

## ЗАНЯТИЯ 1-2. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ И ЕЕ МЕСТО В ПОДГОТОВКЕ СПЕЦИАЛИСТОВ.

*План*

1. *Позиционные и непозиционные системы счисления.*
2. *Перевод чисел в различные системы счисления.*

### ПОНЯТИЕ СИСТЕМЫ СЧИСЛЕНИЯ

Все фантастические возможности вычислительной техники (ВТ) реализуются путем создания разнообразных комбинаций сигналов высокого и низкого уровней, которые условились называть «единицами» и «нулями».

*Система счисления (СС) - это система записи чисел с помощью определенного набора цифр .СС называется позиционной, если одна и та же цифра имеет различное значение, которое определяется ее местом в числе. Десятичная СС является позиционной: 999. Римская СС является непозиционной. Значение цифры X в числе XXI остается неизменным при вариации ее положения в числе. Количество различных цифр, употребляемых в позиционной СС, называется *основанием СС*.*

**Развернутая форма числа** - это запись, которая представляют собой сумму произведений цифр числа на значение позиций.

Например:  $8527=8*10^3+5*10^2+2*10^1+7*10^0$

Развернутая форма записи чисел произвольной системы счисления имеет вид

Пусть

**q** - основание системы счисления,

**n** - число разрядов целой части числа,

**m** - число разрядов дробной части числа,

**a<sub>i</sub>** - цифра числа,

**A<sub>q</sub>** - само число.

Тогда развернутую форму для числа представленного в любой системе счисления можно записать в общем виде следующим образом:

$$A_q = a_{n-1} \cdot q^{n-1} + a_{n-2} \cdot q^{n-2} + \dots + a_0 \cdot q^0 + a_{-1} \cdot q^{-1} + a_{-2} \cdot q^{-2} + \dots + a_{-m} \cdot q^{-m}$$

или

$$A_q = \sum_{i=-m}^{n-1} a_i \cdot q^i$$

где  $q^i$  - называется весом цифры числа

Например: 1) 327.46  $n=3$ ,  $m=2$ ,  $q=10$

$$\begin{aligned} X &= \sum_{i=-2}^{2} a_i q^i = a_2 \cdot 10^2 + a_1 \cdot 10^1 + a_0 \cdot 10^0 + a_{-1} \cdot 10^{-1} + a_{-2} \cdot 10^{-2} = \\ &= 3 \cdot 10^2 + 2 \cdot 10^1 + 7 \cdot 10^0 + 4 \cdot 10^{-1} + 6 \cdot 10^{-2} \end{aligned}$$

2) Возьмем число в десятичной системе счисления, например 247,32, и представим его в следующем виде:

$$247,32 = 2 \cdot 10^2 + 4 \cdot 10^1 + 7 \cdot 10^0 + 3 \cdot 10^{-1} + 2 \cdot 10^{-2}$$

Если основание используемой СС больше десяти, то для цифр вводят условное обозначение со скобкой сверху или буквенное обозначение.

Например: если  $10=A$ , а  $11=B$ , то число  $7A.5B_{12}$  можно расписать так:

$$7A.5B_{12} = B \cdot 12^{-2} + 5 \cdot 12^{-1} + A \cdot 12^0 + 7 \cdot 12^1.$$

В шестнадцатеричной СС основа - это цифры 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 с соответствующими обозначениями 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. Примеры чисел: 17D.ECH, F12AH.

*Двоичная СС* - это система, в которой для записи чисел используются две цифры 0 и 1. Основанием двоичной системы счисления является число 2.

*Двоичный код* числа - запись этого числа в двоичной системе счисления.

Например,

$$0=0_2$$

$$1=1_2$$

$$2=10_2$$

$$3=11_2 \dots$$

$$7=111_2$$

$$120=1111000_2.$$

В ВТ применяют позиционные СС с недесятичным основанием: двоичную, восьмеричную, шестнадцатеричную. Для обозначения используемой СС число снабжают верхним или нижним индексом, в котором записывают основание СС. Другой способ – использование латинских букв после записи числа:

**D** – десятичная СС

**B** – двоичная СС

**O** – восьмеричная СС

**H** – 16-ричная СС.

Несмотря на то, что 10-тичная СС имеет широкое распространение, цифровая ВТ строится на двоичных элементах, т.к. реализовать элементы с 10 четко различимыми состояниями сложно. Историческое развитие ВТ сложилось таким образом, что ЭВМ строятся на базе двоичных цифровых устройств: триггеров, регистров, счетчиков, логических элементов и т.д.

16-ричная и 8-ричная СС используются при составлении программ на языке машинных кодов для более короткой и удобной записи двоичных кодов – команд, данных, адресов и операндов.

Задача перевода из одной СС в другую часто встречается при программировании, особенно, на языке Ассемблера. Например, при определении адреса ячейки памяти. Отдельные стандартные процедуры

языков программирования Паскаль, Бейсик, Си, HTML требуют задания параметров в 16-ричной СС. Для непосредственного редактирования данных, записанных на жесткий диск, также необходимо умение работать с 16-ричными числами. Отыскать неисправность в ЭВМ невозможно без представлений о двоичной СС.

В таблице приведены некоторые числа, представленные в различных СС.

<b>Двоичные числа</b>	<b>Восьмеричные числа</b>	<b>Десятичные числа</b>	<b>Шестнадцатеричные числа</b>
0	0	0	0
1	1	1	1
10	2	2	2
11	3	3	3
100	4	4	4
101	5	5	5
110	6	6	6
111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F

## ПЕРЕВОД ЧИСЕЛ ИЗ ПРОИЗВОЛЬНОЙ СС В ДЕСЯТИЧНУЮ И ОБРАТНО.

*Перевод чисел из произвольной системы в десятичную.* Для перевода числа из любой позиционной СС в десятичную необходимо использовать развернутую форму числа, заменяя, если это необходимо, буквенные обозначения соответствующими цифрами. Например:

$$1101_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 13_{10}$$

$$17D.EC_H = 12 \cdot 16^{-2} + 14 \cdot 16^{-1} + 13 \cdot 16^0 + 7 \cdot 16^1 + 1 \cdot 16^2 = 381.921875$$

*Перевод чисел из десятичной СС в заданную.*

1) Для преобразования целых чисел десятичной системы счисления в число любой системы счисления последовательно выполняют деление нацело на основание СС, пока не получат нуль. Числа, которые возникают как остаток от деления на основание СС, представляют собой последовательную запись разрядов числа в выбранной СС от младшего разряда к старшему. Поэтому для записи самого числа остатки от деления записывают в обратном порядке.

Например:

$$\begin{array}{r} 475 \overline{) 2} \\ 1 \phantom{00} \phantom{00} \phantom{00} \phantom{00} \phantom{00} \phantom{00} \phantom{00} \phantom{00} \\ \underline{1} \phantom{00} \phantom{00} \phantom{00} \phantom{00} \phantom{00} \phantom{00} \phantom{00} \\ 237 \phantom{00} \phantom{00} \phantom{00} \phantom{00} \phantom{00} \phantom{00} \phantom{00} \\ 1 \phantom{00} \phantom{00} \phantom{00} \phantom{00} \phantom{00} \phantom{00} \phantom{00} \\ \underline{1} \phantom{00} \phantom{00} \phantom{00} \phantom{00} \phantom{00} \phantom{00} \\ 118 \phantom{00} \phantom{00} \phantom{00} \phantom{00} \phantom{00} \phantom{00} \\ 0 \phantom{00} \phantom{00} \phantom{00} \phantom{00} \phantom{00} \phantom{00} \\ 59 \phantom{00} \phantom{00} \phantom{00} \phantom{00} \phantom{00} \\ 1 \phantom{00} \phantom{00} \phantom{00} \phantom{00} \phantom{00} \\ \underline{1} \phantom{00} \phantom{00} \phantom{00} \phantom{00} \\ 29 \phantom{00} \phantom{00} \phantom{00} \phantom{00} \\ 1 \phantom{00} \phantom{00} \phantom{00} \phantom{00} \\ \underline{1} \phantom{00} \phantom{00} \phantom{00} \\ 14 \phantom{00} \phantom{00} \phantom{00} \\ 0 \phantom{00} \phantom{00} \phantom{00} \\ 7 \phantom{00} \phantom{00} \\ 1 \phantom{00} \phantom{00} \\ \underline{1} \phantom{00} \\ 3 \phantom{00} \\ 1 \phantom{00} \\ \underline{1} \\ 1 \phantom{00} \\ \underline{1} \\ 0 \end{array}$$

Читая остатки от деления снизу вверх, получим 111011011.

Проверка:

$$1 \cdot 2^8 + 1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 1 + 2 + 8 + 16 + 64 + 128 + 256 = 475_{10}.$$

2) Для преобразования десятичных дробей десятичной СС в число любой СС последовательно выполняют умножение на основание системы счисления, пока дробная часть произведения не станет равной нулю. Полученные целые части являются разрядами числа в новой системе, и их необходимо представлять цифрами этой новой системы счисления. Целые части в дальнейшем отбрасываются.

Например: перевести число  $0.375_{10}$  в двоичную СС.

$$\begin{array}{r}
 0.375 \rightarrow 0 \\
 \underline{\phantom{0.}2} \\
 0.750 \rightarrow 0 \\
 \underline{\phantom{0.}2} \\
 1.50 \rightarrow 1 \\
 \phantom{1.}0.50 \\
 \underline{\phantom{1.}2} \\
 1.00 \rightarrow 1
 \end{array}$$

Полученный результат -  $0.011_2$ .

Необходимо отметить, что не каждое число может быть точно выражено в новой системе счисления, поэтому иногда вычисляют только требуемое количество разрядов дробной части, округляя последний разряд.

## **ПЕРЕВОД МЕЖДУ ОСНОВАНИЯМИ, СОСТАВЛЯЮЩИМИ СТЕПЕНЬ 2.**

Для того, чтобы из *восьмеричной* системы счисления перевести число в *двоичный* код, необходимо каждую цифру этого числа представить триадой двоичных символов. Лишние нули в старших разрядах отбрасываются.

Например:

$$1234.777_8 = 001\ 010\ 011\ 100.111\ 111\ 111_2 = 1\ 010\ 011\ 100.111\ 111\ 111_2$$

$$1234567_8 = 001\ 010\ 011\ 100\ 101\ 110\ 111_2 = 1\ 010\ 011\ 100\ 101\ 110\ 111_2$$

Обратный перевод: каждая триада двоичных цифр заменяется восьмеричной цифрой, при этом, если необходимо, число выравнивается путем дописывания нулей перед целой частью или после дробной.

Например:

$$1100111_2 = 001\ 100\ 111_2 = 147_8$$

$$11.1001_2 = 011.100\ 100_2 = 3.44_8$$

$$110.0111_2 = 110.011\ 100_2 = 6.34_8$$

При переводах между двоичной и шестнадцатеричной СС используются четверки цифр. При необходимости выравнивание выполняется до длины двоичного числа, кратной четырем.

Например:

$$1234.AB77_{16} = 0001\ 0010\ 0011\ 0100.1010\ 1011\ 0111\ 0111_2 = 1\ 0010\ 0011\ 0100.1010\ 1011\ 0111\ 0111_2$$

$$CE4567_{16} = 1100\ 1110\ 0100\ 0101\ 0110\ 0111_2$$

$$0.1234AA_{16} = 0.0001\ 0010\ 0011\ 0100\ 1010\ 1010_2$$

$$1100111_2 = 0110\ 0111_2 = 67_{16}$$

$$11.1001_2 = 0011.1001_2 = 3.9_{16}$$

$$110.0111001_2 = 0110.0111\ 0010_2 = 65.72_{16}$$

При переходе из восьмеричного счисления в шестнадцатеричное счисление и обратно используется вспомогательный двоичный код числа.

Например:

$$1234567_8 = 001\ 010\ 011\ 100\ 101\ 110\ 111_2 = 0101\ 0011\ 1001\ 0111\ 0111_2 = 53977_{16}$$

$$0.12034_8 = 0.001\ 010\ 000\ 011\ 100_2 = 0.0010\ 1000\ 0011\ 1000_2 = 0.2838_{16}$$

$$120.34_8 = 001\ 010\ 000.\ 011\ 100_2 = 0101\ 0000.0111\ 0000_2 = 50.7_{16}$$

$$1234.AB77_{16} = 0001\ 0010\ 0011\ 0100.1010\ 1011\ 0111\ 0111_2 = \\ = 001\ 001\ 000\ 110\ 100.101\ 010\ 110\ 111\ 011\ 100_2 = 11064.526734_8$$

$$CE4567_{16} = 1100\ 1110\ 0100\ 0101\ 0110\ 0111_2 = 110\ 011\ 100\ 100\ 010\ 101\ 100\ 111_2 = 63442547_8$$

$0.1234AA_{16} = 0.0001\ 0010\ 0011\ 0100\ 1010\ 1010_2 = 0.000\ 100\ 100\ 011\ 010\ 010\ 101\ 010_2 = 0.04432252_8$

ЛИТЕРАТУРА: [\[4\]](#), [\[6\]](#), [\[10\]](#), [\[15\]](#), [\[27\]](#)

### ЗАНЯТИЯ 3-4. ИНФОРМАЦИЯ ЕЕ ВИДЫ И СВОЙСТВА. ПРЕДСТАВЛЕНИЕ И КОДИРОВАНИЕ ИНФОРМАЦИИ В КОМПЬЮТЕРАХ

*План.*

1. *Информация и ее виды*
2. *Измерение количества информации*
3. *Свойства информации*

#### **Информация, ее виды.**

Термин "**информация**" происходит от латинского слова "*informatio*", что означает *сведения, разъяснения, изложение*. Несмотря на широкое распространение этого термина, понятие информации является одним из самых дискуссионных в науке. В настоящее время наука пытается найти общие свойства и закономерности, присущие многогранному понятию *информация*, но пока это понятие во многом остается интуитивным и получает различные смысловые наполнения в различных отраслях человеческой деятельности.

*В обиходе* информацией называют любые данные или сведения, которые кого-либо интересуют. Например, сообщение о каких-либо событиях, о чьей-либо деятельности и т.п.

Клод Шеннон, американский учёный, заложивший основы **теории информации** — науки, изучающей процессы, связанные с передачей, приёмом,

*преобразованием и хранением информации, рассматривает информацию как снятую неопределенность наших знаний о чем-то.*

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

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

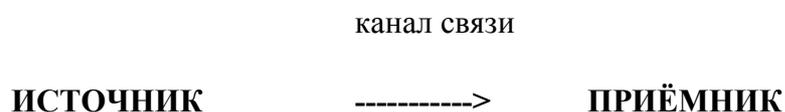
Различают две формы представления информации - *непрерывную (аналоговую) и прерывистую (цифровую, дискретную)*. **Непрерывная форма** характеризует процесс, который не имеет перерывов и может изменяться в любой момент времени и теоретически на любую величину (например, речь человека). **Цифровой сигнал** может изменяться лишь в определенные моменты времени и принимать лишь заранее обусловленные значения. Для преобразования аналогового сигнала в цифровой сигнал требуется провести дискретизацию во времени и квантование по уровню.

**Дискретизация** - это замена непрерывного сигнала последовательностью отдельных во времени отсчетов этого сигнала.

Для преобразования аналогового сигнала в цифровой используется специальный конвертор, называемый **аналого-цифровой преобразователь (АЦП)**. Сигнал на выходе АЦП представляет собой последовательность двоичных чисел, которая может быть записана на лазерный диск или обработана компьютером. Обратная конверсия осуществляется с помощью **цифро-аналогового преобразователя (ЦАП)**. Качество аналого-цифрового преобразования характеризуется разрешением. **Разрешение** - это

количество уровней квантования, используемых для замены непрерывного сигнала цифровым. Еще один показатель качества такого преобразования - **частота дискретизации** - количество преобразований, производимое в секунду. Этот показатель измеряют килогерцами.

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



Пример:

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

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

#### **4. Измерение количества информации**

Какое количество информации содержится, к примеру, в тексте романа "Война и мир", во фресках Рафаэля или в генетическом коде человека? Ответа на эти вопросы наука не даёт и, по всей вероятности, даст не скоро.

*В определенных, весьма широких условиях можно пренебречь качественными особенностями информации, выразить её количество числом, а также сравнить количество информации, содержащейся в различных группах данных.*

В настоящее время получили распространение подходы к определению понятия "количество информации", основанные на том, что информацию, содержащуюся в сообщении, можно нестрого трактовать в смысле её новизны или, иначе, уменьшения неопределённости наших знаний об объекте.

## **5. Подходы к определению количества информации. Формулы Хартли и Шеннона.**

Американский инженер Ральф Хартли в 1928 г. *процесс получения информации рассматривал как выбор одного сообщения из конечного наперёд заданного множества из  $N$  равновероятных сообщений, а количество информации  $I$ , содержащееся в выбранном сообщении, определял как двоичный логарифм  $N$ .*

$$\text{Формула Хартли: } I = \ln N$$

Допустим, нужно угадать одно число из набора чисел от единицы до ста. По формуле Хартли можно вычислить, какое количество информации для этого требуется:  $I = \ln 100 > 6,644$ . Таким образом, сообщение о верно угаданном числе содержит количество информации, приблизительно равное 6,644 единицы информации.

## **6. Свойства информации:**

**Достоверность.** Информация достоверна, если она отражает истинное положение дел. Недостоверная информация может привести к неправильному пониманию или принятию неправильных решений. Достоверная информация со временем может стать недостоверной, так как она обладает свойством устаревать, то есть перестаёт отражать истинное положение дел.

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

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

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

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

**Понятность.** Если ценная и своевременная информация выражена непонятным образом, она может стать бесполезной. Информация становится понятной, если она выражена языком, на котором говорят те, кому предназначена эта информация.

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

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

**ЛИТЕРАТУРА:** [\[4\]](#), [\[6\]](#), [\[10\]](#), [\[15\]](#), [\[27\]](#)

## **ЗАНЯТИЯ 5-6. ПРЕДСТАВЛЕНИЕ И КОДИРОВАНИЕ ИНФОРМАЦИИ В КОМПЬЮТЕРЕ.**

*План*

1. *Кодирование числовой информации*

2. *Кодирование текстовой информации*
3. *Представление графической информации*
4. *Кодирование звука*

Современный компьютер может обрабатывать числовую, текстовую, графическую, звуковую и видео информацию. Все эти виды информации в компьютере представлены в двоичном коде, т. е. используется алфавит мощностью два (всего два символа 0 и 1). Связано это с тем, что удобно представлять информацию в виде последовательности электрических импульсов: импульс отсутствует (0), импульс есть (1). Такое кодирование принято называть двоичным, а сами логические последовательности нулей и единиц - машинным языком.

**Кодирование** – это преобразование данных одного типа через данные другого типа. В ЭВМ применяется система двоичного кодирования, основанная на представлении данных последовательностью двух знаков: 1 и 0, которые называются двоичными цифрами (binary digit – сокращенно bit).

Каждая цифра машинного двоичного кода несет количество информации равное одному биту. Следовательно, две цифры несут информацию 2 бита, четыре разряда --4 бита и т. д. Чтобы определить количество информации в битах, достаточно определить количество цифр в двоичном машинном коде.

**Представление целого положительного числа в компьютере:** Число переводится в двоичную систему, результат дополняется нулями слева в пределах выбранного формата, последний разряд слева является знаковым, в положительном числе он равен 0.

**Для представления целого отрицательного числа нужно:** Абсолютную величину числа перевести в двоичную систему, Результат дополнить нулями слева в пределах выбранного формата,

полученное число перевести в обратный код (0 заменяются 1, а 1 заменяются 0), к полученному коду добавить 1

**Дробные числа представляются в памяти компьютера в форме "с плавающей точкой".** Название происходит от возможности менять положение точки (запятой) в записи числа, представленного в виде:

$\pm m \cdot 10^{\pm p}$ , где  $m$  - мантисса числа,  $p$  - его порядок. Такая форма представления чисел еще называется экспоненциальной.

### **Представление нечисловой информации в памяти компьютера.**

#### **Кодирование текстовой информации.**

В настоящее время большая часть пользователей при помощи компьютера обрабатывает текстовую информацию, которая состоит из символов: букв, цифр, знаков препинания и др.

Традиционно для того чтобы закодировать один символ используют количество информации равное 1 байту, т. е.  $I = 1 \text{ байт} = 8 \text{ бит}$ . При помощи формулы, которая связывает между собой количество возможных событий  $K$  и количество информации  $I$ , можно вычислить сколько различных символов можно закодировать (считая, что символы - это возможные события):

$K = 2^I = 2^8 = 256$ , т. е. для представления текстовой информации можно использовать алфавит мощностью **256** символов.

Суть кодирования заключается в том, что каждому символу ставят в соответствие двоичный код от 00000000 до 11111111 или соответствующий ему десятичный код от 0 до 255.

**Кодовая таблица** – это внутреннее представление символов в компьютере. В качестве стандарта долгое время использовалась таблица ASCII (American Standard Code for Informational Interchange)-Американский стандартный код для обмена информацией. Для хранения двоичного кода одного символа выделен 1 байт=8 бит.

В настоящее время для кодировки русских букв используют пять различных кодовых таблиц (КОИ - 8, CP1251, CP866, Mac, ISO), причем тексты, закодированные при помощи одной таблицы не будут правильно отображаться в другой кодировке. Наглядно это можно представить в виде фрагмента объединенной таблицы кодировки символов.

Одному и тому же двоичному коду ставится в соответствие различные СИМВОЛЫ

Двоичный код	Десятичный код	КОИ8	CP1251	CP866	Mac	ISO
11000010	194	б	В	-	-	Г

В большинстве случаев о перекодировке текстовых документов заботится не пользователь, а специальные программы - *конверторы*, которые встроены в приложения.

*Например*, кодировка символов русского языка Windows – 1251 используется для компьютеров, которые работают под ОС Windows. Другая кодировка для русского языка – это КОИ – 8, которая также широко используется в компьютерных сетях и российском секторе Интернет.

Начиная с 1997 г. последние версии Microsoft Windows&Office поддерживают новую кодировку **Unicode**, которая на каждый символ отводит по 2 байта, а, поэтому, можно закодировать не 256 символов, а 65536 различных символов.

*Задача.*

Два текста содержат одинаковое количество символов. Первый текст записан на русском языке, а второй на языке племени нагури, алфавит которого состоит из 16 символов. Чей текст несет большее количество информации?

Решение.

$I = K * a$  (информационный объем текста равен произведению числа символов на информационный вес одного символа).

Т.к. оба текста имеют одинаковое число символов ( $K$ ), то разница зависит от информативности одного символа алфавита ( $a$ ).

$$2^{a_1} = 32, \text{ т.е. } a_1 = 5 \text{ бит,}$$

$$2^{a_2} = 16, \text{ т.е. } a_2 = 4 \text{ бит.}$$

$$I_1 = K * 5 \text{ бит, } I_2 = K * 4 \text{ бит.}$$

Значит, текст, записанный на русском языке в  $5/4$  раза несет больше информации.

### **Представление графической информации в компьютере**

В середине 50-х годов для больших ЭВМ, которые применялись в научных и военных исследованиях, впервые в графическом виде было реализовано представление данных. В настоящее время широко используются технологии обработки графической информации с помощью ПК. Графический интерфейс пользователя стал стандартом "де-факто" для ПО разных классов, начиная с операционных систем. Вероятно, это связано со свойством человеческой психики: наглядность способствует более быстрому пониманию. Широкое применение получила специальная область информатики, которая изучает методы и средства создания и обработки изображений с помощью программно-аппаратных вычислительных комплексов, - **компьютерная графика**. Без нее трудно представить уже не только компьютерный, но и вполне материальный мир, так как визуализация данных применяется во многих сферах человеческой деятельности. В качестве примера можно привести опытно-конструкторские разработки, медицину (компьютерная томография), научные исследования и др.

Особенно интенсивно технология обработки графической информации с помощью компьютера стала развиваться в 80-х годах. Графическую

информацию можно представлять в двух формах: *аналоговой или дискретной*. Живописное полотно, цвет которого изменяется непрерывно - это пример аналогового представления, а изображение, напечатанное при помощи струйного принтера и состоящее из отдельных точек разного цвета - это дискретное представление.

Путем разбиения графического изображения (*дискретизации*) происходит преобразование графической информации из аналоговой формы в дискретную. При этом производится кодирование - присвоение каждому элементу конкретного значения в форме кода. При кодировании изображения происходит его пространственная дискретизация (разделение изображения на участки, в пределах которых значения тона либо усредняются, либо представлены одним из значений участка). Ее можно сравнить с построением изображения из большого количества маленьких цветных фрагментов (метод мозаики). Все изображение разбивается на отдельные точки, каждому элементу ставится в соответствие код его цвета. При этом качество кодирования будет зависеть от следующих параметров: размера точки и количества используемых цветов. Чем меньше размер точки, а, значит, изображение составляется из большего количества точек, тем выше качество кодирования. Чем большее количество цветов используется (т. е. точка изображения может принимать больше возможных состояний), тем больше информации несет каждая точка, а, значит, увеличивается качество кодирования. Создание и хранение графических объектов возможно в нескольких видах - в виде *векторного, фрактального или растрового изображения*. Отдельным предметом считается *3D (трехмерная) графика*, в которой сочетаются векторный и растровый способы формирования изображений. Она изучает методы и приемы построения объемных моделей объектов в виртуальном пространстве. Для каждого вида используется свой способ кодирования графической информации.

*Растровое изображение.*

При помощи увеличительного стекла можно увидеть, что черно-белое графическое изображение, например из газеты, состоит из мельчайших точек, составляющих определенный узор - *растр*. Во Франции в 19 веке возникло новое направление в живописи - *пуантилизм*. Его техника заключалась в том, что на холст рисунок наносился кистью в виде разноцветных точек. Также этот метод издавна применяется в полиграфии для кодирования графической информации. Точность передачи рисунка зависит от количества точек и их размера. После разбиения рисунка на точки, начиная с левого угла, двигаясь по строкам слева направо, можно кодировать цвет каждой точки. Далее одну такую точку будем называть *пикселем* (происхождение этого слова связано с английской аббревиатурой "picture element" - элемент рисунка). **Объем растрового изображения** *определяется умножением количества пикселей (на информационный объем одной точки, который зависит от количества возможных цветов. Качество изображения определяется разрешающей способностью монитора. Чем она выше, то есть больше количество строк растра и точек в строке, тем выше качество изображения. В современных ПК в основном используют следующие разрешающие способности экрана: 1024 на 768 и 1280 на 1024, 1400x1500 точек и т.д.. Так как яркость каждой точки и ее линейные координаты можно выразить с помощью целых чисел, то можно сказать, что этот метод кодирования позволяет использовать двоичный код для того чтобы обрабатывать графические данные.*

Если говорить о черно-белых иллюстрациях, то, если не использовать полутона, то пиксель будет принимать одно из двух состояний: светится (белый) и не светится (черный). А так как информация о цвете пикселя называется кодом пикселя, то для его кодирования достаточно одного бита памяти: 0 - черный, 1 - белый. Если же рассматриваются иллюстрации в виде комбинации точек с 256 градациями серого цвета (а именно такие в настоящее время общеприняты), то достаточно восьмиразрядного двоичного числа для того чтобы закодировать яркость любой точки. В компьютерной графике чрезвычайно важен цвет. Он выступает как средство усиления зрительного

впечатления и повышения информационной насыщенности изображения. Как формируется ощущение цвета человеческим мозгом? Это происходит в результате анализа светового потока, попадающего на сетчатку глаза от отражающих или излучающих объектов. Принято считать, что цветовые рецепторы человека, которые еще называют колбочками, подразделяются на три группы, причем каждая может воспринимать всего один цвет - *красный, или зеленый, или синий*.

### *Цветовые модели.*

Если говорить о кодировании цветных графических изображений, то нужно рассмотреть принцип декомпозиции произвольного цвета на основные составляющие. Применяют несколько систем кодирования: HSB (англ. *Hue, Saturation, Brightness* — *тон, насыщенность, яркость*), RGB и CMYK (англ. *Cyan, Magenta, Yellow, Key color*). Первая цветовая модель проста и интуитивно понятна, т. е. удобна для человека, вторая наиболее удобна для компьютера, а последняя модель CMYK-для типографий. Использование этих цветовых моделей связано с тем, что световой поток может формироваться излучениями, представляющими собой комбинацию "чистых" спектральных цветов: красного, зеленого, синего или их производных. Различают *аддитивное цветовоспроизведение* (характерно для излучающих объектов) и *субтрактивное цветовоспроизведение* (характерно для отражающих объектов). В качестве примера объекта первого типа можно привести электронно-лучевую трубку монитора, второго типа - полиграфический отпечаток.

1) Модель HSB характеризуется тремя компонентами: оттенок цвета (Hue), насыщенность цвета (Saturation) и яркость цвета (Brightness). Можно получить большое количество произвольных цветов, регулируя эти компоненты. Эту цветовую модель лучше применять в тех графических редакторах, в которых изображения создают сами, а не обрабатывают уже готовые. Затем созданное свое произведение можно преобразовать в цветовую

модель RGB, если ее планируется использовать в качестве экранной иллюстрации, или CMYK, если в качестве печатной. Значение цвета выбирается как вектор, выходящий из центра окружности. Направление вектора задается в угловых градусах и определяет цветовой оттенок. Насыщенность цвета определяется длиной вектора, а яркость цвета задается на отдельной оси, нулевая точка которой имеет черный цвет. Точка в центре соответствует белому (нейтральному) цвету, а точки по периметру - чистым цветам.

2) Принцип метода RGB заключается в следующем: известно, что любой цвет можно представить в виде комбинации трех цветов: красного (Red, R), зеленого (Green, G), синего (Blue, B). Другие цвета и их оттенки получаются за счет наличия или отсутствия этих составляющих. По первым буквам основных цветов система и получила свое название - RGB. Данная цветовая модель является *аддитивной*, то есть любой цвет можно получить сочетанием основных цветов в различных пропорциях. При наложении одного компонента основного цвета на другой яркость суммарного излучения увеличивается. Если совместить все три компонента, то получим ахроматический серый цвет, при увеличении яркости которого происходит приближение к белому цвету.

При 256 градациях тона (каждая точка кодируется 3 байтами) минимальные значения RGB (0,0,0) соответствуют черному цвету, а белому - максимальные с координатами (255, 255, 255). Чем больше значение байта цветовой составляющей, тем этот цвет ярче. Например, темно-синий кодируется тремя байтами ( 0, 0, 128), а ярко-синий (0, 0, 255).

3) Принцип метода CMYK. Эта цветовая модель используется при подготовке публикаций к печати. Каждому из основных цветов ставится в соответствие дополнительный цвет (дополняющий основной до белого). Получают дополнительный цвет за счет суммирования пары остальных основных цветов. Значит, дополнительными цветами для красного является

голубой (Cyan, C) = зеленый + синий = белый - красный, для зеленого - пурпурный (Magenta, M) = красный + синий = белый - зеленый, для синего - желтый (Yellow, Y) = красный + зеленый = белый - синий. Причем принцип декомпозиции произвольного цвета на составляющие можно применять как для основных, так и для дополнительных, то есть любой цвет можно представить или в виде суммы красной, зеленой, синей составляющей или же в виде суммы голубой, пурпурной, желтой составляющей. В основном такой метод принят в полиграфии. Но там еще используют черный цвет (Black, так как буква B уже занята синим цветом, то обозначают буквой K). Это связано с тем, что наложение друг на друга дополнительных цветов не дает чистого черного цвета.

Различают несколько режимов представления цветной графики:

- a) *полноцветный (True Color);*
- б) *High Color;*
- в) *индексный.*

При **полноцветном режиме** для кодирования яркости каждой из составляющих используют по 256 значений (восемь двоичных разрядов), то есть на кодирование цвета одного пикселя (в системе RGB) надо затратить  $8*3=24$  разряда. Это позволяет однозначно определять 16,5 млн цветов. Это довольно близко к чувствительности человеческого глаза. При кодировании с помощью системы CMYK для представления цветной графики надо иметь  $8*4=32$  двоичных разряда.

**Режим High Color** - это кодирование при помощи 16-разрядных двоичных чисел, то есть уменьшается количество двоичных разрядов при кодировании каждой точки. Но при этом значительно уменьшается диапазон кодируемых цветов.

При **индексном кодировании** цвета можно передать всего лишь 256 цветовых оттенков. Каждый цвет кодируется при помощи восьми бит данных.

Но так как 256 значений не передают весь диапазон цветов, доступный человеческому глазу, то подразумевается, что к графическим данным прилагается палитра, без которой воспроизведение будет неадекватным: море может получиться красным, а листья - синими. Сам код точки раstra в данном случае означает не сам по себе цвет, а только его номер (индекс) в палитре. Отсюда и название режима - индексный.

*Соответствие между количеством отображаемых цветов (K) и количеством бит для их кодировки (a) находится по формуле:*

$$K = 2^a.$$

A	K	Достаточно для...
4	$2^4 = 16$	
8	$2^8 = 256$	Рисованных изображений типа тех, что видим в мультфильмах, но недостаточно для изображений живой природы
16 (High Color)	$2^{16} = 65536$	Изображений, которые на картинках в журналах и на фотографиях
24 (True Color)	$2^{24} = 16\,777\,216$	Обработки и передачи изображений, не уступающих по качеству наблюдаемым в живой природе

Двоичный код изображения, выводимого на экран, хранится в видеопамяти. **Видеопамять** - это электронное энергозависимое запоминающее устройство. Размер видеопамяти зависит от разрешающей способности дисплея и количества цветов. Но ее минимальный объем определяется так, чтобы поместился один кадр (одна страница) изображения, т.е. как результат произведения разрешающей способности на размер кода пикселя.

*Двоичный код восьмицветной палитры.*

Цвет	Составляющие		
	к	З	С

Красный	1	0	0
Зеленый	0	1	0
Синий	0	0	1
Голубой	0	1	1
Пурпурный	1	0	1
Желтый	1	1	0
Белый	1	1	1
Черный	0	0	0

### ***Векторное и фрактальное изображения.***

***Векторное изображение*** - это графический объект, состоящий из элементарных отрезков и дуг. Базовым элементом изображения является линия. Как и любой объект, она обладает свойствами: формой (прямая, кривая), толщиной, цветом, начертанием (пунктирная, сплошная). Замкнутые линии имеют свойство заполнения (или другими объектами, или выбранным цветом). Все прочие объекты векторной графики состояются из линий. Так как линия описывается математически как единый объект, то и объем данных для отображения объекта средствами векторной графики значительно меньше, чем в растровой графике. Информация о векторном изображении кодируется как обычная буквенно-цифровая и обрабатывается специальными программами.

К программным средствам создания и обработки векторной графики относятся следующие ГР: CorelDraw, Adobe Illustrator, а также векторизаторы (трассировщики) - специализированные пакеты преобразования растровых изображений в векторные.

***Фрактальная графика*** основывается на математических вычислениях, как и векторная. Но в отличие от векторной ее базовым элементом является сама математическая формула. Это приводит к тому, что в памяти компьютера

не хранятся никаких объектов и изображение строится только по уравнениям. При помощи этого способа можно строить простейшие регулярные структуры, а также сложные иллюстрации, которые имитируют ландшафты.

### **Кодирование звуковой информации**

Мир наполнен самыми разнообразными звуками: тиканье часов и гул моторов, завывание ветра и шелест листьев, пение птиц и голоса людей. О том, как рождаются звуки и что они собой представляют люди начали догадываться очень давно. Еще древнегреческий философ и ученый - энциклопедист Аристотель, исходя из наблюдений, объяснял природу звука, полагая, что звучащее тело создает попеременное сжатие и разрежение воздуха. Так, колеблющаяся струна то разряжает, то уплотняет воздух, а из-за упругости воздуха эти чередующиеся воздействия передаются дальше в пространство - от слоя к слою, возникают *упругие волны*. Достигая нашего уха, они воздействуют на барабанные перепонки и вызывают ощущение звука.

На слух человек воспринимает упругие волны, имеющие частоту где-то в пределах от 16 Гц до 20 кГц (1 Гц - 1 колебание в секунду). В соответствии с этим *упругие волны в любой среде, частоты которых лежат в указанных пределах, называют звуковыми волнами или просто звуком*. В учении о звуке важны такие понятия как *тон* и *тембр* звука. Всякий реальный звук, будь то игра музыкальных инструментов или голос человека, - это своеобразная смесь многих гармонических колебаний с определенным набором частот.

Колебание, которое имеет наиболее низкую частоту, называют *основным тоном*, другие - *обертонами*.

*Тембр* - разное количество обертонов, присущих тому или иному звуку, которое придает ему особую окраску. Отличие одного тембра от другого обусловлено не только числом, но и интенсивностью обертонов, сопровождающих звучание основного тона. Именно по тембру мы легко можем отличить звуки рояля и скрипки, гитары и флейты, узнать голос знакомого человека.

Музыкальный звук можно характеризовать тремя качествами: тембром, т. е. окраской звука, которая зависит от формы колебаний, высотой, определяющейся числом колебаний в секунду (частотой), и громкостью, зависящей от интенсивности колебаний.

Компьютер широко применяют в настоящее время в различных сферах. Не стала исключением и обработка звуковой информации, музыка. До 1983 года все записи музыки выходили на виниловых пластинках и компакт-кассетах. В настоящее время широкое распространение получили компакт-диски. Если имеется компьютер, на котором установлена студийная звуковая плата, с подключенными к ней MIDI-клавиатурой и микрофоном, то можно работать со специализированным музыкальным программным обеспечением.

Условно его можно разбить на несколько видов:

1) *всевозможные служебные программы и драйверы, предназначенные для работы с конкретными звуковыми платами и внешними устройствами, к ним относятся все служебные программы операционной системы;*

2) *аудиоредакторы, которые предназначены для работы со звуковыми файлами, позволяют производить с ними любые операции - от разбиения на части до обработки эффектами;*

3) *программные синтезаторы, которые появились сравнительно недавно и корректно работают только на мощных компьютерах. Они позволяют экспериментировать с созданием различных звуков; и другие.*

А как же происходит кодирование звука? С самого детства мы сталкиваемся с записями музыки на разных носителях: грампластинках, кассетах, компакт-дисках и т.д. В настоящее время существует два основных способа записи звука: *аналоговый и цифровой*. Но для того чтобы записать звук на какой-нибудь носитель его нужно преобразовать в электрический сигнал.

Это делается с помощью микрофона. Самые простые микрофоны имеют мембрану, которая колеблется под воздействием звуковых волн. К мембране присоединена катушка, перемещающаяся синхронно с мембраной в магнитном поле. В катушке возникает переменный электрический ток. Изменения напряжения тока точно отражают звуковые волны.

Переменный электрический ток, который появляется на выходе микрофона, называется *аналоговым* сигналом. Применительно к электрическому сигналу «аналоговый» обозначает, что этот сигнал непрерывен по времени и амплитуде. Он точно отражает форму звуковой волны, которая распространяется в воздухе.

Звуковую информацию можно представить в дискретной или аналоговой форме. Их отличие в том, что при дискретном представлении информации физическая величина изменяется скачкообразно («лесенкой»), принимая конечное множество значений. Если же информацию представить в аналоговой форме, то физическая величина может принимать бесконечное количество значений, непрерывно изменяющихся.

Виниловая пластинка является примером аналогового хранения звуковой информации, так как звуковая дорожка свою форму изменяет непрерывно. Но у аналоговых записей на магнитную ленту есть большой недостаток - старение носителя. За год фонограмма, которая имела нормальный уровень высоких частот, может их потерять. Виниловые пластинки при проигрывании их несколько раз теряют качество. Поэтому преимущество отдают цифровой записи.

В начале 80-х годов появились компакт-диски. Они являются примером дискретного хранения звуковой информации, так как звуковая дорожка компакт - диска содержит участки с различной отражающей способностью. Теоретически эти цифровые диски могут служить вечно, если их не царапать, т.е. их преимуществами являются долговечность и неподверженность

механическому старению. Другое преимущество заключается в том, что при цифровой перезаписи нет потери качества звука.

Во время аналого-цифрового преобразования никакого физического преобразования не происходит. С электрического сигнала как бы снимается отпечаток или образец, являющийся цифровой моделью колебаний напряжения в аудиотракте. Если это изобразить в виде схемы, то эта модель представлена в виде последовательности столбиков, каждый из которых соответствует определенному числовому значению. Цифровой сигнал по своей природе дискретен - то есть прерывист, поэтому цифровая модель не совсем точно соответствует форме аналогового сигнала.

*Семпл* - это промежуток времени между двумя измерениями амплитуды аналогового сигнала .

Дословно *Sample* переводится с английского как «образец». В мультимедийной и профессиональной звуковой терминологии это слово имеет несколько значений. Кроме промежутка времени семплом называют также любую последовательность цифровых данных, которые получили путем аналого-цифрового преобразования. Сам процесс преобразования называют *семплированием*. В русском техническом языке называют его *дискретизацией*.

**Звук** представляет собой звуковую волну с меняющейся амплитудой и частотой. Частота звуковой волны выражается числом колебаний в секунду и измеряется в герцах. *Количество бит, отводимое на один звуковой сигнал, называются глубиной кодирования звука*. Современные звуковые карты обеспечивают 16-, 32- или 64-битную глубину кодирования звука. При кодировании звуковой информации непрерывный сигнал заменяется дискретным, то есть превращается в последовательность электрических импульсов (двоичных нулей и единиц). Важной характеристикой при кодировании звука является частота дискретизации- количество измерений

уровней сигнала за 1 секунду: 1 измерение соответствует 1 Гц, 1000 измерений-1 кГц. Звуковые файлы имеют несколько форматов: MIDI, WAV, MP3. Когда говорят о **видеозаписи**, прежде всего имеют в виду движущееся изображение на экране телевизора или монитора. Процесс превращения непрерывного сигнала в набор кодовых слов называется аналого-цифровым преобразованием. Видеоинформация может храниться в файлах форматов AVI и MPEG.

**ЛИТЕРАТУРА:** [4], [6], [10], [15], [27]

## **ЗАНЯТИЕ 7-8. АРИФМЕТИЧЕСКИЕ ОСНОВЫ РАБОТЫ ЭВМ**

*План*

1. *Арифметическое действие сложения с двоичными числами*
2. *Арифметическое действие вычитание с двоичными числами*
3. *Прямой, обратный, дополнительный код.*

Правила выполнения арифметических действий над двоичными числами задаются таблицами сложения, вычитания и умножения.

<b>Сложение</b>	<b>Вычитание</b>	<b>Умножение</b>
$0+0=0$	$0-0=0$	$0 \times 0=0$
$0+1=1$	$1-0=1$	$0 \times 1=0$
$1+0=1$	$1-1=0$	$1 \times 0=0$
$1+1=10$	$10-1=1$	$1 \times 1=1$

Например:

$\begin{array}{r} 111 \\ + 101 \\ \hline 1100 \end{array}$	1) 1+1=10 2) 1+1=10 3) 1+1+1=11
$\begin{array}{r} 10001 \\ - 101 \\ \hline 1100 \end{array}$	1) 1-1=0 2) 0-0=0  3) 1-0=1   1   10   01 <hr style="width: 100%; border: 0.5px solid black;"/> <div style="display: flex; justify-content: center; gap: 20px;"> <span>  ·   1   01</span> </div> <div style="display: flex; justify-content: center; gap: 20px;"> <span>1 1</span> </div>

Правило выполнения операции сложения одинаково для всех систем счисления: если сумма складываемых цифр больше или равна основанию системы счисления, происходит перенос единицы в следующий слева разряд. При вычитании, если необходимо, делают заем. В ВТ с целью упрощения реализации арифметических операций применяют специальные коды: *прямой, обратный, дополнительный*. За счет этого облегчается определение знака результата операции, а операция вычитания чисел сводится к арифметическому сложению. В результате упрощаются устройства, выполняющие арифметические операции.

*Прямой код* складывается из знакового разряда (старшего) и собственно числа. Знаковый разряд имеет значение

0 – для положительных чисел;

1 – для отрицательных чисел.

Например: прямой код для чисел –4 и 5:

$$-4 \ 4_{10} = 100_2 \ 1\_100$$

$$5 \ 5_{10} = 101_2 \ 0\_101$$

*Обратный код* образуется из прямого кода заменой нулей - единицами, а единиц - нулями, кроме цифр знакового разряда. Для положительных чисел обратный код совпадает с прямым. Используется как промежуточное звено для получения дополнительного кода.

Например:

Прямой код 1\_100 1\_101

Обратный код 1\_011 1\_010

*Дополнительный код* образуется из обратного кода добавлением 1 к младшему разряду.

Например: найти дополнительный код  $-7_{10}$

$$-7_{10} = 111_2$$

Прямой код 1\_111

Обратный код 1\_000

Дополнительный код : 1\_001 ( $1_000 + 1$ )

*Правило сложения двоичных чисел:*

При алгебраическом сложении двоичных чисел с использованием дополнительного кода положительные слагаемые представляют в прямом коде, а отрицательные – в дополнительном коде. Затем производят суммирование этих кодов, включая знаковые разряды, которые при этом рассматриваются как старшие разряды. При возникновении переноса из знакового разряда единицу переноса отбрасывают. В результате получают алгебраическую сумму в прямом коде, если эта сумма положительная, и в дополнительном коде, если сумма отрицательная.

Например: 1) найти разность  $13_{10} - 12_{10}$

В двоичной системе

	$13_{10} = 1101_2$	и	$12_{10} = 1100_2$
Для	13		-12
Прямой код	0_1101		1_1100
Обратный код			1_0011
Дополнительный код			1_0100

Вычитание заменяем сложением:

$$\begin{array}{r} 0\_1101 \\ + 1\_0100 \\ \hline 10\_0001 \end{array}$$

, первую единицу отбрасываем и результат =  $0\_0001$ .

2) найти разность  $8_{10} - 13_{10}$

Для	8		-13
Прямой код	0_1000		1_1101
Обратный код			1_0010
Дополнительный код			1_0011

Вычитание заменяем сложением:

$$\begin{array}{r} 0\_1000 \\ + 1\_0011 \\ \hline 1\_1011 \end{array}$$

В знаковом разряде стоит единица и, значит, результат получен в дополнительном коде.

Перейдем от дополнительного кода к обратному:  $1\_1011 - 1 = 1\_1010$ .

Перейдем от обратного кода к прямому:  $1\_1010 \rightarrow 1\_0101 = -5_{10}$

**ЛИТЕРАТУРА:** [\[4\]](#), [\[6\]](#), [\[10\]](#), [\[15\]](#), [\[27\]](#)

## ЗАНЯТИЯ 9-10. ИСТОРИЯ РАЗВИТИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

*План*

1. Докомпьютерная эпоха

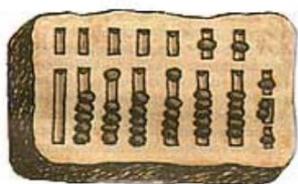
2. Поколения ЭВМ

Вычислительная техника является важнейшим компонентом *процесса вычислений и обработки данных*. Бесконечная вереница всевозможных счетных устройств и сами идеи механизации счета  $\square$  все это уходит в

невообразимо далекие времена.

Уже в палеолите можно найти следы фиксации счета в виде насечек на костяных и каменных изделиях. Для оперативного счета использовался простейший «прибор» - пальцы рук. По-видимому, первым искусственным прибором, облегчавшим счет, была бирка – деревянная палочка, на которой ножом наносились насечки различной формы. Первоначально бирки применялись только для записи (запоминания) чисел, на них велся счет дням, количеству голов скота, величине долга и т. д.

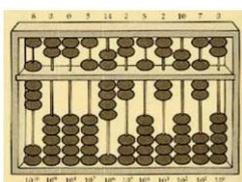
Поистине революционным событием в истории счета было появление приборов, объединяемых общим названием – **абак** (от греч. – доска). Абак мог иметь форму деревянной доски, глиняной плитки или просто очерченного кусочка земли. Важно, что на абаке отмечались места (колонки или строчки) для отдельных разрядов чисел. Камешек, косточка или другой предмет, помещённые в разных колонках, имели различное числовое значение. Вычисления на абаке сводилось к способу выкладывания камешков.



Древнегреческий абак



Римский абак

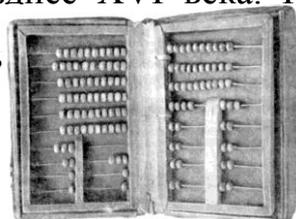


Суаньпань – китайский абак

Соробан - японский абак,  
происходит от китайского суаньпаня,  
который был завезен в Японию в XV-  
XVI веках

Одним из поздних вариантов абака являются обыкновенные конторские счеты.

В Россию абак проник не позднее XVI века. Русскими вариантами абака были «счет костями» и «дощань



Русский «дощаный счет»

Следующий толчок развитию счетного дела был дан шотландским математиком **Д. Непером**, придумавшим специальные «счетные палочки» (1617 г.). У Непера оказалось много последователей, которые совершенствовали его изобретение, создавая при этом немало остроумных и удобных для работы конструкций.



В конце 1620-х годов была изобретена **логарифмическая линейка**.

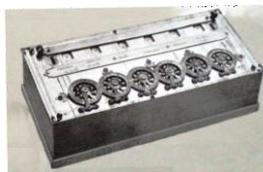
Современная логарифмическая линейка

Но пришло время, когда инструменты ручного счета перестали удовлетворять нуждам науки и инженерного дела, а новые практические потребности всегда рождают новые идеи и вызывают к жизни новые технические средства. Началась эра создания **механических счетных машин**.

Первая «считающая машина» описана профессором математики Тюбингенского университета **Уильямом Шиккардом (1623 г.)**, реализована

в единственном экземпляре и предназначалась для выполнения четырех арифметических операций над 6- значными числами.

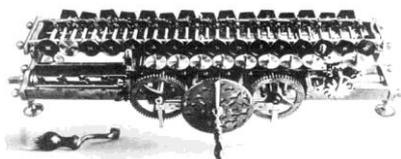
Машина Шиккарда состояла из трех независимых устройств: суммирующего, множительного и записи чисел. Сложение производилось последовательным вводом слагаемых посредством наборных дисков, а вычитание - последовательным вводом уменьшаемого и вычитаемого. Для выполнения операции умножения использовалась идея умножения решеткой. Третья часть машины использовалась для записи числа длиной не более 6 разрядов. Использованная принципиальная схема машины Шиккарда явилась классической - она (или ее модификации) использовалась в большинстве последующих механических счетных машин вплоть до замены механических деталей электромагнитными.



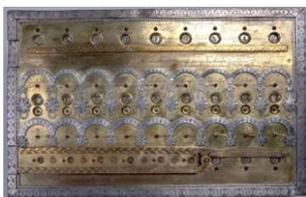
Первую действующую механическую вычислительную машину, которая была не только сконструирована, но и получила распространение, хотя и ограниченное, изобрел известный математик **Блез Паскаль (1642 г.)**. Его механический калькулятор стали называть «Паскалиной», или «Паскалевым колесом». Изобретенный Паскалем принцип связанных шестеренками колес явился основой, на которой строилось большинство механических вычислительных устройств на протяжении следующих трех столетий.

**Готфрид Лейбниц** изготовил механический калькулятор (1673 г.), в

который внес много своих новинок: он добавил в конструкцию подвижную каретку, сделал специальную рукоятку для вращения ступенчатого колеса. Основу машины Лейбница составляли *ступенчатые валики*. Они представляли собой цилиндрики с зубцами разной длины. Эта простая конструкция была великолепным техническим воплощением идеи шестерни с переменным числом зубцов. Именно такое числовое колесо обеспечивало выполнение операций умножения и деления.



У Лейбница были десятки последователей, которые построили множество работоспособных конструкций. Одна из них была создана **около 1770 года** в России механиком **Евной Якобсоном**.



Из более поздних конструкций стоит упомянуть арифмометр (точнее - суммирующую машину) крупного русского математика и механика **П. Л. Чебышева**. Он передал его в Парижский музей искусств и ремесел в **1878** году. Заложенные в арифмометр идеи не претендовали на оригинальность, за исключением одной, которая содержится в названии заметки П. Л. Чебышева «Счетная машина с **непрерывным движением**», опубликованной в 1882 году. В машине Паскаля механизм переноса в

следующий разряд срабатывал дискретно, скачком. У Чебышева этот процесс был постепенным, непрерывным, так как он впервые использовал планетарную передачу.



Одно из последних принципиальных изобретений в механической счетной технике было сделано жителем Петербурга **В. Однером**. Ему удалось сконструировать сравнительно простое колесо с переменным числом зубцов вместо ступенчатых валиков Лейбница. Построенный Однером в **1890** году арифмометр фактически ничем не отличается от современных подобных ему машин. Почти сразу Однер с компаньоном наладил выпуск своих арифмометров – по 500 штук в год. К 1914 году в одной только России насчитывалось более 22 тысяч арифмометров Однера. В первой четверти XX века эти арифмометры были единственными математическими машинами, широко применявшимися в различных областях деятельности человека.



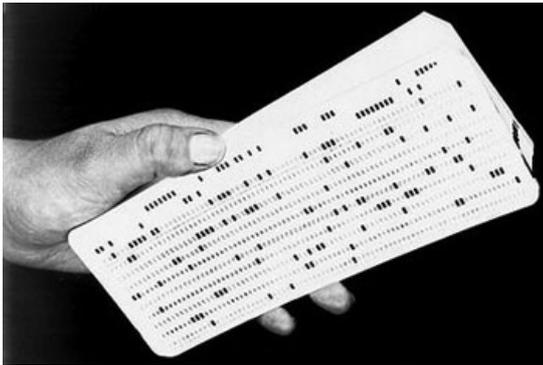
В Советском Союзе самым известным и распространённым калькулятором был механический арифмометр «Феликс», выпускавшийся с 1929 по 1978 год на заводах в Курске, Пензе и Москве.



На протяжении всего XVIII в. На французских фабриках по производству шелковых тканей велись эксперименты с различными механизмами, управлявшими станком при помощи перфорационной ленты, перфорационных карт или деревянных барабанов. Во всех трех системах нить поднималась и опускалась в соответствии с наличием или отсутствием отверстий – так создавался желаемый рисунок ткани. В **1804** году инженер Жозеф- Мари **Жаккар** построил полностью автоматизированный станок, способный воспроизводить сложнейшие узоры. Работа станка программировалась при помощи целой колоды перфокарт, каждая из которых управляла одним ходом челнока. Серия карт могла быть заменена, и смена узора не требовала изменений в механике станка. Это было важной вехой в истории программирования.

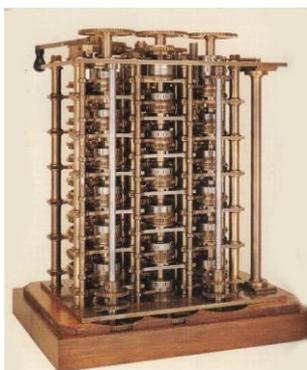


**Перфокарты** - носитель информации, предназначенный для использования в системах автоматической обработки данных. Сделанная из тонкого картона, перфокарта представляет информацию наличием или отсутствием отверстий в определённых позициях карты.



В 1822 г. английский математик **Чарльз Бэббидж** опубликовал научную статью с описанием машины, способной рассчитывать и печатать большие математические таблицы. В том же году он построил пробную модель своей **Разностной машины**, состоящую из шестеренок и валиков, вращаемых вручную при помощи специального рычага. Затем, заручившись поддержкой Королевского общества - самой престижной научной организации Великобритании, - он обратился к правительству с просьбой финансировать создание полномасштабной работающей машины. Эта машина, писал он президенту Королевского общества, возьмет на себя *«невыносимо утомительную работу»*, неизбежную при многократно повторяющихся математических расчетах, которые *«представляют собой самое низкое занятие, недостойное человеческого интеллекта»*. Королевское общество сочло его работу *«в высшей степени достойной общественной поддержки»*, и уже через год британское правительство представило Бэббиджу для реализации его проекта субсидию в 1500 фунтов стерлингов.

На протяжении следующего десятилетия Бэббидж без устали работал над своим изобретением. Первоначально он рассчитывал завершить ее за три года, но Разностная машина становилась все сложнее по мере того, как он ее модифицировал, совершенствовал и конструировал заново. При этом росли сомнения официальных лиц в целесообразности затрат и пользе самого проекта.

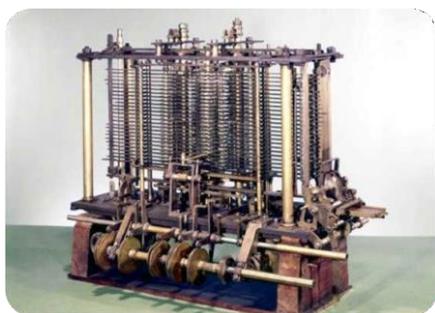


В 1833 г. Бэббидж был готов отказаться от своих планов, связанных с Разностной машиной. Однако, продолжая размышлять на ту же тему, он пришел к идее создания еще более мощной машины. В 1835 г. в письме президенту Брюссельской академии наук дал предварительное описание своего нового проекта. **Аналитическая машина** Бэббиджа в отличие от своей предшественницы должна была не просто решать математические задачи одного определенного типа, а выполнять разнообразные вычислительные операции в соответствии с инструкциями, задаваемыми оператором. По замыслу эта была «машина самого универсального характера» - в действительности не что иное, как первый универсальный программируемый компьютер.

Аналитическая машина должна была иметь такие компоненты, как «мельница» и «склад» (по современной терминологии - арифметическое устройство и память), состоящие из механических рычажков и шестеренок. Бэббидж рассчитывал работать с 50-разрядными числами, сто из которых можно было запоминать. Эти числа должны были храниться в памяти, пока до них не дойдет очередь в арифметическом устройстве. Результаты операции либо отправлялись в память, чтобы также ждать своей очереди, либо распечатывались. Инструкции, или команды, вводились в Аналитическую машину с помощью перфокарт.

*«Можно с полным основанием сказать, что Аналитическая машина точно так же плетет алгебраические узоры, как ткацкий станок Жаккара воспроизводит цветы и листья»,* - писала графиня **Лавлейс**, одна из

немногих, кто понимал, как работает машина и каковы потенциальные области ее применения. Графиню Лавлейс (урожденная Огаста Ада Байрон) иногда называют первым программистом; в ее честь назван язык программирования АДА.



Если Разностная машина имела сомнительные шансы на успех, то Аналитическая машина и вовсе выглядела нереалистичной. Ее просто невозможно было построить и запустить в работу. В своем окончательном виде машина должна была быть не меньше железнодорожного локомотива. Ее внутренняя конструкция представляла собой беспорядочное нагромождение стальных, медных и деревянных деталей, часовых механизмов, приводимых в действие паровым двигателем. Малейшая нестабильность какой-нибудь крошечной детали приводила бы к сто-кратно усиленным нарушениям в других частях, и тогда вся машина пришла бы в бешенство.

Аналитическая машина так и не была построена. Все, что дошло от нее до наших дней, - это ворох чертежей и рисунков, а также небольшая часть арифметического устройства и печатающее устройство, сконструированное сыном Бэббиджа. «Точно так же, как невозможно достигнуть Луны в деревянной ракете с двигателем внутреннего сгорания, так и невозможно было сделать аналитическую машину из механических элементов. Ничего же

другого техника XIX века предоставить не могла».

Лишь через 19 лет после смерти Бэббиджа один из принципов, лежащих в основе идеи Аналитической машины, - использование перфокарт – нашел воплощение в действующем устройстве. Это был статистический табулятор, построенный американцем **Германом Холлеритом** с целью ускорить обработку результатов переписи населения, которая проводилась в США в **1890** году.

«Этот аппарат, - восхищенно писал журнал *Electrical Engineer*, - работает так же безошибочно, как машины бессмертных богов, но намного превосходит их по быстродействию».

Холлерит называл себя «первым инженером-статистиком». Он организовал фирму по производству табуляционных машин (Tabulating Machine Company) и продавал их железнодорожным управлениям и правительственным учреждениям. Предприятию Холлерита сразу же сопутствовал успех, и в дальнейшем оно становилось все более преуспевающим. С годами оно претерпело ряд изменений - слияний и переименований. Последнее такое изменение произошло в **1924** г., за 5 лет до смерти Холлерита, когда была создана фирма **IBM** (International Business Machines Corporation).

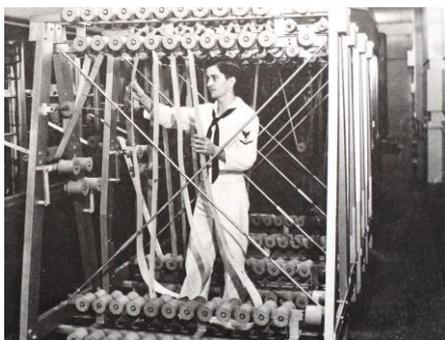


В первые сорок лет XX века в вычислительной технике произошло не так уж много принципиально важных событий. Правда, конструкторы обратили внимание на возможность применения в счетных устройствах

новых элементов – электромагнитных реле.

В середине 30-х годов в Германии инженер **Конрад Цузе** построил вычислительное устройство, работающее на таких реле.

К 1943 году была готова релейная машина «Марк-I» (позднее «Марк-II»), которая воплощала в себе предельные параметры, свойственные этой элементной базе. «Марк-I» имел в длину 15 и в высоту 2,5 метров, содержал 800 тысяч деталей, располагал 60 регистрами для констант, 72 запоминающими регистрами для сложения, центральным блоком умножения и деления, мог вычислять элементарные трансцендентные функции. Машина работала с 23-значными десятичными числами и выполняла операции сложения за 0,3 секунды, а умножения – за 3 секунды.



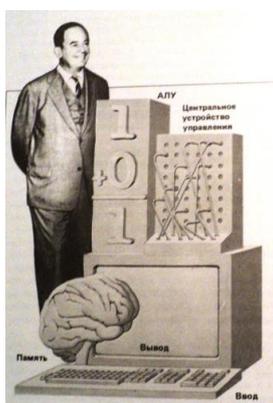
Матрос, обслуживающий машину «Марк-I»

Примерно в то же время в Англии начала работать первая вычислительная машина на реле, которая использовалась для расшифровки сообщений, передававшихся немецким кодированным передатчиком.

#### АРХИТЕКТУРА ФОН НЕЙМАНА

В своем историческом докладе, опубликованном в 1945 г., **Джон фон Нейман** выделил и детально описал пять ключевых компонентов того, что

ныне называют «архитектурой фон Неймана». Чтобы компьютер был и эффективным, и универсальным инструментом, он должен включать следующие структуры: *центральное арифметико-логическое устройство* (АЛУ), *центральное устройство управления*, «дирижирующее» операциями, *запоминающее устройство*, или память, а также *устройства ввода-вывода* информации. Фон Нейман отмечал, что эта система должна работать с двоичными числами, быть электронным, а не механическим устройством и выполнять операции последовательно, одну за другой.



## ПОКОЛЕНИЯ ЭВМ

**Электронная вычислительная машина, ЭВМ** — комплекс технических средств, где основные функциональные элементы (логические, запоминающие, индикационные и др.) выполнены на электронных элементах, предназначенных для автоматической обработки информации в процессе решения вычислительных и информационных задач<sup>4</sup>.

**Компьютер** (англ. Computer — «вычислитель») — устройство или система, способное выполнять заданную, чётко определённую изменяемую последовательность операций. Это чаще всего операции численных расчётов и манипулирования данными, но сюда относятся и операции ввода-вывода. Описание последовательности операций называется **программой**.

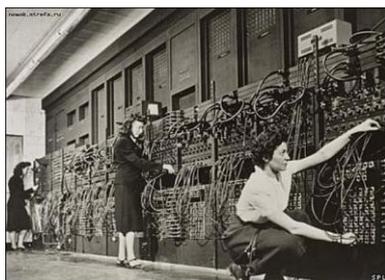
Под термином «**Поколение ЭВМ**» понимают все типы и модели ЭВМ, построенные на одних и тех же научных и технических принципах.

## **ПЕРВОЕ ПОКОЛЕНИЕ ЭВМ**

Машины первого поколения были созданы на основе электронно - вакуумных ламп.

Первая действующая машина, построенная на вакуумных лампах, официально введена в эксплуатацию 15 февраля **1946 года**. В истории компьютеров она известна под названием **ЭНИАК (ENIAC)**, ее авторами были американцы **Эккерт** и **Моучли**. Эту машину пытались использовать для решения некоторых задач, подготовленных фон Нейманом и связанных с проектом атомной бомбы.

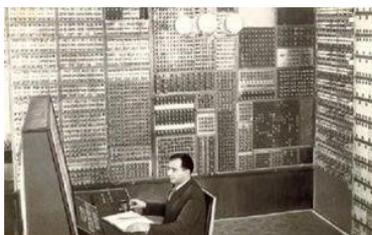
Представление о первом компьютере дают следующие данные. Он содержал 18 тысяч вакуумных ламп, занимал площадь 9-15 метров, весил 30 тонн и потреблял мощность 50 киловатт. ЭНИАК складывал числа за 0,2 миллисекунды, а перемножал за 2,8 миллисекунды. Это было примерно в тысячу раз быстрее, чем на релейных машинах. В компьютере использовалась десятичная система, он оперировал с десятиразрядными числами. ЭНИАК имел существенный недостаток управление им осуществлялось с помощью коммутационной панели. Для изменения программы оператор должен был переключить провода, что занимало не один день. Но подлинным бичом была ужасающая ненадежность компьютера, так как за один день работы успевало выйти из строя около десятка вакуумных ламп.



Новые машины первого поколения сменяли друг друга довольно быстро.

Первый универсальный программируемый компьютер в континентальной Европе был **Z4**, созданный немецким инженером **Конрадом Цузе**. Работа над **Z4** была завершена в сентябре **1950** года.

В **1951** году заработала первая советская электронная вычислительная машина **МЭСМ** («малая электронная счётная машина»), созданная командой учёных под руководством **Сергея Алексеевича Лебедева** из Киевского института электротехники, УССР. Она содержала около 6000 электровакуумных ламп и потребляла 15 кВт. Машина могла выполнять около 3000 операций в секунду.



В **1952** году на свет появилась американская машина **ЭДВАК** (EDVAC). Стоит также отметить построенный ранее, в 1949 году, английский компьютер **ЭДСАК** (EDSAC) – первую машину с хранимой программой.

В **1952** году советские конструкторы ввели в эксплуатацию **БЭСМ** - самую быстродействующую машину в Европе, а в следующем году в СССР начала работать «Стрела» - первая в Европе серийная машина. Среди создателей отечественных машин в первую очередь следует назвать имена С. А. Лебедева, Б. Я. Базилевского, И. С. Брука, Б. И. Рамеева, В. А. Мельникова, М. А. Карцева, А. Н. Мямлина.

Возможности машин первого поколения были достаточно скромны. Так, быстродействие: от 100 («Урал-1») до 20000 операций в секунду (М-20 в 1959 году). Эти цифры определялись в первую очередь инерционностью вакуумных ламп и несовершенством запоминающих устройств. Объем оперативной памяти был крайне мал – в среднем 2048 слов, этого не хватало для размещения сложных программ, не говоря уже о данных. Промежуточная память организовывалась на громоздких и тихоходных магнитных барабанах сравнительно небольшой емкости (5120 слов у БЭСМ-1). Медленно работали и печатающие устройства, а также блоки ввода данных.

В 50-х годах машина **М-20** (главный конструктор – академик С. А. Лебедев) была одной из лучших в мире. На этой машине решалось большинство теоретических и прикладных задач, связанных с развитием самых передовых областей науки и техники. В частности, на комплексах из М-20 обрабатывались данные космических исследований.

ЭВМ первого поколения довольно быстро сошли со сцены, так как не нашли широкого коммерческого применения из-за ненадежности, высокой стоимости, трудности программирования. Это были в основном машины для громоздких расчетов.

## **ВТОРОЕ ПОКОЛЕНИЕ ЭВМ**

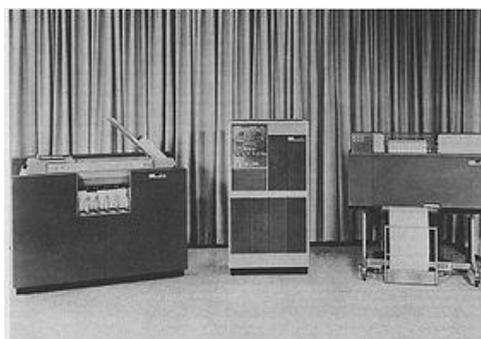
Следующим крупным шагом в истории компьютерной техники стало изобретение **транзистора** в **1947** году. Они стали заменой хрупким и энергоёмким лампам. Машины второго поколения были созданы на основе транзисторов.

Благодаря транзисторам и печатным платам было достигнуто значительное уменьшение размеров и объёмов потребляемой энергии, а также повышение надёжности. Например, IBM 1620 на транзисторах, была размером с офисный стол и ставшая заменой IBM 650 на лампах (цена которой была около 4 млн долл. в пересчёте). В ЭВМ второго поколения впервые стали применять библиотечные программы для решения разнообразных задач, а

также мониторные системы, управляющие трансляцией и исполнением программ.

Однако компьютеры второго поколения по-прежнему были довольно дороги и поэтому использовались только университетами, правительствами, крупными корпорациями.

Компьютеры второго поколения обычно состояли из большого количества печатных плат, каждая из которых содержала от одного до четырёх логических вентилях или триггеров. Второе поколения устройств хранения данных позволяло сохранять уже десятки миллионов символов и цифр.



Компьютер второго поколения **IBM 1401**, выпускавшийся в начале 1960-х, занял около трети мирового рынка компьютеров, было продано более 10 000 таких машин.

«**Сетунь**» была первым компьютером на основе троичной логики, разработана в **1958** году в Советском Союзе. Первыми советскими серийными полупроводниковыми ЭВМ стали «Весна» и «Снег», выпускаемые с 1964 по 1972 год. Пиковая производительность ЭВМ «Снег» составила 300 000 операций в секунду. Машины изготавливались на базе транзисторов с тактовой частотой 5 МГц. Всего было выпущено 39 ЭВМ.

Лучшей отечественной ЭВМ 2-го поколения считается **БЭСМ-6**, созданная в **1966** году.



«Сетунь»



БЭСМ-6

## ТРЕТЬЕ ПОКОЛЕНИЕ ЭВМ

Бурный рост использования компьютеров начался с «третьего поколения» вычислительных машин. Начало этому положило изобретение **интегральной схемы**, которое стало возможным благодаря цепочке открытий сделанных американскими инженерами в 1958—1959 годах. Они решили три фундаментальные проблемы, препятствующие созданию интегральной схемы:

1. **Интеграция.** В 1958 году не существовало способа формирования в кристалле полупроводника множества различных электронных компонентов. Сплавной способ плохо подходил для ИС, новейшая

мезатехнология имела неустранимые проблемы с надёжностью.

2. **Изоляция.** Не существовало эффективного способа электрически изолировать компоненты ИС друг от друга (не считая физической резки кристалла на отдельные приборы).
3. **Соединения.** Не существовало эффективного способа создания электрических соединений между компонентами ИС (не считая чрезвычайно дорогого и трудоёмкого навесного монтажа золотой проволокой).

Элементной базой машин третьего поколения являются интегральные

Элементной базой машин третьего поколения являются  
интегральные схемы.

схемы.

Массовый выпуск интегральных схем начался в 1962 году.

Первая экспериментальная ЭВМ на интегральных схемах была разработана в течение девяти месяцев и завершена в 1961 году. Следующие ее характеристики представляют особый интерес: количество интегральных схем – 587, потребляемая мощность – 16 ватт, вес – 285 грамм, объем – 100 куб. см.

Первая массовая серия машин на интегральных элементах стала выпускаться в 1964 году фирмой IBM. Эта серия известна под названием IBM-360. Она объединяла целое семейство ЭВМ с широким диапазоном производительности, причем совместимых друг с другом. Последнее означало, что машины стало возможно связывать в комплексы, а также без всяких переделок переносить программы, написанные для одной ЭВМ на любую другую из этой серии. Таким образом, впервые выявлено коммерчески выгодное требование стандартизации аппаратного и программного обеспечения.

В СССР первой серийной ЭВМ на интегральных схемах была машина «**Наири-3**», появившаяся в **1970** году.



Со второй половины 60-х годов Советский Союз совместно со странами СЭВ приступил к разработке семейства универсальных машин, аналогичного системе IBM-360. В **1972** году началось серийное производство стартовой, наименее мощной модели Единой системы – **ЭВМ ЕЭ-1010**, а еще через год – пяти других моделей. Их быстродействие находилось в пределах от десяти тысяч (ЕС-1010) до двух миллионов (ЕС-1060) операций в секунду.



Наири -3



ЕС – 1060

В рамках третьего поколения в США была построена уникальная машина ИЛЛИАК-4, в составе которой в первоначальном варианте

планировалось использовать 256 устройств обработки данных, выполненных на монолитных интегральных схемах. ЭВМ должна была обладать быстродействием около одного миллиарда операций в секунду. Однако этой поразительной воображение скорости достигнуть не удалось, так как в процессе реализации проекта выяснилось, что его стоимость (16 миллионов долларов) оказалась заниженной. Поэтому число процессоров пришлось сократить с 256 до 64, также перейти к интегральным схемам с малой степенью интеграции. Сокращенный вариант проекта был завершен в 1972 году, номинальное быстродействие ИЛЛИАК-4 составило 200 миллионов операций в секунду. Почти год этот компьютер был рекордсменом в скорости вычислений, пока в 1973 году не появилась сверхмощная система САТРАН, выполнявшая до 500 миллионов операций в секунду.

Параллельно с компьютерами третьего поколения продолжали выпускаться компьютеры второго поколения.

#### **ЧЕТВЕРТОЕ ПОКОЛЕНИЕ ЭВМ**

К сожалению, начиная с середины 1970-х годов, стройная картина смены поколений нарушается. Все меньше становится 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.* Развитие ЭВМ 4-го поколения пошло по двум направлениям:

1-ое направление — создание суперЭВМ – комплексов многопроцессорных машин.

Быстродействие таких машин достигает нескольких миллиардов операций в секунду. Они способны обрабатывать огромные массивы информации. Сюда входят комплексы ILLIAS-4, CRAY, CYBER, «Эльбрус-1», «Эльбрус-2» и др.



Многопроцессорные вычислительные комплексы (МВК) "Эльбрус-2" активно использовались в Советском Союзе в областях, требующих большого объема вычислений, прежде всего, в оборонной промышленности, управлении космическими полетами, в ядерных исследовательских центрах. Наконец, именно комплексы "Эльбрус-2" с 1991 года использовались в системе противоракетной отрасли. Вычислительные комплексы "Эльбрус-2" эксплуатировались в Центре управления космической обороны и на других военных объектах.



**2-ое направление** — дальнейшее развитие на базе БИС и СБИС микро-ЭВМ и персональных ЭВМ (ПЭВМ).

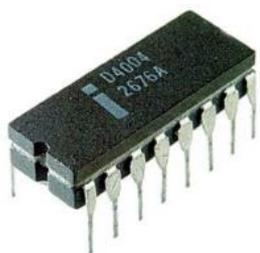
Первыми представителями этих машин являются Apple, IBM - PC (XT, AT, PS/2), «Искра», «Электроника», «Мазовия», «Агат», «ЕС-1840», «ЕС-1841» и др.

Начиная с этого поколения ЭВМ повсеместно стали называть компьютерами. А слово «компьютеризация» прочно вошло в наш быт. Благодаря появлению и развитию персональных компьютеров (ПК),

вычислительная техника становится по-настоящему массовой и общедоступной.



В 1969 году сотрудник компании Intel Тэд Хофф предлагает создать **центральный процессор на одном кристалле**. То есть вместо множества интегральных микросхем создать одну главную интегральную микросхему, которая должна будет выполнять все арифметические, логические операции и операции управления, записанные в машинном коде. Такое устройство получило название — **микропроцессор**. В 1971 году компания Intel выпускает на рынок **первый микропроцессор «Intel 4004»**.



Появление микропроцессоров позволило создать микрокомпьютеры — небольшие недорогие компьютеры, которые могли себе позволить купить маленькие компании или отдельные люди. В 1980-х годах микрокомпьютеры стали повсеместным явлением. Первый массовый домашний компьютер был разработан **Стивом Возняком** — одним из основателей компании Apple Computer. Позже Стив Возняк разработал первый массовый персональный компьютер.



Компьютеры на основе микрокомпьютерной архитектуры, с возможностями, добавленными от их больших собратьев, сейчас доминируют в большинстве сегментов рынка.

Первый по-настоящему мобильный компьютер, спроектированный **Адамом Осборном** специально для работы в поездках, был представлен в **1981 году** и носил название **Osborne 1**. Спрос на первые ноутбуки оказался настолько высок, что компания Адама Осборна стала самой быстрорастущей в Силиконовой Долине. Благодаря уникальным потребительским и техническим характеристикам машины, Osborne Computer Corporation продавала до десяти тысяч экземпляров Osborne 1 ежемесячно. Однако успех не бывает вечным. Руководство компании решило анонсировать новые модели на несколько месяцев раньше срока их реального выхода. Продажи Osborne 1 упали, фирма не смогла выбраться из финансового кризиса и прекратила свое существование.

Необходимо отметить, что впоследствии Compaq, IBM и другие производители компьютерной техники во многом повторяли удачную конструкцию Адама Осборна.



## ПЯТОЕ ПОКОЛЕНИЕ ЭВМ

Программа разработки, так называемого, пятого поколения ЭВМ была принята в Японии в 1982 г. Предполагалось, что к 1991 г. будут созданы принципиально новые компьютеры, ориентированные на решение задач искусственного интеллекта. С помощью языка Пролог и новшеств в конструкции компьютеров планировалось вплотную подойти к решению одной из основных задач этой ветви компьютерной науки - задачи хранения и обработки знаний. Коротко говоря, для компьютеров пятого поколения не пришлось бы писать программ, а достаточно было бы объяснить на "почти естественном" языке, что от них требуется.

Предполагалось, что их элементной базой будут служить не СБИС, а созданные на их базе устройства с элементами искусственного интеллекта. Для увеличения памяти и быстродействия будут использоваться достижения оптоэлектроники и биопроцессоры.

Для ЭВМ пятого поколения ставились совершенно другие задачи, нежели при разработке всех прежних ЭВМ. Если перед разработчиками ЭВМ с I по IV поколений стояли такие задачи, как увеличение производительности в области числовых расчётов, достижение большой ёмкости памяти, то основной задачей разработчиков ЭВМ V поколения являлось создание искусственного интеллекта машины (возможность делать логические выводы из представленных фактов), развитие "интеллектуализации" компьютеров - устранения барьера между человеком и компьютером.

Японский проект ЭВМ пятого поколения повторил трагическую судьбу ранних исследований в области искусственного интеллекта. Более 50-ти миллиардов йен инвестиций были потрачены впустую, проект прекращен, а разработанные устройства по производительности оказались не выше массовых систем того времени. Однако, проведенные в ходе проекта исследования и накопленный опыт по методам представления знаний и

параллельного логического вывода сильно помогли прогрессу в области систем искусственного интеллекта в целом.

Уже сейчас компьютеры способны воспринимать информацию с рукописного или печатного текста, с бланков, с человеческого голоса, узнавать пользователя по голосу, осуществлять перевод с одного языка на другой. Это позволяет общаться с компьютерами всем пользователям, даже тем, кто не имеет специальных знаний в этой области.

Многие успехи, которых достиг искусственный интеллект, используют в промышленности и деловом мире. Экспертные системы и нейронные сети эффективно используются для задач классификации (фильтрация спама, категоризация текста и т.д.). Добросовестно служат человеку генетические алгоритмы (используются, например, для оптимизации портфелей в инвестиционной деятельности), робототехника (промышленность, производство, быт - везде она приложила свою кибернетическую руку), а также многоагентные системы.

**ЛИТЕРАТУРА:** [\[4\]](#), [\[6\]](#), [\[10\]](#), [\[15\]](#), [\[27\]](#)

## **ЗАНЯТИЯ 11-12. КЛАССИФИКАЦИЯ КОМПЬЮТЕРОВ. ТЕНДЕНЦИИ РАЗВИТИЯ ВТ.**

*План*

- 1. Классификация по поколениям*
- 2. Классификация по этапам создания*
- 3. Классификация по принципу действия*

Существуют различные классификации компьютерной техники:

- *по этапам развития (по поколениям);*
- *по архитектуре;*
- *по производительности;*
- *по условиям эксплуатации;*
- *по количеству процессоров;*
- *по потребительским свойствам и т.д.*

**Четких границ между классами компьютеров не существует.** По мере совершенствования структур и технологии производства, появляются новые классы компьютеров, границы существующих классов существенно изменяются.

### **Классификация по поколениям**

**Деление компьютерной техники на поколения — весьма условная, нестрогая классификация вычислительных систем по степени развития аппаратных и программных средств, а также способов общения с компьютером.**

Идея делить машины на поколения вызвана к жизни тем, что за время короткой истории своего развития компьютерная техника проделала большую эволюцию как в смысле **элементной базы** (лампы, транзисторы, микросхемы и др.), так и в смысле **изменения её структуры, появления новых возможностей, расширения областей применения и характера использования.**

Классификация ЭВМ по принципу действия.

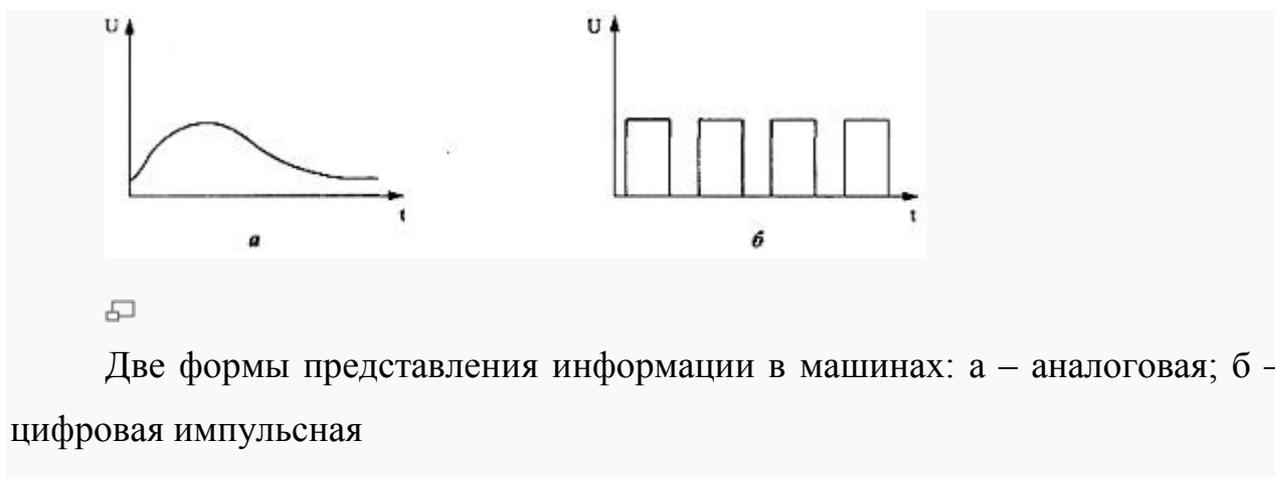
---

Компьютер – комплекс технических средств, предназначенных для автоматической обработки информации в процессе решения вычислительных и информационных задач.

По принципу действия вычислительные машины делятся на три больших класса: аналоговые (АВМ), цифровые (ЦВМ) и гибридные (ГВМ).

Критерием деления вычислительных машин на эти три класса являются форма представления информации, с которой они работают.

- ЦВМ – вычислительные машины дискретного действия, работают с информацией, представленной в дискретной, а точнее, в цифровой форме.
- АВМ - вычислительные машины непрерывного действия, работают с информацией, представленной в непрерывной (аналоговой) форме, то есть в виде непрерывного ряда значений какой-либо физической величины (чаще всего электрического напряжения).
- ГВМ – вычислительные машины комбинированного действия работают с информацией, представленной и в цифровой, и в аналоговой форме; они совмещают в себе достоинства АВМ и ЦВМ. ГВМ целесообразно использовать для решения задач управления сложными быстродействующими техническими комплексами.



Аналоговые вычислительные машины весьма просты и удобны в эксплуатации; программирование задач для решения на них, как правило,

нетрудоемкое; скорость решения задач изменяется по желанию оператора и может быть сделана сколь угодно большой (больше, чем у ЦВМ), но точность решения задач очень низкая (относительная погрешность 2-5 %). На АВМ наиболее эффективно решать математические задачи, содержащие дифференциальные уравнения, не требующие сложной логики.

Наиболее широкое распространение получили ЦВМ с электрическим представлением дискретной информации – электронные цифровые вычислительные машины, обычно называемые просто электронными вычислительными машинами.

#### Классификация ЭВМ по этапам создания.

---

По этапам создания и используемой элементной базе ЭВМ условно делятся на поколения:

- Первое поколение, 50-е годы; ЭВМ на электронных вакуумных лампах.
- Второе поколение, 60-е годы; ЭВМ на дискретных полупроводниковых приборах (транзисторах).
- Третье поколение, 70-е годы; ЭВМ на полупроводниковых интегральных схемах с малой и средней степенью интеграции (сотни – тысячи транзисторов в одном корпусе).
- Четвертое поколение, 80-е годы; ЭВМ на больших и сверхбольших интегральных схемах – микропроцессорах (десятки тысяч – миллионы транзисторов в одном
- Пятое поколение, 90-е годы; ЭВМ с многими десятками параллельно работающих микропроцессоров, позволяющих строить эффективные системы обработки знаний; ЭВМ на сверхсложных микропроцессорах с параллельно-векторной структурой, одновременно выполняющих десятки последовательных команд программы;

- Шестое и последующие поколения; оптоэлектронные ЭВМ с массовым параллелизмом и нейтронной структурой – с распределенной сетью большого числа (десятки тысяч) несложных микропроцессоров, моделирующих архитектуру нейтронных биологических систем.

Каждое следующее поколение ЭВМ имеет по сравнению с предыдущими существенно лучшие характеристики. Так, производительность ЭВМ и емкость всех запоминающих устройств увеличивается, как правило, больше чем на порядок.

### Классификация ЭВМ по назначению

---

По назначению ЭВМ можно разделить на три группы: универсальные (общего назначения), проблемно-ориентированные и специализированные.

**Универсальные ЭВМ** предназначены для решения самых различных инженерно-технических задач: экономических, математических, информационных и других задач, отличающихся сложностью алгоритмов и большим объемом обрабатываемых данных. Они широко используются в вычислительных центрах коллективного пользования и в других мощных вычислительных комплексах.

Характерными чертами универсальных ЭВМ является:

- высокая производительность;
- разнообразие форм обрабатываемых данных: двоичных, десятичных, символьных, при большом диапазоне их изменения и высокой степени их представления;
- обширная номенклатура выполняемых операций, как арифметических, логических, так и специальных;
- большая емкость оперативной памяти;

- развитая организация системы ввода-вывода информации, обеспечивающая подключение разнообразных видов внешних устройств.

**Проблемно-ориентированные ЭВМ** служат для решения более узкого круга задач, связанных, как правило, с управлением технологическими объектами; регистрацией, накоплением и обработкой относительно небольших объемов данных; выполнением расчетов по относительно несложным алгоритмам; они обладают ограниченными по сравнению с универсальными ЭВМ аппаратными и программными ресурсами.

К проблемно-ориентированным ЭВМ можно отнести, в частности, всевозможные управляющие вычислительные комплексы.

**Специализированные ЭВМ** используются для решения узкого круга задач или реализации строго определенной группы функций. Такая узкая ориентация ЭВМ позволяет четко специализировать их структуру, существенно снизить их сложность и стоимость при сохранении высокой производительности и надежности их работы.

К специализированным ЭВМ можно отнести, например, программируемые микропроцессоры специального назначения; адаптеры и контроллеры, выполняющие логические функции управления отдельными несложными техническими устройствами согласования и сопряжения работы узлов вычислительных систем. К таким компьютерам также относятся, например, бортовые компьютеры автомобилей, судов, самолетов, космических аппаратов. Бортовые компьютеры управляют средствами ориентации и навигации, осуществляют контроль за состоянием бортовых систем, выполняют некоторые функции автоматического управления и связи, а также большинство функций оптимизации параметров работы объекта (например, оптимизацию расхода топлива объекта в зависимости от конкретных условий движения). Специализированные мини-ЭВМ, ориентированные на работу с графикой, называют графическими станциями. Специализированные

компьютеры, объединяющие компьютеры предприятия в одну сеть, называют файловыми серверами. Компьютеры, обеспечивающие передачу информации между различными участниками всемирной компьютерной сети, называют сетевыми серверами.

Во многих случаях с задачами специализированных компьютерных систем могут справляться и обычные универсальные компьютеры, но считается, что использование специализированных систем все-таки эффективнее. Критерием оценки эффективности выступает отношение производительности оборудования к величине его стоимости.

#### Классификация ЭВМ по размерам и функциональным возможностям

---

По размерам и функциональным возможностям ЭВМ можно разделить на **сверхбольшие, большие, малые, сверхмалые (микроЭВМ)**.

Функциональные возможности ЭВМ обуславливают важнейшие технико-эксплуатационные характеристики:

- быстродействие, измеряемое усредненным количеством операций, выполняемых машиной за единицу времени;
- разрядность и формы представления чисел, с которыми оперирует ЭВМ;
- номенклатура, емкость и быстродействие всех запоминающих устройств;
- номенклатура и технико-экономические характеристики внешних устройств хранения, обмена и ввода-вывода информации;
- типы и пропускная способность устройств связи и сопряжения узлов ЭВМ между собой (внутримашинного интерфейса);
- способность ЭВМ одновременно работать с несколькими пользователями и выполнять одновременно несколько программ (многопрограммность);

- типы и технико-эксплуатационные характеристики операционных систем, используемых в машине;
- наличие и функциональные возможности программного обеспечения;
- способность выполнять программы, написанные для других типов ЭВМ (программная совместимость с другими типами ЭВМ);
- система и структура машинных команд;
- возможность подключения к каналам связи и к вычислительной сети;
- эксплуатационная надежность ЭВМ;
- коэффициент полезного использования ЭВМ во времени, определяемый соотношением времени полезной работы и времени профилактики.

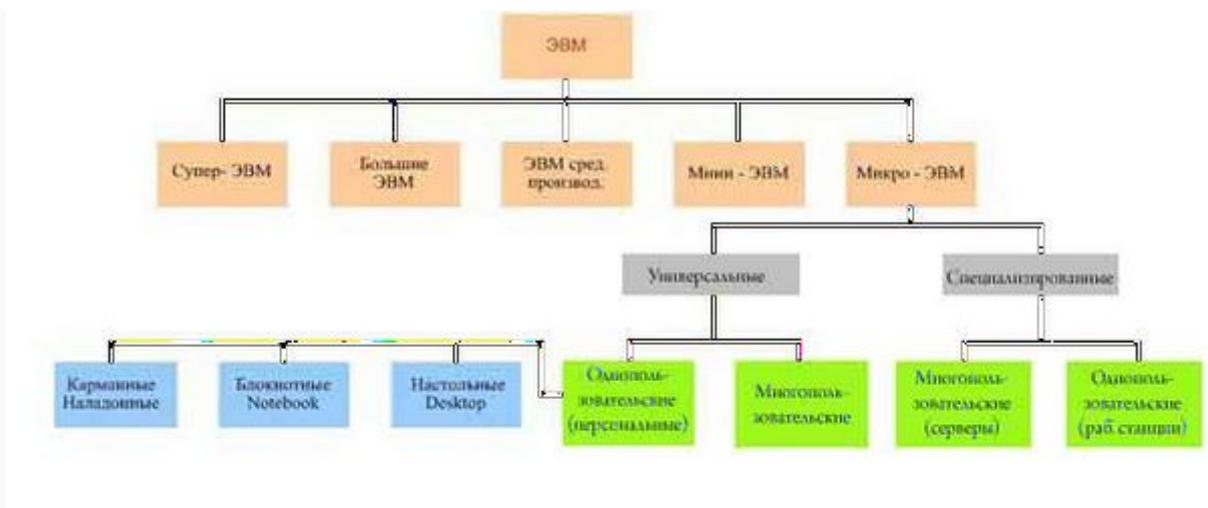


Схема классификации ЭВМ, исходя из их вычислительной мощности и габаритов.

Исторически первыми появились большие ЭВМ, элементная база которых прошла путь от электронных ламп до интегральных схем со сверхвысокой степенью интеграции. Первая большая ЭВМ ЭНИАК была

создана в 1946 году. Эта машина имела массу более 50 т., быстродействие несколько сотен операций в секунду, оперативную память емкостью 20 чисел; занимала огромный зал площадью 100 кв.м.

Производительность больших ЭВМ оказалась недостаточной для ряда задач: прогнозирования метеообстановки, управления сложными оборонными комплексами, моделирования экологических систем и др. Это явилось предпосылкой для разработки и создания суперЭВМ, самых мощных вычислительных систем, интенсивно развивающихся и в настоящее время.

Появление в 70-х годах малых ЭВМ обусловлено, с одной стороны, прогрессом в области электронной элементной базы, а с другой – избыточностью ресурсов больших ЭВМ для ряда приложений. Малые ЭВМ используются чаще всего для управления технологическими процессами. Они более компактны и значительно дешевле больших ЭВМ.

Дальнейшие успехи в области элементной базы и архитектурных решений привели к возникновению супермини-ЭВМ – вычислительной машины, относящейся по архитектуре, размерам и стоимости к классу малых ЭВМ, но по производительности сравнимой с большой ЭВМ.

Изобретение в 1969 году микропроцессора привело к появлению в 70-х годах еще одного класса ЭВМ – микроЭВМ. Именно наличие микропроцессора служило первоначально определяющим признаком микроЭВМ. Сейчас микропроцессоры используются во всех без исключения классах ЭВМ.

### **СуперЭВМ**

К СуперЭВМ относятся мощные многопроцессорные вычислительные машины с быстродействием сотни миллионов – десятки миллиардов операций в секунду.

Типовая модель суперЭВМ 2000 г. по прогнозу будет иметь следующие характеристики:

- высокопараллельная многопроцессорная вычислительная система с быстродействием примерно 100000 MFLOPS;
- емкость: оперативной памяти 10 Гбайт, дисковой памяти 1 – 10 Тбайт (или 1000 Гбайт);
- разрядность 64; 128 бит.

Фирма Cray Research намерена в 2000 г. создать суперЭВМ производительностью 1 TFLOPS = 1000000 MFLOPS.

Создать такую высокопроизводительную ЭВМ по современной технологии на одном микропроцессоре не представляется возможным в виду ограничения, обусловленного конечным значением скорости распространения электромагнитных волн (300000 км/с), ибо время распространения сигнала на расстояние несколько миллиметров (линейный размер стороны микропроцессора) при быстродействии 100 млрд. оп/с становится соизмеримым с временем выполнения одной операции. Поэтому суперЭВМ создаются в виде высокопараллельных многопроцессорных вычислительных систем (МПВС).

Высокопараллельные МПВС имеют несколько разновидностей:

- магистральные (конвейерные) МПВС, в которых процессоры одновременно выполняют разные операции над последовательным потоком обрабатываемых данных; по принятой классификации такие МПВС относятся к системам с многократным потоком команд и однократным потоком данных (МКОД или MISD)
- векторные МПВС, в которых все процессоры одновременно выполняют одну команду над различными данными – однократный поток команд с многократным потоком данных (ОКМД или SIMD).

- матричные МПВС, в которых микропроцессоры одновременно выполняют разные операции над несколькими последовательными потоками обрабатываемых данных (МКМД или МIMD).

В суперЭВМ используются все три варианта архитектуры МПВС:

- структура МIMD в классическом ее варианте (например, в суперкомпьютере BSP фирмы Burroughs

- параллельно-конвейерная модификация, иначе, MMISD, т.е. многопроцессорная MISD- архитектура (например, в суперкомпьютере «Эльбрус 3»).

- параллельно-векторная модификация, иначе, MSIMD, т.е. многопроцессорная SIMD-архитектура (например, в суперкомпьютере Cray 2).

Наибольшую эффективность показала MSIMD-архитектура, поэтому в современных суперЭВМ чаще всего используется именно она (суперкомпьютеры фирм Cray, Fujitsu, NEC, Hitachi и др.)

### **Большие ЭВМ**

Это самые мощные компьютеры. Их применяют для обслуживания очень крупных организаций и даже целых отраслей народного хозяйства. За рубежом компьютеры этого класса называют мэйнфреймами (mainframe). В России за ними закрепился термин **большие ЭВМ**. Штат обслуживания большой ЭВМ составляет до многих десятков человек. На базе таких суперкомпьютеров создают вычислительные центры, включающие в себя несколько отделов или групп:

- **Центральный процессор** — основной блок ЭВМ, в котором непосредственно и происходит обработка данных и вычисление результатов. Обычно центральный процессор представляет собой

несколько стоек аппаратуры и размещается в отдельном помещении, в котором соблюдаются повышенные требования по температуре, влажности, защищенности от электромагнитных помех, пыли и дыма.

- **Группа системного программирования** занимается разработкой, отладкой и внедрением программного обеспечения, необходимого для функционирования самой вычислительной системы. Работников этой группы называют системными программистами. Они должны хорошо знать техническое устройство всех компонентов ЭВМ, поскольку их программы предназначены в первую очередь для управления физическими устройствами. Системные программы обеспечивают взаимодействие программ более высокого уровня с оборудованием, то есть группа системного программирования обеспечивает программно-аппаратный интерфейс вычислительной системы.

- **Группа прикладного программирования** занимается созданием программ для выполнения конкретных операций с данными. Работников этой группы называют прикладными программистами. В отличие от системных программистов им не надо знать техническое устройство компонентов ЭВМ, поскольку их программы работают не с устройствами, а с программами, подготовленными системными программистами. С другой стороны, с их программами работают пользователи, то есть конкретные исполнители работ. Поэтому можно говорить о том, что группа прикладного программирования обеспечивает пользовательский интерфейс вычислительной системы.

- **Группа подготовки данных** занимается подготовкой данных, с которыми будут работать программы, созданные прикладными программистами. Во многих случаях сотрудники этой группы сами вводят данные с помощью клавиатуры, но они могут выполнять и преобразование готовых данных из одного вида в другой. Так, например, они могут получать иллюстрации, нарисованные художниками на бумаге, и

преобразовывать их в электронный вид с помощью специальных устройств, называемых сканерами.

- **Группа технического обеспечения** занимается техническим обслуживанием всей вычислительной системы, ремонтом и наладкой устройств, а также подключением новых устройств, необходимых для работы прочих подразделений.

- **Группа информационного обеспечения** обеспечивает технической информацией все прочие подразделения вычислительного центра по их заказу. Эта же группа создает и хранит архивы ранее разработанных программ и накопленных данных. Такие архивы называют библиотеками программ или банками данных.

- **Отдел выдачи данных** получает данные от центрального процессора и преобразует их в форму, удобную для заказчика. Здесь информация распечатывается на печатающих устройствах (принтерах) или отображается на экранах дисплеев.

К мейнфреймам относятся, как правило, компьютеры, имеющие следующие характеристики:

- производительность не менее 10 MIPS;
- основную память емкостью от 64 до 10000 MIPS;
- внешнюю память не менее 50 Гбайт;
- многопользовательский режим работы (обслуживают одновременно от 16 до 1000 пользователей).

Основные направления эффективного применения мейнфреймов – это решение научно-технических задач, работа в вычислительных системах с пакетной обработкой информации, работа с большими базами данных, управление вычислительными сетями и их ресурсами. Последнее направление – использование мейнфреймов в качестве больших серверов вычислительных сетей часто отмечается специалистами среди наиболее актуальных.

Большие ЭВМ отличаются высокой стоимостью оборудования и обслуживания, поэтому работа таких суперкомпьютеров организована по непрерывному циклу. Наиболее трудоемкие и продолжительные вычисления планируют на ночные часы, когда количество обслуживающего персонала минимально. В дневное время ЭВМ исполняет менее трудоемкие, но более многочисленные задачи. При этом для повышения эффективности компьютер работает одновременно с несколькими задачами и, соответственно, с несколькими пользователями. Он поочередно переключается с одной задачи на другую и делает это настолько быстро и часто, что у каждого пользователя создается впечатление, будто компьютер работает только с ним. Такое распределение ресурсов вычислительной системы носит название принципа разделения времени.

Родоначальником современных больших ЭВМ, по стандартам которой в последние несколько десятилетий развивались ЭВМ этого класса в большинстве стран мира, является фирма IBM.

Среди лучших современных разработок мейнфреймов за рубежом в первую очередь следует отметить: американский IBM 390, IBM 4300, (4331, 4341, 4361, 4381), пришедшие на смену IBM 380 в 1979 году, и IBM ES/9000, созданные в 1990 году, а также японские компьютеры М 1800 фирмы Fujitsu.

### **Мини**

Надежные, недорогие и удобные в эксплуатации компьютеры, обладающие несколько более низкими по сравнению с мейнфреймами возможностями и, соответственно меньшей стоимостью. Такие компьютеры используются крупными предприятиями, научными учреждениями и некоторыми высшими учебными заведениями, сочетающими учебную деятельность с научной. Мини-ЭВМ (и наиболее мощные из них супермини-ЭВМ) обладают следующими характеристиками:

- производительность до 100 MIPS;

- емкость основной памяти – 4-512 Мбайт;
- емкость дисковой памяти - 2-100 Гбайт;
- число поддерживаемых пользователей – 16-512.

Все модели мини-ЭВМ разрабатываются на основе микропроцессорных наборов интегральных микросхем, 16-, 32-, 64-разрядных микропроцессоров. Основные их особенности: широкий диапазон производительности в конкретных условиях применения, аппаративная реализация большинства системных функций ввода-вывода информации, простая реализация микропроцессорных и многомашинных систем, высокая скорость обработки прерываний, возможность работы с форматами данных различной длины.

К достоинствам мини-ЭВМ можно отнести: специфичную архитектуру с большой модульностью, лучше, чем у мейнфреймов, соотношение производительность/цена, повышенная точность вычислений.

Мини-ЭВМ ориентированы на использование в качестве управляющих вычислительных комплексов. Традиционная для подобных комплексов широкая номенклатура периферийных устройств дополняется блоками межпроцессорной связи, благодаря чему обеспечивается реализация вычислительных систем с изменяемой структурой.

Мини-ЭВМ часто применяют для управления производственными процессами. Например, в механическом цехе компьютер может поддерживать ритмичность подачи заготовок, узлов и комплектующих на рабочие места, управлять гибкими автоматизированными линиями и промышленными роботами, собирать информацию с инструментальных постов технического контроля и сигнализировать о необходимости замены изношенных инструментов и приспособлений, готовить данные для станков с числовым программным управлением, а также своевременно информировать цеховые и заводские службы о необходимости выполнения мероприятий по переналадке оборудования. Например, он может помогать экономистам в осуществлении

контроля за себестоимостью продукции, нормировщикам в оптимизации времени технологических операций, конструкторам в автоматизации проектирования станочных приспособлений, бухгалтерии в осуществлении учета первичных документов и подготовки регулярных отчетов для налоговых органов. Для организации работы с мини-ЭВМ тоже требуется специальный вычислительный центр, хотя и не такой многочисленный, как для больших ЭВМ.

Наряду с использованием для управления технологическими процессами мини-ЭВМ успешно применяется для вычислений в многопользовательских вычислительных системах, в системах автоматизированного проектирования, в системах моделирования несложных объектов, в системах искусственного интеллекта.

### **МикроЭВМ**

Компьютеры данного класса доступны многим предприятиям. Организации, использующие микро-ЭВМ, обычно не создают вычислительные центры. Для обслуживания такого компьютера им достаточно небольшой вычислительной лаборатории в составе нескольких человек. В число сотрудников вычислительной лаборатории обязательно входят программисты, хотя напрямую разработкой программ они не занимаются. Необходимые системные программы обычно покупают вместе с микроЭВМ, а разработку нужных прикладных программ заказывают более крупным вычислительным центрам или специализированным организациям.

Программисты вычислительной лаборатории занимаются внедрением приобретенного или заказанного программного обеспечения, выполняют его доводку и настройку, согласовывают его работу с другими программами и устройствами компьютера. Хотя программисты этой категории и не разрабатывают системные и прикладные программы, они могут вносить в них изменения, создавать или изменять отдельные фрагменты. Это требует высокой квалификации и универсальных знаний. Программисты,

обслуживающие микро-ЭВМ, часто сочетают в себе качества системных и прикладных программистов одновременно.

Можно привести следующую классификацию микроЭВМ:

### **Универсальные**

Многопользовательские микроЭВМ – это мощные микроЭВМ, оборудованные несколькими видеотерминалами и функционирующие в режиме разделения времени, что позволяет эффективно работать на них сразу нескольким пользователям.

Персональные компьютеры(ПК) – однопользовательские микроЭВМ удовлетворяющие требованиям общедоступности и универсальности применения, рассчитанные на одного пользователя и управляемые одним человеком. Персональный компьютер должен удовлетворять следующим требованиям:

- стоимость от нескольких сотен до 5-10 тысяч долларов;
- наличие внешних ЗУ на магнитных дисках;
- объём оперативной памяти не менее 4 Мбайт;
- наличие операционной системы;
- способность работать с программами на языках высокого уровня;
- ориентация на пользователя-непрофессионала (в простых моделях).

Портативные компьютеры обычно нужны руководителям предприятий, менеджерам, учёным, журналистам, которым приходится работать вне офиса — дома, на презентациях или во время командировок.

Основные разновидности портативных компьютеров:

**Laptop** (наколенник, от lap — колено и top — вверх). По размерам близок к обычному портфелю. По основным характеристикам

(быстродействие, память) примерно соответствует настольным ПК. Сейчас компьютеры этого типа уступают место ещё меньшим.

**Notebook** (блокнот, записная книжка). По размерам он ближе к книге крупного формата. Имеет вес около 3 кг. Помещается в портфель-дипломат. Для связи с офисом его обычно комплектуют модемом. Ноутбуки зачастую снабжают приводами CD-ROM. Многие современные ноутбуки включают взаимозаменяемые блоки со стандартными разъёмами. Такие модули предназначены для очень разных функций. В одно и то же гнездо можно по мере надобности вставлять привод компакт-дисков, накопитель на магнитных дисках, запасную батарею или съёмный винчестер. Ноутбук устойчив к сбоям в энергопитании. Даже если он получает энергию от обычной электросети, в случае какого-либо сбоя он мгновенно переходит на питание от аккумуляторов.

**Palmtop** (наладонник) — самые маленькие современные персональные компьютеры. Умещаются на ладони. Магнитные диски в них заменяет энергонезависимая электронная память. Нет и накопителей на дисках — обмен информацией с обычными компьютерами идет линиям связи. Если Palmtop дополнить набором деловых программ, записанных в его постоянную память, получится персональный цифровой помощник (Personal Digital Assistant).

### **Специализированные**

Рабочие станции представляют собой однопользовательские мощные микроЭВМ, специализированные для выполнения определенного вида работ (графических, инженерных, издательских и др.)

Несмотря на относительно невысокую производительность по сравнению с большими ЭВМ, микро-ЭВМ находят применение и в крупных вычислительных центрах. Там им поручают вспомогательные операции, для

которых нет смысла использовать дорогие суперкомпьютеры. К таким задачам, например, относится предварительная подготовка данных.

## **Серверы**

Серверы – многопользовательские мощные микроЭВМ в вычислительных сетях, выделенные для обработки запросов от всех станций сети.

Серверы обычно относят к микроЭВМ, но по своим характеристикам мощные серверы скорее можно отнести к малым ЭВМ и даже к мэйнфреймам, а суперсерверы приближаются к суперЭВМ.

Сервер – выделенный для обработки запросов от всех станций вычислительной сети компьютер, предоставляющий этим станциям доступ к общим системным ресурсам (вычислительным мощностям, базам данных, библиотекам программ, принтерам, факсам и др.) и распределяющий эти ресурсы. Такой универсальный сервер часто называют **сервером приложений**.

Серверы в сети часто специализируются. Специализированные серверы используются для устранения наиболее "узких" мест в работе сети: создание и управление базами данных и архивами данных, поддержка многоадресной факсимильной связи и электронной почты, управление многопользовательскими терминалами (принтеры, плоттеры) и др.

**Файл-сервер** ( File Server ) используется для работы с файлами данных, имеет объемные дисковые запоминающие устройства, часто на отказоустойчивых дисковых массивах RAID емкостью до 1 Тбайта.

**Архивационный сервер** (сервер резервного копирования, Storage Express System ) служит для резервного копирования информации в крупных многосерверных сетях, использует накопители на магнитной ленте (стриммеры) со сменными картриджами емкостью до 5 Гбайт; обычно выполняет ежедневное автоматическое архивирование со сжатием

информации от серверов и рабочих станций по сценарию, заданному администратором сети (естественно, с составлением каталога архива).

**Факс-сервер** ( Net SatisFaxion ) – выделенная рабочая станция для организации эффективной многоадресной факсимильной связи с несколькими факс-модемными платами, со специальной защитой информации от несанкционированного доступа в процессе передачи, с системой хранения электронных факсов.

**Почтовый сервер** ( Mail Server ) – то же, что и факс-сервер, но для организации электронной почты, с электронными почтовыми ящиками.

**Сервер печати** ( Print Server , Net Port ) предназначен для эффективного использования системных принтеров.

**Сервер телеконференций** имеет систему автоматической обработки видеоизображений и др.

### **Рабочая станция**

Рабочей станцией называется совокупность аппаратных и программных средств, предназначенных для решения профессиональных задач. Это специализированный высокопроизводительный компьютер для тех, кому необходима надежная и производительная система, гарантирующая стабильную и эффективную работу приложений. Использование рабочих станций позволяет вывести ваше предприятие на новый профессиональный уровень вне зависимости от того, в какой области вы развиваетесь.

Рабочие станции решают широкий спектр задач:

- Инженерно-технические задачи – 3D-проектирование и конструирование, расчетные работы.
- Профессиональная работа с трехмерной графикой – визуализация, 3D-моделирование, мультипликация, спецэффекты.

- Цифровая обработка фото и видео материала - верстка, монтаж, дизайн.
- Работа с большими объемами данных – статистика, аналитика, прогнозирование.

Основные преимущества:

- Эффективность

Решения, использующие последние технологии, позволяют рабочим станциям более эффективно справиться с высокими вычислительными нагрузками. Рабочие станции адаптированы на решение профессиональных задач за счет оптимизации как аппаратной части, так и драйверов.

- Надежность

Повышенная надежность достигается за счет использования только высококачественной компонентной базы, длительному стресс-тестированию на этапе разработки и тотальному контролю качества при производстве изделия.

- Специализация

Отдельным сегментом в линейке рабочих станций являются графические станции, оснащаемые профессиональными видеоадаптерами, созданными специально для решения профессиональных задач, связанных со сложной визуализацией, конструированием и 3D-моделированием, разработкой и производством, созданием медиа контента и научной деятельностью.

- Адаптация к программному обеспечению

Графические станции проходят тестирование и сертифицирование на совместимость и эффективную работу с приложениями от ведущих разработчиков профессионального профильного программного обеспечения, таких как Catia и SolidWorks от Dassault Systemes, AutoCAD и Inventor от

Autodesk, Компас 3D от Аскон, ProEngineer от ProTechnologies, NX от Siemens PLM Software, с продуктами компаний ANSYS, Adobe и многих других.

- Возможности расширения

Платформы рабочих станций предоставляют большую гибкость в модернизации. Большое количество слотов PCI и PCI-E дает возможность установки профильных плат расширения. Большое количество слотов памяти и возможность установки второго процессора в двухпроцессорных системах увеличивает диапазон выбора производительности.

---

Конечно, вышеприведенная классификация весьма условна, ибо мощный современный персональный компьютер, оснащенный проблемно-ориентированным программным и аппаратным обеспечением, может использоваться и как полноправная рабочая станция, и как многопользовательная микроЭВМ, и как хороший сервер, но по своим характеристикам почти не уступающий малым ЭВМ.

**ЛИТЕРАТУРА:** [\[4\]](#), [\[6\]](#), [\[10\]](#), [\[15\]](#), [\[27\]](#)

## **ЗАНЯТИЯ 13-14. АРХИТЕКТУРА И ПРИНЦИПЫ РАБОТЫ ЭВМ**

*План*

- 1. Архитектура ПК*
- 2. Принципы работы ПК*

**Архитектура ЭВМ** - это описание совокупности устройств, блоков ЭВМ

*и принципов взаимодействия компонентов компьютера.*

Основными функциями ЭВМ является обработка и хранение информации, а также обмен информацией с внешними устройствами. В основе функционирования ЭВМ лежит принцип программного управления (*ЭВМ выполняет программу автоматически без вмешательства человека*), то есть вычислительная машина должна руководствоваться программой с последовательным выполнением команд из этой программы, а сама программа, как и сами данные, должна храниться в памяти ЭВМ в виде кодов 0 и 1. Эти принципы работы ЭВМ сформулировал математик *фон Нейман* в 1946 г.

Итак, программы - это управляющие средства компонентами компьютера, которые обеспечивают функционирование компьютера и обработку информации.

**Команда** - это инструкция компонентам ЭВМ, о том, что они должны делать на каждом шагу.

**Программа** - это упорядоченная совокупность команд (*директив*), которые может выполнять ЭВМ в автоматическом режиме.

По фон Нейману вычислительная машина должна состоять из таких основных компонентов.

1. *Запоминающее устройство*, в котором можно было бы записывать двоичные коды и считывать их. Это устройство называется *оперативной памятью* (ОП). Выполнена она в виде ячеек, каждая из которых имеет свой номер, то есть адрес. По адресу можно обратиться к нужной ячейки в операциях записи-считывания.

2. *Арифметико-логическое устройство* (АЛУ). Устройство предназначенное для автоматического выполнения определенного набора арифметических и логических операций над числовыми и символьными данными.

3. *Устройство управления (УУ)*. Это устройство обеспечивает чтение и запись информации в ячейки памяти. Оно также формирует и подает сигналы во все блоки ЭВМ для управления их работой, например, координирует работу АЛУ и внешних устройств. Микропроцессоры в современных компьютерах на одной микросхеме сочетают в себе АЛУ и УУ.

4. *Внешние устройства*. Это, прежде всего, устройства ввода и вывода информации. Такими устройствами являются клавиатура, монитор, принтер и тому подобное.

**ЛИТЕРАТУРА:** [4], [6], [10], [15], [27]

**Занятия 15-16. Системная плата ПК. Магистрально-модульный принцип.**

*План*

1. *Общая характеристика.*

2. *Материнская плата*

**Персональный компьютер** - это многофункциональная, однопользовательская, малогабаритная вычислительная машина, которая предназначена для решения задач обработки и хранения информации.

Основными составными частями персонального компьютера являются:

*системный блок, монитор, клавиатура, манипулятор мыши.* Они составляют минимальную конфигурацию современного персонального компьютера.

Модульный способ конструирования персонального компьютера вместе с магистральным способом обмена информацией и определяет *магистрально-модульный принцип построения компьютера*.

Функциональная схема ПК выглядит, который приведен на рис.



Различные узлы компьютера связаны с микропроцессором и между собой через устройство, которое называется *системной шиной*. Обмен данными происходит через системную шину, которую еще называют *магистралью*.

Магистраль содержит следующие шины:

1. *Шина данных*, по которой информация (данные) передается от МП к любому устройству или наоборот от устройства к МП.
2. *Шина адреса* - совокупность проводов и соответствующих схем, по которым передается в параллель все коды адреса ячейки ОП или портов ввода / вывода.
3. *Шина управления* содержит провода для передачи управления (управляющих сигналов) со стороны микропроцессора во все блоки ПК.
4. *Шина питания*, содержит провода и схемы для подключения блоков ПК к системе электрического питания.

Разрядность шины данных определяет *разрядность компьютера*. например, если шиной данных передается 32 бита в параллель, то ПК является 32- разрядным. Разрядность влияет на производительность компьютера.

Разрядность адресной шины определяет *адресное пространство* - максимальное количество ячеек ОП. Количество адресованных ячеек составляет  $2^n$ , где  $n$  - разрядность адресной шины. Для современных компьютеров используется 32-разрядная адресная шина (для процессоров семьи Pentium) и 64-разрядная (для процессоров семьи Itanium). Разрядность адресной шины определяет максимальное число адресов, по которым может обратиться процессор, т. е. число линий в адресной шине показывает, каким объемом памяти может управлять процессор.

Системная шина обеспечивает три направления передачи информации:

- 1) между микропроцессором и ОП;
- 2) между процессорами и портами ввода / вывода внешних устройств;
- 3) между основной памятью и внешними устройствами (в режиме прямого доступа к памяти).

Благодаря наличию системной шины IBM-совместимые ПК имеют принцип открытой архитектуры, то есть они состоят из нескольких модулей, которые изготавливаются в виде отдельных плат. Модульная структура позволяет пользователю самому комплектовать нужную ему конфигурацию компьютера и облегчает модернизацию компьютера и его ремонт.

Один из важных модулей - это *материнская плата*, на которой размещены микропроцессор, оперативная память, системная шина и слоты расширения для подключения других модулей. Этими модулями являются электронные платы- контроллеры внешних устройств, например, плата видеоконтроллера (видеокарта), создающая сигналы для монитора.

## **1. Общая характеристика**

Все основные узлы ПК расположены внутри системного блока. Системный блок является основным узлом компьютера, он содержит:

- *электронные схемы, управляющие работой ПК (процессор, память, системная плата, контроллеры устройств и т.п.);*

- *накопители на жестких и гибких магнитных дисках, накопители на оптических дисках;*

- *блок питания, который преобразует переменное напряжение сети на низкое постоянное напряжение, необходимое для работы электронных схем и двигателей приводов дисководов и т.д.;*

- *систему охлаждения (вентиляторы и радиаторы), которая обеспечивает необходимый температурный режим.*

Оборудование, размещенное снаружи системного блока, относится ко внешним устройствам ввода / вывода. Это оборудование также называют *периферийными устройствами.*

На фронтальной стороне системного блока находятся:

- *кнопка Power* - для включения / выключения ПК;

- *кнопка Reset* - для перезагрузки компьютера при его зависании, то есть когда в результате ошибки в работе ПК он перестает выполнять ваши команды;

- *два индикатора (лампочки, которые светятся): индикатор питания - горит постоянно и индикатор работы жесткого диска – горит тогда, когда на диск записывается информация или считывается из него.*

На передней панели находится дисковод CD или DVD. Кроме этого, на фронтальную панель для удобства выводят еще дополнительные разъемы (USB, звуковые).

На задней стороне системного блока находится множество различных разъемных соединений для подключения внешних устройств. Это

- два больших разъемных соединения для подключения питания системного блока и монитора (порой эти устройства питаются отдельно),
- разъемное соединения звуковой карты - для подключения колонок, микрофона,
- разъемное соединения видеокарты - для подключения монитора,
- параллельный LPT-порт (в параллель передаются 8 импульсов, несущих информацию) для подключения принтера,
- последовательный СОМ-порт - для подключения модема,
- гнезда PS / 2 - для подключения клавиатуры и мыши (разъемные соединения разного цвета).

Гнезда выполняются различной формы с тонкими штырьками или с дырочками. Поскольку разъемное соединение уникальное то перепутать подключения устройств невозможно.

Корпусы системного блока бывают вертикальные и горизонтальные (Desktop). Современными считаются вертикальные Midi Tower (средний) и Big Tower (большой) корпуса.

Блок питания, как правило, уже встроен в корпус. Блоки питания различаются по мощности: 250 Вт, 300 Вт, 350 Вт, 400 Вт (лучше всего). Этим мощностям должно быть достаточно, чтобы обеспечить энергопотребление всех подключенных к ПК устройств. От эффективности блока питания зависит и стабильность работы всей системы.

## **2. Материнская плата**

Важнейшим узлом ПК является системная (материнская) плата. Основная функция материнской платы - осуществлять взаимосвязь между устройствами ПК. За всеми устройствами компьютера нужен контроль, их работу надо координировать.

**Материнская плата** - это основная электронная схема ПК, от работы которой зависит быстрдействие компьютера и стабильность его работы.

Устройства, из которых состоит материнская плата:

- *системная шина* - магистраль, которая связывает устройства ПК в единое целое. Именно по шине передаются сигналы управления и данные;

- *базовый набор микросхем логики* - чипсет, с помощью которого материнская плата осуществляет контроль над всем, что происходит внутри системного блока. В каждом чипсете есть два моста (чипа): северный, соединяющий между собой процессор, оперативную память и видеошину AGP (Accelerated Graphics Port) и южный, отвечающий за работу со всеми подключенными к этой шине периферийными устройствами. Чипсет является основой любой материнской платы, от него зависит тип процессора, тип памяти и производительность материнской платы;

- *схема BIOS*. Основная функция BIOS - это управление стандартными периферийными устройствами, в частности, дисковыми, клавиатурой, принтером, таймером и т.д. Кроме этого, BIOS отыскивает и загружает в ОП программу-загрузчик операционной системы с системного диска в ОП и осуществляет тестирование аппаратуры компьютера.

Остальные элементы размещаются на отдельных платах и вставляются в разъемные соединения на материнской плате - так называемые *слоты*, имеющие вид длинных гнезд. Количество слотов расширения определяет сколько можно вставить в компьютер дополнительных плат. Видеокарта подключается через специальный слот под названием AGP или PCI Express (Peripheral Component Interconnect), остальные слоты называются PCI. На материнской плате есть слоты для установки ОП. этих слотов может быть от 1 до 4, что позволяет иметь до 4 Гб оперативной памяти. Слоты четко привязаны к типу ОП.

Микропроцессоры устанавливаются на материнской плате в квадратные гнезда, называемые *сокетами*. Эти гнезда похожи между собой, но они

отличаются количеством ножек. Для современных процессоров Pentium IV и Celeron используются материнские платы с гнездами Socket 478. Для различных групп микропроцессоров существуют различные материнские платы с соответствующими гнездами для микропроцессоров, например для процессоров AMD Socket AM2 и AM2 +. Итак, материнскую плату нужно выбирать в соответствии с микропроцессором.

Кроме этого, на материнской плате находятся разъемы (слоты) для установления модулей оперативной памяти, разъем для подключения накопителей жестких дисков, дисководов CD, DVD, FDD, разъем для подключения электропитания. На заднюю стенку ПК с материнской платы выведены разъемные соединения, называемые портами для подключения внешних устройств. Существуют параллельные (LPT) и последовательные (COM) порты. Для последовательного порта присуща последовательная передача данных (бит за битом), а для параллельного - одновременная передача нескольких битов в параллель (по 8 бит).

Параллельный порт предназначен для подключения принтера, сканера. Для него характерна высокая скорость передачи данных - 2 Мб / с. Последовательные порты предназначены для поддержки мыши, модема. для них характерна малая скорость передачи данных - до 112 Кб / с.

Ранее клавиатура имела свое гнездо для подключения к материнской плате, а мышь подключалась к COM-порту. В последствии клавиатуру и мышь решили подключать к одинаковому разъема. Так в 1998г. родился порт PS / 2. Новые конструкции системных плат для подключения клавиатуры и мыши поддерживают встроенный порт USB (Universal Serial Bus).

**ЛИТЕРАТУРА: [4], [6], [10], [15], [27]**

## **ЗАНЯТИЯ 17-18. ПАМЯТЬ ПК: ВНЕШНЯЯ И ВНУТРЕННЯЯ. МИКРОПРОЦЕССОРЫ.**

### *План*

- 1. Внутренняя память ПК*
- 2. Внешняя память ПК*
- 3. Микропроцессор*

Внутренняя память ПК состоит из оперативно запоминающего устройства (ОЗУ, RAM-память, оперативная память) и постоянного запоминающего устройства (ROM BIOS).

***Оперативная память (ОП)*** - это специальные микросхемы, которые состоят из ячеек памяти, предназначенных для временного хранения и текущего изменения информации при работе ПК.

***Постоянная память*** - это энерго независимая память, в которую информация заносится при ее изготовлении. В постоянную память "прошиты" некоторые программы и данные, которые компьютер не может изменить. Эта память предназначена только для считывания информации. Как правило, в постоянной памяти сохраняются программы обслуживания устройств компьютера и инициализации загрузки операционной системы.

ОП используется для хранения данных и программного кода, выполняемого микропроцессором. Любая информация записывается в электронные ячейки памяти в виде двоичных чисел. Расположение информации в памяти называется записью, а получение информации из памяти - считыванием. Во время записи предыдущие данные, которые хранились в ячейках памяти, стираются. В физическую ячейку памяти записывается 1 байт информации. Эта емкость ячейки достаточна, чтобы в нее записать один символ. Каждая ячейка имеет свой адрес. Когда компьютер

отправляет данные в ОП, он запоминает адреса, затем по известному адресу выбирает данные из памяти.

Важнейшими характеристиками ОП является ее *разрядность, емкость и быстродействие*. Еще 10 лет назад ПК с операционной системой Windows 95 работали с 8 Мб ОП. Семь лет назад для ПК полностью хватало 64 -128 Мб ОП. Для работы современных операционных систем и мультимедийных приложений требуется не менее 512 Мб оперативной памяти. Современные ПК имеют и 4 Гб оперативной памяти. Быстродействие ОП зависит от скорости считывания и записи в ячейки памяти.

Кроме объема ОП, актуальным является выбор типа памяти. по принципу работы (принципу хранения информации) RAM можно разделить на динамическую и статическую. Разница между динамической и статической памятью заключается в конструктивных особенностях элементарных ячеек для сохранения отдельных битов. Сейчас для ОП используется динамическая память DRAM. Она построена на микросхемах, требующих для сохранения информации ее периодического обновления (регенерации), то есть на конденсаторах. По своей логической организацией DRAM может быть асинхронной и синхронной. Чтобы обеспечить высокую скорость работы памяти сейчас используется синхронная динамическая память DDR SDRAM. SDRAM означает, что память является синхронной динамической, то есть при работе с памятью SDRAM обеспечивается синхронизация всех входных и выходных сигналов с тактами системного генератора.

Аббревиатура DDR (Double Data Rate) означает двойную скорость передачи данных (до 4 Гб / с и более). В четыре раза большую скорость передачи данных имеет стандарт DDR2. В 2009 основную долю рынка ОП завоевал стандарт DDR3 (логическое продолжение стандарта DDR2). Первые модули памяти DDR3 имеют емкость 1 Гб, последующие - 2 и даже 4 Гб.

В зависимости от форм-фактора различают SIMM-модули и DIMM-модули памяти (появился в 1998г.). В современных ПК используются 184-

контактные DIMM DDR-модули памяти в виде отдельных маленьких плат с напаянными на них микросхемами. В отличие от модулей SIMM, двухрядные модули памяти - модули DIMM имеют электрически независимые контакты с обеих сторон разъема.

Существуют различные стандарты на модули DIMM: DIMM-512 Мб, DIMM-1 Гб, DIMM-2 Гб и др. На материнскую плату в соответствующие гнезда можно вставить 1, 2, 3 и даже 4 микросхемы памяти типа DIMM. Различные структуры ОП отличаются скоростью доступа к памяти и их пропускной способностью.

Статическая память используется в качестве вспомогательной памяти - *кэш-памяти*, которая предназначена для оптимизации работы процессора. Оперативная память работает более медленнее, чем процессор, поэтому он оснащается запоминающим устройством небольшого объема (кэш-памятью) для промежуточного хранения данных.

## **ВНЕШНЯЯ ПАМЯТЬ КОМПЬЮТЕРА**

- 1. Общие характеристики внешней памяти.*
- 2. Винчестеры (HDD).*
- 3. Лазерные диски.*
- 4. Flash-память.*

### **1. Общие характеристики внешней памяти**

Для долговременного хранения информации (программ и данных) на ПК используются различные устройства, принадлежащие к внешней памяти. Внешняя память является долговременной. Если в оперативной памяти данные сохраняются только во время работы программы, то во внешней памяти информация может храниться годами. Поэтому эти устройства называются накопителями. Устройства внешней памяти различают прежде всего по типу носителя информации, а именно:

- жесткие магнитные диски;
- оптические компакт-диски;
- Flash-память.

В системном блоке ПК есть специальные монтажные отсеки. Это позволяет компактно размещать накопители в системном блоке. Все устройства внешней памяти, содержащие диски (накопители), являются устройствами с прямым доступом, то есть время доступа к информации не зависит от расположения информации.

Магнитные диски имеют такое название благодаря наличию тонкого магнитного слоя на своей поверхности. Запись информации на магнитные диски происходит по концентрическим дорожкам. Все концентрические дорожки разбиваются на секторы. **Сектор** - это наименьший участок поверхности диска, на который можно записать данные. В одном секторе, как правило, 512 б, но может быть и 1024 б.

Обмен данных осуществляется целым числом секторов. **Кластер** – это минимальная единица размещения информации, состоящий из нескольких смежных секторов дорожки. Разметка магнитного диска на дорожки и сектора называется **форматированием диска**. Вследствие форматирования дорожкам присваиваются номера. Форматирование осуществляют специальные служебные программы.

При записи / чтении магнитные диски вращаются вокруг своей оси с постоянной скоростью, а магнитная головка подводится к нужной дорожке. Данные на диске хранятся в файлах. *Файлы, хранящиеся в отдельных кластерах, разбросанных на диске, называются фрагментированными*. Для отыскания файлов на первой дорожке находится таблица размещения файлов (FAT).

## 2. Винчестеры (HDD)

Винчестеры служат в современных компьютерах основными средствами массовой памяти. Их параметры постоянно улучшаются. Базой для улучшения является развитие технологии магнитной записи, которая обеспечивает постоянный рост емкости диска при снижении его стоимости.

Одна офисная программа занимает на дисках сотни мегабайт, одна игра может занять несколько Гб, поэтому сейчас дисковое пространство должен занимать Гб и Тб. Чем больше емкость диска, тем меньше относительная стоимость, то есть стоимость одного мегабайта. Современные винчестеры являются надежными в работе (большой срок службы - 5-7 лет) и характеризуются хорошими статистическими показателями (средняя наработка на отказ 500 тыс. - 1 млн часов). Каждый винчестер содержит совокупность дисковых пластин (до четырех), которые размещены на одной оси и покрыты с двух сторон магнитным материалом, на который записываются данные. Данные записываются НЕ как угодно, а в соответствии с физической структурой диска. Магнитная поверхность каждого диска разделена на магнитные дорожки, которые в свою очередь делятся на секторы. Но поскольку дисков в винчестере есть несколько и имеют они по две рабочих поверхности, то дисковое пространство делится еще и на цилиндры. *Цилиндр* - это сумма совпадающих друг с другом дорожек по вертикали и по всем поверхностям. Диски винчестера закреплены на одной оси, которую вращает двигатель. Скорость вращения дисков очень высока. Чем выше скорость вращения, тем больше скорость чтения / записи информации.

На жестком диске ПК размещается операционная система, которая загружается в память сразу после включения компьютера. Каждый жесткий диск для удобства разбивается на несколько разделов. *Образующиеся разделы называются логическими дисками*. Им даются имена: буквы С :, D :, E .... Логический диск с буквой С: является системным.

Числовые характеристики винчестера:

1. *Емкость диска.* Диапазон отформатированных емкостей современных жестких дисков составляет более 250 Гб. Первый жесткий диск (1956 гг.) имел емкость 5 Мб.

2. *Скорость чтения данных.* Современный показатель - это 150, 300 Мб / с.

3. *Скорость вращения диска.* Нынешний стандарт - 7200 об / мин.

4. *Размер кэш-памяти.* **Кэш-память** - быстрая память небольшого объема, в которую компьютер помещает наиболее часто используемые данные, вероятно пригодятся процессору. Размер кэш-памяти в современных моделях винчестеров достигает 16 Мб и более.

5. *Тип интерфейса.* На жестких дисках большинства дисководов имеется несколько разъемов для подключения к системе подачи электропитания и интерфейсного кабеля. Ранее жесткие диски подключались к разъему E-IDE на материнской плате. Сюда подключаются FDD и CD-ROM. Стандарт E-IDE позволяет подключить до 4 дисков.

Контроллер IDE (эл. схема) находится на основной материнской плате. В 2002 появились новые жесткие диски с интерфейсом SATA, пропускная способность которых составляла 150 Мб / с и практически все производители жестких дисков начали их серийное производство. В 2004 максимальная скорость передачи данных по этому интерфейсу увеличилась в два раза и составила 300 Мб / с. Интерфейс SATA несмотря на увеличенную скорость передачи данных имеет и ряд других преимуществ: отсутствие плоских и широких шлейфов, повышенная надежность передачи данных, простота подключения и др. Внедрение этой технологии ставит винчестеры SATA в один ряд с SCSI-дисками. Для интерфейса SCSI нужно приобрести вместе с винчестером контроллер SCSI и установить его в свободный слот на материнской плате. Винчестеры SCSI дороже и ориентированы в основном на применения в профессиональных системах.

### 3. Лазерные диски

Лазерные (оптические) диски выпускаются двух типов: CD-диски, DVD-диски. CD ROM (ROM - Read Only Memory) - устройство для чтения данных с компакт дисков на которых большие объемы информации (эпоха Windows связана с большими объемами информации). Этот дисковод появился на компьютерах в начале 90-х годов.

Компактные (оптические) диски используют технологию лазерной записи и считывания информации (разработана российскими учеными А. Прохоровым, М. Басовым и Ж. Алферовым – лауреатами Нобелевской премии). При этой технологии лазерный луч прожигает ямки, затем при чтении с поверхности диска по-разному отражается свет. Ямка соответствует нулю, а бугорок - единицы. Дорожка, по которой сделана запись, имеет вид спирали. Эта дорожка только одна, в отличие от многих дорожек на магнитном диске.

Теперь выпускаются оптические диски диаметром 120 мм (4,7 дюйма) и 80 мм (3,1 дюйма). Классический CD мог вместить 650 Мб данных или 74 мин аудиоинформации. На такие диски можно записать более 20 тыс. картинок в сжатом формате JPEG. Если хранить только текст, то на диск CD можно разместить более 1000 книг по 300 страниц. Впоследствии появились CD на 700 Мб (80 мин аудиозвучания) и 800 Мб (90 мин).

**Скорость считывания** - это величина вынесена прямо в название устройства. Например, Creative 24x (1997 г.) - это 24-скоростной дисковод. 24 при этом означает, что он в 24 раза быстрее самых первых дисководов, скорость которых была 150 Кб / с (24 умножаем на 150 и получаем 3600 Кб / с). В 2000 появились 52-скоростные дисководы фирмы Kenwood. Этого добились путем расщепления лазерного луча на 6 лучей и смогли читать информацию сразу из 6 дорожек.

Кроме дисководов CD-ROM стали использовать дисководы CD-RW, которые могут записывать диски двух типов: CD-R (однократная запись) и

CD-RW (диски многократной записи). Запись дисков CD-RW приблизительно в два раза медленнее чем CD-R дисков.

В приводах CD-RW указывают три числа: первое - скорость записи, второе - перезаписи, третье - чтение, или меньше число – скорость перезаписи, среднее число - скорость записи, больше всего – скорость чтения, например формула 32x24x48x означает максимальную скорость записи на CD-R 32x, на CD-RW 24x, максимальную скорость чтения 48x.

В начале 1998г. на рынке стали появляться диски и накопители DVD (Digital Video Disk) - многоцелевой цифровой диск). По размеру CD и DVD одинаковые (диаметром 12 см), но DVD в два раза тоньше. На DVD диске дорожки размещены плотнее и лазерный луч с меньшей длиной волны нарезает более плотные ямки (точки). DVD диски могут быть как односторонние так и двусторонние (DS), однослойные и двухслойные (DL).

Односторонние диски DVD выпускаются в запечатанных картриджах, так и без картриджей. Двусторонние диски DVD бывают только в картриджах. Накопитель DVD-ROM, аналогично CD-ROM-у, может считывать информацию как с дисков DVD, так и с дисков CD, так что производство накопителей CD-ROM уже сворачивается.

Для самостоятельной записи существуют две разновидности DVD-дисков:

□ DVD-R, DVD + R, DVD-R DL, DVD + R DL - однократно записываемый диск (аналог CD-R);

□ DVD-RW, DVD + RW - многократно перезаписываемый диск (Аналог CD-RW).

Значение емкостей для DVD-дисков размером 120 мм такие: односторонний однослойный - 4,7 Гб; односторонний двухслойный - 8,5 Гб; двусторонний однослойный - 9,4 Гб; двусторонний двухслойный - 17 Гб.

Если говорить о скоростных характеристиках записывающих DVD-дисководов, то сейчас в большинстве новых моделей универсальных дисководов максимальная скорость записи дисков DVD-R, DVD + R составляет 16x, DVD + RW, DVD + R DL - 8x, дисков DVD-RW и DVD-R DL - 6x, где однократная скорость записи DVD-устройств уже составляет 1350 Кб / с, то есть скорость передачи информации для DVD-дисководов достигает 21 Мбайт / с.

Однако необходимо отметить, что не всегда оправданным является выбор дисководов с максимальной скоростью записи, поскольку нужно иметь сертифицированные диски на соответствующую скорость записи, а это существенно влияет на их цену. Кроме этого возникает проблема надежности записи.

Заметим, что этим двум стандартам дисков уже идут на смену диски нового поколения (Blue-ray диски, сокращенно BD) с высокой плотностью записи - 25 Гб на один слой. Базовое значение скорости 1x для BD составляет 36864 Кб / с, что в 27 раз превышает DVD. Этого эффекта удалось достичь за счет использования в технологии Blue-ray для записи и считывания синего-фиолетового лазера (длина волны 405 нм), вместо красного лазера (длина волны 650 нм), которую использует технология DVD.

### **Flash-память**

В последнее время для внешней памяти используют устройства Flash-памяти (микросхемы в пластиковом корпусе). Подключаются эти устройства к компьютеру по интерфейсу USB (2.0). устройства Flash-памяти имеют небольшие размеры, различные формы корпусов с индикатором, которые загораются при подготовке к доступу к данным.

Основные характеристики Flash-памяти такие: емкость (1, 2, 4, 8, 16, 32 Гб), скорость передачи данных (до 60 Мб / с), надежность (время хранения данных до 10 лет).

**ЛИТЕРАТУРА:** [4], [6], [10], [15], [27]

## **ЗАНЯТИЯ 19-20. ВИДЕОСИСТЕМА ПК. МОНИТОРЫ**

*План*

1. *Видеосистема ПК*
2. *Виды мониторов ПК*

Монитор является основным устройством вывода информации. Сегодня выпускаются разные мониторы (дисплеи) стандарта SVGA. качество изображения на экране монитора определяется как возможностями самого монитора, так и возможностями контроллера SVGA (видеоконтроллера).

Основные параметры монитора: размер экрана и зерна, разрешение, частота кадровой развертки (скорость обновления изображения) и др.

Существует несколько стандартных размеров диагонали экрана: 14 дюймов (36 см), 15 дюймов (39 см), 17 дюймов (44 см), 19 дюймов (49 см), 21 дюйм (54 см) и т.д. Сегодня в основном используются 17-дюймовые мониторы. Большие экраны используются для профессиональной работы.

Еще один фактор, который определяет качество изображения (и соответственно, цену монитора), является размер зерна (0.22, 0.26, 0.28, 0.29 мм). чем меньше зерно, тем лучше изображение. **Зерно** - это минимальная точка (пиксель), которая измеряется в десятых долях миллиметра. Как правило, для 15-дюймовых мониторов размер зерна составляет от 0,28 мм до 0,25 мм. величина зерна на 17-дюймовых мониторах колеблется в диапазоне 0,24 - 0,27 мм.

**Частота вертикальной развертки** - это частота обновления кадров измеряется в Гц. Один герц соответствует одному кадру в секунду. Для

комфортной работы необходимо, чтобы частота вертикальной развертки составляла не менее 85 Гц. Меньшая частота вредна для глаз - мигание быстро утомляет глаза. При частоте вертикальной развертки, что превышает 110 Гц глаз человека уже не замечает никакого мигания.

*Горизонтальная частота развертки показывает, какое количество линий может быть выведено на экран за 1 секунду. Для современных мониторов она составляет от 15 кГц до 100 кГц.*

Параметры мониторов связаны между собой, например, при уменьшении разрешения растет частота развертки и число цветов.

*Разрешение.* Эта величина характеризует качество воспроизведения изображение на мониторе, то есть показывает сколько пикселей может уместиться на вашем экране. Разрешение описывают две величины: количество точек по горизонтали и по вертикали.

Стандартные режимы:

640x480 - для 14 дюймовых мониторов;

800x600 - для 15-дюймовых;

1024x768 - для 17-дюймовых и т.д.

На практике каждый монитор может поддерживать и более высокое разрешение. Для мониторов с электронно-лучевой трубкой разрешение можно менять достаточно гибко.

*Разновидности мониторов.* Существуют два класса мониторов: светоизлучающие (мониторы с электронно-лучевой трубкой (ЭЛТ-мониторы)) и светопропускающие (мониторы на жидких кристаллах).

ЭЛТ-мониторы получают изображение от пучка электронов, попадает на поверхность монитора, который покрыт люминофором. Люминофор - это зернистое вещество, которое испускает свет при бомбардировке его заряженными частицами. Пучок электронов выпускается электронной пушкой и управляется отклоняющей системой через электромагнитное поле так, что

электроны попадают в нужное место на экране. Модулятор регулирует интенсивность этого пучка и обусловленную этим яркость изображения.

Для создания цветного изображения используются три пушки ("красная", "зеленая", "синяя") и на поверхность монитора наносятся три вида люминофора. Когда пучок электронов достигает слоя люминофора он вызывает свечение трех отдельных точек, расположенных настолько близко, что воспринимаются глазом человека как единый смешанный цвет. Перед люминофором ставится специальная маска-решетка, которая сужает пучок и сосредоточивает его на одном из участков люминофора. Без решетки изображение было бы расплывчатым.

*Мониторы на основе дисплеев с жидкими кристаллами (LCD-мониторы).* В LCD-мониторах изображение создается с помощью матрицы пикселей, что формируется не пучком электронов, а жидкими кристаллами. Жидкокристаллическим называется такое состояние вещества при котором оно обладает промежуточными свойствами между свойствами твердого кристалла и жидкости. Жидкие кристаллы обладают оптическими свойствами, то есть под действием электронов их молекулы могут изменять свою ориентацию и вследствие чего изменять свойства светового луча (его интенсивность), проходящего через них, а это позволяет формировать нужное изображение на экране. Цвет в LCD-мониторах получается за счет поворота на определенный угол жидкокристаллических молекул для каждого субпикселя. Промежуточные состояния LCD-ячейки формируют цветной оттенок.

Жидкокристаллические дисплеи имеют фиксированный набор физических пикселей, поэтому LCD-мониторы обладают одним разрешением.

В настоящее время характеристики LCD-мониторов значительно улучшились. Есть мониторы с отличной четкостью и идеальным качеством геометрии изображений. Плюс к этому они не генерируют электромагнитное излучение (являются безопасными для здоровья человека) являются компактными и имеют красивый дизайн. Одно из основных преимуществ

LCD-мониторов - это отсутствие мигания и высокая яркость изображения. К недостаткам LCD-мониторов относится ограниченность диапазонов углов зрения (вертикального и горизонтального), то есть стоит несколько повернуть дисплей, как заметно изменятся яркость и цвета. Выпускают такие мониторы компании LG, BenQ.

Совсем недавно возникли *плазменные дисплеи* (PDP-мониторы). технология PDP базируется на световом разряде в плазме, образующейся при рекомбинации ионизированного газа. Заряженный газ называется плазмой излучает свет в ультрафиолетовом диапазоне, который попадая на люминофор заставляет его частицы светиться, но уже в видимом для человека диапазоне. Пока что PDP-мониторы используются в основном в домашних кинотеатрах и достаточно дороги.

*Действие мониторов на здоровье людей.* Электронно-лучевые трубки являются источником электромагнитного поля, которое негативно влияет на нервную систему и органы зрения человека. Поэтому монитор должен отвечать ряду стандартов безопасности. Наиболее важными являются два стандарта. Стандарт ТСО 92 внедрен в 1992 Шведской конфедерацией профсоюзов. В этом стандарте допустимые уровни электромагнитного поля регламентировались на расстоянии 30 см от экрана и 50 см от других поверхностей корпуса монитора. Опираясь на современные научные исследования утверждать о существенном вреде или о безопасности излучения не представляется возможным. Считается, что наличие значка ТСО 92 свидетельствует о полной безопасности вашего монитора. Согласно ТСО 92 частота изменения картинки должно быть не менее 85 Гц. Стандарт ТСО 98/99 заключается в основном в требованиях к материалам, из которых сделан как самый монитор, так и его упаковка.

**ЛИТЕРАТУРА:** [\[4\]](#), [\[6\]](#), [\[10\]](#), [\[15\]](#), [\[27\]](#)

## **ЗАНЯТИЯ 21-22. УСТРОЙСТВА ВВОДА**

### *План*

- 1. Клавиатура*
- 2. Мышь*
- 3. Сканер*

### **Типы устройств ввода / вывода информации**

Для ввода информации в памяти компьютера существуют различные устройства. Универсальным устройством ввода является клавиатура. К устройствам ввода относятся также манипуляторы типа мышь, джойстик. Для оптического считывания изображений и преобразования их в цифровой код применяются сканеры. В последнее время используются цифровые видеокамеры и фотоаппараты.

Основным устройством вывода информации в ПК является монитор. *Монитор служит для отображения на экране графической и символьной информации.* Для вывода информации на бумагу используют принтер, плоттеры.

### **Клавиатура**

*Современная клавиатура - это сложное устройство, позволяющее вводить данные в ПК. Кроме того, с помощью клавиатуры пользователь может управлять работой компьютера и различных приложений. Кроме панели с клавишами, она содержит электронные схемы, которые преобразуют нажатия клавиш в двоичные коды.*

Клавиши клавиатуры можно разделить на несколько групп. В центре клавиатуры находятся алфавитно-цифровые клавиши. Эти клавиши имеют двойные отметки - верхние надписи работают, когда мы используем латинские буквы, а нижние, когда набираем текст кириллицей. при нажатой

клавише Shift набираются большие буквы, то есть переходим в верхний регистр клавиатуры. Отпустив клавишу Shift снова переходим к набору строчных букв. Набор строчных соответствует нижнему регистру клавиатуры. Для перехода в верхний регистр можно использовать клавишу Caps Lock. Нажатие на эту клавишу переключает режимы нижнего и верхнего регистров.

Справа от алфавитно-цифровых клавиш размещается группа клавиш управления курсором (клавиши со стрелками, Home, End, Page Up, Page Down). Клавиша Home возвращает курсор в начало строки, а End - в конец строки. Клавиши Page Up, Page Down обеспечивают перелистывания видимых частей страниц документа.

В правой части клавиатуры расположена цифровая клавиатура. Эти клавиши используются для ввода цифр и знаков арифметических действий (в режиме Num Lock - светится индикатор Num), или для управления курсором, если режим Num отключен.

Выше алфавитно-цифрового блока находятся функциональные клавиши F1 - F12. В каждом приложении этим клавишам соответствует разное назначение. F1 всегда используется для вызова справки.

Приведем назначение специальных клавиш клавиатуры:

Enter - ввод команды;

Esc - отмена последнего действия, выход из текущего режима программы;

Del - удаление выделенных объектов или символа в тексте, находящегося справа от курсора;

Backspace - удаление символа слева от курсора;

Print Screen - копирование содержимого экрана в буфер.

Всего клавиатура содержит 101-104 клавиши. В ноутбуках обычно используется 88-клавишная клавиатура.

В середине 1990-х годов была разработана эргономичная клавиатура, учитывающей особенности анатомии человека. В ней клавиши разделены на две секции (соответственно для левой и правой руки) и размещены под определенным углом. Такая форма клавиатуры позволяет снизить нагрузку при длительной работе на ПК.

Наиболее распространенными интерфейсами для подключения клавиатур является PS / 2 и USB.

## **Мышь**

Мышь вместе с клавиатурой является неотъемлемым атрибутом ПК. Без нее невозможна работа с большинством современных приложений.

*Мышь - это устройство для позиционирования курсора и управления работой программ.*

Вместе с перемещением мыши по экрану монитора движется указатель мыши. Если указатель мыши навести на объект, то над ним можно выполнить ряд действий. Щелчок левой кнопкой мыши приводит к выделению объекта, щелчок правой кнопкой - к вызову контекстного меню этого объекта.

В комплект поставки мыши входит и драйвер мыши. Драйвер – это программа, управляющая работой устройства.

Мыши делятся на оптико-механические и оптические по виду получения информации о перемещении мыши.

По способу передачи информации в ЭВМ мыши делятся на проводные и беспроводные.

Принцип работы оптико-механической мыши является простым: утяжеленный шарик с резиновым покрытием катится по плоской поверхности и вращает 2 перпендикулярно размещенных валика, которые формируют движение в системе "горизонтально-вертикально", на конце каждого валика есть диск с малыми дырками по кругу. Это колесо крутится между источником света (светодиодом) и приемником (фототранзистором).

Информация о длине светового импульса (чередование света-темноты) преобразуется в электрические сигналы и позволяет определить скорость перемещения курсора на экране и его размещения.

Оптические мыши не имеют движущихся частей. Фотодатчики установлены на нижней поверхности корпуса мыши. В ней установлен цифровой сигнальный процессор DSP (мощность 18 MIPS - первые ПК имели значительно меньшую мощность). Этот процессор в реальном времени сравнивает картинку, которые поступают с миниатюрной видео камеры оптического датчика. По результатам сравнений определяется, в какую сторону и с какой скоростью перемещается мышь. В современных моделях оптических мышей сравнивается от 1,5 до 6 тыс. картинок в секунду). Именно такие оптические мыши сейчас доминируют на рынке.

Большинство мышей подключаются к ПК с помощью тонкого многожильного кабеля. Такой способ простой и дешевый, но лишний провод на столе нежелателен, поэтому были созданы беспроводные мыши.

При перелистывании документов необходимо постоянно переключать внимание с документа на полосу прокрутки (что является неудобным). Можно для этого воспользоваться клавишами PgUp, PgDown. Но при просмотре сайтов Internet руку с мыши убирать неудобно, поскольку большинство операций с гипертекстом выполняется именно мышью. Поэтому данные функции стали выполнять мыши с скроллингом (Microsoft Intelli Mouse). Между двумя кнопками мыши есть колесико, с помощью которого можно прокручивать по вертикали содержимое активного окна. Такая конструкция мыши является сейчас фактическим стандартом.

Но постоянное прокручивание мыши приводило к утомлению пальцев, поэтому вместо колесика стала использоваться кнопка, с помощью которой можно мышь перевести в режим Auto Panning, в котором перемещение мыши приводит не к перемещению курсора, а к прокручиванию содержимого окна - это мыши Easy Mouse.

Наиболее распространенными интерфейсами, используемыми сейчас для подключения мышей является порт мыши PS / 2 и разъем шины USB. Подключение с помощью стандартного последовательного COM-порта и с помощью специального адаптера в виде платы расширения сейчас практически не используются.

## **Сканеры**

*Сканеры - это устройства для ввода в компьютер черно-белых или цветных изображений непосредственно с бумажного документа.*

*Сканирование - это перевод бумажных документов в цифровую форму по точкам. Сканирование - это процесс, в результате которого создается электронный образ бумажного документа.*

В результате сканирования документа создается графический файл, в котором хранится растровое изображение исходного документа, но этот набор точек еще не является документом в электронной форме. Это файл графического формата (например, .bmp, .tiff, .jpeg). Если оригинал содержал текст, отсканированный файл не может быть прочитан текстовым редактором. Нужно еще распознать текст отсканированных документов – это осуществляют программы распознавания текста. Примером программы, которая распознает текст, является FineReader.

Принцип действия сканеров основан на освещении бумажного документа. Затем измеряется отраженный свет в цифровой форме. Наиболее распространенными сейчас являются планшетные сканеры. В планшетных сканерах бумагу кладут на специальную поверхность и далее осуществляется сканирование документа и его ввод в компьютер. Кроме планшетных еще бывают барабанные, рулонные, ручные, проекционные и др. сканеры. Сканеры бывают *черно-белые* (для ввода текста и рисунков, выполненных контуром), *полутоновые* (цвета заменяются различными оттенками серого цвета) и *цветные*. В настоящее время практически все сканеры цветные.

Основная техническая характеристика сканера - это *разрешение сканера* (максимальное количество точек, которое способен различить сканер), которая имеет два показателя: по горизонтали (определяется количеством элементов на линейке фотодетекторов) и по вертикали (определяется шагом движения линейки). Например, 600x300, 600x800, однако часто используют только первое значение. Разрешение сканера измеряют еще *количеством точек на дюйм - dpi*. Сканера нужна такая разрешающая способность:

- в случае текста - для дальнейшего распознавания в программе *FineReader* - 300 dpi в монохромном режиме;

- простой цветной печати - 300 dpi;

- фотопечать - 600 dpi;

- сохранение изображений и просмотр их только на компьютере - 200 dpi.

За точность передачи цветов отвечает второй показатель - *разрядность сканера (глубина цвета)*, которая измеряется в битах. К примеру, разрядность 8 бит соответствует тому, что сканер может распознать  $256 = 2^8$  цветов, или градаций серого, 10 бит - уже  $1024 = 2^{10}$  градаций, 24 бита соответствуют 16700000 цветов (понятно, что в быту такое количество цветов никогда не понадобится - хотя производители сканеров уже заявили о 48-битную разрядность домашних сканеров).

Для качества работы сканера важно, какой тип матрицы использует сканер. Матрица CIS значительно хуже различает цвета и оттенки. В CIS сканеров является невысокое разрешение - до 600 dpi. Хотя они дешевле. Значительно лучше обстоят сканеры с матрицей CCD от фирмы Hewlett-Packard. Профессиональные сканеры по технологии CCD имеют разрешение 1200x2400 dpi.

Способы подключения сканеров к компьютеру:

- сканеры с параллельным или последовательным интерфейсом присоединяются к LPT или COM портам;

- сканеры с интерфейсом USB (работают значительно быстрее) подключаются к USB-порту;

- сканеры с собственной интерфейсной платой. Плата добавляется к сканеру и вставляется в свободный слот на материнской плате (Гнездо PCI или ISA). Эти сканеры относятся к более высокому классу. Они дороже и быстрее работают чем сканеры первых двух категорий.

Фирма номер один на рынке сканеров, как и на рынке принтеров, Hewlett-Packard. Другие фирмы-производители сканеров: Canon, Epson, Mustek.

**ЛИТЕРАТУРА:** [4], [6], [10], [15], [27]

## **ЗАНЯТИЯ 23-24. УСТРОЙСТВА ВЫВОДА ИНФОРМАЦИИ.**

### **ПРИНТЕРЫ**

*План*

- 1. Матричные принтеры*
- 2. Струйные принтеры*
- 3. Лазерные принтеры.*

**Принтер** - это устройство для печати на бумаге различного формата текста, графики, изображений, чертежей. Сейчас наиболее более распространенными являются лазерные, струйные, матричные принтеры.

В конкурентной борьбе явными аутсайдерами являются матричные принтеры. В них скорость и качество печати низкая (бледные, нечеткие буквы). Вторым недостатком матричных принтеров - относительно высокий уровень шума. Для широкого пользования конкуренция идет между лазерными и струйными принтерами.

Одним из преимуществ лазерной печати является высокое качество. Отпечатки НЕ размазываются и не повреждаются при контакте с водой. При печати не коробится лист бумаги (как это бывает при струйной печати). Монохромные лазерные принтеры в более низкой цене производят и фирмы Panasonic, Epson, но Hewlett-Packard является лидером в этой группе. Хотя если сравнить качество печати, то она одинакова, потому что все фирмы, которые выпускают принтеры, используют печатные узлы других фирм, например, доля Canon на рынке печатающих узлов составляет 70%.

Струйные принтеры, хотя и уступают лазерным при черно-белой печати, но позволяют выводить цветные изображения, однако стоимость печати одной страницы на струйном принтере выше чем на лазерном. Поэтому при выборе принтера необходимо исходить прежде всего из сферы применения.

Рассмотрим подробнее принципы действия различных принтеров.

**Матричные принтеры** по качеству печати явно уступают лазерным и струйным. Механизм печати базируется на способе удара. В разных моделях существует 9 или 24 ударных иглы. Почти все матричные принтеры монохромные. Преимуществом матричных принтеров является прочность и надежность принтера, возможность печати на бумаге через копирку (до 6 копий). Также дешевые краска и лента. Цветное изображение на матричных принтерах получается с помощью разноцветных лент. Используется четырехцветная лента, на которую нанесено три основных цвета: голубой, пурпурный, желтый и черный цвет. Разрешение 180 - 300 точек / дюйм.

**Лазерные принтеры.** Первая настольная модель монохромного лазерного принтера, который предназначался для подключения к ПК была

выпущена в 1984 г. Процесс лазерной печати разработан фирмой Херох. На специальном фото чувствительном барабане лучом света создаются области заряженные электронами (картинка рисуется лучом по барабану). Поверхность барабана, обработанная лазером, проходит вдоль картриджа и заряженными областями притягивает порошок тонер, состоящий из частиц красящего пигмента покрытых пластмассой. Затем барабан вращается над листом бумаги, заряженный сильнее барабан, при этом частицы тонера переносятся с барабана на бумагу и разогреваются, образуя водостойкое изображение. В лазерных принтерах используется бумага в виде листов. При лазерной печати область, которая покрывается несколькими точками, превращается в одну большую виртуальную точку. Она может выглядеть светлее, темнее в зависимости от количества реальных точек, формирующих изображение. Это и создает эффект градации серого цвета.

*Одной из основных характеристик принтера является его разрешающая способность. Она измеряется в количестве точек на один дюйм (dpi). Чем выше разрешение принтера, тем больше реальных точек может быть в одной виртуальной точке, а это означает более высокое качество печатного изображения. Разрешение современных монохромных моделей колеблется от 600 до 1200 dpi. Разница между напечатанным текстом 600 dpi и 1200 dpi незначительна, но становится заметной на графических изображениях. Для цветных моделей разрешение составляет 1200 dpi.*

*Второй важной характеристикой принтера является максимальная скорость печати, измеряемая количеством напечатанных страниц формата А4 за одну минуту. Скорость печати монохромных лазерных принтеров составляет от 4 до 40 и более страниц в минуту. Это самая высокая скорость среди других типов принтеров. На цветных принтерах можно печатать до 30 страниц в минуту.*

*Другими характеристиками принтера является время выхода первой страницы, максимальный формат листа бумаги, ресурс фотобарабана,*

*тонер-картриджей, объем памяти принтера (производители оснащают современные лазерные принтеры памятью от 4 до 8 Мб).*

**Современные струйные принтеры** выводят текст и цветную графику и стоят значительно дешевле, чем лазерные. Кроме того, они более компактны, используют меньше энергии, но имеют меньшую скорость печати и высокую стоимость расходных материалов (чернил, картридж). Подобно лазерной печати струйная печать является безударной. Принцип струйной технологии базируется на выстреливании на носитель изображения микрокапли чернил из специального сопла (их называют дюзами). Печатающая головка, содержащая чернила, имеет группу микросопел, каждое из которых в диаметре меньше диаметра человеческого волоса. Цветные устройства струйной печати имеют, как правило, четыре форсунки: три - для основных цветов (голубого, пурпурного, желтого) и одну – для черного. Эта модель цвета называется СМΥК.

Основное преимущество струйной технологии заключается в возможности смешивать цвета, поскольку жидкие части краски наносятся за один проход и они успевают перемешиваться до высыхания краски. Это позволяет получать глубину и резкость цвета такие, которых нельзя достичь при другой технологии.

Важную роль для качества печати играет качество бумаги. Технология струйной печати такова, что лучшего результата можно добиться при использовании специальной бумаги и быстро высыхающих чернил QuickDryLink, которые на соответствующем бумаге обеспечивают устойчивость к свету порядка 20 лет.

В конце 1990 в сегменте струйных принтеров сформировался подкласс *фотопринтеров*. В отличие от струйных принтеров в фото принтерах стали использовать шестицветную модель цвета СсМmΥК, где с и m означают дополнительные светло-голубую и светло -пурпурную краски. На протяжении

многих лет шестицветная схема оставалась стандартом для струйных принтеров непрофессионального класса.

В конце 2003. Компания Hewlet-Packard представила новую эволюцию технологии фотореалистичной печати - HP PhotoRet Pro, в которой уже использовались не шесть, а восемь цветов: к классическому набору были добавлены два оттенка серого (серый и светло-серый). Это дало возможность увеличить количество воспроизводимых цветов к 79,2 млн оттенков. В 2005 компания HP выпустила фотопри́нтер HP Photosmart 8753, который уже использовал девяти цветовую схему (добавили еще синий цвет). Параллельно компании EPSON и Canon выпускают свои модели фотопри́нтеров, в которых реализованы другие технологии фотопечати.

Стоит отметить, что существуют модели фотопри́нтеров, которые могут печатать изображения со сменных носителей и без компьютера. Они обладают многими функциональными возможностями и даже небольшим цветным дисплеем.

В последние годы значительно выросла популярность многофункциональных устройств, объединяющих в себе функции сканера, принтера и копира.

Для получения твердых копий чертежей, широко форматных графических изображений используют перьевые *плоттеры*.

**ЛИТЕРАТУРА:** [4], [6], [10], [15], [27]

## **ЗАНЯТИЯ 25-26. КОНТРОЛЛЕРЫ, АДАПТЕРЫ**

*План*

1. *Контроллер*
2. *Видеоадаптер*
3. *Звуковой адаптер*

**Контроллер** (или *адаптер*, в переводе с английского языка означает *вспомогательное устройство*) - это специальная электронная схема, управляющая работой периферийного устройства (дискетодом, винчестером, монитором и т.д.) и обеспечивает связь этого устройства с материнской платой. Обмен информацией между ОП, микропроцессором и внешним устройством происходит не напрямую, а через специальную схему - *контроллер*.

Отметим, что на всех современных материнских платах уже присутствуют контроллеры клавиатуры, мыши, накопителей, винчестеров (с интерфейсом IDE). К платам, расширяющим возможности ПК относятся плата модема, видеокарта, звуковая карта, сетевая карта и др.

Качество изображения на экране монитора зависит от двух составляющих - от самого монитора и от графического адаптера.

### **ВИДЕОАДАПТЕРЫ (ГРАФИЧЕСКИЕ УСКОРИТЕЛИ)**

**Видеоадаптер** - это обычно отдельная плата, которая вставляется в соответствующий слот на материнской плате и формирует видеосигнал для создания изображений на экране монитора. Команды, которые формируют изображения поступают от микропроцессора к видеоадаптеру, где согласно им конструируется изображения.

Современные видеокарты - это самая сложная составляющая ПК (компьютер в компьютере). На самой видеокарте есть специализированный графический процессор (GPU), который формирует изображение, выводимое на экран и своя оперативная память.

**Экран дисплея** - это прямоугольная матрица отдельных точек (пикселей), которые определяют изображение. Число пикселей по

горизонтали и по вертикале экрана определяет разрешающую способность дисплея, например, 640x480, 1280x1024. Первое число показывает количество пикселей в строке, второе - количество строк.

*Каждому пикселю изображения ставится в соответствие фиксированное число битов (атрибут пикселя) в памяти видеоадаптера. Эта память называется видеопамятью. Для видеопамати используется графическая память GDDR2 GDDR3, микросхемы которой распаиваются на плате графической карты. Ее стандартный объем сейчас составляет от 128 Мб до 2 Гб. Видеоадаптер циклически (75 - 100 раз в секунду) считывает содержимое видео памяти и постоянно формирует изображение на экране монитора, причем цвет пикселя определяется текущим значением его атрибута.*

Программа, выполняемая на ПК в графическом режиме, имеет доступ (чтение / запись) ко всем атрибутам видеопамати. Основой для получения качественных изображений являются графические режимы высокой разрешающей способности и высокой кадровой развертки.

Графические задачи стали настолько сложными, что видеокарты оснастили специализированными графическими процессорами (ускорителями), которые по сложности приближаются к центральному процессору.

Центральный процессор дает видеоадаптеру только общие команды, например, начертить треугольник формы X в области Y экрана. Дальнейшие вычисления с точностью до пикселя берет на себя видеокарта, освобождая от рутинной работы центральный процессор. Видеокарта выполняет эти операции аппаратно, что позволяет ускорить формирование изображения на экране. Так появился термин графический акселератор (ускоритель).

А формирование объемного изображения - значительно сложнее задача. Для создания на экране 3D-картины процессору и графическому ускорителю нужно сначала выделить видимые грани объекта. Следующим шагом будет наложение текстуры на каждую грань. Далее необходимо учесть откуда и

какой падает свет, свойства поверхности объектов (прозрачность, зеркальность). А теперь представьте себе десятки объектов в картинке, которые вращаются, удаляются, приближаются, перекрывают друг друга, попадают под различные источники света, отбрасывают тени и т.д. В результате получается сложная задача, с которой центральный процессор без 3D- ускорителя не смог бы справиться.

Неоснащенных 3D-ускорителями видеокарт сейчас не выпускают, но, с другой стороны, возможности 3D-карт используются в полную силу только в играх. Современную видеокарту можно назвать компьютером в компьютере, поскольку у нее есть свои процессор, память, внутренняя шина передачи данных.

Кроме этого, видеокарта призвана решать задачи мультимедиа. Многие карты сегодня поддерживают вывод изображения на телеэкран и наоборот, осуществляют прием изображений с видеокамеры, телеантенны. Эти операции выполняют видеовход и TV-тюнер. Также современная видеокарта декодирует сжатый видеосигнал, поступающий с DVD-дисков. Вот сколько задач лежит на маленьком *чипсете* - главной микросхеме любой видеокарты.

Потоки графических данных стали настолько интенсивными, что для увеличения быстродействия обмена данными между видеоадаптером и оперативной памятью была разработана отдельная графическая шина AGP (ускоренный графический порт). В настоящее время видеокарты подключаются к материнской плате через локальную шину AGP - ее разрядность 64 бита.

Первые видеокарты были оснащены интерфейсом AGP1x (скорость передачи данных - 256 Мб / с). В конце 1998г. На рынке появились видеокарты AGP2x - со скоростью передачи данных 512 Мб / с, в 1999г. появился режим AGP4x (скорость передачи данных - 1,06 Гб / с). В 2002 появились карты со значком AGP8x (пропускная способность 2,1 Гб / с).

На смену AGP пришел интерфейс PCI Express, который уже передал эстафету в два раза более быстрому PCI Express 2.0, который в свою очередь уже скоро будет заменен на более быстрый PCI Express 3.0.

Современные видеокарты могут поддерживать 24-битный цветной режим (3 байта на 1 пиксель). Этот режим называется True Color и может одновременно отображать на экране 16777216 цветов. Эти видеоадаптеры поддерживают и другие режимы, которые отличаются друг от друга разрешающей способностью и количеством цветов. Способность видеоадаптера отображать большое количество цветов с высокой разрешением может быть обеспечена только соответствующим объемом видеопамяти.

Среди популярных видеоадаптеров можно назвать видеокарты, которые выходят под брендом GeForce. Их выпускает компания NVIDIA. Эти видеокарты именуется по названию используемого графического процессора, например GeForce 8800 GTX, GeForce 9400 GT и др. Объединенное предприятие AMD / ATI выпускает видеокарты на базе процессоров Radeon. новые модели - это карты семейства Radeon HD 3000.

## **ЗВУКОВАЯ КАРТА**

ПК долго обходились без средств воспроизведения звука. С развитием вычислительной техники в середине 80-х годов появилась возможность создавать, хранить и воспроизводить комплексные документы, содержащие текст, звук, язык, графику и видео. Такие документы стали называть **мультимедийными**, а программные и аппаратные средства для работы с такими документами называют средствами **мультимедиа**.

Мультимедийным называют компьютер, который оснащен современными носителями данных (дисковыми CD или DVD) и звуковой картой (SoundBlaster). Звуковая плата в комплексе с аудиоколонками и микрофоном позволяет записывать и воспроизводить на компьютере различные звуки, речь и музыку. Она выполняет преобразование звука из аналоговой формы в цифровую (на входе) и обратное преобразование цифрового звука в аналоговый сигнал (на выходе). Звуковая карта вставляется на материнскую плату в свободный PCI-слот и имеет выход на заднюю панель компьютера. Среди этих выходов является гнезда для подключения колонок, микрофона, MIDI-клавиатуры (копия фортепиано) и др.

Заметим, что почти на половине материнских плат устанавливаются вместо звуковых плат звуковые чипы, то есть звуковые карты интегрируются на материнскую плату. Сейчас часто для высококачественной звуковой системы используются внешние модели звуковых адаптеров, которые подключаются к ПК через интерфейс USB.

Звуковые карты осуществляют декодирование сжатой музыки в формате MP3. MP3 - это формат хранения данных. В формате MP3 на диск CD можно записать 10 - 12 ч звука (100 - 200 записей), на аудио CD – от 10 до 15 записей. Поэтому на звуковые платы поставлены специальные чипы, которые осуществляют декодирование сжатого звука. Средства мультимедиа используются в системах распознавания речи.

## **СЕТЕВАЯ КАРТА. МОДЕМ.**

Когда ПК используется как средство передачи данных по сети, то необходимо установить определенные устройства. При подключении компьютера к локальной сети необходимо сетевая плата. Она устанавливается в PCI-слот на материнской плате. Наиболее распространенными являются комбинированные платы Ethernet, рассчитанные на разные типы кабелей.

Для передачи данных по телефонной линии необходимо устройство, которое может принять аналоговый сигнал из телефонной линии и превратить его в цифровую информацию и наоборот, то есть это устройство осуществляет модуляцию и демодуляцию сигналов (отсюда и название модем).

Модем выполняется в виде отдельного внешнего устройства, который одним выходом подключается к телефонной линии, а второй – к последовательному порту компьютера, или в виде отдельной платы, вставляется в системный блок.

Основной параметр сетевых карт - это скорость передачи данных. Хотя они способны теоретически работать на скорости 57600 бит / с, реальная скорость передачи данных значительно меньше - от 12 до 15 Мб / с. Для работы в сети Internet необходима скорость передачи данных - 28800 бит / с.

## **МИКРОПРОЦЕССОРЫ**

**Микропроцессор (далее МП)** - это специальная сверхбольшая интегральная схема, которая устанавливается на материнской плате. К материнской плате микропроцессор подсоединяется с помощью специальных разъемных соединений (Socket 7, Socket A, Slot 1, Socket 423, Socket 478 и т.д.). Современные микропроцессоры - это одна микросхема, которая изготовлена из полупроводникового кристалла кремния с плотной упаковкой физических элементов, благодаря чему на кристалле площадью около 1 см<sup>2</sup> можно разместить большое количество элементов: транзисторов, конденсаторов и т.д.

Так схемы современных процессоров Pentium содержат более сотни миллионов транзисторов. Последняя модель Itanium 2 содержит 410 млн. транзисторов. Но, поскольку электронным устройствам свойственно нагреваться во время работы, то над корпусом МП размещают небольшой вентилятор и радиатор, которые обеспечивают охлаждение МП в процессе работы.

**Микропроцессор** - это устройство, выполняет две основные функции:

1. *Вычисления согласно программе, которая хранится в оперативной памяти.*

2. *Обеспечивает общее управление аппаратурой компьютера и вычислительными процессами.* При этом МП выполняет:

- чтение и дешифрацию команд из ОП;

- чтение данных из ОП и данных из регистров внешних устройств;

- обработку данных и запись их в ОП.

Для того, чтобы МП знал, что делать, он непрерывно должен получать поток команд. Эти команды составляют программы. Благодаря программе вычисления в ЭВМ происходят автоматически. В программе сложный вычислительный процесс разбивается на множество элементарных команд, которые может выполнять МП. Число команд современного МП - 220. У каждой команды есть свой код.

МП отличаются тремя характеристиками: тактовой частотой, разрядностью и типом (моделью).

**Тактовая частота** *определяет быстрдействие процессора. Количество команд, которые процессор может выполнить за 1 секунду зависит от тактовой частоты.* Каждая команда, которая выполняется в ЭВМ, занимает несколько тактов, поэтому время выполнения команды измеряется в тактах. Продолжительность одного такта зависит от тактовой частоты. Измеряется тактовая частота в мегагерцах (1 МГц соответствует 1

миллиону тактов в секунду). Чем больше тактовая частота, тем меньше продолжительность такта, и тем быстрее работает ПК. Например, МП Intel 8086 работал на тактовой частоте 4,7 МГц, Pentium III на 230 - 400 МГц, современные МП перешли рубеж 3 Гц (3000 МГц). Тактовая частота генерируется тактовым генератором.

Еще одной важной характеристикой процессоров является их разрядность. Процессор оперирует с двоичными числами, представленными как последовательность нулей и единиц. **Разрядность МП** - это количество разрядов двоичных чисел, обрабатываются процессором за один такт в параллель. Микропроцессоры первых ПК были 8-разрядными, а все современные модели МП уже 32- и 64-разрядные.

**Модель МП** определяется особенностью его архитектуры, маркой фирмы производителя и типом процессора.

В IBM-совместимых ПК чаще всего применяются МП фирмы Intel, а также совместимые с ними модели МП других фирм - AMD, Cytrix, IBM и др.

Приведем список моделей МП фирмы Intel в порядке возрастания их производительности:

Intel 8086 (1976 г.), Intel 8088 (1979 г.) - Первые 16-разрядные процессоры имели тактовую частоту 5 ... 10МГц;

Intel 80286 - 16-разрядный процессор с тактовой частотой до 12 МГц;

Intel 80386 (1985 г.) - Первый 32-разрядный процессор. он содержал 275000 транзисторов и имел тактовую частоту до 33 МГц;

Intel 80486 (1989 г.) Содержал 1200000 транзисторов и имел 32 линии адреса, 32 линии данных и работал на частоте до 100 МГц.

Новым этапом в производстве МП стал процессор Pentium (1993 г.).

Впоследствии появились довольно успешный процессор Pentium Pro (1995) и Pentium II, в конце февраля 1999 были анонсированы первые МП Pentium III.

В конце 1999 появилось 9 моделей МП этого типа: Pentium 500E, 550, 533, 600, 700, 733 и т. После появились Intel Pentium 750, 800, 900, 1140. Цифры после названия означают тактовую частоту в МГц.

*Современные микропроцессоры.* В ноябре 2000 компания Intel представила процессор Pentium IV. Архитектура его стала отличаться от архитектуры его предшественников, благодаря чему смогли сильно увеличить частоту процессора.

Первые МП Pentium IV имели частоту 1,4 - 1,5 ГГц и содержали 42 млн транзисторов на площади 217 мм<sup>2</sup> (в два раза больше чем Pentium III). 14 ноября 2002 Intel анонсировала МП Pentium IV 3,06 ГГц. такой тактовой частоты удалось добиться благодаря организации вычислений в несколько потоков. Тактовые частоты последних Pentium находятся в пределах 4 ГГц. В процессорах семейства Pentium используется 64-разрядная шина данных и 32-разрядная шина адреса ( $2^{32} = 4294967296$  ячеек, около 4 Гб ОП)

Ответвлением от процессоров семьи Pentium стали процессоры семейства Xeon, предназначенные для многопроцессорных серверов и процессоры Celeron - более упрощенный и удешевленный вариант процессоров Pentium.

В 2001 году. Появился процессор фирмы Intel - Itanium. Последняя модель Itanium 2 содержит 410 млн транзисторов и имеет разрядность шины данных 128 байт.

Революционным событием на рынке МП стал момент появления (середина 2006) продуктов Intel Core 2 (восьмое поколение микропроцессоров).

Core 2 - это эффективная система взаимодействия нескольких процессорных ядер, но для их эффективной работы необходимо, чтобы программные продукты были адаптированы для многопроцессорных систем. Заметим, что персональный компьютер с процессором Intel Core 2 Duo с

тактовой частотой 2,4 ГГц имеет быстродействие 19,2 Гигафлопс ( $19,2 * 10^{12}$  операций с плавающей точкой в секунду).

Параллельно с фирмой Intel фирма AMD в 1999г. выпустила МП Athlon (K7). Это были модели AMD Athlon 500, 550, 600, впоследствии 650, 700, 750, 800. Кроме K7 на рынке появилась МП Athlon MP и Athlon XP (32-разрядные МП), составившие конкуренцию Pentium IV. В последнее время фирма AMD решила маркировать свои процессоры не реальной тактовой частотой, а так называемым PR-рейтингом. PR-рейтинг новых Athlon XP, MP начинается с отметки 1500+, что соответствует частоте 1,33 ГГц и заканчивается на уровне 2800+.

В октябре 2002 AMD выпустила 2 новых МП: Athlon XP 2700+ и Athlon XP 2800+, который во многих тестах является лучшим чем Pentium IV 2,8 ГГц, хотя дешевле его в 1,5 раза.

В 2003 на рынок поступили МП фирмы AMD 8-го поколения под названием Athlon 64 FX (Hammer). Одноядерный Athlon 64 представлен моделями 2800+, ..., 3400+.

Компания AMD в 2005г. анонсировала выпуск двухъядерных процессоров Athlon 64 X2 для настольных систем и линейку серверных двухъядерных процессоров Opteron. PR-рейтинг Athlon 64 X2 находится в диапазоне от 3800+ до 6000+ (2008 г.).

Как видно, выбрать лидера среди перечисленных МП совсем не просто.

**ЛИТЕРАТУРА:** [\[4\]](#), [\[6\]](#), [\[10\]](#), [\[15\]](#), [\[27\]](#)

## **ЗАНЯТИЯ 27-28. КОМПЬЮТЕРНЫЕ СЕТИ**

*План*

1. Сетевые технологии обработки даны
2. Локальные сети
3. Топология глобальных сетей
4. Глобальные сети
5. Линии связи

## 6. Сетевые технологии обработки данных

Значение компьютерных сетей для современного общества сложно переоценить. Компьютерные сети сегодня являются не просто средством связи технических систем, но и средством коммуникации огромного числа пользователей. В общем смысле компьютерная сеть представляет собой совокупность компьютеров, физически связанных друг с другом, которые передают данные между участниками сети. Компьютер, образующий сеть, называют **хостом** (host хозяин). Хостом может являться как полноценный компьютер, так и, например, сетевой принтер.

Одним из основных понятий в теории компьютерных сетей является термин **ресурс**. Компьютерные сети предоставляют доступ к ресурсам. Ресурсы сети бывают трех видов: аппаратные, программные и информационные (рисунок.1).



Рисунок 1 – Виды ресурсов компьютерных сетей

К аппаратным ресурсам относится, например, жесткий диск, выполняющий функции файлового сервера. В качестве программных сетевых ресурсов может выступать программное обеспечение, установленное на

сервере и позволяющее ускорить процесс обработки информации другим пользователям сети. Информационный ресурс представляет собой данные, хранящиеся на удаленных компьютерах. Сегодня такими ресурсами в наиболее полном объеме располагает глобальная сеть Интернет.

Все компьютерные сети можно разделить на две категории: локальные компьютерные сети и глобальные компьютерные сети. Примером локальной компьютерной сети является сеть Ethernet, а глобальной – сеть Интернет. Такое деление сетей в первую очередь связано с использованием специализированных протоколов. В данном случае под **протоколом** понимаются специализированные стандарты, которые обеспечивают совместимость всех уровней архитектуры компьютерных сетей. Протоколы, определяющие характер взаимодействия компонентов сети, называют аппаратными, а протоколы, обеспечивающие взаимодействие программ и данных, называют программными.



Рисунок .2 – Категории компьютерных сетей

Единый комплект протоколов для всех участников используется в локальных сетях. **Локальные сети**, как правило, достаточно компактны, длина такой сети может составлять от 2 метров до 3 километров, может объединять компьютеры одного этажа, корпуса или группы рядом находящихся зданий. **Глобальные сети** имеют большие территориальные размеры и могут объединять в себе регионы, страны и континенты, а также включать отдельные локальные сети. Глобальные компьютерные сети могут использовать различные сетевые протоколы.

В сетевых технологиях принято различать участников сети. В первую очередь это связано с распределением ресурсов между участниками сети и прав доступа к этим ресурсам. У участников сети права доступа к общим сетевым ресурсам могут значительно различаться. Совокупность приемов разделения и ограничения прав участников компьютерной сети называют **политикой сети**. Управление сетевой политикой называется **администрированием** компьютерных сетей.

### **Топологии компьютерных сетей**

Существует множество вариантов соединения компьютеров в сеть. Самое простое соединение состоит из двух компьютеров, в этом случае в качестве аппаратных средств используются стандартные порты ввода-вывода, а в качестве программного обеспечения может выступать встроенные в операционную систему приложения. Но, как правило, сеть включает в себя более двух компьютеров, в этом случае установка прямого соединения невозможна, тогда компьютерная сеть строиться на основе определенной топологии. **Топология** представляет собой способ организации физических связей компьютеров и прочих сетевых компонентов. В зависимости от вида топологии выбирается необходимое оборудование для монтажа сети, просчитывается возможность расширения сети и выбирается способ ее управления, которое базируется на разделении обязанностей. Существует четыре основные сетевые топологии: шина, звезда, кольцо и ячеистая топология.

#### **Топология шина**

Топология **шина** является наиболее простой из всех топологий и довольно распространенной. В топологии шина все компьютеры подключаются к одному общему кабелю (рисунок 3). Данные в топологии шина передаются всем компьютерам, но принимает их только тот компьютер,

чей адрес совпадает с адресом получателя, который зашифрован в сигнале. В каждый момент времени передачу может вести только один компьютер, все остальные находятся в состоянии приема. Поэтому производительность сети при топологии шина очень сильно зависит от количества компьютеров. Чем больше компьютеров в сети, тем больше время ожидания, следовательно, медленнее работает вся сеть.

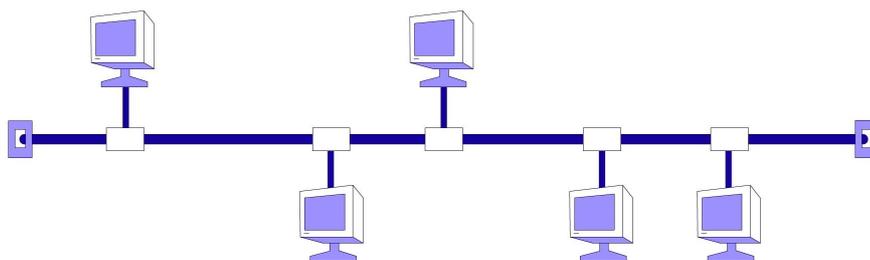


Рисунок 3 – Компьютерная сеть с топологией шина

К параметрам, влияющим на быстродействие сети с топологией шина, относятся тип аппаратного обеспечения сетевых компьютеров, частота, с которой компьютеры передают данные, тип работающих сетевых приложений, тип сетевого кабеля, расстояние между компьютерами в сети.

Электрические сигналы в топологии шина распространяются по всему кабелю. Сигналы, достигшие концов кабеля, отражаются от них, в результате возникает наложение сигналов, что приводит к искажению и ослаблению основного сигнала. Для того, чтобы сигнал не отражался от концов шины, используют специальное устройство, называемое **терминатором**.

Нарушение общего кабеля или терминаторов приводит к выходу из строя участка сети или всей сети в целом. Отключение какого-либо компьютера на работу компьютерной сети не влияет.

### **Топология звезда**

В топологии **звезда** все компьютеры с помощью сегментов кабеля

подключаются к центральному устройству, в качестве которого может выступать отдельный компьютер или концентратор, по этому признаку различают топологии с активной или пассивной звездой соответственно (рисунок 4). Сигналы от передающего компьютера поступают через центральное устройство ко всем остальным компьютерам сети.

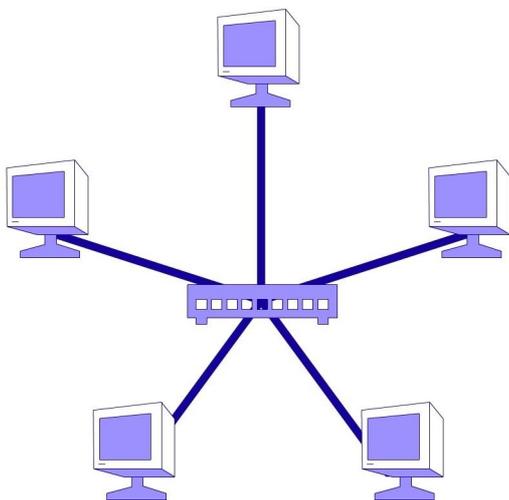


Рисунок 4 – Компьютерная сеть с топологией звезда

Топология звезда имеет четко выделенный центр, к которому подключены все остальные компьютеры сети. Обмен всей информацией происходит через центральный компьютер, который, как правило, имеет более высокие технические характеристики и сложное сетевое оборудование. Центральный компьютер занимается только обслуживанием сети. Такое устройство топологии позволяет повысить отказоустойчивость сети, полный отказ сети возможен только в случае выхода центрального компьютера, отказ одного из периферийных компьютеров на работу остальной сети не влияет.

Сегодня в мире наиболее популярна топология с пассивной звездой. Концентратор в пассивной звезде принимает сигналы от периферийных компьютеров, восстанавливает их и пересылает другим компьютерам. Схема построения пассивной и активной звезды идентичны.

Существенным недостатком топологии звезда является поддержка подключения к центральному компьютеру ограниченного числа

периферийных компьютеров, а также большой расход кабеля при ее проектировании.

### **Топология кольцо**

Топология **кольцо** предполагает последовательную передачу информации в сети от одного компьютера к другому. Данные передаются только в одну сторону. В топологии кольцо каждый компьютер соединяется линиями связи с двумя другими компьютерами (рисунок 5). Причем от одного компьютера он только получает информацию, а другому – только передает. Выделенного центрального устройства в топологии кольцо не существует, все компьютеры равноправны, но может быть выделена специальная рабочая станция, которая управляет сетью и контролирует обмен информацией.

Обмен информацией в топологии кольцо происходит с помощью **маркеров**, которые представляют собой специальную последовательность битов, передающихся по сети. В каждой сети существует только один маркер. Маркер передается по кольцу последовательно от одного компьютера к другому до тех пор, пока его не захватит тот компьютер, который хочет передать данные.

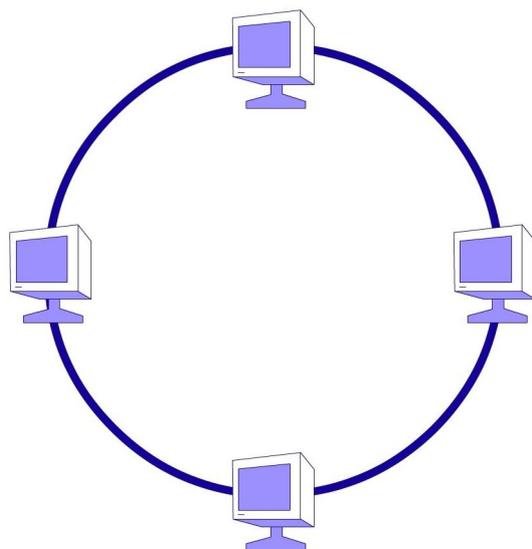


Рисунок 5 – Компьютерная сеть с топологией кольцо

В топологии кольцо каждый компьютер не только принимает сигнал и передает его дальше, но и восстанавливает его, поэтому проблем с затуханием сигнала в кольце не существует, главное, чтобы затухание сигнала не происходило между двумя соседними компьютерами. Благодаря такой ретрансляции можно строить компьютерные сети на основе топологии кольцо протяженностью до нескольких десятков километров. Топология кольцо может поддерживать подключение большого количества компьютеров, устойчива к перегрузкам и обеспечивает работу с очень большими потоками данных. При выходе из строя одного компьютера или повреждении кабеля вся сеть выходит из строя, поэтому к одному компьютеру протягивают два кабеля.

### **Ячеистая топология**

Ячеистая топология предполагает схему соединения компьютеров в сеть, при которой каждый компьютер соединен со всеми рядом стоящими компьютерами (рисунок 6). Эта топология является самой устойчивой к отказам и перегрузкам из всех рассмотренных ранее. Существенным недостатком ячеистой топологии является большие затраты на прокладку кабеля.

В ячеистой топологии сигнал от компьютера отправителя до компьютера получателя может пройти по разным маршрутам, поэтому разрыв какого-либо участка сети на работе всей остальной сети не сказывается. Такая топология используется при построении глобальной сети Интернет.

Кроме базовых топологий существуют их комбинации. Чаще всего используются две комбинированные топологии: звезда-шина и звезда-кольцо.

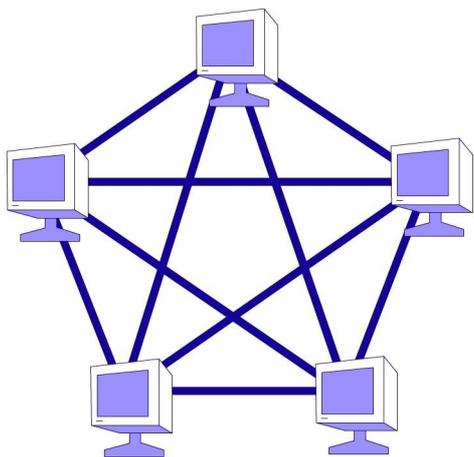


Рисунок 6 – Компьютерная сеть с ячеистой топологией

### **Линии передачи данных в компьютерных сетях**

В большинстве случаев компьютеры в сетях физически соединены между собой с помощью линий связи. В локальных компьютерных сетях информация передается последовательно, бит за битом. Параллельная передача данных также возможна, но физически реализовать ее намного сложнее. Во-первых, при параллельной передаче информации компьютеры соединяются не менее, чем восемью проводами одновременно, что очень дорого даже в случае небольших расстояний между участниками сети. Кроме того, длина прокладываемых линий связи должна быть абсолютно одинакова, что также достаточно тяжело выполнить при увеличении расстояния между компьютерами. Поэтому, несмотря на то, что последовательная передача данных по сети медленнее и сложнее, чем параллельная, используется она на много чаще.

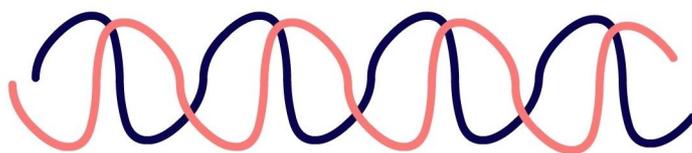
К основным характеристикам линий связи относят следующие параметры: *частотный диапазон сигналов, характеризующий затухание сигнала в линии связи, скорость распространения сигнала, помехозащищенность линии связи, волновое сопротивление кабеля, которое важно для электрических кабелей и зависит от формы и взаиморасположения электрических проводников внутри кабеля, материала*

## **Проводные линии передачи данных**

Существует большое количество различных линий связи, выпускаемых промышленно. Все их виды строго стандартизированы и имеют достаточно точные характеристики. Наибольшую популярность получили проводные линии связи из-за простоты монтажа и низкой стоимости кабеля. Выделяют три основных вида проводных линий связи: кабели на основе витой пары, коаксиальные и оптоволоконные кабели.

### **Витая пара**

Витая пара является сегодня самым дешевым видом линий связи. Кабель на основе витой пары состоит из нескольких пар скрученных изолированных медных проводов, объединенных в одной пластиковой оболочке. Структура



кабеля на основе витой пары представлена рисунке 7.

Рисунок 7 – Структура кабеля на основе витой пары

Как правило, данный вид кабеля состоит из двух или четырех пар витых проводов, является очень гибким и его легко прокладывать в помещениях. Существуют два вида кабелей на основе витой пары: экранированный и неэкранированный. Неэкранированные витые пары обладают слабой защищенностью, подвержены влиянию внешних электромагнитных полей и легко прослушиваются. Для увеличения помехозащищенности используют

экранированные кабели на основе витой пары. В этом случае каждая пара помещается в металлическую оплетку, которая является экраном и значительно снижает воздействие помех на медные провода внутри кабеля и взаимные наводки между парами. Несмотря на все недостатки, наиболее распространенным кабелем является неэкранированная витая пара, так как является более дешевым и легко монтируемым видом кабеля. На основе витой пары в основном строятся сети с топологией звезда или кольцо. Скорость передачи данных в таких линиях связи не превышает 100 Мбит/с.

### **Коаксиальный кабель**

Основу коаксиального кабеля составляет электрический кабель, состоящий из центрального медного провода и металлической оплетки, между которыми находится диэлектрик. Диэлектрик является внутренней изоляцией кабеля. Все составляющие коаксиального кабеля помещены в общую внешнюю оболочку из пластика. На рисунке 8 представлена структура коаксиального кабеля, по которому передается электрический сигнал.

По сравнению с витой парой коаксиальный кабель имеет более высокую помехоустойчивость и допустимые скорости передачи данных. В основном коаксиальный кабель применяют при монтаже сетей с топологией шина. На концах коаксиального кабеля всегда устанавливаются терминаторы для предотвращения внутреннего отражения сигнала, причем волновое сопротивление терминатора должно совпадать с волновым сопротивлением кабеля, а один из терминаторов должен быть заземлен.

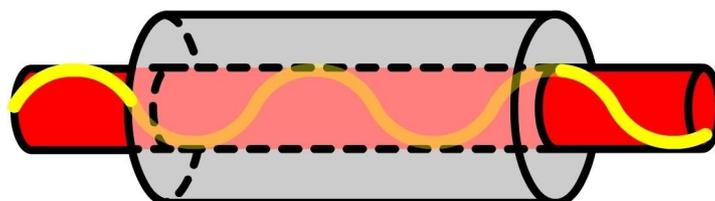


Рисунок 5.8 – Структура коаксиального кабеля

На сегодняшний день коаксиальный кабель используется очень редко из-за его стоимости, сложности в прокладке и низких технических характеристик. Допустимое значение скорости передачи данных в компьютерных сетях, элементы которых связаны с помощью коаксиальных линий связи, равно 500 Мбит/с.

### **Оптоволоконный кабель**

В оптоволоконном кабеле информация распространяется по оптическим волокнам в виде модулированных световых импульсов. Основу оптоволоконного кабеля составляет прозрачное стекловолокно. Стекловолокно передает свет на расстояния до десятков километров с незначительным ослаблением. Структура оптоволоконного кабеля представлена на рисунке 9.

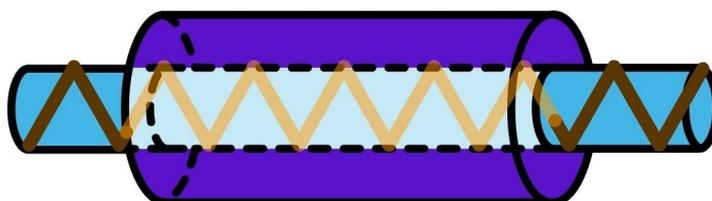


Рисунок 9 – Структура оптоволоконного кабеля

Оптоволокно передает сигналы только в одном направлении, поэтому кабель состоит из двух волокон, одно из которых используется для передачи данных, а другое – для приема данных. Скорость передачи в оптоволоконной сети составляет от 100 Мбит/с, теоретически данные по оптоволоконным линиям связи можно передавать со скоростью до 200 Гбит/с.

По уровню помехозащищенности и секретности передаваемой информации оптоволоконный кабель имеет самые высокие характеристики среди всех линий связи. Прослушать оптоволоконный кабель практически

невозможно, так как при несанкционированном доступе оптоволокну будет повреждено, что приведет к потере работоспособности сети.

### **Беспроводные линии передачи данных**

В последнее время большое распространение получили беспроводные линии связи, основным преимуществом которых является мобильность компьютеров в сети, что достигается отсутствием проводов. Беспроводные сети могут быть гибридными, то есть использовать несколько видов соединения компьютеров в рамках одной сети.

В большинстве случаев данные в беспроводных сетях передаются с помощью инфракрасного излучения и с помощью радиопередачи в узком диапазоне или рассеянном спектре. Рассмотрим каждый из этих видов подробнее.

### **Инфракрасные линии передачи данных**

В инфракрасных беспроводных сетях используют для передачи данных инфракрасные лучи. В подобных системах необходимо генерировать очень мощный сигнал, чтобы сгладить влияние на него других источников света. Скорость передачи в таких сетях высока и составляет 10 Мбит/с (рисунок 10). Различают 4 типа инфракрасных сетей: сети прямой видимости между приемником и передатчиком, сети на рассеянном излучении, когда сигнал от передатчика достигает приемника, отражаясь от стен и потолка, сети на отраженном излучении, широкополосные оптические сети.

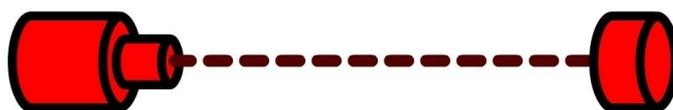


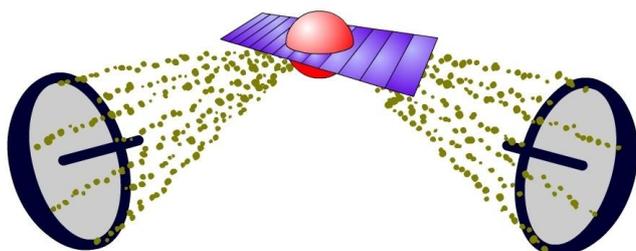
Рисунок 10 – Схема передачи данных в инфракрасных беспроводных сетях

Среди основных достоинств инфракрасных сетей можно отметить высокую скорость передачи данных и отсутствие чувствительности к

электромагнитным полям. К недостаткам использования этого класса сетей можно отнести трудности при передаче сигналов на большие расстояния, а также подверженность помехам со стороны сильных источников света.

### **Радиопередача данных**

Радиоканал использует в качестве носителя информации радиоволны. Скорость передачи данных в радиосетях может достигать 10 Мбит/с и зависит от выбора длины волны и способа кодирования информации. Радиопередача данных применяется как в персональных сетях, так и в глобальных компьютерных сетях. Существует два способа передачи данных по радиосетям в зависимости от ширины полосы передачи данных: узкополосные и широкополосные.



11 – Схема передачи данных в беспроводных радиосетях

Одной из самых популярных технологий беспроводной передачи данных является Bluetooth, которая обеспечивает беспроводную связь между компьютерами, сотовыми телефонами, приборами, датчиками на расстоянии до 100 м. Спецификация Bluetooth описывает пакетный способ передачи данных с разделением информации во времени. Радиообмен происходит в полосе частот от 2400 до 2483,5 МГц, которая применяется во многих странах для безлицензионного доступа.

## **АППАРАТУРА ЛИНИЙ СВЯЗИ**

Как было сказано выше, кроме физической среды передачи данных в состав линии связи входит также аппаратура приема-передачи данных и промежуточная аппаратура.

*Аппаратура приема-передачи данных в компьютерных сетях непосредственно присоединяет компьютеры к линиям связи и отвечает за передачу информации в физическую среду и прием из нее сигналов нужной формы, мощности и частоты.* Примерами аппаратуры передачи данных могут служить сетевой адаптер и модем.

*Сетевой адаптер* (сетевая карта, сетевая интерфейсная карта, NIC – Network Interface Card) – *аппаратный модуль, подключаемый к вычислительному устройству и предназначенный для соединения этого устройства с сетью.* Для подсоединения к сети по технологии Wi-Fi используются беспроводные адаптеры (рис. 1).



Рис. 1. Беспроводной сетевой адаптер

Сетевые карты бывают в виде плат расширения, которые устанавливаются в соответствующий слот материнской платы (рис. 1, 2), в виде отдельных устройств, подключаемых через порт USB, или могут быть

встроенными в материнские платы.



Рис. 2 Сетевой адаптер в виде платы расширения

**Модем** (МОдулятор-ДЕМОдулятор) – устройство, выполняющее преобразование цифровых сигналов (поточков бит) в аналоговую форму, для передачи их по каналам связи аналогового типа (например, телефонным линиям связи), а также преобразование принимаемых аналоговых сигналов в цифровую форму для обработки их вычислительным устройством (рис. 3).



Рис. 3. Внешний вид модема Ascorp

**Промежуточная аппаратура** используется на линиях связи большой протяженности и предназначена для улучшения качества сигнала и создания постоянного канала связи между двумя абонентами сети. К промежуточной аппаратуре относятся **усилители**, повышающие мощность сигнала, **регенераторы**, повышающие мощность и восстанавливающие форму импульсных сигналов, **мультиплексоры**, образующие из нескольких отдельных потоков данных общий поток, который передается по одному

физическому каналу данных, *демультиплексоры*, разделяющие суммарный поток на несколько составляющих его потоков данных, и др.

**ЛИТЕРАТУРА:** [4], [6], [10], [15], [27]

## **ЗАНЯТИЯ 31-32. БЕЗОПАСНОСТЬ, ГИГИЕНА, ЭРГОНОМИКА, РЕСУРСОСБЕРЕЖЕНИЕ.**

*План*

- 1. Безопасность, гигиена*
- 2. Эргономика*
- 3. Ресурсосбережение*

### **Безопасность, гигиена, эргономика, ресурсосбережение**

*Безопасность - состояние защищённости жизненно важных интересов личности, общества, организации, предприятия от потенциально и реально существующих угроз, или отсутствие таких угроз.*

*Гигиена - наука, изучающая влияние факторов внешней среды на организм человека с целью оптимизации благоприятного и профилактики неблагоприятного воздействия.*

*Гигиена труда – наука изучающая воздействие производственной среды и факторов производственного процесса на человека.*

*Эргономика (от греч. érgon — работа и nómos — закон), научная дисциплина, комплексно изучающая человека (группу людей) в конкретных условиях его деятельности в современном производстве. Это наука о том, как люди с их различными физическими данными и особенностями*

жизнедеятельности взаимодействуют с оборудованием и машинами, которыми они пользуются.

**Цель эргономики** состоит в том, чтобы обеспечить комфорт, эффективность и безопасность при пользовании компьютерами уже на этапе разработки клавиатур, компьютерных плат, рабочей мебели и др. для устранения физического дискомфорта и проблем со здоровьем на рабочем месте.

Эргономика возникла в 1920-х годах, в связи со значительным усложнением техники, которой должен управлять человек в своей деятельности. Термин «эргономика» был принят в Великобритании в 1949 году/ В СССР в 1920-е годы предлагалось название «эргология».

Современная эргономика изучает действия человека в процессе работы, скорость освоения им новой техники, затраты его энергии, производительность и интенсивность при конкретных видах деятельности.

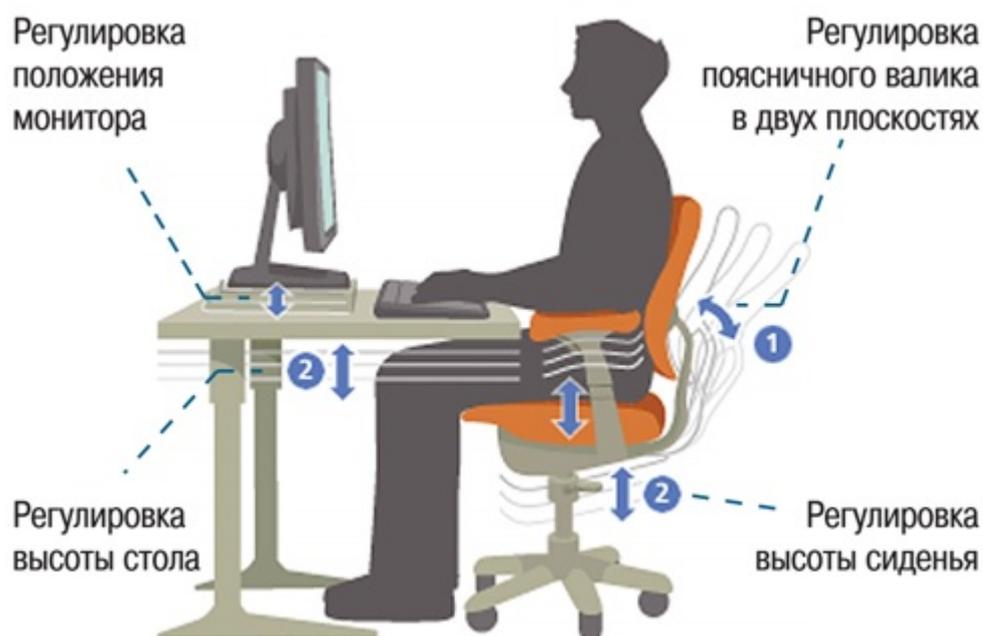
Информатика определяет сферу человеческой деятельности, связанную с процессами хранения, преобразования и передачи информации с помощью компьютера. В процессе изучения информатики надо не только научиться работать на компьютере, но и уметь целенаправленно его использовать для познания и созидания окружающего нас мира. В связи с тем, что всё больше людей проводят много времени перед компьютерными мониторами, ученые многих областей, включая анатомию, психологию и охрану окружающей среды, вовлекаются в изучение правильных, с точки зрения эргономики, условий работы.

Главной частью профилактических мероприятий в эргономике является правильная посадка.

### **Рабочее место.**

Чтобы заниматься было комфортно, чтобы не нанести вреда своему здоровью, Вы должны уметь правильно организовать свое рабочее место.

Правильная рабочая поза позволяет избегать перенапряжения мышц, способствует лучшему кровотоку и дыханию.



**Негативные последствия работы за монитором возникают из-за того, что:**

- а) наш глаз предназначен для восприятия отражённого света, а не излучаемого, как в случае с монитором (телевизором)
- б) пользователю приходится вглядываться в линии и буквы на экране, что приводит к повышенному напряжению глазных

### **Система гигиенических требований.**

Длительная работа с компьютером может приводить к расстройствам состояния здоровья.

Кратковременная работа с компьютером, установленным с грубыми нарушениями гигиенических норм и правил, приводит к повышенному утомлению.

Вредное воздействие компьютерной системы на организм человека является комплексным:

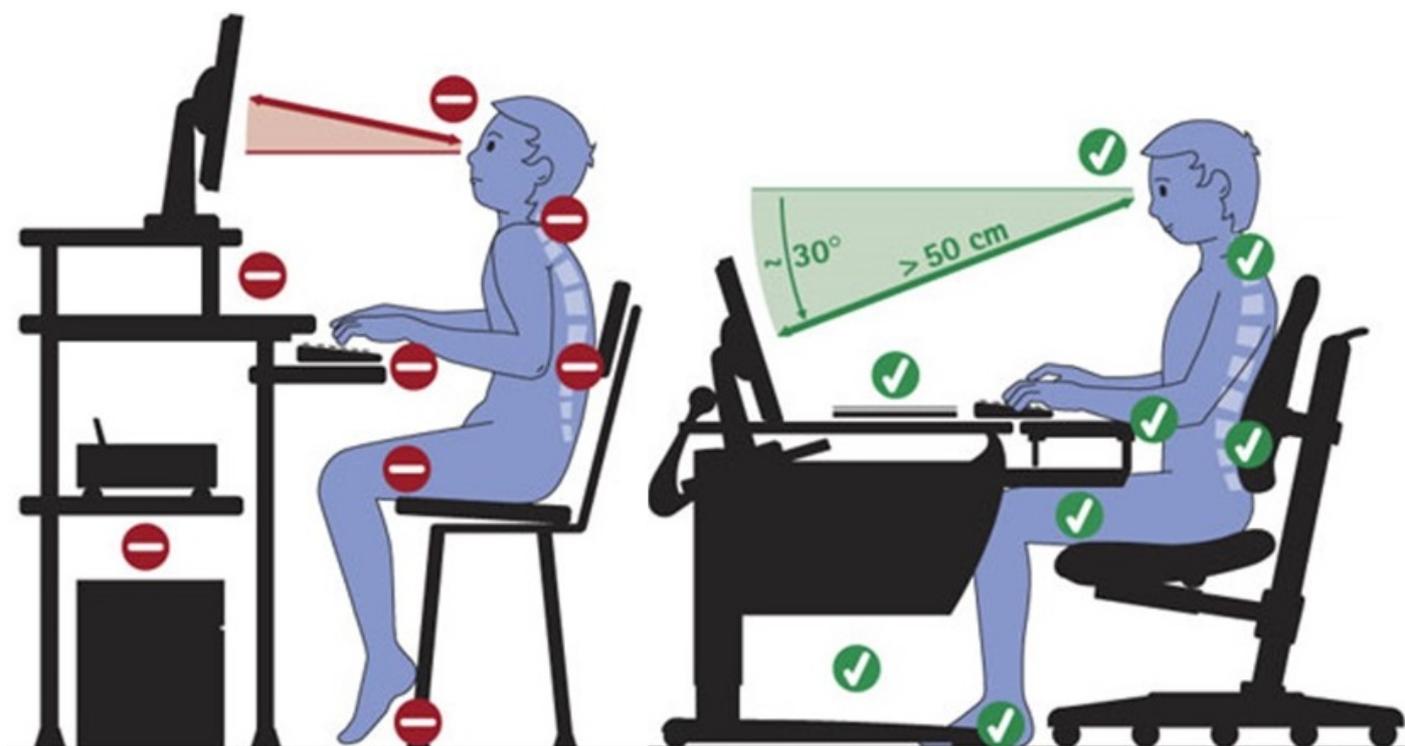
- параметры монитора оказывают влияние на органы зрения
- оборудование рабочего места влияет на органы опорно-двигательной системы
- характер расположения оборудования в компьютерном классе и режим его использования влияет как на общее психофизиологическое состояние организма, так и на органы зрения.

### **Правильная рабочая поза**

- Следует сидеть прямо (не сутулясь) и опираться спиной о спинку кресла. Прогибать спину в поясничном отделе нужно не назад, а, наоборот, немного в перед.
- Колени - на уровне бедер или немного ниже. При таком положении ног не возникает напряжение мышц.
- Нельзя скрещивать ноги, класть ногу на ногу - это нарушает циркуляцию крови из-за сдавливания сосудов. Лучше держать обе стопы на подставке или полу.
- Необходимо сохранять прямой угол (90°) в области локтевых, тазобедренных и голеностопных суставов.
- Экран монитора должен находиться от глаз пользователя на оптимальном расстоянии 60-70 см, но не ближе 50 см с учетом размеров алфавитно-цифровых знаков и символов.
- Не располагайте рядом с монитором блестящие и отражающие свет предметы
- Поверхность экрана должна быть чистой и без световых бликов.

*Хочешь сберечь здоровье?  
Не сиди так!*

*Правильная рабочая поза при  
работе с компьютером*



Ресурсосбережение - это основная результирующая часть НТП (научно-технического прогресса), представляющая собой эколого-социально-экономический эффект, полученный за счет рационализации потребления ресурсов.

В настоящее время вопросы ресурсосбережения приобретают особую актуальность. Ресурсосбережение рассматривается в узком смысле как мероприятия по изысканию резервов на основе снижения отходов и потерь. Сущность ресурсосберегающей деятельности заключается в комплексном использовании ресурсов, максимальном устранении всех видов потерь, возможно более полном вовлечении в хозяйственный оборот вторичных материальных и энергетических ресурсов. Центральными звеньями ресурсосбережения являются экономика, техника, технология и экология.

ЛИТЕРАТУРА: [4], [6], [10], [15], [27]

## ЗАНЯТИЕ 33. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРА

*План*

1. Классификация ПО
2. Системное ПО
3. Инструментальное ПО
4. Прикладное ПО

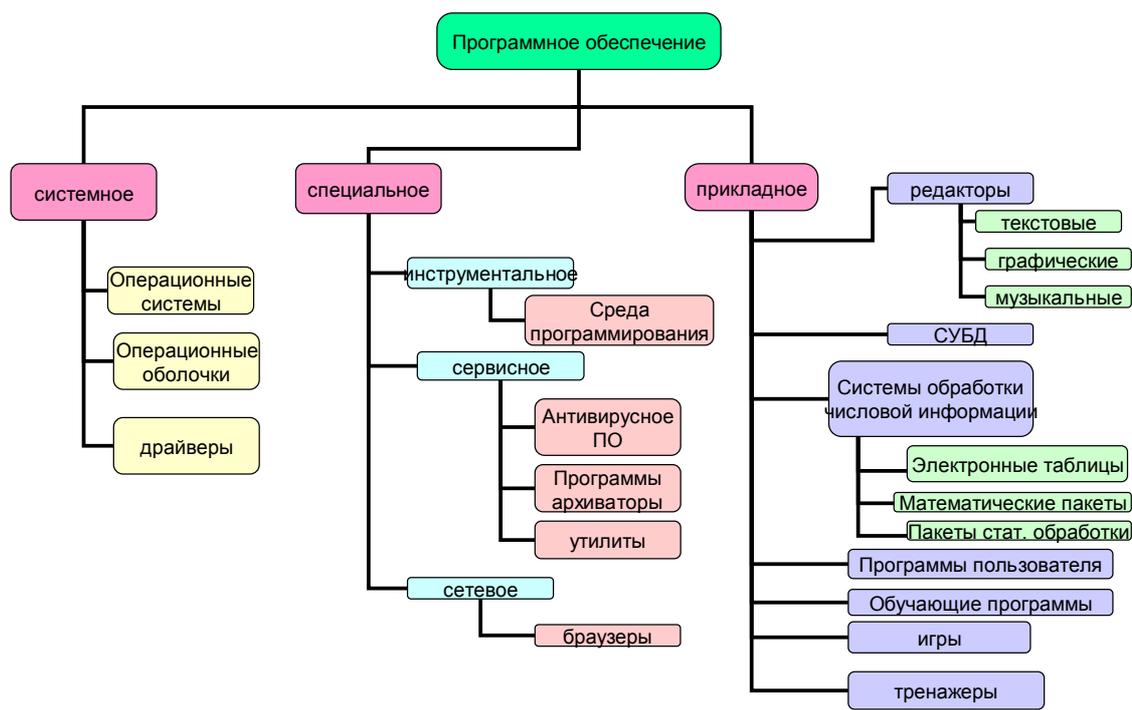
### ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРА

*Программное обеспéчение* — наряду с аппаратными средствами, важнейшая составляющая информационных технологий, включающая компьютерные программы и данные, предназначенные для решения определённого круга задач и хранящиеся на машинных носителях.

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

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

## КЛАССИФИКАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПК



7

Рис. 1. Иерархия программного обеспечения.

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

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

**Системное ПО** используется для обеспечения работы компьютера самого по себе и выполнения прикладных программ. Конкретные виды системного

программного обеспечения включают загрузчики, операционные системы, драйверы устройств, утилиты (сервисные программы). Наиболее общая часть системного программного обеспечения – операционная система.

### **Системное программное обеспечение. Операционные системы (ОС)**

*Операционная система, ОС (OS - operating system) — базовый комплекс компьютерных программ, обеспечивающий управление аппаратными средствами компьютера, работу с файлами, ввод и вывод данных, а также выполнение прикладных программ и утилит.*

При включении компьютера операционная система загружается в память раньше остальных программ и затем служит платформой и средой для их работы.

С 1990-х наиболее распространёнными операционными системами являются ОС семейства Microsoft Windows и системы класса UNIX (особенно Linux).

#### **Основные функции ОС:**

- 1. Загрузка приложений в оперативную память и их выполнение;*
- 2. Стандартизованный доступ к периферийным устройствам (устройствам ввода-вывода);*
- 3. Управление оперативной памятью;*
- 4. Управление энергонезависимой памятью (жесткий диск, компакт-диски и т.д.), как правило, с помощью файловой системы*
- 5. Пользовательский интерфейс.*

#### **Дополнительные функции ОС:**

- 1. Параллельное или псевдопараллельное выполнение задач (многозадачность);*
- 2. Взаимодействие между процессами;*

3. Межмашинное взаимодействие (компьютерная сеть);
4. Защита самой системы, а также пользовательских данных и программ от вредоносных действий пользователей или приложений;
5. Разграничение прав доступа и многопользовательский режим работы (аутентификация, авторизация).

### Установка и загрузка операционной системы (ОС)

Операционные системы распространяются в форме **дистрибутивов** на оптических дисках.



В процессе установки операционной системы файлы ОС копируются с диска дистрибутива на жесткий диск компьютера. После установки файлы ОС хранятся в долговременной памяти на жестком диске, который называется *системным*. Для выполнения программ ОС необходима их загрузка с системного диска в оперативную память.

Загрузка ОС начинается в одном из трех случаев – после:

- включения питания компьютера;
- нажатия кнопки **Reset** на системном блоке компьютера;

В процессе загрузки операционной системы:

- производится тестирование работоспособности процессора, памяти и других аппаратных средств компьютера (краткие диагностические сообщения о процессе тестирования выводятся на экран монитора);
- после окончания загрузки ОС пользователь получает возможность управлять компьютером с использованием графического интерфейса ОС.

## Интегрированные программные продукты и прикладное ПО

Операционная система Microsoft Windows включает в себя стандартные приложения, такие как браузер Internet Explorer, почтовый клиент Outlook Express, проигрыватель Windows Media Player.

Вокруг факта включения таких стандартных продуктов в ОС Windows разгорается много споров и дискуссий, поскольку это создает серьезное препятствие для распространения конкурирующих продуктов.



Рис. 6. Логотип пакета MS Office.

**Приложение** – это программа, дающая возможность обрабатывать текстовую, графическую, числовую, аудио- и видеoinформацию; работать в компьютерных сетях, не владея программированием. Приложение функционирует под управлением определенной операционной системы.

**Приложения общего назначения** нужны каждому пользователю для создания и редактирования документов различного типа и назначения.

Для обработки числовых данных используют **калькуляторы** и **электронные таблицы**. Для редактирования и форматирования используют **текстовые редакторы**. Для создания и редактирования рисунков, создания чертежей и анимации, редактирования видео существуют различные типы **приложений для компьютерной графики**. **Звуковые редакторы** позволяют обрабатывать звуковую информацию. Для воспроизведения звука, анимации и видео были созданы специальные приложения – **мультимедиа проигрыватели**.

В последние годы широкое распространение получили **программы разработки презентаций**. Презентации позволяют демонстрировать на экране слайды, содержащие текст, изображения, анимацию и звук. Поэтому презентации используются в процессе выступлений на конференциях, для рекламы, на уроке и т.д.

Для упорядоченного хранения и обработки больших объемов информации используются **системы управления базами данных**. Они позволяют осуществлять поиск и сортировку данных.

Для глобальных и локальных компьютерных сетей созданы различные **коммуникационные программы**. Для путешествий по Интернету необходимы браузеры, для работы с электронной почтой – почтовые программы, для общения в Интернете – программы интерактивного общения.

**Приложения специального назначения** – используются для профессионального применения в различных сферах деятельности. Это – системы компьютерного черчения, компьютерные словари и энциклопедии, системы автоматического перевода, системы распознавания текста, бухгалтерские программы, электронные учебники и репетиторы, тесты по различным предметам, компьютерные игры.

Приложения для пользователей разрабатывают программисты, которые используют для этого **системы программирования** (например, системы объектно-ориентированного визуального программирования Visual Basic и Delphi).

Для MS Windows существует очень удобный и освоенный большинством пользователей пакет прикладных программ Microsoft Office, включающий:

- *текстовый процессор MS Word,*
- *табличный процессор MS Excel,*
- *органайзер MS Outlook,*
- *приложение для подготовки презентаций MS PowerPoint,*

- *приложение для управления базами данных MS Access.*

Не стоит забывать, что и ОС Windows и абсолютное большинство популярных прикладных программ под Windows имеют лицензию copyright, т.е. являются проприетарным ПО. Соответственно каждая копия такой программы должна приобретаться за деньги.

Используя нелицензионное (пиратское) ПО, защищенное лицензией copyright, вы нарушаете законодательство о защите авторских прав.

### **Драйверы**

С ПК могут сопрягаться разнообразные устройства: видеокарта, звуковая карта, принтер, сканер, манипуляторы, дисководы, цифровые фотоаппараты, сотовые телефоны... Каждое из них имеет свой набор команд – свой «язык». *Чтобы конкретная операционная система могла управлять конкретным устройством, прибегают к помощи программ–«переводчиков», знающих с одной стороны язык команд конкретного устройства, а с другой – язык конкретной операционной системы, под управлением которой должно работать это устройство.*

*Такая программа называется драйвером (driver) и поставляется вместе с устройством его производителем. Производители аппаратного ПО, как правило, также размещают драйверы, созданных ими устройств, на своих web-сайтах.*

## **Задания по теме «Программное обеспечение»**

### **Задание 1**

Продолжи предложение:

- Текстовые редакторы используются для...
- Системы управления базами данных используются для...
- Программы-архиваторы позволяют...
- Графические редакторы используются для...
- Без системных программ работа компьютера станет...
- Электронные таблицы используют для...

## **Задание 2**

### **Найди и исправь ошибки:**

1. Для того, чтобы поработать за компьютером я в первую очередь обязательно включаю монитор и принтер, набираю текст с помощью сканера и сохраняю его в файле оперативной памяти.
2. Для рисования картинки я использую джойстик, а если мне нужно распечатать, я вставляю бумагу в монитор.
3. Чтобы не набирать текст с клавиатуры я использую принтер, а для переноса его на компьютер друга сохраняю его в ПЗУ.
4. Фильмы я смотрю с помощью системного блока, а музыку прослушиваю с помощью графопостроителя, который иногда называют плоттером.
5. Для длительного хранения документов и программ я использую процессор, а оперативная память позволяет мне переносить их на другой компьютер.

**ЛИТЕРАТУРА: [4], [6], [10], [15], [27]**

**ЗАНЯТИЯ 34-35. ОПЕРАЦИОННЫЕ СИСТЕМЫ. УСТАНОВКА И УДАЛЕНИЕ ПРОГРАММ.**

## *План*

- 1. Установка программ*
- 2. Конфигурирование ОС*

### ***Установка операционной системы***

До установки ОС должны быть выполнены следующие этапы подготовки диска: *логическое разбиение и логическое форматирование*. Для логического разбиения предназначены специальные утилиты, например Partition Magic. С помощью этой же утилиты можно отформатировать логический диск под файловую систему.

Этап логического форматирования реализуется при помощи средств программы установки большинства ОС. Например, при установке систем Windows и выбора логического диска для установки, этот диск предлагается отформатировать под NTFS или FAT.

Чаще всего ОС устанавливается с установочного компакт-диска. В случае необходимости потребуется изменить настройки BIOS – в разделе First Boot Device первым загрузочным устройством нужно выбрать CD-ROM.

После того как компьютер загрузится с установочного CD-диска, начнется сам процесс установки ОС. В большинстве случаев этот процесс включает следующие этапы:

#### **I этап. Сбор сведений: выбор параметров установки.**

*-сбор информации о компьютере, анализ конфигурации (проверка того, соответствует ли аппаратная конфигурация компьютера требованиям, необходимым для использования ОС).*

*-выбор варианта установки (обновление или новая).*

*-ознакомление с лицензионным соглашением.*

*-указание серийного номера.*

*-выбор специальных параметров установки: использование многовариантной загрузки, выбор расположения файлов дистрибутива, имя папки для сохранения файлов ОС, специальные возможности, экранная лупа, экранный диктор, языки ввода и раскладки клавиатуры.*

### **II этап. Динамическое обновление.**

*Появление новых аппаратных компонентов влечет за собой появление новых драйверов, которых не было на момент появления ОС и которые отсутствуют в дистрибутиве. Процедура динамического обновления выполняет проверку наличия обновленных драйверов и загрузку самых последних версий драйверов и может выполняться в процессе установки ОС.*

### **III этап. Подготовка к установке, начало установки.**

*-копирование на жесткий диск установочных файлов и перезагрузка системы.*

*-выбор логического диска для установки, его проверка и, возможно, форматирование под файловую систему.*

### **IV этап. Установка.**

*-ввод персональной информации о пользователе (например, имя и организация).*

*-выбор опций языков ввода и региональных стандартов.*

*-ввод имени компьютера и пароля администратора.*

*-настройка параметров отображения даты и времени.*

*-настройка параметров сети.*

### **V этап. Завершение процесса установки.**

*-завершение настройки устройств.*

*-настройка подключения к сети Интернет.*

*-активация операционной системы.*

*-создание учетных записей локальных пользователей.*

*-завершение конфигурирования.*

После установки ОС выполняются следующие действия: установка драйверов к материнской плате и ко всем устройствам.

Разделение всех действий установки ОС на этапы является условным. В зависимости от вида ОС, некоторые действия в процессе инсталляции могут отсутствовать или выполняться на другом этапе.

Установка серверных версий ОС отличается от установки ОС на локальные компьютеры и требует установку и конфигурирование основных компонентов сервера.

### ***Конфигурирование операционной системы***

Любая операционная система предоставляет пользователю ряд инструментов для настройки (конфигурирования) своей среды. Пользователь может выполнить настройки: оборудования, интерфейса, подключения к сети, администрирования и др.

В ОС Windows инструменты для конфигурирования находятся в папке *Панель управления*. Доступ к этой папке осуществляется через Главное меню. После нажатия кнопки «Пуск» в меню следует выбрать пункт Настройка, а затем в подменю - пункт Панель управления. Все инструменты можно разделить на два класса:

*класс инструментов для системной настройки.* Эти инструменты используются для настройки оборудования и установки приложений. К ним можно отнести: установку или удаление программ; установку, тестирование и конфигурирование оборудования; изменение настройки Интернет; установку паролей и др.

*класс инструментов для настройки пользовательского интерфейса.* Эти инструменты позволяют изменять внешний облик системы. К ним можно отнести; установку даты и времени, изменение оформления Рабочего стола

(звук, фон, заставку и др.), поддержку необходимого языка, выбор видов указателя мыши и др.

Еще одним инструментом системы, который может использоваться для конфигурирования является реестр. **Реестр** (от английского registry) - *база данных операционной системы, содержащая конфигурационные сведения. В реестре хранится информация по аппаратной конфигурации компьютера, различные настройки операционной системы и настройки устанавливаемых программ.*

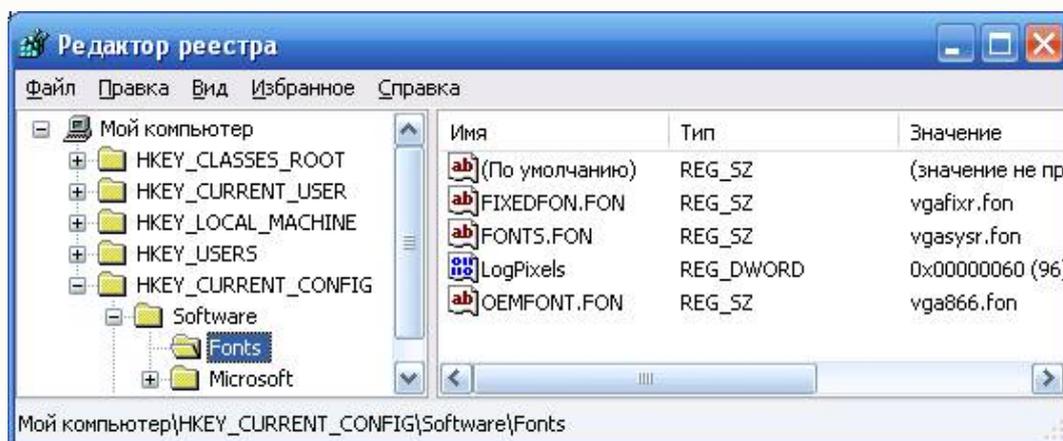
Например, существует возможность настроить практически все параметры среды Windows через реестр. Причем эти настройки зачастую невозможно подкорректировать стандартными средствами и инструментами самой операционной системы. В большинстве случаев, когда все функционирует нормально, необходимости заглядывать в реестр нет. Но иногда возникают ситуации, требующие непосредственного вмешательства в эту базу данных и тогда надо иметь хотя бы приблизительное представление о структуре реестра, его функционировании, месторасположении различной информации и принципов работы с ней.

Программа установки Windows предотвращает использование реестра неопытными пользователями, поэтому после установки Windows редактор реестра отсутствует в главном меню и на рабочем столе. А файл редактора реестра (REGEDIT.EXE) помещается в системную папку Windows в процессе инсталляции.

Самый простой способ открыть редактор реестра – это выполнить команду: [Пуск-Выполнить, набрать regedit] или сделать ярлык файла REGEDIT.EXE на Рабочем столе.

Информация, хранящаяся в иерархической базе данных реестра, собрана в разделы (key), которые содержат один или более подразделов (subkey). Каждый подраздел содержит параметры (value). После запуска редактора

реестра откроется окно программы «Редактор реестра», при этом количество разделов зависит от версии ОС.



С левой стороны расположено дерево реестра, а справа выводятся значения (ключи), содержащиеся в выбранном разделе. Каждый раздел (ветвь) соответствует определенному типу информации о пользователе, аппаратном обеспечении, приложении и т.д.

Для приведенного выше примера, название каждого корневого раздела начинается с HKEY\_, и каждый корневой раздел содержит несколько подразделов.

Назначение корневых разделов следующее:

*HKEY\_CLASSES\_ROOT* – отвечает за настройки рабочего стола, ярлыки, межпрограммные связи, технологии OLE.

*HKEY\_CURRENT\_USER* – хранит настройки текущего пользователя, рабочего стола и др.

*HKEY\_LOCAL\_MACHINE* – отвечает за настройки компьютера, параметры оборудования, драйверы, установленное программное обеспечение и его настройки.

*HKEY\_USERS* – хранит информацию и настройки оболочки Windows всех пользователей компьютера, плюс разграничение доступа.

*HKEY\_CURRENT\_CONFIG* – отвечает за текущую конфигурацию устройств Plug&Play и сведения о конфигурации компьютера с переменным составом аппаратных средств.

**ЛИТЕРАТУРА:** [\[4\]](#), [\[6\]](#), [\[10\]](#), [\[15\]](#), [\[27\]](#)

## **ЗАНЯТИЯ 37-38. ОПТИМИЗАЦИЯ РАБОТЫ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА. УТИЛИТЫ.**

*План*

- 1. Программы архиваторы*
- 2. Антивирусные программы*

### **Программы архиваторы**

Характерной особенностью большинства типов данных является их избыточность. Степень избыточности данных зависит от типа данных. Например, для видеоданных степень избыточности в несколько раз больше чем для графических данных, а степень избыточности графических данных, в свою очередь, больше чем степень избыточности текстовых данных. Другим фактором, влияющим на степень избыточности является принятая система кодирования. Примером систем кодирования могут быть обычные языки общения, которые являются ни чем другим, как системами кодирования понятий и идей для высказывания мыслей. Так, установлено, что кодирование текстовых данных с помощью средств русского языка дает в среднем избыточность на 20-25% большую чем кодирование аналогичных данных средствами английского языка.

Для человека избыточность данных часто связана с качеством информации, поскольку избыточность, как правило, улучшает понятность и восприятие информации. Однако, когда речь идет о хранении и передаче информации средствами компьютерной техники, то избыточность играет отрицательную роль, поскольку она приводит к возрастанию стоимости хранения и передачи информации. Особенно актуальной эта проблема становится в случае обработки огромных объемов информации при незначительных объемах носителей данных. В связи с этим, постоянно возникает проблема уменьшения избыточности или сжатия данных. Если методы сжатия данных применяются к готовым файлам, то часто вместо термина "сжатие данных" употребляют термин "архивация данных", сжатый вариант данных называют **архивом**, а программные средства, которые реализуют методы сжатия называются **архиваторами**.

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

1. Сжатие (архивация) файлов: используется для уменьшения размеров файлов при подготовке их к передаче каналами связи или к транспортированию на внешних носителях малой емкости;
2. Сжатие (архивация) папок: используется как средство уменьшения объема папок перед долгим хранением, например, при резервном копировании;
3. Сжатие (уплотнение) дисков: используется для повышения эффективности использования дискового пространства путем сжатия данных при записи их на носители информации (как правило, средствами операционной системы).

Существует много практических алгоритмов сжатия данных, но все они базируются на трех теоретических способах уменьшения избыточности данных. Первый способ состоит в изменении содержимого данных, второй - в

изменении структуры данных, а третий - в одновременном изменении как структуры, так и содержимого данных.

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

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

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

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

Существует много разных практических методов сжатия без потери информации, которые, как правило, имеют разную эффективность для разных типов данных и разных объемов. Однако, в основе этих методов лежат три теоретических алгоритма:

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

### Алгоритм RLE

В основе алгоритма RLE лежит идея выявления повторяющихся последовательностей данных и замены их более простой структурой, в которой указывается код данных и коэффициент повторения. Например, пусть задана такая последовательность данных, что подлежит сжатию:

1 1 1 1 2 2 3 4 4 4

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

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

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

### **Алгоритмы группы KWE**

В основе алгоритма сжатия по ключевым словам положен принцип кодирования лексических единиц группами байт фиксированной длины. Примером лексической единицы может быть обычное слово. На практике, на роль лексических единиц выбираются повторяющиеся последовательности символов, которые кодируются цепочкой символов (кодом) меньшей длины. Результат кодирования помещается в таблице, образуя так называемый словарь.

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

индекса совпадения и следующего за ним символа, прибавляется в словарь. В следующий раз, если эта фраза появится в сообщении, она может быть использована для построения более длинной фразы, что повышает меру сжатия информации.

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

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

### **Алгоритм Хаффмана**

В основе алгоритма Хаффмана лежит идея кодирования битовыми группами. Сначала проводится частотный анализ входной последовательности данных, то есть устанавливается частота вхождения каждого символа, встречающегося в ней. После этого, символы сортируются по уменьшению частоты вхождения.

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

Пусть задан текст, в котором буква 'A' входит 10 раз, буква 'B' - 8 раз, 'C' - 6 раз, 'D' - 5 раз, 'E' и 'F' - по 4 раза. Тогда один из возможных вариантов кодирования по алгоритму Хаффмана приведен в таблице 1.

Таблица 1.

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

Как видно из таблицы 1, размер входного текста до сжатия равен 37 байт, тогда как после сжатия - 93 бит, то есть около 12 байт (без учета длины словаря). Коэффициент сжатия равен 32%. Алгоритм Хаффмана универсальный, его можно применять для сжатия данных любых типов, но он малоэффективен для файлов маленьких размеров (за счет необходимости сохранения словаря).

На практике программные средства сжатия данных синтезируют эти три "чистых" алгоритмы, поскольку их эффективность зависит от типа и объема данных. В таблице 2 приведены распространенные форматы сжатия и соответствующие им программы-архиваторы, использующиеся на практике.

Таблица 2.

Формат сжатия	Операционная система MS DOS		Операционная система Windows	
		Программа	Программа	Программа

	архивации	разархивации	архивации	разархивации
ARJ	Arj.exe	Arj.exe	WinArj.exe	WinArj.exe
RAR	Rar.exe	Unrar.exe	WinRar.exe	WinRar.exe
ZIP	Pkzip.exe	Pkunzip.exe	WinZip.exe	WinZip.exe

Кроме того, современные архиваторы предоставляют пользователю полный спектр услуг для работы с архивами, основными из которых являются:

1. создание нового архива;
2. добавление файлов в существующий архив;
3. распаковывание файлов из архива;
4. создание самораспаковывающихся архивов (self-extractor archive);
5. создание распределенных архивов фиксированного размера для носителей маленькой емкости;
6. защита архивов паролями от несанкционированного доступа;
7. просмотр содержимого файлов разных форматов без предварительного распаковывания;
8. поиск файлов и данных внутри архива;
9. проверка на вирусы в архиве к распаковыванию;
10. выбор и настройка коэффициента сжатия.

### **Антивирусные программы**

**Компьютерный вирус** - это небольшая программа, написанная программистом высокой квалификации, способная к саморазмножению и

выполнению разных деструктивных действий. На сегодняшний день известно свыше 50 тыс. компьютерных вирусов.

Существует много разных версий относительно даты рождения первого компьютерного вируса. Однако большинство специалистов сходятся на мысли, что компьютерные вирусы, как таковые, впервые появились в 1986 году, хотя исторически возникновение вирусов тесно связано с идеей создания самовоспроизводящихся программ. Одним из "пионеров" среди компьютерных вирусов считается вирус "Brain", созданный пакистанским программистом по фамилии Алви. Только в США этот вирус поразил свыше 18 тыс. компьютеров. В начале эпохи компьютерных вирусов разработка вирусоподобных программ носила чисто исследовательский характер, постепенно превращаясь на откровенно вражеское отношение к пользователям безответственных, и даже криминальных "элементов". В ряде стран уголовное законодательство предусматривает ответственность за компьютерные преступления, в том числе за создание и распространение вирусов.

Вирусы действуют только программным путем. Они, как правило, присоединяются к файлу или проникают в тело файла. В этом случае говорят, что файл заражен вирусом. Вирус попадает в компьютер только вместе с зараженным файлом. Для активизации вируса нужно загрузить зараженный файл, и только после этого, вирус начинает действовать самостоятельно.

Некоторые вирусы во время запуска зараженного файла становятся резидентными (постоянно находятся в оперативной памяти компьютера) и могут заражать другие загружаемые файлы и программы. Другая разновидность вирусов сразу после активизации может быть причиной серьезных повреждений, например, форматировать жесткий диск. Действие вирусов может проявляться по-разному: от разных визуальных эффектов, мешающих работать, до полной потери информации. Большинство вирусов заражают исполнимые программы, то есть файлы с расширением .EXE и

.COM, хотя в последнее время большую популярность приобретают вирусы, распространяемые через систему электронной почты.

Следует заметить, что компьютерные вирусы способны заражать лишь компьютеры. Поэтому абсолютно абсурдными являются разные утверждения о влиянии компьютерных вирусов на пользователей компьютеров.

### **Основные источники вирусов:**

- дискета, на которой находятся зараженные вирусом файлы;
- компьютерная сеть, в том числе система электронной почты и Internet;
- жесткий диск, на который попал вирус в результате работы с зараженными программами;
- вирус, оставшийся в оперативной памяти после предшествующего пользователя.

### **Основные ранние признаки заражения компьютера вирусом:**

- уменьшение объема свободной оперативной памяти;
- замедление загрузки и работы компьютера;
- непонятные (без причин) изменения в файлах, а также изменения размеров и даты последней модификации файлов;
- ошибки при загрузке операционной системы;
- невозможность сохранять файлы в нужных каталогах;
- непонятные системные сообщения, музыкальные и визуальные эффекты и т.д.

### **Признаки активной фазы вируса:**

- исчезновение файлов;

- форматирование жесткого диска;
- невозможность загрузки файлов или операционной системы.

Существует очень много разных вирусов. Условно их можно классифицировать следующим образом:

1) **загрузочные вирусы** или **BOOT-вирусы** заражают boot-секторы дисков. Очень опасные, могут привести к полной потере всей информации, хранящейся на диске;

2) **файловые вирусы** заражают файлы. Делятся на:

- **вирусы, заражающие программы** (файлы с расширением .EXE и .COM);
- **макровирусы** **вирусы**, заражающие файлы данных, например, документы Word или рабочие книги Excel;
- **вирусы-спутники** используют имена других файлов;
- **вирусы семейства DIR** искажают системную информацию о файловых структурах;

3) **загрузочно-файловые вирусы** способные поражать как код boot-секторов, так и код файлов;

4) **вирусы-невидимки** или **STEALTH-вирусы** фальсифицируют информацию прочитанную из диска так, что программа, какой предназначена эта информация получает неверные данные. Эта технология, которую, иногда, так и называют Stealth-технологией, может использоваться как в BOOT-вирусах, так и в файловых вирусах;

5) **ретровирусы** заражают антивирусные программы, стараясь уничтожить их или сделать нетрудоспособными;

б) **вирусы-черви** снабжают небольшие сообщения электронной почты, так называемым заголовком, который по своей сути есть Web-адресом местонахождения самого вируса. При попытке прочитать такое сообщение вирус начинает считывать через глобальную сеть Internet свое 'тело' и после загрузки начинает деструктивное действие. Очень опасные, так как обнаружить их очень тяжело, в связи с тем, что зараженный файл фактически не содержит кода вируса.

Если не принимать меры для защиты от компьютерных вирусов, то следствия заражения могут быть очень серьезными. В ряде стран уголовное законодательство предусматривает ответственность за компьютерные преступления, в том числе за внедрение вирусов. Для защиты информации от вирусов используются общие и программные средства.

**К общим средствам, помогающим предотвратить заражение и его разрушительных последствий относят:**

- резервное копирование информации (создание копий файлов и системных областей жестких дисков);
- избежание пользования случайными и неизвестными программами. Чаще всего вирусы распространяются вместе с компьютерными программами;
- перезагрузка компьютера перед началом работы, в частности, в случае, если за этим компьютером работали другие пользователи;
- ограничение доступа к информации, в частности физическая защита дискеты во время копирования файлов с нее.

К программным средствам защиты относят разные антивирусные программы (антивирусы). Антивирус - это программа, выявляющая и обезвреживающая компьютерные вирусы. Следует заметить, что вирусы в своем развитии опережают антивирусные программы, поэтому даже в случае регулярного

пользования антивирусов, нет 100% гарантии безопасности. Антивирусные программы могут выявлять и уничтожать лишь известные вирусы, при появлении нового компьютерного вируса защиты от него не существует до тех пор, пока для него не будет разработан свой антивирус. Однако, много современных антивирусных пакетов имеют в своем составе специальный программный модуль, называемый эвристическим анализатором, который способен исследовать содержимое файлов на наличие кода, характерного для компьютерных вирусов. Это дает возможность своевременно выявлять и предупреждать об опасности заражения новым вирусом.

### **Различают такие типы антивирусных программ:**

- 1) **программы-детекторы:** предназначены для нахождения зараженных файлов одним из известных вирусов. Некоторые программы-детекторы могут также лечить файлы от вирусов или уничтожать зараженные файлы. Существуют специализированные, то есть предназначенные для борьбы с одним вирусом детекторы и полифаги, которые могут бороться с многими вирусами;
- 2) **программы-лекари:** предназначены для лечения зараженных дисков и программ. Лечение программы состоит в изъятии из зараженной программы тела вируса. Также могут быть как полифагами, так и специализированными;
- 3) **программы-ревизоры:** предназначены для выявления заражения вирусом файлов, а также нахождение поврежденных файлов. Эти программы запоминают данные о состоянии программы и системных областей дисков в нормальном состоянии (до заражения) и сравнивают эти данные в процессе работы компьютера. В случае несоответствия данных выводится сообщение о возможности заражения;
- 4) **лекари-ревизоры:** предназначены для выявления изменений в файлах и системных областях дисков и, в случае изменений, возвращают их в начальное состояние.

5) **программы-фильтры**: предназначены для перехвата обращений к операционной системе, которые используются вирусами для размножения и сообщают об этом пользователя. Пользователь может разрешить или запретить выполнение соответствующей операции. Такие программы являются резидентными, то есть они находятся в оперативной памяти компьютера.

б) **программы-вакцины**: используются для обработки файлов и boot-секторов с целью предупреждения заражения известными вирусами (в последнее время этот метод используется все чаще).

Следует заметить, что выбор одного "наилучшего" антивируса крайне ошибочное решение. Рекомендуется использовать несколько разных антивирусных пакетов одновременно. Выбирая антивирусную программу следует обратить внимание на такой параметр, как количество распознающих сигнатур (последовательность символов, которые гарантированно распознают вирус). Второй параметр - наличие эвристического анализатора неизвестных вирусов, его присутствие очень полезно, но существенно замедляет время работы программы. На сегодняшний день существует большое количество разнообразных антивирусных программ. Рассмотрим коротко, распространенные в странах СНГ.

## **DRWEB**

Один из лучших антивирусов с мощным алгоритмом нахождения вирусов. Полифаг, способный проверять файлы в архивах, документы Word и рабочие книги Excel, выявляет полиморфные вирусы, которые в последнее время, получают все большее распространение. Достаточно сказать, что эпидемию очень опасного вируса OneHalf остановил именно DrWeb. Эвристический анализатор DrWeb, исследуя программы на наличие фрагментов кода, характерных для вирусов, разрешает найти почти 90% неизвестных вирусов. При загрузке программы, в первую очередь DrWeb проверяет самого себя на

целостность, после чего тестирует оперативную память. Программа может работать в диалоговом режиме, имеет удобный настраиваемый интерфейс пользователя.

## **ADINF**

Антивирус-ревизор диска ADINF (Advanced DiskINFoscope) разрешает находить и уничтожать, как существующие обычные, stealth- и полиморфные вирусы, так и совсем новые. Антивирус имеет в своем распоряжении лечащий блок ревизора ADINF - Adinf Cure Module - что может обезвредить до 97% всех вирусов. Эту цифру приводит "Диалогнаука", исходя из результатов тестирования, которое происходило на коллекциях вирусов двух признанных авторитетов в этой области - Д.Н.Лозинского и фирмы Dr.Solomon's (Великобритания).

ADINF загружается автоматически в случае включения компьютера и контролирует boot-сектор и файлы на диске (дата и время создания, длина, контрольная сумма), выводя сообщения про их изменения. Благодаря тому, что ADINF осуществляет дисковые операции в обход операционной системы, обращаясь к функциям BIOS, достигаются не только возможность выявления активных stealth-вирусов, но и высокая скорость проверки диска. Если найден boot-вирус, то ADINF просто восстановит предшествующий загрузочный сектор, который хранится в его таблице. Если вирус файловый, то здесь на помощь приходит лечащий блок Adinf Cure Module, который на основе отчета основного модуля о зараженных файлах сравнивает новые параметры файлов с предыдущими, хранящиеся в специальных таблицах. При выявлении расхождений ADINF восстанавливает предыдущее состояние файла, а не уничтожает тело вируса, как это делают полифаги.

ЛИТЕРАТУРА: [4], [6], [10], [15], [27]

## ЗАНЯТИЯ 39-40. ДОСТУП К ДАННЫМ, РАСПОЛОЖЕННЫМ НА РАБОЧИХ ЛИСТАХ. СОЗДАНИЕ МАКРОСОВ.

*План*

- 1. Введение в макросы*
- 2. Создание макросов*
- 3. Отладка макросов*

Для автоматизации часто выполняемых в Microsoft Excel задач можно записать макрос. **Макрос** представляет собой действие (или набор действий), которое можно выполнять любое количество раз. С помощью макросов можно автоматизировать повторяющиеся действия. Например: быструю вставку группы ячеек, применение сложного форматирования, выполнение операций над текстом, печать данных в определенном формате, анализ выделенного диапазона ячеек и т.д.

При создании макроса записываются щелчки мышью и нажатия клавиш. После создания макроса его можно отредактировать, чтобы изменить выполняемые им действия.

### **Введение в макросы**

**Макрос** – это программа, состоящая из списка команд, которые должны быть выполнены приложением. Макрос служит для объединения нескольких различных действий в одну процедуру. Такой список команд состоит, в основном, из макрооператоров, тесно связанных с командами приложений из Microsoft Office. Большая часть макрооператоров

соответствует командам меню или параметрам, которые задаются в диалоговых окнах.

Excel представляет два способа создания макросов:

1. написать макрос вручную, используя код VBA
2. записать макрос, используя макрорекордер Excel.

Хотя написание макросов вручную является наиболее мощным и гибким методом, использование макрорекордера является наиболее простым подходом.

*Макрорекордер* - это небольшая программа, встроенная в Excel, которая переводит любое действие пользователя на язык программирования VBA и записывает получившуюся команду в программный модуль. Если мы включим макрорекордер на запись, а затем начнем создавать свой еженедельный отчет, то макрорекордер начнет записывать команды вслед за каждым нашим действием и, в итоге, мы получим макрос создающий отчет как если бы он был написан программистом. Такой способ создания макросов не требует знаний пользователя о программировании и VBA и позволяет пользоваться макросами как неким аналогом видеозаписи: включил запись, выполнил операции, перемотал пленку и запустил выполнение тех же действий еще раз. Естественно у такого способа есть свои плюсы и минусы:

- Макрорекордер записывает только те действия, которые выполняются в пределах окна Microsoft Excel. Как только вы закрываете Excel или переключаетесь в другую программу - запись останавливается.

- Макрорекордер может записать только те действия, для которых есть команды меню или кнопки в Excel. Программист же может написать макрос, который делает то, что Excel никогда не умел (сортировку по цвету, например или что-то подобное).

- Если во время записи макроса макрорекордером вы ошиблись - ошибка будет записана. Однако смело можете давить на кнопку отмены последнего

действия (Undo) - во время записи макроса макрорекордером она не просто возвращает Вас в предыдущее состояние, но и стирает последнюю записанную команду на VBA.

Можно выделить три основные разновидности макросов:

1. **командные** – наиболее распространенные макросы, которые обычно состоят из операторов, эквивалентных тем или иным командам меню или параметрам диалоговых окон. Основным предназначением таких макросов является выполнение действий, аналогичных командам меню, т. е. изменение окружения и основных объектов приложения. Например, изменение рабочего листа или рабочего пространства Microsoft Excel, сохранение или вывод на печать и т. п. Таким образом, в результате выполнения макроса вносятся изменения либо в обрабатываемый документ, либо в общую среду приложения;

2. **пользовательские функции** – работают аналогично имеющимся встроенным функциям Microsoft Excel. Отличие этих функций от командных макросов состоит в том, что они используют значения передаваемых им аргументов, производят некоторые вычисления и возвращают результат в точку вызова, но не изменяют среду приложения;

3. **макрофункции (подпрограммы)** – представляют собой сочетание командных макросов и пользовательских функций. Наряду с тем, что они, подобно пользовательским функциям, могут использовать аргументы и возвращать результат, макрофункции, как и командные макросы, способны еще и изменять среду приложения. Чаще всего макрофункции вызываются из других макросов и активно используются для модульного программирования. Если необходимо в различных макросах выполнить ряд одинаковых действий, то эти действия обычно выделяются в отдельную макрофункцию (подпрограмму).

В Excel каждый макрос привязывается к книге и сохраняется в XLSM-файле этой книги. *Как только открывается книга, которая содержит какие-либо макросы, Excel делает эти макросы доступными любой другой открытой в настоящий момент книге.*

Когда записывается макрос, Excel предоставляет три варианта его сохранения:

1. *Эта книга* – сохранение макроса в текущей книге. Необходимо сохранять книгу в одном из поддерживающих макросы форматов (XLSM или XLSB).

2. *Новая книга* – Excel автоматически создаст новую книгу и разместит в ней макрос. Книга затем должна быть сохранена в одном из поддерживающих макросы форматов (XLSM или XLSB).

3. *Личная книга макросов* – Excel сохраняет макрос в специальной скрытой книге, названной Personal.xlsm в папке XLStart. Каждый пользователь с учетной записью имеет отдельную личную книгу.

Можно скопировать или переместить макрос в любую открытую книгу или в личную книгу. Макросы можно использовать для быстрого получения чернового варианта кода, который в дальнейшем можно отредактировать.

Следует помнить о последовательности действий, связанной с разработкой макросов:

1. *Логическая разработка процедуры.* Прежде всего, необходимо определить, что следует получить в результате выполнения макроса и какова логическая последовательность действий для получения данного результата.

2. *Подготовка документа.* Выполните предварительные действия, которые не нужно включать в процедуру (например, создание нового рабочего листа или перемещение в конкретную часть рабочего листа и т. д.).

3. *Запись макроса с помощью макрорекордера.* Макрорекордер представляет собой транслятор, создающий программу (макрос) на языке

VBA, которая является результатом перевода на язык VBA действий пользователя с момента запуска макрорекордера до окончания записи макроса.

**Для создания макроса выполните следующие действия:**

а) перейдите на вкладку *Разработчик* и в группе Код щелкните по кнопке *Запись макроса*;

б) в открывшемся диалоговом окне *Запись макроса* установите параметры записываемой процедуры (ее имя, описание, сочетание клавиш для выполнения записанной процедуры, указание, для каких документов доступен макрос) и войдите в режим записи макроса – на экране, на вкладке *Разработчик* в группе *Код* кнопка *Запись макроса* изменится на кнопку *Остановить запись*; кроме того, кнопка *Пауза* также станет активной (если на время нужно приостановить запись макроса и выполнить другие действия с документом);

в) последовательно выполните все необходимые действия с документом и его содержимым, предусмотренные на первом этапе;

г) остановите запись (кнопка *Остановить запись* в группе *Код* на вкладке *Разработчик*).

#### *4. Просмотр созданной процедуры.*

На вкладке *Разработчик* в группе *Код* нажмите кнопку *Макросы*, в диалоговом окне *Макрос* выберите в списке имя нужного макроса и нажмите кнопку *Изменить*, откроется главное окно редактора Microsoft Visual Basic и окно Модуль (Module) с текстом выбранного макроса, который можно редактировать.

*5. Выполнение макроса.* Нажмите кнопку *Макросы* в группе *Код* на вкладке *Разработчик*, затем в открывшемся диалоговом окне *Макрос* в списке выберите имя макроса и нажмите кнопку *Выполнить*.

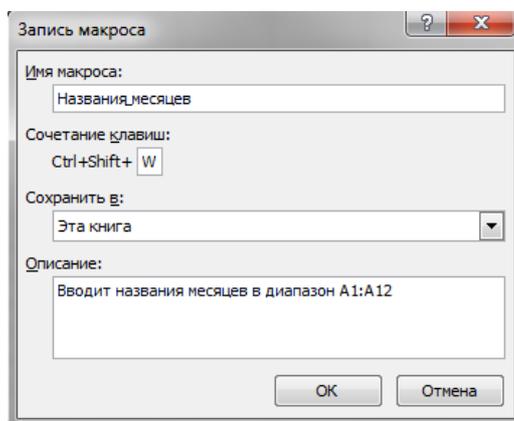
Можно назначить записанной процедуре кнопку и расположить на Панели быстрого доступа для упрощения вызова макроса.

### Пример создания макроса

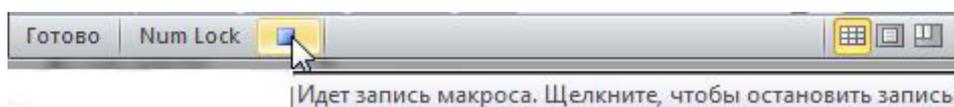
Создадим макрос, который будет выводить на экран столбец с названиями месяцев в ячейках A1:A12.

Для этого выполните следующие действия:

1. Установите курсор в какую-нибудь ячейку книги, за исключением A1.
2. Перейдите на вкладку Разработчик и в группе Код щелкните по кнопке Запись макроса, в открывшемся диалоговом окне Запись макроса установите параметры записываемой процедуры (рис.). Имя макроса может включать буквы, числа и символ подчеркивания, но не может содержать пробелы или другие специальные символы, максимальная длина имени макроса составляет 80 символов.



3. Выполните последовательность действий, которую будет выполнять макрос: установите курсор в ячейку A1, введите слово январь, поместите указатель мыши в правый нижний угол ячейки A1, при этом указатель мыши изменит вид на черный крестик; нажмите кнопку мыши и, удерживая ее нажатой, продолжите выделение до ячейки A12. Нажмите на кнопку Остановить запись на вкладке Разработчик или в строке состояния (рис. 52).



При записи макроса не включался режим записи с относительными ссылками, поэтому названия месяцев будут появляться только в ячейках A1:A12.

Если включен режим абсолютных ссылок, Excel сохраняет абсолютные ссылки для изменяемых ячеек. При выполнении макроса, он затрагивает только эти ячейки.

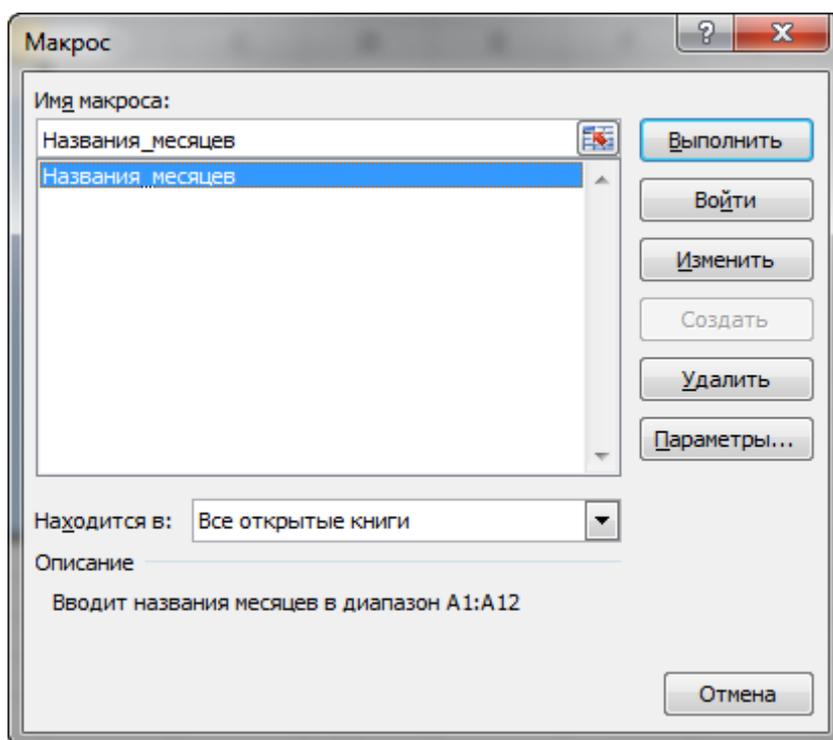
Если выбран режим относительных ссылок, Excel отслеживает, насколько сместились от стартовой позиции. При выполнении макроса, Excel принимает во внимание текущее расположение активной ячейки.

Изменить режим использования ссылок при записи макросов можно кнопкой Относительные ссылки в группе Код на вкладке Разработчик

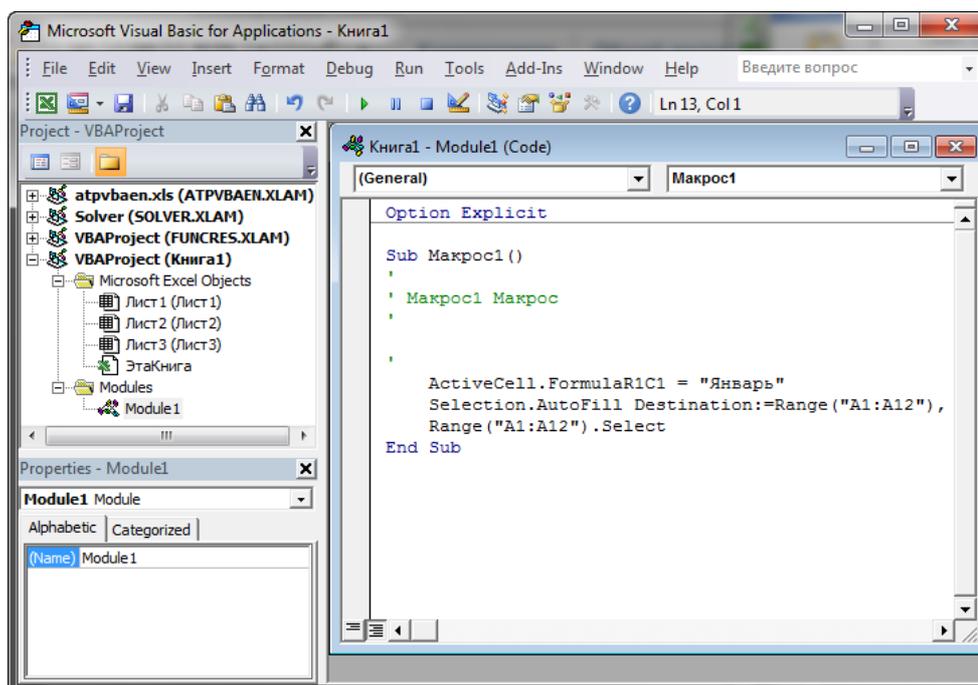
### **Код макроса и его редактирование**

При записи макроса все выполненные действия будут записаны в виде последовательности соответствующих операторов языка VBA. Вся совокупность этих операторов будет называться *исходным кодом* или *кодом макроса*.

Отредактировать текст записанного макроса можно с помощью кнопки Макросы группы Код на вкладке Разработчик или команды меню редактора VBE Tools  Macros (Сервис  Макросы). В результате на экран будет выведено диалоговое окно Макрос, показанное на рис. Выберите из списка перечисленных имен макросов требуемый макрос и щелкните на кнопке Изменить.



Откроется окно редактора VBA с программным кодом выбранного макроса, готовым для редактирования



В этом окне можно добавлять, удалять, копировать, изменять, вырезать или вставлять любые операторы на языке VBA.

Можно использовать макросы для создания заготовок элементов программ на языке VBA. Вместо того чтобы вручную программно описывать

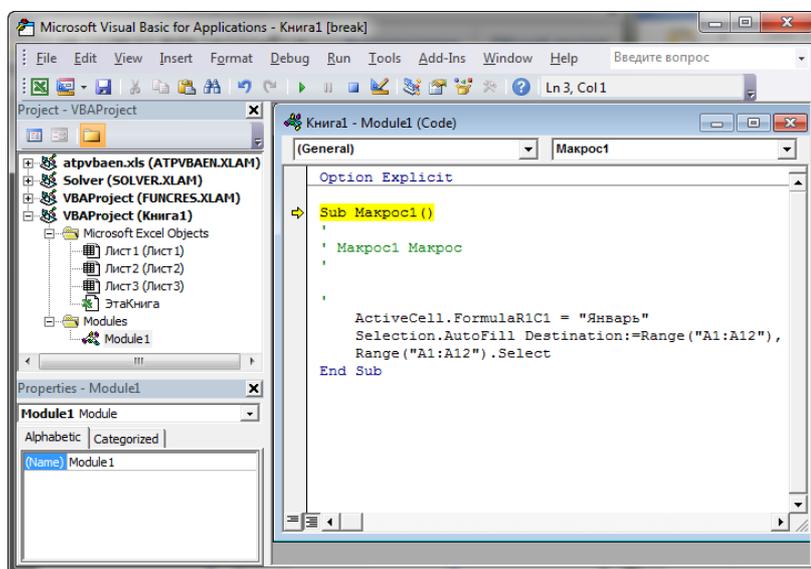
некоторую последовательность действий в приложении Office, нужно просто записать соответствующий макрос, а затем скопировать его код в текст создаваемой программы.

## Отладка макроса

*Отладка макроса* – это инструмент, который дает возможность выполнить код макроса построчно. При выполнении каждой строки макроса в рабочей книге сразу отображаются результаты.

Для отладки макроса выполните следующие действия:

- 1. Откройте макрос и щелкните мышью в любом месте внутри кода.*
- 2. В меню Debug выберите команду Step Into или нажмите клавишу F8. Excel подсветит первую строчку макроса (выражение Sub) желтым цветом и поместит рядом с ней стрелку. Стрелка означает, что строка готова к выполнению (рис.). Нажмите клавишу F8 для выполнения подсвеченной строки.*



Как только Excel выполнит эту строку, он перейдет к следующей строке и подсветит ее желтым цветом. В процессе перехода можно перейти в книгу, чтобы посмотреть результат выполнения предыдущей строки. Затем вернитесь в редактор Visual Basic для продолжения отладки. В любой момент можно

нажать кнопку Run, чтобы выполнить оставшуюся часть кода, или нажать клавишу F5. Чтобы отказаться от выполнения макроса нажмите кнопку Reset.

### **Запуск макроса**

Существует несколько способов выполнения макроса в Microsoft Excel:

1. С помощью команды *Макросы* на ленте в группе *Код* вкладки *Разработчик*.

2. С помощью сочетаний клавиш, назначенных макросу при создании.

3. С помощью кнопки на панели быстрого доступа.

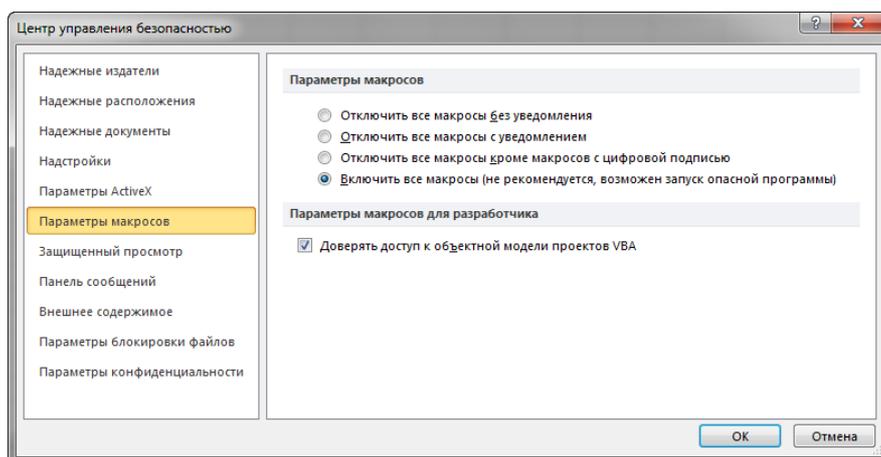
Чтобы добавить на панель быстрого доступа кнопку для запуска макроса, откройте вкладку *Файл*, нажмите кнопку *Параметры* и выберите пункт *Панель быстрого доступа*. В списке *Выбрать команды* из выберите элемент *Макросы*. В левом списке выберите созданный макрос и нажмите кнопку *Добавить*. Чтобы изменить изображение на кнопке макроса, в правом списке выберите макрос и нажмите кнопку *Изменить*. В диалоговом окне *Изменение кнопки* в поле *Символ* выберите нужное изображение для кнопки. В поле *Отображаемое имя* введите текст, который будет отображаться при наведении указателя мыши на кнопку. *Нажмите ОК*, кнопка макроса будет добавлена на панель быстрого доступа.

4. По щелчку области объекта, графического объекта или элемента управления. Чтобы назначить выполнение макроса элементу управления, на вкладке *Разработчик* в группе *Элементы управления* нажмите кнопку *Вставить*, выберите элемент управления формы, например элемент *Кнопка*. «Нарисуйте» элемент управления на рабочем листе, как только отпустите кнопку мыши, появится диалоговое окно *Назначить макрос объекту*, содержащее список доступных макросов, выберите нужный макрос. Для изменения надписи на элементе управления, в контекстном меню выберите команду *Изменить текст*.

5. Кроме того, макросы можно запускать автоматически при открытии книги. Если макрос записан и сохранен с именем Auto\_open, он будет запускаться при каждом открытии содержащей его книги. Другим способом автоматического запуска макроса при открытии книги является написание процедуры на языке VBA в событии Open книги в редакторе VBE. Событие Open представляет собой встроенное событие книги, запускающее код макроса при каждом открытии книги.

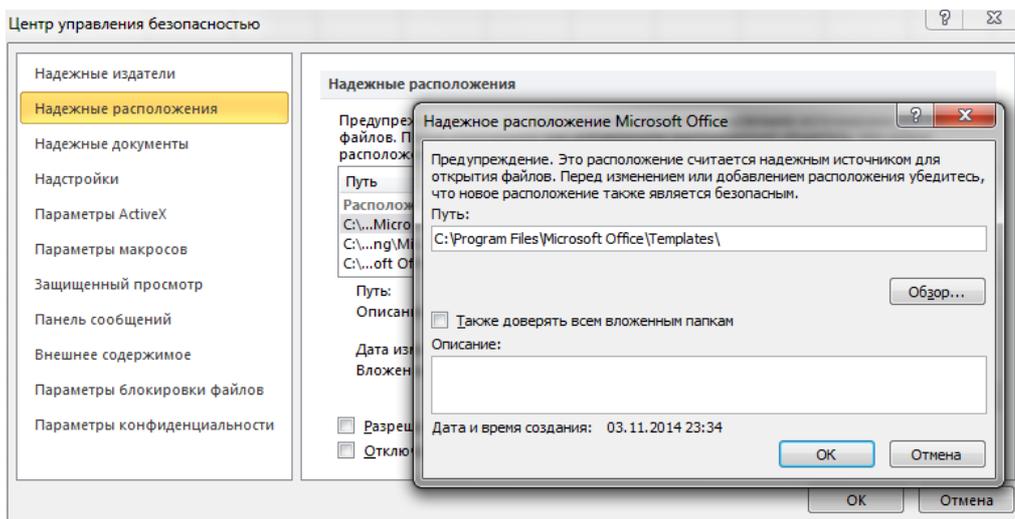
### Параметры безопасности макросов

На вкладке Разработчик в группе Код находится кнопка Безопасность макросов, которая открывает окно Центр управления безопасностью в категории Параметры макросов (рис)



В этом окне можно выбрать требуемый параметр, чтобы предотвратить нежелательное выполнение вредоносного кода, который может содержаться в макросах, полученных из неизвестных источников.

Выбрав слева в окне Центр управления безопасностью категорию Надежные расположения и нажав справа кнопку Добавить новое расположение, можно указать место, откуда файлы, содержащие код VBA, будут открываться без блокирования соответствующих действий (рис.).



**ЛИТЕРАТУРА:** [\[4\]](#), [\[6\]](#), [\[10\]](#), [\[15\]](#), [\[27\]](#)

## **ЗАНЯТИЯ 41-42. АЛГОРИТМИЧЕСКИЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ. СРЕДА ПРОГРАММИРОВАНИЯ VISUAL BASIC FOR APPLICATION**

*План*

- 1. Понятие и свойства алгоритма*
- 2. Способы записи алгоритмов*
- 3. Языки программирования*

Процесс решения задачи на ЭВМ состоит из восьми этапов:

- 1) постановка задачи;*
- 2) математическое описание задачи;*
- 3) алгоритмизация задачи;*
- 4) программирование;*
- 5) разработка тестовой задачи;*

- 6) *перенос программы на машинные носители;*
- 7) *отладка программы;*
- 8) *получение и анализ результатов.*

Рассмотрим более подробно этап алгоритмизации задачи.

### **Понятие и основные свойства алгоритма**

Слово *алгоритм* происходит от *algorithm* – латинского написания имени *аль-Хорезми*, под которым в средневековой Европе знали величайшего математика из Хорезма (город в современном Узбекистане) Мухаммеда бен Мусу, жившего в 783-850 гг., который сформулировал правила выполнения четырех арифметических действий над многозначными числами.

*Алгоритм – это конечная последовательность однозначных предписаний, исполнение которых позволяет с помощью конечного числа шагов получить решение задачи, однозначно определяемое исходными данными.*

#### Свойства алгоритма:

1. *Дискретность.* Это свойство состоит в том, что алгоритм должен представлять процесс решения задачи как последовательность простых шагов. При этом для выполнения каждого шага алгоритма требуется некоторый конечный отрезок времени, т. е. преобразование исходных данных в результат осуществляется во времени дискретно;

2. *Определенность.* Каждая команда алгоритма должна быть четкой, однозначной и не оставлять места для произвола;

3. *Результативность.* Алгоритм должен приводить к решению поставленной задачи за конечное число шагов;

4. *Массовость (универсальность).* Алгоритм решения задачи разрабатывается не для одной конкретной задачи, а для целого класса однотипных задач, различающихся лишь исходными данными.

5. *Однозначность*. Правила и порядок выполнения действий алгоритма имеют единственное толкование.

6. *Конечность* - каждое из действий и весь алгоритм в целом обязательно завершаются.

Алгоритм может быть предназначен для выполнения его человеком или автоматическим устройством, называемым *формальным исполнителем*.

Каждый алгоритм создается в расчете на вполне конкретного исполнителя. Те действия, которые может совершать исполнитель, называются *допустимыми действиями*. Совокупность допустимых действий образует систему команд исполнителя. Алгоритм должен содержать только те действия, которые допустимы для данного исполнителя.

Объекты, над которыми исполнитель может совершать действия, образуют так называемую *среду исполнителя*.

Алгоритм считается *правильным*, если его выполнение дает правильный результат. Соответственно алгоритм *содержит ошибки*, если можно указать такие допустимые исходные данные или условия, при которых выполнение алгоритма либо не завершится вообще, либо не будет получено никаких результатов, либо полученные результаты окажутся неправильными.

Выделяют три крупных класса алгоритмов:

- **вычислительные** алгоритмы, работающие со сравнительно простыми видами данных, такими как числа и матрицы, хотя сам процесс вычисления может быть долгим и сложным;
- **информационные** алгоритмы, представляющие собой набор сравнительно простых процедур, работающих с большими объемами информации (алгоритмы баз данных);

– **управляющие** алгоритмы, генерирующие различные управляющие воздействия на основе данных, полученных от внешних процессов, которыми алгоритмы управляют.

### **Способы записи алгоритмов**

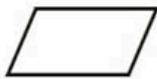
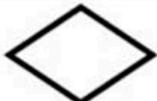
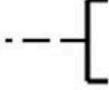
Существуют следующие способы записи алгоритма:

1. *Вербальный*, когда алгоритм описывается на человеческом языке.
2. *Символьный*, когда алгоритм описывается с помощью набора символов.
3. *Графическое описание в виде блок-схемы* (набор связанных между собой геометрических фигур). Для обозначения шагов решения в виде схемы алгоритма используются специальные обозначения (символы). Общепринятыми способами записи являются графическая запись с помощью блок-схем и символьная запись с помощью какого-либо алгоритмического языка.

Описание алгоритма с помощью блок схем осуществляется рисованием последовательности геометрических фигур, каждая из которых подразумевает выполнение определенного действия алгоритма. Порядок выполнения действий указывается стрелками. Написание алгоритмов с помощью блок-схем регламентируется ГОСТом. Внешний вид основных блоков, применяемых при написании блок схем, приведен в таблице.

*Таблица.*

### **Символы для создания блок-схем алгоритмов**

Символ	Название	Назначение
	Данные	Общее обозначение ввода или вывода данных
	Процесс	Обработка данных, операция или группа операций
	Соединитель	Соединение прерванных линий потока
	Предопределенный процесс	Вычисления по подпрограмме (модулю)
	Подготовка	Осуществляет задание изменений параметров цикла
	Решение	Проверка условия
	Терминатор	Вход или выход во внешнюю среду
	Комментарий	Для записи пояснений к алгоритму

4. *Описание на каком-либо языке программирования (программа).*

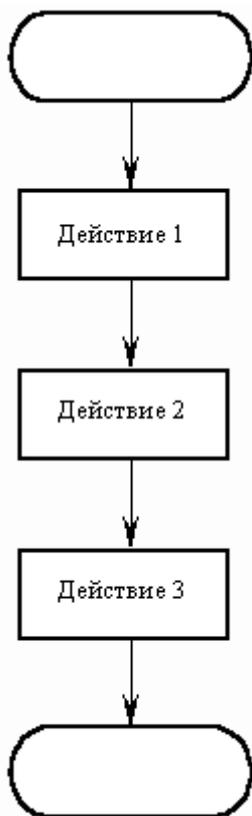
*Программа* – это набор машинных команд, который следует выполнить компьютеру для реализации того или иного алгоритма. *Программа* – это форма представления алгоритма для исполнения его машиной.

### **Разновидности структур алгоритмов**

По структуре алгоритмы разделяют на *линейные, разветвляющиеся и циклические.*

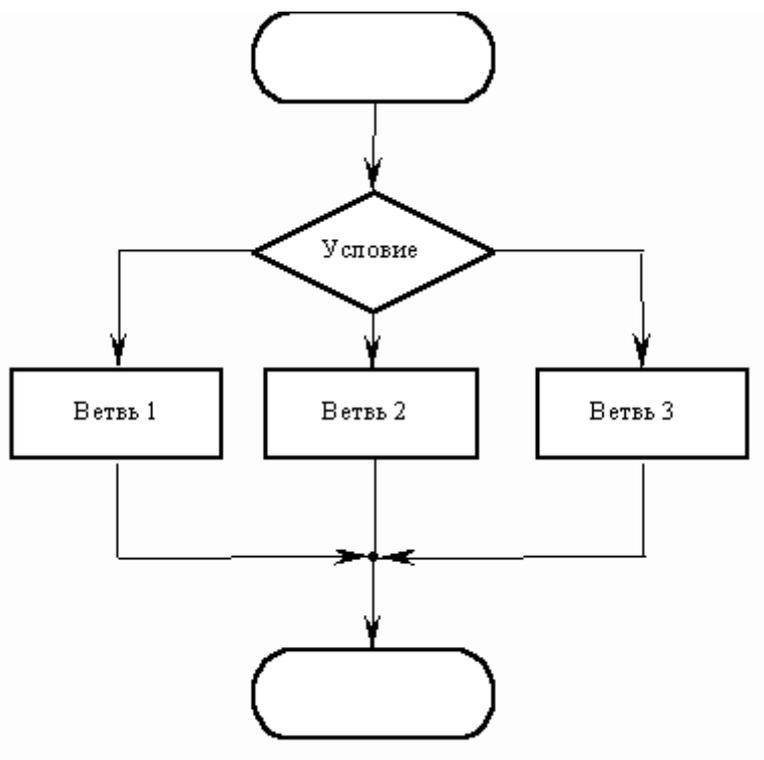
### **Линейные алгоритмы**

*Линейными* называют алгоритмы, в которых операции выполняются последовательно одна за другой, в естественном и единственном порядке следования. В таких алгоритмах все блоки имеют последовательное соединение логической связью передачи информационных потоков. В них могут использоваться все блоки, за исключением блоков проверки условия и модификации. Линейные алгоритмы, как правило, являются составной частью любого алгоритмического процесса.



### **Разветвляющиеся алгоритмы**

При составлении схем алгоритмов часто возникает необходимость проведения анализа исходных данных или промежуточных результатов вычислений и определения дальнейшего порядка выполнения вычислительного процесса в зависимости от результатов этого анализа. *Алгоритмы, в которых в зависимости от выполнения некоторого логического условия происходит разветвление вычислений по одному из нескольких возможных направлений, называют **разветвляющимися**.* Подобные алгоритмы предусматривают выбор одного из альтернативных путей продолжения вычислений. Каждое возможное направление вычислений называется *ветвью*. Логическое условие называют простым, если разветвляющийся процесс имеет две ветви, и сложным, если процесс разветвляется на три и более ветви. Любое сложное логическое условие может быть представлено в виде нескольких простых.



## Циклические алгоритмы

Алгоритм *циклической* структуры предусматривает многократное повторение действий в одной и той же последовательности по одним и тем же математическим зависимостям, но при разных значениях некоторой специально изменяемой величины. Циклические алгоритмы позволяют существенно сократить объем программы за счет многократного выполнения группы повторяющихся вычислений, так называемого *тела цикла*. Специально изменяемый по заданному закону параметр, входящий в тело цикла, называется *переменной цикла*. Переменная цикла используется для подготовки очередного повторения цикла и отслеживания условий его окончания. В качестве переменной цикла используют любые переменные, индексы массивов, аргументы вычисляемых функций и тому подобные величины. Во время выполнения тела цикла параметры переменной цикла изменяются в интервале от начального до конечного значения с заданным шагом. Следовательно, при организации циклических вычислений необходимо предусмотреть задание начального значения переменной цикла,

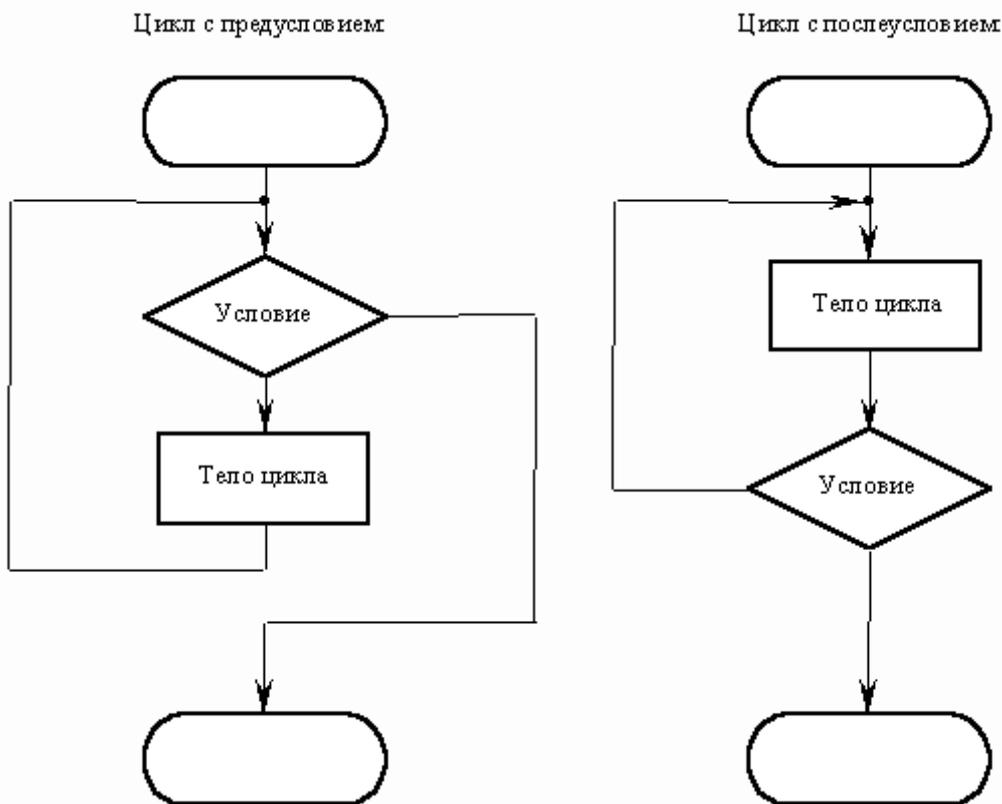
закон ее изменения перед каждым новым повторением и ее конечное значение, при достижении которого произойдет завершение цикла.

Циклы, в теле которых нет разветвлений и других встроенных в них циклов, называют *простыми*. В противном случае их относят к *сложным*.

Циклические алгоритмы разделяют на *детерминированные* и *итерационные*. Циклы, в которых число повторений заранее известно из исходных данных или определено в ходе решения задачи, называют **детерминированными**. Для организации детерминированных циклов наиболее целесообразно использовать блок модификации, внутри которого указывается переменная цикла, ее начальное и конечное значения, а также шаг ее изменения (если шаг изменения равен 1, то его допускается не указывать). Организовать подобный цикл возможно и при использовании блока проверки условия вместо блока модификации, однако при этом несколько усложняется алгоритм и теряется его рациональность.

*Циклы, в которых число повторений неизвестно из исходных данных не определено по ходу решения задачи, называют **итерационными***. В итерационных циклах для организации выхода из тела цикла предусматривается проверка некоторого заранее заданного условия, для чего используют блок проверки условия. В итерационных циклах невозможно использовать блоки модификации, так как при организации таких циклов заранее неизвестно количество изменений переменной цикла и ее конечное значение. В этом случае одно повторение цикла называется **итерацией**.

Различают циклы с **предусловием** и **постусловием**:



## ЯЗЫКИ ПРОГРАММИРОВАНИЯ

На практике в качестве исполнителей алгоритмов используются компьютеры. Для того, чтобы ЭВМ могла выполнять программу, программа должна быть записана по строгим правилам в виде, доступном для обработки на ЭВМ. Программа для такой машины записывается на так называемом **машинном языке**, т. е. *представляет собой последовательность двоичных чисел*. Придумывать и записывать программу на машинном языке достаточно неудобно.

Поэтому возникла идея записывать программу на так называемом **алгоритмическом языке** или **языке программирования**. **Языки программирования** – *специально разработанные искусственные языки, предназначенные исключительно для записи алгоритмов, исполнение которых поручается компьютеру*.

Алгоритмический язык (как и любой другой язык), образуют три его составляющие: алфавит, синтаксис и семантика.

**Алфавит** – фиксированный для данного языка набор символов (букв, цифр, специальных знаков и т.д.), которые могут быть использованы при написании программы.

**Синтаксис** - правила построения из символов алфавита специальных конструкций, с помощью которых составляется алгоритм.

**Семантика** - система правил толкования конструкций языка. Таким образом, **программа** составляется с помощью соединения символов алфавита в соответствии с синтаксическими правилами и с учетом правил семантики.

**Язык программирования** – это набор символов и терминов, который в соответствии с правилами синтаксиса описывает алгоритм решения задачи.

### **Краткая история и классификация языков программирования**

Первые языки программирования были очень примитивными и мало чем отличались от формализованных упорядоченных последовательностей единиц и нулей, понятных компьютеру. Использование таких языков было крайне неудобно с точки зрения программиста, так как он должен был знать числовые коды всех машинных команд, должен был сам распределять память под команды программы и данные.

Для того, чтобы облегчить общение человека с ЭВМ были созданы языки программирования типа Ассемблер. Переменные величины стали изображаться символическими именами. Числовые коды операций заменились на мнемонические обозначения, которые легче запомнить. Язык программирования приблизился к человеческому языку, и отдалился от языка машинных команд.

Один из первых языков программирования – **Фортран** (**Formula Translation**) был создан в середине 50-х годов. Благодаря своей простоте и тому, что на этом языке накоплены большие библиотеки программ Фортран и в наши дни остается одним из самых распространенных. Он используется для инженерных и научных расчетов, для решения задач физики и других наук с развитым математическим аппаратом.

Для решения экономических задач был создан язык программирования - **Кобол**.

Расширение областей применения ЭВМ влечет за собой создание языков, ориентированных на новые сферы применения: **Снобол** – алгоритмический язык для обработки текстовой информации, **Лисп** - алгоритмический язык для обработки символов. Лисп находит широкое применение в исследованиях по созданию искусственного интеллекта.

В 1968 г. был объявлен конкурс на лучший язык программирования для обучения студентов. Победителем стал язык **Алгол-68**, но широкого распространения не получил. Для этого конкурса Никлаус Вирт создал язык **Паскаль**, достаточно простой, удобный, с наличием мощных средств структурирования данных. Хотя Паскаль был разработан как язык для обучения программированию, он впоследствии получил широкое развитие и в настоящее время считается одним из самых используемых языков. Для обучения младших школьников Самуэлем Пайпертом был разработан язык **Лого**. Он отличается простотой и богатыми возможностями.

Широкое распространение в школах в качестве обучающего языка получил язык **Бейсик**, позволяющий взаимодействовать с ЭВМ в режиме непосредственного диалога. Спустя много лет после изобретения Бейсика, он и сегодня самый простой для освоения из десятков языков общецелевого программирования.

Необходимость разработки больших программ, управляющих работой ЭВМ, потребовала создания специального языка программирования **СИ** в

начале 70-х г. Он является одним из универсальных языков программирования. В отличие от Паскаля, в нем заложены возможности непосредственного обращения к некоторым машинным командам и к определенным участкам памяти компьютера. *Си широко используется как инструментальный язык для разработки операционных систем, трансляторов, баз данных и других системных и прикладных программ.* Си – это язык программирования общего назначения, хорошо известный своей эффективностью, экономичностью, и переносимостью. Во многих случаях программы, написанные на Си, сравнимы по скорости с программами, написанными на языке Ассемблера. При этом они имеют лучшую наглядность и их более просто сопровождать. Си сочетает эффективность и мощьность в относительно малом по размеру языке.

Появление функционального программирования привело к созданию языка **Пролог**. Этот язык программирования разрабатывался для задач анализа и понимания естественных языков на основе языка формальной логики и методов автоматического доказательства теорем.

В 80-х г. 20 века был создан язык **Ада**. Этот язык в дополнение к классическим свойствам, обеспечивает программирование задач реального времени и моделирование параллельного решения задач.

Существуют различные **классификации языков программирования**. По наиболее распространенной классификации все языки программирования делят на языки **низкого, высокого и сверхвысокого** уровня.

В группу языков **низкого уровня** входят машинные языки и языки символического кодирования: (**Автокод, Ассемблер**). Операторы этого языка – это те же машинные команды, но записанные мнемоническими кодами, а в качестве операндов используются не конкретные адреса, а символические имена. *Все языки низкого уровня ориентированы на определенный тип компьютера, т. е. являются машинно-зависимыми.* **Машинно-ориентированные языки** – это языки, наборы операторов и

*изобразительные средства которых существенно зависят от особенностей ЭВМ (внутреннего языка, структуры памяти и т.д.).*

Следующую, существенно более многочисленную группу составляют языки программирования **высокого уровня**. Это **Фортран, Алгол, Кобол, Паскаль, Бейсик, Си, Пролог и т.д.** *Эти языки машинно-независимы, т.к. они ориентированы не на систему команд той или иной ЭВМ, а на систему операндов, характерных для записи определенного класса алгоритмов.* Однако программы, написанные на языках высокого уровня, занимают больше памяти и медленнее выполняются, чем программы на машинных языках.

К языкам **сверхвысокого уровня** можно отнести лишь **Алгол-68 и АPL**. Повышение уровня этих языков произошло за счет введения сверхмощных операций и операторов.

Другая классификация делит языки на **вычислительные** и языки **символьной обработки**. К первому типу относят Фортран, Паскаль, Алгол, Бейсик, Си, ко второму типу - Лисп, Пролог, Снобол и др.

В современной информатике можно выделить два основных направления развития языков программирования: **процедурное** и **непроцедурное**.

Процедурное программирование возникло на заре вычислительной техники и получило широкое распространение. В процедурных языках программа явно описывает действия, которые необходимо выполнить, а результат задается только способом получения его при помощи некоторой процедуры, которая представляет собой определенную последовательность действий.

Среди процедурных языков выделяют в свою очередь **структурные** и **операционные языки**. В структурных языках одним оператором записываются целые алгоритмические структуры: ветвления, циклы и т.д. В операционных языках для этого используются несколько операций. Широко

распространены следующие структурные языки: Паскаль, Си, Ада, ПЛ/1. Среди операционных известны Фортран, Бейсик, Фокал.

**Непроцедурное (декларативное)** программирование появилось в начале 70-х годов 20 века, но стремительное его развитие началось в 80-е годы, когда был разработан японский проект создания ЭВМ пятого поколения, целью которого явилась подготовка почвы для создания интеллектуальных машин. К непроцедурному программированию относятся **функциональные и логические языки**.

В **функциональных языках** программа описывает вычисление некоторой функции. Обычно эта функция задается как композиция других, более простых, те в свою очередь разлагаются на еще более простые и т.д. Один из основных элементов в функциональных языках - **рекурсия**, *то есть вычисление значения функции через значение этой же функции от других элементов*. Присваивания и циклов в классических функциональных языках нет.

В **логических языках** программа вообще не описывает действий. Она задает данные и соотношения между ними. После этого системе можно задавать вопросы. Машина перебирает известные и заданные в программе данные и находит ответ на вопрос. Порядок перебора не описывается в программе, а неявно задается самим языком. Классическим языком логического программирования считается Пролог. Построение логической программы вообще не требует алгоритмического мышления, программа описывает статические отношения объектов, а динамика находится в механизме перебора и скрыта от программиста.

Можно выделить еще один класс языков программирования - **объектно-ориентированные языки высокого уровня**. На таких языках не описывают подробной последовательности действий для решения задачи, хотя они содержат элементы процедурного программирования. Объектно-ориентированные языки, благодаря богатому пользовательскому интерфейсу,

предлагают человеку решить задачу в удобной для него форме. Примером такого языка может служить язык программирования визуального общения Object Pascal.

**Языки описания сценариев**, такие как Perl, Python, Rexx, Tcl и языки оболочек UNIX, предполагают стиль программирования, весьма отличный от характерного для языков системного уровня. Они предназначаются не для написания приложения с нуля, а для комбинирования компонентов, набор которых создается заранее при помощи других языков. Развитие и рост популярности Internet также способствовали распространению языков описания сценариев. Так, для написания сценариев широко употребляется язык Perl, а среди разработчиков Web-страниц популярен JavaScript.

**ЛИТЕРАТУРА:** [\[4\]](#), [\[6\]](#), [\[10\]](#), [\[15\]](#), [\[27\]](#)

## **ЗАНЯТИЯ 43-44. ПЕРЕМЕННЫЕ, ТИПЫ ПЕРЕМЕННЫХ, ФУНКЦИИ. ОПЕРАТОРЫ**

*План*

- 1. Основные понятия*
- 2. Операции*
- 3. Переменные*

Основными понятиями в алгоритмических языках являются следующие.

- 1. Имена (идентификаторы)** - *последовательность символов для обозначения объектов программы (переменных, массивов, функций и др.).*

**2. Операции.** Существуют следующие типы операций:

– **арифметические операции:** сложение, обозначается символом “+”; вычитание, обозначается символом “-”; умножение, обозначается символом “\*”; деление, обозначается символом “/” и др. ;

– **логические операции:** операции “логическое и”, “логическое или”, “логическое не” и др.;

– **операции отношения:** меньше, обозначается символом “<”; больше, обозначается символом “>”; меньше или равно, обозначается символами “<=”; больше или равно, обозначается символами “>=”; равно, обозначается символом “=”; не равно, обозначается символами “<>”.

– **операция конкатенации** символьных значений друг с другом, изображается знаком "+".

**3. Ключевые слова** – это слова языка, имеющие строго определенное назначение, которые не могут использоваться в качестве идентификаторов.

**4. Данные** - величины, обрабатываемые программой. Имеется три основных вида данных: **константы, переменные и массивы.**

**5. Константы** - это данные, которые зафиксированы в тексте программы и не изменяются в процессе ее выполнения.

Примеры констант:

**числовые:** 7.5, 12;

**логические:** true (истина), false (ложь);

**символьные:** "A", "+";

**строковые:** "abcde", "информатика".

**6. Переменные** – это данные, которые могут изменять свои значения в ходе выполнения программы. Они обозначаются именами.

Переменные бывают **целые, вещественные, логические, символьные и строковые.**

**7. Массивы** - последовательности однотипных элементов, число которых фиксировано и которым присвоено одно имя. Положение элемента в массиве однозначно определяется его индексами - одним в случае одномерного массива, или несколькими, если массив многомерный.

**8. Выражения** – элементы языка, которые предназначаются для выполнения необходимых вычислений, состоят из констант, переменных, указателей функций, объединенных знаками операций. Выражения записываются в виде линейных последовательностей символов (без подстрочных и надстрочных символов, "многоэтажных" дробей и т. д.), что позволяет вводить их в компьютер, последовательно нажимая на соответствующие клавиши клавиатуры.

Различают выражения **арифметические, логические и строковые.**

**Арифметические выражения** служат для определения одного числового значения. Арифметические выражения записываются по следующим правилам:

1. Нельзя опускать знак умножения между сомножителями и ставить рядом два знака операций.
2. Индексы элементов массивов записываются в скобках.
3. Операции выполняются в порядке старшинства: сначала вычисление функций, затем возведение в степень, потом умножение и деление и в последнюю очередь - сложение и вычитание.
4. Операции одного старшинства выполняются слева направо.

**Логические выражения** описывают некоторые условия, которые могут удовлетворяться или не удовлетворяться. Таким образом, логическое выражение может принимать только два значения - "истина" или "ложь" (да или нет).

В записи логических выражений помимо арифметических операций сложения, вычитания, умножения, деления и возведения в степень используются операции отношения и логические операции.

Значения **строковых выражений** - *тексты*. В них могут входить строковые константы, строковые переменные и строковые функции, разделенные знаком операции конкатенации.

**9. Оператор** – это элемент языка, который задает полное описание некоторого действия, которое необходимо выполнить. Оператор - это наиболее крупное и содержательное понятие языка: каждый оператор представляет собой законченную фразу языка программирования и определяет некоторый вполне законченный этап обработки данных. В состав операторов входят ключевые слова; данные; выражения и т.д.

**10. Стандартная функция** – подпрограмма, заранее встроенная в транслятор языка для вычисления часто употребляемых функций. В качестве аргументов функций можно использовать константы, переменные и выражения.

**11. Программа** - это последовательность инструкций, предназначенных для выполнения компьютером. В настоящее время программы оформляются в виде текста, который записывается в файлы. **Программа на языке программирования** представляет собой совокупность операторов, записанных в соответствии с принятым синтаксисом.

**12. Программирование** – это теоретическая и практическая деятельность решения задачи средствами конкретного языка

программирования и оформления полученных результатов в виде программы. **Программирование (programming)** – это процесс создания последовательности действий (операций), проводимый в целях достижения требуемого результата. Процесс программирования состоит из стадий: формулирования, разработки программы, включая кодирование и тестирование, и, далее, создания новых версий.

На стадии программирования возникает этап **отладки программы** – процесс обнаружения и устранения ошибок в программе, производимой по результатам ее тестирования на компьютере.

После окончательной отладки **программа документируется**, т.е. к ней прилагается описание назначения программы и инструкция по эксплуатации. Только после этого программа становится законченным программным продуктом, подготовленным к реализации как любой иной вид промышленной продукции.

Языки высокого уровня работают через трансляционные программы - **трансляторы**, которые преобразуют исходный код в последовательность команд машинного языка. Существует два основных вида трансляторов: **интерпретаторы**, которые сканируют и проверяют исходный код в один шаг, и **компиляторы**, которые сканируют исходный код для создания текста программы на машинном языке, которая затем выполняется отдельно.

В общем случае программа может иметь **модульную структуру**, т.е. состоять из нескольких программных единиц, связанных между собой командами передачи управления. Такой принцип построения программ называется **модульным**. Программная единица, с первой команды которой начинается выполнение программы, называется головной программой. Остальные программные единицы, входящие в единую программу, называются **подпрограммами**.

**13. Подпрограмма** - это последовательность операторов, которые определены и записаны только в одном месте программы, однако их можно вызвать для выполнения из одной или нескольких точек программы.

**14. Функция** - это программная единица, которая может быть употреблена в выражении. Функция прямо возвращает величину, которая используется при вычислении этого выражения, и, кроме того, может возвращать величины через параметры.

Процесс разработки многомодульных программ эффективнее, особенно если разрабатывается программа большого размера, когда над реализацией проекта может работать несколько программистов, каждый из которых имеет возможность модифицировать фрагменты программы, не мешая работе остальных.

Подпрограммы и функции позволяют создавать большие структурированные программы, которые можно делить на части. Это дает преимущества в следующих ситуациях:

1. Если программа большая, разделение ее на части облегчает создание, тестирование и ее сборку.
2. Если программа большая и повторная компиляция всего исходного текста занимает много времени, разделение ее на части экономит время компиляции.
3. Если процедуру надо использовать в разных случаях разным образом, можно записать ее в отдельный файл и скомпилировать отдельно.

### **Инструментальные системы программирования**

Для популярных языков программирования на ЭВМ существует множество систем программирования. Программисты предпочитают те системы, которые легки в использовании, позволяют получить эффективные программы, имеют богатые библиотеки функций (подпрограмм) и мощные

возможности для отладки разрабатываемых программ. В качестве примеров таких систем программирования можно назвать Delphi, Visual C++, Visual Basic.

Системы программирования прежде всего различаются по тому, какой язык программирования они реализуют. Среди программистов, пишущих программы для персональных компьютеров, наибольшей популярностью пользуются языки Си, Паскаль и Бейсик.

## СТРУКТУРА ПРОГРАММЫ

В среде программирования VBA реализован алгоритмический язык высокого уровня Visual Basic. В алгоритмическом языке программа состоит из операторов. В языках высокого уровня операторы записываются с помощью ключевых слов, имен переменных и знаков действий. В Visual Basic каждый оператор записывается в отдельной строке. Как исключение, несколько операторов можно записывать в одной строке. Тогда их разделяют двоеточием (:).

Количество пропусков и знаков табуляции между отдельными частями оператора не имеет значения.

В тексте программы после апострофа ( ' ) можно записывать **комментарий** - *текст, поясняющий операцию и не является частью программы*.

## ПЕРЕМЕННЫЕ

Переменные используются в программе для хранения данных.

**Переменные** - *это величины, которые могут принимать различные значения*. Переменные различают по именам.

**Имя переменной** - *это совокупность букв, цифр и других символов, начинающийся с буквы и записанных без пробелов*.

Существуют имена, которые нельзя давать переменным. Это - ключевые слова, используемые в языке программирования для написания операторов программы. Среда VBA настроена так, что в тексте программы в окне кода ключевые слова изображаются другим цветом.

В языке Visual Basic для написания имени переменной можно использовать только латинские буквы, цифры и некоторые специальные знаки. В среде VBA для записи имени переменной рядом с английскими буквами можно использовать буквы национального алфавита, который установлен на компьютере. Первым символом имени должна быть буква. При записи имени переменной большие и малые буквы не различаются. В имени переменной нельзя использовать пробелы.

### **ТИП ПЕРЕМЕННОЙ**

Переменные бывают разных типов в зависимости от типа значений, которые они могут принимать (числовые, текстовые и другие) и диапазона их значений (Таблица 3.1).

В языке Visual Basic можно использовать неопределенный тип **Variant**. В переменной этого типа можно правильно хранить и числа, и текст, и дату. Однако она занимает много места в памяти - 16 байт плюс один байт на каждый символ.

В языке Visual Basic тип переменной обычно описывается оператором

#### **Dim переменная As тип**

Можно сразу описывать несколько переменных. К примеру,

#### **Dim Цена As Double Стоимость As Double**

В описании переменной тип можно не указывать, то есть писать:

#### **Dim переменная**

Такая переменная будет иметь тип **Variant** до тех пор, пока ей не будет предоставлено определенное значение. После этого тип переменной будет тип предоставленного значения.

В программе можно использовать описания типа

### **Dim sigma, Цена, Стоимость As Double**

При этом только переменная **Стоимость** будет тип **Double**. Другие переменные будут иметь тип **Variant** до тех пор, пока им НЕ будет присвоено значение.

В алгоритмическом языке Basic можно запретить использовать переменную без описания. Для этого перед описаниями следует записать оператор **Option Explicit**.

Использование этого оператора позволяет избежать ошибок, связанных с неправильной записью переменных и использовании их не по назначению.

Система программирования VBA может автоматически включать оператор **Option Explicit** в код программы. Для этого должна быть включена опция **Require Variable Declaration** на странице **Editor** команды **Tools / Options** главного меню среды VBA.

## **ЧИСЛА**

В математике числа могут быть целыми и дробными, рациональными и иррациональными. Совокупность целых и дробных чисел образует множество действительных чисел. В языке Visual Basic и в других алгоритмических языках числа классифицируются иначе - они отличаются способом записи в памяти: точно или приближенно.

Все числа, которые используются для вычислений, и целые, и дробные, обычно записываются в компьютере *приближенно*. Числа, которые записываются приближенно, в языке Visual Basic называются *действительными*.

Есть задачи, для которых необходимо точно записывать целые числа. Например, если нужно выполнить какое-то действие определенное число раз. Числа, которые записываются точно, называются *целыми*.

В соответствии с этим, переменные для сохранения числовых значений делятся на *целые и действительные*.

В языке Visual Basic переменная целого типа используется для точной записи чисел, а переменная вещественного типа – для приближенного.

## ТЕКСТ

Значением символьной переменной может быть текст последовательность любых символов, которые воспринимает компьютер. *Количество символов при этом называется длиной символьной переменной*. В языке Visual Basic длина символьной переменной может меняться в процессе выполнения программы и может принимать значение от 0 до  $2^{16}-1 = 32767$ .

## ЧИСЛОВЫЕ И ТЕКСТОВЫЕ КОНСТАНТЫ

В тексте программ, кроме операторов и переменных, используют **числовые и текстовые константы**. Целые числа записываются с помощью цифр и знаков, например: 7, +7, 17. Знак (+) перед числом можно не писать.

Для действительных чисел используют две формы записи:

1. *С десятичной точкой*. Например: 0.5, -12.5;
2. *В экспоненциальной форме*. Например, запись 9.3 E-3 означает число  $9.3 \times 10^{-3}$  или 0.0093.

Если целая часть числа равна нулю, то ее можно опускать. Например, число 0.5 можно записать в виде .5, а 0.3 E-2 - .3 E-2.

Текстовая константа - это текст в кавычках. Например: "Его зовут Иван".

Константы могут иметь имена. Константа с именем описывается оператором:

**Const имя = значение**

## **ОПЕРАТОР ПРИСВАИВАНИЯ**

*Переменной можно придать значение с помощью оператора присваивания, который имеет вид*

**переменная = константа**

или

**переменная = выражение**

**Выражение** - это *формула в алгоритмическом языке*. Выражение строится из переменных и констант, которые соединяются знаками действий и могут использовать различные функции. Для определения порядка действий в выражении можно использовать круглые скобки.

Оператор присваивания выполняется так: *вычисляется выражение в правой части оператора, а затем полученное значение предоставляется переменной*. Старое значение переменной при этом исчезает.

Для записи оператора присваивания в языке Visual Basic используется знак равенства. Однако этот знак в Visual Basic имеет иной смысл, чем в математике:

В математике уравнения  $x = x + 1$  не имеет решений. В программировании, если переменная  $x$  имеет значение 1, то после выполнения оператора  $x = x + 1$  эта переменная будет иметь значение 2.

Над числами выполняются арифметические действия с учетом приоритета действий в следующем порядке:

1. Возведение в степень (^)
2. Умножение (\*) и деление - обычное (/) и целое (\)
3. Остаток от деления (Mod)

#### 4. Сложение (+) и вычитание (-).

Для текстовых строк выполняется операция объединения, которая также обозначается знаком (+). Если типы данных разные, то для этого используют операцию (&).

**ЛИТЕРАТУРА:** [\[4\]](#), [\[6\]](#), [\[10\]](#), [\[15\]](#), [\[27\]](#)

### **ЗАНЯТИЯ 45-46. ЛИНЕЙНЫЕ ПРОГРАММЫ**

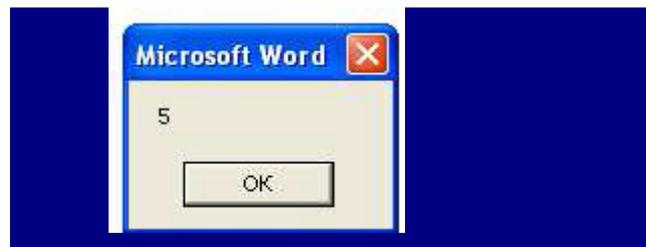
*План*

1. Составление программы для вычисления суммы двух чисел
2. Отладка программы

#### **Программа 1.**

*Эта программа вычисляет и выводит сумму чисел 2 и 3.*

```
Sub Progl ()  
Dim a, b, c As Integer  
a = 2: b = 3: c = a + b  
MsgBox c  
End Sub
```

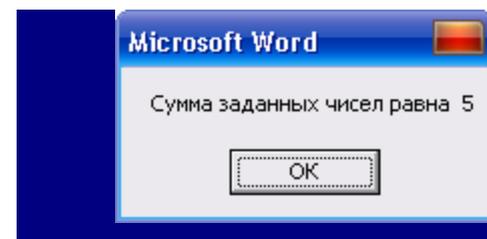


В одной строке программы можно записывать несколько операторов. Тогда их разделяют двоеточием (:). В приведенной программе непонятно какой, результат выводится. Это можно исправить, если сформировать строку вывода так, как в программе:

```

Sub Prog2 ()
  Dim a, b, c As Integer
  Dim Text As String
  a = 2: b = 3: c = a + b
  Text = "Сумма заданных чисел равна" & c
  MsgBox Text
End Sub

```



Общий недостаток обеих программ заключается в том, что они складывают два конкретных числа. Следующая программа находит сумму двух любых целых чисел.

```

Sub Prog3 ()
  Dim a, b, c As Integer
  Dim Text As String
  Text = "Введите первое слагаемое"
  a = Val (InputBox (Text))
  Text = "Введите второе слагаемое"
  b = Val (InputBox (Text))
  c = a + b
  Text = "Сумма заданных чисел" & a & " и " & b & " равна" & c
  MsgBox Text
End Sub

```

В этой программе используется функция **InputBox** ввода данных с помощью диалогового окна (рис. 3.1). Значением функции **InputBox** является строка текста.

Когда оператор **Text =** не помещается в одной строке, используют знак переноса - подчеркивание (  ), который отделяется от других символов по крайней мере одним пробелом. Это означает, что текст на следующей строке

является продолжением данного оператора. Таким образом можно записать оператор, занимающий несколько строк текста программы.

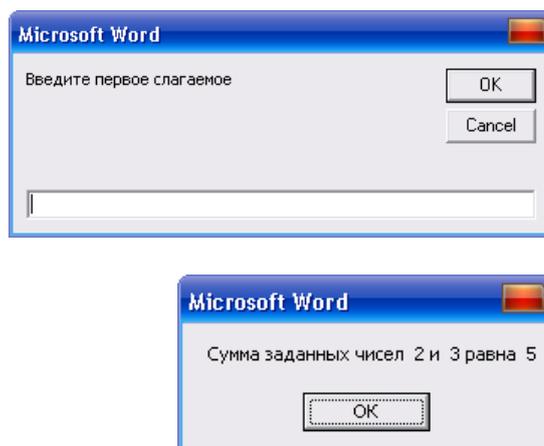


Рис. 3.1 Диалоговые окна ввода и вывода данных

Для преобразования текста в число используют функцию **Val**. Эта функция может использовать в тексте только его часть до появления нецифровых данных. Например, Val ("1615 198th Street N.E. ") даст 1615198.

Функцию **Val** нельзя использовать для ввода дробей с десятичной запятой, а не точкой. В этом случае надо использовать функции преобразования типов данных из следующей Таблицы 3.3.

Таблица 3.3

Функции преобразования типов

Функция	Тип результата
Cint	Integer
CLng	Long
CByte	Byte
CSng	Single
CDbl	Double
CStr	String
CDate	Date

В диалоговых окнах ввода и вывода данных можно использовать заголовки, которые объясняют действия (рис. 3.2).

Как в программе:

```
Sub Prog4 ()  
Dim a, b, c As Integer  
Dim Title, Text As String  
Title = "Сложение чисел"  
Text = "Введите первое слагаемое"  
a = CInt (InputBox (Text, Title))  
Text = "Введите второе слагаемое"  
b = CInt (InputBox (Text, Title))  
c = a + b  
Title = "Результат"  
Text = "Сумм заданных чисел" & a & "и" & b & "равна" & c  
Response = MsgBox (Text,, Title)  
End Sub
```

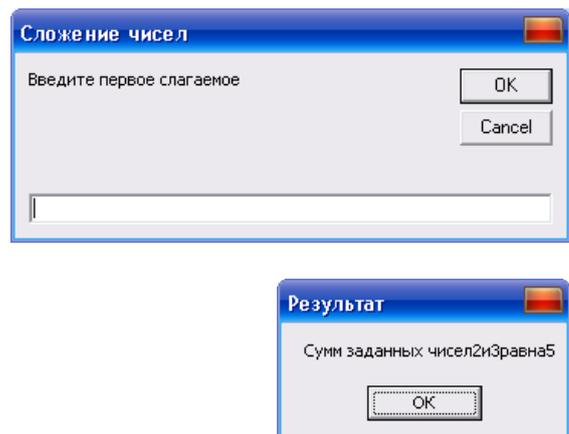


Рис. 3.2 Диалоговые окна программы с заголовками

Обратите внимание на другой способ формирования окна вывода. Здесь уже используется не оператор, а функция **MsgBox**. Ее значение присваивается переменной *Response*. Переменная *Response* не описана в

программе, и поэтому она имеет тип Variant. Значение переменной Response в программе не используется. Она введена только для того, чтобы можно было использовать функцию MsgBox.

Обратите внимание на две запятые в списке параметров у оператора вызова функции MsgBox. Их наличие означает, что функция MsgBox имеет второй параметр, который в нашей программе не используется.

## **Программа 2.**

*Следующая простая программа осуществляет ввод текстовых данных в диалоге.*

Программа запрашивает имя пользователя и здоровается с ним:

```
Sub Hello ()
```

```
Dim name As String
```

```
name = InputBox ("Как Вас зовут?")
```

```
MsgBox "Приветствую Вас" + name + "!"
```

```
End Sub
```

Здесь для объединения фрагментов текста используется знак (+), так как сочетаются элементы одного типа.

Обратите внимание на пробелы, которые используются в константах для формирования правильного текста.

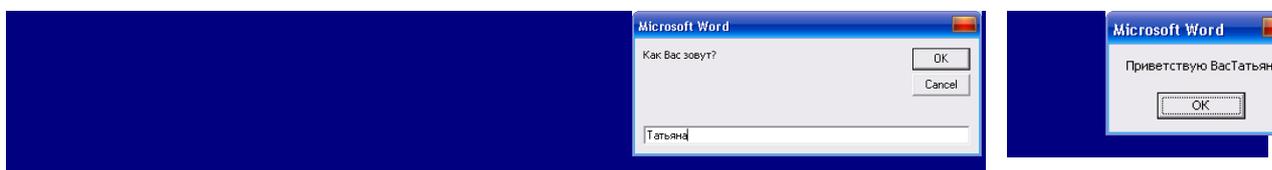


Рис. 3.3 Диалоговые окна программы Hello

## **Программа 3**

Эта программа вычисляет по заданным значениям  $x$  значение

$$y = \frac{x - \sqrt{x^2 + 1}}{x + \frac{1}{x^4 + 1}} e^{\frac{1}{x^2 + 1}} + \sin \frac{\pi}{2} x$$

выражения

```
Sub Formula()
```

```
Dim x, y As Double
```

```
Const pi = 3.14159
```

```
x = Val(InputBox("Введите X = "))
```

```
y = (x - Sqr(x ^ 2 + 1)) / (x + x / (x ^ 4 + 1))
```

```
y = y * Exp(1 / (x ^ 2 + 1))
```

```
y = y + sin(pi / 2 * x)
```

```
MsgBox "y = " & y
```

```
End Sub
```

В таких задачах используются действительные переменные двойной точности типа Double. При вычислении по формулам очень важно правильно построить выражение в формуле.

В программе вычисления по формуле разбиты на три оператора. Это не обязательно, но так легче проверить правильность вычислений. Для того чтобы было удобно записывать и читать формулу, в программе используется константа  $\pi$ .

Операции одного порядка, (+ и -) или (\* и /), выполняются подряд слева направо. Обратите внимание, как в программе записано выражение  $\frac{\pi}{2} x$ .

При вычислении выражений сразу выполняются операции возведения в степень (^), затем умножение и деление (\* и /), а дальше сложение и вычитание.

? Знак умножения в выражениях опускать нельзя!

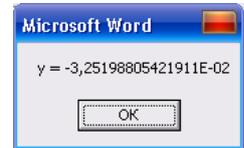
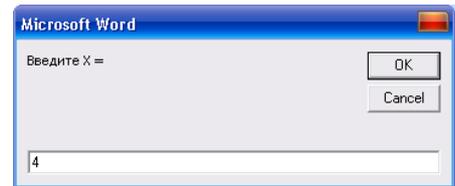


Рис. 3.4 Диалоговые окна программы Formula

ЛИТЕРАТУРА: [4], [6], [10], [15], [27]

## ЗАНЯТИЯ 47-48. ЦИКЛИЧЕСКИЕ АЛГОРИТМЫ. ЦИКЛИЧЕСКИЕ АЛГОРИТМЫ

*План*

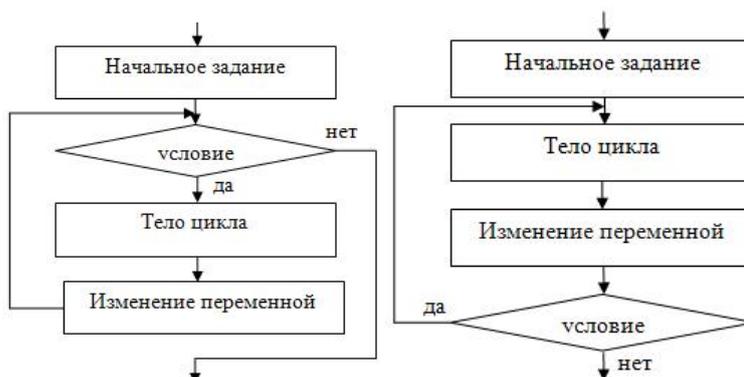
1. *Определение цикла*
2. *Операторы цикла*
3. *Циклические конструкции*

**Циклом** называется часть программы, которая выполняется несколько раз.

Алгоритм с циклом называется **циклическим**.

Повторяющаяся часть программы, называется **телом цикла**.

На рис. приведена логическая схему циклического алгоритма с предусловием (слева) и с постусловием (справа).



Существуют специальные операторы для построения циклических алгоритмов.

### **Оператор цикла Do ... Loop**

Этот оператор применяют, когда число повторений тела цикла заранее неизвестно.

В языке Visual Basic есть два типа оператора Do: *с предусловием и с постусловием.*

Цикл с предусловием проверяет условие выхода из цикла перед выполнением тела цикла:

**Do While** *условие*

*тело цикла*

**Loop**

Если условие верное и принимает значение True, то операторы тела цикла выполняются, а если оно неправильное и принимает значение False, то операторы тела цикла не выполняются, а выполняется оператор программы, который стоит после Loop.

Цикл с постусловием проверяет условие выхода из цикла после выполнения тела цикла:

**Do**

*тело цикла*

**Loop Until** *условие*

Если условие верное и принимает значение True, то работа цикла заканчивается, а если оно неверное и принимает значение False, то цикл повторяется.

Иногда используют другие формы этих операторов: операторы

**Do Until** *условие*

*тело цикла*

**Loop**

а также

**Do**

*тело цикла*

**Loop Until** *условие*

Эти операторы завершают работу цикла, если условие принимает значение True.

Работу оператора цикла можно завершить также с помощью оператора **Exit Do**, который можно поставить внутри тела цикла.

## ПРОГРАММА 1

*Составить программу, которая вводит действительные числа до тех пор, пока не будет введен пустой символ, и находит их среднее значение.*

**Sub СреднееЗнач()**

**Dim a, S As Double**

**Dim xTxt, Title, Text As String**

**Title = "Найти среднее"**

**Text = "Введите следующее число или нажмите ENTER"**

**Text = Text + vbCrLf**

**S = 0**

**n = 0**

**xTxt = InputBox(Text, Title)**

**Do While xTxt <> ""**

**a = CDbI(xTxt)**

**S = n \* S + a**

**Text = Text + " " + xTxt**

**xTxt = InputBox(Text, Title)**

**n = n + 1**

**Loop**

**If n <> 0 Then**

*S = S / n*

*Text = "Среднее значение введенных " & n & " чисел равно " & S*

*Response = MsgBox(Text, , Title)*

*End If*

*End Sub*

## Оператор While ... Wend

Другая, более короткая запись основной формы оператора цикла с предусловием Do While:

**While** *условие*

*тело цикла*

**Wend**

Оператор Do имеет больше возможностей. Однако оператор While проще, короче, и поэтому его часто используют.

## ПРОГРАММА 2

*Составить программу, которая вычисляет сумму цифр данного целого числа.*

Для составления программы используем разработанный алгоритм, имеющий циклическую структуру. Этот цикл заканчивается, когда в результате очередного целого деления получим ноль.

*Sub СуммаЦифр()*

*Dim a, s As Long*

*Dim Title, Text As String*

*Title = "Найти сумму цифр"*

*Text = "Введите любое целое число"*

*x = CLng (InputBox (Text, Title))*

*s = 0*

*a = x*

*While a <> 0*

*s = s + a Mod 10*

*a = a \ 10*

*Wend*

*Text = "Сумма цифр числа" & x & "равна" & s*

*Response = MsgBox (Text,, Title)*

*End Sub*

## Оператор For

Оператор цикла со счетчиком For используют, когда известно количество повторений тела цикла. Это самый популярный оператор цикла.

Оператор цикла со счетчиком имеет вид:

**For** *счетчик = начало To шаг Step конец*

*тело цикла*

**Next**

где

- ✓ счетчик - переменная целого типа,
- ✓ начало, конец, шаг - целые выражения:
  - начало - начальное значение переменной счетчик,
  - конец - конечное значение переменной счетчик,
  - шаг - величина увеличения переменной счетчик при каждом новом повторении тела цикла.

Оператор For работает так:

- 1) Переменной *счетчик* придается значение начало;
- 2) если *счетчик* <= *конец*, то выполняется тело цикла, а иначе тело цикла не выполняется и оператор цикла заканчивает работу;

3) Значение счетчика увеличивается на шаг:

*счетчик = счетчик + шаг*

4) Переходим к пункту 2.

Если шаг = 1, то это значение не задают и используют оператор

**For** *счетчик = начало* **To** *конец*

*тело цикла*

**Next**

После выполнения оператора For значение переменной счетчик не определено.

В теле цикла в операторе со счетчиком For нельзя изменять его параметры счетчик, начало, конец шаг.

### ПРОГРАММА 3

Составить программу для вычисления факториала  $n! = 1 \times 2 \times \dots \times n$ ;  $0! = 1$ .

Алгоритм. Поскольку величина  $n$  известна, для решения этой задачи удобно использовать оператор For:

**Sub** *Fact ()*

**Dim** *i, n* **As Integer**

**Dim** *f* **As Long**

**Dim** *xTxt, Title, Text* **As String**

*Title = "Вычисление факториала N!"*

*Text = "Введите значение N"*

*n = CInt (InputBox (Text, Title))*

*f = 1*

**For** *i = 1* **To** *n*

*f = f \* i*

*Next*

*Text = n & "! =" & F*

*Response = MsgBox (Text,, Title)*

*End Sub*

С ростом  $n$  величина  $n!$  очень быстро растет. Поэтому для результата в программе используется переменная типа Long. Поскольку факториал  $n!$  очень быстро растет, эту программу можно применять для небольших значений  $n$ .

## ПРОГРАММА 4

*Найти все делители данного целого числа.*

Алгоритм. Натуральное число  $n$  называется делителем целого числа  $a$ , если  $a$  делится на  $n$  без остатка, то есть остаток при целом делении  $a$  на  $n$  равен нулю. Очевидно, что всегда делителями целого числа является 1 и само это число, а все остальные делители могут принимать значения от 2 до  $n / 2$ . Простейший алгоритм нахождения делителей заключается в том, чтобы найти остатки от деления  $a$  на все возможные его делители и включить в ответы те, которые дают остаток 0.

*Sub Делители()*

*Dim n, a As Integer*

*Dim Title, Text As String*

*Title = "Нахождение делителей числа"*

*Text = "Введите целое число"*

*a = CInt (InputBox (Text, Title))*

*Text = "делитель числа" & a & "являются: 1;"*

*For n = 2 To (a \ 2)*

*If a Mod n = 0 Then*

*Text = Text & n & ", "*

*End If*

*Next*

*Text = Text & a*

*Response = MsgBox (Text,, Title)*

*End Sub*

**ЛИТЕРАТУРА:** [\[4\]](#), [\[6\]](#), [\[10\]](#), [\[15\]](#), [\[27\]](#)

## **Занятия 51-52. ГРАФИЧЕСКИЕ ВОЗМОЖНОСТИ VBA**

*План*

- 1. Графические возможности алгоритмического языка*
- 2. Элементы управления*
- 3. Графические объекты*

В среде программирования VBA графические возможности алгоритмического языка ограничены. Это объясняется предназначением VBA - расширять возможности документов MS Office.

В среде VBA можно:

- 1. Включать в форму изображения (рис. 12.1) и реагировать на некоторые события, происходящие с изображением, так как и с элементами управления.*
- 2. Вставлять в документы MS Office графические объекты приложений такие, как диаграммы и графики, организационные схемы и блок-схемы, фигурный текст и другие.*
- 3. Выполнять рисунки на активном листе приложения методами векторной графики - рисовать линии, стрелки, круги и эллипсы, трехмерные фигуры и др.*

Среды программирования, ориентированные на разработку программного обеспечения, имеют больше встроенных графических объектов и позволяют выполнять рисование на форме или в пределах специальных элементов управления, которые включаются в форму.

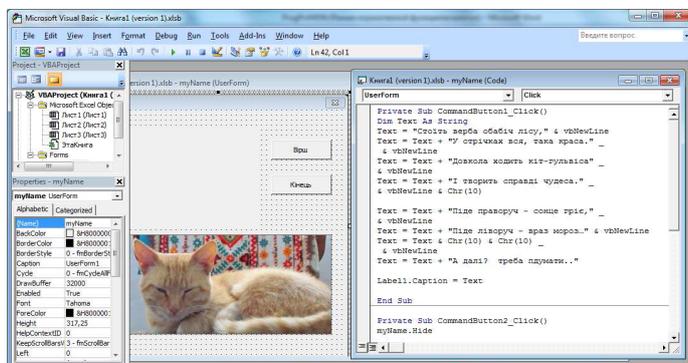


Рис. 12.1 Включение в форму изображения

### Элемент управления Image

Этот элемент управления используется для включения в форму рисунков. Например, изображения товаров или фотографии сотрудников. Элемент управления Image может работать только с графическими форматами .bmp, .gif, .jpg, .wmf, .ico и .cur.

На рис. 12.1 показаны события с элементом управления Image, на которые может реагировать программа в VBA.

### Программа 1

Следующий проект вызывается кнопкой "Сказка" и выполняет следующие действия:

1. При нажатии кнопки "Сказка" на поверхности формы появляется начало сказки о животном, которое изображено на рис. 12.1.
2. Если щелкнуть мышкой на изображении животного, в диалоговом окне появляется сообщение об этом животном (рис. 12.2).

Следующие процедуры реализуют этот проект.

```
Private Sub CommandButton1_Click()
```

```
Dim Text As String
```

```

Text = "У Лукоморья дуб зеленый," & VbNewLine
Text = Text + "Златая цепь на дубе том"& VbNewLine
Text = Text + "И днем, и ночью кот ученый"& VbNewLine
Text = Text + "Все ходит по цепи кругом."& VbNewLine & Chr(10)
Text = Text + "Идет направо – песнь заводит,"& VbNewLine
Text = Text+ "Налево – сказку говорит" & VbNewLine
Text = Text & Chr(10) & Chr(10) & VbNewLine
Label1.Caption = Text
End Sub

Private Sub CommandButton2_Click()
UserForm1.Hide
End Sub

Private Sub Image1_Click()
Dim Text As String
Text = "Этот кот, его зовут Барсик" & VbNewLine
Text = Text + "Он очень красивый." & VbNewLine
Text = Text & Chr(10)
Text = Text + "Он мой питомец"
MsgBox Text
End Sub

```

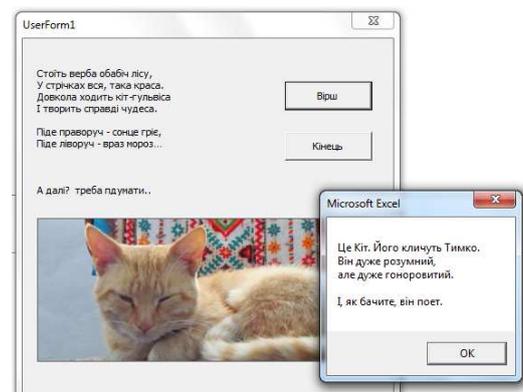


Рис. 12.2 Результат работы программы 1

Анализ. Обратите внимание на управление выводом текста с помощью символов & Chr (13) и & Chr (10). Без них текст будет произвольно переноситься из строки в строку.

## ГРАФИЧЕСКИЕ ОБЪЕКТЫ

Среда VBA позволяют работать и с графическими элементами: линиями, стрелками, фигурами, рамками, текстом и рисунками. VBA позволяет использовать графику как на этапе построения программы (проекта) через объект Image, так и на этапе ее выполнения через элементы управления.

Графические элементы добавляются в программы VBA через объект **Shape** - образ, являющийся одним из свойств активного документа **ActiveDocument**. Множество образов активного документа образует коллекцию Shapes.

Для того, чтобы поместить графический документ на странице активного документа, надо указать координаты размещения.

Координаты точки и размеры элементов задаются в документа и на форме в пунктах:

$$1\text{pt} = 1 \text{ пункт} = 1/72 \text{ дюйма} = 0,035 \text{ см} \quad 1 \text{ дюйм} = 1'' = 2,54 \text{ см}$$

Многие объекты VBA, в том числе и геометрический элементы, имеют свойства, которые выражаются в пунктах:

1. **Left** - расстояние от левого края документа или формы;
2. **Top** - расстояние от верхнего края;
3. **Width** - ширина;
4. **Height** - высота.

Координаты точки в VBA задаются двумя числами: (X; Y) = (Left; Top) и тоже выражаются в пунктах. Для включения геометрической фигуры в документ используют метод коллекции Shapes:

*AddShape (типОбъекта, Left, Top, Width, Height)*, где типОбъекта - тип объекта, который надо включить в документ.

Значением переменной *типОбъекта* есть константа среды программирования VBA, связанная с данной геометрической фигурой. VBA позволяет использовать более 100 геометрических фигур. Изображения этих фигур можно посмотреть на панели инструментов *Рисование* приложения MS Office, а константы для них - в среде программирования VBA - в окне просмотра (браузере) объектов, которое открывается функциональной клавишей F2 в библиотеке Office (рис. 12.3). Этот список можно найти также в справочной службе VBA.

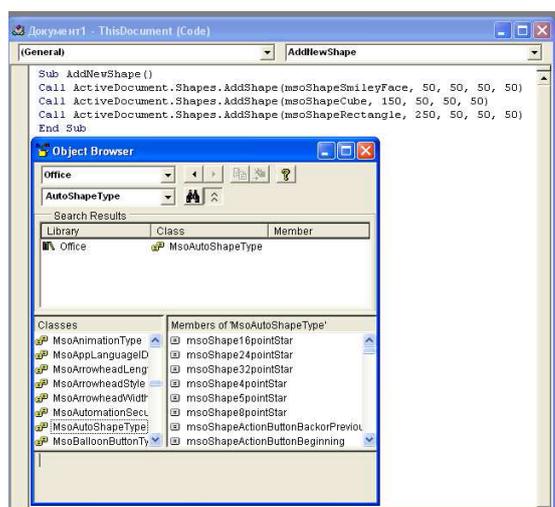


Рис. 12.3 Константы геометрических фигур объекта Shape

Вызов метода AddShape выполняется так, как показано на рис. 12.3. Названия констант на английском языке содержательные, они описывают форму этих объектов. Приведенная на рис. 12.3 программа вводит в документ MS Word изображение лица с улыбкой, куба и прямоугольника (рис. 12.4).

## Программа 2.

```
Sub AddNewShape ()
Call ActiveDocument.Shapes.AddShape (msoShapeSmileyFace, 50, 50, 50, 50)
Call ActiveDocument.Shapes.AddShape (msoShapeCube, 150, 50, 50, 50)
Call ActiveDocument.Shapes.AddShape (msoShapeRectangle, 250, 50, 50, 50)
End Sub
```

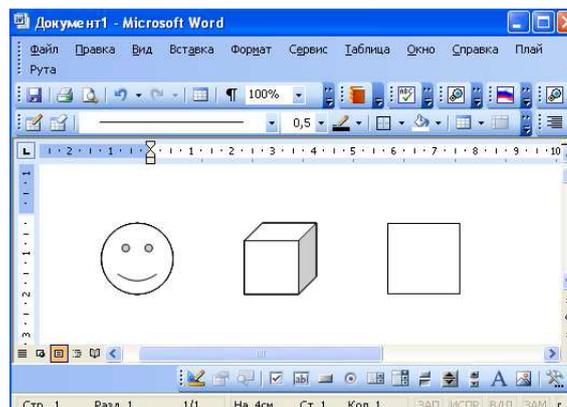


Рис. 12.4 Результат работы программы Рис. 12.3

Анализ. Обратите внимание, что программа включает изображения в документ приложения MS Word, откуда она была вызвана.

В коллекции Shapes есть и другие методы для включения в документ отдельных типов геометрических элементов.

Следующий метод выводит отрезок прямой:

**AddLine (X1, Y1, X2, Y2)**, где (X1, Y1) - координаты начала отрезка, а (X2, Y2) - координаты конца отрезка в пунктах.

Если в документ включены несколько образов, их можно различать по номерам в коллекции Shapes. Так в Программе 2 лицо имеет номер 1, куб - номер 2, а прямоугольник - 3.

Образ получает свой номер, когда он включается в программу.

### Программа 3

Различные методы включения графических элементов в документ приведены в следующей программе. Эта программа включает в документ MS Word четыре геометрические фигуры (рис. 12.5)

```
Sub AddMyShapes()
```

```
' Пример 1. Выводит отрезок прямой толщиной 5 pt
```

```
Call ActiveDocument.Shapes.AddLine(50, 150, 300, 50)
```

```
Call ActiveDocument.Shapes(1).Select
```

```
Selection.ShapeRange.Line.Weight = 5
```

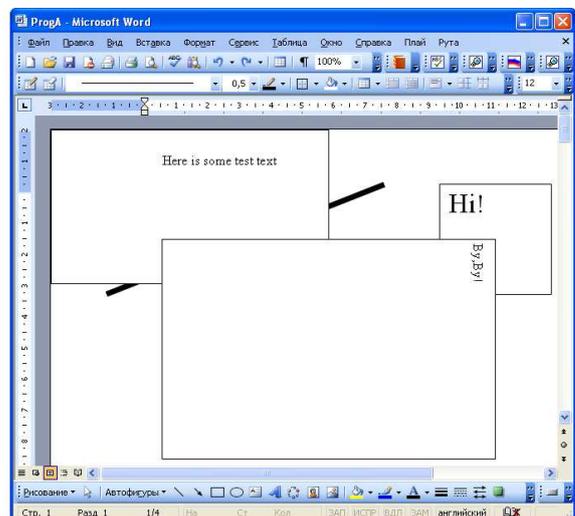
```

' Пример 2. Выводит надпись шрифтом 25 pt в рамке
Call ActiveDocument.Shapes._
AddTextbox(msoTextOrientationHorizontal,350,50,100,100)
Call ActiveDocument.Shapes(2).Select
Selection.Font.Size = 25
Selection.Text = "Hi!"

' Приклад 3. Выводит прямоугольник с надписью внутри
Set myDocument = ActiveDocument
With myDocument.Shapes.AddShape(msoShapeRectangle, _
0, 0, 250, 140).TextFrame
.TextRange.Text = "Here is some test text"
.MarginBottom = 0
.MarginLeft = 100
.MarginRight = 0
.MarginTop = 20
End With

' Пример 4. Выводит вертикальную надпись в рамке
Call ActiveDocument.Shapes._
AddTextbox(msoTextOrientationVertical,100,100,300,200)
Call ActiveDocument.Shapes(4).Select
Selection.Text = "By,By!"
End Sub

```



Анализ. В примере 3 этой программы используется оператор `set`, который присваивает ссылки на объект переменной или свойству объекта. После этого такую переменную можно использовать вместо ссылки на объект.

Пример 3 использует оператор `With ... End With`, который создает новый объект - прямоугольник - и устанавливает значение его свойство - текст внутри прямоугольника, а также отступления текста от границ.

Видим, что в коллекции `Shapes` каждый новый объект имеет свой номер. Этот номер определяется порядком, в котором объекты создавались. ? Объект можно выделить по этому номеру, а затем устанавливать или изменять его свойства. Поскольку создаются геометрические объекты, то при их наложении новый объект может частично или полностью закрывать предыдущий.

**ЛИТЕРАТУРА:** [\[4\]](#), [\[6\]](#), [\[10\]](#), [\[15\]](#), [\[27\]](#)