

**Министерство образования и науки Российской Федерации**

**ФЕДЕРАЛЬНО ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«КАРАЧАЕВО-ЧЕРКЕССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
имени У.Д. АЛИЕВА»**



# **Информатика**

***В помощь студентам и преподавателям***

**Карачаевск  
2018**

УДК 681.142.37  
ББК 32.973я73

Печатается по решению редакционно-издательского совета Карачаево-Черкесского государственного университета им. У. Д. Алиева

**Информатика.** Учебно-методическое пособие. –Карачаевск: КЧГУ, 2018. –167 с.

В пособии содержится необходимый теоретический и практический материал по информатике.

Предназначено для студентов высших учебных заведений, обучающихся по специальностям «Информатика и вычислительная техника», «Прикладная информатика», «Прикладная математика и информатика». Пособие может быть использовано старшеклассниками и учителями для подготовки к ЕГЭ по информатике.

**Авторы -составители:**

М.М. Бостанова(Уурсова)  
З.К. Джаубаева

**Рецензенты:**

Б.И. Уурсова, д.ф.-м., проф.  
П.А. Кочкарова, к.ф.-м. доц.

## СОДЕРЖАНИЕ

<b>ПРЕДИСЛОВИЕ .....</b>	<b>4</b>
<b>I. ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ И СИСТЕМЫ .....</b>	<b>5</b>
I.1. ИНФОРМАЦИЯ И ЕЕ КОДИРОВАНИЕ .....	5
<i>I.1.1. Единицы измерения информации. Количество информации и скорость ее передачи .....</i>	<i>5</i>
<i>I.1.2. Представление цифровой информации. Системы счисления.....</i>	<i>13</i>
I.2. АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ.....	<b>ОШИБКА!</b>
<b>ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.</b>	
<i>I.2.1. Алгоритмы. Свойства алгоритмов. Способы описания алгоритмов .....</i>	<i>Ошибка! Закладка не определена.</i>
<i>I.2.2. Базовые алгоритмические структуры .....</i>	<i>Ошибка! Закладка не определена.</i>
I.3. ОСНОВЫ ЛОГИКИ.....	23
<b>II. ИНФОРМАЦИОННЫЕ И КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ .....</b>	<b>28</b>
II.1. ФАЙЛОВЫЕ СИСТЕМЫ .....	28
II.2. КОДИРОВАНИЕ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ .....	31
II.3. ОБРАБОТКА ИНФОРМАЦИИ В ЭЛЕКТРОННЫХ ТАБЛИЦАХ .....	39
<i>II.3.1. Основные понятия электронных таблиц .....</i>	<i>39</i>
<i>II.3.2. Применение электронных таблиц для расчетов.....</i>	<i>43</i>
II.4. БАЗЫ ДАННЫХ.....	49
II.5. ТЕЛЕКОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ.....	50
<i>II.5.1. Адресация в сети Интернет.....</i>	<i>50</i>

## ПРЕДИСЛОВИЕ

Информация – основной объект изучения информатики, абстрактное понятие, используемое для уточнения бытового содержания этого слова, отражающего систему сведений об окружающем мире.

Авторы надеются, что учебно-методическое пособие будет полезно не только студентам педагогических вузов и «классических» университетов, но и практикующим учителям информатики для самообразования и повышения квалификации. Эта книга может быть полезна и многим другим, изучающим информатику. Современная информатика очень велика по объему и очень динамична.

Учебно-методическое пособие содержит теоретический и практический материал, необходимый при изучении дисциплин «Информатика», «Информатика и программирование». Книга составлена в соответствии с федеральным государственным образовательным стандартом по данным направлениям подготовки. Каждую тему предваряет теоретический материал, затем предлагаются практические задания с решениями.

# **I. ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ И СИСТЕМЫ**

## ***1.1. Информация и ее кодирование***

### ***1.1.1. Единицы измерения информации. Количество информации и скорость ее передачи***

В канале связи сообщение, составленное из символов (букв) одного алфавита, может преобразовываться в сообщение из символов (букв) другого алфавита. Правило, описывающее однозначное соответствие букв алфавитов при таком преобразовании, называют *кодом*. Саму процедуру преобразования сообщения называют *перекодировкой*. Подобное преобразование сообщения может осуществляться в момент поступления сообщения от источника в канал связи (кодирование) и в момент приема сообщения получателем (декодирование). Устройства, обеспечивающие кодирование и декодирование, называются соответственно кодировщиком и декодировщиком. Кодирование заключается в том, что каждому символу ставится в соответствие уникальный двоичный код от 00000000 до 11111111 (или десятичный код от 0 до 255).

Компьютер считают универсальным преобразователем информации. Тексты на естественных языках и числа, математические и специальные символы – одним словом все, что в быту или в профессиональной деятельности может быть

необходимо человеку, должно иметь возможность быть введенным в компьютер.

В силу безусловного приоритета двоичной системы счисления при внутреннем представлении информации в компьютере кодирование «внешних» символов основывается на сопоставлении каждому из них определенной группы двоичных знаков. При этом из технических соображений и из соображений удобства кодирования-декодирования следует пользоваться равномерными кодами, т.е. двоичными группами равной длины.

Наиболее распространены две системы кодирования: EBCDIC (Extended Binary Coded Decimal Interchange Code) и ASCII (American Standard Information Interchange).

Первая – исторически тяготеет к «большим» машинам, вторая чаще используется на мини- и микро-ЭВМ (включая персональные компьютеры). Познакомимся подробнее именно с ASCII, созданной в 1963 г.

В своей первоначальной версии это – система семибитного кодирования. Она ограничивалась одним естественным алфавитом (английским), цифрами и набором различных символов, включая «символы пишущей машинки» (привычные знаки препинания, знаки математических действий и др.) и «управляющие символы».

В следующей версии фирма IBM перешла на расширенную 8-битную кодировку. В ней первые 128 символов совпадают с исходными и имеют коды со старшим битом равным нулю, а остальные коды отданы под буквы некоторых европейских языков, в основе которых лежит латиница, греческие буквы,

математические символы (скажем, знак квадратного корня) и символы псевдографики. С помощью последних можно создавать таблицы, несложные схемы и др. Каждому символу ASCII соответствует 8-битовый двоичный код, например, А- 01000001, С – 01000011 и т.д.

Для представления букв русского языка (кириллицы) в рамках ASCII было предложено несколько версий. Первоначально был разработан ГОСТ под названием КОИ-7, оказавшийся по ряду причин крайне неудачным; ныне он практически не используется, затем КОИ-8 (код обмена информацией, восьмизначный).

В таблице приведена часто используемая в нашей стране модифицированная альтернативная кодировка. В левую часть входят исходные коды ASCII; в правую часть (расширение ASCII) вставлены буквы кириллицы взамен букв, немецкого, французского алфавитов (не совпадающих по написанию с английскими), греческих букв, некоторых спецсимволов.

Знакам алфавита ПК ставятся в соответствие шестнадцатеричные числа по правилу: первая – номер столбца, вторая – номер строки. Например: английская 'А' – код 41, русская 'и' – код А8.

Одним из достоинств этой системы кодировки русских букв является их естественное упорядочение, т.е. номера букв следуют друг за другом в том же порядке, в каком сами буквы стоят в русском алфавите. Это очень существенно при решении ряда задач обработки текстов, когда требуется выполнить или использовать лексикографическое упорядочение слов.

## Таблица кодов ASCII

### Основная таблица ASCII

	00	10	20	30	40	50	60	70
0		▸		0	@	P	`	р
1	☒	◀	!	1	A	Q	a	q
2	☒	↕	"	2	B	R	b	r
3	♥	!!	#	3	C	S	c	s
4	♦	¶	\$	4	D	T	d	t
5	♣	§	%	5	E	U	e	u
6	♠	=	&	6	F	V	f	v
7	•	±	'	7	G	W	g	w
8	☒	↑	(	8	H	X	h	x
9	o	↓	)	9	I	Y	i	y
A	☒	→	*	:	J	Z	j	z
B	♂	←	+	;	K	[	k	{
C	♀	└	,	<	L	\	l	
D	♂	↕	-	=	M	]	m	}
E	♂	▲	.	>	N	^	n	~
F	※	▼	/	?	O	_	o	Δ

### Расширенная таблица ASCII

	80	90	A0	B0	C0	D0	E0	F0
0	А	Р	а	☒	└	└	Р	≡
1	Б	С	б	▮	└	└	с	±
2	В	Т	в	▮	└	└	т	>
3	Г	У	г		└	└	у	<
4	Д	Ф	д	└	-	└	ф	└
5	Е	Х	е	└	└	└	х	└
6	Ж	Ц	ж	└	└	└	ц	÷
7	З	Ч	з	└	└	└	ч	≈
8	И	Ш	и	└	└	└	ш	°
9	Й	Щ	й	└	└	└	щ	.
A	К	Ь	к		└	└	ь	.
B	Л	Ы	л	└	└	▮	ы	└
C	М	Ь	м	└	└	▮	ь	н
D	Н	Э	н	└	=	▮	э	2
E	О	Ю	о	└	└	▮	ю	☒
F	П	Я	п	└	└	▮	я	

Из сказанного выше следует, что даже 8-битная кодировка недостаточна для кодирования всех символов, которые хотелось бы иметь в расширенном алфавите. Все препятствия могут быть сняты при переходе на систему (UNICODE), основанную на 16-разрядном кодировании символов. Шестнадцать разрядов позволяют обеспечить уникальные коды для 65536 различных символов. Каждому символу в этой кодировке соответствует 16-битовый (2-байтовый) двоичный код.



Как было сказано выше, традиционно для кодирования одного символа используется количество информации равное 1 байту (1 байт = 8 битов). Учитывая, что каждый *бит* принимает значение 1 или 0, получаем, что с помощью 1 байта можно закодировать 256 различных символов ( $2^8=256$ ). В компьютере бит является наименьшей возможной единицей информации. Объем информации, записанной двоичными знаками в памяти компьютера или на внешнем носителе информации подсчитывается просто по количеству требуемых для такой записи двоичных символов.

Цифры кодируются по стандарту ASCII в двух случаях – при вводе-выводе и когда они встречаются в тексте. Если цифры участвуют в вычислениях, то осуществляется их преобразование в другой двоичный код.

Возьмем число 57. При использовании в тексте каждая цифра будет представлена своим кодом в соответствии с таблицей ASCII. В двоичной системе это – 00110101 00110111.

При использовании в вычислениях код этого числа будет получен по правилам перевода в двоичную систему и получим – 00111001.

Для удобства использования введены и более крупные, чем бит, единицы количества информации.

**1 байт = 8 бит**

**1 килобайт (кбайт) = 1024 байта =  $2^{10}$  байт**

**1 мегабайт (Мбайт) = 1024 килобайта =  $2^{20}$  байт**

**1 гигабайт (Гбайт) = 1024 мегабайта =  $2^{30}$  байт (миллиард)**

**1 терабайт (Тбайт) = 1024 гигабайта =  $2^{40}$  байт (триллион)**

**1 петабайт (Пбайт) = 1024 терабайт =  $2^{50}$  байт (квадриллион)**

**1 экзабайт (Эбайт) = 1024 петабайта =  $2^{60}$  байт (квинтиллион)**

**1 зеттабайт (Збайт) = 1024 экзабайта =  $2^{70}$  байт (секстиллион)**

**1 йоттабайт (Йбайт) = 1024 зеттабайта =  $2^{80}$  байт (септиллион).**

Для двоичных сообщений в качестве меры используется количество бит в сообщении. Это количество называется информационным объемом сообщения.

Алфавитный подход - это единственный способ измерения информации, который может применяться по отношению к информации, циркулирующей в информационной технике, в компьютерах.

*Алфавит - это конечное множество символов, используемых для представления информации.* Число символов в алфавите называется *мощностью алфавита* (термин взят из математической теории множеств). В основном содержании базового курса информатики алфавитный подход рассматривается лишь с позиции *равновероятного приближения*. Это значит, что допускается предположение о том, что вероятности появления всех символов алфавита в любой позиции в тексте одинаковы.

В рассматриваемом приближении количество информации, которое несет в тексте каждый символ (**i**), вычисляется из уравнения Хартли:

$$H = \log_2 N - \text{формула Хартли.}$$

Для измерения количества информации следует пользоваться формулой алфавитного подхода (формулой Хартли):  $I = M \cdot \log_2 N$ , где  $N$  — количество символов (мощность) алфавита, в котором

записано сообщение,  $M$  — количество символов в записи сообщения (длина сообщения),  $I$  — количество бит информации, содержащиеся в сообщении. Если  $\log_2 N$  не является целым числом, то  $I$  округляется в большую сторону.

В качестве примера определим количество информации, связанное с появлением каждого символа в сообщениях, записанных на русском языке. Будем считать, что русский алфавит состоит из 33 букв и знака “пробел” для разделения слов. По формуле:

$$I = \log_2 34 \approx 5 \text{ бит.}$$

Рассмотрим алфавит, состоящий из двух знаков 0 и 1. Если считать, что со знаками 0 и 1 в двоичном алфавите связаны одинаковые вероятности их появления ( $P(0)=P(1)=0,5$ ), то количество информации на один знак при двоичном кодировании будет равно:

$$I = \log_2 2 = 1 \text{ бит.}$$

Количество информации (в битах), заключенное в двоичном слове, равно числу двоичных знаков в нем.

В рассматриваемом приближении количество информации, которое несет в тексте каждый символ ( $i$ ), вычисляется из уравнения Хартли:

$$2^i = N, \text{ где } N \text{ — мощность алфавита.}$$

Величину  $i$  можно назвать информационным весом символа. Отсюда следует, что количество информации во всем тексте ( $I$ ), состоящем из  $K$  символов, равно произведению информационного

веса символа на  $K$ :  $I = ixK$ . Эту величину можно назвать информационным объемом текста.

Напомним основные положения алфавитного подхода к измерению количества информации и приведем решения примеров на данную тему.

1). Пусть  $A$  - упорядоченное множество из  $N$  элементов, тогда для кодирования каждого элемента двоичным кодом, например, путем нумерации в двоичной системе счисления, требуется  $\log_2 N$  двоичных разрядов (бит). Объем информации  $I$ , содержащейся в сообщении о том, что выбран какой-либо элемент этого множества, равен, соответственно  $\log_2 N$  бит. Если  $N$  не является целой степенью 2, то число  $\log_2 N$ , не является целым, и происходит округление в большую сторону.

2). Если некоторый алфавит содержит  $M$  символов, то информационный объем одного символа этого алфавита в сообщении равен  $\log_2 M$ . Для того чтобы найти информационный объем сообщения, состоящего из символов этого алфавита, следует  $\log_2 M$  умножить на количество символов в сообщении.

3). С помощью  $n$  двоичных разрядов (бит) можно закодировать двоичным кодом все элементы множества мощностью  $2^n$  (т.е. состоящего из  $2^n$  элементов). Информационный объем одного символа, обозначающего элемент данного множества, будет равен  $n$ .

## I.1.2. Представление цифровой информации. Системы счисления

*Системой счисления (СС)* называется способ записи чисел с помощью заданного набора специальных символов.

Различают *позиционные* и *непозиционные* системы счисления (СС). В позиционных системах вес цифры (т.е. тот вклад, который цифра вносит в значение числа) зависит от позиции цифры в записи числа. Позиция цифры называется *разрядом*.

Десятичная система счисления является примером позиционной системы, например, в числе 757,7 первая 7-ка означает 7 сотен, вторая- 7 единиц, а третья-7 десятых долей единицы. В римской же системе в числе XXX вес цифры X в любой позиции равен десяти, т.е эта система - непозиционная.

Любая позиционная система счисления характеризуется своим *основанием*.

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

В позиционной системе счисления (СС) любое число можно представить в виде:

$$N = a_k p^k + a_{k-1} p^{k-1} + \dots + a_1 p^1 + a_0 p^0 + a_{-1} p^{-1} + a_{-2} p^{-2} + \dots,$$

где  $a_i$  - цифра числа (коэффициенты при степенях  $p$ ),  $N$  - число,  $p$  - основание системы счисления ( $p > 1$ ).

Например:

$$A_{10} = 757,7 = 700 + 50 + 7 + 0,7 = 7 * 10^2 + 5 * 10^1 + 7 * 10^0 + 7 * 10^{-1}$$

За основание системы счисления можно принять любое натуральное число: 2, 3, 4 и т.д. Следовательно, **возможно бесчисленное множество позиционных систем:** двоичная, троичная, четверичная и т. д.

Основание	Система счисления	Алфавит
2	Двоичная	0,1
3	Троичная	0,1,2
4	Четверичная	0,1,2,3
5	Пятеричная	0,1,2,3,4
8	Восьмеричная	0,1,2,3,4,5,6,7
10	Десятичная	0,1,2,3,4,5,6,7,8,9
12	Двенадцатеричная	0,1,2,3,4,5,6,7,8,9,A,B
16	Шестнадцатеричная	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Во всех современных ЭВМ для представления числовой информации используется двоичная СС. Это обусловлено:

- более простой реализацией алгоритмов выполнения арифметических и логических операции;
- более надёжной физической реализацией основных функций, так как они имеют всего два состояния (0 и 1);
- экономичностью аппаратной реализации всех схем ЭВМ.

Кроме двоичной СС широко используются системы счисления с основанием, являющимся *целой степенью числа 2*, а именно: восьмеричная и шестнадцатеричная. Они используются в основном для более компактного изображения двоичной

информации, так как запись значения чисел производится существенно меньшим числом знаков.

По данным этой таблицы можно выявить целый ряд закономерностей:

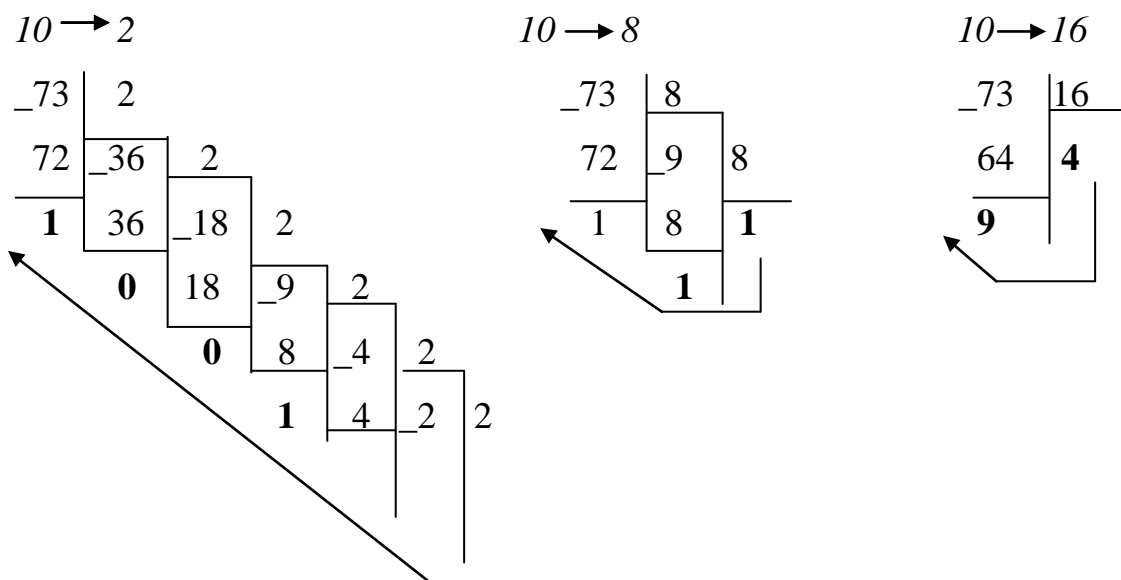
- 0 и 1 имеют единственное и одинаковое представление в любых СС;
- основание СС в любой системе имеет представление 10;
- представление чисел в различных системах счисления допускает однозначное их преобразование из одной системы в другую.

10 - я	2 - я	8 - я	16 - я
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9

10 - я	2 - я	8 - я	16 - я
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13

### *Правила перевод чисел из одной системы счисления в другую*

Рассмотрим системы счисления с основанием  $q(q=2, 8, 16)$ .



$$\begin{array}{r} 0 \quad 2 \quad \overline{1} \\ \hline 0 \quad | \\ \hline \end{array}$$

**Ответ:**  $73_{10} = 1\ 001\ 001_2 = 111_8 = 49_{16}$

При переводе десятичного числа в систему с основанием  $q$  ( $q=2, 8, 16$ ), необходимо это число последовательно делить на  $q$  до тех пор, пока не останется остаток, меньший или равный  $q$ . При этом, число с основанием  $q$  записывается как последовательность остатков от деления, записанных в обратном порядке, например:

При переводе числа из системы с основанием  $q$  ( $q=2, 8, 16$ ) в десятичную систему, надо это число представить в виде суммы степеней основания его системы счисления.

**Примеры:**

$$1011,1_2 = 1*2^3 + 1*2^1 + 1*2^0 + 1*2^{-1} = 11,5_{10}$$

$$276,5_8 = 2*8^2 + 7*8^1 + 6*8^0 + 5*8^{-1} = 190,625_{10}$$

$$1F3_{16} = 1*16^2 + 15*16^1 + 3*16^0 = 499_{10}$$

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

**Примеры:**

$$537,1_8 = 101\ 011\ 111,001_2;$$

$$1A3,F_{16} = 1\ 1010\ 0011,1111_2$$

При переводе *правильной десятичной дроби* в систему счисления с основанием  $q$  необходимо сначала саму дробь, а затем дробные части всех последующих произведений последовательно **умножать** на  $q$ , отделяя после каждого умножения целую часть



произведения. Число в новой системе счисления записывается как последовательность полученных целых частей произведения.

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

### **Арифметические операции в позиционных системах счисления**

Рассмотрим основные арифметические операции: сложение, вычитание, умножение и деление. Правила выполнения этих операций в десятичной системе хорошо известны – это сложение, вычитание, умножение столбиком и деление углом. Эти правила применимы и ко всем позиционным системам счисления. Только таблицами сложения и умножения надо пользоваться особыми для каждой системы.

#### **Сложение**

**Сложение  
в двоичной  
системе**

+	0	1
0	0	1
1	1	10

**Сложение в восьмеричной системе**

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

Таблицы сложения легко составить, используя Правило Счета.

### Сложение в шестнадцатеричной системе

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

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

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

Десятичная:  $15_{10} + 6_{10}$

$$\begin{array}{r} 1 \\ 15 \\ + \\ 6 \\ \hline 21 \end{array}$$

$\left\{ \begin{array}{l} 5 + 6 = 11 = 10 + 1 \\ 1 + 1 = 2 \end{array} \right.$

Двоичная:  $1111_2 + 110_2$

$$\begin{array}{r} 111 \\ 1111 \\ + \\ 0110 \\ \hline 10101 \end{array}$$

$\left\{ \begin{array}{l} 1 + 0 = 1 \\ 1 + 1 = 2 = 2 + 0 \\ 1 + 1 + 1 = 3 = 2 + 1 \\ 1 + 1 = 2 = 2 + 0 \end{array} \right.$

Восьмеричная:  $17_8 + 6_8$

$$\begin{array}{r} 1 \\ 17 \\ + \\ 6 \\ \hline 25 \end{array}$$

$\left\{ \begin{array}{l} 7 + 6 = 13 = 8 + 5 \\ 1 + 1 = 2 \end{array} \right.$

Шестнадцатеричная:  $F_{16} + 6_{16}$

$$\begin{array}{r} 1 \\ F \\ + \\ 6 \\ \hline 15 \end{array}$$

$\left\{ \begin{array}{l} 15 + 6 = 21 = 16 + 5 \end{array} \right.$

**Ответ:**  $15 + 6 = 21_{10} = 10101_2 = 25_8 = 15_{16}$

**Проверка:** Преобразуем полученные суммы к десятичному виду:

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

$$25_8 = 2 * 8^1 + 5 * 8^0 = 16 + 5 = 21$$

$$15_{16} = 1 * 16^1 + 5 * 16^0 = 16 + 5 = 21$$

### Вычитание

**Пример 3.** Вычтем единицу из чисел  $10_2$ ,  $10_8$ ,  $10_{16}$ .

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

Заемы	1	1	1
	1 0	1 0	1 0
	-	-	-
	$\frac{1}{1}$	$\frac{1}{7}$	$\frac{1}{F}$
	$2 - 1 = 1$	$8 - 1 = 7$	$16 - 1 = 15 = F$

**Ответ:**  $10 - 1 = 1_2 = 7_8 = F_{16}$

### Умножение

Выполняя умножение многозначных чисел в различных позиционных системах счисления, можно использовать обычный алгоритм перемножения чисел в столбик, но при этом результаты перемножения и сложения однозначных чисел необходимо заимствовать из соответствующих в рассматриваемой системе таблиц умножения и сложения. В виду чрезвычайной простоты таблицы умножения в двоичной системе, умножение сводится лишь к сдвигам множимого и сложения.

**Умножение  
в двоичной  
системе**

*	0	1
0	0	0
1	0	1

**Умножение в восьмеричной системе**

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

**Пример 7.** Перемножим числа 5 и 6 в различных системах счисления.

Десятичная:  $5_{10} \times 6_{10}$

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

Двоичная:  $101_2 \times 110_2$

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

Восьмеричная:  $5_8 \times 6_8$

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

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

**Проверка:** Преобразуем полученные произведения к десятичному виду:

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

$$36_8 = 3 * 8^1 + 6 * 8^0 = 30$$

### Деление

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

**Пример 9.** Разделим число 30 на число 6 в различных системах счисления.

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

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

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

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

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

**Пример 10.** Разделим число 5865 на число 115.

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

$$\begin{array}{r} 5865 \overline{) 115} \\ - 575 \\ \hline 115 \\ - 115 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1011011101001 \overline{) 1110011} \\ - 1110011 \\ \hline 10001000 \\ - 1110011 \\ \hline 10101100 \\ - 1110011 \\ \hline 1110011 \\ - 1110011 \\ \hline 0 \end{array}$$

Восьмеричная:  $13351_8 : 163_8$

$$\begin{array}{r} 13351 \overline{) 163} \\ - 1262 \\ \hline 531 \\ - 531 \\ \hline 0 \end{array}$$

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

**Проверка:** Преобразуем полученные частные к десятичному

виду:

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

## **I.2. Основы логики**

*Алгебра логики* – это математический аппарат, с помощью которого записывают, вычисляют, упрощают логические высказывания.

*Логическое высказывание* – это повествовательное предложение, в отношении которого можно однозначно сказать, истинно оно или ложно.

**Например:** Предложение «6 – четное число» следует считать высказыванием, так как оно истинно. Предложение «Рим – столица Франции» - тоже высказывание, ложное.

Высказыванием не является, например, предложение «ученик 10 класса» «информатика - интересный предмет», первое ничего не утверждает об ученике, второе – неопределенное предложение. Вопросительные и восклицательные предложения не являются высказываниями, поскольку говорить об их истинности и ложности не имеет смысла.

Из простых высказываний посредством операций создаются сложные, причем их логическое значение зависит от составляющих их простых высказываний.

Каждая логическая связка рассматривается как операция над логическими высказываниями и имеет свое название и обозначение:

**1. Операция, выражаемая словом «не», называется отрицанием** и обозначается чертой над высказыванием ( или знаком  $\neg$  ). Высказывание  $\bar{A}$  истинно, когда  $A$  ложно, и ложно,

когда  $A$  истинно. Пример, «Луна – спутник Земли» ( $A$ ); «Луна – не спутник Земли» ( $\bar{A}$ ).

$A$	$\neg A$
0	1
1	0

**2. Операция, выражаемая связкой «и», называется конъюнкцией** или логическим умножением и обозначается « $\wedge$ » или «\*» (может также обозначаться знаками  $\wedge$  и  $\&$ ). Высказывание  $A \wedge B$  истинно тогда и только тогда, когда оба высказывания  $A$  и  $B$  истинны, в остальных случаях – ложно. Например, высказывание «10 делится на 2 и 5 больше 3» истинно, а высказывания «10 делится на 2 и 5 не больше 3», «10 не делится на 2 и 5 больше 3», «10 не делится на 2 и 5 не больше 3» ложны.

$A$	$B$	$A \wedge B$
0	1	0
0	0	0
1	0	0
1	1	1

**3. Операция, выражаемая связкой «или», называется дизъюнкцией** или логическим сложением и обозначается точкой «+» или знаком  $\vee$ . Высказывание  $A \vee B$  ложно тогда и только тогда, когда оба высказывания  $A$  и  $B$  ложны, истинно во всех остальных случаях. Например, высказывание «10 не делится на 2 или 5 не больше 3» ложно, а высказывания «10 делится на 2 или 5 не больше 3», «10 не делится на 2 или 5 больше 3», «10 не делится на 2 или 5 больше 3», «10 не делится на 2 или 5 больше 3» истинны.



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

**4. Операция, выражаемая связками «если ..., то», «из ... следует», «...влечет...» называется импликацией и обозначается знаком  $\rightarrow$ . Высказывание  $A \rightarrow B$  ложно тогда и только тогда, когда  $A$  истинно, а  $B$  – ложно, истинно во всех остальных случаях. (или «из истины следует ложь»). Например даны два высказывания: «данный четырехугольник - квадрат» ( $A$ ) и «около данного четырехугольника можно описать окружность» ( $B$ ). Рассмотрим составное высказывание  $A \rightarrow B$ , понимаемое как «если данный четырехугольник квадрат, то около него можно описать окружность». Есть три варианта, когда высказывание  $A \rightarrow B$  истинно:**

1)  $A$  истинно и  $B$  истинно, то есть данный четырехугольник квадрат, и около него можно описать окружность;

2)  $A$  ложно и  $B$  истинно, то есть данный четырехугольник не является квадратом, но около него можно описать окружность ( это справедливо не для всякого четырехугольника );

3)  $A$  ложно и  $B$  ложно, то есть данный четырехугольник не является квадратом, и около него нельзя описать окружность.

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

Ложен только один вариант:  $A$  истинно и  $B$  ложно, то есть данный четырехугольник является квадратом, но около него нельзя описать окружность.

**5. Операция, выражаемая связками «тогда и только тогда», «необходимо и достаточно», «...равносильно...» называется эквиваленцией** или **двойной импликацией** и обозначается знаком  $\leftrightarrow$  или  $\sim$ . Высказывание  $A \leftrightarrow B$  истинно тогда и только тогда, когда логические значения  $A$  и  $B$  совпадают.

Например, высказывания:

«24 делится на 6 тогда и только тогда, когда 24 делится на 3»,

«23 делится на 6 тогда и только тогда, когда 23 делится на 3»,

«24 делится на 6 тогда и только тогда, когда 24 делится на 5»,

«21 делится на 6 тогда и только тогда, когда 21 делится на 3»

ложны.

A	B	$A \leftrightarrow B$
0	1	0
0	0	1
1	0	0
1	1	1

Итак, нами рассмотрены пять логических операций: отрицание, конъюнкция, дизъюнкция, импликация и эквиваленция.

**Импликацию** можно выразить через дизъюнкцию и отрицание:

$$A \rightarrow B = \bar{A} \vee B$$

**Эквиваленцию** можно выразить через отрицание, дизъюнкцию и конъюнкцию:

$$A \leftrightarrow B = (\bar{A} \vee B) * (\bar{B} \vee A)$$

**Порядок выполнения логических операций:** отрицание, дизъюнкция и конъюнкция, импликация и эквиваленция.

Если две формулы А и В «одновременно», то есть при одинаковых наборах значений входящих в них переменных, принимают одинаковые значения, то они называются **равносильными**.

### Основные законы алгебры логики

Закон	Для ИЛИ	Для И
Переместительный	$x \vee y = y \vee x$	$x \wedge y = y \wedge x$
Сочетательный	$x \vee (y \vee z) = (x \vee y) \vee z$	$x \wedge (y \wedge z) = (x \wedge y) \wedge z$
Распределительный	$x \wedge (y \vee z) = x \wedge y \vee x \wedge z$	$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$
Правила де Моргана	$\overline{x \vee y} = \overline{x} \wedge \overline{y}$	$\overline{x \wedge y} = \overline{x} \vee \overline{y}$
Идемпотенции	$x \vee x = x$	$x \wedge x = x$
Поглощения	$x \vee (x \wedge y) = x$	$x \wedge (x \vee y) = x$
Склеивания	$(x \wedge y) \vee (\overline{x} \wedge y) = y$	$(x \vee y) \wedge (\overline{x} \vee y) = y$
Операция переменной с ее инверсией	$x \vee \overline{x} = 1$	$x \wedge \overline{x} = 0$
Операция с константами	$x \vee 0 = x$ $x \vee 1 = 1$	$x \wedge 0 = 0$ $x \wedge 1 = x$
Двойного отрицания	$\overline{\overline{x}} = x$	

**Составление таблицы истинности для логической формулы.**

Таблица истинности логической формулы выражает соответствие между возможными наборами значений переменных и значениями формулы.

Набор значений имеет количество равное  $2^n$ , где n- количество высказываний участвующих в формуле.

## II. ИНФОРМАЦИОННЫЕ И КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ

### II.1. *Файловые системы*

При наличии большого числа программ и данных необходим строгий их учет и систематизация. Операционным системам приходится работать с различными потоками данных, разными аппаратными и периферийными устройствами компьютера. Организовать упорядоченное управление всеми этими объектами позволяет файловая система.

Под файлом при этом понимают набор данных на диске, термине или каком-либо другом устройстве. Таким образом, файловая система - это система управления данными. т.е. во всех операционных системах имеющаяся на компьютере информация хранится в виде файлов. Каждый файл имеет имя и располагается на определенном устройстве хранения информации. В виде файлов хранятся программы (такие файлы называются выполняемыми) и документы. Иногда в состав одного приложения или документа входят несколько файлов. Для удобства хранения и поиска файлов они объединены в папки. Синонимами термина "папка", являются слова "каталог" и "директория". Подобно файлам папки имеют свои имена.

Наиболее развитая многоуровневая файловая система UNIX поддерживает иерархическую (древовидную) систему каталогов (рис.1). Папки вложены друг в друга, образуя многоуровневую древовидную структуру. Каждый пользователь может работать в

составе этой структуры со своей системой каталогов (со своим поддеревом). Полное имя файла в данной структуре задает путь переходов между каталогами в логической структуре каталогов.

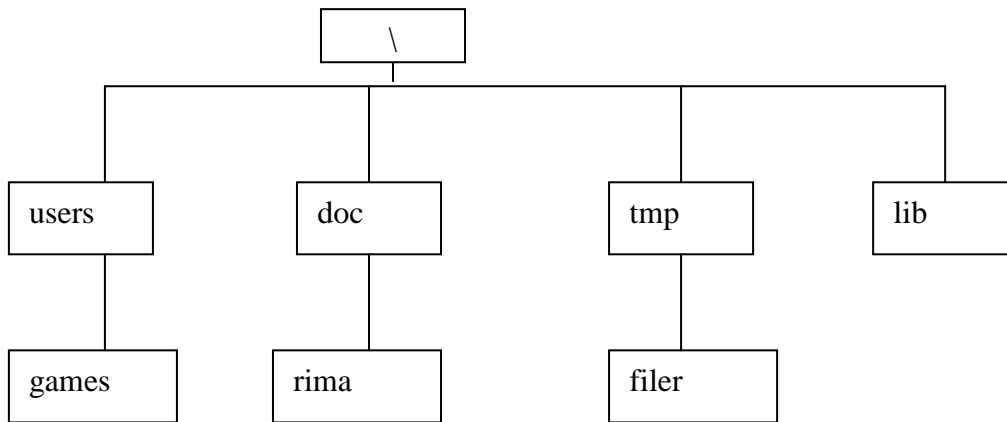


Рис. 1. Иерархическая система каталогов.

Файл обладает уникальным идентификатором (именем), обеспечивающим доступ к файлу. Идентификатор включает в себя собственно имя - буквенно-цифровое обозначение файла, которое может содержать специальные символы (подчеркивание, дефис, ! и т.д.), и расширение имени файла (обычно отделяемое от имени файла точкой). Если имена создаваемых файлов пользователь может задавать произвольно, то в использовании расширений следует придерживаться традиции, согласно которой расширение указывает на тип файла, характер его содержимого. Например, в операционной системе MS-DOS файлы с расширениями:

.com, .exe, .bat	- исполняемые
.txt, .doc	- текстовые
.pas, .bas, .c	- тексты программ на известных языках программирования: Паскаль, Бейсик, Си, соответственно
.for, .dbf	- файлы базы данных.

Известны десятки стандартных расширений, используемых при работе с различными программными системами.

В различных ОС существуют определенные ограничения на длину имени и расширения имени файла. Так, в MS-DOS длина имени файла не должна превышать восьми символов, а расширение - трех. В ОС UNIX ограничения значительно менее жесткие.

Полное имя файла состоит из пути к файлу и имени файла. Путь к файлу представляет собой перечень имен каталогов (папок), которые нужно последовательно посетить, чтобы спуститься к файлу с самого высокого уровня дерева файлов. Во многих операционных системах полное имя файла начинается с имени устройства внешней памяти, на котором он расположен. Имя устройства образуется двумя символами - латинской буквой и следующим за ней двоеточием. После имени устройства следуют имена папок, разделенные символом «\» или «/».

Имена директории, начиная от корневого, образующие путь к файлу, отделяемые при записи друг от друга косой чертой (\ в DOS, / в UNIX), также как и обозначение диска, относятся к идентификатору файла. Например, в MS-DOS:

```
d:\lang\pascal\work\example.pas
```

есть файл с именем `example` и расширением `pas`, указывающем на то, что это текст программы на Паскале, полный путь к которому:

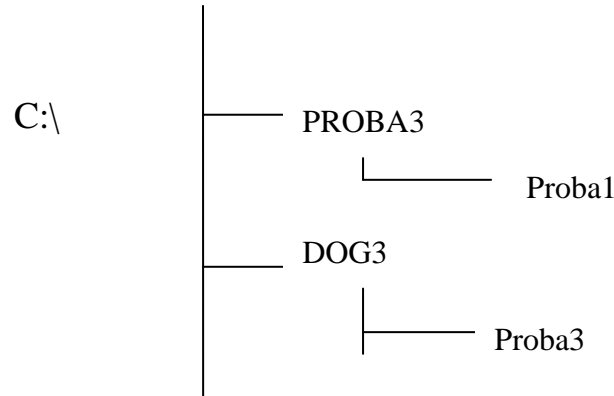
`d:` - это диск `d`;

`\lang\pascal\work` - это структура вложенных директорией, в самом внутреннем из которых находится необходимый файл `example.pas`.

Рассмотрим решение типовых задач по этой теме.

### Пример 1.

Дано дерево каталогов.



Определите полное имя файла Proba3

- 1) C:\ PROBA3
- 2) C:\ PROBA3\ Proba3
- 3) C:\ PROBA3\ Proba1
- 4) C:\ DOG3\ Proba3

**Решение:**

## ***II.2. Кодирование графической информации***

Под графической информацией можно понимать рисунок, чертеж, фотографию, картинку в книге, изображения на экране телевизора или в кинозале и т. д. Для обсуждения общих принципов кодирования графической информации в качестве конкретного, достаточно общего случая графического объекта выберем изображение на экране телевизора. Это изображение состоит из некоторого количества горизонтальных линий – строк. А каждая строка в свою очередь состоит из элементарных мельчайших единиц изображения – точек, которые принято называть пикселями (pixel – Picture's element – элемент картинки).

Весь массив элементарных единиц изображения называют растром (лат. *gastrum* – грабли). Степень четкости изображения зависит от количества строк на весь экран и количества точек в строке, которые представляют разрешающую способность экрана или просто разрешение. Чем больше строк и точек, тем четче и лучше изображение. Например, разрешение 640x480, то есть 640 точек на строку и 480 строчек на экран.

Строки, из которых состоит изображение, можно просматривать сверху вниз друг за другом, как бы составив из них одну сплошную линию. После полного просмотра первой строки просматривается вторая, за ней третья, потом четвертая и т. д. до последней строки экрана. Так как каждая из строк представляет собой последовательность пикселей, то все изображение, вытянутое в линию, также можно считать линейной последовательностью элементарных точек. В рассматриваемом случае эта последовательность состоит из  $640 \times 480 = 307200$  пикселей. Вначале рассмотрим принципы кодирования монохромного изображения, то есть изображения, состоящего из любых двух контрастных цветов – черного и белого, зеленого и белого, коричневого и белого и т. д. Для простоты обсуждения будем считать, что один из цветов – черный, а второй – белый. Тогда каждый пиксель изображения может иметь либо черный, либо белый цвет. Поставив в соответствие черному цвету двоичный код “0”, а белому – код “1” (либо наоборот), мы сможем закодировать в одном бите состояние одного пикселя монохромного изображения. А так как байт состоит из 8 бит, то на



строчку, состоящую из 640 точек, потребуется 80 байтов памяти, а на все изображение – 38 400 байтов.

Однако полученное таким образом изображение будет чрезмерно контрастным. Реальное черно-белое изображение состоит не только из белого и черного цветов. В него входят множество различных промежуточных оттенков – серый, светло-серый, темно-серый и т. д. Если кроме белого и черного цветов использовать только две дополнительные градации, скажем светло-серый и темно-серый, то для того чтобы закодировать цветовое состояние одного пикселя, потребуется уже два бита. При этом кодировка может быть, например, такой: черный цвет – 002, темно-серый – 012, светло-серый – 102, белый – 112.

Общепринятым на сегодняшний день, дающим достаточно реалистичные монохромные изображения, считается кодирование состояния одного пикселя с помощью одного байта, которое позволяет передавать 256 различных оттенков серого цвета от полностью белого до полностью черного. В этом случае для передачи всего раstra из 640x480 пикселей потребуется уже не 38 400, а все 307 200 байтов.

Цветное изображение может формироваться различными способами. Один из них – метод RGB (от слов Red, Green, Blue – красный, зеленый, синий), который опирается на то, что глаз человека воспринимает все цвета как сумму трех основных цветов – красного, зеленого и синего. Например, сиреневый цвет – это сумма красного и синего, желтый цвет – сумма красного и зеленого и т. д. Для получения цветного пикселя в одно и то же место экрана

направляется не один, а сразу три цветных луча. Опять упрощая ситуацию, будем считать, что для кодирования каждого из цветов достаточно одного бита. Нуль в бите будет означать, что в суммарном цвете данный основной отсутствует, а единица – присутствует. Следовательно, для кодирования одного цветного пикселя потребуется 3 бита – по одному на каждый цвет. Пусть первый бит соответствует красному цвету, второй – зеленому и третий – синему. Тогда код 101(2) обозначает сиреневый цвет – красный есть, зеленого нет, синий есть, а код 110(2) – желтый цвет – красный есть, зеленый есть, синего нет. При такой схеме кодирования каждый пиксель может иметь один из восьми возможных цветов. Если же каждый из цветов кодировать с помощью одного байта, как это принято для реалистического монохромного изображения, то появится возможность передавать по 256 оттенков каждого из основных цветов. А всего в этом случае обеспечивается передача  $256 \times 256 \times 256 = 16\,777\,216$  различных цветов, что достаточно близко к реальной чувствительности человеческого глаза. Таким образом, при данной схеме кодирования цвета на изображение одного пикселя требуется 3 байта, или 24 бита, памяти. Этот способ представления цветной графики принято называть режимом True Color (true color – истинный цвет) или полноцветным режимом.

Следует упомянуть еще один часто используемый метод представления цвета, в котором вместо основного цвета используется его дополнение до белого. Если три цвета: красный, зеленый и синий вместе дают белый, то дополнением для красного,

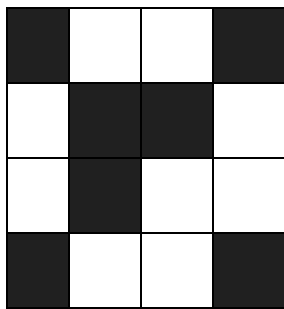
очевидно, является сочетание зеленого и синего, то есть голубой цвет. Аналогичным образом дополнением для зеленого является сочетание красного и синего, то есть пурпурный, а для синего – сочетание красного и зеленого, то есть желтый цвет. Эти три цвета – голубой, пурпурный и желтый с добавлением черного образуют основные цвета в системе кодирования, которая называется CMYK (от Cyan – голубой, Magenta – пурпурный, Yellow – желтый и black – черный). Этот режим также относится к полноцветным, но для передачи состояния одного пикселя в этом случае требуется 32 бита, или четыре байта, памяти, и может быть передано 4 294 967 295 различных цветов.

Полноцветные режимы требуют очень много памяти. Так, для обсуждавшегося выше растра 640x480 при использовании метода RGB требуется 921 600, а для режима CMYK – 1 228 800 байтов памяти. В целях экономии памяти разрабатываются различные режимы и графические форматы, которые немного хуже передают цвет, но требуют гораздо меньше памяти. В частности, можно упомянуть режим High Color (high color – богатый цвет), в котором для передачи цвета одного пикселя используется 16 битов и, следовательно, можно передать 65 535 цветовых оттенков, а также индексный режим, который базируется на заранее созданной таблице цветовых оттенков. Нужный цвет выбирается из этой таблицы с помощью номера – индекса, который занимает всего один байт памяти.

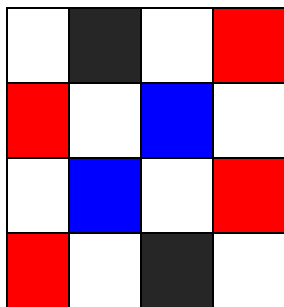
При записи изображения в память компьютера кроме цвета отдельных точек необходимо фиксировать много дополнительной

информации – размеры рисунка, яркость точек и т. д. Конкретный способ кодирования всей требуемой при записи изображения информации образует графический формат. Форматы кодирования графической информации, основанные на передаче цвета каждого отдельного пикселя, из которого состоит изображение, относят к группе растровых или BitMap форматов (bit map – битовая карта).

### Кодирование растровых изображений



1	0	0	1
0	1	1	0
0	1	0	0
1	0	0	1



00	11	00	01
01	00	10	00
00	10	00	01
01	00	11	00

Растровое изображение представляет собой совокупность точек (пикселей) разных цветов.

Наиболее известными растровыми форматами являются BMP, GIF и JPEG форматы. В формате BMP (от BitMap) задается цветность всех пикселей изображения. При этом можно выбрать монохромный режим с 256 градациями или цветной с 16 256 или 16 777 216 цветами. Этот формат требует много памяти. В формате GIF (Graphics Interchange Format – графический формат обмена) используются специальные методы сжатия кода, причем

поддерживается только 256 цветов. Качество изображения немного хуже, чем в формате BMP, зато код занимает в десятки раз меньше памяти. Формат JPEG (Joint Photographic Experts Group - Уединенная группа экспертов по фотографии) использует методы сжатия, приводящие к потерям некоторых деталей. Однако поддержка 16 777 216 цветов все-таки обеспечивает высокое качество изображения. По требованиям к памяти формат JPEG занимает промежуточное положение между форматами BMP и GIF.

### **Кодирование векторных изображений**

Векторное изображение представляет собой совокупность графических примитивов (точка, отрезок, эллипс...). Каждый примитив описывается математическими формулами. Кодирование зависти от прикладной среды.

Растровая графика обладает существенным недостатком – изображение, закодированное в одном из растровых форматов, очень плохо “переносит” увеличение или уменьшение его размеров – масштабирование. Для решения задач, в которых приходится часто выполнять эту операцию, были разработаны методы так называемой векторной графики. В векторной графике, в отличие от основанной на точке – пикселе – растровой графики, базовым объектом является линия. При этом изображение формируется из описываемых математическим, векторным способом отдельных отрезков прямых или кривых линий, а также геометрических фигур – прямоугольников, окружностей и т. д., которые могут быть из них получены. Фирма Adobe разработала специальный язык PostScript

(от poster script – сценарий плакатов, объявлений, афиш), служащий для описания изображений на базе указанных методов. Этот язык является основой для нескольких векторных графических форматов. В частности, можно указать форматы PS (PostScript) и EPS, которые используются для описания как векторных, так и растровых изображений, а также разнообразных текстовых шрифтов. Изображения и тексты, записанные в этих форматах, большинством популярных программ не воспринимаются, они могут просматриваться и печататься только с помощью специализированных аппаратных и программных средств.

Кроме растровой и векторной графики существует еще и фрактальная графика, в которой формирование изображений целиком основано на математических формулах, уравнениях, описывающих те или иные фигуры, поверхности, тела. При этом само изображение в памяти компьютера фактически не хранится – оно получается как результат обработки некоторых данных. Таким способом могут быть получены даже довольно реалистичные изображения природных ландшафтов.

Для успешного решения задач этой темы необходимо понимать взаимосвязь информационного объема растрового изображения, его пространственного и цветового разрешения, а также принципы кодирования цвета элементов растра.

## Справочная информация

Количество цветов		1 пиксель
2 (черное, белое)	$2^1$	1 бит
16	$2^4$	4 бита
256	$2^8$	8 бит
65536 (high color)	$2^{16}$	16 бит
16 777 216 (true color)	$2^{24}$	24 бита

Чтобы запомнить, сколькими битами кодируется 1 пиксель в палитре, достаточно запомнить формулу:  $P=2^b$ , где P- это количество цветов в палитре, а b – это количество бит для кодирования одного пикселя.

### ***II.3. Обработка информации в электронных таблицах***

#### **II.3.1. Основные понятия электронных таблиц**

Программа Excel предназначена для работы с таблицами данных, преимущественно числовых. При формировании таблицы выполняют ввод, редактирование и форматирование текстовых и числовых данных, а также формул. Наличие средств автоматизации облегчает эти операции. Созданная таблица может быть выведена на печать. Документ Excel называется рабочей книгой. Рабочая книга представляет собой набор рабочих листов, каждый из которых имеет табличную структуру и может содержать одну или несколько таблиц. В окне документа в программе Excel

отображаются только текущий рабочий лист, с которым ведется работа.

Каждый рабочий лист имеет название, которое отображается на ярлыке листа, отображаемом в его нижней левой части. С помощью ярлычков можно переключаться к другим листам, входящим в ту же самую рабочую книгу. Чтобы переименовать рабочий лист, надо дважды щелкнуть на его ярлыке. Рабочий лист состоит из строк и столбцов. Столбцы озаглавлены прописными латинскими буквами, а далее 2-х буквенными комбинациями. Всего рабочий лист может содержать до 256 столбцов, пронумерованных от A до IV. Строки последовательно нумеруются цифрами, от 1 до 65536. На пересечении столбцов и строк образуются ячейки рабочей таблицы. Они являются минимальными элементами для хранения данных. Обозначение отдельной ячейки сочетает в себе номера столбца и строки, например, B3. Такое обозначение выполняет функцию ее адреса. Адреса ячеек используются при записи формул. Выделенная рамкой ячейка называется активной. Все операции ввода и редактирования выполняются в активной ячейке. На данные, расположенные в соседних ячейках можно ссылаться в формулах, как на единое целое. Такую группу ячеек называют диапазоном. Наиболее часто используются прямоугольные диапазоны, образующие на пересечении группы последовательно идущих строк и группы последовательно идущих столбцов. Для выделения прямоугольного диапазона следует протянуть указатель от одной угловой линии до противоположной по диагонали.



## **Ввод, редактирование и форматирование данных.**

Отдельная ячейка может содержать данные, относящиеся к одному из трех типов: текст, число или формула, а также оставаться пустой. Программа Excel при сохранении рабочей книги записывает в файл только прямоугольную область рабочих листов, примыкающую к левому верхнему углу (ячейка A1) и содержащую все заполненные ячейки.

Ввод формулы всегда начинается с символа «=» (знак равенства). Ввод данных осуществляют непосредственно в текущую ячейку или в строку формул, располагающуюся в верхней части окна программы непосредственно под панелями инструментов. Место ввода отмечается текстовым курсором. Если начать ввод нажатием алфавитно-цифровых клавиш, данные из текущей ячейки заменяются вводимым текстом. Если щелкнуть на строку формул или дважды на текущей ячейке, старое содержимое ячейки не удаляется и появляется возможность его редактирования. Чтобы завершить ввод, сохранив введенные данные используют кнопку *Enter* в строке формул или саму клавишу *Enter*. Чтобы отменить внесенные изменения и восстановить прежние значения ячейки используют кнопку *Отмена* в строке формул или клавишу *Esc*. Для очистки текущей ячейки или выделенного диапазона проще всего использовать клавишу *Delete*. Текстовые данные по умолчанию выравниваются по левому краю ячейки, а числа по правому. Чтобы изменить формат отображения данных в текущей ячейке или выбранном диапазоне используют команду: **Формат** →

Ячейки. Вкладки этого диалогового окна позволяют выбирать формат записи данных.

Вычисления в таблицах программы Excel осуществляются при помощи формул. Формула может содержать числовые константы, ссылки на ячейки и функции Excel, соединенные знаками математических операций. Скобки позволяют изменять стандартный порядок выполнения действий. Если ячейка содержит формулу, то в рабочем листе отображается текущий результат вычисления этой формулы. Если сделать ячейку текущей, т.е. в ячейке находится курсор, то сама формула отображается в строке формул. называется текущей. Правило использования формул в программе Excel состоит в том, что если значение ячейки действительно зависит от других ячеек таблицы, всегда следует использовать формулу, даже если операцию легко можно выполнить в «уме». Это гарантирует, что последующее редактирование таблицы не нарушит ее целостности и правильности производимых в ней вычислений. Формула может содержать ссылки, т.е. адреса ячеек, содержание которых используются в вычислениях. Это означает, что результат вычисления формулы зависит от числа, находящегося в другой ячейке. Ячейка, содержащая формулу, таким образом, является зависимой.

### **Использование стандартных функций**

Стандартные функции используются в программе Excel только в формулах. Вызов функции состоит в указании в формуле *Имени* функции, после которого в скобках указывается *Список*

*параметров*. Отдельные параметры разделяются в списке точкой с запятой. В качестве параметра может использоваться число, адрес ячейки или произвольные выражения, для вычисления которых также могут использоваться функции.

**Палитра формул.** Если начать ввод формулы щелчком на кнопке *Изменить формулу* в строке формул, под строкой формул появляется *Палитра формул*, обладающая свойствами диалогового окна. Она содержит значение, которое получится, если немедленно закончить ввод формулы. В левой части строки, где раньше располагался номер текущей ячейки, теперь появляется раскрывающийся список функций. Он содержит десять функций, которые использовались последними, а также пункт *Другие функции*. При выборе пункта *Другие функции* запускается *Мастер функций*, облегчающий выбор нужной функции. В списке *Категории* выбирается категория, к которой относятся функции, а в списке *Функция* – конкретная функция данной категории. После щелчка на кнопке *Ок* *Имя функции* заносится в строку формул вместе со скобками, ограничивающими список параметров. Текстовый курсор устанавливается между этими скобками. В ходе ввода параметров функции палитра формул изменяет вид. Если название параметра указано полужирным шрифтом, то параметр является *обязательным* и соответственно поле должно быть заполнено.

### **П.3.2. Применение электронных таблиц для расчетов**

Итоговые вычисления предполагают получение числовых характеристик, описывающих определенный набор данных в

целом. Например, возможно вычисление суммы значений, входящих в набор, среднего значения и других статистических характеристик, количества или доли элементов набора, удовлетворяющих определенным условиям. Для итоговых вычислений применяют ограниченный набор функций, наиболее типичной из которых является функция суммирования (СУММ). Это единственная функция, для применения которой есть отдельная кнопка на стандартной панели инструментов (кнопка Автосумма  $\Sigma$ ).

Прочие функции для итоговых вычислений выбираются отдельным образом, с помощью раскрывающегося списка в строке формул или с помощью мастера функций. Все эти функции относятся к категории – статистические. В их число входят функции ДИСП (вычисляет дисперсию), МАКС (максимальное число в диапазоне), СРЗНАЧ (среднее арифметическое значение), СЧЕТ (подсчет ячеек с числами в диапазоне). Функции итоговых вычислений часто применяют при использовании таблицы Excel в качестве базы данных.

В программе Excel термин диаграмма используется для обозначения всех видов графического представления числовых данных. Построение графического изображения производится на основе ряда данных. Так называют группу ячеек с данными в пределах отдельной строки или столбца. Диаграмма представляет собой вставной OLE-объект, внедренный на один из листов рабочей книги. Для построения диаграммы используют *Мастер диаграмм*, запускаемый щелчком на кнопке *Мастер диаграмм* на

стандартной панели инструментов. На первом этапе работы мастера выбирают форму диаграммы. Доступные формы перечислены в списке *Тип* на вкладке стандартные. Для выбранного типа справа указывается несколько вариантов представления данных (*Палитра Вид*). Второй этап работы мастера служит для выбора данных, по которым будет строиться диаграмма. Третий этап работы мастера (после щелчка на кнопке *Далее*) состоит в выборе оформления диаграммы, где указывается название диаграммы, отображаются маркировки осей координат, сетки линий. На последнем этапе работы мастера (после щелчка на кнопке *Далее*) указывается, следует ли использовать для размещения диаграммы новый рабочий лист или один из имеющихся. Для того, чтобы открыть диалоговое окно для форматирования элемента диаграммы можно через меню *Формат* (для выделенного элемента) или через контекстное меню (команда *Формат*). Для удаления диаграммы используется клавиша *Delete*.

## **Примеры обработки информации в электронных таблицах**

### ***Работа с формулами, абсолютные и относительные ссылки***

В ячейке электронной таблицы может храниться не только значение, введенное пользователем, но и формула, по которой автоматически рассчитывается значение ячейки. Для того, чтобы электронная таблица могла производить вычисления по формулам, как мы отмечали выше они должны быть записаны по определенным правилам. В формулах могут использоваться адреса других ячеек и различные функции, в том числе математические

**sin**, **cos**, **exp** (показательная функция с основанием, равным числу  $e$ ), **abs** (модуль числа) и так далее.

В электронных таблицах принято следующее правило: обычные адреса ячеек в формулах являются относительными. Это означает, что при копировании ячейки, содержащей формулу, в ячейку, отстоящую от исходной на некоторое число столбцов и строк, адреса ячеек в формуле изменяются на такое же число столбцов и строк. Пусть, например, ячейка B2 содержит формулу  $=C2+1$ . При копировании ячейки формула изменится следующим образом:

	A	B	C	D
1	$=B1+1$		$=D1+1$	
2		$=C2+1$		$=E2+1$
3	$=B3+1$			
4			$=D4+1$	

Чтобы адрес не менялся при копировании, он должен быть абсолютным. В абсолютном адресе перед обозначениями строки и столбца ставится знак  $\$$ . Если знак  $\$$  стоит только перед именем столбца, то при копировании будет сохраняться имя столбца, если перед номером строки, то номер строки.

Пусть ячейка B2 содержит формулу  $=\$C\$2+\$C3+C\$4$ . При копировании ячейки формула изменится следующим образом:

	A	B	C	D
1	$=\$C\$2+\$C2+B\$4$		$=\$C\$2+\$C2+B$	
2		$=\$C\$2+\$C3+B\$4$		$=\$C\$2+\$C3+E\$4$
3	$=\$C\$2+\$C4+B\$4$			
4			$=\$C\$2+\$C5+D\$4$	

### Пример 1.

В ячейке A1 электронной таблицы записана формула =D1-\$D2. Какой вид приобретет формула после того, как ячейку A1 скопируют в ячейку B1?

- 1) =E1-\$E2    2) =E1-\$D2    3) =E2-\$D2    4) =D1-\$E2

**Решение:**

Адрес B1 получен из адреса A1 сдвигом на 1 вправо. Также изменятся все относительные адреса столбцов в адресах формулы, а именно, D1 преобразуется в E1. Адреса строк не изменятся, так как формула копируется в пределах одной строки (первой). Адрес \$D2 не изменится, так как здесь адрес столбца абсолютный, и формула приобретет вид =E1-\$D2.

**Ответ: 2.**

**Пример 2.**

Дан фрагмент электронной таблицы:

	A	B	C	D
1	5	2	4	
2	10	1	6	

В ячейку D2 введена формула =A2\*B1+C1

В результате в ячейке D2 появится значение:

- 1) 6      2) 14      3) 16      4) 24

**Решение:**

$$D2=10 \cdot 2+4=24$$

**Ответ: 4**

**Пример 3.**

В электронной таблице значение формулы =СУММ(B1:B2) равно 5. Чему равно значение ячейки B3, если значение формулы =СРЗНАЧ(B1:B3) равно 3?

**Решение:**

Диапазону В1:В3 принадлежат три ячейки В1, В2 и В3. Формула СРЗНАЧ вычисляет среднее арифметическое ячеек диапазона, поэтому  $(В1+В2+В3):3 = 3$ . Следовательно  $В1+В2+В3 = 9$ . Учитывая, что по условию  $В1+В2 = 5$ , получаем, что  $В3 = 4$ .

**Диаграммы и графики**

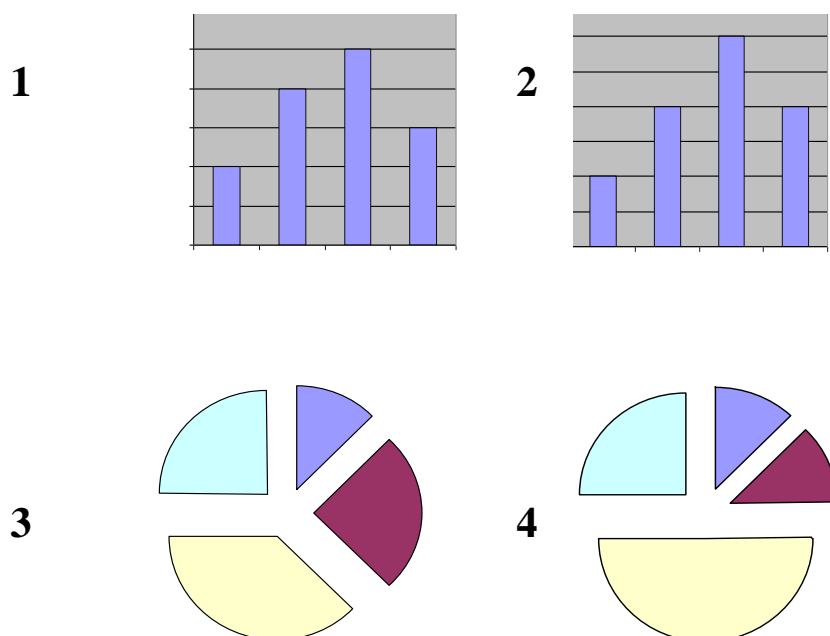
Для решения задач этой темы важно уметь правильно интерпретировать столбчатые и круговые диаграммы, учитывая, что простые столбчатые диаграммы обычно отражают абсолютные значения, а круговые - процентный состав.

**Пример 4.**

Дан фрагмент электронной таблицы:

	А	В
1	=А4-В1	1
2	=А1+2	2
3	=А2+В1	
4	=В1+В2	

После выполнения вычислений была построена диаграмма по значениям диапазона ячеек А1:А4. Укажите получившуюся диаграмму.





**Решение:**

В диапазон A1:A4 входят ячейки A1, A2, A3, A4. Прежде всего, следует вычислить значения ячеек указанного диапазона. Мы можем вычислить значение формулы только в том случае, если известны значения всех фигурирующих в ней ячеек. Поэтому, порядок заполнения таблицы будет таков:

	A	B
1	=A4-B1	1
2	=A1+2	2
3	=A2+B1	
4	3	

	A	B
1	2	1
2	=A1+2	2
3	=A2+B1	
4	3	

	A	B			A	B
1	2	1		1	2	1
2	4	2		2	4	2
3	=A2+B1			3	5	
4	3			4	3	

Итак, значения ячеек диапазона A1:A4 равны 2; 4; 5; 3 соответственно. Такому набору значений соответствует диаграмма под номером 1.

**Ответ: 1.**

## **II.4. Базы данных**

В задачах на эту тему обычно требуется вручную выполнить заданный поисковый запрос к предложенному фрагменту базы данных или самостоятельно сформулировать такой запрос исходя из поставленной цели.

Разберем примеры таких задач.

**Пример 1.**

Сколько записей в нижеследующем фрагменте турнирной таблицы удовлетворяют условию «Место  $\leq 4$  И ( $V > 4$  ИЛИ  $M3 > 12$ )»?

Место	Команда	В	Н	П	О	МЗ	МП
1	Боец	5	3	1	18	9	5
2	Авангард	6	0	3	18	13	7
3	Опушка	4	1	4	16	13	7
4	Звезда	3	6	0	15	5	2
5	Химик	3	3	3	12	14	17
6	Пират	3	2	4	11	13	7

1) 5

2) 2

3) 3

4) 4

**Решение:**

Решая данную задачу, необходимо последовательно применять условие к каждой строке таблицы. Условию удовлетворяют команды: Боец, Авангард, Опушка. У Звезды недостаточно выигрышей и забитых мячей, Химик и Пират заняли слишком низкие места.

**Ответ: 3.****II.5. Телекоммуникационные технологии****II.5.1. Адресация в сети Интернет****Межсетевой протокол (IP)**

С помощью линий связи обеспечивается доставка данных из одного пункта в другой. Различные участки Internet связываются с помощью системы компьютеров (называемых *маршрутизаторами*) соединяющих между собой сети. Это могут быть сети Internet, сети с маркерным доступом, телефонные линии (см. рис.2).

## The Internet

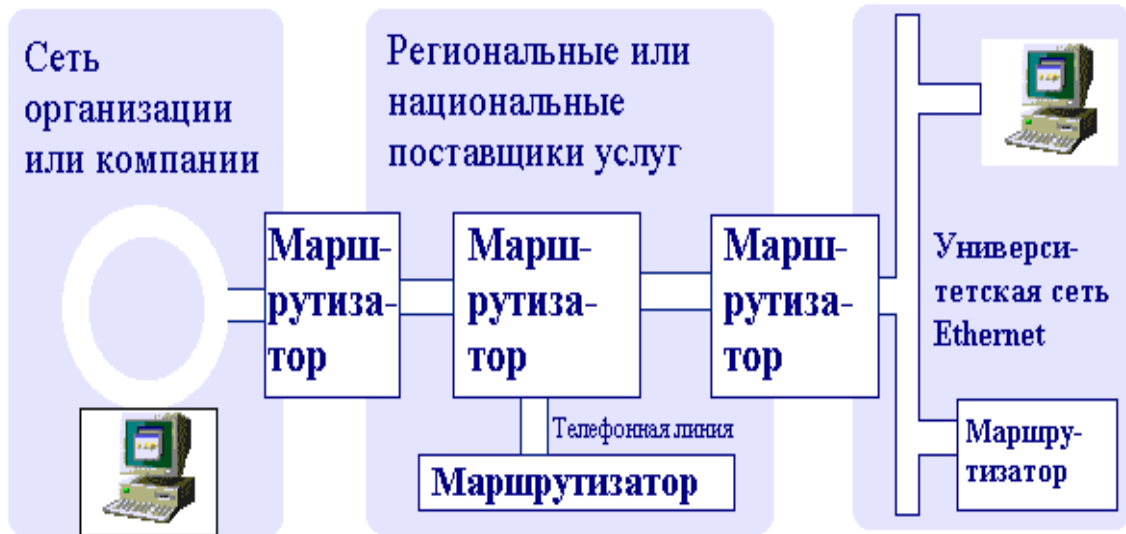


Рис. 2. Аппаратные средства Internet.

Определенные правила регламентируют порядок работы Internet. Эти правила называют **протоколами**. **Межсетевой протокол (Internet Protocol, IP)** отвечает за адресацию, т.е. гарантирует, что маршрутизатор знает, что делать с Вашими данными, когда они поступят. Некоторая адресная информация приводится в начале Вашего сообщения. Она даёт сети достаточно сведений для доставки пакета данных.

Internet-адреса состоят из четырёх чисел, каждое из которых не превышает 256. При записи числа отделяются одно от другого точками, например:

192.112.36.5

128.174.5.6

Адрес фактически состоит из нескольких частей. Поскольку Internet – это сеть сетей, то начало адреса содержит информацию для маршрутизаторов о том, к какой сети относится ваш компьютер. Правая часть адреса служит для того, чтобы сообщить

сети, какой компьютер должен получить этот пакет. Каждый компьютер в Internet имеет свой уникальный адрес.

Поэтому следующий уровень сети даст нам возможность пересылать более крупные порции информации и позаботиться об устранении тех искажений, которые вносит сама сеть. Для решения упомянутых выше проблем используется “протокол управления передачей” (Transmission Control Protocol, TCP), который часто упоминают вместе с протоколом IP.

Информацию, которую Вы хотите передать, TCP разбивает на порции. Каждая порция нумеруется, чтобы можно было проверить, вся ли информация получена, и расположить данные в правильном порядке. Для передачи этого порядкового номера по сети у протокола есть свой собственный “конверт”, на котором “написана” необходимая информация. Порция Ваших данных помещается в конверт TCP. Конверт TCP, в свою очередь, помещается в конверт IP и передается в сеть.

В Интернете для передачи данных различных форматов существуют следующие протоколы:

- *http* (hyper text transfer protocol) – протокол передачи гипертекста, т.е. *Web*-страницы;
- *ftp* (file transfer protocol) – протокол пересылки бинарных или текстовых файлов;
- *smtp* (simple mail transfer protocol) – протокол пересылки электронной почты;
- *pop* (post office protocol) – протокол хранения и получения электронной почты;

- *nntp* (net news transfer protocol) – протокол для телеконференций новостей;
- *telnet* – простая эмуляция терминала для удаленной работы с сервером;
- *usenet* - протокол телеконференций;
- *wais* - протокол баз данных.

Темпы роста Интернета потребовали наличия распределённой интерактивной системы. Эта система называется “доменной системой имён” (Domain Name System, DNS).

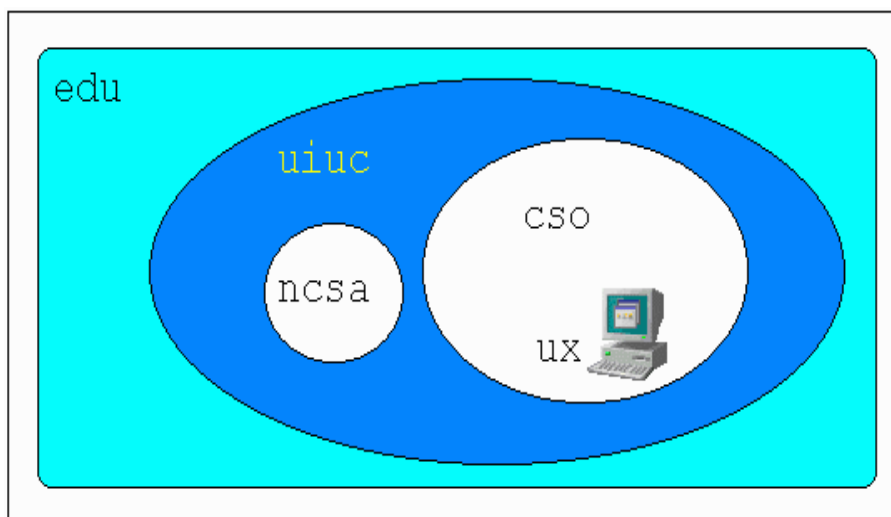
Доменная система имён представляет собой метод назначения имён путём возложения на разные группы пользователей ответственности за подмножества имён. Каждый уровень в этой системе называется доменом. Домены отделяются один от другого точками:

ix.cso.uiuc.edu

nic.ddn.mil

yoyodyne.com

В имени может быть любое число доменов, но более пяти встречается редко. Каждый последующий домен в имени (если смотреть слева направо) больше предыдущего. В имени ix.cso.uiuc.edu элемент ix – имя реального компьютера с IP - адресом. (См. рисунок 3).



*Рис. 3. Структура доменного имени.*

Имя данного компьютера создано и курируется группой `cso`, которая есть не что иное, как отдел, в котором стоит этот компьютер. Отдел `cso` является отделом университета штата Иллинойс (`uiuc`). `Uiuc` входит в национальную группу учебных заведений (`edu`). Таким образом, домен `edu` включает в себя все компьютеры учебных заведений США; домен `uiuc.edu` – все компьютеры университета штата Иллинойс и т.д.

Каждая группа может создавать и изменять все имена, находящиеся под её контролем. Если `uiuc` решит создать новую группу и назвать её `nisa`, она может ни у кого не спрашивать разрешения. Всё, что нужно сделать – это добавить новое имя в свою часть всемирной базы данных, и рано или поздно тот, кому нужно, узнает об этом имени (`nisa.uiuc.edu`). Аналогичным образом `cso` может купить новый компьютер, присвоить ему имя и включить в сеть, не спрашивая ни у кого разрешения. Если все группы, начиная с `edu` и ниже, будут соблюдать правила, и обеспечивать уникальность имён, то никакие две системы в Internet не будут иметь одинакового имени. У Вас могут быть два

компьютера с именем fred, но лишь при условии, что они находятся в разных доменах (например, fred.cso.uiuc.edu и fred.ora.com).

Легко узнать, откуда берутся домены и имена в организации типа университета или предприятия. Но откуда берутся домены “верхнего уровня” типа edu? Они были созданы, когда была изобретена доменная система. Изначально было шесть организационных доменов высшего уровня.

*Таблица 1. Первоначальные домены верхнего уровня.*

№	Домен	Использование
1.	Com	Коммерческие организации
2.	Edu	Учебные заведения (университеты, средние школы и т.д.)
3.	Gov	Правительственные учреждения (кроме военных)
4.	Mil	Военные учреждения (армия, флот и т.д.)
5.	Org	Прочие организации
6.	Net	Сетевые ресурсы

Когда Internet стала международной сетью, возникла необходимость предоставить зарубежным странам возможность контроля за именами находящихся в них систем. Для этой цели создан набор двухбуквенных доменов, которые соответствуют доменам высшего уровня для этих стран. Поскольку ca – код

Канады, то компьютер на территории Канады может иметь такое имя: hockey.guelph.ca

Общее число кодов стран – 300; компьютерные сети существуют приблизительно в 170 из них.

Окончательный план расширения системы присвоения имён ресурсов в Internet был наконец-то объявлен комитетом IАНС (International Ad Hoc Committee). Согласно новым решениям, к доменам высшего уровня, включающим сегодня com, net, org, прибавятся:

- firm – для деловых ресурсов Сети;
- store – для торговли;
- web – для организаций, имеющих отношение к регулированию деятельности в WWW;
- arts – для ресурсов гуманитарного образования;
- rec – игры и развлечения;
- info – предоставление информационных услуг;
- nom – для индивидуальных ресурсов, а также тех, кто ищет свои пути реализации, которые отсутствуют в приведённом убогом списке.

Каждый компьютер, подключенный к сети Интернет, имеет два равноценных уникальных адреса: IP-адрес и доменный адрес.

Электронная почта служит для передачи текстовых сообщений в пределах Интернета, а также между другими сетями электронной почты.

Адреса организованы иерархически как дерево доменов. Домены первого уровня обозначают страну или область



деятельности. Например, **de** - Германия, **org** - учебные организации и пр. Домен второго уровня обозначает одно из имен провайдера, который оказывает услуги по подключению к Интернету, например, gmail.com. При образовании адреса электронной почты к зарегистрированному имени пользователя добавляется символ @ и затем имя домена узла провайдера.

Например: **zoro@trans.com**, где **zoro** - имя пользователя, **trans.com** - почтовый сервер, @ - разделительный символ.

### *Примеры. Адресация в сети Интернет*

Web-страницы в Интернет размещаются на Web-серверах в виде текстовых файлов. Входящие в состав страниц изображения, звуки и другие компоненты тоже размещаются на сервере в виде отдельных файлов. Файлы страниц и их компонентов имеют свои адреса (идентификаторы), называемые унифицированными указателями ресурсов (URL— Uniformed Resource Locator).

Пример URL -адреса: <http://www.mail.ru/chair86/index.htm>

URL-адрес начинается с обозначения протокола Интернета, с помощью которого осуществляется доступ к ресурсу. В данном случае указан HTTP - протокол передачи гипертекста (HyperText Transfer Protocol). После названия протокола следуют двоеточие и две наклонных черты. Далее указывается имя компьютера (www.mail.ru). После имени компьютера следует полное имя файла с ресурсом на сервере, символы "/" разделяют названия каталогов, которые нужно посетить, чтобы добраться до файла. В данном случае страница хранится на сервере в файле index.htm каталога chair86.

## ИСПОЛЬЗУЕМАЯ ЛИТЕРАТУРА

1. Безручко В.Т. Практикум по курсу «Информатика». М., 2009.
2. Зорин М.В., Е.М. Зорина, Информатика. Тестирование в формате ЕГЭ. Реком. по решению заданий. Волгоград. изд. «Учитель». 2009. 223 с.
3. Информатика в школе. Приложение к журналу «Информатика и образование». №2-2006. –М. Образование и информатика, 2006. -96с.
4. Информатика. Универсальные материалы для подготовки учащихся. ЕГЭ. ФИПИ, 2009
5. Информатика: Основы компьютерной грамоты. Начальный курс/ Под ред. Н.В. Макаровой. - СПб.: Питер, 2005.-160с.: ил.
6. Культин Н.Б. TurboPascal в задачах и примерах.-СПб.:БХВ-Петербург, 2005. -256с.
7. Культин Н.Б. Основы программирования в Delphi.-СПб.:БХВ-Петербург, 2006, -608с.
8. Математика и информатика. под ред.В.Д.Будаева и Н.П.Стефановой. учеб. пособ. для ст. пед.вузов. М., 2004.
9. Могилев А.В. Н.И. Пак, Е.К. Хеннер. Информатика. Учебное пособие. 2004.-848с.
10. Могилев А.В. Н.И. Пак, Е.К. Хеннер. Практикум по программированию. Учебное пособие. 2005. -544с/
11. Немнюгин С.А. Turbo Pascal. Учебник. изд. «Питер», 2000.
12. Немнюгин С.А. Практикум по Turbo Pascal. учебное пособие. Изд. «Питер», 2003.

13. Острейковский В.А., Полякова И.В. Информатика. М.: Оникс. 2008. -608с.
14. Петров М.Н., Молочков В.П. Компьютерная графика., 2-ое издание. учебник для вузов. СПб: Питер, 2004. –811с.
15. Сафронов И.К. ЕГЭнциклопедия. Информатика. СПб.:БХВ-Петербург, 2010, -496с.
16. Семакин И.Г., Залогова Л.А., Русаков С.В. и др. Информатика. Бином, -2005, -224с.
17. Семакин И.Г., Хеннер Е.К. Информационные системы и модели. Практикум.-М. -2006. -87 с.
18. Семакин И.Г., Шестаков А.П. Основы программирования. Учебник, изд. «Мастерство», М. 2008.
19. Соболев. Информатика. Учебник. Ростов-на-Дону. –Феникс, 2007. -446с.
20. Таненбаум Э. Архитектура компьютера. 5-е изд. – СПб.:Питер, 2007. -844с.
21. Угринович Н. Информатика и информационные технологии. М., учебник для 10-11 кл. 2003.
22. Фаронов В. Программирование баз данных в Delphi 7. Учебный курс. –СПб.:Питер, 2005. -459с.
23. Фаронов В.В. Delphi. Программирование на языке высокого уровня: Учебник для вузов – СПб.:Питер, 2006.-640с.
24. Фигурнов В.Э, IBM для пользователя. Краткий курс. - М.: ИНФРА-М.. 2006.

# ИНФОРМАТИКА

Учебное пособие

План университета 2017 г., поз.30

Редактор - Н.В. Ефрюкова

Корректор Е.М. Плуговая

Компьютерный набор М.М. Бостанова, З.К. Джаубаева

Компьютерная верстка М.М. Бостанова, З.К. Джаубаева

Подписано в печать

Формат 60x84/16

Бумага типографская

Объем: 10,3 физ.печ.л., 9,6 усл.печ.л., 8,9 уч.изд.л.

Тираж 300 экз.

Издательство Карачаево-Черкесского  
государственного университета им. У.Д. Алиева  
369202 г. Карачаевск, ул. Ленина, 29  
Лицензия ЛР №040310 от 21.10.1997

Отпечатано в типографии Карачаево-Черкесского  
государственного университета им.У.Д. Алиева

369202, г. Карачаевск, ул. Ленина, 46