следует выбрать образец таблицы, при этом можно просмотреть ее поля. Выбрав необходимые поля новой таблицы, мы получим готовую таблицу с отформатированными полями. С помощью мастера таблиц можно создавать некоторые часто встречающиеся структуры данных. Прежде чем заново строить структуру таблицы использование одного из стандартных шаблонов мастера таблиц при построении структуры таблицы позволяет сэкономить время.

Таблица данных похожа на лист электронной таблицы, но ее размеры ограничены. Число ее строк на единицу больше количества записей (последняя пустая строка предназначена для ввода новой записи), а число столбцов равно количеству полей в записи. Заголовки столбцов таблицы соответствуют именам полей, сформированных с помощью мастера создания таблиц.



Мастер создания таблиц

Третий способ - путем ввода данных. Таблица данных похожа на лист электронной таблицы, но ее размеры ограничены. Число ее строк на единицу больше количества записей (последняя пустая строка предназначена для ввода новой записи), а число столбцов равно количеству полей в записи.

Приглашение на работу : таблица					
	Код_Приглашение	Должность	Имя	Отчество	Фамилия
	1	Слесарь	Василий	Петрович	Иванов
▶	2	Доктор	Федор	Иванович	Трошкин
*	(Счетчик)				
Запись: 2 из 2					

Операции над таблицами:

- сортировка записей
- фильтрация записей
- поиск данных по таблице.

В БД данные таблицы хранятся в виде двумерных таблиц. Можно импортировать и связывать таблицы из других СУБД. Одновременно можно открыть до 24 таблиц.

Между таблицами можно устанавливая связи. При связывании применяются три типа отношений.

Отношение «один ко многим»

Отношение «один ко многим» связывает одну запись первой таблицы со многими записями второй таблицы. Оно устанавливается по умолчанию и применяется чаще других типов отношений. Например, один клиент сделал много заказов. Следовательно, одной записи в таблице соответствует много записей из другой таблицы.

Отношение «один ко одному» связывает одну запись в первой таблице с одной записью второй таблицы.

Отношение «многие ко многим» связывает много записей в первой таблице со многими записями второй таблицы.

### ***Контрольные вопросы***

1. Какой способ создания таблиц используется для ввода изменения в структуру таблицы.
2. В каких случаях может быть использован мастер шаблонов при создании таблицы.
3. Чем примечателен способ ввода данных при создании таблицы.
4. Какие типы отношений и в каких случаях могут быть использованы между таблицами.
5. Приведите примеры на использование операций над данными в таблицах.

### ***Темы самостоятельных работ***

*Этап создания таблиц и установки связей между ними*

1. Разработать информационную систему ДЕТСКАЯ\_ПОЛИКЛИНИКА.  
В базу данных по поликлинике ввести списки детей и списки сделанных им прививок, при этом надо иметь возможность просматривать как перечень прививок по каждому ребенку, так и списки детей по каждой прививке.
2. Разработать информационную систему НАЛОГОВАЯ ИНСПЕКЦИЯ. Налоговая инспекция создает базу данных юридических лиц по видам деятельности. Юридическое лицо может иметь несколько видов деятельности. Надо иметь возможность просматривать как список юридических лиц по каждому виду, так и перечень видов деятельности по каждому юридическому лицу.
3. Разработать информационную систему КАФЕДРА. Кафедре ВУЗа требуется вести списки преподавателей и читаемых ими предметов. Каждый преподаватель может читать несколько предметов, а один предмет может читаться несколькими преподавателями. Надо иметь возможность просматривать как список преподавателей по каждому предмету, так и перечень предметов по каждому преподавателю. Требуется также распечатывать анкетные данные о преподавателях.
4. Разработать информационную систему ДЕКАНАТ. Следует вести перечни групп, списки студентов по группам, перечни предметов по группам. А также требуется хранить и распечатывать анкетные данные о студентах.
5. Разработать информационную систему АВТОБАЗА. Необходимо вести учет заказов с перечнем дополнительно устанавливаемых деталей (доукомплектовка), расчет суммы, печать заказа и суммы продаж за определенный период времени.
6. Разработать информационную систему ОКНА&ДВЕРИ. Фирма принимает заказы на изготовление пластиковых окон. Каждый заказ содержит данные о клиенте, размерах окна(высота и ширина). Погонный метр пластика стоит 4900 сум., квадратный метр стекла стоит 3000сум. Кроме пластика окно может комплектоваться ручками, петлями, подоконниками.

Необходимо обеспечить ведение заказов от клиентов со сроком изготовления и расчетом стоимости заказа. Обеспечить просмотр списка заказов(размеры окна, клиент, стоимость) на любой вводимый срок изготовления .

### **Список рекомендованной литературы**

1. Михеева В., Харитоновна И. Microsoft Access 2003 в подлиннике. Наиболее полное руководство. БХВ-Петербург, 2003.
2. Кен Блюттман. Access трюки. Питер, 2006.
3. Лори Фуллер, Кен Кук, Джон Кауфельд. Access 2007 для чайников. М:Диалектика, 2007.

[На оглавление](#)

## **Лекция 10**

### **Организация запросов**

**Цель лекции:** ознакомить с объектом запрос, со способами создания запросов, с типами создаваемых запросов.

**Ключевые слова:** запрос, мастер запросов, конструктор, бланк запроса, условие отбора, итоговый запрос, запрос на обновление, удаление, добавление записи.

#### **Содержание**

Объект Запрос.

Способы создания запросов.

Типы запросов.

Таблицы данных предназначены для хранения информации. Вам, конечно, захочется не только заносить в них данные, но и обрабатывать записанную ранее информацию. Access предлагает множество способов извлечения данных из таблиц. Вы можете запросить данные, руководствуясь определенным правилом отбора, отфильтровать нужные записи или рассчитать результирующие значения с помощью формул.

**Запрос** — это набор условий, согласно которым производится выборка информации из таблиц. Запуск запроса формирует новую таблицу данных, единственным отличием которой от обычных таблиц является то, что с помощью повторных запусков запроса ее данные можно обновлять в соответствии с изменением информации источников данных запроса.

Рассмотрим создание запросов на примере следующей базы.

Учебная база

<b>Заказы</b>				
<b>Код заказа</b>	<b>Стоимость</b>	<b>ДатаПокупки</b>	<b>Код_продавца</b>	<b>Код_Покупателя</b>
3001	18,69	10.03.2009	1007	2008
3002	1900	10.04.2009	1004	2007

Заказы				
Код заказа	Стоимость	ДатаПокупки	Код_продавца	Код_Покупателя
3003	767,1	10.03.2009	1001	2001
3005	5160,45	10.03.2009	1002	2003
3006	1098	10.04.2009	1007	2008
3007	75,75	10.05.2009	1002	2004

Покупатели				
код_покупателя	имя	город	код_продавца	рейтинг
2001	Голубев	Липецк	1001	100
2002	Дмитриева	Архангельск	1003	200
2003	Кустов	Саратов	1002	200
2004	Кузьмин	Подольск	1002	300
2005	Кречко	Липецк	1001	100
2007	Карташов	Архангельск	1004	100
2008	Фролов	Саратов	1007	300

Продавцы			
код_продавца	имя	город	комиссионные
1001	Дугинов	Липецк	0,12
1002	Ольшевский	Саратов	0,13
1003	Борева	Новгород	0,11
1004	Мягкова	Липецк	0,15
1007	Шишкин	Бронницы	0,1

Запросы можно создавать двумя способами:

- с помощью мастера

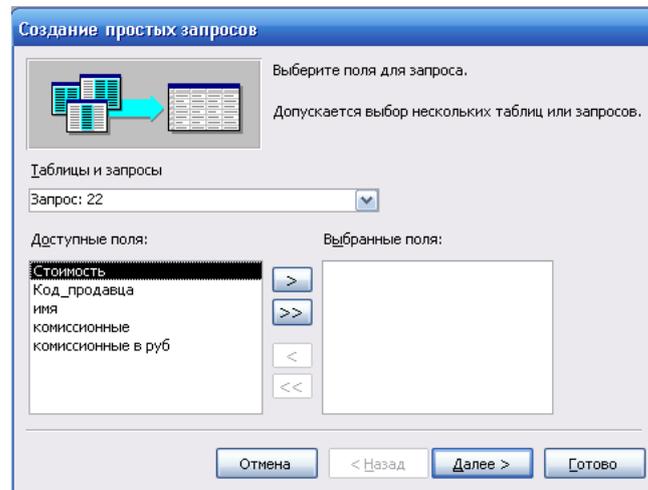
- в режиме конструктора

Мастер запросов

Чтобы упростить задачу пользователя, в состав Access включен мастер запросов, позволяющий автоматизировать процесс построения запроса. Давайте с помощью этого мастера выполним выборку информации из таблиц учебной базы данных .

1. В окне базы данных щелкните на кнопке Запросы
2. Дважды щелкните на значке Создание запроса с помощью мастера.
3. В раскрывающемся списке Таблицы и запросы первого окна мастера выберите таблицу Покупатели .
4. В списке Доступные поля щелкните на строке Город.
5. Щелкните на кнопке >, чтобы переместить выделенное поле в список Выбранные поля.

6. Повторяя шаги 3-5, добавьте в список Выбранные поля поле, Имя таблицы Покупатели, а также поле Стоимость таблицы Заказы.
7. Щелкните на кнопке Далее.
8. Введите имя запроса Перечень контактов и щелкните на кнопке Готово.



Мастер запросов

### Конструктор запросов

Мастер запросов умеет конструировать только простые условия отбора. Чтобы наложить дополнительные ограничения, следует пользоваться конструктором запросов, обеспечивающим полное управление параметрами запроса и построение сложных условий отбора данных.

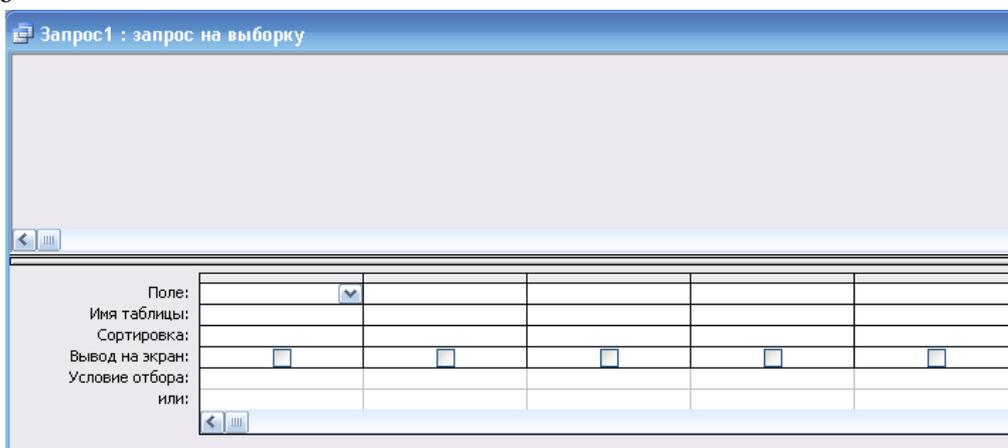
1. Чтобы переключиться в режим конструктора, выберите команду Вид > Конструктор. Окно конструктора показано на рисунке.

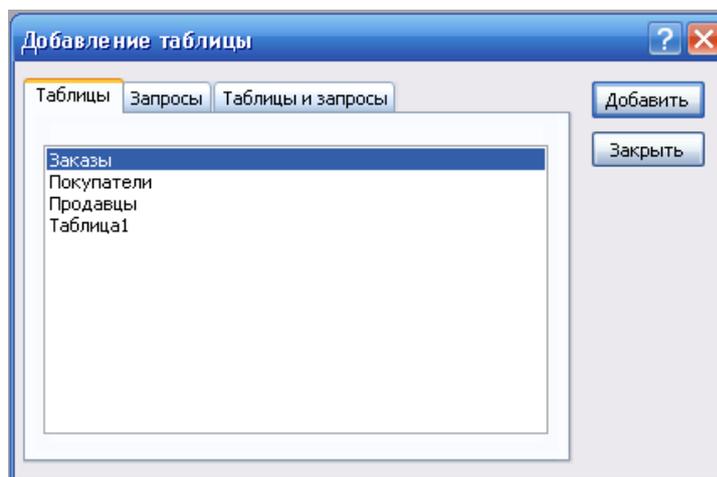
В его верхней части отображаются списки полей таблиц, к которым обращается запрос, и связи между таблицами. Нижняя область содержит бланк выбора полей таблиц, условий отбора и режимов сортировки. Чтобы добавить в запрос еще одно поле, выполните следующие шаги.

2. Переместите указатель на пункт Имя таблицы Контакты.

3. Нажмите кнопку мыши и перетащите поле Имя в верхнюю ячейку четвертой строки бланка запроса. Его имя появится в этой ячейке, а имя соответствующей таблицы — во второй ячейке того же столбца. Третья строка бланка запроса позволяет сортировать результат запроса по тому или иному полю.

4. В раскрывающемся списке третьей ячейки третьего столбца бланка выберите пункт По возрастаню





Конструктор запроса

5. Назначьте тот же режим сортировки для поля Имя. Теперь записи результата запроса будут упорядочены по алфавиту фамилий и имен. Имеющийся вариант связи добавляет в результат запроса только те записи связанных таблиц, в которых значения полей Код\_Контакты равны. Так как для некоторых людей из таблицы Контакты нет записей в таблице Список, то информация о таких людях не включается в результат запроса. Чтобы запрос возвращал данные даже о тех людях (включенных в таблицу Контакты), с которыми не было никаких контактов, нашедших отражение в таблице Список, следует изменить параметры объединения.
6. Дважды щелкните на линии связи.
7. В открывшемся окне диалога Параметры объединения выберите положение переключателя Объединение всех записей из "Контакты" и только тех записей из "Список", в которых связанные поля совпадают
8. Щелкните на кнопке ОК. На одном конце линии связи появится стрелка, указывающая на смену режима объединения.
9. Щелчком на кнопке Вид панели инструментов запустите запрос повторно. Результат выполнения запроса с учетом сортировки и нового режима объединения будет таким, как показано на рис. 17.5.
10. Закройте окно запроса.
11. В ответ на вопрос о необходимости сохранения новой структуры запроса ответьте Да.

## Типы запросов

### Запрос с условием отбора

Добавление в запрос условия отбора позволяет выбирать из таблицы не все записи, а лишь те, которые удовлетворяют определенным критериям. Например, вас могут заинтересовать все покупатели из городов Липецк и Саратов

1. Выделите в окне базы данных значок запроса Покупатели из городов Липецк&Саратов.
2. Щелкните на кнопке Конструктор.

3. В бланке запроса щелкните на ячейке Условие отбора первого столбца правой кнопкой мыши и выберите в контекстном меню команду Построить. Откроется окно построителя выражений.

В левом списке построителя щелкните на папке Операторы.

2. В среднем списке выберите категорию Сравнения.
3. В правом списке дважды щелкните на пункте Between, чтобы добавить этот оператор в поле формулы.
4. Щелчком выделите в поле формулы первый местозаполнитель «Выражение» («Expr»).
5. В левом списке построителя выражений двойным щелчком откройте папку Функции.
6. 9. Щелкните на папке Встроенные функции, содержащей стандартные функции Access.
7. В среднем списке построителя выражений щелкните на пункте Дата/время (Data/Time).
8. В правом списке дважды щелкните на функции DateValue, чтобы заменить ею местозаполнитель «Выражение».
9. Нажмите два раза клавишу →, выделив местозаполнитель «stringexpr».
10. Введите текст "1.12.2010".
11. Повторяя шаги 7-13, замените второй местозаполнитель «Выражение» на выражение DateValue ("31.12.2010").

У вас должна получиться формула Between DateValue ("1.12.2010") And DateValue ("31.12.2010"). Она проверяет условие нахождения даты в интервале от 1 до 31 декабря 2010 г., то есть отбирает те записи, значение поля Дата которых относится к декабрю 2010 г.

15. Щелкните на кнопке ОК. Построенная формула появится в ячейке Условие отбора первого столбца бланка запроса. При выборе записей, относящихся к одному месяцу, точная дата контакта может оказаться несущественной. Совсем выбросить это поле из бланка запроса нельзя, так как оно необходимо для реализации условия отбора. Однако любое поле можно скрыть, то есть не включать в результат запроса.
16. Сбросьте флажок Вывод на экран первого столбца запроса.
17. Щелчком на кнопке Запуск панели инструментов выполните запрос.

Запрос итоговый.

Запросы позволяют не только выбирать записи из таблиц, но и вычислять различные статистические параметры. Например, можно подсчитать суммарное количество контактов и вывести даты первого и последнего контакта с каждым из людей, включенных в таблицу Контакты. Чтобы построить такой запрос в режиме конструктора, выполните следующие действия.

1. В окне базы данных щелкните на кнопке Запросы.
2. Дважды щелкните на значке Создание запроса в режиме конструктора.
3. В открывшемся окне диалога выделите строку Контакты.

Щелчком на кнопке Добавить добавьте выбранную таблицу в верхнюю область конструктора запроса.

4. Выделите пункт Список и снова щелкните на кнопке Добавить.
5. Щелчком на кнопке Закройте закройте окно диалога Добавление таблицы. Списки полей двух таблиц, соединенные линией связи, появятся в окне конструктора.

6. Щелкните на кнопке Групповые операции панели инструментов. В бланке запроса появится дополнительная строка Групповая операция позволяет выполнять статистические операции со значениями конкретных полей.
7. Перетащите поле Фамилия в ячейку Поле первого столбца конструктора.

В ту же ячейку второго столбца перетащите поле Имя таблицы Контакты.

8. В третий, четвертый и пятый столбцы бланка запроса перетащите поле Дата таблицы Список.
9. В раскрывающемся списке ячейки Групповая операция (Total) третьего столбца бланка запроса выберите пункт Min.
10. В той же ячейке четвертого столбца выберите пункт Max.
11. В пятом столбце задайте групповую операцию Count. Групповые операции построенного запроса обработают все записи таблицы Список, соответствующие конкретному человеку из таблицы Контакты, и вместо самих данных таблицы Список выведут в соответствующее поле результата запроса только значение величины, вычисляемой по определенной формуле.

Доступные групповые операции перечислены в таблице.

**Таблица. Групповые операции**

<b>Название</b>	<b>Функция</b>
Условие(Where)	Режим задания условия отбора для поля, по которому не выполняется группировка. Access автоматически делает такое поле скрытым
Выражение (Expression)	Вычисляемое поле, значение которого рассчитывается по сложной формуле
Group By	Поле, определяющее группу записей, по которой вычисляются статистические параметры. К одной группе относятся все записи, для которых значения поля с режимом Group By (Группировка) одинаковы
Last	Последнее значение в группе
First	Первое значение в группе
Var	Вариация значений поля
StDev	Стандартное отклонение величин поля от среднего
Count	Количество записей, соответствующее поле которых не содержит величины Null
Max	Максимальное значение
Min	Минимальное значение
Avg	Среднее значение поля
Sum	Сумма значений поля по всем записям

Last	Последнее значение в группе
First	Первое значение в группе
Var	Вариация значений поля
StDev	Стандартное отклонение величин поля от среднего

Щелкните на кнопке Вид, чтобы выполнить запрос. Появится таблица с пятью столбцами. Два первых столбца содержат фамилии и имена людей. По ним выполняется группировка, то есть расчет значений остальных полей запроса выполняется для записей таблицы Список, сопоставляемых с одним человеком. Как уже говорилось выше, соответствие контакта таблицы Список и человека из таблицы Контакты определяется полями Код\_Контакты, с помощью которых осуществляется связь этих двух таблиц. Третий и четвертый столбцы запроса выводят соответственно дату первого (функция Min) и последнего (функция Max) контакта с данным человеком. Пятый столбец содержит количество записей в таблице Список (функция Count), соответствующих данному человеку, то есть число контактов с ним. Единственный недостаток построенного запроса — это непонятные имена столбцов. Давайте скорректируем их.

14. Щелчком на кнопке Вид вернитесь в конструктор запроса.
15. В ячейке Поле третьего столбца замените имя Дата на текст Дата первого контакта: Дата. Правая часть этого выражения, расположенная правее двоеточия, по-прежнему задает имя поля, а левая определяет название столбца результата запроса. Таким образом, любому столбцу запроса можно назначить произвольное имя.
16. В ячейке Поле четвертого столбца введите Дата последнего контакта: Дата.
17. В первой строке пятого столбца бланка запроса введите Число контактов: Дата.
19. Снова щелкните на кнопке Вид.
20. Закройте запрос.
21. Для сохранения изменений структуры щелкните на кнопке Да.
22. В окне диалога Сохранение (Save As) введите имя Итоговый запрос и щелкните на кнопке ОК.

## Запрос действия

До сих пор вы сталкивались с запросами, выполняющими выборку данных и некоторые вычисления. Однако запросы могут применяться также для добавления, удаления и обновления группы записей таблицы. Такие запросы являются мощным инструментом преобразования данных, они называются *запросами действия*. Предположим, что по каким-то причинам вам понадобилось скорректировать даты контактов, заменив во всех записях таблицы Список, относящихся к 1999 году, месяц ноябрь на декабрь. Подобную операцию трудно проделать вручную, - если в таблице содержится несколько тысяч записей. Запрос действия позволяет быстро решить поставленную задачу.

1. В окне базы данных щелкните на кнопке Таблицы.
2. Выделите таблицу Список, данные которой нужно обновить.
3. В палитре кнопки Новый объект выберите пункт Запрос. Откроется окно диалога Новый запрос, показанное на рис. 17.8. Подобное окно открывается и при щелчке на кнопке Создать (New) окна базы данных. Оно позволяет выбрать наиболее удобный способ создания объекта.

4. Дважды щелкните на строке Конструктор (Design View). Таблица Список, выделенная в окне базы данных на шаге 2, автоматически появится в окне конструктора запросов.
5. Перетащите в бланк запроса поле Дата, значение которого нужно обновлять.
6. В палитре кнопки Тип запроса (Query Type) выберите пункт Обновление (Update Query). Описание всех возможных вариантов запросов, предлагаемых программой Access, приведено в табл. 17.2. Структура бланка запроса видоизменяется в соответствии с типом запроса. В бланке выбранного варианта запроса на обновление появляется поле Обновление (Update To), в которое нужно ввести новое значение поля. Чтобы изменить месяц даты с ноября на декабрь, достаточно прибавить к дате 30 дней.
7. Введите в ячейку Обновление (Update To) формулу [Дата] +30.

**ТАБЛИЦА.** Варианты запросов

<b>Тип</b>	<b>Описание</b>
Выборка (Select) ;	Выборка данных в таблицу результата запроса на основе указанных условий отбора
Перекрестный (Crosstab)	Результат запроса выводит статистические значения (сумму, количество или среднее) для одного из полей таблицы в зависимости от двух параметров других полей таблицы, задающих заголовки строк и столбцов результата перекрестного запроса
Создание таблицы (Make-Table)	Создание новой таблицы в текущей или в другой базе данных на базе информации из имеющихся таблиц
Обновление (Update)	Обновление данных таблицы
Добавление (Append)	Добавление набора записей в таблицу
Удаление (Delete)	Удаление записей таблицы в соответствии с указанным критерием

8. Чтобы обновлялись только даты, относящиеся к ноябрю 2000 года, введите в ячейку Условие отбора формулу Between DateValue ("1.11.2009") And DateValue ("30.11.2009"), которая подробно обсуждалась в третьем упражнении этого занятия.
9. Закройте запрос, сохранив его под именем Обновление.
10. Щелкните на кнопке Таблицы окна базы данных и двойным щелчком на значке Список откройте эту таблицу. В ней есть четыре записи, относящиеся к ноябрю 2000 года.

Запросы действия могут выполнять широкомасштабные изменения данных, которые отменить уже невозможно. Будьте внимательны. Перед запуском таких запросов полезно сделать резервную копию базы данных, чтобы иметь возможность возвратиться к исходному состоянию таблиц. Значки запросов действия снабжены восклицательным знаком, предупреждающим об их особой роли.

11. В окне базы данных щелкните на кнопке Запросы.
12. Дважды щелкните на значке Обновление.

Ответьте Да на вопрос о необходимости запуска запроса действия. Access проинформирует вас об обнаружении четырех записей, удовлетворяющих условию отбора, и попросит подтвердить необходимость их изменения.

13. Щелкните на кнопке Да и изучите изменения, произошедшие с данными таблицы Список.

### ***Контрольные вопросы***

1. Мастер создания запросов. Случаи его применения.
2. Режим конструктора. Его преимущества.
3. Операции, применяемые при создании запросов с условием.
4. Как можно создать итоговые запросы.
5. Приведите примеры на организацию запросов действия.

### ***Темы самостоятельных работ***

#### *Этап создания различного типа запросов*

1. Разработать информационную систему ДЕТСКАЯ\_ПОЛИКЛИНИКА.  
В базу данных по поликлинике ввести списки детей и списки сделанных им прививок, при этом надо иметь возможность просматривать как перечень прививок по каждому ребенку, так и списки детей по каждой прививке.
2. Разработать информационную систему НАЛОГОВАЯ ИНСПЕКЦИЯ. Налоговая инспекция создает базу данных юридических лиц по видам деятельности. Юридическое лицо может иметь несколько видов деятельности. Надо иметь возможность просматривать как список юридических лиц по каждому виду, так и перечень видов деятельности по каждому юридическому лицу.
3. Разработать информационную систему КАФЕДРА. Кафедре ВУЗа требуется вести списки преподавателей и читаемых ими предметов. Каждый преподаватель может читать несколько предметов, а один предмет может читаться несколькими преподавателями. Надо иметь возможность просматривать как список преподавателей по каждому предмету, так и перечень предметов по каждому преподавателю. Требуется также распечатывать анкетные данные о преподавателях.
4. Разработать информационную систему ДЕКАНАТ. Следует вести перечни групп, списки студентов по группам, перечни предметов по группам. А также требуется хранить и распечатывать анкетные данные о студентах.
5. Разработать информационную систему АВТОБАЗА. Необходимо вести учет заказов с перечнем дополнительно устанавливаемых деталей (доукомплектовка), расчет суммы, печать заказа и суммы продаж за определенный период времени.

6. Разработать информационную систему ОКНА&ДВЕРИ. Фирма принимает заказы на изготовление пластиковых окон. Каждый заказ содержит данные о клиенте, размерах окна(высота и ширина). Погонный метр пластика стоит 4900 сум., квадратный метр стекла стоит 3000сум. Кроме пластика окно может комплектоваться ручками, петлями , подоконниками. Необходимо обеспечить ведение заказов от клиентов со сроком изготовления и расчетом стоимости заказа. Обеспечить просмотр списка заказов(размеры окна, клиент, стоимость) на любой вводимый срок изготовления .

### ***Список рекомендованной литературы***

1. Михеева В., Харитонов И. Microsoft Access 2003 в подлиннике. Наиболее полное руководство. БХВ-Петербург, 2003.
2. Кен Блюттман. Access трюки. Питер, 2006.
3. Лори Фуллер, Кен Кук, Джон Кауфельд. Access 2007 для чайников. М:Диалектика, 2007.

[На оглавление](#)

## **Лекция 11**

### **Формы. Панель управления. Создание отчетов.**

**Цель лекции:** ознакомить с назначением объекта - форма, со способами создания, с панелью элементов, со способами подготовки различного типов отчетов по базе.

**Ключевые слова:** объект-форма, конструктор форм, мастер форм, элементы управления формами, отчет, заголовок отчета, верхний и нижний колонтитул отчета, область данных, примечание отчета, предварительный просмотр отчета.

#### **Содержание**

Способы создания форм.  
Элементы управления панели.  
Организация вычислений в формах.  
Способы создания отчетов.  
Оформление отчетов.

Информация баз данных хранится в таблицах, но с ними не очень удобно работать. Для ввода, редактирования и печати информации предназначаются формы и отчеты.

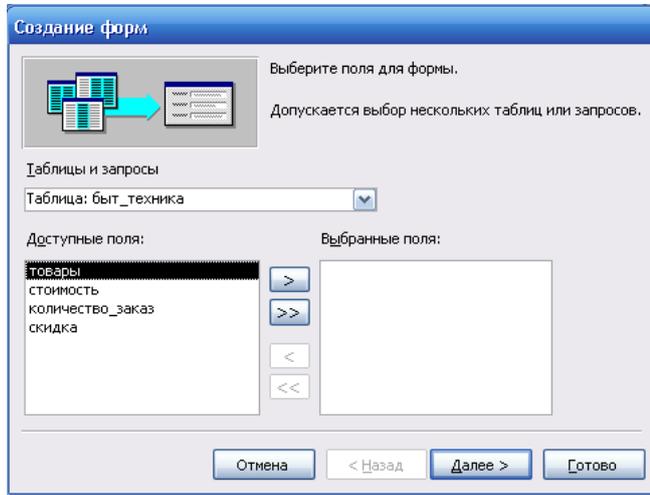
В то время как таблицы и запросы позволяют отобразить на экране длинные списки записей, формы дают возможность сосредоточиться на конкретной записи. Они облегчают ввод, редактирование и восприятие информации, могут содержать вспомогательные подписи и элементы оформления.

Создавать формы можно:

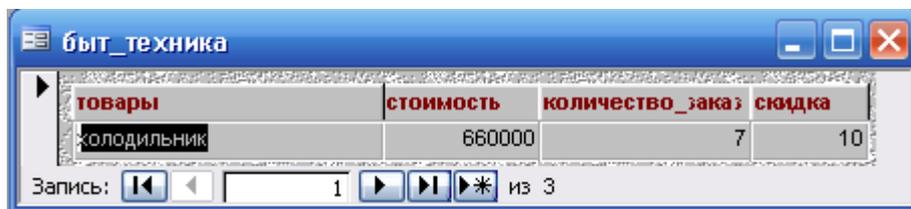
- с помощью мастера форм
- с помощью конструктора форм
- С помощью инструментов автоформы

**Мастер форм** позволяет сэкономить время и быстро сконструировать привлекательную форму для записей любой таблицы.

1. С помощью команды **Файл > Открыть** откройте базу данных.
2. В окне базы данных щелкните на кнопке **Формы**.
3. В списке форм дважды щелкните на значке **Создание формы с помощью мастера**.



4. Щелкните на кнопке **»**, чтобы добавить в список **Выбранные поля** все поля таблицы.
5. В следующем окне диалога выберите для формы **стиль Выровненный** и щелкните на кнопке **Далее**.
6. В списке третьего окна выберите вариант **оформления Камень** или любой другой понравившийся вам стиль и снова щелкните на кнопке **Далее**.



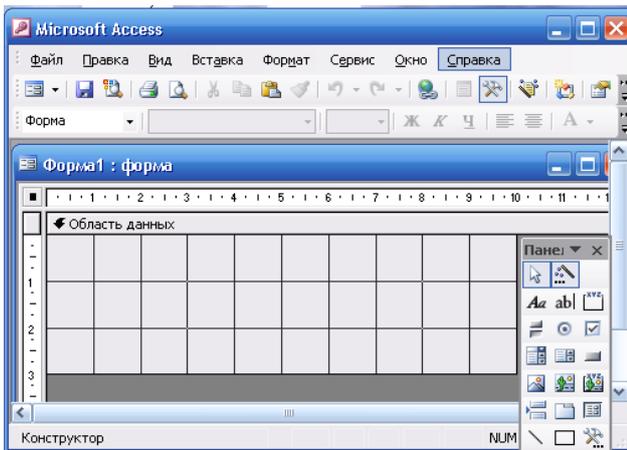
7. В последнем окне мастера щелкните на кнопке **Готово**, не изменяя никаких параметров. Мастер сгенерирует форму и откроет ее в режиме просмотра данных.

Окно формы содержит названия и области отображения для всех полей таблицы, выбранных в первом окне мастера. В нижней части формы расположены кнопки перемещения по записям, аналогичные кнопкам окна таблицы.

**Конструктор форм.** Недостатком форм, создаваемых мастером, является то, что они однообразны и не содержат пояснительных надписей. Чтобы приукрасить форму и расположить поля более удобным образом, следует воспользоваться конструктором форм, который позволяет передвигать и масштабировать элементы формы, связывать их с источником данных и настраивать любые другие параметры формы. Давайте создадим с помощью конструктора форму для таблицы **Быт\_техника**.

1. В окне базы данных дважды щелкните на значке **Создать форму в режиме конструктора**. В окне конструктора присутствуют пустая форма с разметочной сеткой, вертикальная и горизонтальная линейки, помогающие позиционировать объекты. Неотъемлемым атрибутом конструктора форм является панель инструментов **Панель элементов**, помогающая добавлять

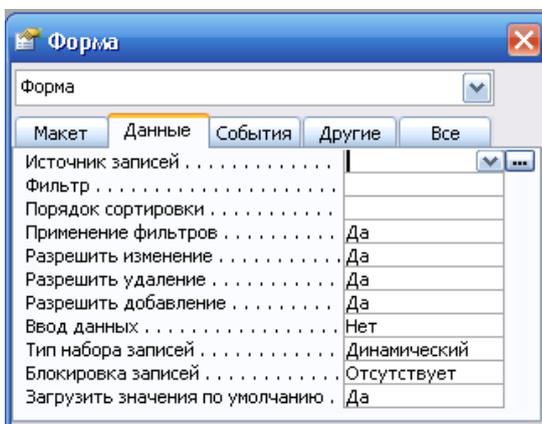
элементы формы. Новую форму необходимо связать с источником данных — таблицей Быт\_техника.



- Щелкните в пустой области конструктора и выберите в контекстном меню команду Свойства.
- На вкладке Данные окна диалога свойств формы щелкните на стрелке поля Источник записей.

Любой элемент формы имеет контекстное меню, в котором есть команда Свойства. Выбор такой команды открывает окно свойств, подобное окну свойств формы. Вкладка Макет содержит параметры форматирования объекта. Вкладка Данные — описание источника данных, различные ограничения и условия, накладываемые на данные поля. Вкладка События позволяет писать функции отклика на различные события (например, перемещение мыши на объект). Параметры, не вошедшие в первые три группы, размещаются на вкладке Другие. Вкладка Все обеспечивает одновременный доступ ко всем параметрам.

- Выберите в раскрывшемся списке пункт Быт\_техника.
- Закройте окно свойств. В конструкторе форм появится небольшое окно со списком полей источника записей. Это окно помогает добавлять элементы управления формы. Его можно скрывать или выводить на экран с помощью кнопки Список полей. Сейчас нужно добавить в форму три области ввода для трех полей таблицы.
- Поместите указатель на пункт Товары окна Быт\_техника, нажмите кнопку мыши и перетащите этот пункт в область формы. Access создаст элемент управления типа поле, а также подпись.



Окно свойств формы

Любой объект формы можно выделить щелчком мыши, при этом вокруг него появляется габаритная рамка с боковыми и угловыми маркерами масштабирования объекта. В левом верхнем углу объекта появляется большой квадрат, за который можно перетаскивать объект в пределах формы.

**Инструменты автоформы** экономят рабочее время, необходимое для создания форм. Они выполняют черновую работу, пользователю остается нанести несколько завершающих штрихов в режиме конструктора

## **Элементы управления**

При открытии конструктора форм на экране появляется панель элементов. Если ее нет, щелкните на кнопке Панель элементов панели инструментов. С помощью кнопок панели элементов в форму можно добавлять различные объекты, типы которых перечислены выше. Элементы управления форм и отчетов сходны между собой, поэтому такая же панель элементов имеется и в конструкторе отчетов, о котором мы поговорим на этом занятии позже, а все операции, рассматриваемые в этом и в следующем упражнении, одинаково применимы как к формам, так и к отчетам.

Элементы управления форм.

**Надпись (Label)** ---Надписи создаются автоматически вместе с элементами типа текстовое поле, поле со списком и т. п. Они разъясняют смысл поля ввода. Дополнительные надписи могут использоваться для заголовков и пояснений

**Поле (Text Box)**-- Используется для ввода и отображения информации полей таблиц и запросов, а также для вывода результатов вычислений

**Группа переключателей (Option Group)**-- Группа переключателей ссылается на некоторое поле таблицы. Каждый переключатель группы соответствует определенному целочисленному значению этого поля

**Поле со списком**- Разрешает как непосредственный ввод значения в поле, так и его выбор в раскрывающемся списке predetermined значений

**Выключатель (Toggle Button)**-- Позволяет вводить информацию типа да/нет. Величине «да» или «истина» соответствует утопленное положение выключателя

**Переключатель (Option Button)**-- Элемент группы переключателей, в которой может быть отмечен только один из них (его кружок выглядит зачерненным)

**Кнопка (Command Button)** --Щелчок на такой кнопке выполняет команду, с помощью которой можно перейти в другую форму, переместиться по записям и выполнить многие другие операции

**Рисунок**--Рисунок любого графического формата с рамкой

**Свободная рамка объекта (Unbound Object Frame)**-- Объект OLE, не связанный ни с каким источником данных

Присоединенная рамка объекта --Объект OLE, связанный с полем данных таблицы или запроса

Разрыв страницы --Линия, по которой формируется перевод страницы при выводе формы или отчета на принтер

Набор вкладок --Позволяет разместить на одном и том же пространстве экрана несколько наборов элементов управления. Удобен в тех случаях, когда элементы легко разделяются на логические группы .

Подчиненная форма/отчет-- Вставляет в форму (или отчет) область с другой формой (или отчетом), связанной с главной

Линия-- Добавляет прямую линию

Прямоугольник --Рисует прозрачный или непрозрачный прямоугольник с цветной границей любой толщины

Другие элементы-- Открывает меню с элементами управления, не вошедшими в панель элементов

Теперь в окне формы может разместиться сразу две записи. Такую форму можно листать с помощью полосы прокрутки. Заголовок и примечание формы отображаются в верхней и нижней части окна, не смещаясь при прокручивании записей

В целом отчеты похожи на формы, но они, как правило, предназначены для вывода информации из базы данных на принтер. Поэтому в отчетах данные форматируют так, чтобы их было удобно размещать на отдельных страницах. Отчеты поддерживают самые разнообразные способы оформления и позволяют группировать данные, разбивая их на логически цельные блоки.

Способы создания отчетов:

- с помощью мастера отчетов
- с помощью конструктора

Мастер отчетов. Чтобы облегчить работу пользователя, в Access имеется специальный мастер, который при недостатке времени позволяет быстро создавать довольно привлекательные отчеты. Давайте поручим мастеру построение отчета о контактах с разными людьми.

1. В окне базы данных щелкните на кнопке Отчеты.
2. Дважды щелкните на значке Создание отчета с помощью мастера.

В отчет нужно добавить как информацию о товарах, так и данные о продаже их. Для этого следует сгруппировать вместе все товары по типам и распечатать их список ниже области с данными об этом товаре.. Такой порядок печати записей позволит быстро найти полный список товаров с любыми данными по их продаже.

3. В раскрывающемся списке Таблицы и запросы первого окна мастера отчета выберите таблицу Контакты.
4. В списке Доступные поля выделите поле Название\_товара.
5. Щелчком на кнопке > переместите выбранное поле в список Выбранные поля.

6. Повторяя шаги 4 и 5, добавьте в список **Выбранные поля** поля **Название\_товара**, **цена**, **скидка**, **продано(в шт)**, **вид\_товара**.

Поле **Код\_товаров** содержит служебную информацию, обеспечивая связь записей двух таблиц. Не помещайте такие поля в отчет никогда, кроме случаев отладки взаимосвязей таблиц базы данных.

7. Выберите в списке **Таблицы и запросы** таблицу **Доп\_по\_быт**.
8. Переместите в список **Выбранные поля** поля **тип\_товара** и **количество\_на\_складе**, добавив их к полям таблицы **Быт\_техника**.

9. Щелкните на кнопке **Далее**.

Во втором окне мастера можно выбрать способ расположения данных — упорядоченные по записям таблицы **Быт\_техника**, а затем уже по записям таблицы **Доп\_по\_быт**, или наоборот. Схема группировки отображается в правой части окна диалога.

10. Оставьте вариант **Контакты** и щелкните на кнопке **Далее**.
11. В следующем окне не назначайте дополнительной группировки. Просто щелкните на кнопке **Далее**.
12. Четвертое окно мастера отчета позволяет задать порядок сортировки записей. Выберите в списке 1 пункт **Дата** и щелкните на кнопке **Далее**.

Следующее окно позволяет выбрать способ расположения полей и ориентацию страницы.

13. Выберите положение переключателя **Структура 1**.
14. Затем щелкните на кнопке **Далее**.
15. В следующем окне мастера выберите вариант оформления **Строгий** и щелкните на кнопке **Далее**.
16. Введите в качестве заголовка текст **Список контактов** и щелкните на кнопке **Готово**.

Мастер сформирует отчет и откроет его в отдельном окне в режиме предварительного просмотра. В нем видно, что сначала идет фамилия первого человека, информация о нем и список контактов с ним, затем располагаются те же данные, касающиеся второго человека, и т. д. В целом отчет выглядит неплохо и может устроить нетребовательного пользователя.

### **Режим конструктора.**

Чтобы переключиться в **режим конструктора**, выберите команду **Вид > Конструктор**

В схеме отчета присутствует несколько разделов. Их назначение указано в таблице **Разделы отчета**. Поля и подписи можно размещать в любом разделе отчета, регулируя этим их положение на странице.

2. Щелчком выделите надпись **Фамилия**.
3. Нажатием клавиши **Delete** удалите надпись.

Разделы отчета

- Заголовок отчета
- Верхний колонтитул
- Область данных

-Нижний колонтитул

-Примечание

4. Повторяя шаги 2 и 3, удалите из раздела Заголовок группы Код\_Контакты надписи Имя, Дата и Описание. Назначение соответствующих полей очевидно и не нуждается в пояснении.
5. Щелкните на горизонтальной линии, расположенной ниже поля Назвакие Компании, правой кнопкой мыши и выберите команду Вырезать.
6. Аналогичным приемом удалите три оставшиеся линии раздела Заголовок группы Код\_Контакты. Нет смысла выводить подписи полей адреса, названия компании и телефона в заголовке каждой группы. Достаточно будет поместить соответствующие надписи в верхний колонтитул страницы.
7. Поместите указатель мыши между заголовками разделов Верхний колонтитул и Заголовок группы Код\_Контакты, чтобы его значок превратился в двунаправленную стрелку. Нажмите кнопку мыши и перетащите заголовок группы вниз, освобождая место для верхнего колонтитула.
8. Щелчком выделите надпись Адрес. Нажмите клавиши Ctrl+X, чтобы вырезать ее.
9. Щелкните на заголовке раздела Верхний колонтитул и нажмите клавиши Ctrl+V, чтобы вставить вырезанную надпись в верхний колонтитул.
10. Выделяя надписи колонтитула и перетаскивая их мышью за маркер верхнего левого угла либо передвигая нажатием клавиши Ctrl и клавиш со стрелками.
11. Щелкните на надписи Адрес. Повторным щелчком на ней включите режим редактирования и замените текст на Почтовый адрес.
12. Описанным выше приемом измените текст подписей НазваниеКомпании и РабочийТелефон соответственно на Компания и Телефон.
13. Переставьте поля раздела Заголовок группы Код\_Контакты, разместив их в две строки. Фамилия и имя — в первой строке, а остальные три поля — во второй.
11. Щелкните правой кнопкой мыши на жирной линии, расположенной в разделе Заголовок отчета. Выберите в контекстном меню команду Копировать (Сору).
12. Щелкните на заголовке раздела Заголовок группы Код\_Контакты и нажмите клавиши Ctrl+V, чтобы вставить в этот раздел дубликат линии.
13. Нажимая клавиши Ctrl+V, переместите линию вниз, под вторую строку полей данных.
14. Снова нажмите Ctrl+V, добавив вторую линию вдоль верхнего края раздела.
15. Перетащите заголовок раздела Область данных вверх, убрав излишнее пустое пространство.
16. Щелчком выделите поле Дата. Увеличьте его горизонтальный размер, перетащив вправо серединный маркер правой границы поля.
20. Растяните поле Описание вправо до границы отчета.
21. С помощью команды Вид > Предварительный просмотр (View > Print Preview) переключите отчет в режим вывода данных. В новом варианте, показанном на рис. 18.18, информация размещается намного компактнее.
22. Закройте отчет. При появлении запроса о необходимости сохранения новой структуры отчета щелкните на кнопке Да.

В нижней части страницы отчета выводятся текущая дата и номер страницы. Эта информация генерируется элементами управления, размещенными в разделе Нижний колонтитул конструктора отчета.

Группировка и сортировка

Отчет, который вы сконструировали в двух предыдущих упражнениях, имеет еще один существенный недостаток. Мастер отчетов автоматически сгруппировал данные таблицы Доп\_по\_быт по их принадлежности к тем или иным записям таблицы Быт\_техника, которая определяется связью между полями Код\_товаров этих двух таблиц. По указанной причине название\_товаров в отчете упорядочены по возрастанию числового параметра Код\_Контакты — идентификатора записей таблицы Контакты. Это неудобно, в отчетах с большим числом записей при такой сортировке будет довольно трудно отыскать нужного человека. Есть смысл изменить порядок сортировки, для этого выполните следующие шаги.

1. Щелкните на кнопке Отчеты окна базы данных.
2. Выделите значок отчета Список контактов.
3. Щелчком на кнопке Конструктор панели инструментов окна базы данных откройте отчет в режиме конструктора.
4. Щелкните на кнопке Сортировка и группировка панели инструментов, чтобы открыть окно диалога,

В этом окне назначены два уровня сортировки. Сначала в порядке возрастания по полю Код\_товаров, а затем в пределах списка контактов, относящихся к одному человеку, по возрастанию даты контакта. На кнопке строки Код\_товаров виден значок, указывающий на то, что записи группируются по данному полю, то есть при переборе записей таблицы Быт\_техника для каждого нового значения поля Код\_товаров генерируется заголовок группы, в котором, как вы должны помнить из предыдущего упражнения, выводятся название\_товаров, цена другие характеристики.

Казалось бы, можно просто изменить поле сортировки Код\_товаров на название\_товаров, и записи в отчете сразу же выстроятся по алфавиту. Вспомните, что новый заголовок группы выводится только для каждого *нового значения* поля, по которому выполняется группировка. Код\_товаров — уникальный идентификатор записей таблицы Контакты, группировка по которому обеспечивает вывод в отчет всех записей этой таблицы. Для сортировки записей по алфавиту название\_товаров следует добавить в список окна диалога Сортировка и группировка соответствующие поля, разместив их *выше* поля группировки Код\_товаров.

При добавлении нового поля сортировки Access автоматически назначает ему порядок сортировки По возрастанию. Чтобы реализовать обратный порядок записей, выберите в раскрывающемся списке ячейки Порядок сортировки соответствующей строки пункт По убыванию.

8. Щелчком на кнопке строки Название\_товаров выделите ее.
9. Поместите указатель на кнопку строки Название\_товаров, нажмите кнопку мыши и перетащите строку на самый верх списка. Отпустите кнопку мыши в тот момент, когда жирная горизонтальная линия, показывающая будущее расположение перемещаемой строки, окажется над первой строкой таблицы.
10. Повторяя шаги 8 и 9, разместите строку Цена Название\_товаров и Код\_товара
11. Закройте окно Сортировка и группировка..

Заголовки тех групп, для которых нет ни одной записи в таблице, в отчет не выводятся.

12. Щелкните на кнопке Вид панели инструментов, чтобы вывести отчет в режиме предварительного просмотра.

Предварительный просмотр

Перед выводом объекта базы данных на принтер нужно настроить параметры страницы и проверить правильность форматирования в режиме предварительного просмотра. Давайте выполним все эти действия на примере таблицы Контакты.

1. В окне базы данных щелкните на кнопке Таблицы.
2. Двойным щелчком на значке Быт\_техника откройте эту таблицу в режиме редактирования данных.
3. Увеличьте ширину столбца Название\_товара, перетащив вправо его правую границу, чтобы адреса не обрезались по длине.
4. Выберите команду Формат > Подтаблица > Развернуть все, чтобы отобразить записи таблицы Список.
5. Выделите столбец Код\_товаров, щелкнув на его заголовке.
6. С помощью команды Формат > Скрыть столбцы скройте выделенный столбец. Этот служебный столбец не нужно печатать.
7. Щелкните на заголовке столбца Ключ подчиненной таблицы.
8. Повторным выполнением команды Формат > Скрыть столбцы уберите и этот столбец.

Режим предварительного просмотра является основным вариантом демонстрации отчетов, поэтому он активизируется двойным щелчком на значке отчета в окне базы данных.

9. Выберите команду Файл > Предварительный просмотр. Вы увидите предполагаемое > оформление бумажной копии таблицы.

### ***Контрольные вопросы***

1. Опишите способы создания форм базы данных.
2. С помощью какого средства можно корректировать созданную форму.
3. В каких случаях используется окно свойств формы.
4. Какие элементы панели управления используются для украшения формы.
5. Как организуются вычисления с помощью форм.
6. Опишите способы создания отчетов по базе данных.
7. Как просматриваются отчеты перед печатью.

### ***Темы самостоятельных работ***

*Этап создания форм и отчетов по базе данных.*

1. Разработать информационную систему ДЕТСКАЯ\_ПОЛИКЛИНИКА.  
В базу данных по поликлинике ввести списки детей и списки сделанных им прививок, при этом надо иметь возможность просматривать как перечень прививок по каждому ребенку, так и списки детей по каждой прививке.
2. Разработать информационную систему НАЛОГОВАЯ ИНСПЕКЦИЯ. Налоговая инспекция создает базу данных юридических лиц по видам деятельности. Юридическое лицо может иметь несколько видов деятельности. Надо иметь возможность просматривать

как список юридических лиц по каждому виду, так и перечень видов деятельности по каждому юридическому лицу.

3. Разработать информационную систему КАФЕДРА. Кафедре ВУЗа требуется вести списки преподавателей и читаемых ими предметов. Каждый преподаватель может читать несколько предметов, а один предмет может читаться несколькими преподавателями. Надо иметь возможность просматривать как список преподавателей по каждому предмету, так и перечень предметов по каждому преподавателю. Требуется также распечатывать анкетные данные о преподавателях.

4. Разработать информационную систему ДЕКАНАТ. Следует вести перечни групп, списки студентов по группам, перечни предметов по группам. А также требуется хранить и распечатывать анкетные данные о студентах.

5. Разработать информационную систему АВТОБАЗА. Необходимо вести учет заказов с перечнем дополнительно устанавливаемых деталей (доукомплектовка), расчет суммы, печать заказа и суммы продаж за определенный период времени.

6. Разработать информационную систему ОКНА&ДВЕРИ. Фирма принимает заказы на изготовление пластиковых окон. Каждый заказ содержит данные о клиенте, размерах окна(высота и ширина). Погонный метр пластика стоит 4900 сум., квадратный метр стекла стоит 3000сум. Кроме пластика окно может комплектоваться ручками, петлями, подоконниками. Необходимо обеспечить ведение заказов от клиентов со сроком изготовления и расчетом стоимости заказа. Обеспечить просмотр списка заказов(размеры окна, клиент, стоимость) на любой вводимый срок изготовления .

### *Список рекомендованной литературы*

1. Михеева В., Харитонов И. Microsoft Access 2003 в подлиннике. Наиболее полное руководство. БХВ-Петербург, 2003.
2. Кен Блюттман. Access трюки. Питер, 2006.
3. Лори Фуллер, Кен Кук, Джон Кауфельд. Access 2007 для чайников. М:Диалектика, 2007.

Ссылки в Интернете.

<http://www.natahaus.ru/>

[www.databases.about.com/od/access.](http://www.databases.about.com/od/access)

[www.microsoft.com](http://www.microsoft.com)  
[iit.metodist.ru](http://iit.metodist.ru)  
[www.300.piter.com](http://www.300.piter.com)

[На оглавление](#)

## **Лекция 12**

### **Назначение Ms Excel и основные возможности. Стандартные операции.**

**Цель лекции:** ознакомить с базовыми понятиями и возможностями электронной таблицы Microsoft Excel.

**Ключевые слова:** рабочая книга, рабочие листы, строки, столбцы, ячейка, адрес ячейки, строка формул, типы данных, типы адресации.

## Содержание

Характеристика приложения

Структура таблицы, строка формул, строка закладок

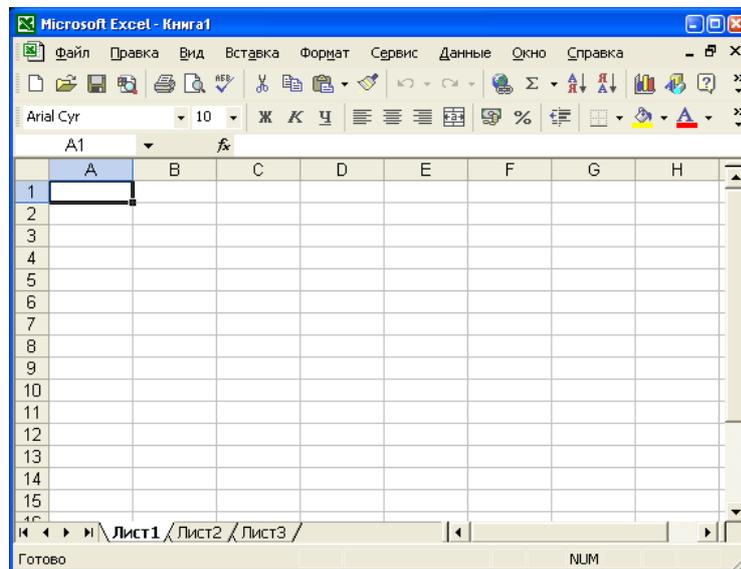
Операции над элементами таблицы.

Типы адресации ячеек.

**Великолепным представителем** семейства программ, обслуживающих вычисления в таблицах является приложение **Microsoft Excel**. Весьма простая в обращении таблица может быть использована деловыми людьми, учеными, бухгалтерами, журналистами. С помощью электронной таблицы можно вести различного рода каталоги, списки, таблицы, составлять статистические и финансовые отчеты, обрабатывать результаты научного эксперимента, вести учет, готовить презентационные материалы. Microsoft Excel достаточно мощная программа, содержащая несколько сотен стандартных функций: математических, финансовых, логических и статистических.

Помимо вычислений и расчетов по сложным формулам и представления результатов графиками и диаграммами (причем любое пересчитывание гигантских колонок чисел моментально отображается в новых изгибах графиков и диаграмм). Excel обладает возможностями форматирования данных хорошего текст - процессора.

При запуске электронной таблицы Excel интерфейс его напоминает интерфейс приложения Microsoft Word, соответствующей версии.



Однако имеются главные отличия:

Во – первых, рабочее поле Excel представляет собой не чистый бланк (как в Word), а пустую таблицу.

Во – вторых, на экране под панелями инструментов **Стандартная** и **Форматирование** (при условии их включения) находится строка, называемая строкой формул. Внизу рабочего поля, т.е. под таблицей находится строка закладок: *Лист1/Лист2/Лист3*... и строка состояния.

Не будем останавливаться на описаниях заголовка окна, строки меню, панелей инструментов, так как они используются почти так же, как и в приложении Microsoft Word, за исключением пункта **Данные**, который отсутствует в меню Word. На имеющихся еще отличиях мы будем останавливаться по ходу изучения материала.

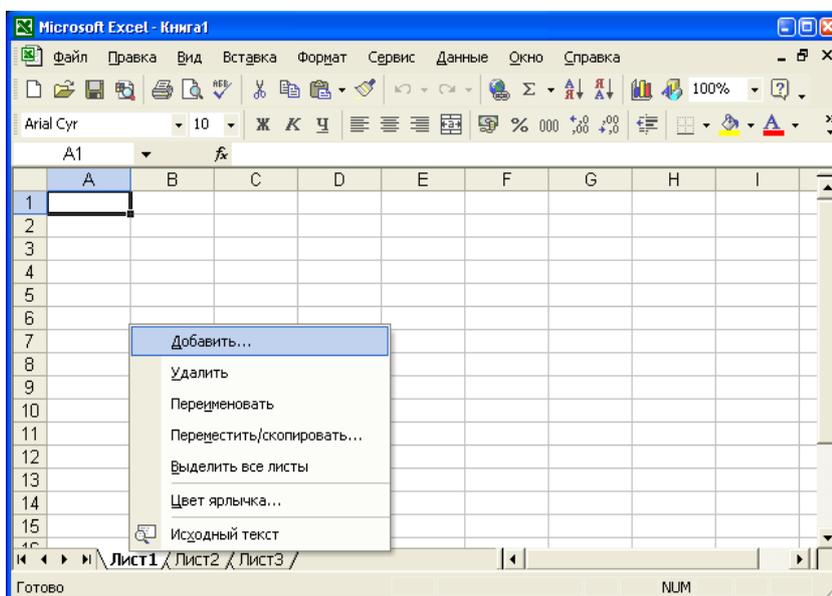
Рассмотрим теперь подробнее строку формул, структуру таблицы, строку закладок и строку состояния.

Строка формул состоит из поля имен и поля формул, в первом отражается адрес клетки, во втором - его содержимое.

Сама таблица состоит из клеток, называемых также ячейками. Каждая таблица содержит 256 столбцов и 65536 строк. Столбцы обозначены большими латинскими буквами от А до Z, затем идут AA-AZ, BA-BZ и т.д. до IV. Строки просто пронумерованы цифрами. Ячейки, оказывающиеся на пересечении пронумерованных строк и столбцов, автоматически получают буквенно-цифровой номер, например А1 или IV.

Строка закладок указывает на имена таблиц. Пользователь может при работе создавать и использовать сразу несколько таблиц. При загрузке Excel обычно в строке закладок имеются имена *Лист1*, *Лист2*, *Лист3* т.е., можно по этим именам переходить на следующую таблицу и работать с ней. Каждый Лист представляет собой чистый бланк, разделенный на ячейки. Все эти листы образуют рабочую книжку, которую можно листать взад и вперед, щелкая кнопкой мыши по ярлыкам или клавишами **Ctrl - PgUp**,

**Ctrl - PgDown**. Для прокручивания ярлыков используются кнопки слева от горизонтальной координатной линейки. Над листами можно выполнить основные операции, которые указаны в таблице:



Кроме того, листы можно добавлять в рабочую книгу следующим образом:

### **Вставка → Лист**

В нижней части окна расположена строка состояния. На правом конце строки располагается надпись **NUM** (при включении **Numlock**), на левом надпись **Готово**, при выполнении очередного действия и имеются поля индикаторов.

Заполненную таблицу (или несколько таблиц, размещенных на рабочих листах) можно сохранить в файл разных форматах. Основным является книга с расширением **.xls**, то есть все рабочие листы одной книги сохраняются в одном файле с указанным расширением. EXCEL является многооконной программой и можно одновременно загружать несколько файлов (рабочих книг). Переход между файлами можно выполнить клавишами **Ctrl+Tab** и **Shift+Ctrl+Tab** или через пункт **Окно** в **Меню**.

Итак, на пересечении строк и столбцов находятся **ячейки**. Обозначение ячейки, состоящее из номера столбца и строки, называется ее **адресом** или **ссылкой**. Одна из ячеек таблицы всегда является активной и выделяется рамкой, которая играет роль курсора.

Ссылаться можно и на группы ячеек, называемые **диапазоном**. Группу ячеек можно выбрать по строке или по столбцу. В этом случае такие диапазоны будут обозначаться,

соответственно *B2:M2* или *A2:A8* (в качестве разделителя используется (:). Группа может представлять собой прямоугольную область и в этом случае диапазон указывается ссылками на верхнюю левую и правую нижнюю ячейки. Например, *A3:F10*.

Для выделения ячеек мышью указатель устанавливает над первой ячейкой диапазона и, удерживая нажатой кнопку мыши, перетаскивают указатель к последней ячейке диапазона. При выделении нескольких ячеек диапазона удерживают нажатой дополнительно клавишу **Ctrl**, пока все ячейки не будут выделены.

Для выделения столбца или строки полностью щелкают соответственно на букве или цифре, обозначающей столбец или строку. Чтобы выделить весь лист, щелкают на кнопке **Выделить все**, расположенной в верхнем левом углу таблицы. Ячейку или диапазон можно именовать, причем эти имена не изменяются при перестройки структуры листа (в отличие от ссылок).

Например, ячейка со ссылкой A1 можно присвоит имя ИТОГ, а диапазону ячеек C3:C9 – имя ВЗНОС присвоение имен выполняется с помощью команд:

***Вставка → Имя → Присвоить,***

или с помощью ввода имени в поле ИМЯ строки формул.

Над элементами таблицы можно выполнить следующие операции:

- добавление строк, столбцов, ячеек
- удаление строк, столбцов, ячеек
- копирование содержимого строк, столбцов, ячеек
- перемещение содержимого строк, столбцов, ячеек

Для перемещения и копирования используется буфер обмена (CLIPBOARD), предназначенный для промежуточного хранения информации на 12 ячеек. С помощью этого буфера фрагменты таблицы можно копировать не только в пределах Excel, но и в другие приложения Windows – программ. Для просмотра буфера обмена необходимо выполнить команды меню

***Правка → Буфер обмена.***

Ячейки являются уникальным «контейнером» для хранения самой разнообразной информации. В них можно поместить различного типа данных. Некоторые данные (рисунки, диаграммы, блок-схемы и другие объекты) можно вставить поверх листа или фоном.

Итак, в ячейки можно ввести:

- числа разного формата (например, 15000; \$ 29,7; 33%);
- текст (например, КВАРТАЛ, ГОД, ОФИС);
- даты и время суток (например, янв. 2005; 11/12/63 или 3:00 pm);
- формулы (например, =B5\*1.001 или =СУММ(B3:B7));
- гиперссылки на адреса Интернета или другие документы.

Число, текст, дата и формула вводятся с клавиатуры. По мере ввода символа появляются активные ячейки и в **строке формул** над листом, где можно также выполнить редактирование данных.

Числа в ячейку можно вводить в разных форматах: десятичном, целом, обыкновенной дробью или экспоненциальной записью. Подробнее форматы будут рассмотрены в следующих лекциях.

Последовательность, состоящая из букв, цифр и специальных знаков, при вводе в ячейки воспринимается как **текст**. Обычно числа в ячейках выравниваются по правому краю, а текст- по левому.

Дату и время Excel хранит в виде серийных чисел. Это облегчает их использование в функциях и формулах. Значение серийного числа определяется на основе подсчета количества дней, прошедших с 1/1/1900.

В ячейки можно вводит разнообразные формулы, содержащие числа, ссылки на ячейки, математические и логические функции, встроенные функции. Ввод формулы начинается со знака «=». Excel хранит формулы во внутреннем формате, выводя формулу в строке формул, а в каждой ячейки, помещает результат вычислений. В формулах можно применять арифметические

операции: сложения (+), вычитания (-), умножения (\*), деления (/), возведения в степень (^) и встроенные функции. Они используются в сочетании с числами, именами и адресами отдельных ячеек, а также диапазонов.

Например, в ячейку F5 можно ввести следующую формулу:

$$= ((F4+C5)/E6)*(B2-D2)+C4^2 \text{ или} \\ = B5+C5+\text{ИТОГ}$$

При вычислении формул действуют обычные правила, принятые в алгебре.

В ячейках можно использовать текст формулы или объединять тексты из разных ячеек. Для этого применяется знак объединения & (амперсant). Например, к вычислениям в таблице необходимо добавить слово «рублей»

	A	B	C
1	B \$	Курс	Доход
2	2500	1.2	300 ← =A2*B2& «рублей»
3	3000	1.2	360 ← =A3*B3& «рублей»
4	4000	1.2	480 ← =A4*B4& «рублей»

В ячейки C2,C3,C4 записываются формулы, указанные выше таблице.

### Типы адресации

Программа Excel позволяет использовать при ссылках на ячейки (или диапазон ячеек) три типа адресации:

- относительная,
- абсолютная,
- поименная.

По умолчанию, ссылки на ячейки в формулах рассматриваются как относительные. Это означает, что при копировании формулы адреса в ссылках автоматически изменяются в соответствии с относительным расположением исходной ячейки и создаваемой копии.

Рассмотрим пример:

	A	B	C	D
1	Площадь рулона	количество рулонов	Цена рулона	Общая стоимость
2	5.5	24	78	1872 ← = B2*C2
3	6	24	125	3000 ← = B3*C3
4	7	24	150	3600 ← = B4*C4
5	8	24	180	4320 ← = B5*C5

В ячейку D2 необходимо записать формулу подсчета общей стоимости рулонов площадью 5.5м<sup>2</sup>, т.е. содержимое ячеек B2 и C2 нужно перемножить, тогда в ячейке D2 ← = B2\*C2. Теперь при копировании этой формулы в ячейку D3 для подсчета стоимости рулонов площадью 6 м<sup>2</sup>, мы получим формулу B3\*C3; соответственно, при копировании в следующие ячейки получим в D4 ← = B4\*C4, в D5 ← = B5\*C5. Таким образом, при копировании относительное указание ссылок в формулах сохраняется, т.е. в ячейках D3, D4, D5 записываются в формулах адреса двух смежных ячеек, находящихся соответственно слева: B3 и C3, B4 и C4, B5 и C5.

Обратите внимание на содержание ячеек в столбце **B**, значения повторяются. Было бы удобнее поместив число в ячейку один раз, затем в формулах ссылаться на нее. В этом случае наша таблица будет выглядеть так:

	A	B	C	D
1	площадь	количество	Цена рулона	Общая стоимость

	рулона	рулонов		
2	5.5	24	78	1872= B2*C2
3	6		125	= B3*C3
4	7		150	= B4*C4
5	8		180	= B5*C5

Если при копировании мы используем относительную адресацию, то получим, как показано в таблице, неверный результат во всех ячейках, кроме D2. для получения правильного результата нам нужно во всех формулах сохранить без изменения адрес B2, т.е. он не должен меняться при переходе на следующие строки. Адресация, при которой адреса ссылок при копировании не изменяется, называется абсолютной. Перед элементами адреса ячейки, использующий абсолютную адресацию, должен стоять знак \$. В нашем примере все формулы в ячейках столбца D должны выглядеть следующим образом:

\$ B \$2\*C3

\$ B \$2\*C4

\$ B \$2\*C5

В ячейку **D2** заносится формула **\$B\$2\*C2** и затем она копируется во все остальные ячейки. При вводе формулы необязательно вводить знак \$ вручную, можно нажать комбинацию клавишей **Ctrl+4** и Excel сделает это сам:

при первом нажатии на B2 → \$ B \$2

при втором → B\$2

при третьем → \$ B2

при четвертом → снова B2

Еще один способ адресации – присвоение имени ячейке. **Имя** обязательно должно начинаться с буквы или символа подчеркивания, остальные символы могут быть буквами, цифрами, символами подчеркивания или точками. Чтобы вставить в формулу имя ячейки или диапазона необходимо выбрать команду

**Вставка → Имя → Присвоить.**

Имена должны быть уникальными, т.е. в пределах одной рабочей книги имена ячеек не должны повторяться. Список всех поименованных ячеек рабочей книги можно посмотреть, щелкнув по стрелочке в поле **Имя**.

Вернемся к нашему примеру. Выделим ячейку **B2** и присвоим ей имя **КОЛИЧЕСТВО**. Теперь в ячейке **D2** вместо указания ссылки **B2** в формуле используем ссылку **КОЛИЧЕСТВО**:  
**D2 ← КОЛИЧЕСТВО\*C2.**

При копировании в ячейках D3, D4 и D5 будут записаны соответственно формулы:

D3 ← **КОЛИЧЕСТВО** \* C3

D4 ← **КОЛИЧЕСТВО** \* C4

D5 ← **КОЛИЧЕСТВО** \* C5

Как бы эта ячейка не меняла свое месторасположение при изменении структуры таблицы, к ней всегда можно обратиться по имени.

### Контрольные вопросы

1. Опишите структуру электронной таблицы.
2. Какие операции можно выполнять над рабочими листами.
3. Что содержит рабочая книга и как она хранится.
4. Ячейка и ссылка на ячейку. Опишите относительную адресацию.
5. В каких случаях используется абсолютная адресация.

### Темы самостоятельных работ

1. Условное форматирование как инструмент для анализа данных.
2. Ошибки при вводе формул. Отслеживание и исправление ошибок.
3. Применение тем и рисунков для оформления страниц в MS EXCEL.
4. Рецензирование документа. Вставка примечаний.
5. Подготовка данных к печати. Область печати.

### *Список рекомендованной литературы*

1. А.Н. Степанов. Информатика. Учебник для ВУЗов. СПб.: Питер, 2006.
2. Б.В. Соболев, А.Б. Галин и др. Информатика. Учебник. Ростов-на-Дону: Феникс, 2007.

[На оглавление](#)

## Лекция 13

### Формулы и функции

**Цель лекции:** ознакомить с вводом формул и средством автозаполнения, с использованием окна Мастера функций при вызове стандартных функций приложения.

**Ключевые слова:** автозаполнение, мастер функций, категории функций, аргументы функций

#### Содержание

Средство автозаполнения  
 Способы вызова и окно Мастера функций  
 Аргументы функций  
 Характеристика категорий функций  
 Логические функции  
 Функции суммирования, среднее значение, число, максимум и минимум.

В приложении Excel реализован механизм, заметно упрощающий копирование формул, ввод повторяющихся данных, известных последовательностей значений (например, месяцев года, числовой шкалы с определенным приращением), называемой автозаполнение. **Автозаполнение** включается при перетаскивании по ячейкам маленького черного квадратика, называемого **маркером заполнения**. Он находится в правом нижнем углу активной ячейки. При установке над маркером указателя мыши он меняется на знак «плюс», т.е. автоматическое заполнение разрешено.

В рассмотренных выше примерах достаточно было ввести формулу в ячейку **D2**, а затем, воспользовавшись механизмом **автозаполнение**, заполнить формулами ячейки **D3, D4, D5**, потянув за «плюсик».

При автоматическом заполнении различных последовательностей данных работают следующие правила:

Вид данных	Последовательность данных	Пример
------------	---------------------------	--------

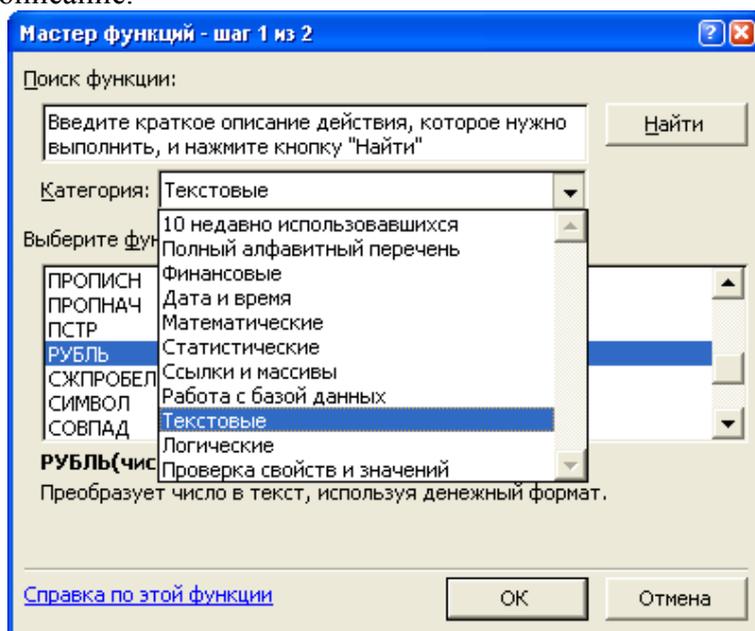
Текст	Текст просто дублируется	Буфер, Буфер, Буфер
Числа	Изменения в зависимости от заданного приращения	100,200,300,400
Текст с числами	Пронумерованные текстовые данные	Шаг1, Шаг2, Шаг3
Дни недели	Последовательность дней недели	Понедельник, вторник, среда
Месяцы	Последовательность месяцев года	Январь, Февраль, Март
Время	Изменения в зависимости от заданного приращения	1:30am,2:00pm, 2:30pm

С помощью формул и функций можно выполнять математическую и статистическую обработку данных листа Excel. Формулы и функции Excel способны обрабатывать величины как конкретных ячеек таблицы, так и целых блоков (например, строк или столбцов). Формулы Excel могут быть очень сложны, а результат их вычислений, в свою очередь, можно использовать в других расчетах. Неоспоримым преимуществом электронной таблицы является то, что при изменении данных листа результаты вычислений моментально обновляются. В библиотеке приложения Excel имеется около 300 функций, с помощью которых можно решать самые разнообразные задачи. Самым простым способом задания формулы является ее непосредственный ввод в строку формул. У этого варианта есть лишь один недостаток — вы должны помнить правила построения формул Excel, названия встроенных функций и методы ссылок на ячейки листа. Excel поддерживает множество стандартных математических функций, которые можно вставлять в формулы.

Список функций можно посмотреть при помощи вызова **Мастера функций**. Сделать это можно :

- **Функции** ;
  - с помощью кнопки **Автосумма**  $\Sigma$ .
- Здесь вы можете увидеть наиболее часто используемые при расчетах функции: *суммировать, число, среднее значение, максимум, минимум*, а также пункт *Другие функции*, выбор которого также открывает окно Мастера функций.

В диалоговом окне **Мастера функций** представлен список категорий, при выделении какой-либо из этих категорий справа в окошке **Выберите Функции** отображается ее состав. Далее, выделив из этого списка имя функции, внизу в окне можно посмотреть ее синтаксическое и текстовое описание.



Функции могут иметь один или несколько аргументов, которые помещаются в скобках, следующих за именем функции. Аргументом может быть ячейка, число или диапазон ячеек.

Например: =СУММ(A5:A9) – подсчитывает сумму значений в ячейках A5, A6, A7, A8 и A9;  
=СРЗНАЧ(G4:G6) – определяет среднее значение чисел в ячейках G4, G5, G6;  
=ЧИСЛО(A1:A10) - определяет количество ячеек в диапазоне(A1:A10);  
=МАКС(B3:C8) - выбирает максимальное из чисел, расположенных в ячейках B3,B4,B5,B6,B7,B8 и C3,C4,C5,C6,C7,C8.

Функции могут входить одна в другую, например:  
=СУММ((F1:F20);МАКС(A9:A19)).

При записи аргументов необходимо соблюдать ее синтаксис, иначе Excel откажется выполнять функцию. Соблюдать эти правила помогает описание, которое выполняется на следующем шаге после выбора функции. На этом этапе появляется окно, помогающее построить запись для выбранной функции без ошибки.

Дадим краткую характеристику основным категориям функций из диалогового окна **Мастера функций**.

Категория **Финансовые**. В эту категорию включено множество функций, вычисляющих проценты по вкладу или кредиту, амортизационные отчисления, норму прибыли и т. д. Как отмечалось выше, список функций выбранной категории можно просмотреть справа от списка категорий в диалоговом окне.

Категория **Дата и Время**. Большинство функций этой категории ведает преобразованиями даты и времени в различные форматы. Две специальные функции **СЕГОДНЯ** и **ТДАТА** вставляют в ячейку текущую дату и время, обновляя их при каждом вызове файла или при внесении любых изменений в таблицу.

Категория **Математические**. Это одна из самых многочисленных категорий в Excel. Здесь представлены все тригонометрические и логарифмические функции, несколько разновидностей функций округления и суммирования (СУММ, СУММПРОИЗВ, СУММКВ, СУММПРАЗНКВ и т.п.)

Категория **Статистические**. Эта категория самая многочисленная (около 80 команд). Представленные функции довольно специфичные. Например, **ФРАСП** – F распределение вероятности или **ГАММАРАСП** - гамма функция. Обычно для подсчета таких величин как дисперсия, коэффициент корреляции или доверительный интервал используются функции из этой категории.

Категория **Ссылки и Массивы**. Функции этой категории позволяют обратиться к массиву данных (по столбцу, строке, прямоугольной области) и получить самую разнообразную информацию о номерах строк и столбцов, об их количестве, об адресе ячейки с нужным числом или текстом и т. п.

Категория **Текстовые**. Команды этой категории выполняют различные операции с текстами: посчитать количество символов в ячейке, определить первый и последний символ, поместить в данную ячейку некоторое количество символов из другой, поменять строчные буквы на прописные, проверить идентичность текстов, найти и заменить текст и т.п.

Категория **Логические**. В этой категории представлены 6 команд. Рассмотрим подробнее функцию **ЕСЛИ**, используемую для проверки значений и формул. Формат этой функции:

=ЕСЛИ (логическое выражение; значение \_если \_ истина;  
значение \_если \_ ложь)

В логическом выражении можно использовать следующие логические операции:

= равно,  
>= больше или равно,  
<= меньше или равно,  
< > не равно.

Если логическое выражение возвращает значение **ИСТИНА**, то вычисляется второй аргумент, если возвращает значение **ЛОЖЬ**, вычисляется третий аргумент. На месте второго и

третьего аргумента могут быть использованы вновь функции **ЕСЛИ**. В этом случае такие функции называется вложенными. Допускается до 7 уровней вложенности функций ЕСЛИ. Пример записи функции:

$$\begin{aligned}
 &=ЕСЛИ (С1 >D1; “УРА!”; “УВЫ”)) \\
 &\quad \text{или} \\
 &=ЕСЛИ (В3>0; МАКС(С3:F3);МИН(С3:F3)) \\
 &\quad \text{или} \\
 &=ЕСЛИ (В1=>86; А1;” ”> ).
 \end{aligned}$$

В функции **ЕСЛИ** можно задать несколько условий (до 30 условий) с помощью союза **И**. В этом случае все условия должны одновременно быть выполненными для того, чтобы логическое выражение в целом приняло значение **ИСТИНА**. Невыполнение хотя бы одного условия приводит к принятию значения **ЛОЖЬ**. Например,

$$=ЕСЛИ (И ( G1>1; G2<1 ); “Верно”; “Не верно”).$$

Функция выполняется следующим образом: проверяется одновременное выполнение условий  $G1 > 1$  и  $G2 < 1$  в ячейках G1 и G2. Если при проверке значение будет **ИСТИНА**, то в ячейку, куда записана эта логическая функция **ЕСЛИ()**, будет записано “Верно”. При невыполнении хотя бы одного из условий будет в ту же ячейку записано “Не верно”.

Аналогично, союз **ИЛИ** позволяет в функции **ЕСЛИ** задать несколько условий. Если хотя бы одно из условий будет выполнено, логическое выражение принимает значение **ИСТИНА**. Значение **ЛОЖЬ** принимается только при одновременном выполнении всех условий. Например,

$$=ЕСЛИ ( ИЛИ (A1<0; A1>0); 1; 0).$$

Если значение в ячейке A1 окажется строго больше 0 или строго меньше 0, то в ячейку, где записана сама функция **ЕСЛИ** заносится значение 1. В случае же невыполнения ни одного из условий – заносится значение 0.

В функции **ЕСЛИ** можно также использовать частицу отрицания **НЕ**, которая инвертирует значение **ИСТИНА** на **ЛОЖЬ** и **ЛОЖЬ** на **ИСТИНА**. Например,

$$=ЕСЛИ ( НЕ (С1>D1* В5); “Угадал”; “Не угадал”).$$

В ячейке с записью функции **ЕСЛИ** будет записано слово “Угадал” при невыполнении условия  $C1 > D1 * B5$  и слово “Не угадал” в случае выполнения.

Рассмотрим еще две функции, включающие в себя проверку условия или критерий.

Функция **СЧЕТЕСЛИ**. Формат записи этой функции:

$$=СЧЕТЕСЛИ (диапазон; критерий).$$

С помощью этой функции подсчитывается количество ячеек в заданном диапазоне (первый аргумент), содержимое которых удовлетворяет указанному критерию (второй аргумент).

Как задается диапазон клеток мы уже знаем, т.е. это могут ячейки по строке ( $A1:F1$ ), по столбцу ( $B10:D20$ ) или может быть прямоугольная область ( $A2:K15$ ). Критерий, заданный в функции, может быть числом, выражением, текстом. Критерий должен быть заключен в кавычки.

Например,

	А	В	С
1	Данные	Количество	
2	Яблоки	32	
3	Апельсины	54	
4	Персики	75	
5	Яблоки	86	
6		2	$2 \leftarrow =СЧЕТЕСЛИ (A2:A5; “Яблоки”)$

По заданной таблице нам нужно определить количество завозов яблок. Для этого в ячейку С6 запишем **=СЧЕТЕСЛИ (A2:A5;”Яблоки”)**. Результатом выполнения функции будет число 2.

Функция **СУММЕСЛИ**. Формат записи этой функции:

**=СУММЕСЛИ** (диапазон; критерий; диапазон \_ суммирования)

Первый аргумент указывает не выбранный для проверки диапазон ячеек. Второй аргумент - условие проверки. Третий аргумент - диапазон ячеек, из которых будут включаться в сумму только те, для которых выполняется критерий. Например,

	А	В	С
1	Стоимость имущества	Комиссионные	
2	100000	7000	
3	200000	14000	
4	300000	21000	
5	400000	28000	
6			63000 ← =СУММЕСЛИ(A2:A5;">16000";B2:B5)

Из диапазона В2:В5 будут для суммирования выбираться только те ячейки, для которых из диапазона А2:А5 выполнено условие >160000.

Категория **Проверка свойств и значений**. Эта категория содержит команды, с помощью которых можно получить информацию о типе данных в ячейках (число, текст, формула и т.д.), о формате, текущей операционной среде, о типе ошибки, возникшей в формуле и т.д.

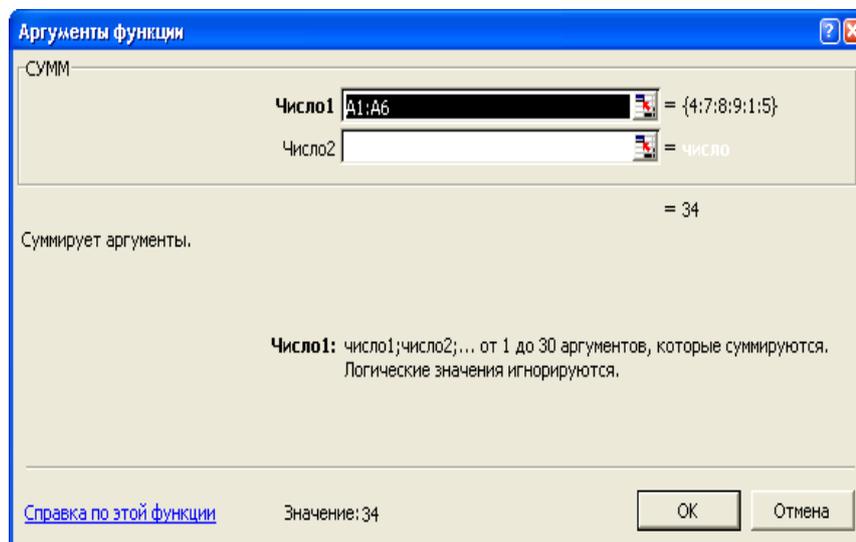
Категория **Работа с базой данных**. Здесь находятся функции для статистического подсчета различных характеристик, например, дисперсии по выборке из базы данных, генеральной совокупности, стандартного отклонения по выборке, операции со столбцами и строками базы данных и т.д.

Рассмотрим работу диалоговых окон на примере вызова функций Excel СУММ, СРЗНАЧ, ЧИСЛО, МАКСИМУМ, МИНИМУМ.

Общая запись функций выглядит так:

**ФУНКЦИЯ** (число1;число2;...), где число1, число2 и т.д.г аргументы функции.

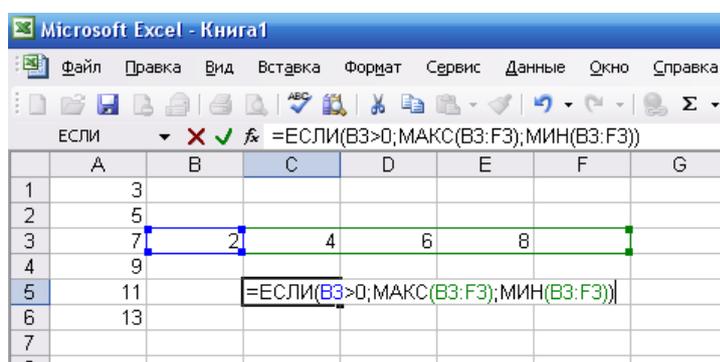
Поскольку эти функции используются часто, на панели инструментов предусмотрена кнопка **Автосумма** , ускоряющая доступ к этим функциям. Рассмотрим на примере работу функции **СУММ**. После вызова запись этой функции окажется в строке формул в виде **СУММ( )**. Аргументы функции можно далее вводить вручную с клавиатуры или выделяя диапазон мышкой, либо на этом этапе открывается окно с полями для ввода аргументов: **Число1**, **Число2** и т. д., куда также вручную можно ввести аргументы функции. Для указания их мышкой необходимо щелкнуть на кнопку , находящуюся справа от поля ввода, и выделить ячейки, содержащие аргументы функции.



При суммировании значений по строке или по столбцу достаточно выделить диапазон ячеек, захватив свободную колонку или снизу строку, и щелкнуть по кнопке  $\Sigma$ . Например, для суммирования данных в таблице достаточно выделить диапазоны A1:A7 и B3:G3. Результат автоматически появится, соответственно, в ячейках A7 и G3.

	A	B	C	D	E	F	G	H
1	3							
2	5							
3	7	2	4	6	8	12	32	
4	9							
5	11							
6	13							
7	48							
8								

При выполнении других функций, обычно, вызывают их в активную ячейку и набирают необходимые аргументы. Например, в таблице в ячейку C5 набрана функция ЕСЛИ() с аргументами в виде условного выражения – B3>0,



и функций МАКС() и МИН(). После выполнения функции вид ее записи можно видеть только в строке формул, в активную ячейку размещается результат вычисления.

### **Контрольные вопросы**

1. Приведите пример записи формулы в ячейку таблицы.
2. Каким образом можно загрузить окно Мастера функций.
3. Какие категории функций представлены в окне Мастера функций.
4. Как можно вызвать нужную вам функцию.
5. Какими могут быть аргументы у функций.
6. Опишите работу наиболее часто используемых функций. Как происходит обращение к этим функциям.
7. Опишите работу логических функций.

### **Темы самостоятельных работ**

1. В заданном списке сотрудников найти фамилию человека, получающего самую большую зарплату (используя функции адреса).
2. Создание связанной таблицы. Управление связанной таблицей в EXCEL-2007.

3. Вычисление итогов. По данным о продажах из разных магазинов различных областей необходимо получить итоговую информацию о сумме и количестве поставки в каждую область, а также сведения по каждому наименованию товара.
4. Создание сводных таблиц.
5. Работа с функциями консолидации.

### *Список рекомендованной литературы*

1. А.Н. Степанов. Информатика. Учебник для ВУЗов. СПб.: Питер, 2006.
2. Б.В. Соболев, А.Б. Галин и др. Информатика. Учебник. Ростов-на-Дону: Феникс, 2007.

[На оглавление](#)

## Лекция 14

### Работа с диаграммами в Excel

**Цель лекции:** получить навыки построения диаграмм со всеми оформительскими элементами, используя окно Мастера диаграмм.

**Ключевые слова:** диаграмма, тип диаграммы, источник данных, легенда диаграммы.

#### Содержание

Вызов Мастера диаграмм.

Шаги построения диаграмм: выбор типа, выбор диапазона.

Оформление диаграммы и ее размещение.

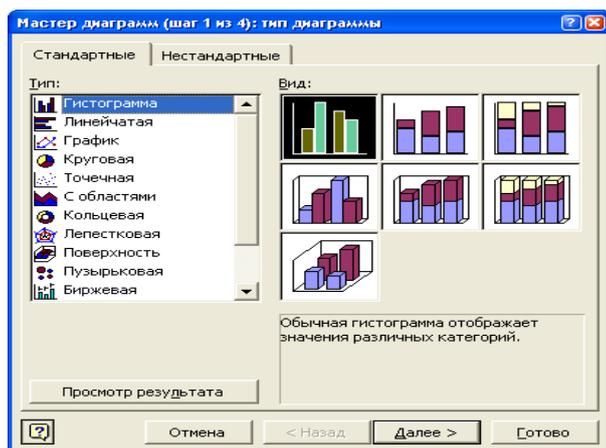
Форматирование и редактирование диаграммы.

Диаграммы в Excel являются наглядным графическим представлением строк и столбцов данных. Они помогают проводить анализы и сравнения данных, увидеть тенденции, которые трудно разглядеть за набором чисел. Кроме того, они являются неизменными элементами презентаций. На диаграмме числовые данные изображаются в виде точек, линий, полос, столбиков, секторов и в другой форме.

Для создания диаграмм необходимо выбрать команду **Диаграмма** в меню **Вставка** или щелкнуть по кнопке  на панели инструментов. При выполнении команды загружается диалоговое окно **Мастера диаграмм**, где пользователь может выбрать тип, формат и другие параметры диаграмм. При создании диаграммы на любом шаге (всего 4 шага) можно щелкнуть по кнопке **Готово**, тогда **Мастер диаграмм** самостоятельно закончит построение.

Рассмотрим шаги построения:

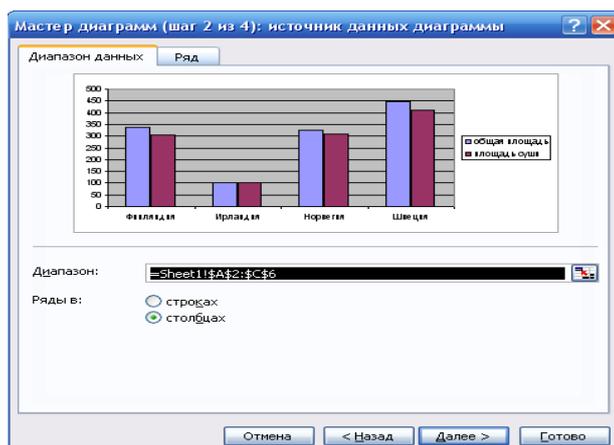
**1 шаг.** На этом шаге программа Excel позволяет выбрать тип диаграммы. Основных типов диаграмм-14, и каждый имеет несколько разновидностей. Например, круговые диаграммы используются для показа соотношений между долями целого, а линейчатые - для наглядного количественного сопоставления. Кроме того, в этом же окне можно посмотреть список нестандартных типов диаграмм.



**2 шаг.** На этом шаге выбирается источник данных. Хотя диапазон ячеек с данными, которые должны попасть на диаграмму, можно выделить до загрузки Мастера диаграмм. В текстовом поле Диапазон окна этого шага будут помещены адреса предварительно выделенных ячеек. Например,

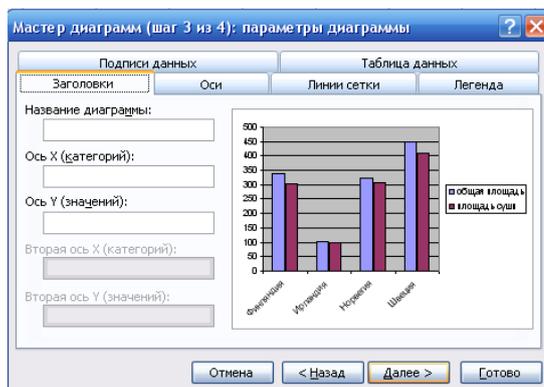
\$A\$2: \$C\$6.

В этом же окне следует установить переключатель, указывающий место расположения данных - в строках или в столбцах (переключатель: ряды в столбцах/столбцах).

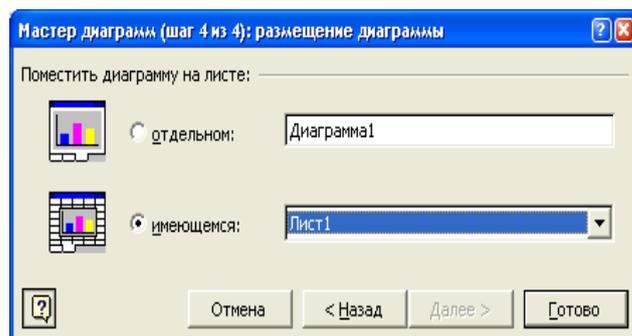


Далее переходим к третьему шагу кнопкой **Далее**.

**3 шаг.** В этом окне задают название диаграммы, название осей, убирают либо включают осевые линии, легенду и прочие детали оформления диаграммы. **Легенда** - это подпись, определяющая графические характеристики (узор или цвета) рядов данных на диаграмме. Далее переходим к заключительному шагу кнопкой **Далее**.



**4 шаг.** На этом шаге можно выбрать место размещения созданной диаграммы - на отдельном листе книги или на одном из уже имеющихся. Затем необходимо нажать клавишу **ГОТОВО**.

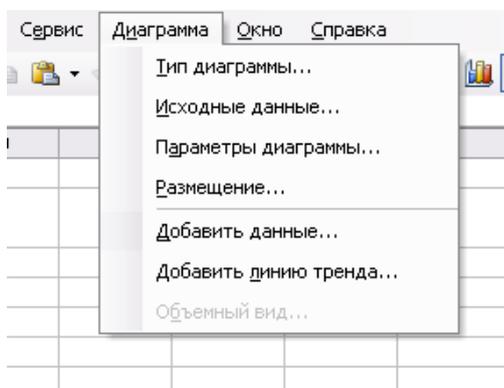


Ценное качество мастера - возможность “шагать назад”. Если необходимо на одной диаграмме отобразить сразу много переменных (данные по нескольким столбцам или рядам), можно выбрать соответствующий тип диаграммы, например, гистограмму.

### Форматирование и редактирование диаграммы

Если требуется изменить вид диаграммы, то сделать это несложно. Причем, можно изменить любые её атрибуты, будь то оси, подписи, легенда, цвета графических элементов, адреса ячеек с исходными данными.

Доступ к параметрам форматирования данных диаграммы осуществляется через -меню **Диаграммы**



-контекстное меню (щелкаем правой кнопкой мыши на предварительно выделенной диаграмме)

или

- панель инструментов **Диаграмма**.

В этом меню имеется команда **Исходные данные**. Если ею воспользоваться, то откроется одноименное диалоговое окно, в котором будут представлены поля, позволяющие изменить адреса диапазонов ячеек, содержащих отображенные на диаграмме данные.

На панели **Диаграмма** также представлены инструменты, предназначенные для преобразования построенных диаграмм. Через данную панель удобно работать с отдельными фрагментами диаграммы. Здесь имеется раскрывающийся список **Элементы Диаграммы**. Он содержит пункты контекстно-связанные с типом диаграммы. Щелчок на соответствующем пункте списка выделяет определенные элементы: оси, заголовки, подписи, данные и прочие, - и позволяет приступить к редактированию.

Щелкая по любому элементу диаграммы (он выделяется черными квадратиками) или по пустому месту внутри диаграммы (выделяется вся диаграмма) можно копировать, удалять, перемещать, уменьшать или растягивать как всю диаграмму, так и отдельные ее элементы.

Между диаграммой и исходными данными существует динамическая связь. Если вы что-то поменяете в таблице, соответственно изменится и диаграмма.

### ***Контрольные вопросы***

1. За какое минимальное количество шагов можно построить диаграмму.
2. На каком шаге можно вынести на диаграмму дополнительные данные.
3. Что такое легенда диаграммы.
4. Какие действия выполняются на последнем шаге построения диаграммы.
5. Как можно редактировать уже готовую диаграмму.

### ***Темы самостоятельных работ***

1. Работа с БД. Функция сортировки.
2. Функция фильтрации.
3. Создание форм
4. Создание сводных диаграмм
5. Подбор параметров.
- 6.

### ***Список рекомендованной литературы***

1. А.Н.Степанов. Информатика. Учебник для ВУЗов. СПб.:Питер, 2006.
2. Б.В.Соболь, А.Б.Галин и др. Информатика. Учебник. Ростов-на-Дону: Феникс, 2007.

[На оглавление](#)

## **Лекция 15**

### **Создание макросов**

**Цель лекции:** ознакомить с возможностями автоматизирования задачи с помощью макроса, научить создавать макросы и управлять ими.

**Ключевые слова:** макрос, запись макроса, запуск и управление макросом.

#### **Содержание**

Определение макроса  
Создание макроса в Microsoft Excel  
Запуск макроса  
Управление макросом

#### **Макросы**

Если требуется периодическое выполнение задачи в Microsoft Excel, можно автоматизировать задачу с помощью макроса. Макрос — это последовательность команд и функций, хранящаяся в

модуле Visual Basic. Ее можно выполнять всякий раз, когда необходимо выполнить данную задачу.

Например, если в ячейки часто вводятся длинные текстовые строки, можно создать макрос для форматирования этих ячеек для охватывания ими текста.

Для создания макроса необходимо установить один из уровней безопасности средний или низкий. Затем выбираются команда Макрос и подкоманда Начать запись. В поле **Имя макроса** вводится имя макроса.

- Первым знаком имени макроса должна быть буква. Остальные знаки могут быть буквами, цифрами или знаками подчеркивания. В имени макроса не допускаются пробелы; в качестве разделителей слов следует использовать знаки подчеркивания.
- Не используйте имя макроса, являющееся ссылкой на ячейку, либо будет выдано соответствующее сообщение об ошибке.

Если этот макрос потребуется запускать нажатием сочетания клавиш на клавиатуре, то вводится буква в поле **Сочетание клавиш**. Допускается использование сочетаний CTRL+*буква* (для строчных букв) или CTRL+SHIFT+*буква* (для прописных букв), где *буква* — любая буквенная клавиша на клавиатуре. Нельзя использовать сочетания клавиш с цифрами и специальными знаками, такими как @ или #. Выбранное сочетание клавиш заменяет все совпадающие стандартные сочетания клавиш Microsoft Excel на то время, пока открыта книга, содержащая данный макрос.

Если этот макрос требуется всегда при работе в Microsoft Excel, то необходимо выбрать вариант **Личная книга макросов**. После завершения действий по созданию макроса нажимается кнопка на панели инструментов **Остановить запись**.

При записи макроса Microsoft Excel сохраняет информацию о каждом шаге выполнения последовательности команд. Последующий запуск макроса вызывает повторение («воспроизведение») команд. Если при записи макроса была допущена ошибка, сделанные исправления также будут записаны. **Visual Basic** хранит каждый записанный макрос в отдельном модуле, присоединенном к книге. Макрос можно запустить, выбрав его из списка в диалоговом окне **Макрос**. Чтобы макрос запускался после каждого нажатия определенной кнопки или сочетания клавиш, можно назначить макросу кнопку на панели инструментов, сочетание клавиш или графический объект на листе.

Чтобы выполнить макрос, находясь в редакторе Visual Basic, выберите команду **Macros** в меню **Tools** В поле **Macro name** введите имя того макроса, который нужно выполнить, а затем нажмите кнопку **Run**.

**Управление макросами.** После записи макроса код программы можно посмотреть в редакторе Visual Basic для исправления ошибок и внесения требуемых изменений в макрос. Например, если макрос, переносящий текст по словам, должен также выделять текст полужирным шрифтом, можно записать другой макрос, выделяющий шрифт полужирным шрифтом, и скопировать инструкции из этого нового макроса в старый макрос, переносящий текст по словам.

Редактор Visual Basic разработан для упрощения написания и редактирования программ начинающими пользователями и предоставляет обширную оперативную справку. Для внесения простейших изменений в макросы нет необходимости изучать программирование на языке Visual Basic. Редактор Visual Basic позволяет редактировать макросы, копировать макросы из одного

модуля в другой, копировать макросы из других книг, переименовывать модули, хранящие макросы и переименовывать макросы.

Работу макроса можно прервать. Для этого необходимо выполнить одно из следующих действий.

- Нажать клавишу ESC, а затем в диалоговом окне **Microsoft Visual Basic** нажать кнопку **End**.
- Чтобы предотвратить автоматическое выполнение макроса при запуске Microsoft Excel, во время запуска необходимо удерживать нажатой клавишу SHIFT.

**Безопасность макросов.** Microsoft Excel обеспечивает защиту от вирусов, которые могут передаваться макросами. Если макросы используются совместно с другими пользователями, макросы могут быть снабжены цифровой подписью, так что пользователи могут проверить надежность источника макросов. Всякий раз при открытии книги, содержащей макросы, можно проверить источник макросов, прежде чем разрешить их.

### *Контрольные вопросы*

1. Дайте определение макроса.
2. Как создается макрос в Excel.
3. Какой уровень безопасности требуется для выполнения макроса.
4. Как можно внести изменения в созданный макрос.
5. Как предотвратить автоматический запуск макроса.

### *Темы для самостоятельных работ*

1. Назначение кнопки панели инструментов для запуска макроса.
2. Назначение графического объекта для запуска макроса.
3. Сообщения об ошибках при выполнении макроса.
4. Удаление макроса.
5. Создание макроса для элемента управления ActiveX.

### *Список рекомендованной литературы*

1. А.Н. Степанов. Информатика. Учебник для ВУЗов. СПб.: Питер, 2006.
2. Б.В. Соболев, А.Б. Галин и др. Информатика. Учебник. Ростов-на-Дону: Феникс, 2007.

[На оглавление](#)