

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новгородский государственный университет имени Ярослава Мудрого»

Старорусский политехнический колледж (филиал)

КОНСПЕКТЫ ЛЕКЦИЙ

ОУД.08. ИНФОРМАТИКА

Специальность 15.02.08 Технология машиностроения

Квалификация техник

Старая Русса
2021 г.

Распределение фонда времени лекционных занятий

№ п/п	Наименование раздела и темы лекции	Вид лекции	Кол-во часов по очной форме обучения
1.	Информация и информационные процессы.		9
1.1	Информация. Информационная деятельность человека.	<i>текущая</i>	4
1.2	Системы счисления	<i>текущая</i>	2
1.3	Основы алгоритмизации	<i>текущая</i>	3
2	Средства информатизации и коммуникационных технологий		7
2.1	История компьютера	<i>текущая</i>	2
2.2	Состав персонального компьютера	<i>текущая</i>	3
2.3	Элементная база ПК	<i>текущая</i>	2
3	Технологии создания и преобразования информационных объектов		12
3.1	Текстовые редакторы	<i>текущая</i>	4
3.2	Графика и звук	<i>текущая</i>	6
3.3	Электронная презентация	<i>текущая</i>	2
4	Технологии работы с информационными структурами		6
4.1	Электронные таблицы	<i>текущая</i>	3
4.2	Базы данных	<i>текущая</i>	3
5	Телекоммуникационные технологии		6
5.1	Компьютерные сети	<i>текущая</i>	2
5.2	Интернет	<i>текущая</i>	4

РАЗДЕЛ 1 ИНФОРМАЦИЯ И ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ

Тема 1.1 Информация. Информационная деятельность человека.

План лекции

1. Введение в дисциплину, предмет и содержание курса. Основные понятия и определения. Этапы информационного развития общества. Информационные ресурсы общества.

2. Понятие информации. Информация и ее свойства. Информация и управление. Информация и моделирование. Единицы измерения информации.

3. Файловая система хранения, поиска и обработки информации.

1. Введение в дисциплину, предмет и содержание курса. Основные понятия и определения. Этапы информационного развития общества. Информационные ресурсы общества.

Термин *информатика* (informatique) возник в 60-х гг. XX века во Франции для названия области, занимающейся автоматизированной обработкой информации с помощью электронных вычислительных машин. Французский термин образован путем слияния слов *information* («информация») и *automatique* («автоматика») и дословно означает «информационная автоматика». Широко распространён также англоязычный вариант этого термина *computer science*, что означает буквально «компьютерная наука».

Согласно определению, введенному Французской Академией, информатика – «наука об осуществляемой с помощью автоматических средств целесообразной обработке информации, рассматриваемой как представление знаний и сообщений в технической, экономической и социальной областях». Эта точка зрения на информатику как науку была закреплена в 1978 г. Международным конгрессом в Японии. На этом конгрессе официально были закреплены за понятием «информатика» области, связанные с разработкой, созданием, использованием, материально-техническим обслуживанием систем

обработки информации, включая компьютеры и их программное обеспечение, а также организационные, коммерческие, административные и социально-политические аспекты *компьютеризации*.

В СССР это направление начало развиваться еще в 50-е гг. в рамках *кибернетики*, но как самостоятельная научная дисциплина признание получила в 1983 г. на сессии годовичного Общего собрания АН СССР. В этом же году академик В.М. Глушков выпустил монографию «Основы безбумажной информатики», в которой связал будущее науки с компьютерными технологиями. Многие ученые поддержали данную точку зрения (среди них можно назвать А.П. Александрова, Е.П. Велихова, А.А. Дородницына, А.П. Ершова, В.С. Михалевича).

В 90-е гг. на первый план выходит тенденция объединения научных и технических средств. Теперь речь идет об информатике как комплексной научной и технической дисциплине, в составе которой выделяются теоретический и практический аспекты. Впервые такое определение информатики как «интегральной научной дисциплины» было предложено профессором Темниковым Ф.Б. в 1963 г., в западной англоязычной науке эти аспекты получили название «informatics» и «computer science». Проиллюстрировать развитие данной точки зрения можно, указав на активное формирование большого количества отраслевых информатик, в определении которых присутствуют теоретические и прикладные аспекты (для примера можно упомянуть биологическую, историческую, правовую, музейную, промышленную информатики).

В США обобщающая концепция получила название информационной науки (information science) и начала формироваться в 80-е гг., а в начале 90-х гг. она становится актуальной по всей Европе. Информационную науку можно определить как междисциплинарную область знаний о различных формах проявления информации, информационных потоках, процессах и системах, о распространении информации в обществе, а также о психологических и социальных последствиях этих явлений и процессов.

В настоящее время продолжает активно использоваться понимание информатики как инфраструктуры, отражающей вопросы использования компьютеров в практической деятельности. На волне подобного отношения к информатике формируется понятие специальной информатики, представляющей собой разработку и изучение специальных прикладных программ, предназначенных для выполнения конкретных действий по конкретным специальностям (информатика для секретаря-референта; информатика для делопроизводителя и т.д.).

Итак, существует множество определений информатики, что связано с многогранностью ее функций, возможностей, форм, методов. Одно из наиболее общих определений такое.

Информатика – это область человеческой деятельности, связанная с процессами преобразования информации с помощью компьютеров и их взаимодействием со средой применения.

Часто возникает путаница понятий «информатика» и «кибернетика». Попробуем разъяснить их сходство и различие.

Кибернетика – это наука об общих принципах управления в различных системах: технических, биологических, социальных и др.

Информатика занимается изучением процессов преобразования информации более широко, практически не решая задачи управления различными объектами, как кибернетика. Информатика появилась благодаря развитию компьютерной техники, базируется на ней и совершенно немыслима без нее. Кибернетика развивается сама по себе и, хотя достаточно активно использует достижения компьютерной техники, совершенно от нее не зависит, т.к. строит различные модели управления объектами.

Информатика имеет широчайший диапазон применения. Её основные направления:

- разработка вычислительных систем и программного обеспечения;
- теория информации;

- методы искусственного интеллекта;
- системный анализ;
- методы машинной графики, анимации, средства мультимедиа;
- средства телекоммуникации;
- разнообразные приложения в других видах хозяйственной и общественной деятельности людей.

2. Понятие информации. Информация и ее свойства. Информация и управление. Информация и моделирование. Единицы измерения информации.

Информация - очень емкое понятие, в которое вмещается весь мир: все разнообразие вещей и явлений, вся история, все тома научных исследований, творения поэтов и прозаиков. И все это отражается в двух формах – непрерывной и дискретной. Обратимся к их сущности. Объекты и явления характеризуются значениями физических величин. Например, массой тела, его температурой, расстоянием между двумя точками, длиной пути (пройденного движущимся телом), яркостью света и т.д. Природа некоторых величин такова, что величина может принимать принципиально любые значения в каком-то диапазоне. Эти значения могут быть сколь угодно близки друг к другу, исчезающе малоразличимы, но все-таки, хотя бы в принципе, различаться, а количество значений, которое может принимать такая величина, бесконечно велико. Такие величины называются непрерывными величинами, а информация, которую они несут в себе, непрерывной информацией. Слово “непрерывность” отчетливо выделяет основное свойство таких величин - отсутствие разрывов, промежутков между значениями, которые может принимать величина. Масса тела - непрерывная величина, принимающая любые значения от 0 до бесконечности. То же самое можно сказать о многих других физических величинах - расстоянии между точками, площади фигур, напряжении электрического тока. Кроме непрерывных существуют иные величины, например, количество людей в комнате, количество электронов в

атоме и т.д. Такого рода величины могут принимать только целые значения, например, 0, 1, 2, ..., и не могут иметь дробных значений. Величины, принимающие не всевозможные, а лишь вполне определенные значения, называют дискретными. Для дискретной величины характерно, что все ее значения можно пронумеровать целыми числами 0,1,2,...

Примеры дискретных величин:

- геометрические фигуры (треугольник, квадрат, окружность);
- буквы алфавита;
- цвета радуги;

Можно утверждать, что различие между двумя формами информации обусловлено принципиальным различием природы величин. В то же время непрерывная и дискретная информация часто используются совместно для представления сведений об объектах и явлениях.

Виды информации, свойства и единицы измерения.

Виды информации, воспринимаемые компьютером: текстовая, числовая, звуковая, графическая, мультимедийная.

Информацию можно сгруппировать в две большие группы: дискретную (прерывную) и аналоговую (непрерывную).

Информация (ресурсы, знания) подразделяется на декларативную (я знаю, что ...) и процедурную (я знаю, как ...)

Свойства информации (с точки зрения бытового подхода к определению информации):

- * релевантность -- способность информации соответствовать нуждам (запросам) потребителя;
- * полнота -- свойство информации исчерпывающе (для данного потребителя) характеризовать отображаемый объект или процесс;
- * своевременность -- способность информации соответствовать нуждам потребителя в нужный момент времени;

* достоверность -- свойство информации не иметь скрытых ошибок. Достоверная информация со временем может стать недостоверной, если устареет и перестанет отражать истинное положение дел;

* доступность -- свойство информации, характеризующее возможность ее получения данным потребителем;

* защищенность -- свойство, характеризующее невозможность несанкционированного использования или изменения информации;

* эргономичность -- свойство, характеризующее удобство формы или объема информации с точки зрения данного потребителя.

1 бит -- минимальная единица измерения информации, при вероятностном подходе к измерению информации, принятом в теории информации, это количество информации, уменьшающее неопределенность знаний в 2 раза.

Связь между единицами измерения информации: ** 1 байт = 8 бит,

* 1 Кб (килобайт) = 1024 байт = 2¹³ бит;

* 1 Мб (мегабайт) = 2¹⁰ (1024) Кб = 2²⁰ (1048576) байт = 2²³ бит;

* 1 Гб (гигабайт) = 2¹⁰ Мб = 2²⁰ Кб = 2³⁰ байт = 2³³ бит;

* 1 Тб (терабайт) = 2¹⁰ Гб = 2²⁰ Мб = 2³⁰ Кб = 2⁴⁰ байт = 2⁴³ бит.

При объемном подходе к измерению информации, характерном для компьютерной обработки данных, информативность сообщения определяется количеством символов, его составляющих.

3. Файловая система хранения, поиска и обработки информации.

Файл- это определенное количество информации (программа или данные), имеющее имя и хранящееся в долговременной (внешней) памяти.

Имя файла. Имя файла состоит из двух частей, разделенных точкой: собственно имя файла и расширение, определяющее его тип (программа, данные и так далее). Собственно имя файлу дает пользователь, а тип файла обычно задается программой автоматически при его создании.

Таблица 1. Типы файлов и расширений

Тип файла	Расширения
Программы	exe, com
Текстовые файлы	txt, doc
Графические файлы	bmp, d1 T,]рдидр
Звуковые файлы	wav, mid
Видеофайлы	avi
Программы на языках программирования	bas, pas и др

Файловая система. На каждом носителе информации (гибком, жестком или лазерном диске) может храниться большое количество файлов. Порядок хранения файлов на диске определяется используемой файловой системой.

Каждый диск разбивается на две области: область хранения файлов и каталог. Каталог содержит имя файла и указание на начало его размещения на диске.

Для дисков с небольшим количеством файлов (до нескольких десятков) может использоваться одноуровневая файловая система, когда каталог (оглавление диска) представляет собой линейную последовательность имен файлов. Такой каталог можно сравнить с оглавлением детской книжки, которое содержит только названия отдельных рассказов.

Таблица 2. Одноуровневый каталог

Имя файла	Номер начального сектора
Файл_1	
Файл_2	
.....	
Файл_112	

Если на диске хранятся сотни и тысячи файлов, то для удобства поиска используется многоуровневая иерархическая файловая система, которая имеет древовидную структуру.

Начальный, корневой каталог содержит вложенные каталоги 1-го уровня, в свою очередь, каждый из последних может содержать вложенные каталоги 2-го уровня и так далее. Необходимо отметить, что в каталогах всех уровней могут храниться и файлы.

Файловая система— это система хранения файлов и организации каталогов.

Путь к файлу.Путь к файлу вместе с именем файла называют иногда полным именем файла.

Операции над файлами.В процессе работы на компьютере наиболее часто над файлами производятся следующие операции:

- копирование (копия файла помещается в другой каталог);
- перемещение (сам файл перемещается в другой каталог);
- удаление (запись о файле удаляется из каталога);
- переименование (изменяется имя файла).

Форматирование дисков.Для того чтобы на диске можно было хранить информацию, диск должен быть отформатирован, то есть должна быть создана физическая и логическая структура диска.

Формирование физической структуры диска состоит в создании на диске концентрических дорожек, которые, в свою очередь, делятся на секторы. Для этого в процессе форматирования магнитная головка дисководов расставляет в определенных местах диска метки дорожек и секторов.

После форматирования гибкого диска 3,5" его параметры будут следующими:

- информационная емкость сектора — 512 байтов;
- количество секторов на дорожке — 18;
- дорожек на одной стороне — 80;
- сторон — 2.

Логическая структура гибких дисков. Логическая структура магнитного диска представляет собой совокупность секторов (емкостью 512 байтов), каждый из которых имеет свой порядковый номер (например, 100).

Сектора нумеруются в линейной последовательности от первого сектора нулевой дорожки до последнего сектора последней дорожки.

На гибком диске минимальным адресуемым элементом является **сектор**.

При записи файла на диск будет занято всегда целое количество секторов, соответственно минимальный размер файла — это размер одного сектора, а максимальный соответствует общему количеству секторов на диске.

Файл записывается в произвольные свободные сектора, которые могут находиться на различных дорожках. Например, Файл_1 объемом 2 Кбайта может занимать сектора 34, 35 и 47, 48, а Файл_2 объемом 1 Кбайт — сектора 36 и 49.

Таблица 3. Логическая структура гибкого диска формата 3,5" (2-я сторона)



Для того чтобы можно было найти файл по его имени, на диске имеется каталог, представляющий собой *базу данных*.

Запись о файле содержит имя файла, адрес первого сектора, с которого начинается файл, объем файла, а также дату и время его создания.

Таблица 4. Структура записей в каталоге

Имя файла	Адрес первого сектора	Объем файла, Кбайт	Дата создания	Время создания
Файл 1			14 01 99	14 29
Файл 2			14 01 99	14 45

Полная информация о секторах, которые занимают файлы, содержится в таблице размещения файлов (FAT — File Allocation Table). Количество ячеек FAT соответствует количеству секторов на диске, а значениями ячеек являются

цепочки размещения файлов, то есть последовательности адресов секторов, в которых хранятся файлы.

Цепочка размещения для файла Файл_1 выглядит следующим образом: в начальном 34-м секторе хранится адрес 35, в 35-м секторе хранится адрес 47, в 47-м — 48, в 48-м — знак конца файла (K).

Для размещения каталога — базы данных и таблицы FAT на гибком диске отводятся секторы со 2 по 33. Первый сектор отводится для размещения загрузочной записи операционной системы. Сами файлы могут быть записаны, начиная с 34 сектора.

Виды форматирования. Существуют два различных вида форматирования дисков: полное и быстрое форматирование. Полное форматирование включает в себя как физическое форматирование (проверку качества магнитного покрытия дискеты и ее разметку на дорожки и секторы), так и логическое форматирование (создание каталога и таблицы размещения файлов). После полного форматирования вся хранившаяся на диске информация будет уничтожена.

Быстрое форматирование производит лишь очистку корневого каталога и таблицы размещения файлов. Информация, то есть сами файлы, сохраняется и в принципе возможно восстановление файловой системы.

Информационная емкость гибких дисков. Рассмотрим различие между емкостью неформатированного гибкого магнитного диска, его информационной емкостью после форматирования и информационной емкостью, доступной для записи данных.

Заявленная емкость неформатированного гибкого магнитного диска формата 3,5" составляет 1,44 Мбайт.

Рассчитаем общую информационную емкость отформатированного гибкого диска:

Количество секторов: $N = 18 \times 80 \times 2 = 2880$.

Информационная емкость:

$512 \text{ байт} \times N = 1\,474\,560 \text{ байт} = 1\,440 \text{ Кбайт} \approx 1,40625 \text{ Мбайт}$.

Однако для записи данных доступно только 2847 секторов, то есть информационная емкость, доступная для записи данных, составляет:

$$512 \text{ байт} \times 2847 = 1\,457\,664 \text{ байт} = 1423,5 \text{ Кбайт} = 1,39 \text{ Мбайт}.$$

Логическая структура жестких дисков. Логическая структура жестких дисков несколько отличается от логической структуры гибких дисков. Минимальным адресуемым элементом жесткого диска является кластер, который может включать в себя несколько секторов. Размер кластера зависит от типа используемой таблицы FAT и от емкости жесткого диска.

На жестком диске минимальным адресуемым элементом является кластер, который содержит не сколько секторов.

Таблица FAT16 может адресовать 2^{16} — 65 536 кластеров. Для дисков большой емкости размер кластера оказывается слишком большим, так как информационная емкость жестких дисков может достигать 150 Гбайт.

Например, для диска объемом 40 Гбайт размер кластера будет равен:

$$40 \text{ Гбайт} / 65536 = 655\,360 \text{ байт} = 640 \text{ Кбайт}.$$

Файлу всегда выделяется целое число кластеров. При размещении на жестком диске большого количества небольших по размеру файлов они будут занимать кластеры лишь частично, что приведет к большим потерям свободного дискового пространства.

Эта проблема частично решается с помощью использования таблицы FAT32, в которой объем кластера принят равным 8 секторам или 4 килобайтам для диска любого объема.

В целях более надежного сохранения информации о размещении файлов на диске хранятся две идентичные копии таблицы FAT.

Преобразование FAT16 в FAT32 можно осуществить с помощью служебной программы Преобразование диска в FAT32, которая входит в состав Windows.

Мы рассмотрели файловую систему, имеющую название FAT, однако в последнее время все большую популярность приобретает файловая система NTFS (New Technology File System - файловая система операционных систем

семейства Windows NT), которая, в частности, используется в Windows NT и Windows XP.

Максимальный размер раздела NTFS в данный момент ограничен лишь размерами «жестких» дисков. Как и любая другая система, NTFS делит все полезное место на кластеры — блоки данных, используемые одновременно. NTFS поддерживает почти любые размеры кластеров — от 512 байт до 64 Кбайт, неким стандартом же считается кластер размером 4 Кбайт. Диск NTFS условно делится на две части. Первые 12 % диска отводятся под так называемую зону MFT (Master File Table). Это БД, представляющая собой общую файловую таблицу, строки которой соответствуют файлам тома NTFS, а столбцы - атрибутам файлов. Запись каких-либо других данных в эту область невозможна. Остальные 88 % диска является обычным пространством для хранения файлов.

Зона MFT поделена на записи фиксированного размера (обычно 1 Кбайт), и каждая запись соответствует какому-либо файлу (в общем смысле этого слова). Первые 16 файлов несут служебный характер и недоступны ОС — они называются *метафайлами*, причем самый первый метафайл - сам MFT. Эти первые 16 элементов MFT - единственная часть диска, имеющая фиксированное положение. Интересно, что вторая копия первых трех записей для надежности — они очень важны - хранится ровно посередине диска. Остальной MFT-файл может располагаться, как и любой другой файл, в произвольных местах диска — восстановить его положение можно с помощью его самого, «зацепившись» за самую основу первый элемент MFT.

Дефрагментация дисков. Замедление скорости обмена данными может происходить в результате фрагментации файлов. Фрагментация файлов (фрагменты файлов хранятся в различных, удаленных друг от друга кластерах) возрастает с течением времени, в процессе удаления одних файлов и записи других.

Так как на диске могут храниться сотни и тысячи файлов в сотнях тысяч кластеров, то фрагментированность файлов будет существенно замедлять

доступ к ним (магнитным головкам придется постоянно перемещаться с дорожки на дорожку) и в конечном итоге приводить к преждевременному износу жесткого диска. Рекомендуется периодически проводить дефрагментацию диска, в процессе которой файлы записываются в кластеры, последовательно идущие друг за другом.

Учебно-методическое и информационное обеспечение

Основная литература:

- 1 Гаврилов, М. В. Информатика и информационные технологии [Электронный ресурс]: учебник для СПО / М. В. Гаврилов, В. А. Климов. — 4-е изд., перераб. и доп. — М. : Издательство Юрайт, 2017. — 383 с. — Режим доступа: <https://www.biblio-online.ru>.
- 2 Илюшечкин, В. М. Основы использования и проектирования баз данных [Электронный ресурс]: учебник для СПО / В. М. Илюшечкин. — испр. и доп. — М. : Издательство Юрайт, 2017. — 213 с. — Режим доступа: <https://www.biblio-online.ru>.
- 3 Казанский, А. А. Объектно-ориентированный анализ и программирование на visual basic 2013 [Электронный ресурс]: учебник для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 290 с. — Режим доступа: <https://www.biblio-online.ru>.
- 4 Казанский, А. А. Прикладное программирование на excel 2013 [Электронный ресурс]: учебное пособие для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 159 с. — Режим доступа: <https://www.biblio-online.ru>.
- 5 Казанский, А. А. Программирование на visual c# 2013 [Электронный ресурс]: учебное пособие для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 191 с. — Режим доступа: <https://www.biblio-online.ru>.
- 6 Семакин И.Г. Основы алгоритмизации и программирования: учеб. для

- сред. проф. образования/ И.Г.Семакин, А.П. Шестаков. – 3-е изд., стер. – М.: Академия, 2012. – 392 с.
- 7 Советов, Б. Я. Базы данных [Электронный ресурс]: учебник для СПО / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 2-е изд. — М. : Издательство Юрайт, 2017. — 463 с. — Режим доступа: <https://www.biblio-online.ru>.
 - 8 Советов, Б. Я. Информационные технологии [Электронный ресурс]: учебник для СПО / Б. Я. Советов, В. В. Цехановский. — 6-е изд., перераб. и доп. — М. : Издательство Юрайт, 2017. — 261 с. — Режим доступа: <https://www.biblio-online.ru>.
 - 9 Федорова Г. Н. Разработка и администрирование баз данных : учеб. для студ. учреждений сред. проф. образования / Г. Н. Федорова. - М.: Академия, 2015. - 313 с.
 - 10 Цветкова, М.С. Информатика и ИКТ : учеб. для студентов учреждений сред. проф. образования / М. С. Цветкова, Л. С. Великович. - 8-е изд., стер. - М. : Академия, 2016. - 332, [1] с., [4] л. ил.
 - 11 Черпаков, И. В. Основы программирования [Электронный ресурс]: учебник и практикум для СПО / И. В. Черпаков. — М. : Издательство Юрайт, 2017. — 219 с. — Режим доступа: <https://www.biblio-online.ru>.

Тема 1.2. Системы счисления

Основные понятия

Система счисления — это способ записи чисел с помощью заданного набора специальных знаков (цифр).

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

План лекции

1. Понятие системы счисления. История развития систем счисления. Позиционные и непозиционные системы счисления. Модель перевода чисел из одной системы счисления в другую.

2. Кодирование информации.

1. Понятие системы счисления. История развития систем счисления. Позиционные и непозиционные системы счисления. Модель перевода чисел из одной системы счисления в другую.

Система счисления — это способ записи чисел с помощью заданного набора специальных знаков (цифр).

Существуют позиционные и непозиционные системы счисления.

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

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

Сама же запись числа 757,7 означает сокращенную запись выражения

$$700 + 50 + 7 + 0,7 = 7 \cdot 10^2 + 5 \cdot 10^1 + 7 \cdot 10^0 + 7 \cdot 10^{-1} = 757,7.$$

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

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

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

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

где a_i – цифры системы счисления; n и m – число целых и дробных разрядов, соответственно.

Например:

Разряды	3	2	1	0	-1	
Число	1	0	1	1		$1_2 = 1*2^3 + 0*2^2 + 1*2^1 + 1*2^0 + 1*2^{-1};$

Разряды	2	1	0	-1	-2	
Число	2	7	6	5	2	$8 = 2*8^2 + 7*8^1 + 6*8^0 + 5*8^{-1} + 2*8^{-2};$

Целые числа в позиционных системах счисления

В каждой системе счисления цифры упорядочены в соответствии с их значениями: 1 больше 0, 2 больше 1 и т.д.

Продвижением цифры называют замену её следующей по величине.

Продвинуть цифру 1 значит заменить её на 2, продвинуть цифру 2 значит заменить её на 3 и т.д. **Продвижение старшей цифры** (например, цифры 9 в десятичной системе) **означает замену её на 0**. В двоичной системе, использующей только две цифры – 0 и 1, продвижение 0 означает замену его на 1, а продвижение 1 – замену её на 0.

Целые числа в любой системе счисления порождаются с помощью

Правила счета:

Для образования целого числа, следующего за любым данным целым числом, нужно *продвинуть* самую правую цифру числа; если какая-либо цифра после продвижения стала нулем, то нужно продвинуть цифру, стоящую слева от неё.

Применяя это правило, запишем первые десять целых чисел

- в двоичной системе: 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001;
- в троичной системе: 0, 1, 2, 10, 11, 12, 20, 21, 22, 100;
- в пятеричной системе: 0, 1, 2, 3, 4, 10, 11, 12, 13, 14;
- восьмеричной системе: 0, 1, 2, 3, 4, 5, 6, 7, 10, 11.

Системы счисления используются специалистами для общения с компьютером

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

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

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

10 - я	2 - я	8 - я	16 - я	10 - я	2 - я	8 - я	16 - я
0	0	0	0	10	1010	12	A
1	1	1	1	11	1011	13	B
2	10	2	2	12	1100	14	C
3	11	3	3	13	1101	15	D
4	100	4	4	14	1110	16	E
5	101	5	5	15	1111	17	F
6	110	6	6	16	10000	20	10
7	111	7	7	17	10001	21	11
8	1000	10	8	18	10010	22	12
9	1001	11	9	19	10011	23	13

Из всех систем счисления **особенно проста** и поэтому **интересна для технической реализации в компьютерах двоичная система счисления.**

Почему люди пользуются десятичной системой, а компьютеры — двоичной?

Люди предпочитают десятичную систему, вероятно, потому, что с древних времен считали по пальцам, а пальцев у людей по десять на руках и

ногах. Не всегда и не везде люди пользуются десятичной системой счисления. В Китае, например, долгое время пользовались пятеричной системой счисления.

А компьютеры используют двоичную систему потому, что она имеет ряд преимуществ перед другими системами:

- для ее реализации нужны **технические устройства с двумя устойчивыми состояниями** (есть ток — нет тока, намагничен — не намагничен и т.п.), а не, например, с десятью, — как в десятичной;
- представление информации посредством только двух состояний **надежно и помехоустойчиво**;
- возможно **применение аппарата булевой алгебры** для выполнения логических преобразований информации;
- двоичная арифметика намного проще десятичной.

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

Восьмеричная и шестнадцатеричная системы счисления

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

Перевод чисел из десятичной системы в двоичную и наоборот выполняет машина. Однако, чтобы профессионально использовать компьютер, следует научиться понимать слово машины. Для этого и разработаны восьмеричная и шестнадцатеричная системы.

Числа в этих системах читаются почти так же легко, как десятичные, требуют соответственно *в три* (восьмеричная) и *в четыре* (шестнадцатеричная) *раза меньше разрядов, чем в двоичной системе* (ведь числа 8 и 16 — соответственно, третья и четвертая степени числа 2).

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

Например:

$$537,1_8 = 101\ 011\ 111,001_2 ; 1A3,F_{16} = 1\ 1010\ 0011,1111_2$$

$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$
 5 3 7 1

$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$
 1 A 3 F

Чтобы перевести число из двоичной системы в восьмеричную или шестнадцатеричную, его нужно разбить влево и вправо от запятой на триады (для восьмеричной) или тетрады (для шестнадцатеричной) и каждую такую группу заменить соответствующей восьмеричной (шестнадцатеричной) цифрой.

Например,

$$10101001,10111_2 = 10\ 101\ 001,101\ 110_2 = 251,56_8$$

$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow$
 2 5 1 5 6

$$10101001,10111_2 = 1010\ 1001,1011\ 1000_2 = A9,B8_{16}$$

$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$
 A 9 B 8

Перевод целого числа из десятичной системы в любую другую позиционную систему счисления

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

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

в двоичную

75		2
1		37
		2
		1
		18
		2
		0
		9
		2
		1
		4
		2
		0
		2
		2
		0
		1

в восьмеричную

75		8
3		9
		8
		1
		1

в шестнадцатеричную

75		16
11		14

Ответ: $75_{10} = 1\ 001\ 011_2 = 113_8 = 4B_{16}$.

Перевод правильной десятичной дроби в любую другую позиционную систему счисления

При переводе правильной десятичной дроби в систему счисления с основанием q необходимо сначала саму дробь, а затем дробные части всех последующих произведений последовательно умножать на q , отделяя после каждого умножения целую часть произведения. Число в новой системе счисления записывается как последовательность полученных целых частей произведения.

Умножение производится до тех пор, пока дробная часть произведения не станет равной нулю. Это значит, что сделан точный перевод. В противном случае перевод осуществляется до заданной точности. Достаточно того количества цифр в результате, которое поместится в ячейку.

Пример: Перевести число 0,35 из десятичной системы в двоичную, восьмеричную и шестнадцатеричную:

в двоичную	в восьмеричную	в шестнадцатеричную
0, 35 2 0 70 2 1 40 2 0 80 2 1 60 2 1 20	0, 35 8 2 80 8 6 40 8 3 20	0, 35 16 5 60 16 9 60

Ответ: $0,35_{10} = 0,01011_2 = 0,263_8 = 0,59_{16}$.

Перевод числа из двоичной (восьмеричной, шестнадцатеричной) системы в десятичную

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

Примеры:

Разряды 3 2 1 0 -1
Число 1 0 1 1, $1_2 = 1 \cdot 2^3 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} = 11,5_{10}$.

Разряды 2 1 0 -1
Число 2 7 6, $5_8 = 2 \cdot 8^2 + 7 \cdot 8^1 + 6 \cdot 8^0 + 5 \cdot 8^{-1} = 190,625_{10}$.

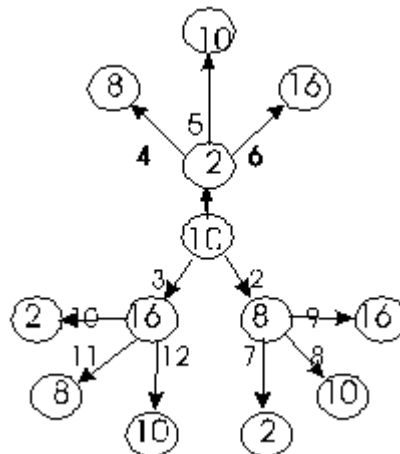
Разряды 2 1 0
Число $1 F 3_{16} = 1 \cdot 16^2 + 15 \cdot 16^1 + 3 \cdot 16^0 = 499_{10}$.

Сводная таблица переводов целых чисел из одной системы счисления в другую

Рассмотрим только те системы счисления, которые применяются в компьютерах — десятичную, двоичную, восьмеричную и шестнадцатеричную.

Для определенности возьмем произвольное десятичное число, например 46, и для него выполним все возможные последовательные переводы из одной системы счисления в другую.

Порядок переводов определим в соответствии с рисунком:



На этом рисунке использованы следующие обозначения:

- в кружках записаны основания систем счисления;
- стрелки указывают направление перевода;
- номер рядом со стрелкой означает порядковый номер

соответствующего примера в сводной таблице 1.

Например:  означает перевод из двоичной системы в шестнадцатеричную, имеющий в таблице порядковый номер 6.

Сводная таблица переводов целых чисел

Таблица 1.

№ п./п	Перевод	№ п./п	Перевод
1	$10 \rightarrow 2$ $\begin{array}{r} 46 \overline{) 23} \quad 2 \\ \underline{0} \quad 1 \quad 11 \quad 2 \\ \quad \underline{1} \quad 5 \quad 2 \\ \quad \quad \underline{1} \quad 2 \quad 2 \\ \quad \quad \quad \underline{0} \quad 1 \end{array}$ Ответ: 101110_2	5	$2 \rightarrow 10$ $\begin{array}{r} 5 \ 4 \ 3 \ 2 \ 1 \ 0 \\ 1 \ 0 \ 1 \ 1 \ 1 \ 0_2 = 2^5 + 2^3 + 2^2 + 2^1 = \\ = 46_{10} \end{array}$ Ответ: 46_{10}
		6	$2 \rightarrow 16$ $101110_2 = 10 \quad 1110_2 = 2E_{16}$ Ответ: $2E_{16}$
		7	$8 \rightarrow 2$ $56_8 = \underbrace{101}_5 \underbrace{110}_6_2$ Ответ: 101110_2
2	$10 \rightarrow 8$ $\begin{array}{r} 46 \overline{) 8} \\ \underline{6} \quad 5 \end{array}$ Ответ: 56_8	8	$8 \rightarrow 10$ $\begin{array}{r} 1 \ 0 \\ 5 \ 6_8 = 5 \cdot 8^1 + 6 \cdot 8^0 = 40 + 6 = \\ = 46_{10} \end{array}$ Ответ: 46_{10}
		9	$8 \rightarrow 16$ $5 \ 6_8 = \underbrace{101}_5 \underbrace{110}_6_2 = 10 \quad 1110_2 = 2E_{16}$ Ответ: $2E_{16}$
№ п./п	Перевод	№ п./п	Перевод
3	$10 \rightarrow 16$ $\begin{array}{r} 46 \overline{) 16} \\ \underline{14} \quad 2 \end{array}$ Ответ: $2E_{16}$	10	$16 \rightarrow 2$ $2E_{16} = 0010 \quad 1110_2 = 101110_2$ Ответ: 101110_2
		11	$16 \rightarrow 8$ $2E_{16} = 10 \quad 1110_2 = \underbrace{101}_5 \underbrace{110}_6_2 = 56_8$ Ответ: 56_8
4	$2 \rightarrow 8$ $101110_2 = \underbrace{101}_5 \underbrace{110}_6_2 = 56_8$ Ответ: 56_8	12	$16 \rightarrow 10$ $\begin{array}{r} 1 \ 0 \\ 2 \ E_{16} = 2 \cdot 16^1 + E \cdot 16^0 = \\ = 32 + 14 = 46_{10} \end{array}$ Ответ: 46_{10}

Арифметические операции в позиционных системах счисления

Рассмотрим основные арифметические операции: *сложение*, *вычитание*, *умножение* и *деление*. Правила выполнения этих операций в десятичной системе хорошо известны — это *сложение*, *вычитание*, *умножение столбиком* и *деление углом*. Эти правила применимы и ко всем другим позиционным

системам счисления. Только *таблицами сложения и умножения* надо пользоваться *особыми* для каждой системы.

Сложение

Таблицы сложения легко составить, используя Правило Счета.

Сложение в двоичной системе **Сложение в восьмеричной системе**

+	0 1
0	0 1
1	1 10

+	0 1 2 3 4 5 6 7
0	0 1 2 3 4 5 6 7
1	1 2 3 4 5 6 7 10
2	2 3 4 5 6 7 10 11
3	3 4 5 6 7 10 11 12
4	4 5 6 7 10 11 12 13
5	5 6 7 10 11 12 13 14
6	6 7 10 11 12 13 14 15
7	7 10 11 12 13 14 15 16

Сложение в шестнадцатиричной системе

+	0 1 2 3 4 5 6 7 8 9 A B C D E F
0	0 1 2 3 4 5 6 7 8 9 A B C D E F
1	1 2 3 4 5 6 7 8 9 A B C D E F 10
2	2 3 4 5 6 7 8 9 A B C D E F 10 11
3	3 4 5 6 7 8 9 A B C D E F 10 11 12
4	4 5 6 7 8 9 A B C D E F 10 11 12 13
5	5 6 7 8 9 A B C D E F 10 11 12 13 14
6	6 7 8 9 A B C D E F 10 11 12 13 14 15
7	7 8 9 A B C D E F 10 11 12 13 14 15 16
8	8 9 A B C D E F 10 11 12 13 14 15 16 17
9	9 A B C D E F 10 11 12 13 14 15 16 17 18
A	A B C D E F 10 11 12 13 14 15 16 17 18 19
B	B C D E F 10 11 12 13 14 15 16 17 18 19 1A
C	C D E F 10 11 12 13 14 15 16 17 18 19 1A 1B
D	D E F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C
E	E F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D
F	F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E

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

Пример 1. Сложим числа 15 и 6 в различных системах счисления.

Десятичная: $15_{10} + 6_{10}$

Двоичная: $1111_2 + 110_2$

Восьмеричная: $17_8 + 6_8$

$$\begin{array}{r}
 1 \\
 + 15 \\
 + 6 \\
 \hline
 21 \\
 \hline
 \begin{array}{l}
 5+6=11=10+1 \\
 1+1=2
 \end{array}
 \end{array}$$

$$\begin{array}{r}
 1\ 1\ 1 \\
 + 1\ 1\ 1\ 1 \\
 + 0\ 1\ 1\ 0 \\
 \hline
 1\ 0\ 1\ 0\ 1 \\
 \hline
 \begin{array}{l}
 1+0=1 \\
 1+1=2=2+0 \\
 1+1+1=3=2+1 \\
 1+1=2=2+0
 \end{array}
 \end{array}$$

$$\begin{array}{r}
 1 \\
 + 17 \\
 + 6 \\
 \hline
 25 \\
 \hline
 \begin{array}{l}
 7+6=13=8+5 \\
 1+1=2
 \end{array}
 \end{array}$$

Шестнадцатеричная: $F_{16} + 6_{16}$

Ответ: $15+6 = 21_{10} = 10101_2 =$

$25_8 = 15_{16}.$

$$\begin{array}{r}
 1 \\
 + F \\
 + 6 \\
 \hline
 15 \\
 \hline
 15+6=21=16+5
 \end{array}$$

Проверка.

Преобразуем

полученные суммы к десятичному виду:

$$10101_2 = 2^4 + 2^2 + 2^0 = 16+4+1=21,$$

$$25_8 = 2 \cdot 8^1 + 5 \cdot 8^0 = 16 + 5 = 21,$$

$$15_{16} = 1 \cdot 16^1 + 5 \cdot 16^0 = 16+5 = 21.$$

Пример 2. Сложим числа 15, 7 и 3.

Десятичная: $15_{10} + 7_{10} + 3_{10}$ **Двоичная:** $1111_2 + 111_2 + 11_2$ **Восьмеричная:** $17_8 + 7_8 + 3_8$

$$\begin{array}{r}
 1 \\
 + 15 \\
 + 7 \\
 + 3 \\
 \hline
 25 \\
 \hline
 \begin{array}{l}
 5+7+3=15=10+5 \\
 1+1=2
 \end{array}
 \end{array}$$

$$\begin{array}{r}
 1\ 1+1\ 1 \\
 + 1\ 1\ 1\ 1 \\
 + 1\ 1\ 1 \\
 + 1\ 1 \\
 \hline
 1\ 1\ 0\ 0\ 1 \\
 \hline
 \begin{array}{l}
 1+1+1=3=2+1 \\
 1+1+1+1=4=2 \cdot 2+0 \\
 1+1=2=2+0 \\
 1+1+1=3=2+1
 \end{array}
 \end{array}$$

$$\begin{array}{r}
 2 \\
 + 17 \\
 + 7 \\
 + 3 \\
 \hline
 31 \\
 \hline
 \begin{array}{l}
 7+7+3=17=2 \cdot 8+1 \\
 2+1=3
 \end{array}
 \end{array}$$

Шестнадцатеричная:

Ответ: $5+7+3 = 25_{10} = 11001_2 =$

$F_{16} + 7_{16} + 3_{16}$

$31_8 = 19_{16}.$

$$\begin{array}{r}
 + F \\
 + 7 \\
 + 3 \\
 \hline
 19 \\
 \hline
 15+7+3=25=16+9
 \end{array}$$

Проверка:

$$11001_2 = 2^4 + 2^3 + 2^0 = 16+8+1=25,$$

$$31_8 = 3 \cdot 8^1 + 1 \cdot 8^0 = 24 + 1 = 25,$$

$$19_{16} = 1 \cdot 16^1 + 9 \cdot 16^0 = 16+9 = 25.$$

Пример 3. Сложим числа 141,5 и 59,75.

Десятичная: $141,5_{10} + 59,75_{10}$ **Двоичная:** $10001101,1_2 + 111011,11_2$

$$\begin{array}{r}
 111 \\
 + 141,50 \\
 \hline
 59,75 \\
 \hline
 201,25
 \end{array}$$

$0+5=5$
 $5+7=12=10+2$
 $1+9+1=11=10+1$
 $4+5+1=10=10+0$
 $1+1=2$

$$\begin{array}{r}
 11111111 \\
 + 10001101,1 \\
 \hline
 111011,11 \\
 \hline
 11001001,01
 \end{array}$$

$1+0=1$
 $1+1=2=2+0$
 $1+1=2=2+0$
 $1+1+1=3=2+1$
 $1+1=2=2+0$
 $1+1=2=2+0$
 $1+1+1=3=2+1$
 $1+1=2=2+0$

Восьмеричная: $215,4_8 + 73,6_8$

Шестнадцатеричная: $8D,8_{16} + 3B,C_{16}$

$$\begin{array}{r}
 111 \\
 + 215,4 \\
 \hline
 73,6 \\
 \hline
 311,2
 \end{array}$$

$4+6=10=8+2$
 $5+3+1=9=8+1$
 $1+7+1=9=8+1$
 $2+1=3$

$$\begin{array}{r}
 11 \\
 + 8D,8 \\
 \hline
 3B,C \\
 \hline
 C9,4
 \end{array}$$

$8+12=20=16+4$
 $13+11+1=25=16+9$
 $8+3+1=12=C_{16}$

Ответ: $141,5 + 59,75 = 201,25_{10} = 11001001,01_2 = 311,2_8 = C9,4_{16}$

Проверка. Преобразуем полученные суммы к десятичному виду:

$$11001001,01_2 = 2^7 + 2^6 + 2^3 + 2^0 + 2^{-2} = 201,25$$

$$311,2_8 = 3 \cdot 8^2 + 1 \cdot 8^1 + 1 \cdot 8^0 + 2 \cdot 8^{-1} = 201,25$$

$$C9,4_{16} = 12 \cdot 16^1 + 9 \cdot 16^0 + 4 \cdot 16^{-1} = 201,25$$

Вычитание

Пример 4. Вычтем единицу из чисел 10_2 , 10_8 и 10_{16}

Двоичная: $10_2 - 1_2$ Восьмеричная: $10_8 - 1_8$ Шестнадцатеричная: $10_{16} - 1_{16}$

$$\begin{array}{r}
 1 \\
 - 10 \\
 \hline
 1 \\
 \hline
 1
 \end{array}$$

$2-1=1$

$$\begin{array}{r}
 1 \\
 - 10 \\
 \hline
 1 \\
 \hline
 7
 \end{array}$$

$8-1=7$

$$\begin{array}{r}
 1 \\
 - 10 \\
 \hline
 1 \\
 \hline
 F
 \end{array}$$

Заемы
 $16-1=15=F_{16}$

Пример 5. Вычтем единицу из чисел 100_2 , 100_8 и 100_{16} .

$$\begin{array}{r}
 1 \\
 - 100 \\
 \hline
 1 \\
 \hline
 11
 \end{array}$$

$2-1=1$
 $1-0=1$

$$\begin{array}{r}
 1 \\
 - 100 \\
 \hline
 1 \\
 \hline
 77
 \end{array}$$

$8-1=7$
 $7-0=7$

$$\begin{array}{r}
 1 \\
 - 100 \\
 \hline
 1 \\
 \hline
 FF
 \end{array}$$

Заемы
 $16-1=15=F_{16}$
 $1+1=2$

Двоичная: $100_2 - 1_2$ Восьмеричная: $100_8 - 1_8$ Шестнадцатеричная: $100_{16} - 1_{16}$

Пример 6. Вычтем число $59,75$ из числа $201,25$.

Десятичная: $201,25_{10} - 59,75_{10}$ Двоичная: $11001001,01_2 - 111011,11_2$

$$\begin{array}{r}
 1 \quad 1 \\
 - \quad 201,25 \\
 \quad 59,75 \\
 \hline
 141,50 \\
 \begin{array}{l}
 5-5=0 \\
 10+2-7-5 \\
 10-9=1 \\
 9-5=4 \\
 2-1=1
 \end{array}
 \end{array}$$

Восьмеричная: $311,2_8 - 73,6_8$

$$\begin{array}{r}
 1 \quad 1 \quad 1 \\
 - \quad 311,2 \\
 \quad 73,6 \\
 \hline
 215,4 \\
 \begin{array}{l}
 8+2-6=4 \\
 8-3=5 \\
 8-7=1
 \end{array}
 \end{array}$$

$$\begin{array}{r}
 1 \quad 1 \quad 1 \quad \text{Заемы} \\
 - \quad 11001001,01 \\
 \quad 00111011,11 \\
 \hline
 10001101,10 \\
 \begin{array}{l}
 1-0=1 \\
 0-0=0 \\
 1-1=0 \\
 1-1=0 \\
 2-1=1 \\
 1-0=1
 \end{array}
 \end{array}$$

Шестнадцатеричная: $C9,4_{16} - 3B,C_{16}$

$$\begin{array}{r}
 1 \quad 1 \\
 - \quad C9,4 \\
 \quad 3B,C \\
 \hline
 8D,8 \\
 \begin{array}{l}
 16+4-12=8 \\
 16+8-11=13=D_{16} \\
 12-1-3=8
 \end{array}
 \end{array}$$

Ответ: $201,25_{10} - 59,75_{10} = 141,5_{10} = 10001101,1_2 = 215,4_8 = 8D,8_{16}$.

Проверка. Преобразуем полученные разности к десятичному виду:

$$10001101,1_2 = 2^7 + 2^3 + 2^2 + 2^0 + 2^{-1} = 141,5;$$

$$215,4_8 = 2 \cdot 8^2 + 1 \cdot 8^1 + 5 \cdot 8^0 + 4 \cdot 8^{-1} = 141,5;$$

$$8D,8_{16} = 8 \cdot 16^1 + D \cdot 16^0 + 8 \cdot 16^{-1} = 141,5.$$

Умножение

Выполняя умножение многозначных чисел в различных позиционных системах счисления, можно использовать обычный алгоритм перемножения чисел в столбик, но при этом результаты перемножения и сложения однозначных чисел необходимо заимствовать из соответствующих рассматриваемой системе таблиц умножения и сложения.

Умножение в двоичной системе **Умножение в восьмеричной системе**

*	0	1
0	0	0
1	0	1

*	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	10	12	14	16
3	0	3	6	11	14	17	22	25
4	0	4	10	14	20	24	30	34
5	0	5	12	17	24	31	36	43
6	0	6	14	22	30	36	44	52
7	0	7	16	25	34	43	52	61

Ввиду чрезвычайной простоты таблицы умножения в двоичной системе, умножение сводится лишь к сдвигам множимого и сложениям.

Пример 7. Перемножим числа 5 и 6.

Десятичная: $5_{10} \cdot 6_{10}$ Двоичная: $101_2 \cdot 110_2$ Восьмеричная: $5_8 \cdot 6_8$

$$\begin{array}{r} \times 5 \\ 6 \\ \hline 30 \end{array}$$

$$\begin{array}{r} \times 101 \\ 110 \\ \hline 101 \\ 101 \\ \hline 11110 \end{array}$$

$$\begin{array}{r} \times 5 \\ 6 \\ \hline 36 \end{array}$$

Ответ: $5 \cdot 6 = 30_{10} = 11110_2 = 36_8$.

Проверка. Преобразуем полученные произведения к десятичному виду:

$$11110_2 = 2^4 + 2^3 + 2^2 + 2^1 = 30;$$

$$36_8 = 3 \cdot 8^1 + 6 \cdot 8^0 = 30.$$

Пример 8. Перемножим числа 115 и 51.

Десятичная: $115_{10} \cdot 51_{10}$ Двоичная: $1110011_2 \cdot 110011_2$ Восьмеричная: $163_8 \cdot 63_8$

$$\begin{array}{r} \times 115 \\ 51 \\ \hline 115 \\ 575 \\ \hline 5865 \end{array}$$

$$\begin{array}{r} \times 1110011 \\ 110011 \\ \hline 1110011 \\ 1110011 \\ \hline 1110011 \\ 1110011 \\ \hline 1011011101001 \end{array}$$

$$\begin{array}{r} \times 163 \\ 63 \\ \hline 531 \\ 1262 \\ \hline 13351 \end{array}$$

Ответ: $115 \cdot 51 = 5865_{10} = 1011011101001_2 = 13351_8$.

Проверка. Преобразуем полученные произведения к десятичному виду:

$$1011011101001_2 = 2^{12} + 2^{10} + 2^9 + 2^7 + 2^6 + 2^5 + 2^3 + 2^0 = 5865;$$

$$13351_8 = 1 \cdot 8^4 + 3 \cdot 8^3 + 3 \cdot 8^2 + 5 \cdot 8^1 + 1 \cdot 8^0 = 5865.$$

Деление

Деление в любой позиционной системе счисления производится по тем же правилам, как и деление углом в десятичной системе. В двоичной системе деление выполняется особенно просто, ведь очередная цифра частного может быть только нулем или единицей.

Пример 9. Разделим число 30 на число 6.

Десятичная: $30_{10} : 6_{10}$ Двоичная: $11110_2 : 110_2$ Восьмеричная: $36_8 : 6_8$

$$\begin{array}{r} 30 \overline{) 6} \\ 30 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 11110 \overline{) 110} \\ 110 \\ \hline 110 \\ 110 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 36 \overline{) 6} \\ 36 \\ \hline 0 \end{array}$$

Ответ: $30 : 6 = 5_{10} = 101_2 = 5_8$.

Пример 10. Разделим число 5865 на число 115.

Десятичная: $5865_{10} : 115_{10}$ **Двоичная:** $1011011101001_2 : 1110011_2$

$$\begin{array}{r} - \quad 5865 \mid 115 \\ \quad 575 \quad \mid 51 \\ \hline \quad 115 \\ - \quad 115 \\ \hline \quad 0 \end{array}$$

$$\begin{array}{r} - \quad 1011011101001 \mid 1110011 \\ \quad 1110011 \quad \mid 110011 \\ \hline \quad 1000100 \\ - \quad 1110011 \\ \hline \quad 10101100 \\ - \quad 1110011 \\ \hline \quad 1110011 \\ - \quad 1110011 \\ \hline \quad 0 \end{array}$$

Восьмеричная: $13351_8 : 163_8$

$$\begin{array}{r} - \quad 13351 \mid 163 \\ \quad 1262 \quad \mid 63 \\ \hline \quad 531 \\ - \quad 531 \\ \hline \quad 0 \end{array}$$

Ответ: $5865 : 115 = 51_{10} = 110011_2 = 63_8$.

Проверка. Преобразуем полученные частные к десятичному виду:

$$110011_2 = 2^5 + 2^4 + 2^1 + 2^0 = 51; \quad 63_8 = 6 \cdot 8^1 + 3 \cdot 8^0 = 51.$$

Пример 11. Разделим число 35 на число 14.

$$\begin{array}{r} - \quad 35 \mid 14 \\ \quad 28 \quad \mid 2,5 \\ \hline \quad 70 \\ - \quad 70 \\ \hline \quad 0 \end{array}$$

$$\begin{array}{r} - \quad 100011 \mid 1110 \\ \quad 1110 \quad \mid 10,1 \\ \hline \quad 1110 \\ - \quad 1110 \\ \hline \quad 0 \end{array}$$

Десятичная: $35_{10} : 14_{10}$

Двоичная: $100011_2 : 1110_2$

Восьмеричная: $43_8 : 16_8$

$$\begin{array}{r} - \quad 43 \mid 16 \\ \quad 34 \quad \mid 2,4 \\ \hline \quad 70 \\ - \quad 70 \\ \hline \quad 0 \end{array}$$

Ответ: $35 : 14 = 2,5_{10} = 10,1_2 = 2,4_8$.

Проверка. Преобразуем полученные частные к десятичному виду:

$$10,1_2 = 2^1 + 2^{-1} = 2,5;$$

$$2,4_8 = 2 \cdot 8^0 + 4 \cdot 8^{-1} = 2,5.$$

2. Кодирование информации

Любой текст состоит из последовательности символов. Символами могут быть буквы, цифры, знаки препинания, знаки математических действий, круглые и квадратные скобки и т.д. Особо обратим внимание на символ "пробел", который используется для разделения слов и предложений между собой. Хотя на бумаге или экране дисплея "пробел" - это пустое, свободное место, этот символ ничем не "хуже" любого другого символа. На клавиатуре компьютера или пишущей машинки символу "пробел" соответствует специальная клавиша.

Текстовая информация, как и любая другая, хранится в памяти компьютера в двоичном виде. Для этого каждому символу ставится в соответствие некоторое неотрицательное число, называемое кодом символа, и это число записывается в память ЭВМ в двоичном виде. Конкретное соответствие между символами и их кодами называется системой кодировки.

В современных ЭВМ, в зависимости от типа операционной системы и конкретных прикладных программ, используются 8-разрядные и 16-разрядные коды символов. Использование 8-разрядных кодов позволяет закодировать 256 различных знаков, этого вполне достаточно для представления многих символов, используемых на практике. При такой кодировке для кода символа достаточно выделить в памяти один байт. Так и делают: каждый символ представляют своим кодом, который записывают в один байт памяти.

В персональных компьютерах обычно используется система кодировки **ASCII (American Standard Code for Information Interchange** - американский стандартный код для обмена информацией). Он введен в 1963 г. и ставит в соответствие каждому символу семиразрядный двоичный код. Легко определить, что в коде ASCII можно представить 128 символов.

В системе ASCII закреплены две таблицы кодирования **базовая** и **расширенная**. Базовая таблица закрепляет значения кодов от 0 до 127, а расширенная относится к символам с номерами от 128 до 255.

Первые 32 кода базовой таблицы, начиная с нулевого, отданы производителям аппаратных средств. В этой области размещаются управляющие коды, которым не соответствуют ни какие символы языков. Начиная с 32 по 127 код размещены коды символов английского алфавита, знаков препинания, арифметических действий и некоторых вспомогательных символов.

Кодировка символов русского языка, известная как **кодировка Windows-1251**, была введена "извне" - компанией Microsoft, но, учитывая широкое распространение операционных систем и других продуктов этой компании в России, она глубоко закрепились и нашла широкое распространение.

Другая распространённая кодировка носит название **КОИ-8** (код обмена информацией, восьмизначный) - её происхождение относится к временам действия Совета Экономической Взаимопомощи государств Восточной Европы. Сегодня кодировка КОИ - 8 имеет широкое распространение в компьютерных сетях на территории России и в российском секторе Интернета.

Международный стандарт, в котором предусмотрена кодировка символов русского языка, носит название **ISO (International Standard Organization - Международный институт стандартизации)**. На практике данная кодировка используется редко.

Универсальная система кодирования текстовых данных.

Если проанализировать организационные трудности, связанные с созданием единой системы кодирования текстовых данных, то можно прийти к выводу, что они вызваны ограниченным набором кодов (256). В то же время, очевидно, что если, кодировать символы не восьмизначными двоичными числами, а числами с большим разрядом то и диапазон возможных значений кодов станет на много больше. Такая система, основанная на 16-разрядном кодировании символов, получила название **универсальной - UNICODE**. Шестнадцать разрядов позволяют обеспечить уникальные коды для 65 536 различных символов - этого поля вполне достаточно для размещения в одной таблице символов большинства языков планеты.

Несмотря на тривиальную очевидность такого подхода, простой механический переход на данную систему долгое время сдерживался из-за недостатков ресурсов средств вычислительной техники (в системе кодирования UNICODE все текстовые документы становятся автоматически вдвое длиннее). Во второй половине 90-х годов технические средства достигли необходимого уровня обеспечения ресурсами, и сегодня мы наблюдаем постепенный перевод документов и программных средств на универсальную систему кодирования. Ниже приведены таблицы кодировки ASCII.

Базовая таблица кодировки ASCII											
32 пробел	48 0	64 @	80 P	96 `	112 p						
33 !	49 1	65 A	81 Q	97 a	113 q						
34 "	50 2	66 B	82 R	98 b	114 r						
35 #	51 3	67 C	83 S	99 c	115 s						
36 \$	52 4	68 D	84 T	100 d	116 t						
37 %	53 5	69 E	85 U	101 e	117 u						
38 &	54 6	70 F	86 V	102 f	118 v						
39 '	55 7	71 G	87 W	103 g	119 w						
40 (	56 8	72 H	88 X	104 h	120 x						
41)	57 9	73 I	89 Y	105 i	121 y						
42 *	58 :	74 J	90 Z	106 j	122 z						
43 +	59 ;	75 K	91 [	107 k	123 {						
44 ,	60 <	76 L	92 \	108 l	124						
45 -	61 =	77 M	93]	109 m	125 }						
46 .	62 >	78 N	94 ^	110 n	126 ~						
47 /	63 ?	79 O	95 _	111 o	127						

Кодировка Windows 1251							
128 Ъ	144 ђ	160 Ѣ	176 •	192 А	208 Р	224 а	240 р
129 Ѓ	145 'ѣ	161 ѣ	177 ±	193 Б	209 С	225 б	241 с
130 ,	146 'ѥ	162 ѥ	178	194 В	210 Т	226 в	242 т
131 'ѧ	147 'Ѧ	163 Ј	179	195 Г	211 У	227 г	243 у
132 .	148 "ѧ	164 Ѧ	180 'ѧ	196 Д	212 Ф	228 д	244 ф
133 ...	149 'ѧ	165 Г	181 'ѧ	197 Е	213 Х	229 е	245 х
134 †	150 -ѧ	166 'ѧ	182 'ѧ	198 Ж	214 Ц	230 ж	246 ц
135 ‡	151 -ѧ	167 'ѧ	183 'ѧ	199 З	215 Ч	231 з	247 ч
136 'ѧ	152 'ѧ	168 Е	184 'ѧ	200 И	216 Ш	232 и	248 ш
137 ‰	153 'ѧ	169 'ѧ	185 Na	201 Ё	217 Щ	233 й	249 щ
138 Љ	154 ъ	170 Ё	186 е	202 К	218 Ъ	234 к	250 ъ
139 'ѧ	155 'ѧ	171 •	187 •	203 Л	219 Ы	235 л	251 ы
140 Њ	156 ъ	172 -ѧ	188	204 М	220 Ь	236 м	252 ь
141 К	157 к	173 -ѧ	189 S	205 Н	221 Э	237 н	253 э
142 Ћ	158 ѣ	174 @	190 s	206 О	222 Ю	238 о	254 ю
143 Ќ	159 џ	175 i	191 i	207 П	223 Я	239 п	255 я

Для представления текстовой информации (прописные и строчные буквы русского и латинского алфавитов, цифры, знаки и математические символы) достаточно 256 различных знаков. По формуле можно вычислить, какое количество информации необходимо, чтобы закодировать каждый знак:

$$N = 2^i \Rightarrow 256 = 2^i \Rightarrow 2^8 = 2^i \Rightarrow i = 8 \text{ битов.}$$

Для обработки текстовой информации на компьютере необходимо представить ее в двоичной знаковой системе. Для кодирования каждого знака требуется количество информации, равное 8 битам, т. е. длина двоичного кода знака составляет восемь двоичных знаков. Каждому знаку необходимо поставить в соответствие уникальный двоичный код из интервала от 00000000 до 11111111 (в десятичном коде от 0 до 255) (табл. 2).

Таблица 2. Кодировки знаков

Двоичный код	Десятичный код	КОИ-8	Windows	MS-DOS	Mac	ISO
00000000	0					
...						
00001000	8	удаление последнего символа (клавиша {Backspace})				
...						
00001101	13	перевод строки (клавиша {Enter})				
...						
00100000	32	клавиша {Пробел}				
00100001	33	!				
...						
01011010	90	Z				
...						
01111111	127	␣				
10000000	128	—	Ъ	А	А	к
...						
11000010	194	б	В	—	—	т
...						
11001100	204	л	М			ь
...						
11011101	221	щ	Э	_	Ё	н
...						
11111111	255	ь	я	нераз. пробел	нераз. пробел	п

Различные кодировки знаков. Присваивание знаку конкретного двоичного кода - это вопрос соглашения, которое фиксируется в кодовой таблице. В существующих кодовых таблицах первые 33 кода (десятичные коды с 0 по 32) соответствуют не знакам, а операциям (перевод строки, ввод пробела и т. д.).

Десятичные коды с 33 по 127 являются интернациональными и соответствуют знакам латинского алфавита, цифрам, знакам арифметических операций и знакам препинания.

Десятичные коды с 128 по 255 являются национальными, т. е. в различных национальных кодировках одному и тому же коду соответствуют разные знаки. К сожалению, в настоящее время существуют пять различных кодовых таблиц для русских букв (Windows, MS-DOS, КОИ-8, Mac, ISO (табл. 2 и 3), поэтому тексты, созданные в одной кодировке, не будут правильно отображаться в другой.

Таблица 3. Десятичные коды некоторых символов в различных кодировках

Символ	Windows	MS-DOS	КОИ-8	Mac	ISO	Unicode
А	192	128	225	128	176	1040
В	194	130	247	130	178	1042
М	204	140	237	140	188	1052
Э	221	157	252	157	205	1069
я	255	239	241	223	239	1103

Например, в кодировке *Windows* последовательность числовых кодов 221, 194, 204 образует слово "ЭВМ", тогда как в других кодировках это будет бессмысленный набор символов.

К счастью, в большинстве случаев пользователь не должен заботиться о перекодировках текстовых документов, так как это делают специальные программы-конверторы, встроенные в операционную систему и приложения.

В последние годы широкое распространение получил новый международный стандарт кодирования текстовых символов Unicode, который отводит на каждый символ 2 байта (16 битов). По формуле можно определить количество символов, которые можно закодировать согласно этому стандарту:

$$N = 2^i = 2^{16} = 65\,536.$$

Такого количества символов оказалось достаточно, чтобы закодировать не только русский и латинский алфавиты, цифры, знаки и математические символы, но и греческий, арабский, иврит и другие алфавиты.

Учебно-методическое и информационное обеспечение

Основная литература:

- 1 Гаврилов, М. В. Информатика и информационные технологии [Электронный ресурс]: учебник для СПО / М. В. Гаврилов, В. А. Климов. — 4-е изд., перераб. и доп. — М. : Издательство Юрайт, 2017. — 383 с. — Режим доступа: <https://www.biblio-online.ru>.
- 2 Илюшечкин, В. М. Основы использования и проектирования баз данных [Электронный ресурс]: учебник для СПО / В. М. Илюшечкин. — испр. и доп. — М. : Издательство Юрайт, 2017. — 213 с. — Режим доступа: <https://www.biblio-online.ru>.
- 3 Казанский, А. А. Объектно-ориентированный анализ и программирование на visual basic 2013 [Электронный ресурс]: учебник для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 290 с. — Режим доступа: <https://www.biblio-online.ru>.
- 4 Казанский, А. А. Прикладное программирование на excel 2013 [Электронный ресурс]: учебное пособие для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 159 с. — Режим доступа: <https://www.biblio-online.ru>.
- 5 Казанский, А. А. Программирование на visual c# 2013 [Электронный ресурс]: учебное пособие для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 191 с. — Режим доступа: <https://www.biblio-online.ru>.
- 6 Семакин И.Г. Основы алгоритмизации и программирования: учеб. для сред. проф.образования/ И.Г.Семакин, А.П. Шестаков. – 3-е изд., стер. – М.: Академия, 2012. – 392 с.

- 7 Советов, Б. Я. Базы данных [Электронный ресурс]: учебник для СПО / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 2-е изд. — М. : Издательство Юрайт, 2017. — 463 с. — Режим доступа: <https://www.biblio-online.ru>.
- 8 Советов, Б. Я. Информационные технологии [Электронный ресурс]: учебник для СПО / Б. Я. Советов, В. В. Цехановский. — 6-е изд., перераб. и доп. — М. : Издательство Юрайт, 2017. — 261 с. — Режим доступа: <https://www.biblio-online.ru>.
- 9 Федорова Г. Н. Разработка и администрирование баз данных : учеб. для студ. учреждений сред. проф. образования / Г. Н. Федорова. - М.: Академия, 2015. - 313 с.
- 10 Цветкова, М.С. Информатика и ИКТ : учеб. для студентов учреждений сред. проф. образования / М. С. Цветкова, Л. С. Великович. - 8-е изд., стер. - М. : Академия, 2016. - 332, [1] с., [4] л. ил.
- 11 Черпаков, И. В. Основы программирования [Электронный ресурс]: учебник и практикум для СПО / И. В. Черпаков. — М. : Издательство Юрайт, 2017. — 219 с. — Режим доступа: <https://www.biblio-online.ru>.

Тема 1.3. Основы алгоритмизации

План лекции

- 1 Понятие алгоритма. Свойства алгоритмов. Способы задания алгоритмов.
- 2 Виды алгоритмических конструкций. Системы и технологии программирования.
- 3 Введение в язык программирования. Синтаксис программы. Семантика программы.

Понятие алгоритма. Свойства алгоритмов. Способы задания алгоритмов.

Алгоритм - это последовательность действий, приводящих к решению поставленной задачи.

Алгоритм применительно к ПК - точное предписание, т.е. набор операций и правил их чередования, при помощи которого, начиная с некоторых исходных данных, можно решить задачу фиксированного типа.

(Algorithmi (Аль-Хорезми) - среднеазиатский математик XI в., впервые описавший правила выполнения четырёх арифметических действий).

Примеры выполнения определённого алгоритма: рецепт приготовления вкусного блюда, инструкция по сборке мебели, описание химического опыта, инструкции и описания из журналов по рукоделью, вязке и шитью, руководства по игровым и полезным компьютерным программам.

Исполнитель алгоритма - человек или устройство (в частности, процессор ЭВМ), умеющий выполнять определённый набор действий. Такой набор действий - система команд исполнителя.

Алгоритмы в зависимости от цели, начальных условий задачи, путей ее решения и определения действий исполнителя подразделяются на:

- механические алгоритмы, или детерминированные, жесткие (например, алгоритм работы машины, двигателя и т. п.);
- гибкие алгоритмы, например стохастические, т. е. вероятностные и эвристические.

Механический алгоритм задает определенные действия, обозначая их в единственной и достоверной последовательности, обеспечивая тем самым однозначный требуемый или искомый результат.

Вероятностный (стохастический) алгоритм дает программу решения задачи несколькими путями или способами, приводящими к вероятному достижению результата.

Эвристический алгоритм (от греческого слова "эврика" "Я нашел") -это такой алгоритм, в котором достижение конечного результата программы действий однозначно не предопределено, так же как не обозначена вся последовательность действий, не выявлены все действия исполнителя.

Свойства алгоритма:

понятность для исполнителя – содержание предписания о выполнении только таких действий, которые входят в систему команд исполнителя;

дискретность (прерывность) – выполнение команд алгоритма последовательно, с точной фиксацией моментов окончания выполнения одной команды и начала выполнения следующей;

определённость - каждое правило алгоритма должно быть чётким и однозначным;

результативность (конечность) - алгоритм должен приводить к решению за конечное число шагов;

массовость (универсальность) - алгоритм должен выполняться для любого набора исходных данных, удовлетворяющих условию задачи.

Способы задания алгоритмов:

- **словесный;**
- **графический;**
- **табличный;**
- **программный.**

Словесный способ записи задается в произвольном изложении на естественном языке.

Графический способ записи - в виде блок-схем. Блок-схема - последовательность блоков, соединенных линиями передачи (ветвями).

Табличный способ записи - в виде таблицы, устанавливающей зависимость результата от исходных данных.

Программный способ записи - в виде текста на каком-либо языке программирования.

Виды алгоритмических конструкций. Системы и технологии программирования.

Элементарные шаги алгоритма можно объединить в следующие алгоритмические конструкции: линейные (последовательные), разветвляющиеся, циклические с предусловием и циклические с постусловием.

Любой алгоритм можно составить, используя эти четыре алгоритмические конструкции.

Линейной называют алгоритмическую конструкцию, реализованную в виде последовательности действий (шагов), в которой каждое действие (шаг) алгоритма выполняется ровно один раз, причем после каждого i -го действия (шага) выполняется $(i+1)$ -е действие (шаг), если i -е действие — не конец алгоритма.

Разветвляющейся (или ветвящейся) называется алгоритмическая конструкция, обеспечивающая выбор между двумя альтернативами в зависимости от значения входных данных. При каждом конкретном наборе входных данных разветвляющийся алгоритм сводится к линейному. Различают неполное (если – то) и полное (если – то – иначе) ветвления. Полное ветвление позволяет организовать две ветви в алгоритме (то или иначе), каждая из которых ведет к общей точке их слияния, так что выполнение алгоритма продолжается независимо от того, какой путь был выбран (рис. 1). Неполное ветвление предполагает наличие некоторых действий алгоритма только на одной ветви (то), вторая ветвь отсутствует, т.е. для одного из результатов проверки никаких действий выполнять не надо, управление сразу переходит к точке слияния.

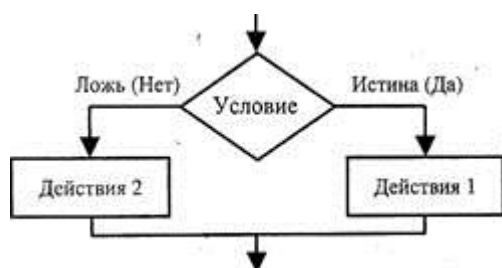


Рис. 1. Полное ветвление

Циклической (или циклом) называют алгоритмическую конструкцию, в которой некая, идущая подряд группа действий (шагов) алгоритма может выполняться несколько раз, в зависимости от входных данных или условия задачи. Группа повторяющихся действий на каждом шагу цикла называется

телом цикла. Любая циклическая конструкция содержит в себе элементы ветвящейся алгоритмической конструкции.

Цикл с предусловием

В данной циклической структуре сначала проверяется значение условного выражения (условие) перед выполнением очередного шага цикла. Если значение условного выражения истинно, выполняется тело цикла. После чего управление вновь передается проверке условия и т.д. Эти действия повторяются до тех пор, пока условное выражение не примет значение ЛОЖЬ. При первом же несоблюдении условия цикл завершается. Количество шагов цикла заранее не определено и зависит от входных данных задачи. Особенностью цикла с предусловием является то, что если изначально условное выражение ложно, то тело цикла не выполнится ни разу.

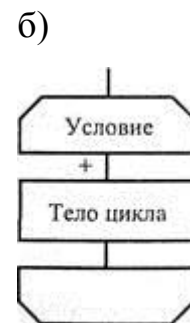
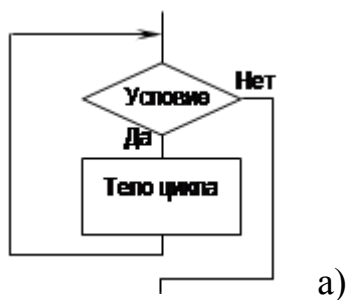


Рис. 2. Блок-схема цикла с предусловием: два варианта изображения с помощью условного блока а) и с помощью блока границы цикла б)

Цикл с постусловием

Как и в цикле с предусловием, в циклической конструкции с постусловием заранее не определено число повторений тела цикла, оно зависит от входных данных задачи. В отличие от цикла с предусловием, тело цикла с постусловием всегда будет выполнено хотя бы один раз, после чего проверяется условие. В этой конструкции тело цикла будет выполняться до тех пор, пока значение условного выражения ложно (условие "окончания" цикла). Как только оно становится истинным, выполнение команды прекращается. Возможно построение цикла и с условием "продолжения" цикла, т.е. тело цикла

будет выполняться до тех пор, пока значение условия истинно. Блок-схема данной конструкции представлена на рис. двумя способами: с помощью условного блока а и с помощью блока управления б.

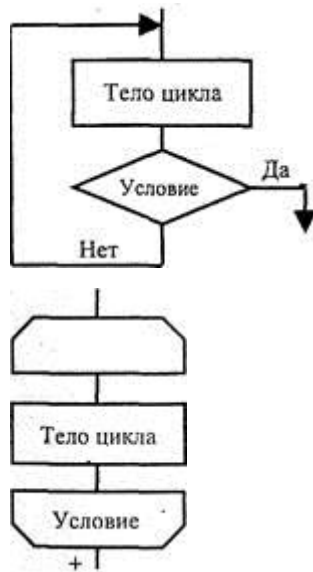


Рис. Блок-схема цикла с постусловием

Рассмотрим три типа циклических алгоритмов: цикл с параметром (который называют арифметическим циклом), цикл с предусловием и цикл с постусловием (их называют итерационными).

Существует разновидность цикла с предусловием, называемая арифметический цикл. В арифметическом цикле число его шагов (повторений) однозначно определяется правилом изменения параметра, которое задается с помощью начального (N) и конечного (K) значений параметра и шагом (h) его изменения. Т.е., на первом шаге цикла значение параметра равно N , на втором – $N + h$, на третьем – $N + 2h$ и т.д. На последнем шаге цикла значение параметра не больше K , но такое, что дальнейшее его изменение приведет к значению, большему, чем K .

Например, вывести 10 раз слово «Привет!». Его блок-схема использует специальный блок начала арифметического цикла с указанием, что переменная i в нем будет изменяться от 1 до 10 с шагом 1.

Введение в язык программирования. Синтаксис программы.

Семантика программы

Стандартизация языков программирования

Язык программирования может быть представлен в виде набора спецификаций, определяющих его синтаксис и семантику.

Для многих широко распространённых языков программирования созданы международные стандарты. Специальные организации проводят регулярное обновление и публикацию спецификаций и формальных определений соответствующего языка. В рамках таких комитетов продолжается разработка и модернизация языков программирования и решаются вопросы о расширении или поддержке уже существующих и новых языковых конструкций.

Типы данных

Современные цифровые компьютеры обычно являются двоичными и данные хранят в двоичном (бинарном) коде (хотя возможны реализации и в других системах счисления). Эти данные как правило отражают информацию из реального мира (имена, банковские счета, измерения и др.), представляющую высокоуровневые концепции.

Особая система, по которой данные организуются в программе, — это *система типов* языка программирования; разработка и изучение систем типов известна под названием теория типов. Языки могут быть классифицированы как системы с *статической типизацией* и языки с *динамической типизацией*.

Статически-типизированные языки могут быть в дальнейшем подразделены на языки с *обязательной декларацией*, где каждая переменная и объявление функции имеет обязательное объявление типа, и языки с *выводимыми типами*. Иногда динамически-типизированные языки называются *латентно-типизированными*.

Структуры данных

Системы типов в языках высокого уровня позволяют определять сложные, составные типы, так называемые структуры данных. Как правило, структурные типы данных образуются как декартово произведение базовых (атомарных) типов и ранее определённых составных типов.

Основные структуры данных (списки, очереди, хеш-таблицы, двоичные деревья и пары) часто представлены особыми синтаксическими конструкциями в языках высокого уровня. Такие данные структурируются автоматически.

Семантика языков программирования

Существует несколько подходов к определению семантики языков программирования.

Наиболее широко распространены разновидности следующих трёх: операционного, денотационного (математического) и деривационного (аксиоматического).

При описании семантики в рамках операционного подхода обычно исполнение конструкций языка программирования интерпретируется с помощью некоторой воображаемой (абстрактной) ЭВМ.

Деривационная семантика описывает последствия выполнения конструкций языка с помощью языка логики и задания пред- и постусловий. Денотационная семантика оперирует понятиями, типичными для математики — множества, соответствия, а также суждения, утверждения и др.

Язык программирования строится в соответствии с той или иной базовой моделью вычислений парадигмой программирования.

Несмотря на то, что большинство языков ориентировано на императивную модель вычислений, задаваемую фоннеймановской архитектурой ЭВМ, существуют и другие подходы. Можно упомянуть языки со стековой вычислительной моделью (Forth, Factor, Postscript и др.), а также функциональное (Лисп, Haskell, ML и др.) и логическое программирование (Пролог) и язык Рефал, основанный на модели вычислений, введённой советским математиком А.А. Марковым-младшим.

В настоящее время также активно развиваются проблемно-ориентированные, декларативные и визуальные языки программирования.

Компилируемые и интерпретируемые языки

Языки программирования могут быть разделены на компилируемые и интерпретируемые.

Программа на компилируемом языке при помощи специальной программы компилятора преобразуется (компилируется) в набор инструкций для данного типа процессора (машинный код) и далее записывается в исполнимый модуль, который может быть запущен на выполнение как отдельная программа. Другими словами, компилятор переводит исходный текст программы с языка программирования высокого уровня в двоичные коды инструкций процессора.

Если программа написана на интерпретируемом языке, то интерпретатор непосредственно выполняет (интерпретирует) исходный текст без предварительного перевода. При этом программа остаётся на исходном языке и не может быть запущена без интерпретатора. Можно сказать, что процессор компьютера — это интерпретатор машинного кода.

Кратко говоря, компилятор переводит исходный текст программы на машинный язык сразу и целиком, создавая при этом отдельную исполняемую программу, а интерпретатор выполняет исходный текст прямо во время исполнения программы.

Разделение на компилируемые и интерпретируемые языки является несколько условным. Так, для любого традиционно компилируемого языка, как, например, Паскаль, можно написать интерпретатор. Кроме того, большинство современных «чистых» интерпретаторов не исполняют конструкции языка непосредственно, а компилируют их в некоторое высокоуровневое промежуточное представление (например, с разыменованием переменных и раскрытием макросов).

Для любого интерпретируемого языка можно создать компилятор — например, язык Лисп, изначально интерпретируемый, может компилироваться без каких бы то ни было ограничений. Создаваемый во время исполнения программы код может так же динамически компилироваться во время исполнения.

Как правило, скомпилированные программы выполняются быстрее и не требуют для выполнения дополнительных программ, так как уже переведены

на машинный язык. Вместе с тем, при каждом изменении текста программы требуется её перекомпиляция, что создаёт трудности при разработке. Кроме того, скомпилированная программа может выполняться только на том же типе компьютеров и, как правило, под той же операционной системой, на которую был рассчитан компилятор. Чтобы создать исполняемый файл для машины другого типа, требуется новая компиляция.

Интерпретируемые языки обладают некоторыми специфическими дополнительными возможностями (см. выше), кроме того, программы на них можно запускать сразу же после изменения, что облегчает разработку. Программа на интерпретируемом языке может быть зачастую запущена на разных типах машин и операционных систем без дополнительных усилий.

Однако интерпретируемые программы выполняются заметно медленнее, чем компилируемые, кроме того, они не могут выполняться без дополнительной программы-интерпретатора.

Некоторые языки, например, Java и C#, находятся между компилируемыми и интерпретируемыми. А именно, программа компилируется не в машинный язык, а в машинно-независимый код низкого уровня, байт-код. Далее байт-код выполняется виртуальной машиной. Для выполнения байт-кода обычно используется интерпретация, хотя отдельные его части для ускорения работы программы могут быть транслированы в машинный код непосредственно во время выполнения программы по технологии компиляции «на лету» (Just-in-time compilation, JIT). Для Java байт-код исполняется виртуальной машиной Java (Java Virtual Machine, JVM), для C# — Common Language Runtime.

Подобный подход в некотором смысле позволяет использовать плюсы как интерпретаторов, так и компиляторов. Следует упомянуть также оригинальный язык Форт (Forth) имеющий и интерпретатор и компилятор.

Современные языки программирования рассчитаны на использование ASCII, то есть доступность всех *графических* символов ASCII является необходимым и достаточным условием для записи любых конструкций языка. *Управляющие* символы ASCII используются ограниченно: допускаются только

возврат каретки CR, перевод строки LF и горизонтальная табуляция HT (иногда также вертикальная табуляция VT и переход к следующей странице FF).

Подробнее по этой теме см.: Переносимый набор символов.

Ранние языки, возникшие в эпоху 6-битных символов, использовали более ограниченный набор. Например, алфавит Фортрана включает 49 символов (включая пробел): A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 = + - * / () . , \$ ' :

Заметным исключением является язык APL, в котором используется очень много специальных символов.

Использование символов за пределами ASCII (например, символов KOI8-R или символов Юникода) зависит от реализации: иногда они разрешаются только в комментариях и символьных/строковых константах, а иногда и в идентификаторах. В СССР существовали языки, где все ключевые слова писались русскими буквами, но большую популярность подобные языки не завоевали (исключение составляет Встроенный язык программирования 1С: Предприятие).

Подробнее по этой теме см.: Русские языки программирования.

Расширение набора используемых символов сдерживается тем, что многие проекты по разработке программного обеспечения являются международными. Очень сложно было бы работать с кодом, где имена одних переменных записаны русскими буквами, других — арабскими, а третьих — китайскими иероглифами. Вместе с тем, для работы с текстовыми данными языки программирования нового поколения (Delphi 2006, C#, Java) поддерживают Unicode.

Процедурные языки программирования

Процедурное (императивное) программирование является отражением архитектуры традиционных ЭВМ, которая была предложена фон Нейманом в 1940-х годах. Теоретической моделью процедурного программирования служит алгоритмическая система под названием Машина Тьюринга.

Программа на процедурном языке программирования состоит из последовательности операторов (инструкций), задающих процедуру решения задачи. Основным является оператор присваивания, служащий для изменения содержимого областей памяти. Концепция памяти как хранилища значений, содержимое которого может обновляться операторами программы, является фундаментальной в императивном программировании.

Выполнение программы сводится к последовательному выполнению операторов с целью преобразования исходного состояния памяти, то есть значений исходных данных, в заключительное, то есть в результаты. Таким образом, с точки зрения программиста имеются программа и память, причем первая последовательно обновляет содержимое последней.

Процедурный язык программирования предоставляет возможность программисту определять каждый шаг в процессе решения задачи. Особенность таких языков программирования состоит в том, что задачи разбиваются на шаги и решаются шаг за шагом. Используя процедурный язык, программист определяет языковые конструкции для выполнения последовательности алгоритмических шагов.

Basic

Бейсик (от BASIC, сокращение от англ. *Beginner's All-purpose Symbolic Instruction Code* — универсальный код символических инструкций для начинающих; англ. *basic* — основной, базовый) — семейство высокоуровневых языков программирования.

Был разработан в 1963 году профессорами Дартмутского колледжа Томасом Куртом (*Thomas E. Kurtz*) и Джоном Кемени (*John G. Kemeny*).

Язык предназначался для обучения программированию и получил широкое распространение в виде различных диалектов, прежде всего, как язык для домашних микрокомпьютеров.

Предпосылки

До середины 1960-х компьютеры (ЭВМ) были слишком дорогими машинами, использовавшимися в основном для научно-технических задач, и

выполнявшими задачи не в интерактивном режиме (т.н. *пакетная обработка*), когда общение с машиной осуществлялось через её оператора.

Языки программирования этой поры, как и компьютеры, на которых они использовались, были разработаны для работы с ними обученных технических специалистов и программистов. Поскольку машины были дорогими, то и машинное время было дорого — поэтому на первом месте стояла скорость выполнения программы (скорость счёта).

Однако, в течение 1960-х цены на компьютеры стали падать так, что даже небольшие компании могли их себе позволить; их быстродействие всё увеличивалась и наступило время, когда стало возможно организовать более эффективную непосредственную работу пользователей с компьютерами в так называемом режиме разделения времени (*time-sharing*).

Рождение

Бейсик был придуман в 1963 году преподавателями Дартмутского Колледжа Джоном Кемени и Томасом Куртцом, и под их руководством был реализован командой студентов колледжа. Со временем, когда стали появляться другие диалекты, этот «изначальный» диалект стали называть Dartmouth BASIC.

Бейсик был спроектирован так, чтобы студенты могли без затруднений писать программы, используя терминалы с разделением времени. Он предназначался для более «простых» пользователей, не столько заинтересованных в скорости исполнения программ, сколько просто в возможности использовать компьютер для решения своих задач не имея специальной подготовки.

При проектировании языка использовались следующие восемь принципов: новый язык должен

- быть простым в использовании *для начинающих*
- быть языком программирования *общего назначения*
- предоставлять возможность *расширения функциональности*,

доступную опытным программистам

- быть *интерактивным*
- предоставлять ясные сообщения об ошибках
- быстро работать на небольших программах
- не требовать понимания работы аппаратного обеспечения
- защищать пользователя от операционной системы

Язык был основан частично на Фортран Пи частично наАлгол-60, с добавлениями, делающими его удобным для работы в режиме разделения времени и, позднее, обработки текста и матричной арифметики. Первоначально Бейсик был реализован на мейнфрейме GE-265 с поддержкой множества терминалов. Вопреки распространённому убеждению, в момент своего появления это былкомпилируемыйязык.

Взрывной рост

Несмотря на то что язык уже использовался на нескольких миникомпьютерах, его настоящее распространение началось с его появления на микрокомпьютере Altair 8800. Многие языки программирования были слишком большими чтобы поместиться в небольшую память, которую пользователи таких машин могли себе позволить. Для машин с таким медленным носителем как бумажная лента (позднее — аудиокассета) и без подходящего текстового редактора такой небольшой язык как Бейсик был отличной находкой. В1975 годуМайкрософт(тогда это были лишь двое —Билл ГейтсиПол Аллен, при участииМонте Давидова) выпустилаAltair BASIC. Затем его версии появились на другой платформе под лицензией и скоро в использовании были уже миллионы копий и вариантов; один из вариантов,Applesoft BASIC, стал стандартным языком наApple II. Для операционной системыCP/Mбыл создан диалектBASIC-80, надолго определивший развитие языка.

В 1979 годуМайкрософтобсуждала с несколькими поставщиками компьютеров (включаяIBM) лицензирование интерпретатора Бейсик на их машинах. Одна из версий (ROM BASIC) была включена вПЗУIBM PC— компьютер мог автоматически загружаться в Бейсик. Так как IBM не придавала тогда большого значения персональным компьютерам (основным полем её

деятельности былимейнфреймы), то, вопреки своим принципам, она разрешила Microsoft продавать интерпретатор отдельно. Это сделало последнюю пионером в выпуске ПО нового поколения — не привязанного к конкретной аппаратуре и поставляемого отдельно от компьютера.

Синтаксис языка напоминает Fortran, и многие элементы — явные заимствования из него. Язык задумывался для обучения, поэтому его конструкции максимально просты. Как и в других языках программирования, ключевые слова взяты из английского языка. Основных типов данных два: строки и числа. С появлением версии Visual Basic, а также различных его модификаций (таких как VBA), в языке появились многие другие типы данных и дополнения, типичные для современных языков программирования (например, такие как объектная модель). Объявление переменных не требует специальной секции (в отличие, например, от Паскаля). Объявление переменной — это первое её использование.

Ранние версии бейсика (такие как GWBASIC, QBasic) существенно отличаются от современных диалектов и на сегодня практически не используются.

Pascal

Паскаль (англ. *Pascal*) — высокоуровневый язык программирования общего назначения. Один из наиболее известных языков программирования, широко применяется в промышленном программировании, обучении программированию в высшей школе, является базой для большого числа других языков. Был создан Никлаусом Виртом в 1968/9 годах (опубликован в 1970-м) после его участия в работе комитета разработки стандарта языка Алгол-68.

Паскаль был создан как язык для обучения процедурному программированию (хотя, по словам Вирта, язык нельзя считать только учебным, поскольку язык, непригодный для написания реальных программ, для обучения использоваться не должен). Название языку дано в честь

выдающегося французского математика, физика, литератора и философа Блеза Паскаля.

Паскаль был создан Никлаусом Виртом в 1968/9 годах после его участия в работе комитета разработки стандарта языка Алгол-68. Он был опубликован в 1970 году Виртом как небольшой и эффективный язык, чтобы способствовать хорошему стилю программирования, использовать структурное программирование и структурированные данные.

Компилятор Паскаля был написан на самом Паскале, используя «метод раскрутки», когда создается ядро языка, с постепенным наращиванием новых возможностей.

Особенностями языка являются строгая типизация и наличие средств структурного (процедурного) программирования. Паскаль был одним из первых таких языков. По мнению Н. Вирта, язык должен способствовать дисциплинированию программирования, поэтому, наряду со строгой типизацией, в Паскале сведены к минимуму возможные синтаксические неоднозначности, а сам синтаксис автор постарался сделать интуитивно понятным даже при первом знакомстве с языком.

Тем не менее, первоначально язык имел ряд ограничений: невозможность передачи функциям массивов переменной длины, отсутствие нормальных средств работы с динамической памятью, ограниченная библиотека ввода-вывода, отсутствие средств для подключения функций написанных на других языках, отсутствие средств отдельной компиляции и т.п. Подробный разбор недостатков языка Паскаль того времени был выполнен Брайаном Керниганом в статье «Почему Паскаль не является моим любимым языком программирования» (интересно, что эта статья вышла в начале 1980-х, когда уже существовал язык Модула-2, потомок Паскаля, избавленный от большинства его пороков, а также более развитые диалекты Паскаля). Некоторые недостатки Паскаля были исправлены в ISO-стандарте 1982 года, в частности, в языке появились открытые массивы, давшие возможность

использовать одни и те же процедуры для обработки одномерных массивов различных размеров.

Необходимо заметить, что многие недостатки языка не проявляются или даже становятся достоинствами при обучении программированию. Кроме того, по сравнению с основным языком программирования в академической среде 70-х (которым был Фортран, обладавший гораздо более существенными недостатками), Паскаль представлял собой значительный шаг вперед. В начале 1980-х годов в СССР для обучения школьников основам информатики и вычислительной техники академик А.П. Ершов разработал алголо-паскалеподобный «алгоритмический язык».

Наиболее известной реализацией Паскаля, обеспечившая широкое распространение и развитие языка является Turbo Pascal фирмы Borland, выросшая затем в объектный Паскаль для DOS (начиная с версии 5.5) и Windows и далее в Delphi, в которой использовались значительные расширения языка.

Диалекты Паскаля, применяемые в Turbo Pascal для DOS и Delphi для Windows, из-за отсутствия других успешных коммерческих реализаций стали популярны.

Объектный Паскаль от Borland и в отсутствие отраслевой стандартизации стал сильно отличаться от классического Паскаля.

Язык программирования — формальная знаковая система, предназначенная для записи программ, задающих алгоритмы в форме, понятной для исполнителя (например, компьютера). Язык программирования определяет набор лексических, синтаксических и семантических правил, используемых при составлении компьютерной программы. Он позволяет программисту точно определить то, на какие события будет реагировать компьютер, как будут храниться и передаваться данные, а также какие именно действия следует выполнять над этими данными при различных обстоятельствах. Со времени создания первых программируемых машин человечество придумало уже более восьми с половиной тысяч языков программирования. Каждый год их число

пополняется новыми. Некоторыми языками умеет пользоваться только небольшое число их собственных разработчиков, другие становятся известны миллионам людей. Профессиональные программисты иногда применяют в своей работе более десятка разнообразных языков программирования.

Си

Си (англ. C) — стандартизированный процедурный язык программирования, разработанный в начале 1970-х годов сотрудниками Bell Labs Кеном Томпсоном и Денисом Ритчи как развитие языка Би. Си был создан для использования в операционной системе UNIX. С тех пор он был импортирован на многие другие операционные системы и стал одним из самых используемых языков программирования. Си ценят за его эффективность; он является самым популярным языком для создания системного программного обеспечения. Его также часто используют для создания прикладных программ. Несмотря на то, что Си не разрабатывался для новичков, он активно используется для обучения программированию. В дальнейшем синтаксис языка Си стал основой для многих других языков (см.: Си-подобный синтаксис).

Для языка Си характерны лаконичность, современный набор конструкций управления потоком выполнения, структур данных и обширный набор операций.

Язык программирования Си отличается минимализмом. Авторы языка хотели, чтобы программы на нём легко компилировались с помощью однопроходного компилятора, после компиляции каждой элементарной составляющей программы соответствовало весьма небольшое число машинных команд, а использование базовых элементов языка не задействовало библиотеку во времени выполнения. Однопроходный компилятор компилирует программу, не возвращаясь назад, к уже откомпилированному тексту. Поэтому использованию функции должно предшествовать её объявление. Код на Си можно легко писать на низком уровне абстракции, почти как на ассемблере. Иногда Си называют «универсальным ассемблером» или «ассемблером высокого уровня», что отражает различие языков ассемблера для

разных платформ и единство стандарта Си, код которого может быть скомпилирован без изменений практически на любой модели компьютера. Си часто называют языком *среднего уровня* или даже *низкого уровня*, учитывая то, как близко он работает к реальным устройствам.

Компиляторы Си разрабатываются сравнительно легко благодаря относительно низкому уровню языка и скромному набору элементов. Поэтому данный язык доступен на самых различных платформах (возможно, круг этих платформ шире, чем у любого другого существующего языка). К тому же, несмотря на свою низкоуровневую природу, язык позволяет создавать переносимые программы и поддерживает программиста в этом. Программы, соответствующие стандарту языка, могут компилироваться на самых различных компьютерах.

Си (как и ОС UNIX, с которой он долгое время был связан) создавался программистами и для программистов, круг которых был бы ненамного шире круга разработчиков языка. Несмотря на это, область использования языка значительно шире задач системного программирования.

Си создавался с одной важной целью: сделать более простым написание больших программ с минимумом ошибок по правилам процедурного программирования, не добавляя лишних накладных расходов на итоговый код программы компилятором, как это всегда делают языки очень высокого уровня, такие как Бейсик. С этой стороны Си имеет следующие важные особенности:

- простую языковую базу, из которой вынесены в библиотеки многие существенные возможности, вроде математических функций или функций управления файлами;
- ориентацию на процедурное программирование, обеспечивающую удобство применения структурного стиля программирования;
- систему типов, предохраняющую от бессмысленных операций;
- использование препроцессора для, например, определения макросов и включения файлов исходным кодом;

- непосредственный доступ к памяти компьютера через использование указателей;
- минимальное число ключевых слов;
- передачу параметров в функцию по значению, а не по ссылке (при этом передача по ссылке выполняется с помощью указателей);
- указатели на функции и статические переменные
- области действия имён;
- записи— определяемые пользователем собирательные типы данных (структуры), которыми можно манипулировать как одним целым;

Вот некоторые особенности других языков программирования, которых не имеет Си:

- автоматическое управление памятью;
- поддержка объектно-ориентированного программирования (при этом первые версии C++ генерировали код программы на языке Си);
- вложенные функции (существуют компиляторы языка Си реализующие эту функцию, например компилятор GNU);
- полиморфизм функций и операторов;
- встроенная поддержка многозадачности и сети
- функции высшего порядка
- карринг.

Несмотря на то, что в Си нет столь многого, а это было важно в начале, язык был хорошо принят, потому что он позволял быстро создавать компиляторы для новых платформ, а также позволял программистам довольно точно представлять, как выполняются их программы. Благодаря этому программы, написанные на Си, эффективнее написанных на многих других языках. Как правило, лишь оптимизированный вручную код на ассемблере может работать ещё быстрее, потому что он даёт полный контроль над машиной, однако развитие современных компиляторов вместе с усложнением современных процессоров быстро сократило этот разрыв.

Одним из последствий высокой эффективности и переносимости Си стало то, что многие компиляторы, интерпретаторы и библиотеки других языков высокого уровня часто выполнены на языке Си.

Объектно-ориентированные языки программирования

Объектно-ориентированный язык программирования (ОО-язык) — язык, построенный на принципах объектно-ориентированного программирования.

В основе концепции объектно-ориентированного программирования лежит понятие *объекта* - некоей субстанции, которая объединяет в себе поля (данные) и методы (выполняемые объектом действия).

Например, объект "человек" может иметь поля "имя", "фамилия" и иметь методы "есть" и "спать". Соответственно, мы можем использовать в программе операторы Человек. Имя: ="Иван" и Человек. Есть(пища).

Си++

Компилируемый статически типизированный язык программирования общего назначения. Поддерживает разные парадигмы программирования, но, в сравнении с его предшественником — языком Си, — наибольшее внимание уделено поддержке объектно-ориентированного и обобщённого программирования.^[1]

Название «Си++» происходит от Си, в котором унарный оператор ++ обозначает инкремент переменной.

В 1990-х годах язык стал одним из наиболее широко применяемых языков программирования общего назначения.

При создании Си++ стремились сохранить совместимость с языком Си. Большинство программ на Си будут исправно работать и с компилятором Си++. Си++ имеет синтаксис, основанный на синтаксисе Си.

Язык возник в начале 1980-х годов, когда сотрудник фирмы Bell Laboratories Бьёрн Страуструп придумал ряд усовершенствований к языку Си под собственные нужды. До начала официальной стандартизации язык развивался в основном силами Страуструпа в ответ на запросы программистского сообщества. В 1998 году был ратифицирован международный

стандарт языка Си++: ISO/IEC 14882:1998 «Standard for the C++ Programming Language»; после принятия технических исправлений к стандарту в 2003 году — нынешняя версия этого стандарта — ISO/IEC 14882:2003. Ранние версии языка, известные под именем «С с классами», начали появляться с 1980 года. Идея создания нового языка берёт начало от опыта программирования Страуструпа для диссертации. Он обнаружил, что язык моделирования Симула (Simula) имеет такие возможности, которые были бы очень полезны для разработки большого программного обеспечения, но работает слишком медленно. В то же время язык BCPL достаточно быстр, но слишком близок к языкам низкого уровня и не подходит для разработки большого программного обеспечения. Страуструп начал работать в Bell Labs над задачами теории очередей (в приложении к моделированию телефонных вызовов). Попытки применения существующих в то время языков моделирования оказались неэффективными. Вспоминая опыт своей диссертации, Страуструп решил дополнить язык Си (преемник BCPL) возможностями, имеющимися в языке Симула. Язык Си, будучи базовым языком системы UNIX, на которой работали компьютеры Bell, является быстрым, многофункциональным и переносимым. Страуструп добавил к нему возможность работы с классами и объектами. В результате, практические задачи моделирования оказались доступными для решения как с точки зрения времени разработки (благодаря использованию Симула-подобных классов) так и с точки зрения времени вычислений (благодаря быстрдействию Си). В начале в Си были добавлены классы (с инкапсуляцией), производные классы, строгая проверка типов, inline-функции и аргументы по умолчанию. Разрабатывая *Си с классами* (позднее Си++), Страуструп также написал программу cfront — транслятор, перерабатывающий исходный код *Си с классами* в исходный код простого Си. Новый язык, неожиданно для автора, приобрёл большую популярность среди коллег и вскоре Страуструп уже не мог лично поддерживать его, отвечая на тысячи вопросов. В 1983 году произошло переименование языка из *Си с классами* в Си++. Кроме того, в него были добавлены новые возможности, такие как виртуальные функции, перегрузка

функций и операторов, ссылки, константы, пользовательский контроль над управлением свободной памятью, улучшенная проверка типов и новый стиль комментариев (`//`). Его первый коммерческий выпуск состоялся в октябре 1985 года. В 1985 году вышло также первое издание «Языка программирования Си++», обеспечивающее первое описание этого языка, что было чрезвычайно важно из-за отсутствия официального стандарта. В 1989 году состоялся выход Си++ версии 2.0. Его новые возможности включали множественное наследование, абстрактные классы, статические функции-члены, функции-константы и защищённые члены.

В 1990 году вышло «Комментированное справочное руководство по С++», положенное впоследствии в основу стандарта. Последние обновления включали шаблоны, исключения, пространства имён, новые способы приведения типов и булевский тип.

Стандартная библиотека Си++ также развивалась вместе с ним. Первым добавлением к стандартной библиотеке Си++ стали потоки ввода/вывода, обеспечивающие средства для замены традиционных функций Си `printf` и `scanf`. Позднее самым значительным развитием стандартной библиотеки стало включение в неё Стандартной библиотеки шаблонов.

После многих лет работы совместный комитет ANSI-ISO стандартизировал Си++ в 1998 году (ISO/IEC 14882:1998 — Язык программирования Си++). В течение нескольких лет после официального выхода стандарта комитет обрабатывал сообщения об ошибках и в итоге выпустил исправленную версию стандарта Си++ в 2003 году. В настоящее время рабочая группа MOC (ISO) работает над новой версией стандарта под кодовым названием C++09 (ранее известный как C++0X), который должен выйти в 2009 году.

Никто не обладает правами на язык Си++, он является свободным. Однако сам документ стандарта языка (за исключением черновиков) не доступен бесплатно. Название «Си++» было придумано Риком Масситти (Rick

Mascitti) и впервые было использовано в декабре 1983 года. Ранее, на этапе разработки, новый язык назывался «Си с классами».

Имя, получившееся в итоге, происходит от оператора Си «++» (увеличение значения переменной на единицу). Имя «C+» не было использовано потому, что является синтаксической ошибкой в Си и, кроме того, это имя было занято другим языком. Язык также не назван «D», поскольку «является расширением Си и не пытается устранять проблемы путём удаления элементов Си». Стандарт Си++ на 1998 год состоит из двух основных частей: ядра языка и стандартной библиотеки. Стандартная библиотека Си++ вобрала в себя разрабатывавшуюся одновременно со стандартом библиотеку шаблонов STL. Сейчас название STL официально не употребляется, однако в кругах программистов на Си++ это название используется для обозначения части стандартной библиотеки, содержащей определения шаблонов контейнеров, итераторов, алгоритмов и функторов. Стандарт Си++ содержит нормативную ссылку на стандарт Си от 1990 года и не определяет самостоятельно те функции стандартной библиотеки, которые заимствуются из стандартной библиотеки Си.

Кроме того, существует огромное количество библиотек Си++, не входящих в стандарт. В программах на Си++ можно использовать многие библиотеки Си.

Стандартизация определила язык программирования Си++, однако за этим названием могут скрываться также неполные, ограниченные, достандартные варианты языка. В первое время язык развивался вне формальных рамок, спонтанно, по мере ставившихся перед ним задач. Развитию языка сопутствовало развитие кросс-компилятора `g++`. Новшества в языке отражались в изменении номера версии кросс-компилятора. Эти номера версий кросс-компилятора распространялись и на сам язык, но применительно к настоящему времени речь о версиях языка Си++ не ведут.

JAVA

Java — объектно-ориентированный язык программирования, разрабатываемый компанией Sun Microsystems. Приложения Java обычно компилируются в специальный байт-код, поэтому они могут работать на любой виртуальной Java-машине (JVM) независимо от компьютерной архитектуры. Дата официального выпуска — 23 мая 1995 года.

В отношении произношения в русском языке, как и в ряде других, образовались две различные нормы — заимствованная англоязычная /'dʒɑ:və/ («джава») и традиционно-национальная «ява», соответствующая традиционному произношению названия острова Ява. Компания Sun придерживается англоязычного произношения во всех странах мира. Иногда в обиходе используют также жаргонное слово «Жаба» (например, изображение жабы есть на календариках группы российских пользователей Java (Java Users Group)).

Java — так называют не только сам язык, но и платформу для создания и исполнения приложений на основе данного языка.

Изначально язык назывался Oak («дуб») и разрабатывался Джеймсом Гослингом для программирования бытовых электронных устройств. Впоследствии он был переименован в Java и стал использоваться для написания клиентских приложений и серверного программного обеспечения. Назван в честь марки кофе Java, любимого некоторыми программистами, поэтому на официальной эмблеме языка изображена чашка с парящим кофе.

Программы на Java транслируются в байт-код, выполняемый виртуальной машиной Java (JVM) — программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор, но с тем отличием, что байтовый код, в отличие от текста, обрабатывается значительно быстрее.

Достоинство подобного способа выполнения программ — в полной независимости байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Другой важной особенностью технологии Java является гибкая система безопасности благодаря

тому, что исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером) вызывают немедленное прерывание.

Часто к недостаткам концепции виртуальной машины относят то, что исполнение байт-кода виртуальной машиной может снижать производительность программ и алгоритмов, реализованных на языке Java. Данное утверждение было справедливо для первых версий виртуальной машины Java, однако в последнее время оно практически потеряло актуальность. Этому способствовал ряд усовершенствований:

- применение технологии трансляции байт-кода в машинный код непосредственно во время работы программы (JIT-технология) с возможностью сохранения версий класса в машинном коде,
- широкое использование платформенно-ориентированного кода (native-код) в стандартных библиотеках,
- аппаратные средства, обеспечивающие ускоренную обработку байт-кода (например, технология Jazelle, поддерживаемая некоторыми процессорами фирмы ARM).

Идеи, заложенные в концепцию и различные реализации среды виртуальной машины Java, вдохновили множество энтузиастов на расширение перечня языков, которые могли бы быть использованы для создания программ, исполняемых на виртуальной машине.^[13] Эти идеи нашли также выражение в спецификации общезыковой инфраструктуры CLI, заложенной в основу платформы .NET компанией Microsoft.

Учебно-методическое и информационное обеспечение

Основная литература:

- 1 Гаврилов, М. В. Информатика и информационные технологии [Электронный ресурс]: учебник для СПО / М. В. Гаврилов, В. А. Климов.

- 4-е изд., перераб. и доп. — М. : Издательство Юрайт, 2017. — 383 с. — Режим доступа: <https://www.biblio-online.ru>.
- 2 Илюшечкин, В. М. Основы использования и проектирования баз данных [Электронный ресурс]: учебник для СПО / В. М. Илюшечкин. — испр. и доп. — М. : Издательство Юрайт, 2017. — 213 с. — Режим доступа: <https://www.biblio-online.ru>.
 - 3 Казанский, А. А. Объектно-ориентированный анализ и программирование на visual basic 2013 [Электронный ресурс]: учебник для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 290 с. — Режим доступа: <https://www.biblio-online.ru>.
 - 4 Казанский, А. А. Прикладное программирование на excel 2013 [Электронный ресурс]: учебное пособие для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 159 с. — Режим доступа: <https://www.biblio-online.ru>.
 - 5 Казанский, А. А. Программирование на visual c# 2013 [Электронный ресурс]: учебное пособие для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 191 с. — Режим доступа: <https://www.biblio-online.ru>.
 - 6 Семакин И.Г. Основы алгоритмизации и программирования: учеб. для сред. проф.образования/ И.Г.Семакин, А.П. Шестаков. – 3-е изд., стер. – М.: Академия, 2012. – 392 с.
 - 7 Советов, Б. Я. Базы данных [Электронный ресурс]: учебник для СПО / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 2-е изд. — М. : Издательство Юрайт, 2017. — 463 с. — Режим доступа: <https://www.biblio-online.ru>.
 - 8 Советов, Б. Я. Информационные технологии [Электронный ресурс]: учебник для СПО / Б. Я. Советов, В. В. Цехановский. — 6-е изд., перераб. и доп. — М. : Издательство Юрайт, 2017. — 261 с. — Режим доступа: <https://www.biblio-online.ru>.

- 9 Федорова Г. Н. Разработка и администрирование баз данных : учеб. для студ. учреждений сред. проф. образования / Г. Н. Федорова. - М.: Академия, 2015. - 313 с.
- 10 Цветкова, М.С. Информатика и ИКТ : учеб. для студентов учреждений сред. проф. образования / М. С. Цветкова, Л. С. Великович. - 8-е изд., стер. - М. : Академия, 2016. - 332, [1] с., [4] л. ил.
- 11 Черпаков, И. В. Основы программирования [Электронный ресурс]: учебник и практикум для СПО / И. В. Черпаков. — М. : Издательство Юрайт, 2017. — 219 с. — Режим доступа: <https://www.biblio-online.ru>.

РАЗДЕЛ 2 СРЕДСТВА ИНФОРМАТИЗАЦИИ И КОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

Тема 2.1 История компьютера

План лекции

1. История развития вычислительной техники
2. Поколения электронно-вычислительных машин.

1. История развития вычислительной техники

Стремительное развитие цифровой вычислительной техники (ВТ) и становление науки о принципах ее построения и проектирования началось в 40-х годах XX века, когда технической базой ВТ стала электроника и микроэлектроника, а основой для развития архитектуры компьютеров (называемых ранее ЭВМ) – достижения в области искусственного интеллекта.

До этого времени в течение почти 500 лет ВТ сводилась к простейшим устройствам для выполнения арифметических операций над числами. Основой практически всех изобретенных за 5 столетий устройств было зубчатое колесо, рассчитанное на фиксацию 10 цифр десятичной системы счисления. Первый в мире эскизный рисунок тринадцатиразрядного десятичного суммирующего устройства на основе таких колес принадлежит Леонардо да Винчи.

Первым реально осуществленным механическим цифровым вычислительным устройством стала "Паскалина" великого французского ученого Блеза Паскаля, которая представляла собой 6-ти (или 8-ми) разрядное устройство, на зубчатых колесах, рассчитанное на суммирование и вычитание десятичных чисел (1642 г.).

Через 30 лет после "Паскалины" в 1673 г. появился "арифметический прибор" Готфрида Вильгельма Лейбница - двенадцатиразрядное десятичное устройство для выполнения арифметических операций, включая умножение и деление.

В конце XVIII века во Франции произошли два события, имеющие принципиальное значение для дальнейшего развития цифровой вычислительной техники. К таким событиям относятся:

- изобретение Жозефом Жакардом программного управления ткацким станком с помощью перфокарт;
- разработка Гаспаром де Прони, технологии вычислений, разделившей численные вычисления на три этапа: разработка численного метода, составление программы последовательности арифметических действий, проведение собственно вычислений путем арифметических операций над числами в соответствии с составленной программой.

Указанные новшества позже были использованы англичанином Чарльзом Беббиджем, осуществившим, качественно новый шаг в развитии средств ВТ – *переход от ручного к автоматическому выполнению вычислений по составленной программе*. Им был разработан проект Аналитической машины - механической универсальной цифровой вычислительной машины с программным управлением (1830-1846 гг.). Машина состояла из пяти устройств: арифметическое (АУ); запоминающее (ЗУ); управления (УУ); ввода (УВВ); вывода (УВ).

Именно из таких устройств и состояли первые ЭВМ, появившиеся спустя 100 лет. АУ строилось на основе зубчатых колес, на них же предлагалось реализовать ЗУ (на тысячи 50-разрядных чисел). Для ввода данных и программы использовались перфокарты. Предполагаемая скорость вычислений - сложение и вычитание за 1 сек, умножение и деление - за 1 мин. Помимо арифметических операций имелась команда условного перехода.

Следует отметить, что хотя и были созданы отдельные узлы машины, всю машину из-за ее громоздкости создать не удалось. Только зубчатых колес для нее понадобилось бы более 50 000. Изобретатель намеревался использовать паровую машину для приведения в действие своей аналитической машины .

В 1870 г. (за год до смерти Беббиджа) английский математик Джевонс сконструировал первую в мире "логическую машину", позволяющую механизировать простейшие логические выводы.

Создателями логических машин в дореволюционной России стали Павел Дмитриевич Хрущев (1849-1909) и Александр Николаевич Щукарев (1884-1936), работавшие в учебных заведениях Украины.

Гениальную идею Беббиджа осуществил американский ученый Говард Айкен, создавший в 1944 г. первый в США релейно-механический компьютер. Ее основные блоки – арифметики и памяти – были исполнены на зубчатых колесах. Если Беббидж намного опередил свое время, то Айкен, используя все те же зубчатые колеса, в техническом плане при реализации идеи Беббиджа использовал устаревшие решения.

Следует отметить, что десятью годами ранее, в 1934 г. немецкий студент Конрад Цузе, работавший над дипломным проектом, решил сделать цифровую вычислительную машину с программным управлением. В этой машине впервые в мире была использована двоичная система исчисления. В 1937 г. машина Z1 произвела первые вычисления. Она была двоичной 22-х разрядной с плавающей запятой с памятью на 64 числа, и работала на чисто механической (рычажной) основе.

В том же 1937 г., когда заработала первая в мире механическая двоичная машина Z1, Джон Атанасов (болгарин по происхождению, живший в США) начал разработку специализированного компьютера, впервые в мире применив электронные лампы (300 ламп).

В 1942-43 годах в Англии была создана (с участием Алана Тьюринга) вычислительная машина "Колоссус". Эта машина, состоящая из 2000 электронных ламп, предназначалась для расшифровки радиোগрамм германского вермахта. Поскольку работы Цузе и Тьюринга были секретными, о них в то время знали немногие и они не вызвали какого-либо резонанса в мире.

Только в 1946 г. появилась информация об ЭВМ "ЭНИАК" (электронный цифровой интегратор и компьютер), созданной в США Д. Мочли и П.

Эккертом, с применением электронной техники. В машине использовалось 18 тысяч электронных ламп, и она выполняла около 3-х тыс. операций в сек. Однако, машина оставалась десятичной, а ее память составляла лишь 20 слов. Программы хранились вне оперативной памяти.

Почти одновременно, в 1949-52 гг. ученые Англии, Советского Союза и США (Морис Уилкс, ЭВМ "ЭДСАК", 1949 г.; Сергей Лебедев, ЭВМ "МЭСМ", 1951 г.; Исаак Брук, ЭВМ "М1", 1952 г.; Джон Мочли и Преспер Эккерт, Джон фон Нейман ЭВМ "ЭДВАК", 1952 г.), создали ЭВМ с хранимой в памяти программой.

В общем случае выделяют **пять поколений** ЭВМ.

Первое поколение (1945-1954) характеризуется появлением техники на электронных лампах. Это эпоха становления вычислительной техники. Большинство машин первого поколения были экспериментальными устройствами и строились с целью проверки тех или иных теоретических положений. Вес и размеры этих компьютеров были такими, что они нередко требовали для себя отдельных зданий.

Основоположниками компьютерной науки по праву считаются Клод Шеннон – создатель теории информации, Алан Тьюринг – математик, разработавший теорию программ и алгоритмов, и Джон фон Нейман - автор конструкции вычислительных устройств, которая до сих пор лежит в основе большинства компьютеров. В те же годы возникла еще одна новая наука, связанная с информатикой, – кибернетика – наука об управлении как одном из основных информационных процессов. Основателем кибернетики является американский математик Норберт Винер.

Во втором поколении (1955-1964) вместо электронных ламп использовались транзисторы, а в качестве устройств памяти стали применяться магнитные сердечники и магнитные барабаны - далекие предки современных жестких дисков. Все это позволило резко уменьшить габариты и стоимость компьютеров, которые тогда впервые стали строиться на продажу.

Но главные достижения этой эпохи принадлежат к области программ. Во втором поколении впервые появилось то, что сегодня называется операционной системой. Тогда же были разработаны первые языки высокого уровня - Фортран, Алгол, Кобол. Эти два важных усовершенствования позволили значительно упростить и ускорить написание программ для компьютеров.

При этом расширялась сфера применения компьютеров. Теперь уже не только ученые могли рассчитывать на доступ к вычислительной технике, поскольку компьютеры нашли применение в планировании и управлении, а некоторые крупные фирмы даже начали компьютеризовать свою бухгалтерию, предвосхищая этот процесс на двадцать лет.

В третьем поколении (1965-1974) впервые стали использоваться интегральные схемы - целые устройства и узлы из десятков и сотен транзисторов, выполненные на одном кристалле полупроводника (микросхемы). В это же время появляется полупроводниковая память, которая и по сей день используется в персональных компьютерах в качестве оперативной.

В эти годы производство компьютеров приобретает промышленный размах. Фирма IBM первой реализовала серию полностью совместимых друг с другом компьютеров от самых маленьких, размером с небольшой шкаф (меньше тогда еще не делали), до самых мощных и дорогих моделей. Наиболее распространенным в те годы было семейство System/360 фирмы IBM, на основе которого в СССР была разработана серия ЕС ЭВМ. Еще в начале 60-х появляются первые миникомпьютеры - небольшие маломощные компьютеры, доступные по цене небольшим фирмам или лабораториям. Миникомпьютеры представляли собой первый шаг на пути к персональным компьютерам, пробные образцы которых были выпущены только в середине 70-х годов.

Между тем количество элементов и соединений между ними, уместающихся в одной микросхеме, постоянно росло, и в 70-е годы интегральные схемы содержали уже тысячи транзисторов.

В 1971 г. фирма Intel, выпустив первый микропроцессор, который предназначался для только-только появившихся настольных калькуляторов. Этому изобретению суждено было произвести в следующем десятилетии настоящую революцию. Микропроцессор является главной составляющей частью современного персонального компьютера.

На рубеже 60-х и 70-х годов двадцатого столетия (1969 г) зародилась первая глобальная компьютерная сеть ARPA, прототип современного Интернета. В том же 1969 г. одновременно появились операционная система Unix и язык программирования C ("Си"), оказавшие огромное влияние на программный мир и до сих пор сохраняющие свое передовое положение.

Четвертое поколение (1975 – 1985) характеризуется все меньшим количеством принципиальных новаций в компьютерной науке. Прогресс идет в основном по пути развития того, что уже изобретено и придумано, прежде всего за счет повышения мощности и миниатюризации элементной базы и самих компьютеров.

Самая главная новация четвертого поколения – это появление в начале 80-х годов персональных компьютеров. Благодаря персональным компьютерам вычислительная техника становится по-настоящему массовой и общедоступной. Несмотря на то, что персональные и миникомпьютеры по-прежнему в вычислительных мощностях отстают от больших машин, львиная доля новшеств, таких как графический пользовательский интерфейс, новые периферийные устройства, глобальные сети, связана появлением и развитием именно этой техники.

Большие компьютеры и суперкомпьютеры, конечно же, продолжают развиваться. Но теперь они уже не доминируют на компьютерной арене, как было раньше.

Некоторые характеристики вычислительной техники четырех поколений приведены в табл. 1.1.

Таблица 1.1

Поколения вычислительной техники

Поколение	I	II	III	IV
Годы	1945-1954	1955-1964	1965-1974	1975-1985
Основной элемент	Эл. лампа	Транзистор	Интегральная схема	Большая интегральная схема (микропроцессор)
Количество ЭВМ в мире (шт.)	Сотни	Тысячи	Десятки тысяч	Миллионы
Размеры ЭВМ	Большие	Значительно меньше	Мини-ЭВМ	микроЭВМ
Быстродействие (условное)операций/ сек	Несколько единиц	Несколько десятков	Несколько тысяч	Несколько десятков тысяч
Носитель информации	Перфокарта, Перфолента	Магнитная лента	Диск	Гибкий диск

Пятое поколение (1986 до настоящего времени) в значительной мере определяется результатами работы японского Комитета научных исследований в области ЭВМ, опубликованными в 1981г. Согласно этому проекту ЭВМ и вычислительные системы пятого поколения кроме высокой производительности и надежности при более низкой стоимости с помощью новейших технологий, должны удовлетворять следующим качественно новым функциональным требованиям:

- обеспечить простоту применения ЭВМ путем реализации систем ввода/вывода информации голосом, а также диалоговой обработки информации с использованием естественных языков;

- обеспечить возможность обучаемости, ассоциативных построений и логических выводов;
- упростить процесс создания программных средств путем автоматизации синтеза программ по спецификациям исходных требований на естественных языках;
- улучшить основные характеристики и эксплуатационные качества вычислительной техники для удовлетворения различных социальных задач, улучшить соотношения затрат и результатов, быстродействия, легкости, компактности ЭВМ;
- обеспечить разнообразие вычислительной техники, высокую адаптируемость к приложениям и надежность в эксплуатации.

В настоящее время ведутся интенсивные работы по созданию оптоэлектронных ЭВМ с массовым параллелизмом и нейронной структурой, представляющих собой распределенную сеть большого числа (десятки тысяч) несложных микропроцессоров, моделирующих архитектуру нейронных биологических систем

Учебно-методическое и информационное обеспечение

Основная литература:

- 1 Гаврилов, М. В. Информатика и информационные технологии [Электронный ресурс]: учебник для СПО / М. В. Гаврилов, В. А. Климов. — 4-е изд., перераб. и доп. — М. : Издательство Юрайт, 2017. — 383 с. — Режим доступа: <https://www.biblio-online.ru>.
- 2 Илюшечкин, В. М. Основы использования и проектирования баз данных [Электронный ресурс]: учебник для СПО / В. М. Илюшечкин. — испр. и доп. — М. : Издательство Юрайт, 2017. — 213 с. — Режим доступа: <https://www.biblio-online.ru>.
- 3 Казанский, А. А. Объектно-ориентированный анализ и программирование на visual basic 2013 [Электронный ресурс]: учебник для СПО / А. А.

- Казанский. — М. : Издательство Юрайт, 2017. — 290 с. — Режим доступа: <https://www.biblio-online.ru>.
- 4 Казанский, А. А. Прикладное программирование на excel 2013 [Электронный ресурс]: учебное пособие для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 159 с. — Режим доступа: <https://www.biblio-online.ru>.
 - 5 Казанский, А. А. Программирование на visual c# 2013 [Электронный ресурс]: учебное пособие для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 191 с. — Режим доступа: <https://www.biblio-online.ru>.
 - 6 Семакин И.Г. Основы алгоритмизации и программирования: учеб. для сред. проф.образования/ И.Г.Семакин, А.П. Шестаков. – 3-е изд., стер. – М.: Академия, 2012. – 392 с.
 - 7 Советов, Б. Я. Базы данных [Электронный ресурс]: учебник для СПО / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 2-е изд. — М. : Издательство Юрайт, 2017. — 463 с. — Режим доступа: <https://www.biblio-online.ru>.
 - 8 Советов, Б. Я. Информационные технологии [Электронный ресурс]: учебник для СПО / Б. Я. Советов, В. В. Цехановский. — 6-е изд., перераб. и доп. — М. : Издательство Юрайт, 2017. — 261 с. — Режим доступа: <https://www.biblio-online.ru>.
 - 9 Федорова Г. Н. Разработка и администрирование баз данных : учеб. для студ. учреждений сред. проф. образования / Г. Н. Федорова. - М.: Академия, 2015. - 313 с.
 - 10 Цветкова, М.С. Информатика и ИКТ : учеб. для студентов учреждений сред. проф. образования / М. С. Цветкова, Л. С. Великович. - 8-е изд., стер. - М. : Академия, 2016. - 332, [1] с., [4] л. ил.
 - 11 Черпаков, И. В. Основы программирования [Электронный ресурс]: учебник и практикум для СПО / И. В. Черпаков. — М. : Издательство Юрайт, 2017. — 219 с. — Режим доступа: <https://www.biblio-online.ru>.

Тема 2.2 Состав персонального компьютера

План лекции

- 1 Архитектура персонального компьютера. Основные функции ПК. Устройства ввода информации. Устройства вывода информации.
- 2 Память персонального компьютера. Виды памяти. Программное обеспечение ПК. Защита информации.

Архитектура персонального компьютера. Основные функции ПК.

Устройства ввода информации. Устройства вывода информации.

Компьютер (англ. computer — вычислитель) представляет собой программируемое электронное устройство, способное обрабатывать данные и производить вычисления, а также выполнять другие задачи манипулирования символами.

Существует два основных класса компьютеров:

- цифровые компьютеры, обрабатывающие данные в виде числовых двоичных кодов;
- аналоговые компьютеры, обрабатывающие непрерывно меняющиеся физические величины (электрическое напряжение, время и т. д.), которые являются аналогами вычисляемых величин.

Поскольку в настоящее время подавляющее большинство компьютеров являются цифровыми, далее будем рассматривать только этот класс компьютеров и слово "компьютер" употреблять в значении "цифровой компьютер". Основу компьютеров образует аппаратура (HardWare), построенная, в основном, с использованием электронных и электромеханических элементов и устройств. Принцип действия компьютеров состоит в выполнении программ (SoftWare) — заранее заданных, четко определённых последовательностей арифметических, логических и других операций.

Любая компьютерная программа представляет собой последовательность отдельных команд. Команда — это описание операции, которую должен выполнить компьютер. Как правило, у команды есть свой код (условное обозначение), исходные данные (операнды) и результат. Например, у команды "сложить два числа" операндами являются слагаемые, а результатом — их сумма. А у команды "стоп" операндов нет, а результатом является прекращение работы программы. Результат команды вырабатывается по точно определенным для данной команды правилам, заложенным в конструкцию компьютера. Совокупность команд, выполняемых данным компьютером, называется системой команд этого компьютера.

Компьютеры работают с очень высокой скоростью, составляющей миллионы – сотни миллионов операций в секунду.

Персональные компьютеры, более чем какой-либо другой вид ЭВМ, способствуют переходу к новым компьютерным информационным технологиям, которым свойственны:

- дружественный информационный, программный и технический интерфейс с пользователем;
- выполнение информационных процессов в режиме диалога с пользователем;
- сквозная информационная поддержка всех процессов на основе интегрированных баз данных;
- так называемая “безбумажная технология”.

Компьютер — это многофункциональное электронное устройство для накопления, обработки и передачи информации. Под архитектурой компьютера понимается его логическая организация, структура и ресурсы, т. е. средства вычислительной системы, которые могут быть выделены процессу обработки данных на определенный интервал времени.

В основу построения большинства ЭВМ положены принципы, сформулированные в 1945 г. Джоном фон Нейманом:

1. Принцип программного управления (программа состоит из набора команд, которые выполняются процессором автоматически друг за другом в определённой последовательности).

2. Принцип однородности памяти (программы и данные хранятся в одной и той же памяти; над командами можно выполнять такие же действия, как и над данными).

3. Принцип адресности (основная память структурно состоит из нумерованных ячеек).

ЭВМ, построенные на этих принципах, имеют классическую архитектуру (архитектуру фон Неймана). Архитектура ПК определяет принцип действия, информационные связи и взаимное соединение основных логических узлов компьютера:

- центрального процессора;
- основной памяти;
- внешней памяти;
- периферийных устройств.

Основные электронные компоненты, определяющие архитектуру процессора, размещаются на основной плате компьютера, которая называется системной или материнской (MotherBoard). А контроллеры и адаптеры дополнительных устройств либо сами эти устройства, выполняются в виде плат расширения (DaughterBoard — дочерняя плата) и подключаются к шине с помощью разъёмов расширения, называемых также слотами расширения (англ. slot — щель, паз)

Функционально-структурная организация

Основные блоки ПК и их значение

Архитектура компьютера обычно определяется совокупностью ее свойств, существенных для пользователя. Основное внимание при этом уделяется структуре и функциональным возможностям машины, которые можно разделить на основные и дополнительные. Основные функции определяют назначение ЭВМ: обработка и хранение информации, обмен

информацией с внешними объектами. Дополнительные функции повышают эффективность выполнения основных функций: обеспечивают эффективные режимы ее работы, диалог с пользователем, высокую надежность и др. Названные функции ЭВМ реализуются с помощью ее компонентов: аппаратных и программных средств.

Структура компьютера — это некоторая модель, устанавливающая состав, порядок и принципы взаимодействия входящих в нее компонентов. Персональный компьютер — это настольная или переносная ЭВМ, удовлетворяющая требованиям общедоступности и универсальности применения. Достоинствами ПК являются:

- малая стоимость, находящаяся в пределах доступности для индивидуального покупателя;
- автономность эксплуатации без специальных требований к условиям окружающей среды;
- гибкость архитектуры, обеспечивающая ее адаптивность к разнообразным применениям в сфере управления, науки, образования, в быту;
- "дружественность" операционной системы и прочего программного обеспечения, обуславливающая возможность работы с ней пользователя без специальной профессиональной подготовки;
- высокая надежность работы (более 5 тыс. ч наработки на отказ).

Структура персонального компьютера

Рассмотрим состав и назначение основных блоков ПК.

Примечание. Здесь и далее организация ПК рассматривается применительно к самым распространенным в настоящее время IBM PC-подобным компьютерам.

Микропроцессор (МП) — это центральный блок ПК, предназначенный для управления работой всех блоков машины и для выполнения арифметических и логических операций над информацией.

В состав микропроцессора входят:

- устройство управления (УУ) — формирует и подает во все блоки машины в нужные моменты времени определенные сигналы управления (управляющие импульсы), обусловленные спецификой выполняемой операции и результатами предыдущих операций; формирует адреса ячеек памяти, используемых выполняемой операцией, и передает эти адреса в соответствующие блоки ЭВМ; опорную последовательность импульсов устройство управления получает от генератора тактовых импульсов;
- арифметико-логическое устройство (АЛУ) — предназначено для выполнения всех арифметических и логических операций над числовой и символьной информацией (в некоторых моделях ПК для ускорения выполнения операций к АЛУ подключается дополнительный математический сопроцессор);
- микропроцессорная память (МПП) — служит для кратковременного характера, записи и выдачи информации, непосредственно используемой в вычислениях в ближайшие такты работы машины, ибо основная память (ОП) не всегда обеспечивает скорость записи, поиска и считывания информации, необходимую для эффективной работы быстродействующего микропроцессора. Регистры — быстродействующие ячейки памяти различной длины (в отличие от ячеек ОП, имеющих стандартную длину 1 байт и более низкое быстродействие);
- интерфейсная система микропроцессора — реализует сопряжение и связь с другими устройствами ПК; включает в себя внутренний интерфейс МП, буферные запоминающие регистры и схемы управления портами ввода-вывода (ПВВ) и системной шиной. Интерфейс (interface) — совокупность средств сопряжения и связи устройств компьютера, обеспечивающая их эффективное взаимодействие. Порт ввода-вывода (I/O — Input/Output port) — аппаратура сопряжения, позволяющая подключить к микропроцессору другое устройство ПК.

Генератор тактовых импульсов

Он генерирует последовательность электрических импульсов; частота генерируемых импульсов определяет тактовую частоту машины. Промежуток времени между соседними импульсами определяет время одного такта работы машины или просто такт работы машины.

Частота генератора тактовых импульсов является одной из основных характеристик персонального компьютера и во многом определяет скорость его работы, ибо каждая операция в машине выполняется за определенное количество тактов.

Системная шина

Это основная интерфейсная система компьютера, обеспечивающая сопряжение и связь всех его устройств между собой. Системная шина включает в себя:

- кодовую шину данных (КШД), содержащую провода и схемы сопряжения для параллельной передачи всех разрядов числового кода (машинного слова) операнда;
- кодовую шину адреса (КША), включающую провода и схемы сопряжения для параллельной передачи всех разрядов кода адреса ячейки основной памяти или порта ввода – вывода внешнего устройства;
- кодовую шину инструкций (КШИ), содержащую провода и схемы сопряжения для передачи инструкций (управляющих сигналов, импульсов) во все блоки машины;
- шину питания, имеющую провода и схемы сопряжения для подключения блоков ПК к системе энергоснабжения.

Системная шина обеспечивает три направления передачи информации:

- между микропроцессором и основной памятью;
- между микропроцессором и портами ввода-вывода внешних устройств;
- между основной памятью и портами ввода-вывода внешних устройств (в режиме прямого доступа к памяти).

Не блоки, а точнее их порты ввода – вывода, через соответствующие унифицированные разъемы (стыки) подключаются к шине единообразно:

Непосредственно или через контроллеры (адаптеры). Управление системной шины осуществляется микропроцессором либо непосредственно, либо, что чаще, через дополнительную микросхему- *контроллер шины*, формирующий основные сигналы управления.

Основная память (ОП)

Она предназначена для хранения и оперативного обмена информацией с прочими блоками машины. ОП содержит два вида запоминающих устройств: постоянное запоминающее устройство (ПЗУ) и оперативное запоминающее устройство (ОЗУ).

ПЗУ служит для хранения неизменяемой (постоянной) программной и справочной информации, позволяет оперативно только считывать хранящуюся в нем информацию (изменить информацию в ПЗУ нельзя).

ОЗУ предназначено для оперативной записи, хранения и считывания информации (программ и данных), непосредственно участвующей в информационно-вычислительном процессе, выполняемом ПК в текущий период времени. Главными достоинствами оперативной памяти являются ее высокое быстродействие и возможность обращения к каждой ячейке памяти отдельно (прямой адресный доступ к ячейке). В качестве недостатка ОЗУ следует отметить невозможность сохранения информации в ней после выключения питания машины (энергозависимость).

Внешняя память

Она относится к внешним устройствам ПК и используется для долговременного хранения любой информации, которая может когда-либо потребоваться для решения задач. В частности, во внешней памяти хранится все программное обеспечение компьютера. Внешняя память содержит разнообразные виды запоминающих устройств, но наиболее распространенными, имеющимися практически на любом компьютере, являются накопители на жестких (HDD) и гибких (HD) магнитных дисках.

Назначение этих накопителей — хранение больших объемов информации, запись и выдача хранимой информации по запросу в оперативное

запоминающее устройство. В качестве устройств внешней памяти используются также запоминающие устройства на магнитной дискете, накопители на оптических дисках (CD-ROM-Compact Disk Read Only, DVD, Memory-компакт-диск с памятью, только читаемой) и другие.

Источник питания

Это блок, содержащий системы автономного и сетевого энергоснабжения ПК.

Таймер

Это внутримашинные электронные часы, обеспечивающие при необходимости автоматический съем текущего момента времени (год, месяц, часы, минуты, секунды и доли секунд). Таймер подключается к автономному источнику питания — аккумулятору и при отключении машины от сети продолжает работать.

Внешние устройства (ВУ)

Это важнейшая составная часть любого вычислительного комплекса. Достаточно сказать, что по стоимости ВУ иногда составляют 50 – 80 % всего ПК. От состава и характеристик ВУ во многом зависят возможность и эффективность применения ПК в системах управления и в народном хозяйстве в целом.

ВУ ПК обеспечивают взаимодействие машины с окружающей средой пользователями, объектами управления и другими ЭВМ. ВУ весьма разнообразны и могут быть классифицированы по ряду признаков. Так, по назначению можно выделить следующие виды ВУ:

- внешние запоминающие устройства (ВЗУ) или внешняя память ПК;
- диалоговые средства пользователя;
- устройства ввода информации;
- устройства вывода информации;
- средства связи и телекоммуникации.

Диалоговые средства пользователя включают в свой состав видеомониторы (дисплеи), реже пультовые пишущие машинки (принтеры с клавиатурой) и устройства речевого ввода-вывода информации

Видеомонитор (дисплей) — устройство для отображения вводимой и выводимой из ПК информации.

Устройства речевого ввода – вывода относятся к средствам мультимедиа. Устройства речевого ввода — это различные микрофонные акустические системы, "звуковые мыши", например, со сложным программным обеспечением, позволяющим распознавать произносимые человеком буквы и слова, идентифицировать их и закодировать.

Устройства речевого вывода — это различные синтезаторы звука, выполняющие преобразования цифровых кодов в буквы и слова, воспроизводимые через динамики или звуковые колонки, подсоединенные к компьютеру.

К устройствам ввода информации относятся:

- клавиатура — устройство для ручного ввода числовой, текстовой и управляющей информации в ПК;
- графические планшеты (диджитайзеры) для ручного ввода графической информации, изображений путем перемещения по планшету специального указателя (пера); при перемещении пера автоматически выполняются считывание координат его местоположения и ввод этих координат в ПК;
- сканеры — для автоматического считывания с бумажных носителей и ввода в ПК машинописных текстов, графиков, рисунков, чертежей; в устройстве кодирования сканера в текстовом режиме считанные символы после сравнения с эталонными контурами специальными программами преобразуются в коды ASCII, а в графическом режиме считанные графики и чертежи преобразуются в последовательности двумерных координат;

- манипуляторы (устройства указания): джойстик-рычаг, мышь, трекбол-шар в оправе, световое перо и др. — для ввода графической информации на экран дисплея путем управления движением курсора по экрану с последующим кодированием координат курсора и вводом их в ПК;
- сенсорные экраны — для ввода отдельных элементов изображения, программ или команд с полиэкрана дисплея в ПК.

К устройствам вывода информации относятся:

- принтеры — печатающие устройства для регистрации информации на бумажный носитель;
- графопостроители (плоттеры) — для вывода графической информации (графиков, чертежей, рисунков) из ПК на бумажный носитель; плоттеры бывают векторные с вычерчиванием изображения с помощью пера и растровые: термографические, электростатические, струйные и лазерные. По конструкции плоттеры подразделяются на планшетные и барабанные. Основные характеристики всех плоттеров примерно одинаковые: скорость вычерчивания — 100 – 1000 мм/с (у лучших моделей возможны цветное изображение и передача полутонов); наибольшая разрешающая способность и четкость изображения — у лазерных плоттеров, но они самые дорогие.

Устройства связи и телекоммуникации для связи с приборами и другими средствами автоматизации (согласователи интерфейсов, адаптеры, цифроаналоговые и аналого-цифровые преобразователи и т. п.) и для подключения ПК к каналам связи, к другим ЭВМ и вычислительным сетям (сетевые интерфейсные платы, "стыки", мультиплексоры передачи данных, модемы).

В частности сетевой адаптер является внешним интерфейсом ПК и служит для подключения его к каналу связи для обмена информацией с другими ЭВМ, для работы в составе вычислительной сети. В глобальных сетях функции сетевого адаптера выполняет модулятор- демодулятор.

Многие из названных выше устройств относятся к условно выделенной группе — средствам мультимедиа. Средства мультимедиа (multimedia — многосредовость) — это комплекс аппаратных и программных средств, позволяющих человеку общаться с компьютером, используя самые разные, естественные для себя среды: звук, видео, графику, тексты, анимацию и так далее.

К средствам мультимедиа относятся: устройства речевого ввода и вывода информации; широко распространенные уже сейчас сканеры (поскольку они позволяют автоматически вводить в компьютер печатные тексты и рисунки); высококачественные видео- (video-) и звуковые (sound-) платы, платы видеозахвата (videograbber), снимающие изображение с видеомаягнитофона или видеокамеры и вводящие его в ПК; высококачественные акустические и видеовоспроизводящие системы с усилителями, звуковыми колонками, большими видеоэкранами. Но, пожалуй, еще с большим основанием к средствам мультимедиа относят внешние запоминающие устройства большой емкости на оптических дисках, часто используемые для записи звуковой и видеоинформации

Стоимость компакт-дисков (CD) при их массовом тиражировании невысока, а, учитывая их большую емкость (650 – 700 Мбайт, а новых типов 8 Гбайт и выше), высокие надежность и долговечность, стоимость хранения информации на CD для пользователя оказывается несравнимо меньшей, нежели на магнитных дисках. На компакт-дисках организуются обширные базы данных, целые библиотеки; на CD представлены словари, справочники, энциклопедии; обучающие и развивающие программы по общеобразовательным и общим предметам.

CD широко используется, например, при изучении иностранных языков, правил дорожного движения, бухгалтерского учета, законодательства вообще и налогового законодательства в частности. И все это сопровождается текстами и рисунками, речевой информацией и мультипликацией, музыкой и видео. В чисто бытовом аспекте CD можно использовать для хранения аудио- и

видеозаписей, т. е. использовать вместо магнитных аудиокассет и видеокассет. Следует упомянуть, конечно, и о большом количестве программ компьютерных игр, хранимых на CD.

Таким образом, CD-ROM открывает доступ к огромным объемам разнообразной и по функциональному назначению, и по среде воспроизведения информации, записанной на компакт-дисках.

Прерывание — временный останов выполнения одной программы в целях оперативного выполнения другой, а в данный момент более важной (приоритетной) программы. Прерывания возникают при работе компьютеры постоянно. Достаточно сказать, что все процедуры ввода – вывода информации выполняются по прерываниям, например прерывания от таймера возникают и обслуживаются контроллером прерываний 18 раз в секунду (естественно, пользователь их не замечает).

Контроллер прерываний обслуживает процедуры прерывания, принимает запрос на прерывание от внешних устройств, определяет уровень приоритета этого запроса и выдает сигнал прерывания в МП. МП, получив этот сигнал, приостанавливает выполнение текущей программы и переходит к выполнению специальной программы обслуживания того прерывания, которое запросило внешнее устройство. После завершения программы обслуживания восстанавливается выполнение прерванной программы.

Внутримашинный системный интерфейс — система связи и сопряжения узлов и блоков ЭВМ между собой — представляет собой совокупность электрических линий связи (проводов), схем сопряжения с компонентами компьютера, протоколов (алгоритмов) передачи и преобразования сигналов.

Существует два варианта организации внутримашинного интерфейса:

1. Многосвязный интерфейс: каждый блок ПК связан с прочими блоками своими локальными проводами; интерфейс применяется, как правило, только в простейших бытовых устройствах.

2. Односвязный интерфейс: все блоки ПК связаны друг с другом через общую или системную шину.

В подавляющем большинстве современных ПК в качестве системного интерфейса используется системная шина. Структура и состав системной шины были рассмотрены ранее. Важнейшими функциональными характеристиками системной шины являются: количество обслуживаемых ею устройств и ее пропускная способность, т. е. максимально возможная скорость передачи информации. Пропускная способность шины зависит от ее разрядности (есть шины 8-, 16-, 32- и 64- разрядные) и тактовой частоты, на которой шина работает.

В качестве системной шины в разных ПК использовались и могут использоваться:

- шины расширений — шины общего назначения, позволяющие подключать большое число самых разнообразных устройств;
- локальные шины, специализирующиеся на обслуживании небольшого количества устройств определенного класса.

Шины расширений:

- шина Multibus 1 имеет две модификации: PC/XT bus (personal Computer eXtended Technology) — ПК с расширенной технологией и PC/AT bus (PC Advachnology — ПК с усовершенствованной технологией);
- шина PC/XT bus — 8-разрядная шина данных и 20-разрядная шина адреса, рассчитанная на тактовую частоту 4,77 МГц; имеет 3 линии для адаптерных прерываний и 3 канала для прямого доступа в память (каналы DMA — Direct Memory Access). Шина адреса ограничивала адресное пространство микропроцессора величиной 1 Мбайт. Используется с МП 8086, 8088;
- шина PC/At bus — 16 разрядная шина данных и 24-разрядная шина адреса, рабочая тактовая частота до 8 МГц, но может использоваться и МП с тактовой частотой 16 МГц, так как контроллер шины может делить частоту пополам; имеет 7 линий для адаптерных прерываний и 4 канала DMA. Используется с МП 80286;

- шина ISA (Industry Standard Architecture — архитектура промышленного стандарта) — 16-разрядная шина данных и 24-разрядная шина адреса, рабочая тактовая частота 16 МГц, но может использоваться и МП с тактовой частотой 50 МГц (коэффициент деления увеличен); по сравнению с шинами PC/XT и PC/AT увеличено количество линий аппаратных прерываний с 7 до 15 и каналов прямого доступа к памяти DMA с 7 до 11. Благодаря 24-разрядной шине адреса адресное пространство увеличилось с 1 до 16 Мбайт. Теоретическая пропускная способность шины данных равна 16 Мбайт/с, но реально она ниже, около 3 – 5 Мбайт/с, ввиду ряда особенностей ее использования. С появлением 32-разрядных высокоскоростных МП шина ISA стала существенным препятствием увеличения быстродействия ПК;
- шина EISA (Extended ISA) — 32-разрядная шина данных и 32-разрядная шина адреса, создана в 1989г. Адресное пространство шины 4 Гбайта, пропускная способность 33 Мбайт/с, причем скорость обмена по каналу МП - КЭШ - ОП определяется параметрами микросхем памяти, увеличено число разъемов расширений, (теоретически может подключаться до 15 устройств, практически до — 10). Улучшена система прерываний, шина EISA обеспечивает автоматическое конфигурирование системы и управление DMA; полностью совместима с шиной ISA(есть разъемы для подключения ISA),шина поддерживает многопроцессорную архитектуру вычислительных систем. Шина EISA весьма дорогая и применяется в скоростных ПК, сетевых серверах и рабочих станциях;
- шина MCA (Micro Channel Architecture) — 32-разрядная шина, созданная фирмой IBM в 1987г. для машин PC/2, пропускная способность 76 Мбайт/с, рабочая частота 10-20МГц. По своим прочим характеристикам близка к шине EISA, но не совместима ни с ISA, ни с EISA. Поскольку ЭВМ PS/2 не получили широкого распространения, в первую очередь ввиду отсутствия наработанного обилия прикладных программ, шина MCA также используется не очень широко.

Современные вычислительные системы характеризуются:

- стремительным ростом быстродействия микропроцессоров (например, МП Pentium может выдавать данные со скоростью 528 Мбайт/с по 64-разрядной шине данных) и некоторых внешних устройств (так, для отображения цифрового полноэкранного видео с высоким качеством необходима пропускная способность 22 Мбайт/с);
- появлением программ, требующих выполнения большого количества интерфейсных операций (например, программы обработки графики в Windows, работа в среде Multimedia).

В этих условиях пропускной способности шин расширения, обслуживающих одновременно несколько устройств, оказалось недостаточно для комфортной работы пользователей, ибо компьютеры стали подолгу "задумываться".

Разработчики интерфейсов пошли по пути создания локальных шин, подключаемых непосредственно к шине МП, работающих на тактовой частоте МП, (но не на внутренней рабочей его частоте) и обеспечивающих связь с некоторыми скоростными внешними по отношению к МП, устройствами: основной и внешней памятью, видеосистемами.

Сейчас существуют два основных стандарта универсальных локальных шин — VLB и PCI:

- Шина VLB (VESA Local Bus — локальная шина VESA) — разработана в 1992г. Ассоциацией стандартов видео оборудования (VESA — Video Electronics Standards Association), поэтому часто ее называют шиной VESA.
- Шина VLB, по существу, является расширением внутренней шины МП для связи с видеоадаптером и реже с винчестером, на подходе 64-разрядный вариант шины. Реальная скорость передачи данных по VLB-80 Мбайт/с (теоретически достижимая — 132 Мбайт/с).

Недостатки шины:

- рассчитана на работу МП 80386,80486, не адаптирована для процессоров Pentium, Pentium Pro, Power PC;
- жесткая зависимость от тактовой частоты МП (каждая шина VLB рассчитана только на конкретную частоту);
- малое количество подключаемых устройств — к шине VLB могут подключаться только четыре устройства;
- отсутствует арбитраж шины — могут быть конфликты между подключаемыми устройствами.

Шина PCI (Peripheral Component Interconnect — соединение внешних устройств) разработана в 1993 г. фирмой Intel. Шина PCI является на много более универсальной, чем VLB. Имеет свой адаптер, позволяющий ей настраиваться на работу с любым МП: 80486, Pentium, Pentium Pro, Power PC и др.; она позволяет подключать 10 устройств самой разной конфигурации с возможностью автоконфигурирования, имеет свой "арбитраж", средства управления передачей данных. Шина PCI пока еще весьма дорогая.

Разрядность PCI — 32 бита с возможностью расширения до 64 бит, теоретическая пропускная способность 132 Мбайта/с (реальная вдвое ниже). Шина PCI хотя и является локальной, выполняет и многие функции шины расширения, в частности, шины расширения ISA, EISA, MCA (а она совместима с ними) при наличии шины PCI подключаются не посредственно к МП (как это имеет место при использовании шины VLB), а к самой шине PCI (через интерфейс расширения).

Следует иметь в виду, что использование в ПК шин VLB и PCI возможно только при наличии соответствующей VLB — или PCI-материнской платы. Выпускаются материнские платы с мультишинной структурой, позволяющей использовать ISA/EISA, VLB и PCI, так называемые материнские платы с шиной VIP (по начальным буквам VLB, ISA и PCI).

Но в настоящее время платы с шинами VLB не производится и отмирает шина ISA, появились новые шины, такие как AGP, предназначенные для

видеоадаптеров с высокой пропускной способностью или так называемые 3D ускорители.

Функциональные устройства ПК

Основными характеристиками ПК являются:

1. Быстродействие.
2. Производительность.
3. Тактовая частота.

Единицами измерения быстродействия служат:

- МИПС (MIPS — Mega Instruction Per Second) — миллион операций над числами с фиксированной запятой (точкой);
- МФЛОПС (MFLOPS — Mega Floating Operations Second) — миллион операций над числами с плавающей запятой (точкой);
- КОПС (KOPS — Kilo Operations Per Second) для низко производительных ЭВМ — тысяча неких усредненных операций над числами;
- ГФЛОПС (GFLOPS — Giga Floating Operations Per Second) — миллиард операций в секунду над числами с плавающей запятой (точкой).

Оценка производительности ЭВМ всегда приблизительная, ибо при этом ориентируются на некоторые усредненные или, наоборот, на конкретные виды операций. Реально при решении различных задач используются и различные наборы операций. Поэтому для характеристики ПК вместо производительности обычно указывают тактовую частоту, более объективно определяющую быстродействие машины. И так как каждая операция требует для своего выполнения вполне определенного количество тактов. Зная тактовую частоту, можно достаточно точно определить время выполнения любой машинной операции.

1. Разрядность машины и кодовых шин интерфейса. Разрядность — это максимальное количество разрядов двоичного числа, над которым одновременно может выполняться машинная операция, в том числе и операция передачи информации; чем больше разрядность, тем, при прочих равных условиях, будет больше и производительность ПК.

2. Типы системного и локальных интерфейсов. Разные типы интерфейсов обеспечивают разные сроки передачи информации между узлами машины, позволяют подключать разное количество внешних устройств и различные их виды.

3. Емкость оперативной памяти. Емкость оперативной памяти измеряется чаще всего в мегабайтах (Мбайт). Напоминаем: 1 Мбайт = 1024 Кбайта = 1024 байт. Многие современные прикладные программы при оперативной памяти емкостью меньше 32 Мбайт просто не работают, либо работают, но очень медленно. Следует иметь в виду, что увеличение емкости основной памяти в два раза, помимо всего прочего, дает повышение эффективной производительности ЭВМ при решении сложных задач примерно в 1,7 раза.

4. Емкость накопителя на жестких магнитных дисках (винчестер). Емкость винчестера измеряется обычно в мегабайтах или гигабайтах (1 Гбайт = 1024 Мбайта).

5. Тип и емкость накопителей на гибких магнитных дисках и лазерных компакт дисков. Сейчас применяются накопители на гибких магнитных дисках, использующие дискеты размером 3,5 и 5,25 дюйма (практически уже не применяются) (1 дюйм = 25,4 мм). Первые имеют стандартную емкость 1,44 Мбайта, вторые 1,2 Мбайта. Также применяются накопители на компакт дисках в связи с их низкой стоимостью и большой емкостью, размером 650 и 700 Мб, применяются лазерные перезаписываемые диски CD-RW емкостью 650 – 700 Мб. Применяются и такой тип накопителя как DVD. Высокие технологии и высокая стоимость, но и большая емкость до 24 Gb.

6. Виды и емкость КЭШ-памяти. КЭШ-память — это буферная, недоступная для пользователей быстродействующая память, автоматически используемая компьютером для ускорения операций с информацией, хранящейся в более медленно действующих запоминающих устройствах. Например, для ускорения операций с основной памятью организуется регистровая КЭШ-память внутри микропроцессора (КЭШ-память первого уровня) или вне микропроцессора на материнской плате (КЭШ-память второго

уровня); для ускорения операций с дисковой памятью организуется КЭШ-память на ячейка электронной памяти. Следует иметь в виду, что наличие КЭШ-памяти емкостью 256 Кбайт увеличивает производительность ПК примерно на 20 %. Встречается емкость КЭШ-памяти и 512 Кбайт

7. Тип видеомонитора (дисплея) и видеоадаптера.

8. Тип принтера.

9. Наличие математического сопроцессора. Математический сопроцессор позволяет в десятки раз ускорить выполнение операций над двоичными числами с плавающей запятой и над двоично-кодированными десятичными числами.

10. Имеющееся программное обеспечение и вид операционной системы.

11. Аппаратная и программная совместимость с другими типами ЭВМ. Аппаратная и программная совместимость с другими типами ЭВМ означает возможность использования на компьютере соответственно тех же технических элементов и программного обеспечения, что и на других типах машин.

12. Возможность работы в вычислительной сети

13. Возможность работы в многозадачном режиме. Многозадачный режим позволяет выполнять вычисления одновременно по нескольким программам (многопрограммный режим) или для нескольких пользователей (многопользовательский режим). Совмещение во времени работы нескольких устройств машины, возможное в таком режиме, позволяет значительно увеличить эффективное быстродействие ЭВМ.

14. Надежность. Надежность — это способность системы выполнять полностью и правильно все заданные ей функции. Надежность ПК измеряется обычно средним временем наработки на отказ

15. Стоимость.

16. Габариты и масса.

Микропроцессоры

Микропроцессор, иначе, центральный процессор. Центральный процессор (CPU, от англ. Central Processing Unit) — это основной рабочий

компонент компьютера, который выполняет арифметические и логические операции, заданные программой, управляет вычислительным процессом и координирует работу всех устройств компьютера. Центральный процессор в общем случае содержит в себе:

- арифметико-логическое устройство;
- шины данных и шины адресов;
- регистры;
- счетчики команд;
- кэш — очень быструю память малого объема (от 8 до 512 Кбайт);
- математический сопроцессор чисел с плавающей точкой.

Современные процессоры выполняются в виде микропроцессоров. Физически микропроцессор представляет собой интегральную схему — тонкую пластинку кристаллического кремния прямоугольной формы площадью всего несколько квадратных миллиметров, на которой размещены схемы, реализующие все функции процессора. Кристалл-пластинка обычно помещается в пластмассовый или плоский керамический корпус и соединяется золотыми проводками с металлическими штырьками, чтобы его можно было присоединить к системной плате компьютера. В вычислительной системе может быть несколько параллельно работающих процессоров; такие системы называются многопроцессорными.

Первый микропроцессор был выпущен в 1971 г. фирмой Intel (США) — МП 4004. В настоящее время выпускается несколько сотен различных микропроцессоров, но наиболее популярными и распространенными являются микропроцессоры фирмы Intel и AMD.

Структура микропроцессора

Устройство управления является функционально наиболее сложным устройством ПК. Оно вырабатывает управляющие сигналы, поступающие по кодовым шинам инструкций во все блоки машины. Сюда включаются:

- Регистр команд — запоминающий регистр, в котором хранится код команды: код выполняемой операции и адреса операндов, участвующих в

операции. Регистр команд расположен в интерфейсной части МП, в блоке регистров команд.

- Дешифратор операций — логический блок, выбирающий в соответствии с поступающим из регистра команд кодом операции (КОП) один из множества имеющихся у него выходов.

- Постоянное запоминающее устройство микропрограмм — хранит в своих ячейках управляющие сигналы (импульсы), необходимые для выполнения в блоках ПК операций обработки информации. Импульс по выбранному дешифратором операции (в соответствии с кодом операции) считывает из ПЗУ микропрограмм необходимую последовательность управляющих сигналов.

- Узел формирования адреса (находится в интерфейсной части МП) — устройство, вычисляющее полный адрес ячейки памяти (регистра) по реквизитам, поступающим из регистра команд и регистров МПП.

- Кодовые шины данных, адреса и инструкций — часть внутренней шины микропроцессора. В общем случае УУ формирует управляющие сигналы для выполнения следующих основных процедур:

- о выборки из регистра-счетчика адреса команды МПП адреса ячейки ОЗУ, где хранится очередная команда программы;

- о выборки ИЗ ячеек ОЗУ кода очередной команды и приема считанной команды в регистр команд;

- о расшифровки кода операции и признаков выбранной команды;

- о считывания из соответствующих расшифрованному коду операции ячеек ПЗУ микропрограмм управляющих сигналов (импульсов), определяющих во всех блоках машины процедуры выполнения заданной операции, и пересылки управляющих сигналов в эти блоки;

- о считывания из регистра команд и регистров МПП отдельных составляющих адресов операндов (чисел), участвующих в вычислениях, и формирования полных адресов операндов;

- о выборки операндов (по сформированным адресам) и выполнения заданной операции обработки этих операндов;
- о записи результатов операции в память;
- о формировании адреса следующей команды программы.

Арифметико-логическое устройство предназначено для выполнения арифметических и логических операций преобразования информации. Функционально АЛУ состоит обычно из двух регистров, сумматора и схем управления (местного устройства управления).

Сумматор — вычислительная схема, выполняющая процедуру сложения поступающих на ее вход двоичных кодов; сумматор имеет разрядность двойного машинного слова.

Регистры — быстродействующие ячейки памяти различной длины: регистр 1 (Pr1) имеет разрядность двойного слова, а регистр 2 (Pr2) — разрядность слова. При выполнении операции в Pr1 помещается первое число, участвующее в операции, а по завершении операции — результат; в Pr2 — второе число, участвующее в операции (по завершении операции информация в нем не изменяется). Регистр 1 может принимать информацию с кодовых шин данных, и выдавать информацию с этих шин.

Схемы управления принимают по кодовым шинам инструкций управляющие сигналы от устройства управления и преобразуют их в сигналы для управления работой регистров и сумматора АЛУ. АЛУ выполняет арифметические операции (+, -, *, :) только над двоичной информацией с запятой, фиксированной после последнего разряда, т. е. только над целыми двоичными числами.

Выполнение операций над двоичными числами с плавающей запятой и над двоично-кодированными десятичными числами осуществляется или с привлечением математического сопроцессора, или по специально составленным программам.

Микропроцессорная память — память небольшой емкости, но чрезвычайно высокого быстродействия (время обращения к МПП, т. е. время,

необходимое на поиск, запись или считывание информации из этой памяти, измеряется наносекундами). Она предназначена для кратковременного хранения, записи и выдачи информации, непосредственно в ближайшие такты работы машины участвующей в вычислениях; МПП используется для обеспечения высокого быстродействия машины, ибо основная не всегда обеспечивает скорость записи, поиска и считывания информации, необходимую для эффективной работы быстродействующего микропроцессора. Микропроцессорная память состоит из быстродействующих регистров с разрядностью не меньше машинного слова. Количество и разрядность регистров в разных микропроцессорах различны.

Регистры микропроцессора делятся на регистры общего назначения и специальные. Специальные регистры применяются для хранения различных адресов (адреса команды, например), признаков результатов выполнения операций и режимов работы ПК (регистр флагов, например) и др. Регистры общего назначения являются универсальными и могут использоваться для хранения любой информации, но некоторые из них тоже должны быть обязательно задействованы при выполнении ряда процедур.

Интерфейсная часть МП предназначена для связи и согласования МП системной шиной ПК, а также для приема, предварительного анализа команд выполняемой программы и формирования полных адресов операндов и команд.

Интерфейсная часть включает в свой состав адресные регистры МПП, узел формирования адреса, блок регистров команд, являющийся буфером команд в МП, внутреннюю интерфейсную шину МП и схемы управления шиной и портами ввода – вывода.

Порты ввода – вывода — это пункты системного интерфейса ПК, через которые МП обменивается информацией с другими устройствами. Всего портов у МП может быть 65536. Каждый порт имеет адрес — номер порта, соответствующий адресу ячейки памяти, являющейся частью устройства ввода-вывода, использующего этот порт, а не частью основной памяти компьютера. Порт устройства содержит аппаратуру сопряжения и два регистра памяти —

для обмена данными и обмена управляющей информацией. Некоторые внешние устройства используют и основную память для хранения больших объемов информации, подлежащей обмену. Многие стандартные устройства (НЖМД, НГМД, клавиатура, принтер, сопроцессор и др.) имеют постоянно закрепленные за ними порты ввода – вывода.

Схема управления шиной и портами выполняет следующие функции:

- формирование адреса порта и управляющей информации для него (переключение порта на прием или передачу и др.);
- прием управляющей информации от порта, информации о готовности порта и его состоянии;
- организацию сквозного канала в системном интерфейсе для данных между портом устройства ввода – вывода и МП.

Схема управления шиной и портами использует для связи с портами кодовые шины инструкций, адреса и данные системной шины: при доступе к порту МП посылает сигнал по КШИ, который оповещает все устройства ввода-вывода, что адрес на КША является адресом порта, а затем посылает и сам адрес порта. То устройство, адрес порта которого совпадает, дает ответ о готовности, после чего по КШД осуществляется обмен данными.

Последовательность работы блоков ПК

Программа хранится во внешней памяти ПК. При запуске программы в работу пользователь выдает запрос на ее исполнение в дисковую операционную систему (DOS — Disc Operation System) компьютера. Запрос пользователя — это ввод имени исполняемой программы в командную строку на экране дисплея. Главная программа DOS-Command.com обеспечивает перезапись машинной (исполняемой) программы из внешней памяти в ОЗУ, в которой находится начало (первая команда) этой программы.

После этого автоматически начинается выполнение команд программы друг за другом. Каждая программа требует для своего исполнения нескольких тактов работы машины (такты определяются периодом следования импульсов от генератора тактовых импульсов). В первом такте выполнения любой

команды производятся считывание кода самой команды из ОЗУ по адресу, установленному в регистре-счетчике адреса, и запись этого кода в блок регистров команд устройства управления. Содержание второго и последующих тактов исполнения определяется результатами анализа команды, записанной в блок регистров команд, т. е. зависит уже от конкретной команды.

Пример. При выполнении ранее рассмотренной машинной команды: СЛ 0103 5102.

будут выполнены следующие действия:

- второй такт: считывание из ячейки 0103 ОЗУ первого слагаемого и перемещение его в АЛУ;
- третий такт: считывание из ячейки 5102 ОЗУ второго слагаемого и перемещение его в АЛУ;
- четвертый такт: сложение в АЛУ переданных туда чисел и формирование суммы;
- пятый такт: считывание из АЛУ суммы чисел и запись ее в ячейку 0103.

В конце последнего (в данном случае пятого) такта выполнения команды в регистр-счетчик адреса команд МПП будет добавлено число, равное количеству байтов, занимаемых кодом выполненной команды программы. Поскольку емкость одной ячейки памяти ОЗУ равна 1 байту и команды программы в ОЗУ размещены последовательно друг за другом, в регистре-счетчике адреса команд будет сформирован адрес следующей команды машинной программы, и машина приступит к ее исполнению и т. д. Команды будут выполняться последовательно одна за другой, пока не завершится вся программа. После завершения программы управление будет передано обратно в программу Command.com операционной системы.

Память персонального компьютера. Виды памяти. Программное обеспечение ПК. Защита информации.

Память компьютера построена из двоичных запоминающих элементов — битов, объединенных в группы по 8 битов, которые называются байтами.

(Единицы измерения памяти совпадают с единицами измерения информации). Все байты пронумерованы. Номер байта называется его адресом. Байты могут объединяться в ячейки, которые называются также словами. Для каждого компьютера характерна определенная длина слова — два, четыре или восемь байтов. Это не исключает использования ячеек памяти другой длины (например, полуслово, двойное слово). Как правило, в одном машинном слове может быть представлено либо одно целое число, либо одна команда. Однако допускаются переменные форматы представления информации.

Таблица 1.

Разбиение памяти на слова для четырехбайтовых компьютеров.

Байт 0	айт 1	айт 2	айт 3	айт 4	айт 5	айт 6	айт 7
полуслово		полу слово		полу слово		полу слово	
слово				слово			
двойное слово							

Широко используются и более крупные производные единицы объема памяти: Килобайт, Мегабайт, Гигабайт, а также, в последнее время, Терабайт и Петабайт.

Современные компьютеры имеют много разнообразных запоминающих устройств, которые сильно отличаются между собой по назначению, временным характеристикам, объёму хранимой информации и стоимости хранения одинакового объёма информации. Различают два основных вида памяти — внутреннюю и внешнюю. В состав внутренней памяти входят оперативная память, кэш-память и специальная память.

Оперативная память (ОЗУ, англ. RAM, Random Access Memory — память с произвольным доступом) — это быстрое запоминающее устройство не очень большого объёма, непосредственно связанное с процессором и предназначенное для записи, считывания и хранения выполняемых программ и данных, обрабатываемых этими программами.

Оперативная память используется только для временного хранения данных и программ, так как, когда машина выключается, все, что находилось в ОЗУ, пропадает. Доступ к элементам оперативной памяти прямой — это означает, что каждый байт памяти имеет свой индивидуальный адрес. Объем ОЗУ обычно составляет 32 – 512 Мбайта, а для эффективной работы современного программного обеспечения желательно иметь не менее 256 Мбайт ОЗУ. Обычно ОЗУ исполняется из интегральных микросхем памяти DRAM (Dynamic RAM — динамическое ОЗУ). Микросхемы DRAM работают медленнее, чем другие разновидности памяти, но стоят дешевле.

Каждый информационный бит в DRAM запоминается в виде электрического заряда крохотного конденсатора, образованного в структуре полупроводникового кристалла. Из-за токов утечки такие конденсаторы быстро разряжаются, и их периодически (примерно каждые 2 миллисекунды) подзаряжают специальные устройства. Этот процесс называется регенерацией памяти (Refresh Memory). Современные микросхемы имеют ёмкость 1 – 16 Мбит и более. Они устанавливаются в корпуса и собираются в модули памяти.

Наиболее распространены модули типа DIMM и SIMM. В модуле SIMM элементы памяти собраны на маленькой печатной плате длиной около 10 см. Ёмкость таких модулей неодинаковая — 256 Кбайт, 1, 2, 4, 8, 16, 32 и 64 Мбайта. Различные модули SIMM могут иметь разное число микросхем — девять, три или одну, и разное число контактов — 30 или 72.

Важная характеристика модулей памяти — время доступа к данным, которое обычно составляет 60 – 80 наносекунд.

В настоящее время SIMM'ы практически не применяются. На их смену пришли DIMM, а на смену DIMM приходят DDR и RIMM, но по сравнению с DIMM они имеют немного большую стоимость и соответственно повышенную скорость обмена.

Кэш (англ. cache), или сверхоперативная память — очень быстрое ЗУ небольшого объёма, которое используется при обмене данными между микропроцессором и оперативной памятью для компенсации разницы в

скорости обработки информации процессором и несколько менее быстродействующей оперативной памятью.

Кэш-памятью управляет специальное устройство — контроллер, который, анализируя выполняемую программу, пытается предвидеть, какие данные и команды вероятнее всего понадобятся в ближайшее время процессору, и подкачивает их в кэш-память. При этом возможны как "попадания", так и "промахи". В случае попадания, то есть, если в кэш подкачаны нужные данные, извлечение их из памяти происходит без задержки. Если же требуемая информация в кэше отсутствует, то процессор считывает её непосредственно из оперативной памяти. Соотношение числа попаданий и промахов определяет эффективность кэширования.

Кэш-память реализуется на микросхемах статической памяти SRAM (Static RAM), более быстродействующих, дорогих и малоёмких, чем DRAM. Современные микропроцессоры имеют встроенную кэш-память, так называемый кэш первого уровня размером 8 – 16 Кбайт. Кроме того, на системной плате компьютера может быть установлен кэш второго уровня ёмкостью от 64 Кбайт до 256 Кбайт и выше.

К устройствам специальной памяти относятся постоянная память (ROM), перепрограммируемая постоянная память (Flash Memory), память CMOS RAM, питаемая от батарейки, видеопамять и некоторые другие виды памяти.

Постоянная память (ПЗУ, англ. ROM, Read Only Memory — память только для чтения) — энергонезависимая память, используется для хранения данных, которые никогда не потребуют изменения. Содержание памяти специальным образом “зашивается” в устройстве при его изготовлении для постоянного хранения. Из ПЗУ можно только читать. Перепрограммируемая постоянная память (Flash Memory) — энергонезависимая память, допускающая многократную перезапись своего содержимого с дискеты.

Прежде всего, в постоянную память записывают программу управления работой самого процессора. В ПЗУ находятся программы управления дисплеем,

клавиатурой, принтером, внешней памятью, программы запуска и остановки компьютера, тестирования устройств.

Важнейшая микросхема постоянной или Flash-памяти — модуль BIOS. BIOS (Basic Input/Output System — базовая система ввода-вывода) — совокупность программ, предназначенных для:

- автоматического тестирования устройств после включения питания компьютера;
- загрузки операционной системы в оперативную память.

Роль BIOS двоякая: с одной стороны, это неотъемлемый элемент аппаратуры (Hardware), а с другой стороны — важный модуль любой операционной системы (Software).

CMOS RAM — это память с невысоким быстродействием и минимальным энергопотреблением от батарейки. Используется для хранения информации о конфигурации и составе оборудования компьютера, а также о режимах его работы. Содержимое CMOS изменяется специальной программой Setup, находящейся в BIOS (англ. Set-up — устанавливать, читается "сетап").

Для хранения графической информации используется видеопамять. Видеопамять (VRAM) — разновидность оперативного ЗУ, в котором хранятся закодированные изображения. Это ЗУ организовано так, что его содержимое доступно сразу двум устройствам — процессору и дисплею. Поэтому изображение на экране меняется одновременно с обновлением видеоданных в памяти.

Внешняя память (ВЗУ) предназначена для длительного хранения программ и данных, и целостность её содержимого не зависит от того, включен или выключен компьютер. В отличие от оперативной памяти, внешняя память не имеет прямой связи с процессором. В состав внешней памяти компьютера входят:

- накопители на жёстких магнитных дисках;
- накопители на гибких магнитных дисках;
- накопители на компакт-дисках;

- накопители на магнитооптических компакт-дисках;
- накопители на магнитной ленте (стримеры) и др.

Гибкий диск, дискета (англ. floppy disk) — устройство для хранения небольших объёмов информации, представляющее собой гибкий пластиковый диск в защитной оболочке. Используется для переноса данных с одного компьютера на другой и для распространения программного обеспечения. Дискета состоит из круглой полимерной подложки, покрытой с обеих сторон магнитным окислом и помещенной в пластиковую упаковку, на внутреннюю поверхность которой нанесено очищающее покрытие. В упаковке сделаны с двух сторон радиальные прорезы, через которые головки считывания/записи накопителя получают доступ к диску.

Способ записи двоичной информации на магнитной среде называется магнитным кодированием. Он заключается в том, что магнитные домены в среде выстраиваются вдоль дорожек в направлении приложенного магнитного поля своими северными и южными полюсами. Обычно устанавливается однозначное соответствие между двоичной информацией и ориентацией магнитных доменов. Информация записывается по концентрическим дорожкам (трекам), которые делятся на секторы. Количество дорожек и секторов зависит от типа и формата дискеты. Сектор хранит минимальную порцию информации, которая может быть записана на диск или считана. Ёмкость сектора постоянна и составляет 512 байтов.

На дискете можно хранить от 360 Килобайт до 2,88 Мегабайт информации. В настоящее время наибольшее распространение получили дискеты со следующими характеристиками: диаметр 3,5 дюйма (89 мм), ёмкость 1,44 Мбайт, число дорожек — 80, количество секторов на дорожках — 18.

Дискета устанавливается в накопитель на гибких магнитных дисках (англ. floppy-disk drive), автоматически в нем фиксируется, после чего механизм накопителя раскручивается до частоты вращения 360 мин^{-1} . В накопителе вращается сама дискета, магнитные головки остаются неподвижными. Дискета

вращается только при обращении к ней. Накопитель связан с процессором через контроллер гибких дисков.

Если гибкие диски — это средство переноса данных между компьютерами, то жесткий диск — информационный склад компьютера. Накопитель на жёстких магнитных дисках (англ. HDD — Hard Disk Drive), или винчестерский накопитель — это наиболее массовое запоминающее устройство большой ёмкости, в котором носителями информации являются круглые алюминиевые пластины — платтеры, обе поверхности которых покрыты слоем магнитного материала. Используется для постоянного хранения информации — программ и данных.



Рис. 1. Жесткий диск со снятой крышкой корпуса.

Как и у дискеты, рабочие поверхности платтеров разделены на кольцевые концентрические дорожки, а дорожки — на секторы. Головки считывания-записи вместе с их несущей конструкцией и дисками заключены в герметически закрытый корпус, называемый модулем данных. При установке модуля данных на дисковод он автоматически соединяется с системой, подкачивающей очищенный охлажденный воздух.

Поверхность платтера имеет магнитное покрытие толщиной всего лишь в 1,1 мкм, а также слой смазки для предохранения головки от повреждения при опускании и подъёме на ходу. При вращении платтера над ним образуется воздушный слой, который обеспечивает воздушную подушку для зависания головки на высоте 0,5 мкм над поверхностью диска.

Винчестерские накопители имеют очень большую ёмкость: от сотен Мегабайт до десятков Гбайт или даже сотни Гбайт. У современных моделей скорость вращения шпинделя достигает 5600 — 7200 оборотов в минуту, среднее время поиска данных — 10 мс, максимальная скорость передачи

данных до 40 Мбайт/с. В отличие от дискеты, винчестерский диск вращается непрерывно. Винчестерский накопитель связан с процессором через контроллер жесткого диска. Все современные накопители снабжаются встроенным кэшем (64 Кбайт и более), который существенно повышает их производительность.



Рис. 2. Накопители на компакт-дисках.

CD-ROM состоит из прозрачной полимерной основы диаметром 12 см и толщиной 1,2 мм. Одна сторона покрыта тонким алюминиевым слоем, защищенным от повреждений слоем лака. Двоичная информация представляется последовательным чередованием углублений (pits — ямки) и основного слоя (land — земля). На одном дюйме (2,54 см) по радиусу диска размещается 16 тысяч дорожек с информацией. Для сравнения — на дюйме по радиусу дискеты всего лишь 96 дорожек. Ёмкость CD до 780 Мбайт. Информация заносится на диск на заводе и не может быть изменена.

Достоинства CD-ROM:

- При малых физических размерах CD-ROM обладают высокой информационной ёмкостью, что позволяет использовать их в справочных системах и в учебных комплексах с богатым иллюстративным материалом; один CD, имея размеры примерно дискеты, по информационному объёму равен почти 500 таким дискетам.
- Считывание информации с CD происходит с высокой скоростью, сравнимой со скоростью работы винчестера.
- CD просты и удобны в работе, практически не изнашиваются.
- CD не могут быть поражены вирусами.
- На CD-ROM невозможно случайно стереть информацию.
- Стоимость хранения данных (в расчете на 1 Мбайт) низкая.

В отличие от магнитных дисков, компакт-диски имеют не множество кольцевых дорожек, а одну — спиральную, как у грампластинок. В связи с этим, угловая скорость вращения диска не постоянна. Она линейно уменьшается в процессе продвижения читающей магнитной головки к центру диска.

Для работы с CD ROM нужно подключить к компьютеру накопитель CD-ROM (CD-ROM Drive), в котором компакт-диски сменяются как в обычном проигрывателе. Накопители CD-ROM часто называют проигрывателями CD-ROM или приводами CD-ROM.

Участки CD, на которых записаны символы "0" и "1", отличаются коэффициентом отражения лазерного луча, посылаемого накопителем CD-ROM. Эти отличия улавливаются фотоэлементом, и общий сигнал преобразуется в соответствующую последовательность нулей и единиц. Многие накопители CD-ROM способны воспроизводить обычные аудио-CD. Это позволяет пользователю, работающему за компьютером, слушать музыку в фоновом режиме.

Есть CD-RW для записи на специальные компакт диски CD-R от 650 – 700 Мб и CD-RW для неоднократной записи емкостью от 650 – 700 Мб. Со временем на смену CD-ROM могут прийти цифровые видеодиски DVD (читается "ди-ви-ди"). Эти диски имеют тот же размер, что и обычные CD, но вмещают до 28 Гбайт данных, т. е. по объёму заменяют семь и более стандартных дисков CD-ROM. В скором времени ёмкость дисков DVD возрастет до 48 Гбайт. DVD диски бывают 1, 2 и 4-х слойные, для проигрывания DVD дисков необходим DVD-ROM.

Накопитель на магнитооптических компакт-дисках CD-MO (Compact Disk-Magneto Optical). Диски CD-MO можно многократно использовать для записи, но они не читаются на традиционных дисководы CD-ROM. Ёмкость от 128 Мбайт до 2,6 Гбайт. Записывающий накопитель CD-R (Compact Disk Recordable) способен, наряду с прочтением обычных компакт-дисков, записывать информацию на специальные оптические диски. Ёмкость 650

Мбайт. Накопитель WARM (Write And Read Many times), позволяет производить многократную запись и считывание. Накопитель WORM (Write Once, Read Many times), позволяет производить однократную запись и многократное считывание. Накопитель ZIP и JAZZ на дисках емкостью от 100 Mb до 2,2 Gb.

Стример (англ. tape streamer) — устройство для резервного копирования больших объёмов информации. В качестве носителя здесь применяются кассеты с магнитной лентой ёмкостью 1 – 2 Гбайта и больше. Стримеры позволяют записать на небольшую кассету с магнитной лентой огромное количество информации. Встроенные в стример средства аппаратного сжатия позволяют автоматически уплотнять информацию перед её записью и восстанавливать после считывания, что увеличивает объём сохраняемой информации. Недостатком стримеров является их сравнительно низкая скорость записи, поиска и считывания информации.

В последнее время всё шире используются накопители на сменных дисках, которые позволяют не только увеличивать объём хранимой информации, но и переносить информацию между компьютерами. Объём сменных дисков — от сотен Мбайт до нескольких Гигабайт.

Учебно-методическое и информационное обеспечение

Основная литература:

- 1 Гаврилов, М. В. Информатика и информационные технологии [Электронный ресурс]: учебник для СПО / М. В. Гаврилов, В. А. Климов. — 4-е изд., перераб. и доп. — М. : Издательство Юрайт, 2017. — 383 с. — Режим доступа: <https://www.biblio-online.ru>.
- 2 Илюшечкин, В. М. Основы использования и проектирования баз данных [Электронный ресурс]: учебник для СПО / В. М. Илюшечкин. — испр. и доп. — М. : Издательство Юрайт, 2017. — 213 с. — Режим доступа: <https://www.biblio-online.ru>.

- 3 Казанский, А. А. Объектно-ориентированный анализ и программирование на visual basic 2013 [Электронный ресурс]: учебник для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 290 с. — Режим доступа: <https://www.biblio-online.ru>.
- 4 Казанский, А. А. Прикладное программирование на excel 2013 [Электронный ресурс]: учебное пособие для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 159 с. — Режим доступа: <https://www.biblio-online.ru>.
- 5 Казанский, А. А. Программирование на visual c# 2013 [Электронный ресурс]: учебное пособие для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 191 с. — Режим доступа: <https://www.biblio-online.ru>.
- 6 Семакин И.Г. Основы алгоритмизации и программирования: учеб. для сред. проф.образования/ И.Г.Семакин, А.П. Шестаков. – 3-е изд., стер. – М.: Академия, 2012. – 392 с.
- 7 Советов, Б. Я. Базы данных [Электронный ресурс]: учебник для СПО / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 2-е изд. — М. : Издательство Юрайт, 2017. — 463 с. — Режим доступа: <https://www.biblio-online.ru>.
- 8 Советов, Б. Я. Информационные технологии [Электронный ресурс]: учебник для СПО / Б. Я. Советов, В. В. Цехановский. — 6-е изд., перераб. и доп. — М. : Издательство Юрайт, 2017. — 261 с. — Режим доступа: <https://www.biblio-online.ru>.
- 9 Федорова Г. Н. Разработка и администрирование баз данных : учеб. для студ. учреждений сред. проф. образования / Г. Н. Федорова. - М.: Академия, 2015. - 313 с.
- 10 Цветкова, М.С. Информатика и ИКТ : учеб. для студентов учреждений сред. проф. образования / М. С. Цветкова, Л. С. Великович. - 8-е изд., стер. - М. : Академия, 2016. - 332, [1] с., [4] л. ил.

- 11 Черпаков, И. В. Основы программирования [Электронный ресурс]: учебник и практикум для СПО / И. В. Черпаков. — М. : Издательство Юрайт, 2017. — 219 с. — Режим доступа: <https://www.biblio-online.ru>.

Тема 2.3 Элементная база ПК

План лекции

- 1 Логические функции и схемы ПК
- 2 Логические выражения и таблицы истинности

Логические функции и схемы ПК

Логика очень древняя наука. Ещё в античные времена была известна формальная логика, позволяющая делать заключения о правильности какого-либо суждения не по его фактическому содержанию, а только по форме его построения. Например, уже в древности был известен закон исключения третьего. Его содержательная трактовка была такова: «Во время своих странствований Платон был в Египте ИЛИ не был Платон в Египте». В такой форме это или любое другое выражение будут правильны (тогда говорили: истинно). Ничего другого быть не может: Платон либо был, либо не был в Египте - третьего не дано.

Другой закон логики - закон непротиворечивости. Если сказать: «Во время своих странствий Платон был в Египте И не был Платон в Египте», то очевидно, любое высказывание, имеющее такую форму, всегда будет ложно. Если из теории следуют два противоречащих друг другу вывода, то такая теория безусловно неправильная (ложная) и должна быть отвергнута.

Ещё один закон, известный в древности - закон отрицания: «Если НЕ верно, что Платон НЕ был в Египте, то значит, Платон был в Египте».

Формальная логика основана на “высказываниях”. “Высказывание” - это основной элемент логики, определяемый как повествовательное предложение, относительно которого можно однозначно сказать, истинное или ложное утверждение оно содержит.

Например: Листья на деревьях опадают осенью. Земля прямоугольная. Первое высказывание содержит истинную информацию, а второе - ложную. Вопросительное, побудительное и восклицательное предложения не являются высказываниями, так как в них ничего не утверждается и не отрицается.

Пример предложений, не являющихся высказываниями: Не пейте сырую воду! Кто не хочет быть счастливым?

Высказывания могут быть и такими: $2 > 1$, $H_2O + SO_3 = H_2SO_4$. Здесь используются языки математических символов и химических формул.

Приведённые выше примеры высказываний являются простыми. Но из простых высказываний можно получить сложные, объединив их с помощью логических связок. Логические связки - это слова, которые подразумевают определённые логические связи между высказываниями. Основные логические связки издавна употребляются не только в научном языке, но и в быденном, - это “и”, “или”, “не”, “если ... то”, “либо ... либо” и другие известные нам из русского языка связки. В рассмотренных нами трёх законах формальной логики использовались связки “и”, “или”, “не”, “если ... то” для связи простых высказываний в сложные.

Высказывания бывают общими, частными и единичными. Общее высказывание начинается со слов: всё, все, всякий, каждый, ни один. Частное высказывание начинается со слов: некоторые, большинство и т.п. Во всех других случаях высказывание является единичным.

Формальная логика была известна в средневековой Европе, она развивалась и обогащалась новыми законами и правилами, но при этом вплоть до 19 века она оставалась обобщением конкретных содержательных данных и её законы сохраняли форму высказываний на разговорном языке.

В 1847 году английский математик Джордж Буль, преподаватель провинциального университета в маленьком городке Корке на юге Англии разработал алгебру логики.

Алгебра логики очень проста, так как каждая переменная может принимать только два значения: истинно или ложно. Трудность изучения алгебры логики возникает из-за того, что для обозначения переменных принимают символы 0 и 1, которые по написанию совпадают с обычными арифметическими единицей и

нулём. Но совпадение это только внешнее, так как смысл они имеют совсем иной.

Логическая 1 означает, что какое-то событие истинно, в противоположность этому логический 0 означает, что высказывание не соответствует истине, т.е. ложно. Высказывание заменилось на логическое выражение, которое строится из логических переменных ($A, B, X, \dots$) и логических операций (связок).

В алгебре логики знаки операций обозначают лишь три логические связки ИЛИ, И, НЕ.

Логические выражения и таблицы истинности

1. Логическая операция ИЛИ (Дизъюнкция). Логическую функцию принято задавать в виде таблицы. В левой части этой таблицы перечисляются все возможные значения аргументов функции, т.е. входные величины, а в правой указывается соответствующее им значение логической функции. Для элементарных функций получается таблица истинности данной логической операции. Для операции ИЛИ таблица истинности имеет вид:

A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

Операцию ИЛИ называют также логическим сложением, и потому её можно обозначать знаком «+».

Рассмотрим сложное единичное высказывание: «Летом я поеду в деревню или в туристическую поездку». Обозначим через A простое высказывание «Летом я поеду в деревню», а через B - простое высказывание «Летом я поеду в туристическую поездку». Тогда логическое выражение сложного высказывания имеет вид $A+B$, и оно будет ложным только, если ни одно из простых высказываний не будет истинным.

2. Логическая операция И (Конъюнкция). Таблица истинности для этой функции имеет вид:

A	B	A&B
0	0	0
0	1	0
1	0	0
1	1	1

Из таблицы истинности следует, что операция И - это логическое умножение, которое ничем не отличается от традиционно известного умножения в обычной алгебре. Операцию И можно обозначить знаком по-разному:

$$Y=A\wedge B \quad Y=A\&B \quad Y=A*B \quad Y=AB$$

В формальной логике операции логического умножения соответствуют связки и, а, но, хотя.

3. Логическая операция НЕ. Эта операция является специфичной для алгебры логики и не имеет аналога в обычной алгебре. Она обозначается чертой над значением переменной, либо знаком приставки перед значением переменной:

$$\overline{A} \text{ или } \neg A$$

Читается в обоих случаях одинаково «Не А». Таблица истинности для этой функции имеет вид:

A	$\neg A$
0	1
1	0

В вычислительной технике операцию НЕ называют отрицанием или инверсией, операцию ИЛИ - дизъюнкцией, операцию И - конъюнкцией. Набор логических функций “И”, “ИЛИ”, “НЕ” является функционально полным набором или базисом алгебры логики. С помощью него можно выразить любые другие логические функции, например операции “строгой дизъюнкции”, “импликации” и “эквивалентности” и др. Рассмотрим некоторые из них.

Логическая операция “строгая дизъюнкция”. Этой логической операции соответствует логическая связка “либо ... либо”. Таблица истинности для этой функции имеет вид:

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Операция “строгая дизъюнкция” выражается через логические функции “И”, “ИЛИ”, “НЕ” любой из двух логических формул:

$$A \oplus B = A\bar{B} + \bar{A}B \text{ или } A \oplus B = (\bar{A} + \bar{B})(A + B)$$

и иначе называется операцией неравнозначности или “сложения по модулю 2”, так как при сложении чётного количества единиц, результатом будет “0”, а при сложении нечётного числа единиц, результат станет равен “1”.

Логическая операция “импликация”. Выражение, начинающееся со слов если, когда, коль скоро и продолжающееся словами то, тогда, называется условным высказыванием или операцией «импликация». Таблица истинности для этой функции имеет вид:

A	B	$A \rightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

Операцию “импликация” можно обозначить по-разному:

$$Y = A \rightarrow B \text{ и } Y = A \supset B$$

Эти выражения эквивалентны и читаются одинаково: «Игрек равен импликации от А и В». Операция “импликация” выражается через логические функции “ИЛИ”, “НЕ” в виде логической формулы

$$A \rightarrow B = \bar{A} + B \text{ или } \neg(A \rightarrow B) = A * \bar{B}$$

Логическая операция “эквивалентность” (равнозначность). Этой логической операции соответствуют логические связки “если и только если”, «тогда и только тогда, когда». Таблица истинности для этой функции имеет вид:

A	B	$A \sim B$
0	0	1
0	1	0
1	0	0
1	1	1

Операция “эквивалентность” обозначается по-разному. Выражения

$$A \sim B, A \leftrightarrow B \text{ или } A \equiv B$$

обозначают одно и то же, и можно сказать, что A эквивалентна B, если и только если они равнозначны. Логическая операция “эквивалентность” выражается через логические функции “И”, “ИЛИ”, “НЕ” в виде логической формулы

$$A \leftrightarrow B = A \cdot B + \bar{A} \cdot \bar{B} \text{ или } A \leftrightarrow B = (\bar{A} + B) * (A + \bar{B})$$

С помощью алгебры логики можно очень кратко записать законы формальной логики и дать им математически строгое доказательство.

1. Закон исключения третьего можно записать в виде

В ИЛИ (НЕ В)=ИСТИННО или ещё короче: $B \vee \neg B = 1$.

2. Закон непротиворечивости записан в виде

В И (НЕ В)=ЛОЖНО, а короче: $B \wedge \neg B = 0$.

3. Закон отрицания отрицания представляет собой одно

из основных выражений алгебры логики: $\overline{\overline{B}} = B$.

В алгебре логики, как в элементарной, справедливы переместительный (закон коммутативности), сочетательный (закон ассоциативности) и распределительный (закон дистрибутивности) законы, а также аксиома идемпотентности (отсутствие степеней и коэффициентов) и др., в записях которых используются логические переменные, принимающие только два значения - логический ноль и логическая единица. Применение этих законов позволяет производить упрощение логических функций, т.е. находить для них выражения, имеющие наиболее простую форму. Основные аксиомы и законы алгебры логики приведены в таблице:

Аксиомы: свойства констант 0 и 1:	$1+A=1$ $0*A=0$ $0+A=A$ $1*A=A$
идемпотентность:	$A+A=A$ $A*A=A$
Закон исключения третьего:	$A+\neg A=1$
Закон непротиворечивости:	$A*\neg A=0$
Закон отрицания:	$\neg(\neg A)=A$
Законы коммутативности:	$A+B=B+A$ $A*B=B*A$
Законы ассоциативности:	$A+B+C=A+(B+C)$ $A*B*C=A*(B*C)$
Законы дистрибутивности:	$A*(B+C)=A*B+A*C$ $A+(B*C)=(A+B)*(A+C)$
Законы де Моргана:	$\neg(A+B)=\neg A*\neg B$ $\neg(A*B)=\neg A+\neg B$
Законы поглощения:	$A+A*B=A$ $A*(A+B)=A$

Примеры использования основных аксиом и законов:

$$1) \bar{A} + A*B = (\bar{A} + A)*(\bar{A} + B) = \bar{A} + B$$

при преобразовании логической функции использовались законы дистрибутивности, исключения третьего и четвёртое свойство констант.

$$2) (A+B+B*C)AB = (A+B)AB = AAB + BAB = AB + AB = AB$$

при преобразовании данной логической функции использовались закон поглощения, первый закон дистрибутивности и две аксиомы идемпотентности.

Учебно-методическое и информационное обеспечение

Основная литература:

- 1 Гаврилов, М. В. Информатика и информационные технологии [Электронный ресурс]: учебник для СПО / М. В. Гаврилов, В. А. Климов. — 4-е изд., перераб. и доп. — М. : Издательство Юрайт, 2017. — 383 с. — Режим доступа: <https://www.biblio-online.ru>.
- 2 Илюшечкин, В. М. Основы использования и проектирования баз данных [Электронный ресурс]: учебник для СПО / В. М. Илюшечкин. — испр. и доп. — М. : Издательство Юрайт, 2017. — 213 с. — Режим доступа: <https://www.biblio-online.ru>.

- 3 Казанский, А. А. Объектно-ориентированный анализ и программирование на visual basic 2013 [Электронный ресурс]: учебник для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 290 с. — Режим доступа: <https://www.biblio-online.ru>.
- 4 Казанский, А. А. Прикладное программирование на excel 2013 [Электронный ресурс]: учебное пособие для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 159 с. — Режим доступа: <https://www.biblio-online.ru>.
- 5 Казанский, А. А. Программирование на visual c# 2013 [Электронный ресурс]: учебное пособие для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 191 с. — Режим доступа: <https://www.biblio-online.ru>.
- 6 Семакин И.Г. Основы алгоритмизации и программирования: учеб. для сред. проф.образования/ И.Г.Семакин, А.П. Шестаков. – 3-е изд., стер. – М.: Академия, 2012. – 392 с.
- 7 Советов, Б. Я. Базы данных [Электронный ресурс]: учебник для СПО / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 2-е изд. — М. : Издательство Юрайт, 2017. — 463 с. — Режим доступа: <https://www.biblio-online.ru>.
- 8 Советов, Б. Я. Информационные технологии [Электронный ресурс]: учебник для СПО / Б. Я. Советов, В. В. Цехановский. — 6-е изд., перераб. и доп. — М. : Издательство Юрайт, 2017. — 261 с. — Режим доступа: <https://www.biblio-online.ru>.
- 9 Федорова Г. Н. Разработка и администрирование баз данных : учеб. для студ. учреждений сред. проф. образования / Г. Н. Федорова. - М.: Академия, 2015. - 313 с.
- 10 Цветкова, М.С. Информатика и ИКТ : учеб. для студентов учреждений сред. проф. образования / М. С. Цветкова, Л. С. Великович. - 8-е изд., стер. - М. : Академия, 2016. - 332, [1] с., [4] л. ил.

- 11 Черпаков, И. В. Основы программирования [Электронный ресурс]: учебник и практикум для СПО / И. В. Черпаков. — М. : Издательство Юрайт, 2017. — 219 с. — Режим доступа: <https://www.biblio-online.ru>.

РАЗДЕЛ 3 ТЕХНОЛОГИИ СОЗДАНИЯ И ПРЕОБРАЗОВАНИЯ ИНФОРМАЦИОННЫХ ОБЪЕКТОВ

Тема 3.1 Текстовые редакторы

План лекции

- 1 Технологии обработки текстовой информации
- 2 Текстовый процессор Microsoft Word
- 3 Использование шаблонов документов и других средств, повышающих эффективность работы с текстом
- 4 Программы верстки оригинал-макетов

Технологии обработки текстовой информации

Редакторы текстов программ рассчитаны на редактирование программ на том или ином языке программирования. Редакторы текста, и рассчитаны на тексты программ, и выполняют следующие функции:

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

Редакторы документов — программы для обработки документов, ориентированные на работу с текстами, имеющие структуру документа, т. е.

состоящими из разделов, страниц, абзацев, предложений, слов и т.д. Следовательно, редакторы для обработки документов обеспечивают такие функции, как:

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

Редакторы текстов программ, можно использовать для создания и корректировки небольших документов. Однако при необходимости серьезной работы с документами лучше использовать редакторы, ориентированные на работу с документами.

Современные текстовые процессоры предоставляют пользователю широкие возможности по подготовке документов. Это и функции редактирования, допускающие возможность любого изменения, вставки, замены, копирования и перемещения фрагментов в рамках одного документа и между различными документами, контекстного поиска, функции форматирования символов, абзацев, страниц, разделов документа, верстки, проверки грамматики и орфографии, использования наряду с простыми текстовыми элементами списков, таблиц, рисунков, графиков и диаграмм.

Значительное сокращение времени подготовки документов обеспечивают такие средства автоматизации набора текста, как автотекст и автозамена, использование форм, шаблонов и мастеров типовых документов.

Наличие внешней памяти компьютера обеспечивает удобное длительное хранение подготовленных ранее документов, быстрый доступ к ним в любое время.

Существенно упрощают процедуру ввода данных сканеры и голосовые устройства. Существующие системы распознавания текстов, принимаемых со сканера, включают функцию экспорта документа в текстовые редакторы.

Широкий спектр печатающих устройств в сочетании с функциями подготовки документа к печати, предварительного просмотра, обеспечивает получение высококачественных черно-белых и цветных копий на бумаге и прозрачной пленке.

Таким образом, современные программы предусматривают множество функций, позволяющих готовить текстовую часть документа на типографическом уровне. Кроме того, современные программы позволяют включать в текст графические объекты: рисунки, диаграммы, фотографии. Благодаря этим возможностям файл, представляющий собой текстовый документ, может содержать, помимо алфавитно-цифровых символов, обширную двоичную информацию о форматировании текста, а также графические объекты.

Форматы текстовых файлов

ТХТ

Один из первых (если не первый) текстовых форматов, широко используется по сей день. Так как текст хранится в виде последовательности символов, то размер файла в байтах равен числу символов плюс непечатаемые символы (знак пробела, табуляции, знак конца абзаца и другие - их еще называют знаками форматирования). За счет этого достигается малый размер файла. Однако возможности по форматированию подобных документов сильно ограничены. По сути - это просто текст. Текстовые данные могут храниться не только в контейнерах с расширением ТХТ, это расширение не является

обязательными и служит для облегчения выбора программы, которая открывает файл по умолчанию.

RTF

Свободный межплатформенный формат хранения размеченных текстовых документов, созданный Microsoft в 1987 году. Ныне он широко распространен, поэтому большинство современных текстовых редакторов его поддерживают. Создав RTF на платформе Windows, он прекрасно будет читаться, и редактироваться на других платформах (Apple, Linux и другие). Стандарт де-факто в полиграфии.

RTF позволяет производить и сохранять достаточно сложное форматирование, вставлять сноски, колонтитулы, рисунки, таблицы и формулы, хотя в этом он все же уступает формату DOC. Уступает он DOC и в объеме файлов: сложные документы более компактно хранятся в DOC-файлах (простые - наоборот). Однако RTF выигрывает спор с DOC в отношении безопасности, так как не использует макросы. Поэтому зараженные макровирусами файлы Word можно «вылечить» сохранением в RTF-формат. Кроме того, формат RTF устойчив к повреждению файла. Если в DOC-файле изменить хотя бы один байт, то он уже не откроется в Word. А повреждение файла в формате RTF может привести только к потере поврежденного куска текста.

DOC

Сначала это расширение использовалось для обозначения простых текстовых файлов без форматирования, однако в начале 90-х Microsoft фактически его «приватизировала». Поэтому сейчас DOC ассоциируется только с продуктами этой компании. Этот формат обеспечивает большие возможности по форматированию текста (включены сценарии, макросы). За счет этого ухудшилась совместимость с текстовыми редакторами сторонних разработчиков. В файле этого формата содержится огромное количество

информации о шрифтах, начертании символов, абзацных отступах и интервалах, даже если все это вам совсем не нужно. Именно из-за этой дополнительной информации файл, содержащий только текст, превышает размер RTF-файла. Однако при включении в документ различных графических элементов и изображений DOC выигрывает в размере и обеспечивает большую совместимость. В отличие от TXT и RTF DOC является бинарным форматом, что делает его нечитабельным в простых текстовых редакторах. К примеру, «Блокнот» может просматривать некоторые RTF-файлы. Популярен наравне с RTF.

DOCX

С появлением Office 2007 компания Microsoft перешла на новые форматы, базирующиеся на Office Open XML (визуально отличаются тем, что к расширениям добавлена буква «x» на конце). Формат представляет собой zip-архив, содержащий текст в виде XML, графику и другие данные. Для уменьшения размера файла используется ZIP-компрессия. Документы обратно совместимы с Office 2000/XP/2003, только если установлен Microsoft Office Compatibility Pack. Если вы пользуетесь последней версией Office и планируете передавать файлы кому-либо, сохраняйте документы в RTF или DOC.

ODT/ODF

ODF - общее наименование открытого формата документов для офисных приложений (текст, таблицы, рисунки, базы данных, презентации). Текстовые данные хранятся в файлах с расширением ODT. Стандарт был разработан индустриальным сообществом OASIS и основан на XML-формате. 1 мая 2006 года принят как международный стандарт ISO/IEC 26300. Федеральное Агентство по Техническому Регулированию и Метрологии, входящее в состав Министерства Промышленности и Торговли Российской Федерации, издало приказ об утверждении формата OpenDocument (ODF) в качестве национального стандарта. (Национальный стандарт ГОСТ Р ИСО/МЭК 26300-

2010 «Информационная технология. Формат Open Document для офисных приложений (OpenDocument) v1.0», идентичный международному стандарту ISO/IEC 26300:2006). Введен в действие - 1 июня 2011 года. ODF доступен для всех и может быть использован без ограничений. Эдакая бесплатная альтернатива закрытым форматам Microsoft. Для того чтобы производить чтение и запись формата ODF в продуктах Microsoft, был выпущен плагин Sun ODF Plugin for Microsoft Office. Поддержка ODF в Microsoft Office 2007 была введена с выпуском с Service Pack 2. К сожалению, по-прежнему уступает по распространенности RTF и DOC.

HTML

Стандартный язык разметки документов в интернете (расширение .htm/html). Веб-страницы создаются при помощи языка HTML (или XHTML). HTML был разработан британским ученым Тимом Бернерсом-Ли в 1991 году как язык для обмена научной и технической документацией, пригодный для использования людьми, не являющимися специалистами в области верстки. Текст с разметкой HTML должен был без стилистических и структурных искажений воспроизводиться на различных устройствах. Однако позже активное внедрение мультимедийного и графического оформления на-рушило эти планы. Для просмотра HTML-документов не нужны специальные редакторы, достаточно стандартных средств, встроенных в ОС. По открытости, индексируемости, конвертируемости и читаемости превосходит любые другие форматы. К сожалению, графика обычно сохраняется в отдельной папке.

PDF

Кроссплатформенный формат электронных документов, созданный фирмой Adobe Systems с использованием ряда возможностей языка PostScript. В первую очередь предназначен для представления в электронном виде полиграфической продукции. Для просмотра можно использовать официальную бесплатную программу Adobe Reader, а также программы других

разработчиков. Удобен тем, что решена проблема со слетевшим форматированием, неправильным отображением внедренных графических элементов, отсутствием тех или иных шрифтов. Файл на любой платформе будет отображен в таком же виде, в каком он был создан. Традиционным способом создания PDF-документов является следующий: документ как таковой готовится в своей программе, а затем экспортируется в PDF. Некоторые программы имеют возможность для прямого экспорта (без использования виртуального принтера). Например, OpenOffice.org. Стандарт де-факто для большинства документации.

DjVu

Технология сжатия изображений с потерями, разработанная специально для хранения сканированных документов - книг, журналов, рукописей и пр., где наличие формул, схем, рисунков и рукописных символов делает чрезвычайно трудоемким их полноценное распознавание. Также является эффективным решением, если необходимо передать все нюансы оформления, например, исторических документов. Очень распространен, многие библиотеки используют его для хранения отсканированных научных книг. DjVu иногда называют «тексто-графическим» форматом. Суть технологии DjVu заключается в автоматическом разбиении изображения на несколько участков (например, текст, логотип фирмы и растровая фотография), для каждого из которых выбирается оптимальный алгоритм сжатия. Кроме того, DjVu-файл может содержать встроенное интерактивное оглавление и активные области - ссылки, что позволяет реализовывать удобную навигацию. Дает выигрыш в размере файла по сравнению с GIF-форматом в среднем в полтора-два десятка раз.

XML формат

Существует довольно много текстовых форматов, созданных для одного конкретного устройства или программы. Например, электронные книги. К ним можно отнести Rocket e-book (.rb), Microsoft Reader (.lit), PalmDoc, MobiPocket

(.pro) и т.д. Как правило, все они созданы с помощью языка XML. Самым удачным и самым распространенным из них является формат FictionBook (FB2). На данный момент это самый прогрессивный и перспективный формат для электронных книг. Единственный его недостаток - длительные временные затраты при подготовке начального текста. Что окупается удобством чтения. В FictionBook упор сделан на структурирование документа: с помощью тегов можно выделять различные области текста (главы, заголовки, цитаты, врезки). Как все будет выглядеть на экране, зависит от программы-ридера. Если требуется оформить документ определенным образом, имеется возможность присоединения таблицы стилей.

Текстовые редакторы

Текстовый редактор Word Pad

Редактор Word Pad – текстовый редактор, позволяющий создавать и редактировать достаточно длинные тексты.

Редактор Word Pad, входящий в состав Windows, не является достаточно мощным, но с большинством повседневных задач – с написанием письма, студенческого реферата, изготовлением поздравительной открытки – Word Pad справляется вполне успешно.

С помощью WordPad можно:

- работать со шрифтами, использовать разнообразное начертание и цвет шрифтов;
- сохранять тексты как в своем собственном формате, так и в других популярных форматах (в том числе в формате Microsoft Word);
- вставлять в текст картинки разнообразных форматов.

Текстовый редактор Microsoft Word 2003

Текстовый редактор Microsoft Word на сегодняшний день является одной из лучших профессиональных программ для обработки текста. Он также

представляет собой удобную издательскую систему для верстки и макетирования книг и периодических изданий.

Microsoft Office – пакет, предназначенный для выполнения различных операций с документами. В отличие от других аналогичных по функциям программ приложения, входящие в Microsoft Office, интегрированы в единую систему, что обеспечивает их эффективную работу с документами, включающими разные по типу элементы (например, документ Word может содержать таблицу Excel и часть базы данных Access).

Современные текстовые редакторы могут выполнять следующие функции:

- запоминать и вставлять символы или фрагменты текста;
- использовать большое количество различных шрифтов
- автоматически переносит по слогам слова, не уместающиеся в строке;
- проверять и исправлять орфографические и синтаксические ошибки;
- заменять повторяющиеся слова синонимами;
- вставлять иллюстрации, включая фотографии;
- автоматически заменять символ, слово или фрагмент во всем тексте;
- вставлять таблицы и диаграммы;
- автоматически нумеровать страницы;
- автоматически создавать сноски, алфавитный указатель и оглавление;
- форматировать текст в виде газетных колонок;
- устанавливать размер печатной страницы, полей и отступов и т.д.

В различных редакторах предусмотрены и другие возможности, в частности и встроенный язык программирования, позволяющий автоматизировать многие операции. Любой текстовый редактор позволяет сохранить информацию в виде документа и распечатать ее на бумаге, однако Word умеет гораздо больше. Word – многофункциональная программа для редактирования текстов и графики, создания веб-страниц и обработки документов, в которые, кроме того, легко вставить даже аудио- и

видеофрагменты. С помощью Word можно создавать практически любые документы, публиковать их в электронном виде или выводить на печать в виде бумажных копий.

Работа с Microsoft Word 2003

Теперь подробно рассмотрим некоторые команды меню:

1. Меню «Файл»

Создать - Открывает область задач «Создание документа»

Открыть - Открывает окно диалога «Открытие документа»

Заккрыть - Закрывает документ без прекращения работы текстового редактора Word

Сохранить - Сохраняет документ под тем же именем и в том же формате

Сохранить как - Сохраняет документ под готовым именем или в новом формате

Сохранить как Web – страницу - Сохраняет документ для просмотра в программе Internet Explorer

Версии - Сохраняет версии документа в этом или в новом файле

Пред. просмотр Web – страницы - Запускает программу Internet Explorer и загружает в ее окно текущий документ

Параметры страницы - Открывает окно диалога, в котором можно изменить параметры страницы

Предварительный просмотр - Показывает, как будет выглядеть документ в напечатанном виде

Печать - Открывает окно диалога для установления параметров печати открытого документа

Отправить - Открывает дополнительное меню, которое используется при работе в Сети

Свойства - Открывает окно диалога, в котором можно ознакомиться со свойствами документа и изменить некоторые из них

(Список документов в нижней части меню) - Открывает документ, по имени которого щелкнули

Выход - Завершает работу текстового редактора Word

2. Меню «Правка»

Отменить - Отменяет сделанные изменения в документе

Повторить - Повторяет сделанные изменения в документе

Вырезать - Удаляет вырезанный фрагмент в буфер обмена

Копировать - Копирует выделенный фрагмент в буфер обмена

Буфер обмена - Открывает область задач «Буфер обмена»

Вставить - Вставляет содержимое буфера обмена в то место документа, в котором находится текстовый курсор

Специальная вставка - Вставляет содержимое буфера обмена в разных форматах

Вставить как гиперссылку - Вставляет содержимое буфера обмена как гиперссылку на документ или часть документа, откуда был скопирован фрагмент

Очистить - Открывает дополнительное меню:

Форматы - приводит форматирование документа к стандартному;

Содержимое – по действию аналогично нажатию на клавишу Delete

Выделить все - Выделяет весь документ

Найти - Открывает окно диалога для поиска слова или фрагмента текста в документе

Заменить - Открывает окно диалога для замены слова или фрагмента текста в документе

Перейти - Открывает окно диалога для быстрого перехода к нужному месту (например, к странице) в документе

Связи - Открывает окно диалога, с помощью которого можно изменять связи данного документа с другими

Объект - Открывает дополнительное меню:

Изменить – редактирование объекта в окне Word;

Открыть – открытие в отдельном окне программы, в которой был создан объект, с одновременной его загрузкой;

Преобразовать – преобразование объекта

3. Меню «Вид»

Обычный - Обычный режим просмотра документа

Web – документ - Просмотр документа как Web – страницы

Разметка страницы - Показывает документ так, как он будет напечатан: с номерами страниц, колонтитулами, сносками, графикой

Структура - Показывает только заголовки заданного уровня

Область задач - Открывает ту область задач, которая была закрыта последней

Панели инструментов - Открывает список доступных панелей инструментов

Линейка - Открывает или скрывает горизонтальную и вертикальную линейки форматирования

Схема документа - Открывает в левой части экрана дополнительное окно с названиями заголовков всех уровней, имеющихся в документе

Колонтитулы - Переводит текстовый курсор в область верхних и нижних колонтитулов

Сноски - Открывает окно с текстами всех сносок, имеющихся в документе

Разметка - Показывает все примечания и записанные исправления, сделанные в документе

Во весь экран - Убирает с экрана строку меню, панели инструментов, линейки, полосы прокрутки, чтобы текст документа занимал весь экран

Масштаб - Открывает окно диалога, с помощью которого можно изменять масштаб документа

4. Меню «Вставка»

Разрыв - Открывает окно диалога для принудительного начала новой страницы или нового раздела в документе

Номера страниц - Открывает окно диалога для вставки в документ номеров страниц

Дата и время - Открывает окно диалога, с помощью которого можно вставить в документ текущую дату или время в одном из возможных форматов

Автотекст - Открывает дополнительное меню со списком слов и предложений, которые могут быть вставлены в документ автоматически

Поле - Открывает поле диалога для вставки в документ автоматически обновляемых полей

Символ - Открывает окно диалога для вставки в документов символов, которых нет на клавиатуре

Примечание - Открывает область проверки, в которую можно записывать изменения и примечания в документе

Ссылка - Открывает дополнительное меню:

Сноска – открывает окно диалога для вставки в документ сносок или для их изменения;

Название – открывает окно диалога, с помощью которого можно задавать название автоматически изменяемых номеров таблиц, схем, рисунков и т.д.;

Перекрестная ссылка – открывает окно диалога для записи ссылки на другую часть документа или на другой документ;

Оглавление и указатели – открывает окно диалога для вставки в документ оглавления или указателей (предметных, именных, рисунков и т.д.)

Рисунок - Открывает дополнительное меню для выбора источников рисунков

Организационная диаграмма - Открывает окно диалога, с помощью которого в документ вставляют заполненные организационные диаграммы

Файл - Открывает окно диалога для вставки в документ другого файла

Объект - Открывает окно диалога для выбора объекта, вставляемого в документ. Перечень объектов зависит от программ, установленных на компьютере

Закладка - Вставляет закладку в документ

Гиперссылка - Вставляет гиперссылку на другой документ, файл или фрагмент данного документа

3. Пользовательский интерфейс Word 2003

При запуске текстового редактора Word так выглядит основное окно

Строка состояния полоса прокрутки

4. Общий вид окна Word и основные приемы работы с документами

Первая строка основного окна, являющаяся строкой заголовка, содержит название пакета – Microsoft Word.

Строка меню, начинающаяся со слова “Файл”, содержит названия главных режимов меню, каждый из которых в свою очередь раскрывается в список команд. По умолчанию при создании нового документа панель инструментов всегда изображается на экране, но при желании панель инструментов можно убрать с экрана. На экране можно отразить ту или иную панель в зависимости от потребности.

Ниже панели инструментов располагается окно документа. В строке заголовок документа выводится имя документа, точнее, имя файла, в котором он находится. По умолчанию новым документам присваиваются имена. При дальнейшей работе имена документов можно изменить.

Следующая строка называется линейкой. Эта строка позволяет увидеть и изменить положение отступов и табуляторов для текущего абзаца текста. Ниже линейки форматирования располагается рабочая зона, в которую вводятся текст, формулы, иллюстрации и другие объекты документа.

Предпоследняя строка экрана содержит линейку горизонтальной прокрутки. Она позволяет просматривать правую часть широкого документа, не поместившуюся в окне. Перемещение окна по горизонтали вдоль страницы документа осуществляется с помощью кнопок со стрелками и движка. Справа от этой линейки в той же строке экрана расположены кнопки для перелистывания страниц большого документа в обе стороны и установки первой или последней страницы.

В правой части экрана расположена линейка вертикальной прокрутки для просмотра длинных документов.

В последнюю строку экрана – строку состояния выводится вспомогательная информация об активном документе (номер текущей страницы и их общее количество).

Для запуска необходимо щелкнуть по пиктограмме данного приложения. Сразу после запуска на экране откроется окно с пустым документом. При создании нового документа в Word мы видим пустое окно для ввода.

Для ввода текста достаточно подвести курсор к нужному месту документа и набрать текст.

В Word можно работать с несколькими документами одновременно. Каждый создаваемый или открываемый документ размещается на экране в своем собственном окне.

Одно из преимуществ работы с несколькими документами одновременно заключается в легкости копирования или перемещения текста из одного в другое, что упрощает создание двух различных версий одного базового документа или заимствования из существующего документа при построении нового.

4 Создание и редактирование документов

Текстовые редакторы можно разделить на группы по их мощности и возможностям:

- простые текстовые редакторы (Windows Блокнот) – редактирование текста, простейшее форматирование шрифта;
- текстовые процессоры (Microsoft Word) – больше возможностей по созданию документов (вставка списков и таблиц, средства проверки орфографии, сохранение исправлений и другое);
- настольные издательские системы (Adobe PageMaker) – макетирование книг, журналов и газет.

Помимо этого существуют приложения для подготовки Web-страниц и сайтов (Microsoft FrontPage, Macromedia Dreamweaver).

Рассмотрим основные функции текстовых редакторов на примере приложения Microsoft Word.

1 этап.

Создание документа с использованием шаблона.

Запускаем редактор Word. Выполняем команду [Файл-Создать...]. На появившейся панели Создание выбираем Шаблон – пустую заготовку документа определенного назначения (обычный документ, визитная карточка, Web-страница). Шаблоны задают структуру документа, которую пользователь заполняет определенным содержанием, например, вводит символы с клавиатуры.

Давайте введем в документ какой-нибудь текст.

Для создания документов со сложной структурой используют Мастера.

2 этап.

Ознакомимся со свойствами документа.

Командой [Файл-Свойства] вызываем диалоговую панель для текущего документа Свойства: Документ.doc. На вкладке Статистика расположена информация о составе документа (количество страниц, абзацев, строк, слов, символов и другое).

Если вы хотите посмотреть свойства другого документа, его надо сначала открыть командой [Файл-Открыть...].

3 этап.

Редактирование документа Поиск и замена.

Производя копирование, перемещение (фрагмент вырезается и вставляется в другое место документа) и удаление, мы редактируем документ.

Чтобы найти и заменить неоднократно встречающееся слово на другое, можно использовать функцию редактирования Найти и заменить. Командой [Правка-Заменить...] вызываем диалоговую панель Найти и заменить. На вкладке Заменить в поле Найти: вводим последовательность символов, которую необходимо найти. Причем можно использовать звездочку (*) для маскировки произвольного числа символов или знак вопроса (?), который заменяет любой

единичный символ. (Это используется, например, если искомое слово в тексте содержится с разными окончаниями). В поле Заменить на: также вводим соответствующую последовательность символов.

А теперь замените в вашем документе все буквы «а» на «о».

4 этап.

Вставка объектов в документ.

В редакторе Word в документ можно встроить изображения, анимацию, звук и даже видеофрагменты. Это возможно благодаря механизму встраивания и внедрения объектов OLE (с ним вы уже знакомились). Этот механизм реализует копирование и вставку объекта из одного приложения в другое.

Вставим в наш документ рисунок.

Для этого вводим команду [Вставка-Объект...], появится диалоговая панель Вставка объекта. На вкладке Создание в списке Тип объекта: выберем пункт Точечный рисунок, после чего загрузится графический редактор Paint, в котором можно создать требуемый рисунок.

Если вы хотите вставить в документ готовый рисунок, то на вкладке Создание из файла с помощью кнопки Обзор выберите требуемый файл.

5 этап.

Вставка фигурного текста в документ.

Фигурный текст называется объектом WordArt

Добавление объекта WordArt

1. На панели инструментов Рисование нажмите кнопку Добавить объект WordArt .

2. Выберите нужный объект WordArt, а затем нажмите кнопку ОК.

3. В диалоговом окне Изменение текста WordArt введите нужный текст.

Выполните одно из следующих действий:

чтобы изменить тип шрифта, выберите нужный шрифт в поле Шрифт;

чтобы изменить размер шрифта, выберите размер в поле Размер;

чтобы сделать текст полужирным, нажмите кнопку Полужирный;

чтобы сделать текст наклонным, нажмите кнопку Курсив

6 этап.**Вставка автофигур в документ**

Текстовый редактор позволяет вставлять геометрические фигуры, которые называются автофигурами .

Добавление автофигуры

Выполните одно из следующих действий.

На панели инструментов Рисование нажмите кнопку Автофигуры, выберите категорию, а затем щелкните нужную фигуру.

Щелкните место в презентации, в которое нужно вставить автофигуру.

Добавление нескольких фигур

На панели инструментов Рисование нажмите кнопку Автофигуры, выберите категорию, наведите указатель на пунктирную линию и перетащите меню в нужное место.

Дважды щелкните автофигуру, которую нужно вставить несколько раз, и последовательно щелкните мышью те места презентации, где требуется вставить эту автофигуру.

Совет

Чтобы добавить круг или квадрат, нажмите на панели инструментов Рисование кнопку Овал или Прямоугольник . Для вставки нескольких таких фигур в презентацию, дважды щелкните одну из этих кнопок.

7 этап.**Проверка орфографии и синтаксиса. Сохранение исправлений.**

Для проверки правильности написания слов и построения предложений используются специальные программные модули, встроенные в текстовые процессоры и редакционно-издательские системы. Они содержат словари и грамматические правила для нескольких языков.

Если, набирая текст в редакторе Word, вы допустили орфографическую ошибку, она будет выделена красной волнистой линией, а синтаксическая ошибка выделится зеленой.

Если в процессе ввода текста, вы допустите опечатку (например, наберете ДВа вместо Два), сработает функция Автозамена, которая автоматически исправляет наиболее часто встречающиеся опечатки.

Когда вы закончили работать с документом или внесли какое-то исправление, его надо сохранить командами [Файл-Сохранить] или [Файл-Сохранить как...].

Во втором случае появится панель Сохранение документа. В ней вы можете выбрать место на компьютере, куда надо сохранить файл. В поле Имя файла: вводите название документа, которое хотите ему присвоить, а в раскрывающемся списке Тип файла: выбираете требуемый формат.

О форматах мы поговорим позднее.

8 этап.

Вставка колонтитулов

Колонтитулами называют области, расположенные в верхнем и нижнем поле каждой страницы документа.

В колонтитул обычно вставляется текст и/или рисунок (номер страницы, дата печати документа, эмблема организации, название документа, имя файла, фамилия автора и т. п.), который должен быть напечатан внизу или вверху каждой страницы документа.

Чтобы начать работу с колонтитулом, выберите в меню Вид команду Колонтитулы.

Вставка колонтитулов

Создание одинакового верхнего или нижнего колонтитула на каждой странице

1. Выберите в меню Вид команду Колонтитулы, чтобы открыть на странице область верхнего или нижнего колонтитула
2. Чтобы создать верхний колонтитул, введите текст или графический элемент в область верхнего колонтитула.

3. Чтобы создать нижний колонтитул, на панели инструментов Колонтитулы нажмите кнопку Верхний/нижний колонтитул для перемещения к нижнему колонтитулу, а затем введите текст или графический элемент.

4. При необходимости форматируйте текст с помощью кнопок, расположенных на панели инструментов Форматирование.

5. После завершения создания колонтитулов нажмите кнопку Закреть на панели инструментов Колонтитулы.

Показать Советы

По умолчанию содержимое колонтитула автоматически выравнивается по левому краю. Может потребоваться выровнять содержимое колонтитула по центру или включить несколько элементов (например, дату с выравниванием по левому краю и номер страницы с выравниванием по правому краю). Для выравнивания элемента по центру нажмите клавишу TAB; для выравнивания элемента по правому краю нажмите клавишу TAB дважды.

· Для ввода текста в область колонтитулов также можно использовать кнопки панели инструментов Колонтитулы.

Показать Создание уникального колонтитула для первой страницы

На первой странице документа можно не использовать колонтитул или создать уникальный колонтитул для первой страницы документа или первой страницы каждого раздела в документе.

1. Если документ разбит на разделы, щелкните в нужном разделе или выделите несколько разделов, которые нужно изменить. Если документ не разбит на разделы, щелкните в любом месте документа.

2. В меню Вид выберите команду Колонтитулы.

3. На панели инструментов Колонтитулы нажмите кнопку Параметры страницы.

4. Откройте вкладку Макет.

5. Установите флажок Различать колонтитулы первой страницы и нажмите кнопку ОК.

6. Если нужно, на панели инструментов Колонтитулы используйте соответственно кнопку Переход к предыдущему или Переход к следующему, чтобы перейти в область Верхний колонтитул первой страницы или в область Нижний колонтитул первой страницы.

7. Создайте верхний или нижний колонтитул для первой страницы документа или раздела.

Если требуется, чтобы на первой странице отсутствовал колонтитул, оставьте поле колонтитула пустым.

Чтобы задать колонтитулы для остальных страниц документа или раздела, на панели инструментов Нумерация страниц

В Microsoft Word можно нумеровать страницы двумя способами. При каждом способе нумерации номера страниц отображаются в колонтитулах в верхней и нижней части страницы.

Добавление номеров страниц в колонтитулы

1. В меню Вставка выберите команду Номера страниц.
2. В поле Положение укажите расположение номеров страниц (внизу или вверху страницы).
3. В поле Выравнивание укажите вид выравнивания (по левому краю, по правому краю, по центру, внутри или снаружи страницы).
4. Если на первой странице номер не нужен, снимите флажок Номер на первой странице.

Выберите другие нужные параметры.

Добавление номеров страниц с дополнительными сведениями, такими как дата и время

1. В меню Вид выберите команду Колонтитулы.
2. Чтобы разместить номера страниц внизу страницы, нажмите кнопку Верхний/нижний колонтитул на панели инструментов Колонтитулы.
3. На панели инструментов Колонтитулы нажмите кнопку Номер страницыЗначок кнопки.
4. Выберите другие нужные параметры.

Примечание. Номер страницы автоматически вставляется в левое поле колонтитула. Чтобы переместить его в центр или вправо, щелкните кнопкой мыши перед номером страницы в режиме Колонтитулы и нажмите клавишу TAB.

1. Колонтитулы нажмите кнопку Переход к следующему, а затем создайте нужный колонтитул.

9 этап.

Печать файла.

Для начала надо посмотреть, как будет выглядеть документ на бумаге, так как его вид зависит от принтера.

Командой [Файл-Предварительный просмотр] задаем режим Предварительного просмотра. Если вы хотите посмотреть сразу несколько страниц одновременно на экране, то надо изменить масштаб просмотра документа командой [Вид-Масштаб...].

Если во время просмотра вы хотите иметь возможность форматировать и редактировать документ, то перед запуском режима Предварительный просмотр необходимо выбрать режим Разметка страницы командой [Вид-Разметка страницы].

Выйти из просмотра можно, нажав клавишу {Esc}.

Если же вы хотите вывести документ на печать, то командой [Файл-Печать] необходимо вызвать диалоговую панель Печать, которая позволяет установить параметры печати: выбрать принтер, число копий и номера страниц, выводимых на печать.

5 Проверка правописания

Представляет собой компьютерную программу, осуществляющую проверку заданного текста на предмет наличия в нём орфографических, пунктуационных, а также стиливых ошибок. Найденные ошибки или опечатки отмечаются специальным образом — обычно для этого используется подчеркивание. В некоторых случаях пользователю помимо указания на места

возможных ошибок предоставляется возможность выбрать один из правильных вариантов написания, а также может выводиться комментарий, объясняющий, каким образом следует поправить текст.

Проверка правописания может быть встроена как отдельная функция в некую программную систему, например, текстовый редактор, почтовый клиент, электронный словарь или поисковую систему. А также она может быть выполнена в виде самостоятельной программы. В этом случае она обычно обладает возможностью интеграции с другими приложениями. Такими возможностями, например, обладает GNU Aspell для Unix-подобных операционных систем, а также кроссплатформенная Hunspell.

6 Форматирование текста. Форматирование символов

Форматирование символов заключается в задании их шрифта/ размера и начертания. Если всем ясно что такое размер (Рис. 1, в центре), то о шрифте и начертании следует рассказать более подробно.



Рис. 1 Примеры форматирования символов

Шрифтом называется ранее разработанный набор изображений всех букв алфавита, цифр, знаков препинания и специальных символов. Одни и те же символы, отформатированные с использованием различных шрифтов, могут сильно отличаться

друг от друга (Рис. 1, слева). Вместе с Windows поставляется несколько Шрифтов кроме того имеется возможность установки в Систему Дополнительных шрифтов широко представленных на рынке.

Символы отформатированные с помощью одного шрифта могут иметь различные размеры и начертание. Под начертанием понимается внешний вид символов: обычный полужирный курсив (то есть наклонный) или

подчеркнутый (Рис. 1, справа). При этом возможно сочетание нескольких видов начертания.

Проще всего выполнить форматирование символов с помощью управляющих элементов расположенных на панели форматирования. Для этого используются три элемента: список шрифтов список размеров и кнопки выбора начертания. В последних версиях программ вместо кнопок на панели инструментов расположены только рисунки а обрамляющие рамки появляются только тогда, когда к рисунку подводится указатель мыши. Но мы все равно, для простоты будем называть их кнопками.

Список шрифтов выглядит как название шрифта на панели форматирования Шрифт с названием Times New Roman входит в состав Windows. Возможно у вас выбран другой шрифт но в любом случае вы можете найти на панели инструментов элемент с названием шрифта. Подведите указатель мыши к значку расположенному в правой части списка шрифтов после чего нажмите и отпустите левую кнопку мыши то есть щелкните мышью на кнопке. Ниже появится список установленных у вас. шрифтов (Рис. 2). Если вы щелкнете мышью на названии одного из шрифтов то он будет назначен выделенному фрагменту текста. Однако в Windows бывает установлено достаточно много шрифтов и они могут не поместиться в появившейся рамке. Чтобы просмотреть все варианты следует воспользоваться специальной полосой расположенной в правой части списка. Сверху и снизу полосы имеются кнопки. Они предназначены для передвижения по списку шрифтов. Нажмите несколько раз кнопку и вы передвинетесь вниз по списку шрифтов, а нажатие кнопки позволит передвинуться вверх. Найдя нужный шрифт следует щелкнуть на его названии мышью, и он будет выбран.



Рис. 2 Выбор шрифтов

Выбор нужного шрифта является важным делом, и мы остановимся на нем подробнее. Рядом с названиями большинства шрифтов можно увидеть значок или . Эти значки обозначают, что данный шрифт будет одинаково хорошо выглядеть на экране и при печати на принтере. Кроме того, такие шрифты позволяют выбрать любой размер, сохраняя при этом высокое качество. Мы рекомендуем использовать только шрифты, отмеченные значками или .

Текстовый редактор WordPad добавляет к названию шрифта особенности выбранного языка (Кириллица или Cyrillic). В списке , расположенном в панели форматирования вы можете выбрать нужный язык для выбранного шрифта.

Какие же шрифты следует использовать при создании документов? Тут нет точных рекомендаций, однако чаще всего для форматирования заголовков, надписей и выделения небольших фрагментов используется шрифт Arial, а для форматирования основного текста используется Times New Roman. Если вам нужно, чтобы документ напоминал напечатанный на пишущей машинке, следует использовать шрифт Courier New. Для оригинальных надписей можно рекомендовать шрифт Comic Sans Ms.

Выбор размера символов очень похож на выбор шрифта. Правее списка шрифтов в панели инструментов находится список размеров . Нажмите на

кнопку , и рядом появится перечень возможных размеров. Точно так же, как в списке шрифтов вы можете выбрать любой размер символов.

Но что означает десятый размер шрифта? Много это или мало? В текстовых редакторах используется типографская единица измерения, называемая пунктами. Один пункт равен примерно 0,353 миллиметра, соответственно 10 пунктов равно примерно трем с половиной миллиметрам. Высота прописных букв шрифта равна приблизительно трем четвертям указанного размера, а строчных - немного более половины. Для большинства документов используется шрифт размером 12 пунктов. Многостраничные документы, книги и брошюры лучше форматировать шрифтом десятого или одиннадцатого размера.

Чтобы часть текста отличалась от остального часто используют различное начертание. Для изменения начертания используют три кнопки на панели форматирования: кнопка  или в английской версии используется для выбора полужирного начертания, кнопка  , для использования курсива и для подчеркивания. После того, как вы нажмете нужную кнопку, она останется в нажатом состоянии. Повторное нажатие отменит выбранное начертание, и кнопка будет отжата. Если вы одновременно нажмете две или все три кнопки, то будет выбрана комбинация начертания.

После того, как вы познакомились с различными вариантами форматирования символов, надо поэкспериментировать с изменением форматирования введенного текста. Выделите весь текст и выберите для него шрифт Times New Roman размером 12. После этого выделите несколько слов и выберите шрифт Arial. Нажмите кнопки  и  , чтобы слова изменили начертание на полужирный курсив. Попробуйте самостоятельно изменить форматирование различных фрагментов текста.

Рассмотрим еще один способ форматирования символов. Установите текстовый курсор в середине текста. При этом ни один фрагмент текста не выделен. Выберите шрифт Courier New размером 24. Внешне ничего не изменилось, но текстовый редактор изменил форматирование для вновь

вводимых символов. Введите несколько ело», и вы увидите, что новые символы будут отформатированы указанным шрифтом

Форматируются только выделенные символы. Если в тексте не выделен фрагмент, после изменения форматирования, по-новому будут выглядеть все вновь введенные символы.

Форматирование символов позволяет значительно украсить документ, однако придать ему законченный вид позволит форматирование абзацев.

7 Форматирование абзацев

Если вам понадобится поместить заголовок документа в центре строки, что вы сделаете? Добавьте в начало строки множество пробелов? Этого делать не надо ни в коем случае! Следует просто установить выравнивание по центру для абзаца, в котором содержится заголовок. Что такое выравнивание, можно понять из примера (Рис. 1).

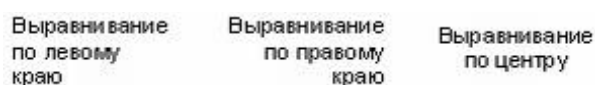


Рис 1. Разные варианты выравнивания

Чтобы выполнить форматирование одного абзаца, достаточно установить в нем текстовый курсор. Выделять абзац не обязательно. Если же вы хотите изменить форматирование нескольких абзацев то следует выделить их. Также требуется выделить весь текст чтобы изменить форматирование всех абзацев документа. После того, как вы установили курсор в нужном абзаце или выделили несколько абзацев, можно изменить форматирование с помощью кнопок на панели форматирования.

Для назначения определенного выравнивания используют кнопки: для выравнивания по левому краю, для выравнивания по правому краю, для выравнивания по центру. Рисунки на кнопках показывают, как будет выравниваться текст и их легко запомнить. К сожалению, в текстовом редакторе WordPad отсутствует возможность выравнивания по обоим краям, она имеется только в более мощных редакторах, таких как, например. Word.

Возвращаясь к созданию заголовков, давайте попробуем это сделать. Установите курсор в начало текста введите несколько слов заголовка и нажмите клавишу чтобы выделить заголовок в отдельный абзац. Снова установите курсор в начало текста после чего нажмите кнопку на панели инструментов. Заголовок будет расположен точно по центру документа.

Выделите все оставшиеся абзацы и нажмите кнопку С помощью выравнивания удобно делать подписи к письмам.

Установите курсор в самый конец текста нажмите клавишу , чтобы создать новый абзац и введите в нем свое имя. После этого нажмите кнопку и имя разместится в правой части документа.

Кроме выравнивания вы можете задать дополнительные отступы текста от края документа для любого абзаца настроив их в специальном диалоге. Выберите команду меню Формат * Абзац (Format * Para graph). На экране появится диалог настройки параметров абзаца (Рис. 2).

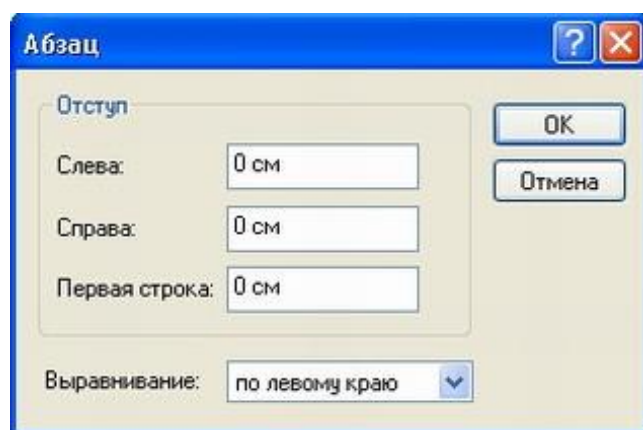


Рис. 2. Настройка отступов и выравнивания

Список, расположенный в нижней части диалога, позволяет выбрать выравнивание, однако для этой цели удобнее использовать кнопки на панели форматирования. В группе полей Отступ (Indentation) задаются отступы текста от края. Вы можете ввести отступы справа и слева. Кроме того, можно задать отступ первой строки, чтобы получить эффект, называемый красной строкой.

При задании отступов слева и справа, а также отступа первой строки, вы можете получить примерно такое форматирование текста.

Выравнивание по левому краю вместе с отступом слева часто используется в шапке документа.

Задав все параметры, нажмите кнопку ОК. Диалог закроется, и форматирование выделенных абзацев будет изменено. Если до вызова диалога не было выделено ни одного абзаца, изменится форматирование абзаца, в котором находился текстовый курсор.

Изменить отступы можно также с помощью специальных значков, расположенных на линейке (Рис. 3). В левой части линейки находится маркер , отмечающий отступ слева, и маркер , отмечающий отступ первой строки. Маркер размещающийся справа, определяет отступ справа. Установите указатель мыши на любом маркере, нажмите кнопку мыши и, не отпуская ее, передвиньте маркер вправо или влево, после чего отпустите кнопку мыши. Вместе с изменением позиции маркера поменяются и отступы выделенного абзаца. Выделяя различные абзацы и перемещая маркеры отступа, вы легко сможете создать нужное оформление документа.



Рис. 2. Линейка с маркерами и символами табуляции

Щелкнув мышью в любом месте линейки, вы добавите символ табуляции. Табуляция может помочь при необходимости вводить слова в точно отведенные позиции, например при создании списков и таблиц (Рис. 4). После ввода текста в первый столбец вы нажимаете клавишу и текстовый курсор устанавливается в позицию, отмеченную символом табуляции. Вы вводите текст во второй столбец и снова нажимаете клавишу. Для перехода к новой строке следует начать новый абзац нажав клавишу, после чего следует аналогично ввести текст в столбцы, используя табуляцию.

Первый столбец	Второй столбец	Третий столбец
12134567	8912345	456123
951235	751248	568942
124578	325489	356874

Рис. 4 Использование табуляции

Символы табуляции также можно перетащить с помощью мыши. Если при перетаскивании символа будет выделено несколько абзацев, то в этих абзацах изменится расположение текста. Помните, что позиции табуляции связаны с конкретным абзацем, и в другом абзаце может быть установлена совсем другая табуляция. Для точного задания позиций табуляции следует выбрать команду меню Формат * Табуляция (Format * Tabs). На экране появится диалог настройки табуляции (Рис. 5). В списке представлены все имеющиеся позиции табуляции. Введя новое Значение в поле ввода и нажав кнопку Установить (Set), вы добавите еще одну позицию табуляции. Для удаления позиции следует выделить ее, щелкнув мышью на нужном элементе списка, после чего нажать кнопку Очистить (Clear). Чтобы удалить всю табуляцию из текущего абзаца, нажмите кнопку Очистить все (Clear All). После того как все нужные позиции табуляции установлены, нажмите кнопку ОК. Диалог закроется, а на линейке будут отмечены новые позиции табуляции. Использование табуляции существенно облегчает создание разнообразных таблиц. Советуем не забывать об этой удобной возможности редактора WordPad.

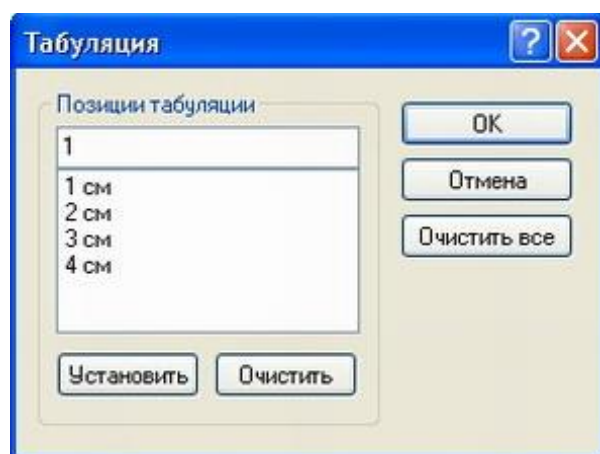


Рис. 5 Настройка табуляции

Для создания различных списков часто используются специальные символы, называемые маркерами, которые располагаются в начале списка (Рис.

6). Чаще всего в качестве маркеров используются небольшие черные кружки и текстовый редактор WordPad позволяет очень просто создавать такие списки.

Я люблю фрукты:

- Яблоки,
- Груши,
- Апельсины.

Рис. 6 Использование маркеров в списке

Установите курсор в конце одного из абзацев текста и нажмите клавишу, чтобы создать новый абзац. Нажмите кнопку, расположенную на панели форматирования, и в начале абзаца появится маркер. Теперь введите несколько названий, расположив каждое из них в отдельном абзаце. Все вновь введенные абзацы будут помечены маркерами. Чтобы убрать маркер из любого абзаца, требуется установить курсор в этот абзац и повторно нажать кнопку.

Выбрав необходимый шрифт, его размер и начертание, а также применив к абзаца выравнивание, установив отступы и добавив в списки маркеры, вы отформатировали свой текст. С помощью описанных возможностей легко создать красивый профессионально оформленный документ. Хотя мощные текстовые редакторы, такие как Word предлагают несравненно больше вариантов форматирования символов и абзацев, для работы с простыми документами вполне хватит описанных нами возможностей.

8 Списки

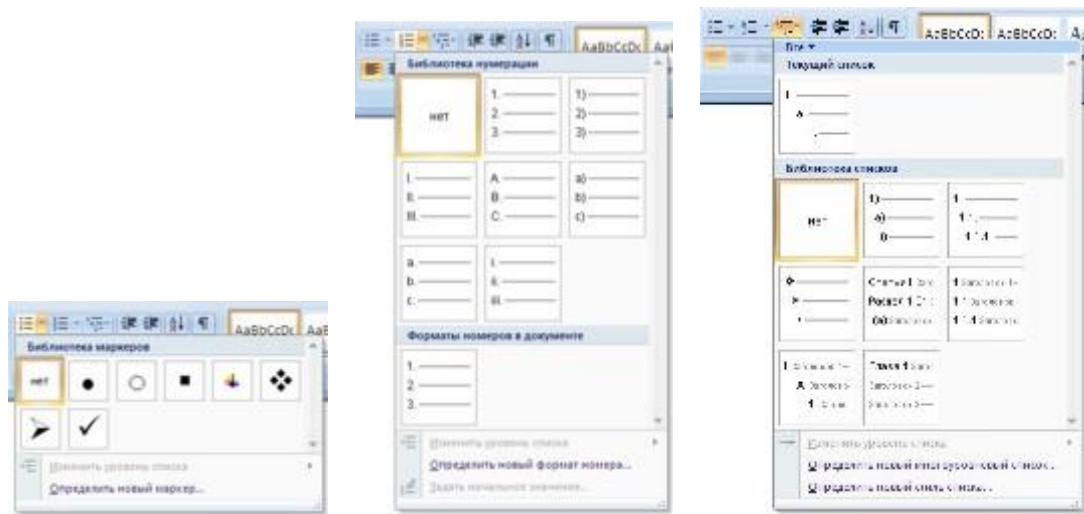
Работа со списками Word 2007

Для работы со списками служат пять верхних кнопок панели "Абзац".



Списки - это фрагменты текста, пункты которого отмечены

специальными знаками. Списки могут быть маркированными, нумерованными и многоуровневыми.



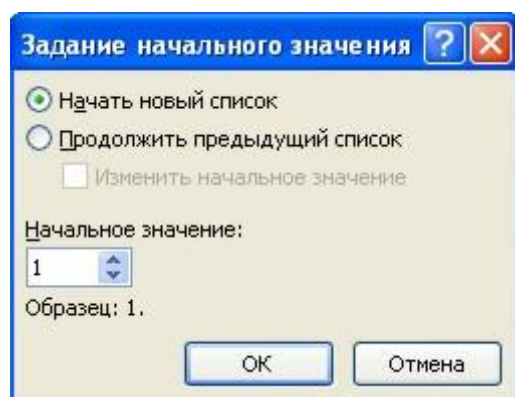
Список можно создавать изначально, а можно из уже существующего текста.

Если необходимо сделать список из уже существующего документа, то надо выделить фрагмент текста, который подлежит форматированию и выбрать тип списка. При этом выделенный текст будет разбит по пунктам списка согласно абзацам (каждый абзац - это новый пункт списка). Во время выбора типа списка при наведении курсора на соответствующий вариант выделенный текст будет сразу предварительно форматироваться, давая пользователю быстро оценить пригодность того или иного варианта.

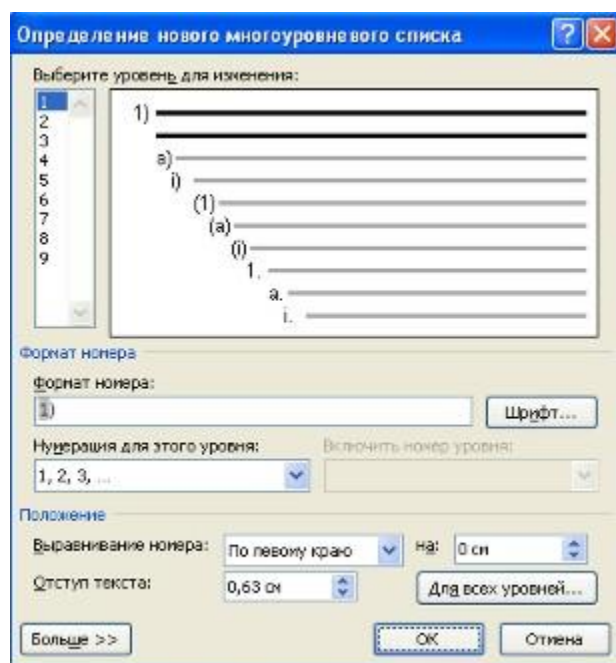
При формировании многоуровневого списка, чтобы задать создание маркеров очередного уровня можно использовать клавишу Tab (либо кнопку "Увеличить отступ" на панели "Абзац"). Вернуться к вводу данных предыдущего уровня можно, нажав сочетание Shift+Tab (либо кнопку "Уменьшить отступ" на панели "Абзац").

При работе с маркированными и нумерованными списками можно создавать свой стиль оформления. Для этого нужно в соответствующих диалоговых окнах (см. выше) выбрать пункт "Определить новый маркер" или "Определить новый формат номера".

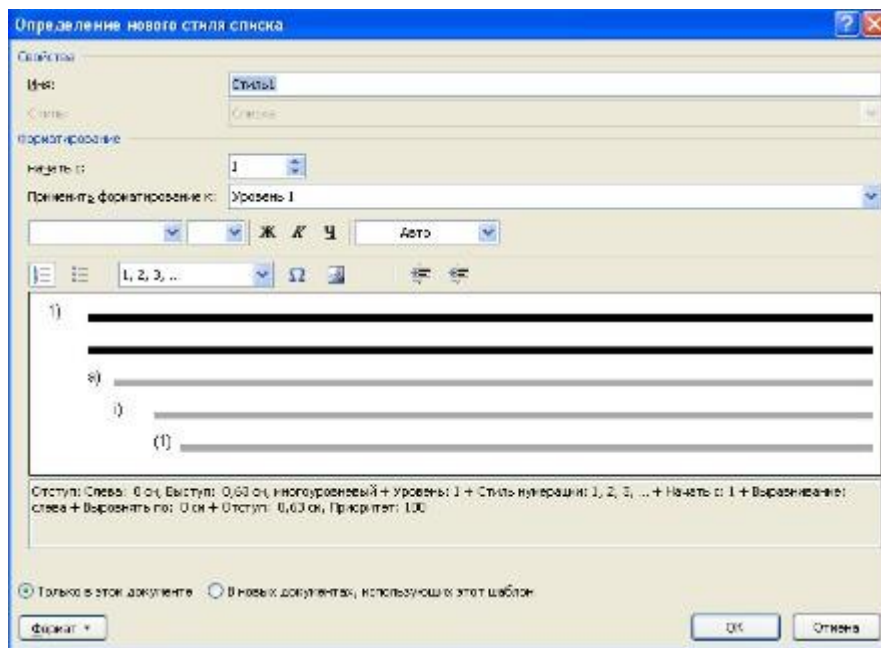
Иногда бывает необходимо в нумерованном списке начать список не с первого номера. Для этой цели служит пункт "Задать начальное значение". В появившемся окне в зависимости от поставленной задачи надо установить переключатель в одно из двух положений: "Начать новый список" или "Продолжить предыдущий список". В поле "Начальное значение" задайте номер первого пункта списка.



При необходимости редактирования многоуровневого списка, щелкните кнопкой мыши на кнопке "Многоуровневый список" и в появившемся окне - "Определить новый многоуровневый список..". Здесь можно настроить формат номера, расстояние, тип шрифта и другие параметры списка.



Если необходимо сформировать новый стиль списка, то необходимо воспользоваться пунктом "Определить новый стиль списка". В появившемся окне можно настроить все необходимые параметры стиля, а также задать область действия нового формата.



Напоследок можно сказать, что Ворд автоматически создает новый нумерованный список, когда абзац начинается с цифры "один" с точкой.

На следующем занятии будет дано практическое применение использованию списочного форматирования.

9 Стили и шаблоны

Стиль форматирования — набор элементов форматирования (шрифта, абзаца и пр.), имеющий уникальное имя. Любой абзац документа Word оформлен определенным стилем, стандартным или пользовательским.

В Word существует четыре основных вида стилей:

- стили символов;
- стили абзацев;
- стили таблиц;

- **стили списков.**

Стилевое форматирование имеет ряд преимуществ перед ручным:

- *экономит время.* Применить стиль как набор элементов форматирования значительно быстрее, чем применять их один за другим.
- *способствует единообразию оформления документа.* При ручном форматировании одинаковые по смыслу форматирования разделы могут отличаться своими форматами, применение стиля же вносит строгость в оформление документа.
- *позволяет быстро изменить вид отдельных элементов во всем документе.* В этом случае достаточно внести изменения в стиль, и оформление вступит в силу во всем документе.

Стили абзацев обычно объединяют элементы форматирования символов и абзацев и определяют вид абзаца. При этом должны быть заданы все элементы форматирования для абзаца.

Стили символов содержат один или несколько элементов форматирования, не требуя полного определения всех элементов форматирования для символа.

К одному и тому же тексту может быть применен стиль абзаца, стиль символов и ручное форматирование. Они выстраиваются в определенную иерархию: ручное форматирование имеет преимущество перед символьным, а стиль символов имеет преимущество перед стилем абзаца.

Если стиль необходимо применить к одному абзацу, то достаточно установить курсор в любое место этого абзаца или выделить необходимый фрагмент.

Стиль можно применить одним из способов.

- Панель инструментов **Форматирование**, список **Стиль**, выбрать нужный стиль.
- Панель задач **Стили и форматирование**, выбрать нужный стиль (для отображения панели задач использовать меню **Формат ==> Стили и форматирование** (рис. 1)).

- Установить курсор в абзац-образец, щелкнуть кнопку  на панели инструментов **Стандартная**, применить стиль по образцу к нужному абзацу, переместившись к нему.

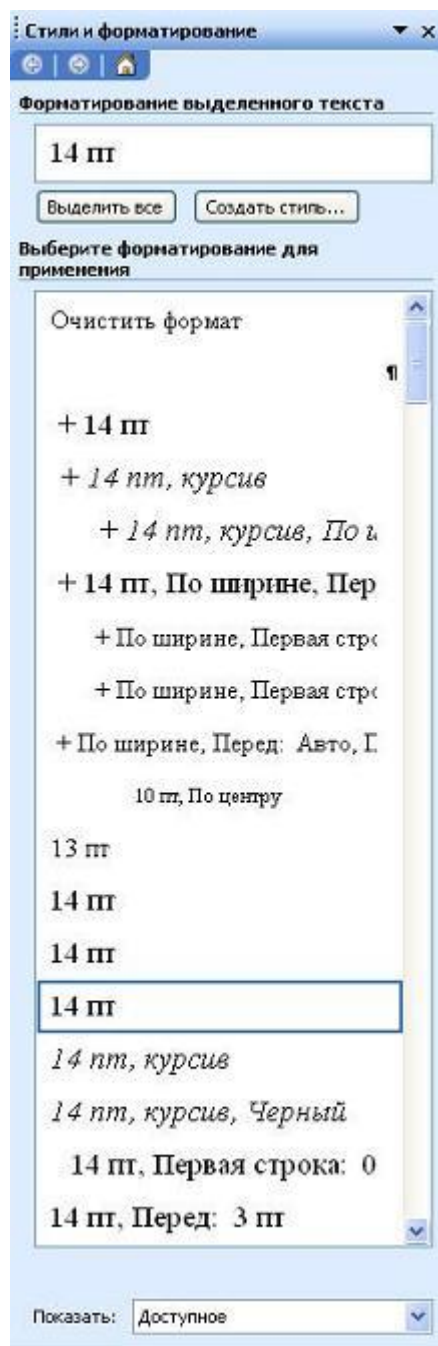


Рис. 1. Панель задач **Стили и форматирование**

Используя панель задач **Стили и форматирование**, можно выделить все фрагменты, имеющие одинаковый стиль форматирования. Это можно сделать, например, так. Установить текстовый курсор в абзац, имеющий нужное

форматирование (стиль отобразится в окне **Форматирование** выделенного текста панели задач), после чего выбрать на панели задач **Выделить все**, затем выполнить с выделенным текстом необходимое действие. Можно также использовать контекстное меню.

Учебно-методическое и информационное обеспечение

Основная литература:

- 1 Гаврилов, М. В. Информатика и информационные технологии [Электронный ресурс]: учебник для СПО / М. В. Гаврилов, В. А. Климов. — 4-е изд., перераб. и доп. — М. : Издательство Юрайт, 2017. — 383 с. — Режим доступа: <https://www.biblio-online.ru>.
- 2 Илюшечкин, В. М. Основы использования и проектирования баз данных [Электронный ресурс]: учебник для СПО / В. М. Илюшечкин. — испр. и доп. — М. : Издательство Юрайт, 2017. — 213 с. — Режим доступа: <https://www.biblio-online.ru>.
- 3 Казанский, А. А. Объектно-ориентированный анализ и программирование на visual basic 2013 [Электронный ресурс]: учебник для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 290 с. — Режим доступа: <https://www.biblio-online.ru>.
- 4 Казанский, А. А. Прикладное программирование на excel 2013 [Электронный ресурс]: учебное пособие для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 159 с. — Режим доступа: <https://www.biblio-online.ru>.
- 5 Казанский, А. А. Программирование на visual c# 2013 [Электронный ресурс]: учебное пособие для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 191 с. — Режим доступа: <https://www.biblio-online.ru>.
- 6 Семакин И.Г. Основы алгоритмизации и программирования: учеб. для сред. проф.образования/ И.Г.Семакин, А.П. Шестаков. – 3-е изд., стер. – М.: Академия, 2012. – 392 с.

- 7 Советов, Б. Я. Базы данных [Электронный ресурс]: учебник для СПО / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 2-е изд. — М. : Издательство Юрайт, 2017. — 463 с. — Режим доступа: <https://www.biblio-online.ru>.
- 8 Советов, Б. Я. Информационные технологии [Электронный ресурс]: учебник для СПО / Б. Я. Советов, В. В. Цехановский. — 6-е изд., перераб. и доп. — М. : Издательство Юрайт, 2017. — 261 с. — Режим доступа: <https://www.biblio-online.ru>.
- 9 Федорова Г. Н. Разработка и администрирование баз данных : учеб. для студ. учреждений сред. проф. образования / Г. Н. Федорова. - М.: Академия, 2015. - 313 с.
- 10 Цветкова, М.С. Информатика и ИКТ : учеб. для студентов учреждений сред. проф. образования / М. С. Цветкова, Л. С. Великович. - 8-е изд., стер. - М. : Академия, 2016. - 332, [1] с., [4] л. ил.
- 11 Черпаков, И. В. Основы программирования [Электронный ресурс]: учебник и практикум для СПО / И. В. Черпаков. — М. : Издательство Юрайт, 2017. — 219 с. — Режим доступа: <https://www.biblio-online.ru>.

Тема 3.2 Графика и звук

План лекции

- 1 Технологии обработки графической информации. Графика в профессии.
- 2 Видеомонтаж. Автоматизированное проектирование.
- 3 Звук. Синтезаторы звука на персональном компьютере.

Технологии обработки графической информации. Графика в профессии.

Все изображения, с которыми работают программы машинной графики, разделяются на два класса: растровые и векторные. В терминологии машинной графики растровые или точечным изображением принято называть массив пикселей — одинаковых по размеру и форме плоских геометрических фигур

(чаще всего — квадратов или кругов), расположенных в узлах регулярной (то есть состоящей из ячеек одинаковой формы и размера) сетки. Для каждого пиксела тем или иным способом задается цвет (обычно цвета кодируются числами фиксированной разрядности). **Представление растрового изображения в памяти компьютера - это массив сведений о цвете всех пикселов, упорядоченный определенным образом (например, по строкам, как в телевизионном изображении).** Наиболее близким аналогом такого явления виртуального мира, как пиксельное изображение, в реальном мире является мозаика. Пиксельное изображение состоит из равномерно расположенных на плоскости элементов одинакового размера и формы (пикселов), подобно мозаике, состоящей из кусочков цветного стекла — смальты. При соблюдении определенных условий (главные из них — небольшие размеры фрагментов смальты и достаточно большое удаление зрителя от поверхности изображения) отдельные кусочки смальты, составляющие мозаичное изображение, не видны — глаз зрителя воспринимает изображение как единое целое.

Векторным изображением в компьютерной графике принято называть совокупность более сложных и разнообразных геометрических объектов. Номенклатур таких объектов может быть более или менее широкой, но, как правило, в нее не включаются простейшие геометрические фигуры (круги, эллипсы, прямоугольники, многоугольники, отрезки прямых и дуги кривых линий). Важнейшая особенность векторной графики состоит в том, что для каждого объекта определяются управляющие параметры, конкретизирующие его внешний вид. Например, для окружности такими управляющими параметрами являются диаметр, цвет, тип и толщина линии, а также цвет внутренней области. Представление векторного изображения в памяти компьютера сложнее, чем пиксельного (хотя, как правило, при этом оно намного компактнее). Несколько упрощая, можно считать, что оно является перечнем всех объектов, из которых составлено изображение, причем для каждого объекта указано, к какому классу объектов он принадлежит, и

Приведены значения всех управляющих параметров. Подобрать аналог векторному изображению в реальном мире не так-то просто. Впрочем, на эту роль вполне может претендовать тот человечек, которого в детстве рисовали, наверное, все, приговаривая: *«Точка, точка, запятая, минус, рожица кривая, палка, палка, огуречик...»*. Последняя фраза, по сути дела, представляет собой перечисление объектов векторного изображения.

ПРИМЕЧАНИЕ Из-за того, что в этой фразе отсутствуют какие-либо намеки на значения управляющих параметров, человечки у всех получались разные.

Процесс вывода пиксельного изображения на экран или бумагу достаточно прост - на экране пикселу соответствует группа из трех частиц люминофора, светящихся различными цветами, принтер изображает пиксеты каплями чернил или пятнами тонера (*красящего порошка*). К устройствам, непосредственно фиксирующим векторные изображения, относятся, пожалуй, только довольно редко встречающиеся вне стен конструкторских бюро **графопостроители**. Почти всегда векторное изображение перед выводом (или непосредственно в процессе вывода) преобразуется в точечное — в компьютерной графике этот процесс называется рендерингом.

ПРИМЕЧАНИЕ 2. На основании приведенного выше описания двух классов изображений вы можете сделать вывод, что с векторными изображениями работать значительно сложнее и что область их применения весьма узкая. В очень многих случаях решающую роль играют специфические достоинства и недостатки пиксельных и векторных изображений.

Основной недостаток пиксельного изображения состоит в том, что размер пикселей является фиксированным. Из-за этого в случае изменения размера изображения возникают крайне нежелательные эффекты. При его увеличении между плотно «прижатыми» друг к другу пиксетами появляется свободное место. Заполнить его, строго говоря, нечем, разве что размещая на свободных местах копии находящихся рядом пикселей. Это эквивалентно увеличению размера пиксета при увеличении изображения. Однако сильно

увеличивать размер пиксела нельзя — слишком крупные пиксели перестанут восприниматься глазом зрителя как однородное изображение, видимость смыкания разрушится. Этот эффект хорошо известен профессиональным фотографам, которые говорят про чрезмерно увеличенную фотографию: «полезло зерно». И в самом деле, увеличенное точечное изображение приобретает отчетливо видимую зернистую структуру, а это нужно только при создании специфических художественных эффектов. **В машинной графике подобное явление называется искажением растривания.** При этом под растриванием подразумевается процесс преобразования векторного изображения в растровое (или одного пиксельного изображения в другое со сменой характеристик пикселей). При уменьшении пиксельного изображения с сохранением прежнего размера пикселей неизбежно приходится выбрасывать некоторые пиксели, что приводит к потере части содержащейся в изображении информации. Не спасает положения и уменьшение размеров пиксела, поскольку устройства отображения информации (дисплеи, полиграфические машины и принтеры) все равно не могут воспроизводить слишком маленькие пиксели — в результате детали изображения становятся плохо различимыми. Более того, оказывается, что **размеры пиксельного изображения при сохранении исходного размера пикселей можно увеличивать лишь кратно - в два, три и т. д. раза.** Если это условие не соблюдается, на изображении может возникать муар — волнообразные полосы, точки или клетки. Избавиться от муара, не искажая само изображение, не так-то просто.

Вторым, не менее существенным, недостатком пиксельных изображений является отсутствие внутренней структуры, соответствующей структуре изображенных объектов. Попробуем разобраться в этом на примере. Если на точечном изображении мы видим мужчину в галстук-бабочке и со значком на лацкане, это — результат работы нашего мозга, выделившего в изображении такие объекты, как значок и галстук. Чтобы идентифицировать соответствующие этим объектам пиксели при работе с программой редактирования пиксельной графики, придется немало потрудиться. Если,

работая над изображением, вы удалите значок, то после этого придется еще как-то заполнять образовавшуюся после удаления пикселей значка «дыру» в изображении — фактически, дорисовывать его. Еще больше вам придется потрудиться, если вы захотите слегка поправить покосившийся галстук

Третий недостаток пиксельных изображений связан с тем, что для их хранения необходим большой объем памяти. При работе с точечными изображениями высокой четкости и сравнительнобольшого размера нередки случаи, когда объемы соответствующих им файлов составляют сотни мегабайтов. Работа с такими громоздкими объектами зачастую оказывается не под силу даже самым современным и мощным компьютерам. **Работе с векторным изображением присуща большая гибкость.** Чтобы увеличить или уменьшить его, требуется всего лишь изменить один управляющий параметр изображения в целом — масштаб. При этом размер файла с векторным изображением не увеличится ни на один байт. Внесенные изменения будут учтены при рендеринге, и четкость изображения не пострадает. В отличие от пиксельного изображения, степень структуризации векторного изображения может быть произвольной. Она определяется создающим его художником. Ссоставляя изображение значка на лацкане из двух кругов и текстовой надписи, можно объединить эти объекты в группу и даже дать ей имя «Значок», чтобы потом было легче найти эти объекты. Удаление этой группы приведет к исчезновению значка, но не к возникновению «дыры» на пиджаке — ведь изображение пиджака составлено из других объектов, которые просто становятся видны в том месте, где раньше был значок. У вас не будет затруднений, если вы также захотите «поправить галстук» — достаточно изменить значение угла поворота группы объектов, из которых составлено его изображение. *Объем файлов с векторными изображениями в большинстве случаев намного меньше объема файлов с изображениями пиксельными.* В завершение сравнительного анализа классов изображений отметим, что преобразование векторного изображения в пиксельное (растрирование или рендеринг) представляет собой достаточно простой и абсолютно формальный

процесс, который выполняется большинством программ машинной графики без вмешательства пользователя. Преобразование же пиксельного изображения в векторное (векторизация или трассировка) в подавляющем большинстве случаев требует не просто вмешательства, а творческого участия пользователя

Видеомонтаж. Автоматизированное проектирование.

Видео (от лат. video — смотрю, вижу) — под этим термином понимают широкий спектр технологий записи, обработки, передачи, хранения и воспроизведения визуального и аудиовизуального материала на мониторах. Когда в быту говорят «видео» — то обычно имеют в виду видеоматериал, телесигнал или кинофильм, записанный на физическом носителе (видеокассете, видеодиске и т. п.).

Количество (частота) кадров в секунду — это число неподвижных изображений, сменяющих друг друга при показе 1 секунды видеоматериала и создающих эффект движения объектов на экране. Чем больше частота кадров в секунду, тем более плавным и естественным будет казаться движение. Минимальный показатель, при котором движение будет восприниматься однородным — примерно 10 кадров в секунду (это значение индивидуально для каждого человека). В традиционном плёночном кинематографе используется частота 24 кадра в секунду. Системы телевидения PAL и SECAM используют 25 кадров в секунду (англ. 25 fps или 25 Герц), а система NTSC использует 29,97 кадров в секунду. Компьютерные оцифрованные видеоматериалы хорошего качества, как правило, используют частоту 30 кадров в секунду.

Оборудование для обработки видео на компьютере.

Для записи видеоинформации необходимо:

- специальная плата или устройство для оцифровки видеоизображения;
- видеомагнитофон или видеокамера;

- программное обеспечение для записи и редактирования цифрового видео.
- звуковая карта (если плата видеозахвата не поддерживает возможности захвата звука).

Видеокарта (видеоадаптер). Совместно с монитором *видеокарта* образует *видеоподсистему* персонального компьютера. Физически видеоадаптер выполнен в виде отдельной *дочерней платы*, которая вставляется в один из слотов материнской платы и называется *видеокартой*. Видеоадаптер взял на себя функции *видеоконтроллера*, *видеопроцессора* и *видеопамяти*.

За время существования персональных компьютеров сменилось несколько стандартов видеоадаптеров: *MDA (монохромный)*; *CGA (4 цвета)*; *EGA (16 цветов)*; *VGA (256 цветов)*. В настоящее время применяются видеоадаптеры *SVGA*, обеспечивающие по выбору воспроизведение до 16,7 миллионов цветов с возможностью произвольного выбора разрешения экрана из стандартного ряда значений.

Плата оцифровки видео

Можно воспользоваться простейшей аналоговой картой видеозахвата или TV-тюнером. При этом существуют следующие особенности такой платы. Она должна:

- показывать и захватывать аналоговое видео со скоростью потока данных, ограничиваемым только устройством записи;
- захватывать видео с произвольными размерами кадра, в частности, с разрешением 352×288 (необходимое для стандарта MPEG-1);
- захватывать видео как через композитный вход, так и через S- Video.

Основные форматы видео файлов

Audio Video Interleaved (.AVI)* - формат, разработанный Microsoft для записи и воспроизведения видео в операционной системе Windows. При записи в этом формате используются несколько различных алгоритмов сжатия

(компрессии) видеоизображения. Среди них Cinepak, Indeo video, Motion-JPEG (M-JPEG) и др. Но только M-JPEG был признан среди них как международный стандарт для сжатия видео. Первоначально для захвата и воспроизведения видео использовались возможности программного комплекта Video for Windows, разработанного Microsoft. Компания Microsoft разработала два формата, призванных заменить формат AVI: *Advanced Streaming Format* (*.ASF) и *Advanced Authoring Format* (*.AAF).

Windows Media Video (*.WMV) - новый формат видео от Microsoft, который приходит на смену формату AVI. В его основе Windows Video Codec, разработанный на базе стандарта MPEG-4.

Quick Time Move (*.MOV) - наиболее распространенный формат для записи и воспроизведения видео, разработанный фирмой Apple для компьютеров Macintosh в рамках технологии Quick Time. Включает поддержку не только видео, но и звука, текста, потоков MPEG, расширенного набора команд MIDI, векторной графики, панорам и объектов (QT) и трехмерных моделей. Поддерживает несколько различных форматов сжатия видео, в том числе MPEG, а также свой собственный метод компрессии.

MPEG (*.MPG, *.MPEG) - формат для записи и воспроизведения видео, разработанный группой экспертов по движущимся изображениям (MPEG). Имеет собственный алгоритм компрессии. В настоящее время активно используются для записи цифрового видео. Наиболее широкое распространение нашли два формата: MPEG-I и MPEG-2. Они различаются по объему и качеству получаемой видеoinформации и признаны международными стандартами для сжатия видео. В настоящее время наряду с MPEG-I и MPEG-2 используется новый формат MPEG-4. Он позволяет сжать информацию с большим коэффициентом сжатия.

Digital Video (*.DV) - формат, разработанный для цифровых видеокамер и видеомagneтофонов. Кодер-декодер (кодек) определен ведущими мировыми производителями электроники, чтобы его могли поддерживать производители в своих платах с интерфейсом FireWare и комплексных решениях для

редактирования цифрового видео. Формат не является компактным, поэтому необходимо его преобразование в MPEG.

Кодеки для видеоинформации

Кодек является сокращением от слов компрессор и декомпрессор, это любая технология для сжатия и обратного восстановления данных. Кодеки могут быть реализованы на уровне программного обеспечения, аппаратной части или в их комбинации. Кодеки используются для того, чтобы сделать большие видео файлы намного меньше, делая их пригодными для распространения по сети, локальной или всемирной, или любого другого метода передачи файлов.

Часть кодеков пользователь получает при установке операционной системы Windows. Это те кодеки, которые используются в файлах формата AVI. Если требуется пользоваться для записи форматом QuickTime, то нужно установить его поддержку.

Cinepak - создавался фирмой SuperMac для компьютеров с процессорами Motorola 68030 и Intel386 и с однокоростными дисковыми CD-ROM. Файлы AVI, в которых использован этот кодек, могут быть переведены в формат QuickTime и, наоборот, без переупаковки.

Sorenson Video - использует усовершенствованные алгоритмы векторного квантования и компенсации движения и адаптивное управление потоком. Оптимизирован для работы со скоростями от 2 до 100 Кбайт/с. Качество изображения значительно превышает *Cinepak* даже при меньших размерах файла. Сжатие очень медленное. Включен в стандартную поставку QuickTime.

Indeo Video Interactive (IVI) - видеокодек для форматов AVI и MOV требует компьютеры класса Pentium. Кодек основан на wavelet - компрессии.

MPEG - кодек, который, является официальным стандартом для сжатия и видеоинформации. В нем применяется дискретное косинусное преобразование (DCT, или ДКП) с межкадровым предсказанием. Существует несколько версий этого стандарта (MPEG-1, MPEG-2; MPEG-4).

MPEG-4 задает правила организации объектно-ориентированной среды. Он имеет дело не просто с потоками и массивами медиаданных, а с медиа объектами, например, аудио, видео, аудио/видео, графическими (плоскими и трехмерными), текстовыми. При этом *MPEG-4* обеспечивает наилучшую эффективность сжатия. Используется алгоритм "сжатия повышенной эффективности" (ACE - Advanced Coding Efficiency). Можно обрабатывать звуковые сигналы в диапазоне от 2 до 24 Кбит/с, а видеосигнал - от 5 до 10 Мбит/с. Благодаря такой масштабируемости, аудио и видеоданные можно адаптировать к реальному применению. Этим обусловлен более универсальный характер *MPEG-4* по сравнению с *MPEG-2*. Наиболее интересны практические достижения в создании видеокодеков, построенных на основе стандарта *MPEG-4*. Среди них - Windows Media Codec и DivX.

Программы для видеомонтажа

Чтобы превратить оцифрованную информацию в готовый продукт, ее необходимо обработать: разместить монтажные эпизоды, задать эффекты и переходы между ними, добавить титры и пояснительные тексты, отредактировать звуковое сопровождение, наконец, смонтировать готовый фильм. Для этого можно использовать следующие программы: Windows Movie Maker, Pinnacle Studio, VideoStudio, Video Wave, Media Studio Pro, Adobe Premiere, Speed Razor Pro, Adobe After Effects, Cannopus Edius и др.

Основные операции с видео файлами

1. Импорт
2. Добавление/удаление кадров
3. Изменение размера кадра
4. Разбиение кадра на фрагменты
5. Добавление эффектов и переходов
6. Тоновая коррекция кадров
7. Добавление интерактивных элементов управления

8. Добавление текста (титры, бегущая строка, текст на кадре и т.д.)

Звук. Синтезаторы звука на персональном компьютере.

Звук – это механические колебания среды: воздуха, воды и т.д., воспринимаемые слуховым аппаратом человека. То, что мы слышим – это результат обработки колебательных движений барабанной перепонки уха, представленный в виде сигналов нервной системы. Вне среды переноса звуковых волн звук не существует. Однако звуковые колебания можно перевести на другой носитель: изменить представление информации, не теряя ее фактически. Обычно звуковые колебания переносят на сигналы радиоволн.

Звуковую информацию можно также представить в математической форме, в виде периодических функций времени. Это представление обычно записывают в виде формулы

$$A(t) = a_0 + \sum_{k>0} (a_k \cos \omega_k t + b_k \sin \omega_k t)$$

Здесь $A(t)$ – амплитуда звукового сигнала, а t – время.

Все слышимые звуки являются результатом воздействия звуковых волн. На магнитной ленте, виниловой пластинке звук сохраняется в виде непрерывного электрического сигнала, определяющего изменение звуковых волн. Звук, создаваемый электрическими волнами, называют аналоговым.

Звук может храниться на цифровых носителях, т.е. быть представленным в виде набора цифр. Любая цифровая техника или программа работают со звуком, представленным в цифровом виде. Таким образом, для переноса звука на цифровой носитель, необходимо осуществить его аналогово-цифровое преобразование. Такое преобразование состоит из трех этапов:

дискретизация – представление непрерывного сигнала в виде последовательного набора отдельных амплитуд;

квантование – разделение каждой амплитуды на заданное число уровней;

кодирование – запись данных позиции и уровня амплитуды в цифровом виде.

На практике преобразования звуковой информации из непрерывной формы в дискретную выполняются электронными устройствами, называемыми *аналого-цифровыми преобразователями* (АЦП) и *цифро-аналоговыми преобразователями* (ЦАП). Современные звуковые карты могут обеспечить кодирование 65536 различных уровней сигнала или состояний. Для определения количества бит, необходимых для кодирования, решим показательное уравнение: $65536 = 2^I$, т.к. $65536 = 2^{16}$, то $I = 16$ бит.

Таким образом, современные звуковые карты обеспечивают 16-битное кодирование звука. При каждой выборке значению амплитуды звукового сигнала присваивается 16-битный код.

Количество выборок в секунду может быть в диапазоне от 8000 до 48000, т.е. частота дискретизации аналогового звукового сигнала может принимать значения от 8 до 48 КГц. При частоте 8 КГц качество дискретизированного звукового сигнала соответствует качеству радиотрансляции, а при частоте 48 КГц – качеству звучания аудио-CD. Следует также учитывать, что возможны как моно-, так и стерео-режимы.

Можно оценить информационный объем моно-аудио-файла длительностью звучания 1 секунду при среднем качестве звука (16 бит, 24 КГц). Для этого количество бит на одну выборку необходимо умножить на количество выборок в 1 секунду:

$$16 \text{ бит} * 24000 = 384000 \text{ бит} = 48000 \text{ байт} \approx 47 \text{ Кбайт}.$$

Устройства для работы со звуком

Звуковая карта явилась одним из наиболее поздних усовершенствований персонального компьютера. Она подключается к одному из слотов материнской платы в виде дочерней карты и выполняет вычислительные операции, связанные с обработкой звука, речи, музыки. Звук воспроизводится через **внешние звуковые колонки**, подключаемые к выходу звуковой карты.

Специальный разъем позволяет отправить звуковой сигнал на внешний усилитель. Имеется также разъем для подключения *микрофона*, что позволяет записывать речь или музыку и сохранять их на жестком диске для последующей обработки и использования.

Основным параметром звуковой карты является *разрядность*, определяющая количество битов, используемых при преобразовании сигналов из аналоговой в цифровую форму и наоборот. Чем выше разрядность, тем меньше погрешность, связанная с оцифровкой, тем выше качество звучания. Минимальным требованием сегодняшнего дня являются 16 разрядов, а наибольшее распространение имеют 32-разрядные и 64-разрядные устройства.

Форматы звуковых файлов.

Звук в компьютере хранится в файлах, имеющих различные способы представления информации. Перечислим основные форматы хранения звуковой информации.

WAVE (*.wav) – наиболее широко распространенный звуковой формат. Используется операционной системой Windows для хранения звуковых файлов. В его основе лежит формат RIFF (Resource Interchange File Format), позволяющий сохранять данные в структурированном виде.

Стандарт MPEG-1 представляет собой, целый комплект аудио и видео стандартов. Согласно стандартам ISO (International Standards Organization), аудио часть MPEG-1 включает в себя три алгоритма различных уровней сложности: Layer 1 (уровень 1), Layer 2 (уровень 2) и Layer 3 (уровень 3). Общая структура процесса кодирования одинакова для всех уровней MPEG-1 . Вместе с тем, несмотря на схожесть уровней в общем подходе к кодированию, уровни различаются по целевому использованию и задействованным в кодировании внутренним механизмам. Для каждого уровня определен свой формат записи выходного потока данных и, соответственно, свой алгоритм декодирования.

MPEG Layer 3 (*.mp3) - формат звуковых файлов с потерями качества, разработанный для сохранения звуков, отличных от человеческой речи. Используется для оцифровки музыкальных записей.

Windows Media Audio (*.wma) - формат звуковых файлов, предложенный фирмой Microsoft. Кодек Windows Media Audio 8 обеспечивает качество, аналогичное mp3, при размерах файлов втрое меньших.

MIDI (*.mid) - цифровой интерфейс музыкальных инструментов (Musical Instrument Digital Interface). MIDI определяет обмен данными между музыкальными и звуковыми синтезаторами разных производителей. Интерфейс MIDI представляет собой протокол передачи музыкальных нот и мелодий. Но данные MIDI не являются цифровым звуком: это сокращенная форма записи музыки в числовой форме.

Программное обеспечение для редактирования звука.

Наиболее известными в настоящее время являются следующие программы для обработки звука: Sound Forge, GoldWave, Adobe Audition и др.

Основные операции со звуком.

1. Запись.
2. Добавление/удаление звуковой дорожки.
3. Изменение размера звуковой дорожки.
4. Разбиение звуковой дорожки на фрагменты.
5. Редактирование звуковой кривой.
6. Изменение громкости звучания.

Учебно-методическое и информационное обеспечение

Основная литература:

- 1 Гаврилов, М. В. Информатика и информационные технологии [Электронный ресурс]: учебник для СПО / М. В. Гаврилов, В. А. Климов.

- 4-е изд., перераб. и доп. — М. : Издательство Юрайт, 2017. — 383 с. — Режим доступа: <https://www.biblio-online.ru>.
- 2 Илюшечкин, В. М. Основы использования и проектирования баз данных [Электронный ресурс]: учебник для СПО / В. М. Илюшечкин. — испр. и доп. — М. : Издательство Юрайт, 2017. — 213 с. — Режим доступа: <https://www.biblio-online.ru>.
 - 3 Казанский, А. А. Объектно-ориентированный анализ и программирование на visual basic 2013 [Электронный ресурс]: учебник для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 290 с. — Режим доступа: <https://www.biblio-online.ru>.
 - 4 Казанский, А. А. Прикладное программирование на excel 2013 [Электронный ресурс]: учебное пособие для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 159 с. — Режим доступа: <https://www.biblio-online.ru>.
 - 5 Казанский, А. А. Программирование на visual c# 2013 [Электронный ресурс]: учебное пособие для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 191 с. — Режим доступа: <https://www.biblio-online.ru>.
 - 6 Семакин И.Г. Основы алгоритмизации и программирования: учеб. для сред. проф.образования/ И.Г.Семакин, А.П. Шестаков. – 3-е изд., стер. – М.: Академия, 2012. – 392 с.
 - 7 Советов, Б. Я. Базы данных [Электронный ресурс]: учебник для СПО / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 2-е изд. — М. : Издательство Юрайт, 2017. — 463 с. — Режим доступа: <https://www.biblio-online.ru>.
 - 8 Советов, Б. Я. Информационные технологии [Электронный ресурс]: учебник для СПО / Б. Я. Советов, В. В. Цехановский. — 6-е изд., перераб. и доп. — М. : Издательство Юрайт, 2017. — 261 с. — Режим доступа: <https://www.biblio-online.ru>.

- 9 Федорова Г. Н. Разработка и администрирование баз данных : учеб. для студ. учреждений сред. проф. образования / Г. Н. Федорова. - М.: Академия, 2015. - 313 с.
- 10 Цветкова, М.С. Информатика и ИКТ : учеб. для студентов учреждений сред. проф. образования / М. С. Цветкова, Л. С. Великович. - 8-е изд., стер. - М. : Академия, 2016. - 332, [1] с., [4] л. ил.
- 11 Черпаков, И. В. Основы программирования [Электронный ресурс]: учебник и практикум для СПО / И. В. Черпаков. — М. : Издательство Юрайт, 2017. — 219 с. — Режим доступа: <https://www.biblio-online.ru>.

Тема 3.3 Электронная презентация

План лекции

- 1 Система компьютерной презентации MS Power Point

Система компьютерной презентации MS Power Point

Microsoft Power Point – это компонент Microsoft Office, создающий файлы, называемые презентацией.

Основным компонентом презентации является слайд. Слайд это специальная цветная картинка, посвященная определенной теме, которая может содержать разнообразную информацию. В процессе разработки презентации слайды можно переставлять, удалять, добавлять новые или же просто изменять содержание существующих слайдов.

Слайд обычно состоит из шести частей.

- **Заголовок** располагается в верхней части слайда.
- **Текст слайда** - это его основная часть. Чаще всего он представляет собой маркированный или нумерованный список. Текст, отображаемый на слайде, вводится в специальное окно, называемое *заполнителем*.

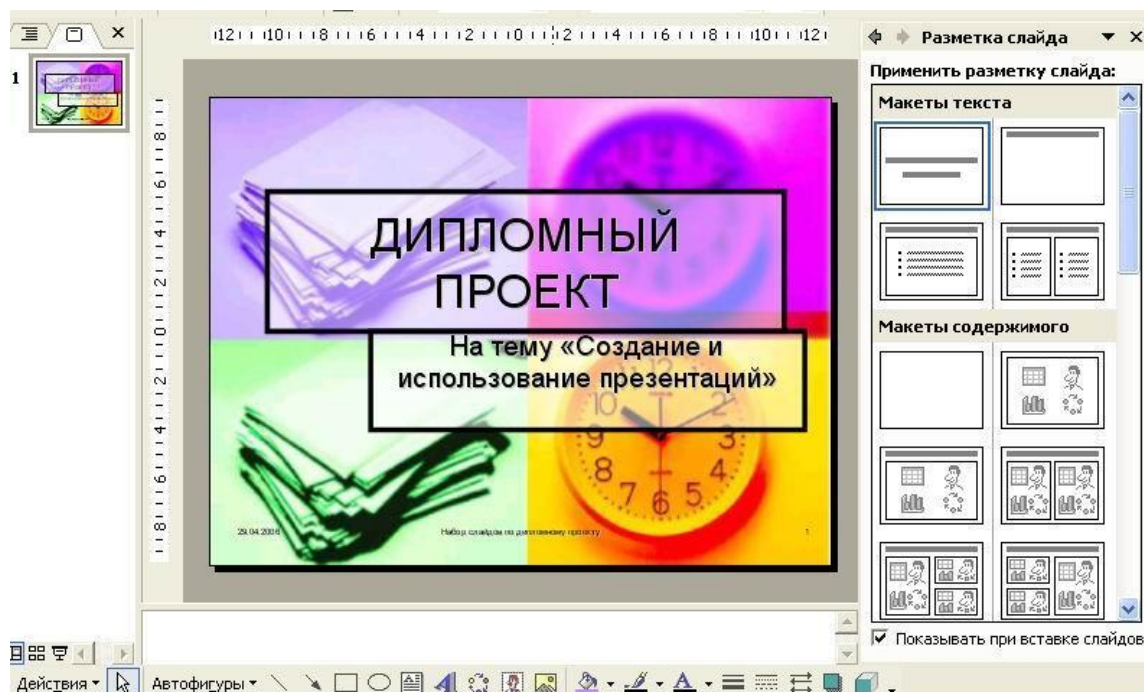
- Слайды могут содержать **рисунки, диаграммы, видеоклипы, звуковые файлы** и другую информацию. Для их размещения на слайде существует определенное заполнителем место.
- **Дату и время** ставят на слайде в правом нижнем углу.
- **Нижний колонтитул**, если он необходим, располагают в нижней части слайда.
- **Номер слайда** отображают в правом нижнем углу.

Обычно презентация начинается с **тительного** слайда, который содержит основную информацию о предстоящем докладе (рис. 1.1):

- название темы,
- цель презентации,
- имя докладчика и другие вводные сведения.

Титульный слайд не следует путать с заголовком слайдов.

панель структуры презентации заголовок слайда



дата и время нижний колонтитул номер слайда

В РР заложено множество профессионально разработанных шаблонов и цветовых схем, в которых заранее заданы расположение объектов, фон слайда и другие его характеристики. Если же никакие предлагаемые стандартные

варианты вас не устраивают, всегда можно создать слайд по своему собственному дизайну.

В зависимости от того, с помощью какого оборудования данная презентация будет представляться, необходимо учитывать ряд тонкостей в ее оформлении.

- **Монитор компьютера.** Показ презентации на мониторе компьютера хорош в том случае, если она предназначена для небольшой группы людей. При создании такой презентации нельзя перегружать слайды большим количеством информации.
- **Проекционный экран.** На проекционном экране обычно отображается копия экрана вашего монитора в увеличенном размере. Естественно, этот вариант подходит для просмотра большей аудиторией, а в каждый слайд в этом случае можно включить намного больше деталей.
- **Web.** Размещение презентации на Web-странице в Internet.
- **Прозрачные транспаранты ("прозрачки").** Бывают черно-белыми или цветными и демонстрируются с помощью специального проекционного аппарата на экран. В этом случае не следует включать в слайды анимацию и другие специальные мультимедийные эффекты.
- **Цветные или черно-белые 35-миллиметровые слайды.**

Учебно-методическое и информационное обеспечение

Основная литература:

- 1 Гаврилов, М. В. Информатика и информационные технологии [Электронный ресурс]: учебник для СПО / М. В. Гаврилов, В. А. Климов. — 4-е изд., перераб. и доп. — М. : Издательство Юрайт, 2017. — 383 с. — Режим доступа: <https://www.biblio-online.ru>.
- 2 Илюшечкин, В. М. Основы использования и проектирования баз данных [Электронный ресурс]: учебник для СПО / В. М. Илюшечкин. — испр. и

- доп. — М. : Издательство Юрайт, 2017. — 213 с. — Режим доступа: <https://www.biblio-online.ru>.
- 3 Казанский, А. А. Объектно-ориентированный анализ и программирование на visual basic 2013 [Электронный ресурс]: учебник для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 290 с. — Режим доступа: <https://www.biblio-online.ru>.
 - 4 Казанский, А. А. Прикладное программирование на excel 2013 [Электронный ресурс]: учебное пособие для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 159 с. — Режим доступа: <https://www.biblio-online.ru>.
 - 5 Казанский, А. А. Программирование на visual c# 2013 [Электронный ресурс]: учебное пособие для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 191 с. — Режим доступа: <https://www.biblio-online.ru>.
 - 6 Семакин И.Г. Основы алгоритмизации и программирования: учеб. для сред. проф.образования/ И.Г.Семакин, А.П. Шестаков. – 3-е изд., стер. – М.: Академия, 2012. – 392 с.
 - 7 Советов, Б. Я. Базы данных [Электронный ресурс]: учебник для СПО / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 2-е изд. — М. : Издательство Юрайт, 2017. — 463 с. — Режим доступа: <https://www.biblio-online.ru>.
 - 8 Советов, Б. Я. Информационные технологии [Электронный ресурс]: учебник для СПО / Б. Я. Советов, В. В. Цехановский. — 6-е изд., перераб. и доп. — М. : Издательство Юрайт, 2017. — 261 с. — Режим доступа: <https://www.biblio-online.ru>.
 - 9 Федорова Г. Н. Разработка и администрирование баз данных : учеб. для студ. учреждений сред. проф. образования / Г. Н. Федорова. - М.: Академия, 2015. - 313 с.

- 10 Цветкова, М.С. Информатика и ИКТ : учеб. для студентов учреждений сред. проф. образования / М. С. Цветкова, Л. С. Великович. - 8-е изд., стер. - М. : Академия, 2016. - 332, [1] с., [4] л. ил.
- 11 Черпаков, И. В. Основы программирования [Электронный ресурс]: учебник и практикум для СПО / И. В. Черпаков. — М. : Издательство Юрайт, 2017. — 219 с. — Режим доступа: <https://www.biblio-online.ru>.

РАЗДЕЛ 4 ТЕХНОЛОГИИ РАБОТЫ С ИНФОРМАЦИОННЫМИ СТРУКТУРАМИ

Тема 4.1 Электронные таблицы

План лекции

1. Персональный компьютер как вычислитель. Моделирование электронной таблицы
2. Примеры моделирование в электронной таблице

1 Персональный компьютер как вычислитель. Моделирование электронной таблицы

Одной из самых продуктивных идей в области компьютерных информационных технологий стала идея электронной таблицы.

Электронной называется таблица, создаваемая в компьютере как универсальное программное средство для автоматизации расчётов над табличными данными.

Многие фирмы - разработчики программного обеспечения для ПК создали свои версии табличных процессоров - прикладных программ, предназначенных для работы с электронными таблицами. Из них наибольшую известность приобрели Lotus 1-2-3 фирмы Lotus Development, Supercalc фирмы Computer Associates, Multiplan и Excel фирмы Microsoft. Отечественные школьные компьютеры также оснащены упрощенными (учебными) версиями табличных процессоров.

Табличные процессоры (ТП) - удобный инструмент для экономистов, бухгалтеров, инженеров, научных работников - всех тех, кому приходится работать с большими массивами числовой информации. Эти программы позволяют создавать таблицы, которые (в отличие от реляционных баз данных) являются динамическими, т. е. содержат так называемые "вычисляемые поля", значения которых автоматически пересчитываются по заданным формулам при изменении значений исходных данных, содержащихся в других полях.

Благодаря ТП создаются документы - **электронные таблицы** (ЭТ). Электронная таблица (документ) создается в памяти компьютера. В дальнейшем ее можно просматривать, изменять, записывать на магнитный диск для хранения, печатать на принтере.

Среда ТП:

Рабочим полем табличного процессора является экран дисплея, на котором электронная таблица представляется в виде матрицы. ЭТ, подобно шахматной доске разделена на клетки, которые принято называть ячейками таблицы. Строки и столбцы таблицы имеют обозначения. Чаще всего строки имеют числовую нумерацию, а столбцы - буквенные (буквы латинского алфавита) обозначения. Как и на шахматной доске, каждая клетка имеет свое имя (адрес), состоящее из имени столбца и номера строки, например: A1, C13, F24 и т.п.

Но если на шахматной доске всего $8 \times 8 = 64$ клетки, то в электронной таблице ячеек значительно больше. Например, у табличного процессора Excel таблица максимального размера содержит 256 столбцов и 65536 строк. Поскольку в латинском алфавите всего 26 букв, то, начиная с 27-го столбца, используются двухбуквенные обозначения, также в алфавитном порядке: AA, AB, AC,..., AZ, BA, BB, BC,..., BZ, CA...

Последний, 256-й столбец имеет имя IV (не путайте с римской цифрой). Значит, существуют ячейки с такими, например, именами: DL67, HZ10234 и т. п.

На экране дисплея видна не вся электронная таблица (документ), а только ее часть. Документ в полном объеме хранится в оперативной памяти, а экран можно считать окном, через которое пользователь имеет возможность просматривать его.

В ТП Excel реализована возможность работы одновременно с несколькими таблицами, расположенными на разных листах. Пользователь может "перелистывать" эти листы, как в книге.

При заполнении таблицы и при ее просмотре большую роль играет табличный курсор - прямоугольник, выделенный цветом. Табличный курсор всегда занимает клетку таблицы, которая называется текущей клеткой. Перемещая курсор по таблице, мы тем самым перемещаем "окно" по документу, хранящемуся в оперативной памяти, и делаем видимым ту или иную его часть.

Важным элементом среды табличного процессора является **меню команд**. С его помощью управляют работой электронной таблицы. Меню команд может быть словесным (в Supercalc) или пиктографическим (в Excel).

Панель диалога обеспечивает взаимодействие пользователя с табличным процессором и может содержать следующие строки: строку состояния, строку запроса, строку ввода и строку помощи, расположенные в нижней части экрана.

Строка состояния предназначена для выдачи информации о текущей клетке. Строка запроса содержит возможные варианты ответа на запросы табличного процессора. Если пользователь не ведет диалог с табличным процессором, то строка запроса содержит некоторую дополнительную информацию: ширину текущего столбца, объем свободной памяти, размер таблицы с данными (активной таблицы).

Строка ввода предназначена для индикации данных, которые пользователь набирает перед вводом в клетки таблицы.

Строка помощи может содержать расшифровку текущей команды, индикатор состояния табличного процессора (не путать со строкой состояния). Можно выделить несколько основных таких состояний:

- "ожидание" - ожидание набора данных или команд;
- "меню" - ожидание выбора команды из меню команд;
- "ввод" - ввод данных;
- "редактирование" - редактирование данных в строке ввода.

Данные в электронной таблице

Все данные таблицы размещаются в ячейках. Содержимым ячейки может быть текст, числовое значение или формула. Табличный процессор должен "знать", какого типа данное хранится в конкретной ячейке таблицы, для того чтобы правильно интерпретировать ее содержимое. Текст и числа рассматриваются как константы. Изменить их можно только путем редактирования соответствующих ячеек. Формулы же автоматически пересчитывают свои значения, как только хотя бы один их операнд был изменен.

Примеры записи формул:

$$\begin{aligned} &2.5*A1+B2*C3; \\ &(B3-C1)/(B3+C1); \\ &F7/2+G7/3; \\ &(A5-1)^2 \end{aligned}$$

Правила записи формул подобны правилам записи арифметических выражений в языках программирования. Только здесь в качестве идентификаторов переменных выступают имена ячеек таблицы. Кроме арифметических операций формулы могут содержать стандартные функции. У каждого ТП свой набор стандартных функций.

Режимы работы и система команд ТП.

Можно выделить следующие режимы работы табличного процессора: формирование электронной таблицы; управление вычислениями; режим

отображения формул; графический режим; работа электронной таблицы как базы данных.

Система команд тесно связана с режимами работы электронной таблицы. Как правило, команды реализуются через меню команд или через функциональные клавиши.

Рассмотрим подробнее режимы работы электронных таблиц и команды, связанные с ними.

1. Режим формирования электронных таблиц предполагает заполнение и редактирование документа. Базовые команды формирования таблиц можно разбить на две группы:

- команды, изменяющие содержимое клеток (очистить, редактировать, копировать);
- команды, изменяющие структуру таблицы (удалить, вставить, переместить).

2. Режим управления вычислениями. Все вычисления начинаются с клетки, расположенной на пересечении первой строки и первого столбца электронной таблицы. Вычисления проводятся в естественном порядке, т. е. если в очередной клетке находится формула, включающая адрес еще не вычисленной клетки, то вычисления по этой формуле откладываются до тех пор, пока значение в клетке, от которой зависит формула, не будет определено.

При каждом вводе нового данного в клетку документ пересчитывается заново - реализуется автоматический пересчет. В некоторых табличных процессорах существует возможность установки ручного пересчета, т.е. таблица пересчитывается заново только при подаче специальной команды.

3. Режим отображения формул задает индикацию содержимого клеток на экране. Обычно этот режим выключен и на экране отображаются значения, вычисленные на основании содержимого клеток.

4. Графический режим дает возможность отображать числовую информацию в графическом виде, чаще всего в виде диаграмм. Команды графического режима можно разбить на две группы:

- команды описания диаграмм (задают данные, которые будут выведены в

графическом виде, задают тип диаграмм и т.д.);
- команды вывода диаграмм.

5. Работа в режиме баз данных реализована в профессиональных ТП. Возможность искать и выбирать данные из таблицы позволяет использовать электронную таблицу в качестве несложной базы данных. При работе с базами данных приходится иметь дело с такими понятиями, как файл, запись, поле данных. В электронных таблицах файлом является сама таблица, записями - строки таблицы, полями - клетки таблицы.

Электронные таблицы успешно применяются при проведении учёта материальных ценностей, планово-финансовых и инженерно-научных расчётов, в бухгалтерском учёте и при оформлении различных документов, в конструировании и при обработке результатов социологических опросов населения или всевозможных наблюдений. ЭТ может использоваться также для моделирования и расчёта многих временных процессов: равноускоренное движение (физика), размножение (биология), скорость реакции (химия), ее можно применять для ведения, например, домашней бухгалтерии и планирования использования денежных средств.

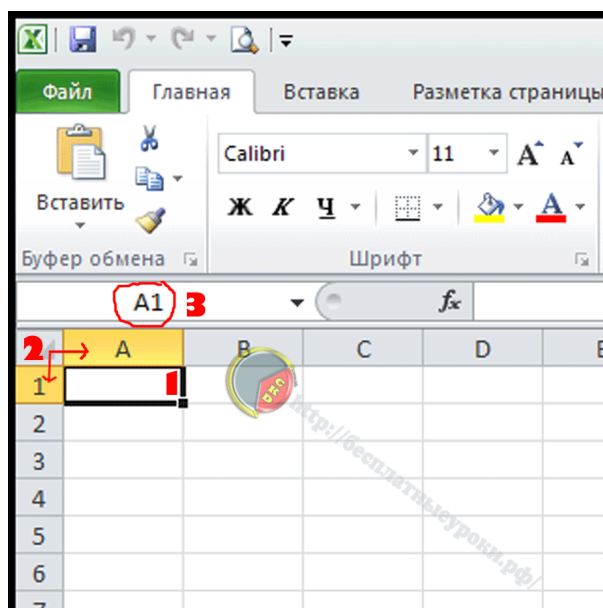
2 Примеры моделирование в электронной таблице

Как в любой другой электронной таблице для простоты в Microsoft Excel используются буквенно-цифровое обозначение адреса ячеек. Текущую ячейку пользователь может определить несколькими способами:

1 способ: На самой рабочей таблице текущая ячейка выделена табличным курсором (черный прямоугольник, обрамляющий ячейку) 1

2 способ: В обрамлении окна рабочей таблицы будут выделены цветом и жирным начертанием буква столбца и цифра строки, на пересечении которых находится текущая ячейка 2.

3 способ: В строке формул, в левой ее части, будет отображен адрес текущей ячейки 3.



Каждая ячейка в рабочей таблице имеет свой индивидуальный адрес, который будет использоваться в формулах. Например, адрес C5 указывает на то, что данные будут браться из ячейки, расположенной на пересечении столбца C и строки 5.

Для указания адреса ячейки с другого листа в правой части строки формул необходимо использовать выражение:

= [Название листа] ! [Адрес ячейки]

Например, Лист5!B10 – означает, что данные будут браться из ячейки B10 расположенной на листе с именем Лист5.

Нумерация столбцов производится буквами латинского алфавита от A до Z, а затем сочетанием двух букв AA, AB, AC и т.д. Нумерация строк производится цифрами.

{social}

Форматы данных. Способы ввода и оформления данных

Файл Microsoft Office Excel 2007 можно сохранить в другом формате. Для этого нажмите кнопку **Microsoft Office**  и выберите пункт **Сохранить как...** Форматы файлов, доступные в диалоговом окне **Сохранение документа**, изменяются в зависимости от типа активного листа (лист электронной таблицы, диаграмма или другой тип листа).

Файл, созданный с другим форматом (в предыдущей версии Microsoft Office Excel или в другой программе), можно открыть, нажав кнопку **Microsoft Office**  и выбрав команду **Открыть**. Если это не общая книга, программа предложит сохранить файл в формате Office Excel 2007. Если файл создан в предыдущей версии Microsoft Excel, его можно сохранить в исходном формате.

ПРИМЕЧАНИЕ. При сохранении файла в другом файловом формате часть форматирования, данных и специальных свойств может быть утрачена. Дополнительные сведения см. в разделе Возможности и форматирование, которые не сохраняются при преобразовании файлов Microsoft Excel.

Форматы Excel

ФОРМАТ	РАСШИРЕНИЕ	ОПИСАНИЕ
Книга Excel	XLSX	Стандартный формат файлов Office Excel 2007, основанный на языке XML. В этом формате нельзя сохранять код макросов Microsoft Visual Basic для приложений (VBA) и листы макросов Microsoft Office Excel 4.0 (XLM).
Лист Excel (код)	XLSM	Формат Office Excel 2007 на основе XML с поддержкой макросов. Позволяет сохранять код макросов VBA и листы макросов Microsoft Excel 4.0 (XLM).
Двоичная книга Excel	XLSB	Формат двоичных файлов Office Excel 2007 (BIFF12).
Шаблон	XLTX	Стандартный формат файлов Office Excel 2007 для шаблонов Excel. Не позволяет сохранять код VBA-макросов, а также листы макросов Microsoft Excel 4.0 (XLM).
Шаблон (код)	XLTM	Формат файлов Office Excel 2007 для шаблонов Excel с поддержкой макросов. Позволяет сохранять код макросов VBA и листы макросов Microsoft Excel 4.0 (XLM).
Книга Microsoft Excel 97 — 2003	XLS	Формат двоичных файлов Excel 97 — 2003 (BIFF8).
Шаблон Excel 97 — 2003	XLT	Формат двоичных файлов Excel 97 — 2003 (BIFF8) для хранения шаблонов Excel.
Книга Microsoft Excel 5.0/95	XLS	Формат двоичных файлов Excel 5.0/95 (BIFF5).
XML-таблица 2003	XML	Формат файлов XML-таблиц 2003 (XMLSS).
Данные XML	XML	Формат данных XML.
Настройка Microsoft Excel	XLAM	Формат файлов надстроек Office Excel 2007, обеспечивающих возможность выполнения дополнительного

		кода. Основан на XML и поддерживает макросы. Позволяет использовать проекты VBA и листы макросов Excel 4.0 (XLM).
Надстройка Excel 97 — 2003	XLA	Надстройка Excel 97 — 2003, дополнительная программа, предназначенная для выполнения дополнительного кода. Поддерживает использование проектов VBA.
Книга Excel 4.0	XLW	Формат файла Microsoft Excel 4.0, в котором сохраняются только листы электронных таблиц, диаграмм и макросов. В Office Excel 2007 можно открыть книгу в этом формате, но сохранить в нем файл Excel нельзя.

4 Быстрый ввод

Если Вам часто приходится вводить даты и время в ячейки, то Вам должна понравиться идея писать их сокращенно, без точек-дробей-двоеточий - просто как число. Чтобы в заданном диапазоне ячеек листа, например, число 250699 автоматически превращалось в 25.06.1999, а 1125 в 11:25.

	А	В	С
1	Дата	Время	
2			
3			
4			
5			
6			
7			
8			

Для этого щелкните по ярлычку листа, куда будут вводиться даты и время и выберите команду **Исходный текст (Source Code)**. В открывшееся окно редактора Visual Basic скопируйте и вставьте следующий код:

```

1 Private Sub Worksheet_Change(ByVal Target As Range)
2 Dim vVal
3 Dim StrVal As String
4 Dim dDate As Date
5
6 If Target.Cells.Count > 1 Then Exit Sub
7 If Not Intersect(Target, Range("A2:A10")) Is Nothing Then
8 With Target
9 StrVal = Format(.Text, "000000")
10 If IsNumeric(StrVal) And Len(StrVal) = 6 Then
11 Application.EnableEvents = False
12 dDate = DateValue(Left(StrVal, 2) & "/" & Mid(StrVal, 3, 2) & "/" & Right(StrVal, 2))

```

```

13      .NumberFormat = "dd/mm/yyyy"
14      .Value = CDate(DateSerial(Year(dDate), Month(dDate), Day(dDate)))
15      End If
16      End With
17      End If
18
19      If Not Intersect(Target, Range("B2:B10")) Is Nothing Then
20          With Target
21              vVal = Format(.Value, "0000")
22              If IsNumeric(vVal) And Len(vVal) = 4 Then
23                  Application.EnableEvents = False
24                  .Value = Left(vVal, 2) & ":" & Right(vVal, 2)
25                  .NumberFormat = "[h]:mm"
26              End If
27          End With
28      End If
29      Application.EnableEvents = True
30
31      End Sub

```

Диапазоны A2:A10 и B2:B10 замените на свои области листа, куда будут подобным образом вводиться даты и время, соответственно.

5 Сложный ввод

Заполнение сложных форм, документов, и заявлений (по одной букве в ячейке)

Выглядят заполняемые документы примерно так: (щелкните на картинке для увеличения)

Почтовый индекс	{ и н д е к }	Код региона	6 6
Район	{ р а й о н }		
Город	{ г о р о д }		
Населенный пункт	{ н а с е л е н н ы й п у н к т }		
Улица(проспект, переулок и т.д.)	{ у л и ц а }		

Наша задача - настроить этот шаблон так, чтобы программа заполнения документов подставляла в него текстовые данные, а шаблон сам, при помощи формул, распределял вставленный текст по

отдельным ячейкам.

Самый простой, на мой взгляд, способ добиться желаемого, — подставлять данные в «невидимые» ячейки, а потом, при помощи формулы =ПСТР(), подтягивать отдельные буквы нужные ячейки.

(для ручной вставки текста в подобные бланки есть **специальный макрос**, но в нашем случае он не подойдёт)

Сделать ячейку «невидимой» — очень просто. Для этого достаточно назначить шрифту ячейки белый цвет.

А формулы будут иметь такой вид:

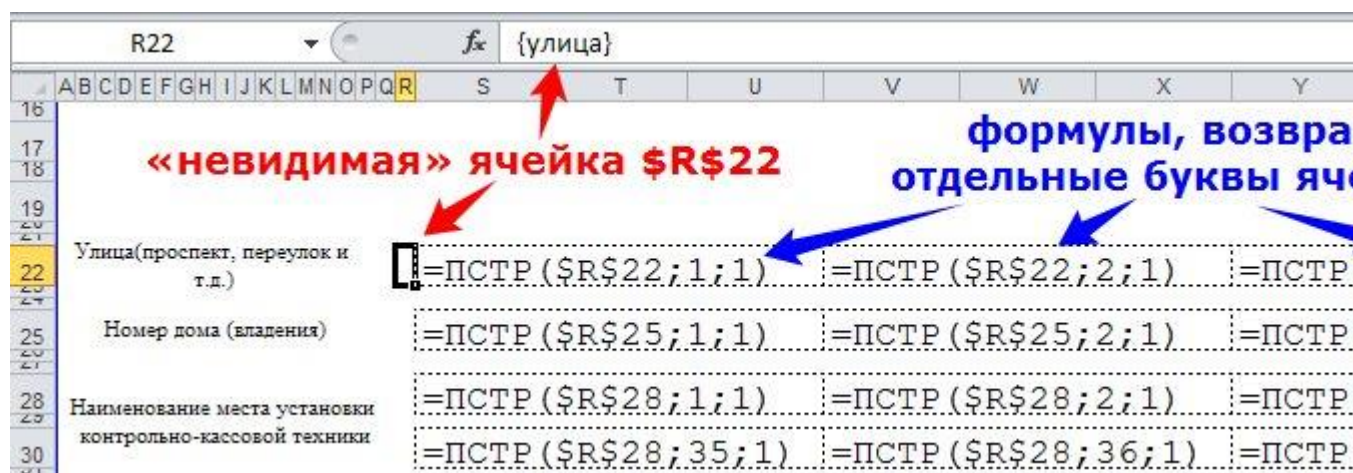
=ПСТР(АдресНевидимойЯчейки;1;1)

=ПСТР(АдресНевидимойЯчейки;2;1)

=ПСТР(АдресНевидимойЯчейки;3;1)

и т.д.

На следующем скриншоте показано, как это сделано: (щелкните на картинке для увеличения)



Теперь осталось выбрать место для «невидимой» ячейки, вписать в неё код поля (например, {улица}), и проставить формулы во все ячейки-клеточки.

Делать это вручную - достаточно долго, да и легко ошибиться при вводе формул.

Для автоматизации настройки подобных шаблонов, в программе

заполнения документов имеется специальная опция.

Включив опцию «Вставлять формулы для разбивки текста по буквам»,

вам станет доступен макрос, который при нажатии Ctrl + Shift + V

1. возьмёт текст из буфера обмена Windows (предполагается, что перед этим вы скопировали код поля в исходной таблице)
2. вставит этот текст в ПЕРВУЮ из выделенных ячеек, назначив этой ячейке белый цвет шрифта
3. пропишет формулы =ПСТР() с нужными параметрами, СО ВТОРОЙ ДО ПОСЛЕДНЕЙ выделенной ячейки

Учебно-методическое и информационное обеспечение

Основная литература:

- 1 Гаврилов, М. В. Информатика и информационные технологии [Электронный ресурс]: учебник для СПО / М. В. Гаврилов, В. А. Климов. — 4-е изд., перераб. и доп. — М. : Издательство Юрайт, 2017. — 383 с. — Режим доступа: <https://www.biblio-online.ru>.
- 2 Илюшечкин, В. М. Основы использования и проектирования баз данных [Электронный ресурс]: учебник для СПО / В. М. Илюшечкин. — испр. и доп. — М. : Издательство Юрайт, 2017. — 213 с. — Режим доступа: <https://www.biblio-online.ru>.
- 3 Казанский, А. А. Объектно-ориентированный анализ и программирование на visual basic 2013 [Электронный ресурс]: учебник для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 290 с. — Режим доступа: <https://www.biblio-online.ru>.
- 4 Казанский, А. А. Прикладное программирование на excel 2013 [Электронный ресурс]: учебное пособие для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 159 с. — Режим доступа: <https://www.biblio-online.ru>.

- 5 Казанский, А. А. Программирование на visual c# 2013 [Электронный ресурс]: учебное пособие для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 191 с. — Режим доступа: <https://www.biblio-online.ru>.
- 6 Семакин И.Г. Основы алгоритмизации и программирования: учеб. для сред. проф.образования/ И.Г.Семакин, А.П. Шестаков. – 3-е изд., стер. – М.: Академия, 2012. – 392 с.
- 7 Советов, Б. Я. Базы данных [Электронный ресурс]: учебник для СПО / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 2-е изд. — М. : Издательство Юрайт, 2017. — 463 с. — Режим доступа: <https://www.biblio-online.ru>.
- 8 Советов, Б. Я. Информационные технологии [Электронный ресурс]: учебник для СПО / Б. Я. Советов, В. В. Цехановский. — 6-е изд., перераб. и доп. — М. : Издательство Юрайт, 2017. — 261 с. — Режим доступа: <https://www.biblio-online.ru>.
- 9 Федорова Г. Н. Разработка и администрирование баз данных : учеб. для студ. учреждений сред. проф. образования / Г. Н. Федорова. - М.: Академия, 2015. - 313 с.
- 10 Цветкова, М.С. Информатика и ИКТ : учеб. для студентов учреждений сред. проф. образования / М. С. Цветкова, Л. С. Великович. - 8-е изд., стер. - М. : Академия, 2016. - 332, [1] с., [4] л. ил.
- 11 Черпаков, И. В. Основы программирования [Электронный ресурс]: учебник и практикум для СПО / И. В. Черпаков. — М. : Издательство Юрайт, 2017. — 219 с. — Режим доступа: <https://www.biblio-online.ru>.

Тема 4.2 Базы данных

План лекции

- 1 База данных как модель информационной структуры. Компьютерная база данных – система организации, хранения, доступа, обработки и поиска информации
- 2 Модель расчета оплаты труда в табличной базе данных

База данных как модель информационной структуры. Компьютерная база данных – система организации, хранения, доступа, обработки и поиска информации

Термин «база данных» появился в 60-х годах XX века. *База данных* (БД) – это любая совокупность связанной информации, объединённой вместе по определённому признаку. Например, в качестве базы данных можно рассматривать расписание движения поездов или книгу регистрации данных о заказах покупателей и выполнении заказов.

Компьютерная база данных – это реализованная с помощью компьютера информационная модель, отражающая состояние объектов и их отношения. Информационной моделью (или структурой данных) называют совокупность взаимосвязанных данных.

Основным назначением БД является быстрый поиск содержащейся в них информации. Ручные или бумажные БД имеют существенный недостаток – их очень трудно модифицировать. Компьютерные БД позволяют устранить многие проблемы, свойственные ручным (бумажным) БД, обладают большой компактностью.

При этом в качестве базы данных рассматриваются только набор данных, организованных определённым образом. Для того, чтобы создать свою собственную базу данных, надо прежде всего решить, из каких элементов она будет состоять (например: фамилия, имя, национальность, год рождения, адрес и т. д.) Из этих элементов можно составить записи, т. е. объединить их. Объединяя элементы в запись, мы устанавливаем между ними определённую связь. Наличие таких связей и определяет *структуру* базы данных.

Базы данных, соответственно типам информационных структур, делят на три класса:

- табличные (реляционные);
- сетевые;
- иерархические.

С понятием базы данных тесно связано понятие **системы управления базой данных (СУБД)**. Это комплекс программных средств, предназначенных для создания структуры новой базы, ее наполнения содержимым, редактирования содержимого и отбора отображаемых данных в соответствии с заданным критерием, упорядочения, оформления и последующей выдачи на устройства вывода или передачи по каналам связи.

В мире существует множество систем управления базами данных: dBase, Paradox, FoxPro, Clipper, Oracle и т. д. Несмотря на то, что они могут по – разному работать с разными объектами и предоставляют пользователю различные функции и средства, большинство СУБД опираются на единый устоявшийся комплекс основных понятий. Это дает возможность рассмотреть одну систему и обобщить ее понятия, приемы и методы на весь класс СУБД. В качестве такой системы мы выбрали СУБД Microsoft Access.

Реляционная база данных. База данных, созданная в СУБД Access, является реляционной базой данных. В основе реляционной модели базы данных лежит понятие отношения (relation). Концепция реляционной модели предложена Е. Ф. Коддом в 1970 г. Основным объектом этой базы являются взаимосвязанные двумерные таблицы. Таблица понятна, обозрима и привычна для человека. Каждая таблица состоит из строк и столбцов, которые в компьютерных базах данных называются записями и полями, соответственно. Изменив состав полей базовой таблицы, мы изменяем структуру базы данных и, соответственно, получаем новую базу данных.

Для однозначного определения каждой записи таблица должна иметь уникальный ключ (первичный ключ). Этот ключ может состоять из одного или нескольких полей. По значению ключа отыскивается единственная запись.

Связи между таблицами базы данных дают возможность совместно использовать данные из разных таблиц. В нормализованной реляционной базе данных связи характеризуются отношениями типа один-к-одному (1:1) или один-ко-многим (1:¥). Связь каждой пары таблиц обеспечивается одинаковыми полями в них – ключом связи. Ключом связи всегда является уникальный ключ главной таблицы в связи. В подчиненной таблице он называется внешним ключом.

Схема данных. В СУБД Access процесс создания реляционной базы данных включает создание схемы данных. Схема данных наглядно отображает таблицы и связи между ними и обеспечивает использование связей при обработке данных. В схеме данных устанавливаются параметры обеспечения связной целостности в базе данных.

Поскольку СУБД Access является одним из приложений Windows, входящих в интегрированную систему Office, интерфейс окна программы и его основные компоненты – меню, панели инструментов, справочная система, а также приемы работы с клавиатурой и мышью используются в работе аналогично другим приложениям (Word, Excel).

Поля базы данных не просто определяют структуру базы – они еще определяют групповые свойства данных, записываемых в ячейки, принадлежащие каждому из полей.

Основные свойства полей таблиц баз данных СУБД Microsoft Access:

Имя поля – определяет, как следует обращаться к данным этого поля при автоматических операциях с базой (по умолчанию имена полей используются в качестве заголовков столбцов таблиц);

Тип поля – определяет тип данных, которые могут содержаться в данном поле;

Размер поля – определяет предельную длину (в символах) данных в ячейках, принадлежащих полю;

Маска ввода – определяет форму, в которой вводятся данные в поле (средство автоматизации ввода данных);

Подпись – определяет заголовок столбца таблицы для данного поля (если подпись не указана, то в качестве заголовка столбца используется свойство Имя поля);

Значение по умолчанию – то значение, которое вводится в ячейки поля автоматически (средство автоматизации ввода данных);

Условие на значение – ограничение, используемое для проверки правильности ввода данных;

Сообщение об ошибке – текстовое сообщение, которое выдается автоматически при попытке ввода в поле ошибочных данных;

Обязательное поле – свойство, определяющее обязательность заполнения данного поля при наполнении базы;

Пустые строки – свойство, разрешающее ввод пустых строковых данных (в основном это касается текстовых данных);

Индексированное поле – если поле обладает этим свойством, все операции, связанные с поиском или сортировкой записей по значению, хранящемуся в данном поле, существенно ускоряются. По этому полю так же проверяются значения записей на наличие повторов.

Свойства полей различаются в зависимости от типа данных. Базы данных Microsoft Access работают со следующими типами данных:

Текстовый – тип данных, используемый для хранения обычного неформатированного текста ограниченного размера (до 255 символов);

Поле Мемо – специальный тип для хранения больших объемов текста (до 65535 символов). Физически в поле хранится указатель на другое место базы данных, в котором хранится непосредственно текст, но для пользователя такое разделение незаметно;

Числовой – тип данных для хранения действительных чисел;

дата/время – тип данных для хранения календарной даты и текущего времени;

денежный – тип данных для хранения денежных сумм;

счетчик – специальный тип данных для хранения уникальных (не повторяющихся в поле) натуральных чисел с автоматическим наращиванием;

логический – тип для хранения логических данных (могут принимать только два значения, например, да или нет);

поле объекта OLE – специальный тип данных, предназначенный для хранения объектов OLE, например, мультимедийных;

гиперссылка – специальное поле для хранения адресов URL WEB-объектов Интернета. При щелчке на ссылке автоматически происходит запуск броузера и воспроизведение объекта в его окне.

Перед созданием базы данных должна быть определена логическая структура базы – состав таблиц, их структура и межтабличные связи.

Объектами базы данных помимо таблиц являются также **запросы, формы, отчеты, макросы и модули**, создание которых существенно упрощает работу пользователя с массивами данных.

Формы используются для ввода и просмотра таблиц в окне формы. Формы позволяют ограничить объем информации, отображаемой на экране, и представить её в требуемом виде. С помощью Мастера можно создать форму, поместив в неё поля исходной таблицы, расположенные в соответствии с одним из заранее созданных шаблонов. С помощью конструктора форм можно создавать формы любой степени сложности.

Запрос является средством извлечения информации из базы данных, причём данные могут быть распределены среди нескольких таблиц. Результатом выполнения запроса является таблица, которая может быть использована наряду с другими таблицами базы данных при обработке данных. В Microsoft Access для формирования запросов используется способ, получивший название запроса по образцу. Используя это средство, на основании визуальной информации можно извлечь нужные данные из одной или нескольких таблиц.

Отчет формируется для создания бумажного документа, т.е. для распечатки данных.

Макросы предназначены для автоматизации часто выполняемых операций. Каждый макрос содержит одну или несколько макрокоманд, каждая из которых выполняет определённое действие, например, открывает форму или печатает отчёт.

Для реализации практических задач пользователя разработчику приходится использовать средства программирования: язык макрокоманд и язык Visual Basic for Applications (VBA). **Макросы и модули**, созданные программистом, оперируют запросами, формами и отчётами и объединяют разрозненные действия в единую задачу пользователя.

В окне базы данных Access наряду со списком объектов представлены ярлыки для быстрого запуска Мастеров или Конструкторов создания нового объекта.

Мастера Access позволяют автоматизировать процесс создания таблиц базы данных, форм, запросов, отчётов и страниц доступа к данным.

Размещение базы данных. Все таблицы базы данных, а также другие объекты Access – формы, запросы, отчёты, макросы и модули, построенные для этой базы, и внедренные объекты – могут размещаться на диске в одном файле базы данных формата .mdb. Это упрощает технологию ведения базы данных и приложения пользователя. Обеспечивается высокая компактность размещения всех объектов базы данных. Приложение базы данных которая создается программой VBA, может быть скомпилирована и сохранено в файле приложения формата .mde. При этом исходные программы на VBA удаляются, а база данных сжимается, что значительно сокращает размер файла. После компиляции объекты базы данных не могут быть изменены.

При работе с базой данных коллективного пользования в сети с файловым сервером Access предоставляет возможность записать в отдельный файл от базы данных на сервере те объекты, которые составляют приложение пользователя. Этот файл размещается на всех компьютерах пользователей, работающих с общей базой данных. Приложение можно модифицировать в соответствии с потребностями пользователя.

В Access включены средства разработки проекта – приложения, обеспечивающего работу с базой данных, размещенной на SQL – сервере. Проект размещается в файле .adp на компьютере пользователя. При создании проекта пользователь может создать базу данных на SQL – сервере или использовать уже существующую.

Учебно-методическое и информационное обеспечение

Основная литература:

- 1 Гаврилов, М. В. Информатика и информационные технологии [Электронный ресурс]: учебник для СПО / М. В. Гаврилов, В. А. Климов. — 4-е изд., перераб. и доп. — М. : Издательство Юрайт, 2017. — 383 с. — Режим доступа: <https://www.biblio-online.ru>.
- 2 Илюшечкин, В. М. Основы использования и проектирования баз данных [Электронный ресурс]: учебник для СПО / В. М. Илюшечкин. — испр. и доп. — М. : Издательство Юрайт, 2017. — 213 с. — Режим доступа: <https://www.biblio-online.ru>.
- 3 Казанский, А. А. Объектно-ориентированный анализ и программирование на visual basic 2013 [Электронный ресурс]: учебник для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 290 с. — Режим доступа: <https://www.biblio-online.ru>.
- 4 Казанский, А. А. Прикладное программирование на excel 2013 [Электронный ресурс]: учебное пособие для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 159 с. — Режим доступа: <https://www.biblio-online.ru>.
- 5 Казанский, А. А. Программирование на visual c# 2013 [Электронный ресурс]: учебное пособие для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 191 с. — Режим доступа: <https://www.biblio-online.ru>.
- 6 Семакин И.Г. Основы алгоритмизации и программирования: учеб. для сред. проф.образования/ И.Г.Семакин, А.П. Шестаков. – 3-е изд., стер. –

- М.: Академия, 2012. – 392 с.
- 7 Советов, Б. Я. Базы данных [Электронный ресурс]: учебник для СПО / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 2-е изд. — М. : Издательство Юрайт, 2017. — 463 с. — Режим доступа: <https://www.biblio-online.ru>.
 - 8 Советов, Б. Я. Информационные технологии [Электронный ресурс]: учебник для СПО / Б. Я. Советов, В. В. Цехановский. — 6-е изд., перераб. и доп. — М. : Издательство Юрайт, 2017. — 261 с. — Режим доступа: <https://www.biblio-online.ru>.
 - 9 Федорова Г. Н. Разработка и администрирование баз данных : учеб. для студ. учреждений сред. проф. образования / Г. Н. Федорова. - М.: Академия, 2015. - 313 с.
 - 10 Цветкова, М.С. Информатика и ИКТ : учеб. для студентов учреждений сред. проф. образования / М. С. Цветкова, Л. С. Великович. - 8-е изд., стер. - М. : Академия, 2016. - 332, [1] с., [4] л. ил.
 - 11 Черпаков, И. В. Основы программирования [Электронный ресурс]: учебник и практикум для СПО / И. В. Черпаков. — М. : Издательство Юрайт, 2017. — 219 с. — Режим доступа: <https://www.biblio-online.ru>.

РАЗДЕЛ 5 ТЕЛЕКОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ

Тема 5.1 Компьютерные сети

План лекции

- 1 Компьютерная сеть как средство массовой коммуникации
- 2 Виды компьютерных сетей

Компьютерная сеть как средство массовой коммуникации

Сети передачи данных или компьютерные сети появились в результате конвергенции компьютерных и телекоммуникационных технологий.

Компьютерная сеть – это набор компьютеров, связанных коммуникационной системой и снабженных соответствующим программным обеспечением (сетевая операционная система), которое позволяет пользователям сети работать автономно или получать доступ к информационным и аппаратным ресурсам сети. Сеть могут образовывать компьютеры различных типов и разной производительности – суперкомпьютеры, мейнфреймы, персональные компьютеры, и т.д. *Коммуникационная система* включает в себя каналы связи, образованные через различные физические среды передачи, и сетевые устройства – маршрутизаторы, коммутаторы, мосты, повторители и пр.

В процессе обмена информацией между компьютерами в сети пересылаются *короткие сообщения строго определенного формата*, которые называются **пакетами** (или **кадрами**).

Компьютер – это совокупность аппаратных средств (hardware), способная выполнять определенную последовательность операций, предписанных встроенным программным обеспечением (firmware) и загружаемым программным обеспечением (software).

Общее руководство всеми действиями в компьютере осуществляет операционная система (ОС). ОС – это базовое программное обеспечение, которое руководит распределением и использованием всех аппаратных и программных ресурсов компьютера, то есть обеспечивает взаимодействие и функционирование всех его компонентов.

Сетевая операционная система дополнительно обеспечивает обработку, хранение и передачу данных в компьютерной (информационной) сети. В ее функции входит обеспечение адресации объектов, функционирование сетевых служб, обеспечение безопасности данных, управление сетью.

Буфер – это устройство для промежуточного хранения данных. Обычно используется для компенсации разницы в скорости обработки информации.

Приложение или **Прикладной процесс пользователя** – это различные процедуры ввода, хранения, обработки и выдачи информации, выполняемые в интересах пользователей и описываемые прикладными программами.

Компьютерная сеть организации выполняет следующие задачи:

- обеспечивает **совместное использование** аппаратных и информационных **ресурсов** (принтеры, сканеры, базы данных) любыми (авторизованными) пользователями сети, независимо от физического расположения ресурса или пользователя.
- создает коммуникационную среду для работников предприятия: общение при помощи электронной почты e-mail, создание совместных документов в режиме on-line, проведение видеоконференций.
- обеспечивает возможность интернет-коммерции, которая является перспективной и быстро развивающейся областью.

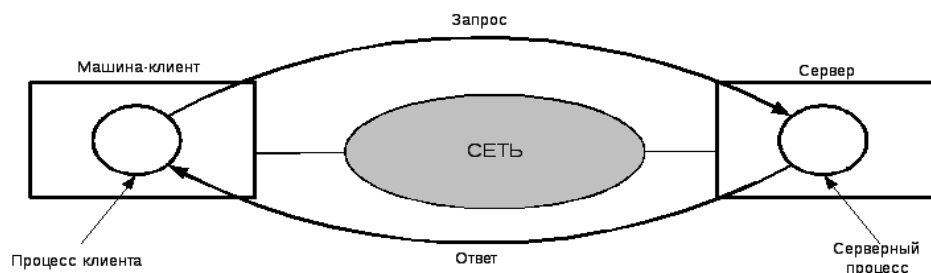
Взаимодействие процессов в компьютерных сетях выполняется в соответствии с моделью «клиент-сервер».

Серверы – это мощные вычислительные машины, управляющие определенным видом ресурсов сети. Этот же термин используется для обозначения программного обеспечения (soft, софт), установленного на этой машине. Различают файл-серверы, серверы приложений, факс-серверы, почтовые серверы, WEB-серверы и др. Часто сервер располагается в отдельном помещении и обслуживается системным администратором.

Менее мощные компьютеры пользователей идентифицируются в сети, как **клиенты**. Они могут размещаться в пределах офиса или иметь удаленный доступ к информации и программам, хранящимся на сервере. Клиентскую машину часто называют рабочей станцией.

Рабочие станции – это ЭВМ, которые используются профессиональными пользователями для работы с сетевыми ресурсами. Этот термин отделяет их от персональных компьютеров, обеспечивающих работу основной массы

непрофессиональных пользователей, и которые обычно работают в автономном режиме.



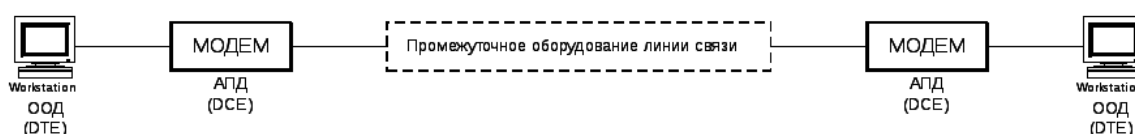
Модель «клиент-сервер»

В соответствии с этой моделью в работе сети выделяют два процесса: серверный и клиентский. Программа-клиент посылает запрос серверу через сеть и ожидает ответ. При принятии запроса программа-сервер выполняет определенные вычисления или ищет запрашиваемые данные, и затем отправляет ответ. *Эта же модель* используется, когда пользователь получает доступ к Интернет-сайту. При этом WEB-сервер играет роль серверной машины, а пользовательский компьютер – роль клиентской машины.

Для подключения к среде передачи сетевого устройства – компьютера пользователя, рабочей станции, сетевого сервера – используется *сетевой адаптер* (сетевая интерфейсная карта, network interface card - NIC) и *внешние интерфейсы* (разъемы, порты). К основным функциям адаптера относятся:

1. Гальваническая развязка со средой передачи – кабелем.
2. Кодирование и декодирование сигналов.
3. Идентификация своего физического адреса в принимаемом пакете.
Физический адрес адаптера хранится в специальном регистре или прошит в постоянной памяти ППЗУ (PROM).
4. Преобразование параллельного кода в последовательный код при передаче и обратное преобразование при приеме.
5. Промежуточное хранение данных и служебной информации в буфере.
Согласование скорости пересылки данных между компьютером и адаптером и скорости передачи в сети.

6. Выявление конфликтных ситуаций в сети и контроль состояния сети. Эта функция наиболее важна в сети с топологией «шина» и со случайным методом доступа к среде передачи. Возможные конфликты адаптер должен разрешать самостоятельно.
7. Подсчет контрольной суммы, которая используется для обнаружения ошибок в принятом блоке данных.



В общем случае в комплексе аппаратно-программных средств, подключаемых к среде передачи можно выделить:

- DTE (data terminal equipment) – оконечное оборудование данных, выполняющее функции ввода/вывода, обработки и хранения данных. Обычно это компьютер, где работает приложение.
- DCE (data circuit-terminating equipment) – аппаратура передачи данных (сетевой адаптер рабочей станции в локальной сети, модем).

На DCE возлагаются функции преобразования сигнала в вид, подходящий для передачи по линии связи (формат среды передачи). Фактически DCE служит интерфейсом между компьютером и средой передачи (например, между компьютером и оптоволокном, или между компьютером и телефонной линией).

Модем – это специальный тип DCE-устройства, применяемый для передачи данных от DTE в среду, эквивалентную полосовому фильтру, с ограничением полосы пропускания сверху и снизу (например, от компьютера в телефонную линию или от компьютера микроволновому передатчику). Модем модулирует сигнал *аналоговой* несущей последовательностью данных и выполняет обратное преобразование.

В сетях передачи данных используется *узкополосная* и *широкополосная* передача:

Узкополосная система (baseband) использует цифровой способ передачи сигнала. Передача ведется в *исходной полосе* частот, не происходит переноса спектра сигнала в другие частотные области. Именно в этом смысле система называется узкополосной.

Цифровой сигнал имеет широкий спектр и теоретически занимает бесконечную полосу частот. На практике ширина спектра передаваемого сигнала определяется частотами его *основных гармоник*, которые дают основной энергетический вклад в формирование сигнала. Сигнал занимает практически всю полосу пропускания линии.

Узкополосная передача используется в сетях с технологией Ethernet (более 95% всех существующих локальных сетей).

Широкополосная система (broadband) использует для передачи данных аналоговые несущие и разделяет полосу пропускания линии связи на отдельные подканалы. Широкополосная система обеспечивает передачу мультимедийной информации - передачу данных на скорости 128 Кбит/с и более, голоса, видео. Примером являются сети xDSL-доступа в Интернет.

Виды компьютерных сетей

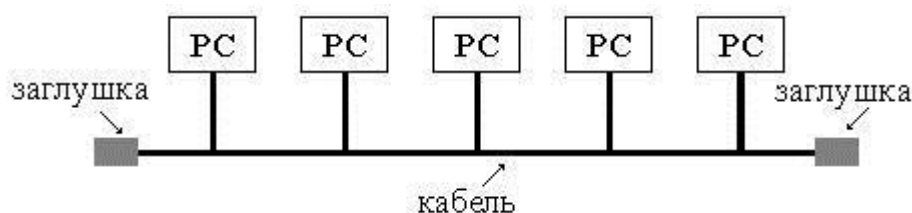
1. **Глобальная сеть** – здесь может быть неограниченное число пользователей и неограниченные расстояния между ними. Скорость передачи данных здесь невелика (т. к. большое расстояние и много ПК, а также из-за плохой надежности сети приходится повторно предавать данные). Сеть Internet является глобальной.
2. **Локальные сети** – связывают ограниченное число ПК. Здесь более качественная связь между ПК, чем в глобальных сетях. Обычная скорость в ЛВС до 10 Мбит/с и выше. Это достигается применением кабелей, в том числе оптоволоконных. Надежность связи в ЛВС должна быть выше, т. к. при наличии сбоев в работе сети исправление ошибок достигается путем

повторной передачи данных, что уменьшает скорость передачи информации в сети в целом.

3. **Корпоративные сети** – объединяют большое количество ПК на всех территориях отдельного предприятия. Они могут быть сложно связаны, и покрывать город, регион и даже континент. Для связи между ПК в таких сетях применяются телефонные каналы, радиоканалы, спутниковая связь.

4. *Топология локальных компьютерных сетей*

1. *Общая шина*



При использовании шинной топологии компьютеры (PC – рабочая станция) соединяются в одну линию, по концам которой устанавливают терминаторы (заглушки). Терминаторы представляют собой резисторы, устанавливаемые на обоих концах сегмента для согласования волнового сопротивления кабеля. Сигнал, дошедший до конца сегмента, поглощается терминатором - это позволяет избавиться от паразитных отраженных сигналов в сети. Если терминаторы не устанавливать, отраженный от конца кабеля сигнал снова попадает в кабель - этот отраженный сигнал будет являться в данном случае помехой и может породить множество проблем вплоть до полной неработоспособности сети. Преимущества шинной топологии заключаются в простоте организации сети и низкой стоимости. Недостатком является низкая устойчивость к повреждениям - при любом обрыве кабеля вся сеть перестает работать, а поиск повреждения весьма затруднителен

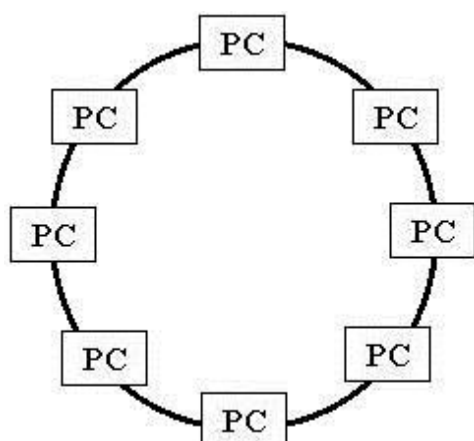
2. *Топология Звезда*



При использовании топологии "звезда", каждый компьютер подключается к специальному концентратору (хабу). Преимуществом этой топологии является ее устойчивость к повреждениям кабеля - при обрыве перестает работать только один из узлов сети и поиск повреждения значительно упрощается. Недостатком является более высокая стоимость из-за наличия HUB. Количество рабочих станций в этой схеме определяется конструкцией хаба.

3. Топология Кольцо

При такой топологии узлы сети образуют виртуальное кольцо (концы кабеля соединены друг с другом). Каждый узел сети соединен с двумя соседними. Эту топологию активно продвигает фирма IBM (сети Token Ring). Преимуществом кольцевой топологии является ее высокая надежность (за счет избыточности), однако стоимость такой сети достаточно высока за счет расходов на адаптеры, кабели и дополнительные приспособления.



- Данные между PC пересылаются по кольцу. Каждый ПК в кольце вынужден пересылать все данные, даже не адресованные конкретно этому ПК. Получается регенерация сигнала на каждом ПК, при которой

сигнал усиливается и компенсируется его затухание . При этом размеры сети могут достигать нескольких десятков километров.

Недостатки:

- Отказ в работе любой РС ведет к отказу сети.
- Для настройки и переконфигурации одного ПК, придется отключать сеть.
- Трудно поддерживать сеть, если кольцевая схема большая.

Учебно-методическое и информационное обеспечение

Основная литература:

- 1 Гаврилов, М. В. Информатика и информационные технологии [Электронный ресурс]: учебник для СПО / М. В. Гаврилов, В. А. Климов. — 4-е изд., перераб. и доп. — М. : Издательство Юрайт, 2017. — 383 с. — Режим доступа: <https://www.biblio-online.ru>.
- 2 Илюшечкин, В. М. Основы использования и проектирования баз данных [Электронный ресурс]: учебник для СПО / В. М. Илюшечкин. — испр. и доп. — М. : Издательство Юрайт, 2017. — 213 с. — Режим доступа: <https://www.biblio-online.ru>.
- 3 Казанский, А. А. Объектно-ориентированный анализ и программирование на visual basic 2013 [Электронный ресурс]: учебник для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 290 с. — Режим доступа: <https://www.biblio-online.ru>.
- 4 Казанский, А. А. Прикладное программирование на excel 2013 [Электронный ресурс]: учебное пособие для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 159 с. — Режим доступа: <https://www.biblio-online.ru>.
- 5 Казанский, А. А. Программирование на visual c# 2013 [Электронный ресурс]: учебное пособие для СПО / А. А. Казанский. — М. : Издательство Юрайт, 2017. — 191 с. — Режим доступа: <https://www.biblio-online.ru>.
- 6 Семакин И.Г. Основы алгоритмизации и программирования: учеб. для

- сред. проф. образования/ И.Г.Семакин, А.П. Шестаков. – 3-е изд., стер. – М.: Академия, 2012. – 392 с.
- 7 Советов, Б. Я. Базы данных [Электронный ресурс]: учебник для СПО / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 2-е изд. — М. : Издательство Юрайт, 2017. — 463 с. — Режим доступа: <https://www.biblio-online.ru>.
 - 8 Советов, Б. Я. Информационные технологии [Электронный ресурс]: учебник для СПО / Б. Я. Советов, В. В. Цехановский. — 6-е изд., перераб. и доп. — М. : Издательство Юрайт, 2017. — 261 с. — Режим доступа: <https://www.biblio-online.ru>.
 - 9 Федорова Г. Н. Разработка и администрирование баз данных : учеб. для студ. учреждений сред. проф. образования / Г. Н. Федорова. - М.: Академия, 2015. - 313 с.
 - 10 Цветкова, М.С. Информатика и ИКТ : учеб. для студентов учреждений сред. проф. образования / М. С. Цветкова, Л. С. Великович. - 8-е изд., стер. - М. : Академия, 2016. - 332, [1] с., [4] л. ил.
 - 11 Черпаков, И. В. Основы программирования [Электронный ресурс]: учебник и практикум для СПО / И. В. Черпаков. — М. : Издательство Юрайт, 2017. — 219 с. — Режим доступа: <https://www.biblio-online.ru>.

Тема 5.2 Интернет

План лекции

- 1 Интернет страница и редакторы для ее создания
- 2 Личные сетевые сервисы в Интернет
- 3 Пример работы в телеконференциях на основе Skype
- 4 Сетевая этика и культура

Интернет страница и редакторы для ее создания

Веб-страница – это фактически минисайт. Поэтому к ней предъявляются те же требования, что и к сайту.

Интернет-страница может сэкономить время на общение.

Размещение информации на интернет-странице

При размещении информации на web странице нет таких ограничений на объем, как на печатных площадях газет и журналов, в эфире радио и телевидения. Можно разместить сколько угодно много материалов сколько угодно большого объема.

Информацию на странице очень легко редактировать, обновлять. Причем замена информации может проводиться в любое время суток.

Что очень важно, размещение информации на веб странице – это достаточно недорогой способ ее распространения. При этом информация становится доступной для аудитории, находящейся в любом конце света.

Интернет-страница должна иметь как качественное содержание, так и качественное же оформление, техническую поддержку.

Содержание веб страницы направлено в первую очередь на определенную аудиторию, являющуюся покупателями товаров или услуг компании.

Контент интернет-страницы

С точки зрения основного содержания для всех посетителей web страницы на ней должна быть следующая информация:

- профиль компании (оборот, прибыль, структура, штат, перечень товаров или услуг, регион, регистрационные свидетельства, лицензии и т.д.),
- история компании (год основания, учредители, динамика развития, интересные факты и т.д.),

- описание и характеристики товаров и услуг,
- применяемые технологии,
- сырье,
- условия сотрудничества (оптовые и розничные цены, гарантия, сервисное обслуживание и т.д.),
- фирменный стиль (логотип, товарный знак, слоган и т.д.),
- заявление о миссии компании ее первого лица,
- биографии руководителей компании,
- контактные адреса, телефоны, ФИО основных менеджеров (с фотографиями),
- новости компании,
- новости региональных представительств (партнеров),
- подборка официальных документов,
- годовые отчеты компании,
- электронные презентации новых продуктов,
- данные тестов,
- отзывы специалистов,
- отзывы потребителей,
- публикации СМИ,
- ссылки на сайты партнеров и клиентов,
- часто задаваемые вопросы (FAQ – Frequently asked questions) и ответы на них;
- обратная связь (адрес обычной и электронной почты, телефон, факс),
- основная информация на английском языке (на других языках),
- и т.д.

Естественно, указанные телефоны и адреса должны быть действующими.

Подача контента на интернет-странице

Качественной интернет-странице следует не только содержать важную, полезную информацию, но и быть удобной. На ней должна соблюдаться четкая логика подачи материалов, ссылок: главное, свежее – сверху, дополнительное – ниже, дальше.

Не стоит просто переносить печатные тексты в интернет-среду, у электронных материалов есть своя специфика чтения.

Хотя у электронных изданий нет таких ограничений по объему, как у бумажных, не следует слишком растягивать материалы. Человек должен получать достаточное, но не излишнее количество информации. Считается, что на чтение интернет текста уходит на 50% больше времени, чем на обычный печатный. Поэтому на web странице большинство материалов просматриваются и только незначительная часть из них прочитывается.

Соответственно, все материалы на веб странице рекомендуется делать на 50% короче традиционных печатных аналогов. Каждый отдельный материал должен вмещаться в экран компьютера и состоять из нескольких абзацев.

Текстовые файлы лучше давать не одним массивом, а разбивать на удобные для чтения куски.

Интернет-страница не должна много «весить», иначе они долго загружаются, и человек может передумать их читать, перейдет на другую web страницу. Это зависит прежде всего от количества иллюстраций.

Также как в обычном печатном тексте, на интернет-странице не должно быть большого количества разных шрифтов и выделений, от которых рябит в глазах (тем более что не известно, как они отобразятся на мониторе посетителя).

При создании веб страницы не стоит использовать так называемые «фреймы» – они отображаются не у всех пользователей. По этой же

причине лучше использовать не самые «продвинутые», а наиболее популярные программы без «навороченной» анимации.

Любой традиционно скучный печатный документ, типа отчета, в электронной среде можно оживить фотографиями, видео, звуками, вставками интервью и т.д.

Действительно творчески оформленные интернет-страницы безусловно привлекают внимание, но страницы (подчеркнем еще раз) не должны быть слишком «тяжелыми».

Не стоит размещать не только слишком большие тексты, тяжелую графику, но и большой набор фотографий. Фотографии лучше давать маленького размера, предоставляя человеку возможность при необходимости увеличить ее самостоятельно.

Используемое видео должно быть как можно «легче» с минимумом смен кадров, изменений цвета и т.д.

Дизайн web страницы должен соответствовать содержанию и работать на общий имидж компании. Так на каждой странице веб страницы должны присутствовать логотип и название компании.

По окончании разработки интернет-страницы стоит посмотреть на него глазами посетителей. Это может помочь выявить недостатки и улучшить как содержательную, так и оформительскую стороны.

Изначально следует заложить такие технические принципы, чтобы в дальнейшем необходимые изменения можно было вносить легко, без переустройства всей web страницы.

У веб страницы должен быть очень простой интернет адрес.

Хорошо, когда название интернет-страницы совпадает в написании с названием компании. Тогда оно лучше запоминается и легче находится пользователями.

По этой же причине желательно, чтобы название web страницы было коротким. Возможно подойдет узнаваемое и принятое широким кругом читателей сокращение (например, «НР» – «Hewlett Packard»).

Личные сетевые сервисы в Интернет

Личные сетевые сервисы принадлежат конкретному пользователю, защищены логином и паролем, доступ к ним осуществляется только персонально владельцем сервиса. Примером такого сервиса может служить личный ящик электронной почты, личный сайт или раздел (блог), личный телефонный номер и пароль для его использования, личная страница на сайте дистанционного обучения или дистанционного тестирования. Все личные сервисы в Интернете требуют *персонализации*, т.е. регистрации со стороны пользователя, по итогам чего пользователь получает личный логин и пароль и пр.

Личная информационная деятельность в Интернете имеет в основном поисковый и коммуникативный характер. Рассмотрим коммуникативный вид сетевой деятельности на примере электронной почты, а поисковый — на примере работы пользователя с поисковыми серверами.

Для того чтобы с кем то переписываться по электронной почте, нужно иметь свой почтовый ящик. Его можно завести на любом поисковом или почтовом сервере. Теперь на многих серверах общего назначения имеется почта.

В почтовом сервисе предлагаются папки сортировки сообщений: «Входящие», «Сомнительные», «Отправленные», «Черновики», «Корзина». В этих папках можно хранить письма, а по желанию — очищать их. Необходимо отметить, что непрочитанные письма отмечаются более темным цветом.

Используя электронную почту, часто невозможно передать файл большого размера. Как правило, современные почтовые серверы накладывают ограничение на размер пересылаемого файла — 5 или 10 Мбайт. Для обмена

большими файлами многими провайдерами социальных сервисов в Интернете предоставляется услуга *файлообмена*.

Каждый пользователь Интернета использует поисковые сервисы. Для выхода в Интернет потребуется специальная программа-браузер. Самыми популярными на сегодняшний день программами-браузерами являются Internet Explorer и Netscape Navigator.

В верхней части панели браузера находится меню. Среди поисковых серверов популярны www.google.ru, <http://list.mail.ru/>, www.yahoo.com, www.afort.ru и др. Современные поисковые сервисы включают в свою сферу новости, репортажи, телепередачи, интернет-СМИ. Примером такого расширенного поисково-информационного сервиса является www.km.ru. Информацию о последних новостях можно получить на сайтах www.utro.ru, www.news.ru, www.gaseta.ru, информацию о новостях бизнеса — на www.rbc.ru, о погоде на www.pogoda.ru. Не только новости, но и видеорепортажи доступны для пользователя Интернета.

Очень интересная возможность — использование *web-камер*, которые с определенной периодичностью, измеряемой иногда секундами, передают изображение по линиям Интернета.

Одна из форм личного сервиса — регистрация и предоставление тезисов для *конференции*, в том числе видео. Существуют конференции самой разнообразной тематики: общественные, технические, медицинские, образовательные, научные и т.д.

Популярным личным сетевым сервисом стали личные сайты — *блоги*. Можно создать свой блог, например, зарегистрировавшись на сайте www.pedsovet.org.

Важнейшее место в личных сетевых сервисах занимает *дистанционное обучение*. Оно позволяет выбрать свою траекторию обучения, а среда сформирует для вас журнал успеваемости.

Сетевая этика и культура

Итак, начало Интернет было положено 5 декабря 1969 г., когда три компьютера в Калифорнии (Лос-Анджелес, Санта-Барбара, Менлоу-Парк) и один компьютер в штате Юта были соединены друг с другом. Этот момент можно считать официальным началом ARPAnet - сети, которая в итоге мутировала в Интернет.

На сегодняшний день в мире существует около 2 млн. серверов и по последним данным около 260 млн. пользователей в Интернет. По прогнозам ведущих компаний Великобритании, 20% их продаж в 2000 году будет сделано непосредственно через Интернет... Таким образом Интернет становится тотальным в планетарных масштабах источником всевозможнейшей информации и средой интерактивного общения. Если ЭТОГО нет в Интернете, значит, ЭТОГО не существует в природе!

За время существования Сети появился огромный, совершенно самостоятельный, неимоверными темпами набирающий в весе социокультурный пласт, ей посвященный, ею же сгенерированный. Здесь есть все необходимое для того, чтобы попытаться в нем (или с его помощью) существовать: наука, философия, искусство, этика, религия, эзотерика, экономика, виртуальный секс, хакеры, "Ассоциация подпольной музыки в Интернете"... Короче, здесь есть абсолютно ВСЕ.

Интернет – это большая сложная транспортно-информационная система из грибовидных (дипольных) структур, шляпка каждой из которых (собственно диполи) представляет собой мозг человека, сидящего за компьютером, в совокупности с самим компьютером, который как бы является искусственным продолжением мозга, а ножки - например, телефонная сеть, соединяющая компьютеры, или эфир, через который передаются радиоволны. В отличие от многих других объектов природы Интернет является уникальным объектом, что создает дополнительные сложности в его исследовании.[2]

Итак, Интернет - объект природы. Причем уникальный. Занятно! Тем более что за этой глобальной экспансией Сети никто не стоит, Интернет никем

не контролируется и сознательно никем никуда не направляется в своем развитии.

Если бы в Сети существовали свои собственные реальные формы жизни, то для части представителей этих сетевых форм жизни белковое человечество выступало бы в качестве Творца (или Внешнего Ненаблюдаемого Наблюдателя). Кстати, не такое уж и невероятное допущение.

Многие провидцы, как их называют, принципиально не исключают самой возможности возникновения жизни собственно в Сети. Так, Стивен Хокинг, выдающийся современный английский астрофизик и космолог, еще пять лет назад заявил, что компьютерные вирусы следует считать одной из форм жизни. По его мнению, хотя компьютерные вирусы не имеют собственного "обмена веществ", они вполне могут "паразитировать на зараженном компьютере, используя его метаболизм". Точно так же настоящие вирусы паразитируют на пораженных ими организмах. "Единственная созданная человеком форма жизни, - подчеркнул Хокинг, имея в виду компьютерные вирусы, - служит исключительно для разрушения. Мы создали жизнь по своему образу и подобию".

Интернет как физическое явление развивается самостоятельно, то есть саморазвивается. Понять тип динамики этого процесса еще предстоит.

Естественно, здесь же поднимается вопрос и этики в Интернете. У нового пользователя сети кажущееся отсутствие этики вызывает беспокойство. В действительности сеть есть общество очень этичное, просто лишь этика его слегка отличается от обычной. Вернее будет сказать, что этика в сети слегка опережает в своём развитии этику обычного общества.

Сеть, являясь передним краем внедрения новых информационных и коммуникационных технологий, пользуется правом изменения и нововведения. Выделяются зачастую две пересекающиеся посылки сетевой этики, два ее принципиальных аспекта:

- ü проявление индивидуальности уважается и поощряется;
- ü Сеть – это действительно хорошо и ее следует защищать.

На пользователях Интернет это сказывается следующим образом.

В обычном обществе каждый может претендовать на индивидуализм, т.е. личную свободу, но во многих случаях индивидуализм должен идти на компромисс с интересами определенной общности, слоя, достаточно большой социальной группы.

Та же ситуация имеет место и в сети. Вы общаетесь, когда вам хочется и как вам хочется - разъездов не требуется. География уже не имеет значения. Ваш собеседник может быть в сети где угодно (и почти где угодно в мире). Поэтому возможно любое совершенно необычное сообщество. Возможно даже создание конкурирующих групп. Одни группы могут выбрать «встречи» по e-mail, другие - по доскам объявлений, третьи - через создание общих файлов, а некоторые еще как-нибудь. Люди свободны действовать так, как им нравится. Уже не нужно присоединяться к большой группе, чтобы обладать критической массой, и поэтому каждый может входить в некоторую более мелкую группу. Гонения за взгляды и вкусы строго воспрещены, за такие гонения любого могут выставить из сети. Никто поэтому не скажет, что так делать нельзя. Ценность информации для человека - личное дело этого человека: ею он может жить и за нее умереть, другой же не даст за нее и ломанного гроша. В открытом свободном обществе это должен понимать каждый, но все же многие пользователи сети, увы, очень нервны, что иногда накаляет страсти и, тем самым, создает неудобства другим пользователям, что снижает полезность Сети.

Индивидуализм, в то же время, имеет и обратную сторону. Благодаря ему сеть является прекрасным местом для поиска разнообразной информации и встречи с разными людьми. Он же может подвергнуть испытанию вашу либеральность. Существует великое множество самых разных мнений о приемлемом поведении. Большую часть времени в сети вы проводите, общаясь через ваш компьютер с другим компьютером где-то еще. Как вы себя при этом ведете, другим людям неизвестно. Вы можете, вообще говоря, вытворять весьма нехорошие вещи, все это остается на вашей, и только вашей, совести.

Конечно, кое-кто все же может узнать о ваших мерзких проделках, например, оператор сетевого узла, а может и нет. А те, кто все же узнает, станут ли они заботиться о вашем поведении или оставят все это на вашей же совести? Ведь у такого поведения есть и другая сторона - страдающая, которой как-нибудь можете оказаться и вы сами. Например, многие пользователи считают, что вполне честно заполучить к себе все файлы, до которых они смогут добраться. Даже до частной переписки, не говоря уже о ваших передовых разработках. Об этом вам всегда следует помнить, когда вы включаете свою машину в сеть. Если вы не намереваетесь допускать всех и вся к своим файлам, вам не следует их помещать туда, куда из сети можно попасть. Лучше бы, конечно, было, если бы все люди были достаточно воспитаны и сознательны, но так как коммунизма нам, видно, не видать, то и защита своей личной информации пока еще звучит актуально.

Активные пользователи Интернета находят его чрезвычайно полезным как для работы, так и для игры, и поскольку доступ в Сеть обычно не требует личных расходов (по крайней мере, больших), они рассматривают эти ценности как нечто, что должно защищать. Угрозы Всемирной паутины исходят из двух общих источников:

- Ø ненамеренное чрезмерное использование;
- Ø политическое давление.

Намеренное чрезмерное использование подразумевает полное осознание последствий такого использования и оценку его целесообразности. Это дело ума, чести и совести человека, решившегося на это. Если он может всем доказать, что такое использование необходимо и оправдано высшими соображениями о благе человечества, что ж, значит надо.

Сети создавались с определенной целью, коммерческое присоединение производится компаниями в определенных целях. Никто не может преследовать лицо, которое использует эти соединения не по назначению, но желание делать это отбивается по-другому и другими (не владельцами). Если вы используете компьютер работодателя в небольших личных целях, на

это, скорее всего, посмотрят сквозь пальцы. Небольшие затраты коммерческого сетевого времени не по предназначению тоже, наверное, будут молчаливо позволены. И только, если некто делает нечто совершенно непотребное, например, общегосударственный день сетевой игры или т.п., что начинает создавать проблемы многим или большие кому-либо, следует принимать меры по пресечению такого разбазаривания общественного достояния или частной собственности. Частную собственность разбазаривать Вам никто из капиталистов не даст.

Несообразное использование происходит не только в виде игр или других развлечений. Оно может происходить и в форме неразумного использования доступных ресурсов. Сеть совсем не призвана подменять неудовлетворительное пользовательское оборудование. Например, технически возможно использовать доступный удаленный диск в другом полушарии, оправдываясь тем, что ваш наниматель не хочет купить винчестер за \$333 для вашей рабочей станции, морально же это совершенно неприемлемо. Вполне возможно диск вам требуется для проведения ценного исследования, но плата за такое использование памяти через сеть совершенно непомерна. Сеть создана для облегчения доступа к уникальным ресурсам, а не для бесплатного доступа к рядовым.

Большинство продвинутых пользователей и владельцев сетей вполне нормальные люди, ничто человеческое им не чуждо. Они, как и другие, с удовольствием играют в игры, читают новости. В сети они работают регулярно. Если они видят, что работа сети начинает как-то портиться безо всяких видимых причин, они изучают это. Обычно это весьма квалифицированные специалисты и далеко не глупые люди: если они обнаруживают, что в данном районе сетевой поток вдруг возрос на порядки (в десять, сто, раз), они понимают, что это далеко неспроста, совсем не случайность. Им захочется узнать причину. Если вы и есть эта причина и ваше использование неприемлемо, скорее всего, вы получите вежливое послание по e-mail с просьбой прекратить такое поведение. Потом вы можете получить менее

вежливые послания, в конце концов, кто-нибудь свяжется с вашим местным «сетевиком». Вы можете кончить полной потерей доступа или же вашим работодателям или общине впредь придется больше платить за доступ, что их вряд ли осчастливит.

Саморегулирование важно также и из-за политики, окружающей сеть. Никто в здравом рассудке не станет полагать, что сеть живет вообще без проблем и злоупотреблений. Однако, если эти проблемы не разрешаются самим сетевым сообществом, а выносятся в печать и на обсуждение в высших кругах власти, всем только хуже.

Стоит еще перечислить некоторые сферы, считающиеся в сети «политически вредными» и которых, соответственно следует избегать:

- v чрезмерные игры;
- v чрезмерное несообразное использование;
- v злобное, надоедливое и иное антиобщественное поведение;
- v умышленное вредительство или вмешательство в дела других, в их оборудование и программное обеспечение;
- v обнародование файлов непристойного содержания.[3]

Очень трудно выбивать субсидии для сети в государственном бюджете, если как раз перед слушаниями о финансировании средства массовой информации рассказывают общественности о злоупотреблениях в сети. Например, Сеть клеймят рассадником секса, насилия и порнографии. И так как файлов такого содержания в Сети действительно наверное больше всех, то, с одной стороны, значит так и есть, а с другой работы над сетевой этикой ведутся уже все упорнее и глубже.

Учебно-методическое и информационное обеспечение

Основная литература:

- 1 Цветкова, М.С. Информатика и ИКТ : учеб. для студентов учреждений сред. проф. образования / М. С. Цветкова, Л. С. Великович. - 8-е изд., стер. - М. : Академия, 2016. - 332, [1] с., [4] л. ил.