

Макшидовская М.А.



Москва, 2019

Школа №109



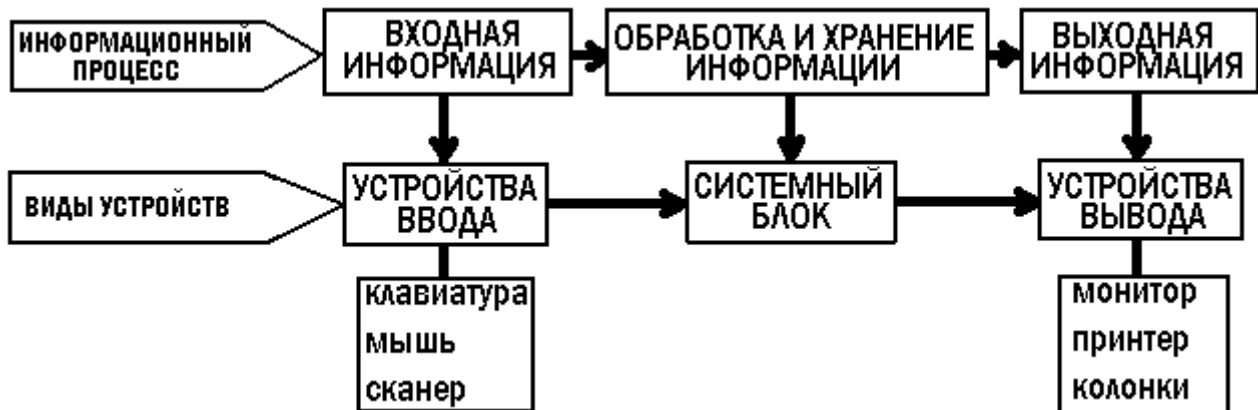
Состав компьютера. Системы счисления.

Перевод чисел из одной системы счисления в другую

Темы для самостоятельного повторения:

1. Состав компьютера (материал на сайте <http://matem-109.ru>, Информатика, Материалы, Виртуальный учебник «Состав компьютера»).

Схема основных устройств компьютера (минимальный набор)



2. Непозиционные и позиционные системы счисления (материал на сайте <http://matem-109.ru>, Информатика, Материалы, Системы счисления).

Повторение:

В позиционной системе счисления любое целое число может быть представлено:

в развёрнутой форме	в свёрнутой (естественной) форме
$A_p = \pm a_k \cdot p^k + a_{k-1} \cdot p^{k-1} + \dots + a_1 \cdot p^1 + a_0 \cdot p^0$	$A_p = \pm (a_k a_{k-1} \dots a_1 a_0)_p$

Перевод целого числа из двоичной системы в десятичную:

Пример: $10010_2 = 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 18_{10}$.

Перевод целого десятичного числа в двоичную систему методом деления на новое основание:

Правило перевода	Десятичное число делится на основание 2 двоичной системы счисления нацело, из остатков записывается двоичное число:	Ответ: $12_{10} = 1100_2$
-------------------------	---	---

Перевод целого десятичного числа в двоичную систему методом подбора степени 2:

Пример. Переведём число 273_{10} в двоичное число. Для этого 273 представим как сумму степеней числа 2 – ближайшая степень $2^8 = 256$ ($273 - 256 = 17$), следующая – $2^4 = 16$ ($17 - 16 = 1$), следующая – $2^0 = 1$. Вместо $2^7, 2^6, 2^5, 2^3, 2^2, 2^1$ ставим 0:

$$273_{10} = 256 + 0 + 0 + 0 + 16 + 0 + 0 + 0 + 1 =$$

$$1 \cdot 2^8 + 0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 =$$

$$100010001_2$$

Перевод чисел из одной системы счисления в другую.

Для выполнения заданий вам потребуется таблица соответствия десятичной, шестнадцатеричной восьмеричной и двоичной систем счисления (см. Приложение 2, стр. 54).

Перевод целого числа из шестнадцатеричной и восьмеричной систем в десятичную:

где p – основание системы счисления, k, n – номер цифры в числе и степень, в которую возводится основание системы счисления.

Примеры:

$$1234_{10} = 1 \cdot 1000 + 2 \cdot 100 + 3 \cdot 10 + 4 \cdot 1 = 1 \cdot 10^3 + 2 \cdot 10^2 + 3 \cdot 10^1 + 4 \cdot 10^0.$$

свёрнутая форма

развёрнутая форма

$$153_8 = 1 \cdot 8^2 + 5 \cdot 8^1 + 3 \cdot 8^0 = 107_{10}.$$

$$2B1_{16} = 2 \cdot 16^2 + 11 \cdot 16^1 + 1 \cdot 16^0 = 689_{10}.$$

Перевод десятичного числа в восьмеричную и шестнадцатеричную системы

Правило перевода целого числа	Десятичное число делится на основание новой системы счисления нацело, из остатков записывается число в соответствующей системе счисления
В шестнадцатеричную $\begin{array}{r l} 331 & 16 \\ \hline 320 & 20 \quad 16 \\ \hline 11 & 16 \quad 1 \quad 16 \\ \hline B & 4 \quad 0 \quad 0 \\ & 1 \end{array}$ ← старший разряд	В восьмеричную $\begin{array}{r l} 461 & 8 \\ \hline 456 & 57 \quad 8 \\ \hline 5 & 56 \quad 7 \quad 8 \\ \hline & 1 \quad 0 \quad 0 \\ & 7 \end{array}$ ← старший разряд
Ответ: $331_{10} = 14B_{16}$	Ответ: $461_{10} = 715_8$

Перевод двоичного числа в шестнадцатеричную и восьмеричную системы счисления

Алгоритм перевода целого числа	Исходное двоичное число разбивается либо на тетрады (по четыре цифры, начиная справа), либо на триады (по три цифры, начиная справа).
Алгоритм перевода дробного числа	Исходное двоичное число разбивается либо на тетрады (по четыре цифры, начиная слева), либо на триады (по три цифры, начиная слева).

См. Приложение 1 (Таблица соответствия систем счисления).

Пример 1. Перевести 1001111_2 в шестнадцатеричную систему:

- 1) Разбиваем на тетрады (справа налево): $1001111 = 100\ 1111 = 0100\ 1111$;
- 2) По таблице соответствия: $0100 = 4$, $1111 = F$;
- 3) Ответ: $1001111_2 = 4F_{16}$.

Пример 2. Перевести 1001111_2 в восьмеричную систему:

- 1) Разбиваем на триады (справа налево): $1001111 = 1\ 001\ 111 = 001\ 001\ 111$
- 2) По таблице соответствия: $001 = 1$, $111 = 7$;
- 3) Ответ: $1001111_2 = 117_8$.

Задания:

- а) Перевести в десятичное число: 10101_2 ; 201_8 ; $1CF_{16}$.
- б) Перевести десятичное число в двоичное методом подбора степеней числа 2: 132_{10} , 573_{10} , 333_{10} .
- в) Перевести числа 78_{10} ; 91_{10} ; 102_{10} и 125_{10} в двоичную, шестнадцатеричную и восьмеричную системы.
- г) Перевести числа 101011011_2 ; $1010001,11001_2$; 10010001110_2 в восьмеричную и шестнадцатеричную системы счисления.



Сравнение чисел в различных системах счисления

Чтобы сравнить числа, заданные в различных системах счисления, необходимо представить их в одной системе счисления.

Пример. Сравните числа 134_8 и $2C_{16}$.

Решение 1. Представим оба числа в десятичной системе:

$$134_8 = 1 \cdot 8^2 + 3 \cdot 8^1 + 4 \cdot 8^0 = 92_{10}, \quad 2C_{16} = 2 \cdot 16^1 + 12 \cdot 16^0 = 44_{10}.$$

Так как $92_{10} > 44_{10}$, то $134_8 > 2C_{16}$.

Решение 2. Представим оба числа в двоичной системе:

$$134_8 = 1011100_2, \quad 2C_{16} = 101100_2. \text{ Так как } 1011100_2 > 101100_2, \text{ то } 134_8 > 2C_{16}.$$

Решение 3. Представим число $2C_{16}$ в восьмеричной системе:

$$2C_{16} = 101100_2 = 54_8. \text{ Так как } 134_8 > 54_8, \text{ то } 134_8 > 2C_{16}.$$

Задания:

а) Сравните:

10001001_2 и 10101_{16} ; 225_{10} и 225_{16} ; 101010_8 и 1111_{10} ; $F45_{16}$ и 5432_8 .

б) Перечислите целые десятичные числа, которые удовлетворяют неравенству: $1111101_2 < x < 82_{16}$.

Арифметические действия в двоичной системе счисления

Рассмотрим основные арифметические действия (сложение, вычитание, умножение) в двоичной системе счисления. Для этого удобно пользоваться специальными таблицами: сложения и умножения.

Таблица сложения

+	0	1
0	0	1
1	1	10

Таблица умножения

*	0	1
0	0	0
1	0	1

Пример 1. Выполните сложение:

$$1001000_2 + 11011_2 = 1100011_2$$

$$\begin{array}{r} 1001000 \\ + 11011 \\ \hline 1100011 \end{array}$$

Пример 2. Выполните вычитание:

$$1001000_2 - 11011_2 = 101101_2$$

$$\begin{array}{r} 1001000 \\ - 11011 \\ \hline 101101 \end{array}$$

Пример 3. Выполните умножение:

$$10010_2 \cdot 1101_2 = 11101010_2$$

$$\begin{array}{r} 10010 \\ \times 1101 \\ \hline 1001 \\ 1001 \\ 1001 \\ 1001 \\ \hline 11101010 \end{array}$$

Задания:

1. Сложите числа:

а) 1011101_2 и 1110111_2 ; б) 1011101_2 и 101011_2 ; в) 1011_2 , 11_2 и 1111_2 ; г) 1011_2 , 1101_2 и 111_2 ;

2. Вычтите:

а) 111_2 из 10100_2 ; б) 1011_2 из 1101_2 ; в) 1111_2 из 10010_2 ; г) 10001_2 из 111011_2 ;

3. Перемножьте числа:

а) 101101_2 и 101_2 ; б) 111101_2 и 1101_2 ; в) 101111_2 и 1011_2 ; г) 101_2 и 1111001_2 ;



Домашнее задание. Подготовьтесь к тестированию по составу компьютера и системам счисления.



Измерение информации.

Содержательный и алфавитный подходы

1. Материалы по теме находятся на сайте <http://matem-109.ru>,

- Информатика, Материалы, Измерение количества информации;

- Информатика, Материалы, Презентация «Различные подходы к измерению количества информации».

2. Вопросы по теме:

1) Определение и свойства информации. Формы представления информации. Язык (определение), виды языков (краткая характеристика);

1. Понятие «информация» относится к числу наиболее общих понятий, применяемых для описания картины мира. Невозможно дать общее определение информации, универсальное для всех областей человеческой деятельности. С точки зрения информатики «информация» - это сведения об окружающих нас объектах.

2. Виды информации:

По способу получения	По способу представления (фиксации)
зрительная (визуальная)	числовая
звуковая (аудиальная)	текстовая
обонятельная	графическая
вкусовая	звуковая
тактильная (осязательная)	видеоинформация

3. Информацию можно представить и передать в различной форме:

☑ **Знаковой письменной**, состоящей из различных знаков, среди которых принято выделять:

❖ **Символьную** в виде текста, чисел, специальных символов;

❖ **Графическую** (например, географическая карта);

❖ **Табличную** (например, таблица, задающая функцию).

☑ **В виде жестов или сигналов** (регулировщик на улице, сигналы светофора, язык глухонемых).

☑ **Устной или словесной.**

4. **Свойства информации:** информация должна быть ПОНЯТНОЙ, точной, достоверной, полезной, актуальной, полной.

5. Независимо от формы представления и способа передачи информации, она всегда передается с помощью какого-либо языка.

Естественные языки	Сформировались в процессе развития человечества. Разговорный язык (устный и письменный)
Формальные языки	Созданы специально для определённых целей (язык математики, физики, химии ..., языки программирования)
Алфавит	Основа любого языка. Набор знаков (символов), из которых формируется сообщение

2) Схема обработки информации (информационный процесс):



3) 1 БИТ (определение и значение) и единицы информации (см. учебник 8 класса);

4) Формула для определения количества информации (содержательный подход) (Информатика, Материалы, Измерение количества информации);

5) Формула для определения количества информации (алфавитный подход, информационная ёмкость знака, количество информации в сообщении) (Информатика, Материалы, Измерение количества информации).

3. Примеры задач.

Пример 1. После экзамена по информатике, который сдавали ваши друзья, объявляются оценки («2», «3», «4» или «5»). Какое количество информации будет нести сообщение об оценке учащегося А, который выучил лишь половину билетов.

Решение. Опыт показывает, что для учащегося А все четыре оценки (события) равновероятны. Поэтому число возможных событий $N = 4$, и можно воспользоваться формулой: $N = 2^i$

$$4 = 2^i, i = 2 \text{ (т.к. } 2^2 = 4)$$

Ответ: количество информации равно 2 битам.

Пример 2. Какое количество вопросов достаточно задать вашему собеседнику, чтобы наверняка определить месяц, в котором он родился?

Решение. Если опрашивать человека напрямую, то возможно придётся задать до 11 вопросов (в случае рождения в декабре месяце). **Правильная стратегия состоит в том, что вопросы нужно задавать так, чтобы количество возможных вариантов каждый раз уменьшалось вдвое.** Тогда количество возможных событий в каждом из полученных подмножеств будет одинаково и их отгадывание равновероятно. В этом случае на каждом шаге ответ («Да» или «Нет») будет нести максимальное количество информации (1 бит).

Допустим, человек родился в сентябре. Всего в году 12 месяцев

	Вопрос	Ответ	Вывод
1	Вы родились после июня?	Да	Человек родился с июля по декабрь. В рассмотрении остаётся 6 месяцев ($12 : 2$)
2	Вы родились после сентября?	Нет	Человек родился с июля по сентябрь. В рассмотрении остаётся 3 месяца ($6 : 2$)

3	Вы родились в июле?	Нет	В рассмотрении остаётся два месяца — август и сентябрь
4	Вы родились в августе?	Нет	Человек родился в сентябре.

Возможно, что мы обошлись бы и меньшим количеством вопросов, если бы сразу «угадали» месяц рождения — сентябрь.

Ответ: необходимо задать 4 (или 3) вопроса.

Пример 3. Какова мощность алфавита, с помощью которого записано сообщение, содержащее 2048 символов, если его объем составляет 1,25 Кбайт.

Решение. Если считать появление каждого знака алфавита в тексте событиями равновероятными, то для определения количества информации можно воспользоваться формулой

$N = 2^i$, где N – мощность алфавита (количество знаков в алфавите), i – информационный вес одного знака этого алфавита (бит).

Количество информации, которое несет один знак алфавита тем больше, чем больше знаков входят в этот алфавит, т.е. чем больше мощность алфавита (количество знаков, входящих в алфавит).

Количество информации, содержащееся в сообщении, закодированном с помощью знаковой системы, равно количеству информации, которое несет один знак алфавита, умноженному на число знаков в сообщении:

$I = i \cdot K$, где I – количество информации в сообщении (бит), i – информационный вес одного знака этого алфавита (бит), K – количество знаков в сообщении.

1) Переведем информационный объем сообщения из кбайт в биты:

$I = 1,25 \cdot 1024 \cdot 8 = 10240$ (бит) (1 кбайт = 1024 байт, 1 байт = 8 бит. Можно считать и по другому: $1,25$ кбайт = $1,25 \cdot 2^{10} \cdot 2^3$ бит = $1,25 \cdot 2^{13}$ бит).

2) Определим количество бит, приходящееся на один символ:

$10\ 240 \text{ бит} : 2\ 048 = 5$ (бит) (Можно считать по-другому: количество знаков в сообщении $2048 = 2^{11}$. Если объем сообщения составляет $1,25 \cdot 2^{13}$, то информационный вес одного знака составляет $1,25 \cdot 2^{13} \text{ бит} : 2^{11} = 1,25 \cdot 2^2 = 1,25 \cdot 4 = 5$ (бит)).

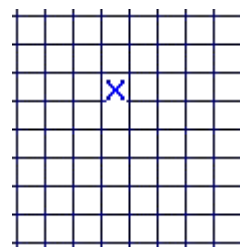
3) $N = 2^i = 2^5 = 32$ (знака).

Ответ: мощность алфавита (количество знаков в алфавите) равна 32.

4. Задачи

1. Какое количество информации несет в себе сообщение о том, что нужная вам программа находится на одной из восьми дискет?

2. Какое количество информации получит второй игрок при игре в крестики-нолики на поле 8x8, после первого хода первого игрока, играющего крестиками?



3. В рулетке общее количество лунок равно 128. Какое количество информации мы получаем в зрительном сообщении об остановке шарика в одной из лунок?
4. Происходит выбор одной карты из колоды в 32 карты. Какое количество информации мы получаем в зрительном сообщении о выборе определенной карты?
5. Какое количество вопросов достаточно задать вашему собеседнику, чтобы точно определить день и месяц его рождения?
6. Заполните пропуски числами:
- а) 5 Кбайт = _____ байт = _____ бит,
 - б) _____ Кбайт = _____ байт = 12288 бит;
 - в) _____ Кбайт = _____ байт = 2^{13} бит;
 - г) _____ Гбайт = 1536 Мбайт = _____ Кбайт;
 - д) 512 Кбайт = $2^{\text{—}}$ байт = $2^{\text{—}}$ бит (определить степень числа 2).
7. В кодировке КОИ-8 каждый символ кодируется одним байтом. Определите информационный объем сообщения из 20 символов в этой кодировке.
8. Скорость передачи данных через ADSL-соединение равна 256000 бит/с. Передача файла через данное соединение заняла 16 секунд. Определите размер файла в Кбайт.
9. Какова мощность алфавита, с помощью которого записано сообщение, содержащее 2048 символов, если его объем составляет $1/512$ часть одного мегабайта.
10. Пользователь компьютера, хорошо владеющий навыками ввода информации с клавиатуры, может вводить в минуту 100 знаков. Мощность алфавита, используемого в компьютере равна 256. Какое количество информации в байтах может ввести пользователь в компьютер за 1 минуту?
- 11*. Система оптического распознавания символов позволяет преобразовывать отсканированные изображения страниц документа в текстовый формат со скоростью 4 страницы в минуту и использует алфавит мощностью 65536 символов. Какое количество информации будет нести текстовый документ после 5 минут работы приложения, страницы которого содержат 40 строк по 50 символов.



1. Задачи на шифрование информации.

При передаче радиогаммы было потеряно разбиение на буквы, но известно, что в радиогамме могли использовать только следующие буквы:

Определите текст радиограммы. В ответе укажите, сколько букв было в исходной радиограмме.

2) Некоторый алгоритм из одной цепочки символов получает новую цепочку следующим образом. Сначала записывается исходная цепочка символов, после нее записывается исходная цепочка символов в обратном порядке, затем записывается буква, следующая в русском алфавите за той буквой, которая в исходной цепочке стояла на последнем месте. Получившаяся цепочка является результатом работы алгоритма. Например, если исходная цепочка символов была **ЛЕС**, то результатом работы алгоритма будет цепочка **ЛЕССЕЛТ**. Дана цепочка символов **ЕН**. Какая цепочка символов получится, если к данной цепочке применить алгоритм дважды (то есть к данной цепочке применить алгоритм, а затем к результату его работы еще раз применить алгоритм)?

2. Двоичное кодирование текстовой информации.

- процесс, при котором каждому текстовому символу (букве, знаку и т.д.) ставится в соответствие двоичный код (некоторое сочетание 0 и 1).

Одним из способов кодирования текстовой информации - кодирование с помощью **кодовой таблицы ASCII**. При этом способе кодирования на 1 символ выделяется 1 байт (8 бит) информации. При таком способе кодирования можно закодировать всего **256 символов**.

Пример: найти двоичный код слова Щит.	Кодовая таблица ASCII (основная часть) + CP-1251 (расширение для русской части)															
1. Найдём десятичный код каждой буквы с учётом регистра: Щ – 217, и – 232, т – 242. 2. Переведём каждое десятичное число в двоичное: $217_{10} = 128 + 64 + 16 + 8 + 1 = 2^7 + 2^6 + 2^4 + 2^3 + 2^0 = 11011001_2$, $232_{10} = 128 + 64 + 32 + 8 = 11101000_2$, $242_{10} = 128 + 64 + 32 + 16 + 2 = 11110010_2$. Ответ: двоичный код слова Щит - 110110011110100011110010	sp	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
	Á	à	,	è	„	…	†	‡	€	%	É	<	й	Й	ó	ú
	á	‘	’	“	”	•	–	—	ë	™	é	>	ò	й	ó	ú
	nbsp	ÿ	Ы	Э	х	Ы	І	§	Ё	©	Ю	«	¬	shy	®	Я
	°	±	Ы	Э	’	µ	¶	•	ё	№	ю	»	Э	ю	я	я
	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я

Какое количество информации приходится на слово Щит?

Ответ: двоичный код слова Щит содержит 24 бит (3 байта) информации.

Задание 1. Найдите двоичный код слова Щит с помощью этой таблицы.

Задание 2. На компьютере в кодировке КОИ-8 набрали слово ФАЙЛ и отправили по e-mail. Что прочитают на компьютере в кодировке CP 1251 в этом сообщении?

Русская часть кодовой таблицы КОИ-8

—		Г	Г	Л	Л	Т	Т	Т	Т	Т	Т	Т	Т	Т	Т	Т
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144
145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161
162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178
179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195
196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212
213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229
230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246
247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263

В настоящее время на платформе Windows применяется новый стандарт **Unicode**, который отводит под один символ не один байт, а два, и поэтому с его помощью можно закодировать не 256, а $N = 2^{16} = 65536$ различных символов. Полная спецификация стандарта Unicode включает в себя все существующие, вымершие и искусственно созданные алфавиты мира, а также множество математических, музыкальных, химических и прочих символов.

□ Компьютеры фирмы Apple, работающие под управлением операционной системы Mac OS, используют свою собственную кодировку **Mac**.

□ Также, Международная организация по стандартизации (International Standards Organization, ISO) утвердила в качестве стандарта для русского языка еще одну кодировку под названием **ISO 8859-5**.

3. Двоичное кодирование графической информации.

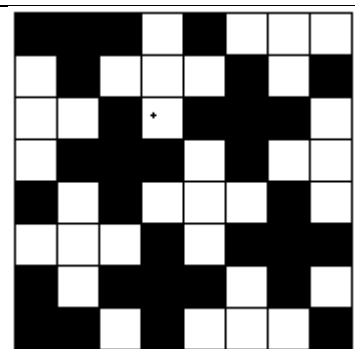
- процесс, при котором каждому мельчайшему элементу рисунка (изображения) - пикселю - ставится в соответствие двоичный код. В данном случае речь идёт о кодировании **растровой графики** (состоящей из мельчайших точек - **пикселей**). Существует ещё **векторная графика**, принцип кодирования которой совершенно другой.

Пример 1. Пусть надо закодировать фрагмент, в котором использовано всего **два** цвета: **белый** и **чёрный**.

Договоримся, что "**белый**" пиксель обозначим, например **0**, а "**чёрный**" - **1**. Получается, что каждая "**точка**" (**пиксель**) содержит **1 бит** информации.

Тогда первая "строка" кода изображения будет следующей: **11101000**, а вторая - **01000101**. Тогда для всего фрагмента получим следующий код:

```
11101000
01000101
00101110
01110100
10100010
00010111
10111010
11010001
```

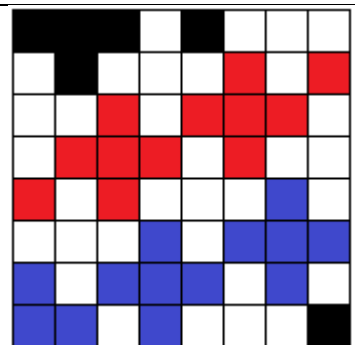


1	1	1	0	1	0	0	0
0	1	0	0	0	1	0	1
0	0	1	0	1	1	1	0
0	1	1	1	0	1	0	0
1	0	1	0	0	0	1	0
0	0	0	1	0	1	1	1
1	0	1	1	1	0	1	0
1	1	0	1	0	0	0	1

Несложно подсчитать, что в каждой строке содержится **8 бит** (**1 байт**) информации, строк всего получилось 8. Таким образом, всё изображение составляет **8 x 8 = 64 бит** (или **1 x 8 = 1 байт**).

Пример 2. Пусть надо закодировать фрагмент, в котором использовано уже **4** цвета: **белый**, **чёрный**, **красный** и **синий**.

В этом случае для каждой точки уже мало 1 бита. Попробуем договориться так: **белый** пиксель, например, обозначим как **00**, **чёрный** - **11**, для **красного** выберем сочетание **01**, для **синего** - **10**. Теперь получается, что **каждый пиксель** содержит **2 бита** информации.



Тогда код первой строки изображения будет таким: 11 11 11 00 11 00 00 00 (для удобства рассмотрения код каждого пикселя отделён пробелом). Для всего фрагмента получим следующий код: В каждой строке фрагмента 8 точек (пикселей), на каждый приходится 2 бита информации, поэтому на всю строку приходится 16 бит информации. Если строк 8, то всё изображение содержит 128 бит или 16 байт информации.	<pre>1111110011000000 0011000000010001 0000010001010100 0001010100010000 0100010000001000 0000001000101010 1000101010001000 1010001000000011</pre>
---	--

Можно заметить закономерность:

2 цвета - каждый пиксель содержит **1 бит** информации ($2 = 2^1$);

4 цвета - каждый пиксель содержит **2 бита** информации ($4 = 2^2$).

Поэтому можно предположить, что, если потребуется закодировать фрагмент, содержащий **8 цветов**, то для кодирования пикселя потребуется уже **3 бита** ($8 = 2^3$).

Таким образом, количество цветов, которое можно закодировать, подчиняется той же формуле для измерения количества информации:

$N = 2^i$, только **N** в данном случае – количество кодируемых цветов, **i** – сколько бит информации необходим для кодирования одной точки (пикселя) изображения.

В настоящее время каждая **точка** растрового изображения кодируется **24 битами (3 байтами)** или **32 битами (4 байтами)**.

Таким способом можно закодировать 2^{24} (т.е., чтобы узнать, сколько это, надо будет перемножить между собой 24 двойки!) или 2^{32} (!) разных цветов и оттенков.

4. Двоичное кодирование числовой информации.

Представление целых чисел. Целые числа могут представляться в компьютере со знаком или без знака

Рассмотрим представление целых чисел без знака. Целые числа без знака обычно занимают в памяти компьютера один или два байта. В однобайтовом формате принимают значения от 00000000_2 до 11111111_2 . В двухбайтовом формате - от $00000000\ 00000000_2$ до $11111111\ 11111111_2$.

Диапазоны значений целых чисел без знака

Формат числа в байтах	Диапазон	
	Запись с порядком	Обычная запись
1	0 ... 2^8-1	0 ... 255
2	0 ... $2^{16}-1$	0 ... 65535

Примеры:

а) число $72_{10} = 1001000_2$ в однобайтовом формате:	Номера разрядов Биты числа	7 6 5 4 3 2 1 0 0 1 0 0 1 0 0 0
б) это же число в двубайтовом формате:	Номера разрядов Биты числа	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
в) число 65535 в двубайтовом формате:	Номера разрядов Биты числа	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Целые числа со знаком* (для физико-математической группы) обычно занимают в памяти компьютера один, два или четыре байта, при этом левый (старший) разряд содержит информацию о знаке числа.

Диапазоны значений целых чисел со знаком

Формат числа в байтах	Диапазон	
	Запись с порядком	Обычная запись
1	$-2^7 \dots 2^7-1$	-128 ... 127
2	$-2^{15} \dots 2^{15}-1$	-32768 ... 32767
4	$-2^{31} \dots 2^{31}-1$	-2147483648 ... 2147483647

При **однобайтовом формате** для знака отводится один разряд, а для цифр абсолютной величины – семь разрядов.

В компьютерной технике применяются три формы кодирования целых чисел со знаком: **прямой код, обратный код, дополнительный код**.

Обратный и дополнительный код позволяют заменить разнообразные арифметические операции операцией **сложения**:

Положительные числа в прямом, обратном и дополнительном кодах выглядят одинаково – в знаковом разряде стоит цифра «0»:	Число $1_{10}=1_2$ 0 0 0 0 0 0 0 1 Знак числа "+"	Число $127_{10}=1111111_2$ 0 1 1 1 1 1 1 1 Знак числа "+"
Отрицательные числа в прямом коде:	Прямой код числа - 1 1 0 0 0 0 0 0 1 Знак числа "-"	Прямой код числа - 127 1 1 1 1 1 1 1 1 Знак числа "-"
Отрицательные числа в обратном коде (получается инвертированием цифр двоичного кода модуля числа):	Число: -1 Код модуля числа: 0 0000001 Обратный код числа: 1 1111110 1 1 1 1 1 1 1 0	Число: -127 Код модуля числа: 0 1111111 Обратный код числа: 1 0000000 1 0 0 0 0 0 0 0
Отрицательные числа в дополнительном коде (получается из обратного кода прибавлением единицы к младшему разряду):	Дополнительный код числа - 1 1 1 1 1 1 1 1 1	Дополнительный код числа - 127 1 0 0 0 0 0 0 1

Обычно отрицательные десятичные числа **при вводе** в машину **автоматически преобразуются в обратный или дополнительный двоичный код** и в таком виде хранятся, перемещаются и участвуют в операциях. **При выводе** таких чисел из машины происходит **обратное преобразование** в отрицательные десятичные числа.



Домашнее задание. Подготовьтесь к тестированию по кодированию и измерению количества информации.



ОСНОВЫ ЛОГИКИ

1. Составьте конспект по материалам лекции:

- 1) Логика (определение);
- 2) Понятие (определение);
- 3) Структура понятия;
- 4) Основные виды отношений между множествами;
- 5) Примеры графического отображения отношений;
- 6) Высказывание (определение);
- 7) Виды высказываний по содержанию;
- 8) Виды высказывание по форме;
- 9) Логическая форма высказывания;
- 10) Умозаключение (определение);
- 11) Виды умозаключений;
- 12) Доказательство.

2. Задачи

1. Отобразите с помощью диаграммы Эйлера-Венна соотношения между следующими объёмами понятий:
а) целые и натуральные числа; б) чётные и нечётные числа.
2. Определите, что является субъектом, предикатом и связкой в следующих суждениях:
а) Сканер – это устройство ввода информации;
б) Луна является спутником Земли;
в) Атом состоит из ядра и электронов.
3. Определите, какое высказывание является простым, какое составным (сложным):
а) Через точку, не лежащую на прямой, проходит только одна прямая, параллельная данной;
б) Прямые α и β перпендикулярны;
в) Пропорция – это равенство двух отношений;
г) Прямые, которые не пересекаются, являются параллельными;
д) Если прямая пересекает одну из двух параллельных прямых, то она пересекает и вторую прямую;
е) На тело, погружённое в жидкость или газ, действует выталкивающая сила, равная весу жидкости (газа), вытесненной данным телом.
4. Определите, какое высказывание является истинным, какое ложным:

- а) Непересекающиеся прямые всегда параллельны;
- б) Если прямые скрещиваются, то они не пересекаются;
- в) Две пересекающиеся плоскости образуют четыре двугранных угла;
- г) Все линейные углы одного двугранного угла равны;
- д) Комар – разновидность таракана;
- е) Если бросить камень вверх, то он становится невесомым.

5*. При каких значениях числа X предикат первого порядка не $((X > 8) \text{ или } (X < -3))$ примет значение: а) ложь; б) истина.



Базовые логические операции

1. Составьте конспект по материалам лекции:

- 1) Цель создания алгебры высказываний;
- 2) Логические переменные и их значения;
- 3) Базовые логические операции;
- 4) Определение и обозначение конъюнкции, дизъюнкции и инверсии;
- 5) Таблицы истинности для конъюнкции, дизъюнкции и инверсии;
- 6) Логические выражения (функции).

Пример 1. Требуется определить истинность логического выражения:
 $F = \langle 2 \times 2 = 5 \text{ или } 2 \times 2 = 4 \text{ и } 2 \times 2 \neq 5 \text{ или } 2 \times 2 \neq 4 \rangle$.

Решение. Пусть выражение $A = \langle 2 \times 2 = 5 \rangle$ – ложно (0), $B = \langle 2 \times 2 = 4 \rangle$ – истинно (1). Перепишем исходное выражение $F = \langle 2 \times 2 = 5 \text{ или } 2 \times 2 = 4 \text{ и } 2 \times 2 \neq 5 \text{ или } 2 \times 2 \neq 4 \rangle$ с использованием введенных обозначений:

$$F = \langle A \text{ или } B \text{ и не } A \text{ или не } B \rangle$$

$$F = (A \vee B) \wedge (\neg A \vee \neg B) \text{ (или обозначим по-другому } F = (A \vee B) \& (A \vee B))$$

Решим задачу с использованием базовых логических операций для данного условия: $F = (A \vee B) \wedge (\neg A \vee \neg B) = (0 \vee 1) \wedge (1 \vee 0) = 1 \wedge 1 = 1$.

Пример 2. Но под выражениями A и B могут выступать любые высказывания, которые могут быть как истинными, так и ложными. Если требуется выяснить истинность логического выражения при всех возможных сочетаниях значений логических переменных, то нужно составить **таблицу истинности** для логического выражения.

Выясним истинность логического выражения $F = (A \vee B) \wedge (\neg A \vee \neg B)$ при всех возможных сочетаниях значений логических переменных.

Решение. Составим таблицу истинности, где в качестве исходных данных переберём все возможные сочетания значений логических переменных А и В:

A	B	$A \vee B$	$\neg A$	$\neg B$	$(\neg A \vee \neg B)$	$F = (A \vee B) \wedge (\neg A \vee \neg B)$
1	1					
1	0					
0	1					
0	0					

2. Задачи

1. Для какого из указанных значений числа X истинно выражение $(X < 3) \& \neg(X < 2)$?

- 1) 1 2) 2 3) 3 4) 4

2. Определите истинность составного высказывания: $(\overline{A} \& \overline{B}) \& (C \vee D)$, состоящего из простых высказываний:

A = {Принтер – устройство вывода информации},
 B = {Процессор – устройство хранения информации},
 C = {Монитор – устройство вывода информации},
 D = {Клавиатура – устройство обработки информации}.

3. В таблице приведены запросы к поисковому серверу. Расположите обозначения запросов в порядке **возрастания** количества страниц, которые найдет поисковый сервер по каждому запросу. Для обозначения логической операции “ИЛИ” в запросе используется символ $\vee$, а для логической операции “И” – $\&$.

А	Литература & Экзамен
Б	Литература $\vee$ Экзамен $\vee$ Билеты
В	Литература $\vee$ Билеты
Г	Литература & Экзамен & Билеты

4. Найдите значения логических выражений:

а) $(1 \vee 1) \vee (1 \vee 0)$; б) $((1 \vee 0) \vee 1) \vee 1$; в) $(0 \vee 1) \vee (1 \vee 0)$; г) $(0 \& 1) \& 1$; д) $1 \& (1 \& 1) \& 1$;
 е) $((1 \vee 0) \& (1 \& 1)) \& (0 \vee 1)$; ж) $((1 \& 0) \vee (1 \& 0)) \vee 1$; з) $((0 \& 0) \vee 0) \& (1 \vee 1)$.

5. Даны простые высказывания: A = {5>3}, B = {2=3} и C = {4<2}.

Определите истинность составных высказываний:

а) $(A \vee B) \& C$; б) $(A \& C) \vee (B \& C)$; в) $(A \& B) \vee C$; г) $(A \vee C) \& (A \& B)$.



Домашнее задание. Подготовьтесь к тестированию по основам логики и базовым логическим операциям.



Основные алгоритмические конструкции

1. Повторение: Алгоритмы. Определение и свойства. Способы описания

Любая среда программирования содержит инструменты для описания, создания, изменения определенных объектов. Способ применения инструмента задается инструкцией (командой языка программирования), но прежде чем писать команды на языке программирования, необходимо подумать и составить алгоритм.

Алгоритм	Совокупность правил выполнения определенных действий, обеспечивающих решение задачи, содержит описание действий.
Способы описания	- словесное (устное или письменное); - графическое (например: описание с помощью чертежей способов построения геометрических фигур)
Блок-схема	Разновидность графического способа описания алгоритма. Каждый шаг алгоритма помещается внутри блока специальной формы

2. Объекты блок-схемы

Обозначение	Содержание
	Начало или конец алгоритма. Внутри блока слова «Начало» или «Конец»
	Выполняемое действие записываем внутри прямоугольника
	Условие выполнения действий записывается внутри ромба
	Последовательность выполнения (влево и вверх линии со стрелками, вправо и вниз линии без стрелок)
	Блок ввода исходных данных
	Блок вывода результата, документ

3. Виды алгоритмов

Элементарные алгоритмы	Простейшие алгоритмы, из которых можно составлять алгоритмы любой сложности. В любом сложном алгоритме можно вычленить составляющие его элементарные алгоритмы. На основании составленных алгоритмов можно написать программу на любом языке программирования
-------------------------------	--

Вид алгоритма	Определение	Блок-схема
Линейный (последовательный)	Алгоритм, который описывает последовательно выполняющиеся действия, следующие одно за другим	<pre> graph TD A[Начало] --> B[Команда 1] B --> C[Команда 2] C --> D[Команда 3] D --> E[Конеч] </pre>
Циклический	Алгоритм, который описывает повторяющиеся действия	<pre> graph TD A[Начало] --> B[n раз] B --> C[Команда 1] C --> D[...] D --> E[Команда n] E --> F[Конеч] E -- loop --> B </pre>
Разветвляющийся	(неполная форма) алгоритм, в котором в зависимости от условий некоторые действия надо пропустить	<pre> graph TD A[Действия1] --> B{Условие} B -- Да --> C[Действия2] B -- Нет --> D[Действия3] C --> D </pre>
	(полная форма) алгоритм, в котором в зависимости от условия, совершается одна или другая последовательность действий	<pre> graph TD A[Действия1] --> B{Условие} B -- Да --> C[Действия3] B -- Нет --> D[Действия2] C --> E[Действия4] D --> E </pre>

4. Задача. Составьте блок-схему алгоритма по следующему тексту: «Мы готовим яичницу. Для этого возьмём 2 яйца, соль, подсолнечное масло и сковороду. Поставим сковороду на плиту и включим её. Нальём немного подсолнечного масла. Если мы готовим яичницу-глазунью, то просто разобьём два яйца над сковородой, если нет, то сначала выльем содержимое яиц в чашку, взобьём венчиком и выльем содержимое на сковороду. После этого жарим яичницу, пока не приготовится».

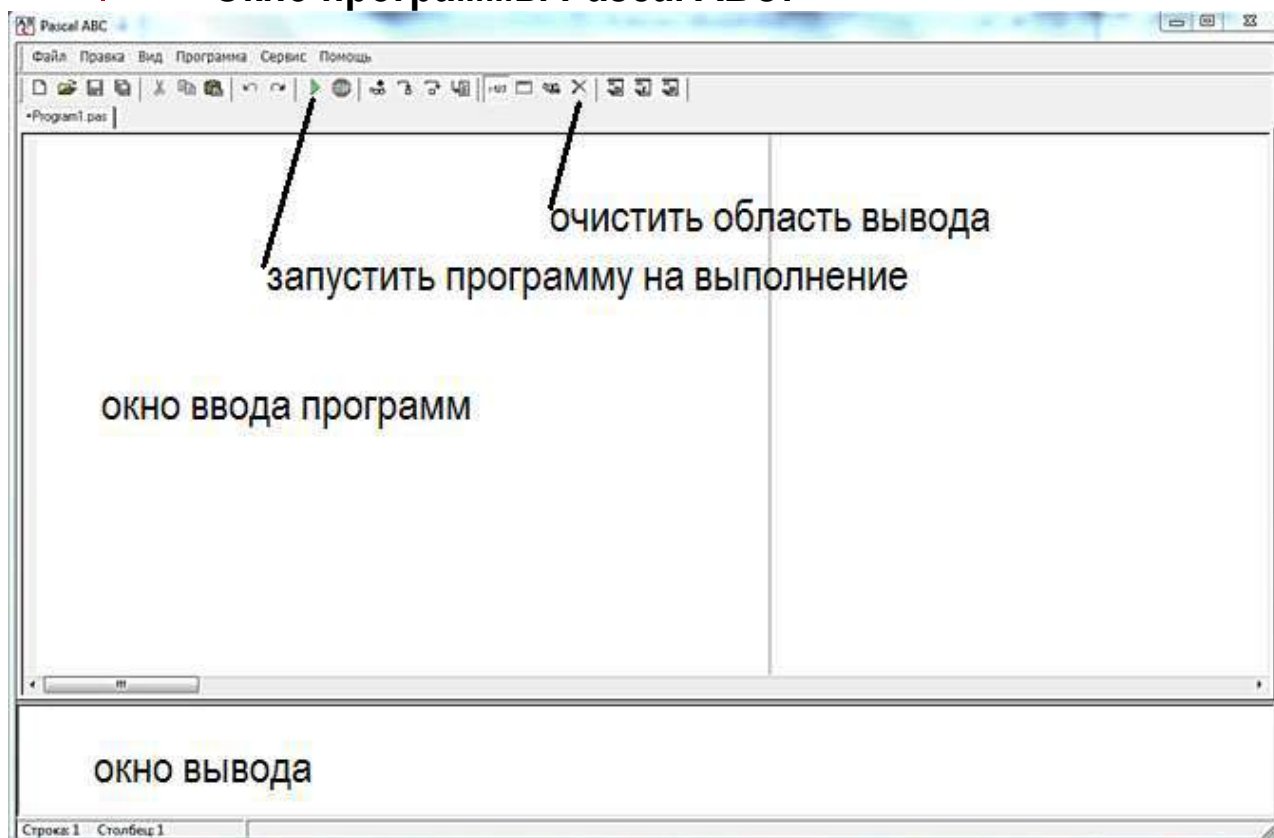


Язык программирования Pascal ABC

1. Начало



Окно программы Pascal ABC:



Типы переменных

В начале программы задаются все переменные, которые будут использованы, а также их тип.

Переменная – предназначена для хранения и обработки данных в программах. В программе представлена **именем** и служит для обращения к данным определённого **типа**, конкретное **значение** которых хранится в ячейках оперативной памяти.

Некоторые типы переменных:

Тип	Название и возможные значения	Объём памяти
integer	Целый (диапазон чисел от -2147483648 до 2147483647)	4 байт
byte	Беззнаковый целый (диапазон чисел от 0 до 255)	1 байт
word	Беззнаковый целый (диапазон чисел от 0 до 65536)	2 байт
char	Символьный (символы в кодировке Windows)	1 байт
string	Строковый (символы в кодировке Unicode, в строке не более 255 символов)	До 256 байт
boolean	Логический (принимают значение True или False)	1 байт
real	Вещественный (диапазон чисел от $5.0 \cdot 10^{-324}$ до $1.7 \cdot 10^{308}$)	8 байт

Имя переменной – имя каждой переменной (идентификатор) уникально и не может меняться в процессе выполнения программы. Может состоять из различных символов (чаще, латинских), но должно **обязательно начинаться с буквы и не должно включать точку «.»**. Число символов в имени – не более 255.

Арифметические выражения – могут входить переменные числового типа и числа.

Арифметические операции:

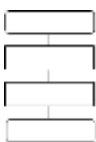
Запись арифметических операций	Арифметические операции
$A+B$	Сложение A и B
$A-B$	Вычитание B из A
$A*B$	Умножение A на B
A/B	Деление A на B
$()$	Скобки

Строковые выражения – входят выражения строкового типа, строки и строковые функции. Это любая последовательность символов, заключённая в апострофы («'»), строковые переменные, строковые функции.

Логические выражения – входят логические переменные, числа, числовые или строковые переменные или выражения, которые сравниваются между собой с использованием операций сравнения (>, <, >=, <= и т.д.). Логические выражения могут принимать только два значения: «истина» («True») или «ложь» («False»). Над элементами логических выражений могут производиться логические операции: and – логическое умножение, or – логическое сложение, not – логическое отрицание.



Первые программы



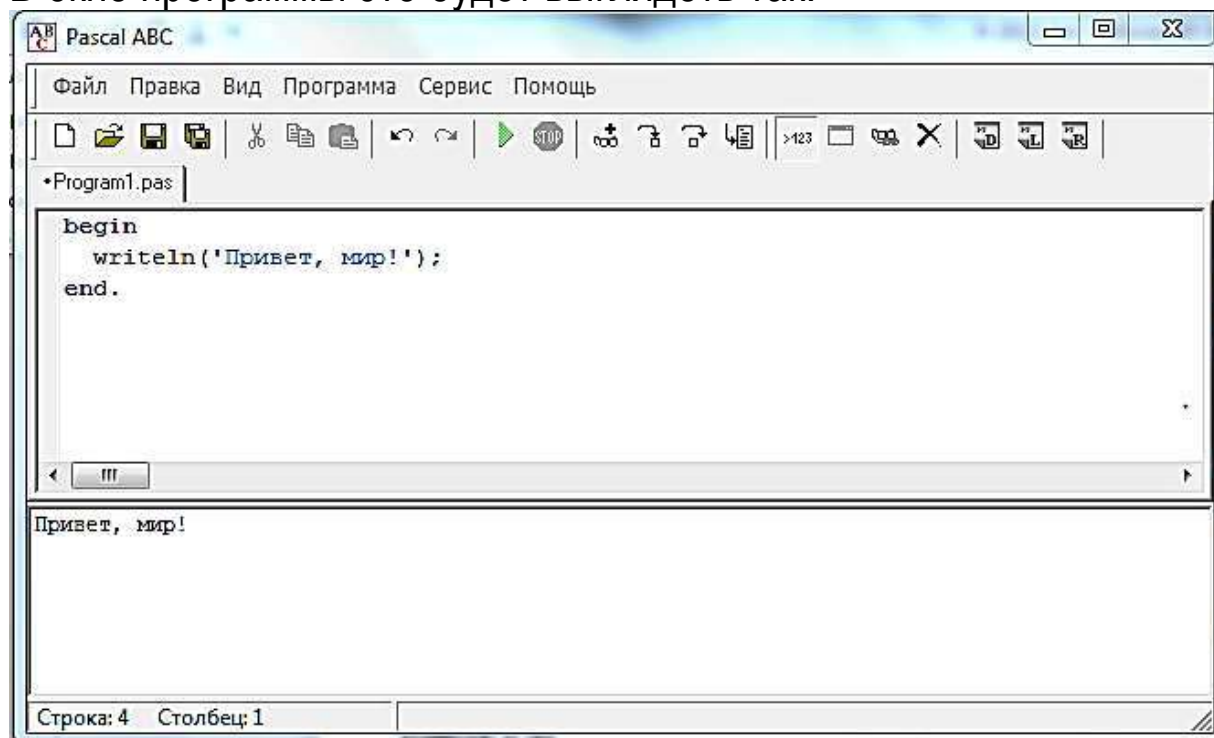
Во всех программах, разобранных ниже, используется только линейный алгоритм (вспомните определение и общую блок-схему линейного алгоритма).

Задание 1. Первую программу на любом языке программирования часто называют «Привет, мир!». Откройте Pascal ABC и в окне программ запишите:

```
begin
    writeln('Привет, мир!');
end.
```

Запустите программу на выполнение (F9 или нажмите )

В окне программы это будет выглядеть так:



Что и как написано в программе:

begin end.	Конструкция задаёт начало и конец программы. Всё, что должна выполнять программа, размещается между этими ключевыми словами.
writeln()	Оператор вывода (буквально: «напиши строку»). Курсор переходит на следующую строку.
'Привет, мир!'	Обратите внимание: текст, который необходимо вывести на печать, обязательно помещается в апострофы: 'текст'
;	Этот знак ставится обязательно в конце каждой строки (за исключением строк, содержащих ключевые слова).

Измените текст приветствия, добавьте ещё строки. Сохраните файл с именем **Фамилия_1** в свою папку.

Задание 2. Программа вычисления площади круга ($S = \pi r^2$):

```
const Pi = 3.1415;
var
  r: real; // радиус круга
  S: real; // площадь круга
begin
  write('Введите радиус круга: ');
  readln(r);
  S := Pi*r*r;
  writeln('Площадь круга равна ', S);
end.
```

Что и как написано в программе:

const Pi = 3.1415;	Задана постоянная (обозначено ключевым словом const) с именем Pi, которой сразу присвоено значение (= 3.1415). Обратите внимание – дробная часть отделена от целой точкой, а не запятой, как мы привыкли!
var	Ключевое слово, обозначающее, что далее будут описаны переменные («;» не ставится!).
 r: real; S: real;	Заданы переменные с именем r и S (это обозначено знаком «:»), определён тип данных (real). Обратите внимание: каждая строка завершена знаком «;».
// радиус круга // площадь круга	// - знак, обозначающий, что далее написан комментарий (в этом примере: что обозначают данные переменные). Комментарий «исключён» из программы, он не исполняется.
write();	Оператор вывода. Курсор не переходит на следующую строку. Чтобы было понятно, что следует делать, внутри скобок помещено предложение в апострофах, которое будет выведено в окне вывода. Можно не вводить пояснение, а оставить оператор в таком виде: write(). В этом случае программа также будет работать. Попробуйте провести такой эксперимент.
readln(r);	Оператор ввода, считывающий то, что вы ввели (буквально: «читай строку»), и присваивающий введённое значение переменной с именем r (обратите внимание – знак «:» в данном случае не ставится, он используется только тогда, когда переменная задаётся).
S := Pi*r*r;	Ввод расчётной формулы для переменной S. В данном случае использовано сочетание «:=», т.к. переменной с именем S присваивается значение, зависящее от значения, которое мы введём для переменной r.
writeln('Площадь круга равна ',S);	Уже знакомый оператор вывода результата. Обратите внимание: текстовая часть вывода результата внутри скобок помещена в апострофы, затем поставлена запятая («»), после которых выводится вычисленное значение переменной S (которое уже не является текстом, это вычисляемое значение).

Наберите программу, поэкспериментируйте с выводом и вводом данных, сохраните файл с именем **Фамилия_2** в свою папку.

Задание 3. Вычисление значения a^8 с помощью введения в программу вспомогательной переменной (основная идея: $a^8 = (((a)^2)^2)^2$).

```
var
a: real;
a2,a4,a8: real; //вспомогательные переменные
begin
  write('Введите значение a: ');
  readln(a);
  a2 := a * a;
  a4 := a2 * a2;
  a8 := a4 * a4;
  writeln(a, ' в степени 8 = ', a8);
end.
```

Обратите внимание: вывод результата состоит из трёх частей – [введённое значение переменной], [текстовая часть (выделена апострофами)], [вычисленное значение переменной]. Каждую часть отделяем запятой («,»), текстовая часть заключена в апострофы («'»), включая необходимые пробелы (пробел – это такой же текстовый символ).

Введите программу, проведите эксперимент с вводом и выводом текстовой части результата. Сохраните свой файл с именем **Фамилия_3** в свою папку.

Задания для самостоятельной проработки.

4) Составьте программу для вычисления суммы, разности, произведения и частного двух произвольных чисел.

5) Вычислите расстояние между двумя точками на прямой ($r = |x_A - x_B|$, где x_A и x_B – координаты точек A и B). Указание: расстояние между точками – число положительное, поэтому в формуле использован модуль. Стандартная функция, вычисляющая модуль: **abs ()**.

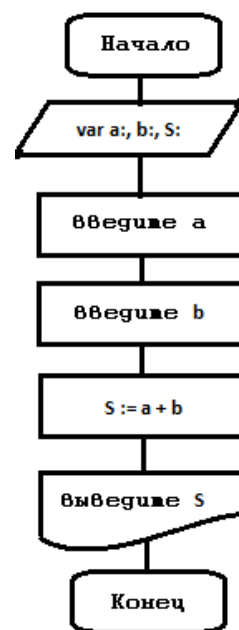
6) Вычислите значение гипотенузы в прямоугольном треугольнике по известным значениям катетов (теорема Пифагора). Указание: согласно теореме Пифагора вам потребуется вычислить арифметический квадратный корень: **sqrt ()**.

7) Бросание кубиков (вычислить сумму выпавших очков на верхних гранях кубиков при двух бросаниях). Указание: в данном случае вводимая переменная может принимать только целое положительное значение, поэтому тип переменной задаём **integer**. Выпадение каждой грани – равновероятное событие, но значение может получиться только от 1 до 6.

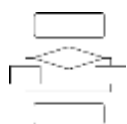
Для выбора случайного значения используется стандартная функция **Random (n)**, где **n** – количество чисел в выборке (целых, начиная с 0). Поэтому, если мы зададим **Random (6)**, то может быть выбрано любое из чисел 0, 1, 2, 3, 4, 5. Но на кубике первое значение – 1, а последнее – 6, поэтому необходимо к полученному случайному числу прибавить 1.

8) Обмен значений двух переменных с использованием третьей вспомогательной переменной (эта классическая задача очень часто используется в других, более сложных задачах). Основная идея: вводится вспомогательная переменная (например, **v**), которой присваивается значение первой переменной (например, **x**), тогда далее можно присвоить переменной **x** значение переменной **y**, а следующим шагом присвоить переменной **y** значение вспомогательной переменной (подумайте, какой результат выдаст программа, если этого не сделать, можете провести соответствующий эксперимент).

9) Выделение цифр из двузначного числа. Указание: в данном случае используются стандартные функции **a div n** (целая часть числа при делении **a** на **n**) и **a mod n** (дробная часть числа при делении **a** на **n**). Т.е., если в качестве значения числа **x** взять, например, число 45, то выражение **x div 10** «сработает» так: $45 : 10 = 4,5$, в качестве результата выйдет цифра 4, а выражение **x mod 10**: $45 : 10 = 4,5$, результат будет – цифра 5. В качестве эксперимента (чтобы посмотреть, как это работает) вы можете «случайно» ввести трёхзначное число. Оцените результат эксперимента.



Внимание! Решить надо все задачи, каждый файл сохраните в свою папку с именем **Фамилия_№**, где **№** - это номер соответствующей задачи. **Домашнее задание в тетради:** выполните блок-схему алгоритма к каждой задаче. Например, для задачи «найти сумму двух чисел **a** и **b**»:

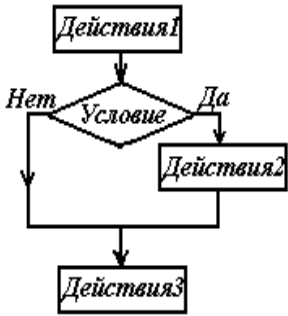
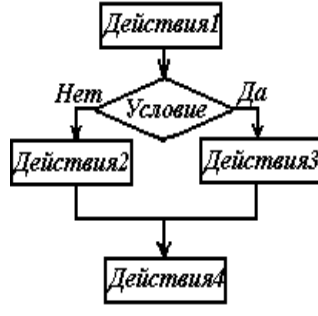


2. Разветвляющийся (условный) алгоритм



Определение. Алгоритм, в котором в зависимости от условия выполняется та или иная последовательность действий, называется разветвляющимся (условным) алгоритмом.

Блок-схема разветвляющегося алгоритма:

(неполная форма) алгоритм, в котором только выполнение условия ведёт к выполнению некоторых действий. 	(полная форма) алгоритм, в котором в зависимости от выполнения условия, совершается одна или другая последовательность действий. 
---	--



Условный оператор if.

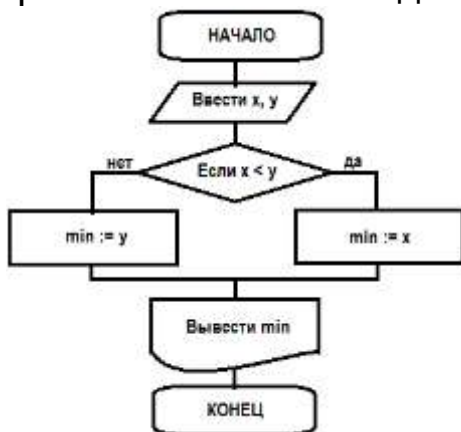
В зависимости от алгоритма используются конструкции:

if [условие] then [действие] (буквально: **если** выполнится условие, **затем** выполни действие);

if [условие] then [действие] else [действие] ... (буквально: **если** выполнится условие, **затем** выполни действие, **иначе** выполни действие ...).

Задание 1. Программа определения минимума из двух значений.
Это стандартная задача по программированию. Часто входит составной частью в более сложные программы.

Описание программы: пользователь вводит два числа, программа сравнивает их и выводит наименьшее значение.



Из блок-схемы видно, что нам необходимо ввести три переменные: x , y и $\min$ – x и y служат для ввода данных, $\min$ определяется внутри программы. Кроме того, видно, что нам достаточно сравнить x и y ($x < y$). Это условие может выполняться (тогда искомое значение – x) иначе – искомое значение y . Поэтому используется конструкция: **if... then... else**.

```

var
    x,y: integer;
    min: integer;
begin
    write('Введите x и y: ');
    readln(x,y);
    if x<y then
        min := x
    else min := y;
    writeln('Минимум = ',min);
end.

```


Что и как написано в программе:

<pre>var x,y: integer; min: integer;</pre>	Объявлены переменные – x , y , min . Для простоты задан тип данных integer . Хотя можно задать и real (не забудьте, что десятичная дробь вводится с точкой).
<pre>if x<y then min := x else min := y;</pre>	Если значение $x < y$, то переменной min присваивается значение x , иначе – присваивается значение y . Обратите внимание: вся конструкция воспринимается как одна строка.

Набрав программу в окне программ и протестировав её на целых числах, проведите следующие эксперименты: а) измените тип данных на **real** и протестируйте программу на десятичных дробях; б) введите одно значение положительное, другое отрицательное (минус не отделяется от числа пробелом!).

Сохраните файл с исходным вариантом программы (**Фамилия_1_1**) и с изменённым вариантом программы (**Фамилия_1_2**) в свою папку.

Задание 2. Программа определения чётности чисел.

Это также стандартная программа. Основная идея: **если** число делится на 2 без остатка (остаток равен 0), **то** число чётное, **иначе** – нечётное.

```
var x: integer;
begin
  write('Введите x: ');
  readln(x);
  if x mod 2 = 0 then
    writeln('Это чётное число')
  else writeln('Это нечётное число');
end.
```

Комментарии:

- т.к. понятие «чётное число» применимо только к целым числам, поэтому тип данных задан **integer**;

- использована уже знакомая стандартная функция **mod** (**дробная часть результата деления одного числа на другое**) – в данном случае эта функция использована в качестве условия: $x \bmod 2 = 0$ (дробная часть результата деления числа x на 2 равна 0).

- конструкция **if... then... else...** использована, т.к. при делении числа на 2 возможны только **два исхода**: остаток равен 0 (дробная часть результата деления – 0) или не равен 0 (дробная часть результата деления – 5).

Сохраните файл с именем **Фамилия_2** в свою папку.

Задание 3. Упорядочение чисел по возрастанию.

Ещё одна стандартная программа. Задача решается уже знакомым нам методом введения вспомогательной переменной:

```
var
  x,y: integer;
  v: integer;
begin
  write('Введите x,y: ');
  readln(x,y);
  if x>y then
    begin
      v := x;
      x := y;
      y := v;
    end;
  writeln('Результат
упорядочения по возрастанию:
',x,', ',y);
end.
```

Комментарий. «Обмен значениями» начнётся только в случае, если $x > y$. Это достигнуто тем, что сам процесс обмена значениями выделен операторными скобками:

```
begin
  v := x;
  x := y;
  y := v;
end;
```

Поэтому **end**, закрывающий операторные скобки, отделяется знаком «;».

Сохраните файл с именем **Фамилия_3** в свою папку.

Задание 4. Наименование времени года по номеру месяца.

Месяцы нумеруются от 1 (январь) до 12 (декабрь). При этом зима – это 12, 1 и 2 месяца; весна – 3, 4 и 5; лето – 6, 7 и 8; осень – 9, 10, 11. Сама программа выглядит так:

```
var
  Month: integer;
  Season: string;
begin
  write('Введите номер месяца: ');
  readln(Month);
  if (Month=1) or (Month=2) or (Month=12) then
    Season := 'Зима'
  else if (Month=3) or (Month=4) or (Month=5) then
    Season := 'Весна'
  else if (Month=6) or (Month=7) or (Month=8) then
    Season := 'Лето'
  else Season := 'Осень';
  writeln('Это ',Season)
end.
```

- т.к. месяц задаётся своим номером (т.е. это всегда целое число), то формат для переменной **Month – integer**;

- время года – это слово. Поэтому здесь мы впервые встречаемся с новым форматом **string** (строка символов в кодировке Unicode);

- в данном случае мы встречаемся со сложным условием (присвоить одно название времени года можно трём различным месяцам), поэтому три значения переменной **Month** должны быть соединены логическим сложением – **or**. Например:

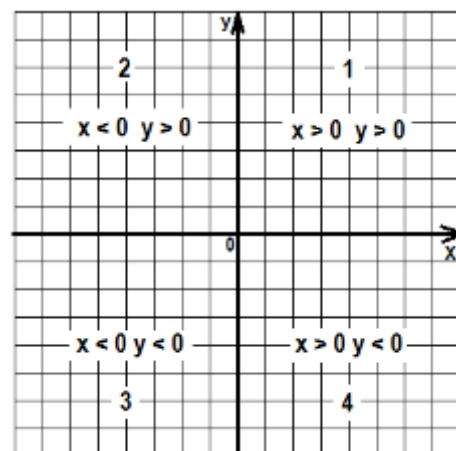
```
if (Month=1) or (Month=2) or (Month=12) then  
    Season := 'Зима'
```

(Буквально: «если месяц = 1 или месяц = 2, или месяц = 12, то время года – Зима»).

Сохраните файл с именем **Фамилия_4** в свою папку.

Задание 5. Определение координатной четверти, в которой находится точка.

Координатная плоскость разделена осями ОХ и ОУ на четыре координатные четверти. При этом в первой координатной четверти x и y принимают положительное значение, во второй – x принимает отрицательное, а y положительное значение, в третьей – x и y принимают отрицательное значение, в четвёртой – x принимает положительное, а y отрицательное значение.

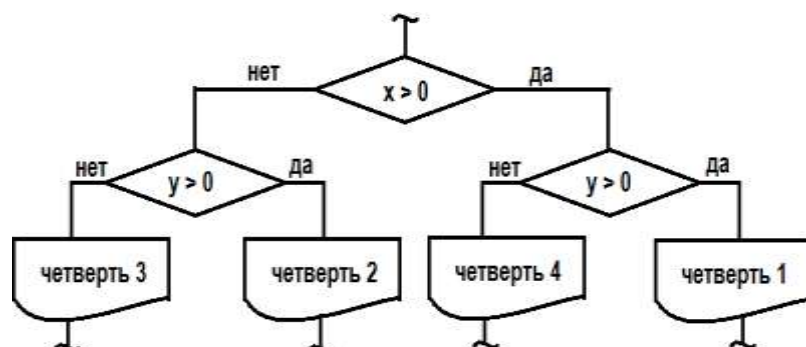


```
var  
    x,y: integer;      // Координаты точки  
    Quater: integer;  // Номер четверти  
begin  
    write('Введите координаты точки: ');  
    readln(x,y);  
    if (x>0) and (y>0) then  
        Quater := 1;  
    if (x>0) and (y<0) then  
        Quater := 4;  
    if (x<0) and (y>0) then  
        Quater := 2;  
    if (x<0) and (y<0) then  
        Quater := 3;  
    writeln('Номер четверти = ',Quater);  
end.
```

Часть программы, содержащая условия, могла бы быть написана по-другому (см. таблицу ниже). В данном случае конструкция **if... then... else...** позволяет «укоротить» программу, сделать её более рациональной.

Выполняя задание, проведите эксперимент с первым и вторым вариантом решения задачи.

```
if x>0 then
  if y>0 then
    Quater := 1
  else Quater := 4
else
  if y>0 then
    Quater := 2
  else Quater := 3;
```



Сохраните исходный файл с именем **Фамилия_5_1** и изменённый с именем **Фамилия_5_2** в свою папку.

Задания для самостоятельной проработки.

6) Составьте программу, определяющую, состоит ли число из одинаковых цифр. Основная идея: необходимо использовать стандартные функции **mod** и **div**. Целая часть от деления числа на 10 должна совпадать с дробной частью от деления этого же числа на 10.

7) Составьте программу, проверяющую, является ли введённое число двузначным. Основная идея: целое число является двузначным, если оно не меньше 10 (т.е. больше 9) и не превышает 99 (т.е. меньше 100). Остальные числа двузначными не являются. Требуется объединить эти два условия. Например, так: **(x ≥ 10) and (x < 100)**. (Подумайте, как ещё можно задать это условие).

8) Напишите программу решения квадратного уравнения. Основная идея: при вводе трёх чисел – коэффициентов квадратного уравнения (a, b, c) – получаем один из трёх ответов: «уравнение не имеет корней», «корень уравнения равен ...», «корни уравнения ... и ...».



Внимание! Решить надо все задачи, каждый файл сохраните в свою папку с именем **Фамилия_№**, где № - это номер соответствующей задачи. **Домашнее задание в тетради:** выполните блок-схему алгоритма к каждой задаче.



Домашнее задание. Подготовьтесь к самостоятельной работе по линейному и разветвляющемуся алгоритмам в Pascal ABC.



Моделирование

1. Модель

1. Модель — это аналог (заместитель) оригинала, отражающий некоторые его свойства.

Исходный объект	Модель	Что отображается в модели		
		Свойства	Действия	Среда
Медведь	Плюшевый мишка	Внешний облик	Не отражены	Не отражена
Автомобиль	Игрушечная машинка	Внешний вид Основные узлы	Перемещение под действием вращения колес	Не отражена
Пингвин	Объемная композиция в зоологическом музее	Внешний облик	Не отражены	Антарктический пейзаж, приметы климата

Модель может быть близка к оригиналу, а может быть совершенно не него непохожа.

2. Разнообразие моделей зависит от целей создания моделей.

Например: игрушки — помогают ребенку познакомиться с окружающим миром, велотренажер создан для замены катания на велосипеде, искусственное сердце помогает человеку выжить.

3. Модели бывают двух видов: материальные (предметные) и нематериальные (абстрактные).

Примеры материальных моделей: игрушки, велотренажер, искусственное сердце, глобус, географическая карта и т.д.

Примеры нематериальных моделей: формулы при решении задач по математике, физике, химии; диспетчер аэропорта следит за движением самолётов на экране компьютера; Коперник в 15 веке создал математическую модель Солнечной системы.

И, тем не менее, прежде чем выполнить какую-нибудь модель, человек создает **мысленную модель объекта или явления.**

4. Что нужно, чтобы создать модель?

Чтобы создать модель объекта или явления, необходимо собрать о нем необходимую **информацию**. Собрав же информацию совершенно необязательно строить материальную модель объекта. Можно создать **информационную модель**.

Информационная модель — целенаправленно отобранная информация об объекте, которая отражает наиболее существенные для исследователя свойства этого объекта.

Например:

Художник напишет пейзаж с коттеджем (его цель — отразить красоту природы и дома в ней), а архитектор начертит план дома (его цель — построить дом как можно лучше).

Художник нарисует цветок с целью показать его красоту, биолог начертит схему цветка, опишет отдельные его части, так как его цель — изучить и классифицировать цветок, врач и фармацевт исследуют химический состав цветка с целью использовать его в лечении.

Форма представления информационной модели может быть любая!

Но для создания информационной модели в компьютере удобно представлять информационную модель в **табличном виде**:

Объект	Параметры		Действия	Среда
	Название	Значения		

Например: Табличное представление информационной модели диска. Цель: изучение носителей информации.

Объект	Параметры		Действия	Среда
	Название	Значения		
Диск	Размер	Ø120 мм	Вставлять в устройство чтения и записи. Форматировать. Записывать информацию. Считывать информацию. Хранить информацию.	Компьютер, Дисковод
	Объем	4,7 Гб		
	Вид	DVD-RW		

5. Содержание информации в таблице зависит от цели моделирования:

Примеры:

Цель: сбор сведений об успеваемости в районе.

Фамилия, имя	Школа	Класс	Средний балл
Соколов Павел	110	7а	4,6

Цель: сбор сведений для школьного медкабинета.

Фамилия, имя	Возраст	Рост	Вес	Прививки	Хронические заболевания
Соколов Павел	12	160	40	Реакция Манту 05.09.05	Сколиоз

Цель: сбор сведений для классного руководителя.

Фамилия, имя	Дата рождения	Адрес	Домашний телефон	Характер	Увлечения
Соколов Павел	15.06.92	Ленинский пр-т, 6-1-12	(495) 135-04-04	Общительный, лидер, организатор	Музыка, компьютеры, бобслей

Вопрос: Какие модели позволяют создать уроки истории, географии, физики, математики, русского языка?

2. Основные этапы моделирования.

1. Моделирование — исследование объектов путем построения и изучения их моделей.

Возникает законный вопрос: почему нельзя исследовать объект или явление «напрямую»? Ну, например, если мы хотим выяснить длину экватора Земли можно пустить человека с измерительным землемерным прибором вдоль экватора. Представили?

Зачем же создают модели:

☑ **Оригинал может не существовать в настоящем** (например: реконструкция жизни людей прошлого на уроках истории, или фантастическая книга о людях будущего, или конструктор создаёт новый прибор и моделирует отдельные его узлы для исследования);

☑ **Оригинал может быть сложным**, с большим количеством свойств, связей, отношений. Модель в этом случае упрощена, по ней исследуются только отдельные свойства (например, биологи создают модели и схемы живых организмов или различных процессов, которые в них происходят);

☑ **Модель может быть обобщением реально существующих объектов** (например: манекенщица демонстрирует модель одежды, которую потом будут носить многие, или наука география, в которой изучаются обобщенные модели рельефа, климата);

☑ **Оригинал может быть недоступен для исследования** (кратер на Луне, атом или молекула, далекие звезды, даже наше Солнце).

2. Моделями служат:

☑ Для материальных объектов — наглядные пособия, чертежи копии;

☑ Для явлений природы — модели явлений: например в 18 веке Рихман исследуя природу молнии создал модель молнии;

☑ Для процессов — модели процессов: например экологи создали модель выброса ядовитого вещества и его влияния на окружающую среду.

3. Этапы исследования объекта или явления.

При исследовании объекта или явления человек обязательно проходит через три этапа: сначала определяется прототип существующего или существовавшего явления или объекта, затем производится моделирование, потом выясняется, достигли ли исследователи требуемого результата:

ПРОТОТИП	Существующий, существовавший или проектируемый объект, явление или процесс
▼	
МОДЕЛИРОВАНИЕ	Центральный этап исследования объекта
▼	
«ПРИНЯТИЕ РЕШЕНИЯ»	В результате моделирования и исследования свойств прототипа на модели получают новую информацию и «принимают решение» — создавать ли новый объект или требуется создание новой модели и т.д.

4. Этапы моделирования

I этап	Постановка задачи
II этап	Разработка модели
III этап	Эксперимент
IV этап	Анализ результатов моделирования

I этап. Постановка задачи.

☑ **Описание задачи.** Все задачи делятся на две большие группы: «Что будет, если...?» и «Как сделать, чтобы...?». Практические же задачи чаще всего **комплексные**: состоят из двух частей. Например, перед учителем на каждом уроке стоит задача «Что будет, если объяснить новый материал?» и «Как сделать так, чтобы его поняли?».

☑ **Определение цели моделирования.**

Если задача формулируется как «Что будет, если...?», то цель моделирования — определение последствий воздействия на объект для принятия решения или эффективности управления объектом, процессом или явлением.

Если задача формулируется «Как сделать, чтобы...?», то цель моделирования — познание окружающего мира или создание объектов с заданными свойствами.

☑ **Формализация задачи.** Её проводят в виде поиска ответов на вопросы, уточняющие общее описание задачи.

II этап. Разработка модели.

☑ **Информационная модель.**

Если мы знаем цель моделирования, то это определит выбор существенных данных, сложность модели, параметры объектов, найденные при формализации задачи.

При моделировании учитывают лишь некоторые свойства объекта-прототипа, потому что учесть все свойства объекта невозможно, да и не нужно.

Варианты построения информационной модели:

1. **Создание нескольких моделей.** Начинают с самой простой, а затем ее усложняют и уточняют, пока она не окажется подходящей. Так делают, **если задача не очень точно сформулирована**, если неизвестно точное количество необходимых параметров.

2. **Создание одной модели, если задача заранее упрощена, цель четко поставлена, параметры, которые необходимо учесть в модели, чётко определены** (например, задачи, которые решают в алгебре или физике).

В результате получают информационную модель в виде таблицы, схемы, текста.

☑ **Компьютерная модель.**

До создания компьютеров после информационной модели делали макет или пытались смоделировать явление в лабораторных условиях. С появлением же компьютеров появилась возможность создавать **компьютерную модель**.

Это **информационная модель, реализованная средствами программной среды** (например, схема, выполненная в графическом редакторе, или диаграмма,

построенная специальными программами, или динамическая (меняющаяся) модель явления, позволяющая посмотреть его изменения в течение времени. Дизайнеры используют специальные программы, позволяющие «собрать» интерьер помещения и «рассмотреть» его со всех сторон).

Первоначально с помощью компьютерной модели решались только вычислительные задачи, теперь же компьютерные модели позволяют выстроить и образную модель объекта или явления (например, программистам удалось смоделировать движение воздушных масс над поверхностью земли или диспетчеры движения поездов работают с компьютерной схемой движения).

III этап. Эксперимент.

Эксперимент — *опыт, который производится с объектом или моделью* (например, опыты по физике или биологии, испытание новой продукции, лекарства). Для таких опытов создаются специальные установки, либо испытывается непосредственно продукт. Лабораторные и натурные эксперименты требуют больших материальных затрат и времени, иногда могут быть опасны для жизни и здоровья исследователей и окружающих, но они необходимы (например, испытание космических кораблей — это очень долго, дорого и неудобно).

Компьютерный эксперимент позволяет иногда обойтись без сложных натурных экспериментов, компьютер сам моделирует объект или явление, позволяет его исследовать со всех интересующих исследователя сторон.

Этапы компьютерного эксперимента:

1. Составляется **план эксперимента** для всестороннего тестирования модели (проверки правильности построенной модели). Тест — это набор исходных данных, позволяющих определить правильность построения модели. Например: в ЛогоМирах составленная программа — это модель, а запущенная процедура — это тестирование программы-модели.
2. **Проведение исследования**, соответствуют ли полученные результаты ожидаемым, исследование свойств модели, ее параметров, жизнеспособности (т.е. работает ли она).

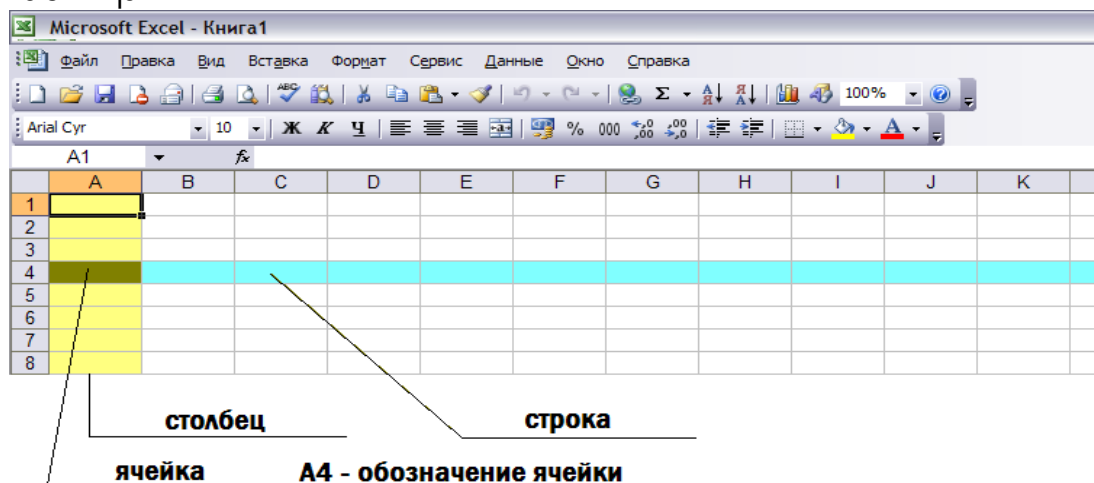
IV этап. Анализ результатов моделирования.

После этого этапа решается вопрос продолжения исследования. Если результат исследования соответствует поставленным целям, то исследование закончено. Если нет — выясняется, на каком этапе моделирования произошла ошибка. Тогда либо исправляется формулировка задачи (неверные исходные данные), либо меняют формализацию задачи (уточняют параметры, необходимые для работы), либо исправляют информационную или компьютерную модель.



Моделирование в среде MS Excel

MS Excel – разновидность электронной таблицы. Это программа обработки числовых данных, хранящая и обрабатывающая данные в прямоугольных таблицах.



1. Структура электронных таблиц

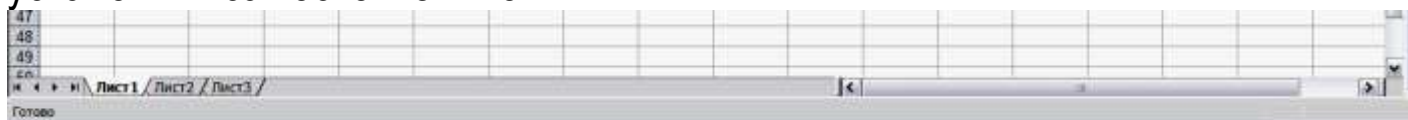
Электронная таблица состоит из **столбцов** (заголовков – буква) и **строк** (заголовков – число).

Ячейка – место пересечения **столбца** и **строки**.

Адрес ячейки – заголовок столбца + заголовок строки.

Рабочий лист (электронная таблица) – основной тип документа, используемый для хранения и обработки данных

Имя листа (по умолчанию) – "Лист1", "Лист2" и т.д. **Имя листа** можно установить самостоятельно.

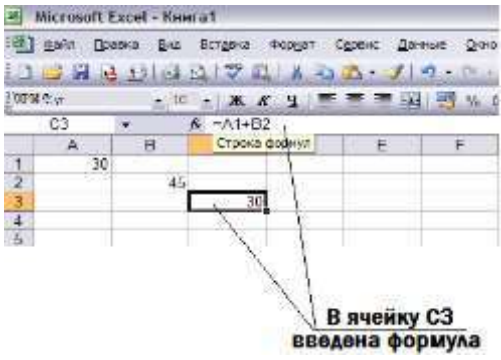


Рабочая книга – файл электронных таблиц, состоящий из нескольких листов.

2. Типы и форматы данных

Типы данных

Число	Числа могут быть записаны в обычном числовом или экспоненциальном формате (195,2 или 1,952E+02) По умолчанию выравниваются в ячейке по правому краю (для удобства выравнивания по разрядам)
Текст	Последовательность символов, состоящих из букв, цифр и пробелов (например, "32 Мегабайт") По умолчанию выравнивается в ячейке по левому краю (традиционный способ письма)

Формулы	 Начинается со знака равенства. Может включать в себя числа, имена ячеек, функции и знаки математических действий Например: =A1+B2 (сложение чисел, содержащихся в ячейках A1 и B2) =A1*9 (умножение числа, содержащегося в ячейке A1 на число 9) В ячейке отображается не формула, а результат выполнения введенной операции.
-----------------------	---

Формат данных (зависит от решаемой задачи) [Меню Формат - Ячейки...]

Основные форматы

Числовой	Используется для отображения чисел (по умолчанию отображаются два десятичных знака) - 195,20
Экспоненциальный	Применяется, если число, отображаемое в ячейке, имеет большое количество разрядов и не помещается в ячейке (2000000000 - 2,00E+09)
Денежный	Применяется для бухгалтерских расчётов (отображаются суммы в руб. и коп.)
Дата и время	Позволяют хранить значения временных данных
Текстовый	Предназначен для хранения текстов
Процентный	Предназначен для отображения чисел в виде процентов (1 автоматически отобразится как 100%)

3. Упражнение. Относительные и абсолютные ссылки

Электронная таблица состоит из **столбцов** (заголовков – буква) и **строк** (заголовков – число).

Ячейка – место пересечения **столбца** и **строки**.

Адрес ячейки – заголовок столбца + заголовок строки (**A1**).

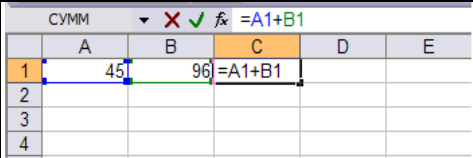
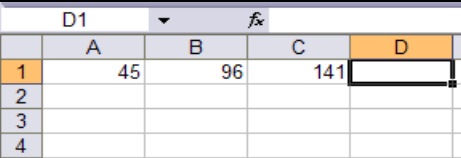
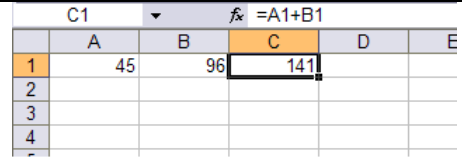
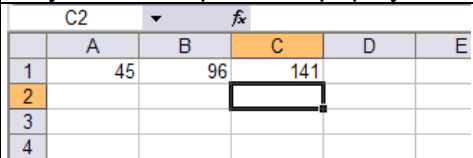
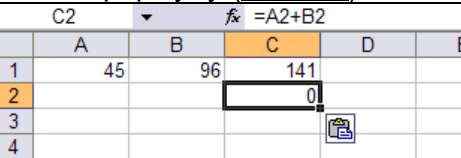
Ссылка на адрес ячейки – используется в формулах.

Виды ссылок – **относительные** и **абсолютные** (различия проявляются при копировании формулы из одной ячейки в другую).

Относительная ссылка – используется для указания адреса ячейки, вычисляемого относительно ячейки, в которой находится формула

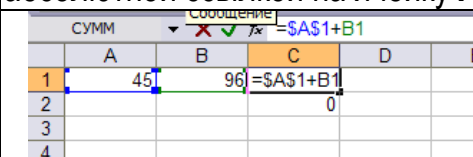
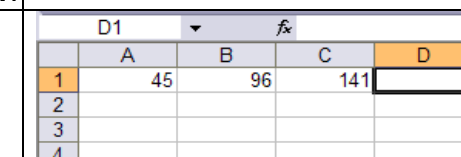
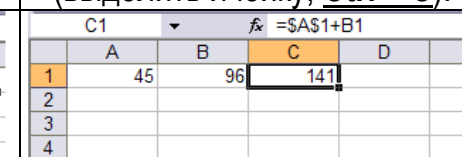
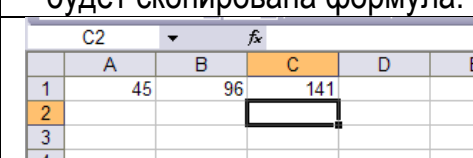
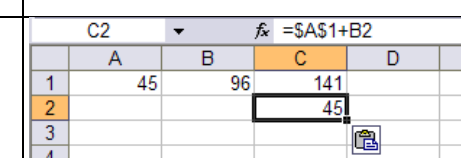
Вид: A1.

Пример 1:

1. В ячейку C1 введена формула с относительными ссылками:	2. Результат работы формулы:	3. Копировать формулу (выделить ячейку, Ctrl + C):
		
4. Выделить ячейку, в которую будет скопирована формула:	5. Вставить скопированную формулу (Ctrl + V):	Обратите внимание на адреса Ячеек в формуле, получившиеся после вставки
		

Абсолютная ссылка – используется для указания фиксированного адреса ячейки. **Перед неизменяемым значением адреса ячейки используется знак "\$"**
Вид – \$A\$1 (в формуле обращение будет всегда к ячейке **A1**).

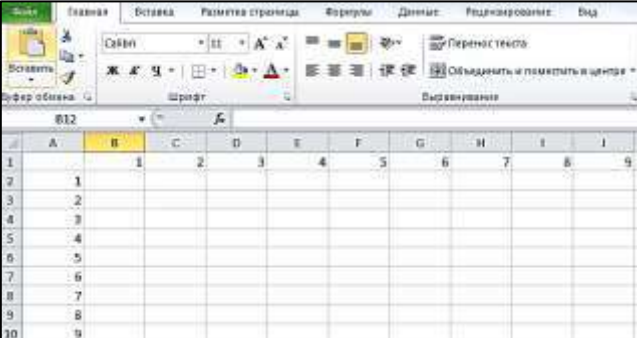
Пример 2:

1. В ячейку C1 введена формула с абсолютной ссылкой на ячейку A1 :	2. Результат работы формулы:	3. Копировать формулу (выделить ячейку, Ctrl + C):
		
4. Выделить ячейку, в которую будет скопирована формула:	5. Ctrl + V :	Обратите внимание на адреса Ячеек в формуле, получившиеся после вставки и на то, чем отличается результат вставки от первого случая.
		

Проект 1. Создание табличных информационных моделей. Таблица квадратов

Этап 1.

1. Запустить **Excel** (Пуск – Все программы - ... или нажать на иконку ).
2. Ввести числа в ячейки так, как показано на рисунке:

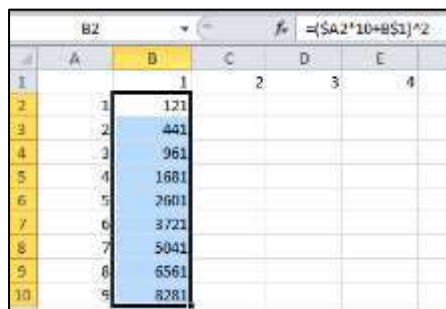


	A	B	C	D	E	F	G	H	I	J
1		1	2	3	4	5	6	7	8	9
2										
3										
4										
5										
6										
7										
8										
9										
10										

3. Ввести в ячейку **B2** формулу: **=(A2*10+B\$1)^2** (**\$A2** – обращение только к столбцу **A**, **B\$1** – обращение только к строке **1**, ^2 - возведение в квадрат);

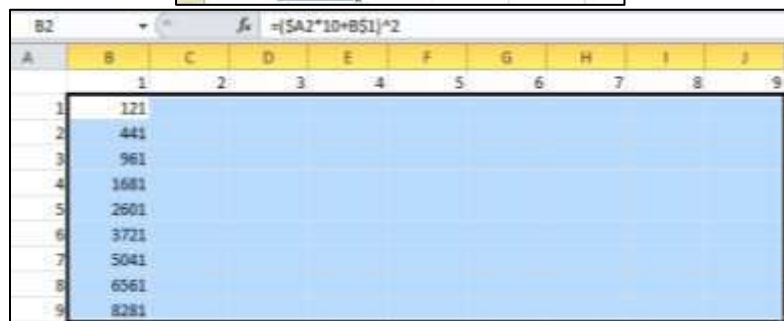
4. Выделить ячейки **B2 – B10**,

нажать на стрелку у кнопки  (вкладка **Главная**, область задач **Редактирование**) и выбрать **Вниз**. Получится:



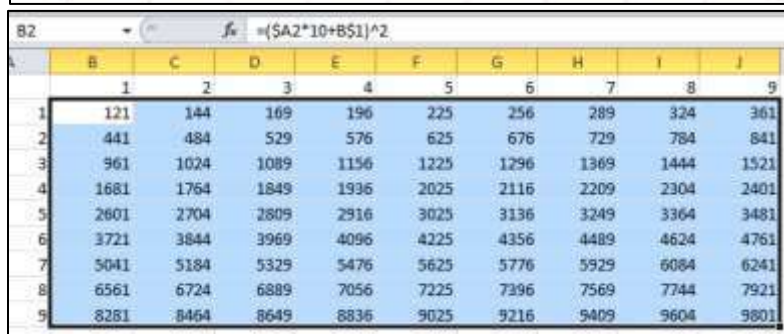
	A	B	C	D	E	F
1		1	2	3	4	
2		121				
3		441				
4		961				
5		1681				
6		2601				
7		3721				
8		5041				
9		6561				
10		8281				

5. Выделить ячейки с **B2 - J2 до B10 - J10**:



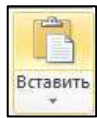
	A	B	C	D	E	F	G	H	I	J
1		1	2	3	4	5	6	7	8	9
2		121								
3		441								
4		961								
5		1681								
6		2601								
7		3721								
8		5041								
9		6561								
10		8281								

6. Нажать на стрелку у кнопки  (вкладка **Главная**, область задач **Редактирование**) и выбрать **Вправо**. Получится:



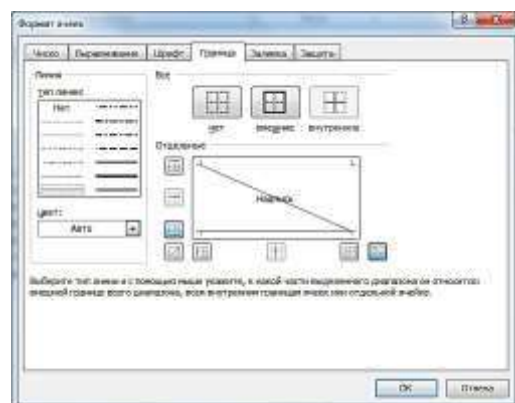
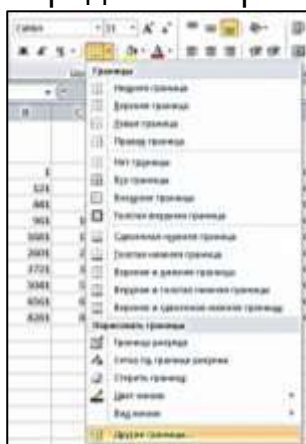
	A	B	C	D	E	F	G	H	I	J
1		1	2	3	4	5	6	7	8	9
2		121	144	169	196	225	256	289	324	361
3		441	484	529	576	625	676	729	784	841
4		961	1024	1089	1156	1225	1296	1369	1444	1521
5		1681	1764	1849	1936	2025	2116	2209	2304	2401
6		2601	2704	2809	2916	3025	3136	3249	3364	3481
7		3721	3844	3969	4096	4225	4356	4489	4624	4761
8		5041	5184	5329	5476	5625	5776	5929	6084	6241
9		6561	6724	6889	7056	7225	7396	7569	7744	7921
10		8281	8464	8649	8836	9025	9216	9409	9604	9801

7. Выделите область **A1:J10**,  (вкладка **Главная**, **Буфер обмена**, **Вырезать**);

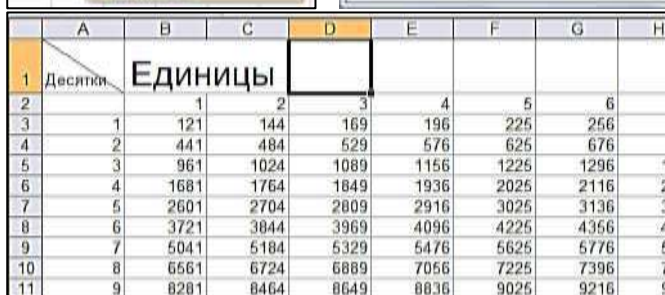
8. Выделить ячейку **A2**,  (вкладка **Главная**, **Буфер обмена**);

9. Подведите курсор к нижней границе заголовка **строки 1**  (курсор должен превратиться в двойную стрелку) и передвиньте границу вниз.

10. Выделите ячейку **A1**, нажмите на стрелку у кнопки , в открывшемся меню выберите **Другие границы**. В открывшемся окне выберите косую границу.



11. Введите в ячейку **A1** слово **ДЕСЯТКИ**, а в ячейку **B1** - **ЕДИНИЦЫ**. Измените шрифт и размер шрифта на своё усмотрение:



	A	B	C	D	E	F	G	H
1	Десяти	Единицы						
2		1	2	3	4	5	6	
3	1	121	144	169	196	225	256	
4	2	441	484	529	576	625	676	
5	3	961	1024	1089	1156	1225	1296	
6	4	1681	1764	1849	1936	2025	2116	
7	5	2601	2704	2809	2916	3025	3136	
8	6	3721	3844	3969	4096	4225	4356	
9	7	5041	5184	5329	5476	5625	5776	
10	8	6561	6724	6889	7056	7225	7396	
11	9	8281	8464	8649	8836	9025	9216	

12. Выделите область **A1:J1**, установите **Внешние границы** . Вокруг таблицы появится граница:

Десятки \ Единицы	1	2	3	4	5	6	7	8	9
1	121	144	169	196	225	256	289	324	361
2	441	484	529	576	625	676	729	784	841
3	961	1024	1089	1156	1225	1296	1369	1444	1521
4	1681	1764	1849	1936	2025	2116	2209	2304	2401
5	2501	2704	2809	2916	3025	3136	3249	3364	3481
6	3721	3844	3969	4096	4225	4356	4489	4624	4761
7	5041	5184	5329	5476	5625	5776	5929	6084	6241
8	6561	6724	6889	7056	7225	7396	7569	7744	7921
9	8281	8464	8649	8836	9025	9216	9409	9604	9801

13. Выделите область **A2:J10**, установите  (**Все границы**):

Десятки \ Единицы	1	2	3	4	5	6	7	8	9
1	121	144	169	196	225	256	289	324	361
2	441	484	529	576	625	676	729	784	841
3	961	1024	1089	1156	1225	1296	1369	1444	1521
4	1681	1764	1849	1936	2025	2116	2209	2304	2401
5	2501	2704	2809	2916	3025	3136	3249	3364	3481
6	3721	3844	3969	4096	4225	4356	4489	4624	4761
7	5041	5184	5329	5476	5625	5776	5929	6084	6241
8	6561	6724	6889	7056	7225	7396	7569	7744	7921
9	8281	8464	8649	8836	9025	9216	9409	9604	9801

14. На предварительном просмотре это будет выглядеть так:

Десятки \ Единицы	1	2	3	4	5	6	7	8	9
1	121	144	169	196	225	256	289	324	361
2	441	484	529	576	625	676	729	784	841
3	961	1024	1089	1156	1225	1296	1369	1444	1521
4	1681	1764	1849	1936	2025	2116	2209	2304	2401
5	2501	2704	2809	2916	3025	3136	3249	3364	3481
6	3721	3844	3969	4096	4225	4356	4489	4624	4761
7	5041	5184	5329	5476	5625	5776	5929	6084	6241
8	6561	6724	6889	7056	7225	7396	7569	7744	7921
9	8281	8464	8649	8836	9025	9216	9409	9604	9801

4. Можно изменить цвет и начертание шрифта. Сохраните проект в папку своего класса с именем **Фамилия_1**.

Если выделять отдельные ячейки, то в строке формул видно:

$=(\$A8*10+D\$2)^2$

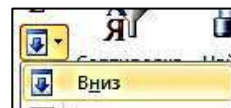
или

$=(\$A6*10+G\$2)^2$

, т.е. во всех формулах обращение к столбцу **A** (строка меняется) для **десятков** и строке **2** (столбец меняется) для **единиц**.

Проект 2. Создание графических моделей. Построение графиков

1. Запустить Excel.
2. Ввести в столбец A числа от -4 до 4 с шагом 0,5.
3. Ввести в ячейку B1 формулу: $= A1^2$.
4. Выделить ячейки B1:B17.

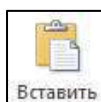


5. Вкладка Главная – область задач Редактирование – кнопка

6. Выделить ячейки A1:B17.

7. Вырезать  (вкладка Главная, область задач Буфер обмена).

8. Выделить ячейку A2.



9. Вставить  (вкладка Главная, область задач Буфер обмена).

	A	B	C
1			
2	-4	16	

10. Опустить границу строки 1:

11. Ввести в ячейку A1 заголовок: Таблица для функции $y = x^2$.

Чтобы получить надстрочную "2" - выделить её — вкладка Главная — область задач

Шрифт () — установите  надстрочный.

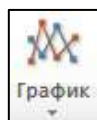
	A	B	C
1	Таблица для функции $y = x^2$		
2	-4	16	

Должно получиться примерно так:

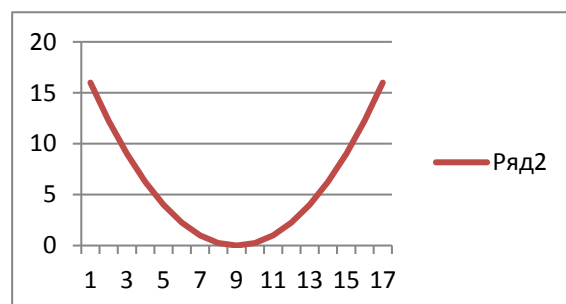
12. Выделите область B2:B18

13. Запустите Мастер диаграмм: вкладка Вставка —

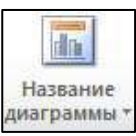
область задач Диаграммы — График

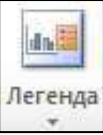


выбрать . Получится примерно так:



14. На появившейся вкладке Макет – область задач Подписи:

— Название диаграммы  — Над диаграммой. Введите надпись:

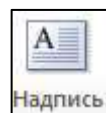
«График функции $y = x^2$ ». Легенда  — Нет.

15. На вкладке Макет — область задач Оси: Оси  — Основная

вертикальная ось — Нет. Сетка  — Вертикальные линии сетки по основной оси — Основные линии сетки.

16. На вкладке Макет — область задач Вставка —  — вставить линию со стрелкой (изменить её толщину) — расположить её вертикально. Вставить линию с стрелкой — расположить её горизонтально.

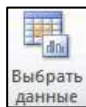
17. На вкладке Макет — область задач Вставка —



— вставить название оси Y, далее, аналогично — вставить название оси X. Получится примерно так:



Дополнительно: выделить значения Горизонтальной



оси – Выбрать данные — Подписи горизонтальной оси (по категориям) — Изменить — ввести: =Лист1!\$A\$2:\$A\$18. Получится так:

Можно добавить промежуточные линии сетки и растянуть диаграмму до обычного вида графика $y = x^2$.



18. Сохраните свой проект в папку класса с именем Фамилия_2.

Дополнительное задание*. Построить график функции $y = 2x^2 - 3x + 6$.

Проект 3. Статистический анализ табличных моделей. Использование встроенных функций и построение диаграмм

1. Запустить Excel.
2. Введите таблицу (левый столбец – в столбец А, правый – в В).
3. В ячейку С1 введите заголовок «Экспорт газа в %».
4. Выделите ячейки В2:В18 — вкладка Главная — область задач Редактирование — нажмите кнопку



- в ячейке В18 появится суммарное значение. В ячейку А18 введите слово «всего:».

5. В ячейку С2 введите формулу: =В2/\$В\$18. Выделите ячейки С2:С17 и нажмите кнопку Заполнить



— Вниз (вкладка Главная, область задач Редактирование). Не снимая выделение измените формат на Процентный:  и увеличьте разрядность:  (вкладка Главная — область задач Число).

6. Не снимая выделение выберите вкладка Вставка — область задач Диаграмма.



Выберите вид диаграммы – Гистограмма

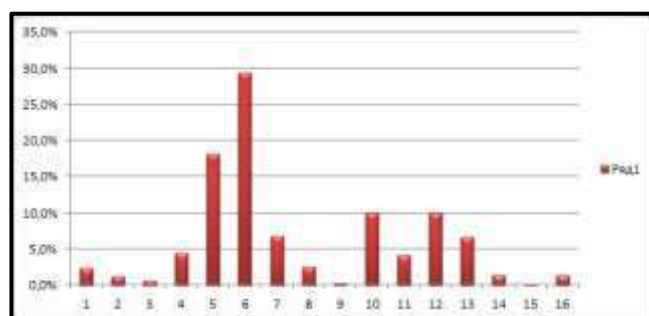


Выберите вид диаграммы

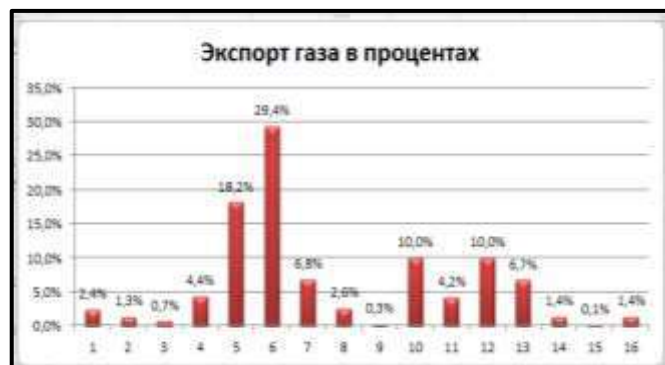
7. В появившейся вкладке Конструктор в области задач Стили диаграмм выберите любой понравившийся вам стиль диаграммы,



например, такой: . Должно получиться примерно следующее:



8. На вкладке Макет в области задач Подписи установите: Легенда – нет, Подписи данных – У вершины снаружи, Название диаграммы – Над диаграммой – Экспорт газа в процентах. Получится примерно так:



9. В ячейках D5:D10 сделайте надписи

10. Поставьте курсор в ячейку E5. Введи знак «=» и нажмите на



значок ввода формул. Найдите функцию СРЗНАЧ(), введите её и в скобках впишите аргумент B2:B17. Выполните аналогичные действия для других значений, выбирая нужную функцию (МЕДИАНА(), ДИСП(), МАКС(), МИН()).

Для вычисления Размаха введите формулу: = E8 – E9.

11. Сохраните проект папку своего класса с именем Фамилия_3.

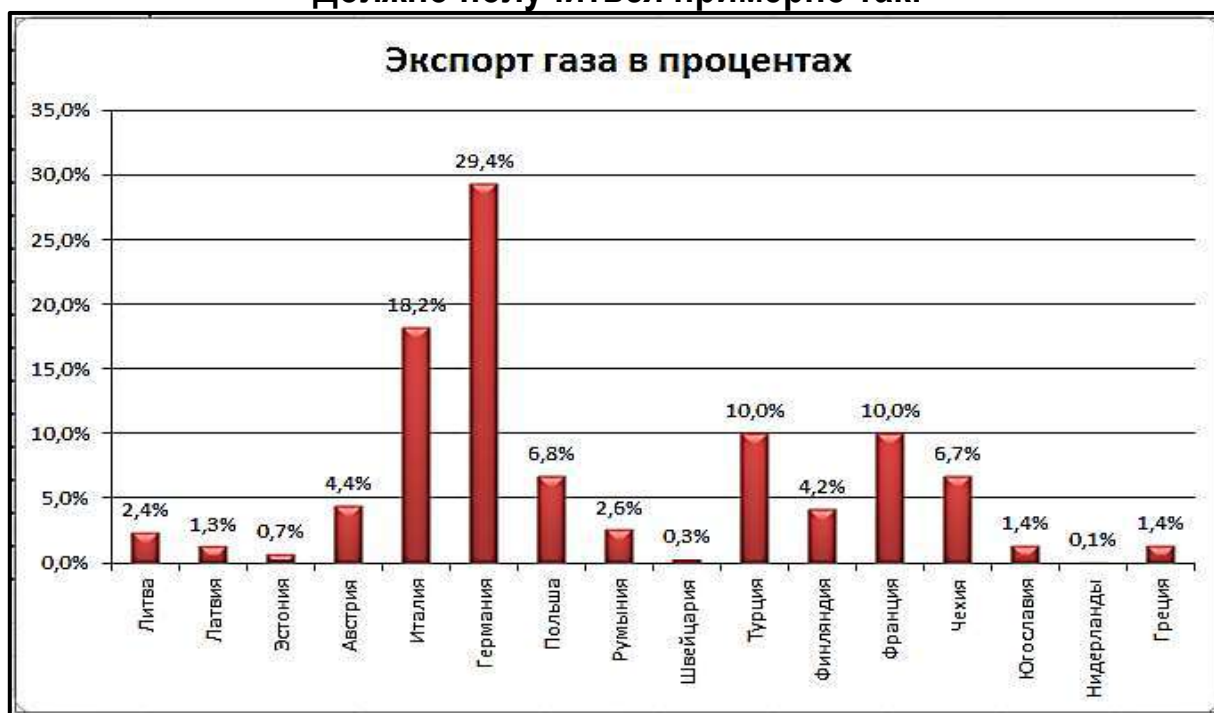
Дополнительно:

— выделите значения горизонтальной оси (числа 1 — 16 снизу), вкладка Конструктор — область задач Данные — кнопка Выбрать данные — Подписи горизонтальной оси (по категориям) — Изменить — введите диапазон: =Лист1!\$A\$2:\$A\$17.

— (не снимая выделение) на вкладке Формат — область задач Текущий фрагмент — Формат выделенного — меню Выравнивание — Направление текста — Повернуть весь текст на 270°.

Среднее:
Медиана:
Дисперсия:
Наибольшее:
Наименьшее:
Размах:

Должно получиться примерно так:



Проект 4. «Круговые диаграммы в электронных таблицах Excel. Использование реальных данных»

1. Материал взят с портала <http://budget.mos.ru>. Выберите одно из направлений, например:



Например вы выбрали «Адресную инвестиционную программу».

2. Откройте MS Excel. Введите в ячейку **A1** надпись «Адресная инвестиционная программа». В ячейки **A2 – A7** введите наименования разделов инвестиционной программы (расширьте столбец **A**).

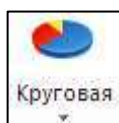
3. Заполните ячейки **B2 – B7** соответствующими данными.

4. Выделите ячейки **B2 – B8** и нажмите кнопку – Сумма (вкладка Главная, область задач Редактирование). В ячейке **B8** появится сумма данных ячеек **B2 – B7**.

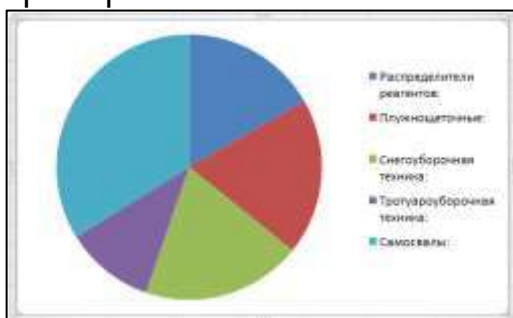
5. В ячейку **C2** введите формулу: $= B2/\$B\8 . Установите для этой ячейки формат (вкладка Главная, область задач Число).

6. Выделите ячейки **C2 – C7** и нажмите кнопку – Вниз (вкладка Главная, область задач Редактирование).

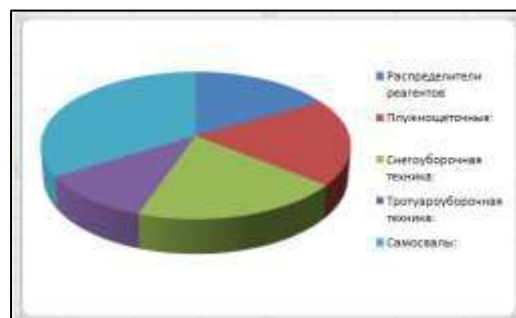
7. Выделите область ячеек **A2:B7**, на вкладке Вставка, в области задач Диаграммы



выберите тип диаграммы и выберите любой из понравившихся вам видов Круговой диаграммы, например, такой или такой . Должно получиться примерно так:



или так:



На вкладке Конструктор можете установить любой понравившийся стиль.

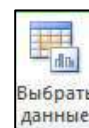


8. На вкладке Макет, нажмите кнопку  и выберите подходящее для вас расположение подписей данных.

9* (дополнительно). Можно поменять исходные цвета частей диаграммы. Для этого на диаграмме выделите какую-нибудь часть, на вкладке Макет найдите Формат выделенного (область задач Текущий фрагмент, слева на ленте), в появившемся окне откройте вкладку Заливка, выберите любой вид заливки и установите её.

10. Выделите ячейки **C2 – C7**, на вкладке Вставка, область задач Диаграмма, выберите тип диаграммы Круговая, выберите любой из понравившихся вам видов Круговой диаграммы. По желанию – установите один из стилей.

11. На вкладке Макет, нажмите кнопку Подписи данных и выберите подходящее для вас расположение подписей данных.



12. На второй диаграмме выделите легенду, на вкладке Конструктор нажмите в области задач Данные. В открывшемся окне нажмите Изменить (Подписи горизонтальной оси (категории)) и введите **=Лист1!A2:A7**.

13. Расположите обе диаграммы рядом. Сравните значения данных и общий вид обеих диаграмм. Свои наблюдения запишите в свободной ячейке под таблицей данных.

14. Сохраните свою работу в папке своего класса под своей фамилией.

 Подготовьтесь к контрольной работе по моделированию в среде MS Excel и тестированию по основным понятиям темы «Моделирование».



Гипертекст. Создание HTML-страницы в приложении Блокнот

1. Краткая теория

При создании Web-сайта используется язык разметки гипертекстовых документов - HTML:

Текстовый документ	+	Управляющие символы (тэги)	=	Web-страница
--------------------	---	----------------------------	---	--------------

Достоинства HTML-документа:

■ Малый информационный объём;

■ Возможность просмотра на компьютере с любой операционной системой.

- 1) Вид Web-страницы задаётся **тэгами**, заключёнными в угловые скобки: <HTML>.
- 2) Тэги могут быть одиночными или парными.
- 3) Парные тэги, например открывающий <HTML> и закрывающий </HTML> — называются **контейнеры**.

4) Список основных тэгов:

<HTML></HTML>	В этот контейнер помещается HTML-код страницы
<HEAD></HEAD>	В этот контейнер помещается заголовок Web-страницы
<TITLE></TITLE>	В этот контейнер помещается название Web-страницы
<BODY></BODY>	В этот контейнер помещается основное содержание страницы: текст, таблицы, бегущая строка, ссылки на графические и звуковые файлы и т.д. <BODY bgcolor="#FFFFCC"> -- задан цвет фона страницы.

2. Создание HTML-страницы

- 1) Создать в папке своего класса свою папку. Имя папки – фамилия, набранная латинскими буквами.

Основное правило: имена всех файлов, имеющих отношение к HTML-странице (сама страница, вставленные рисунки, подгружаемые файлы и т.д.) должны быть обозначены **латинскими буквами**.

- 2) Открыть Блокнот и написать следующие тэги:

```

<HTML>
<HEAD>
<TITLE> МОЯ СТРАНИЦА </TITLE>
</HEAD>
<BODY> Эта страница расскажет обо мне.
</BODY>
</HTML>

```

- 3) **Сохранить этот файл как index.htm в своей папке.** Откройте этот файл в Браузере. Обратите внимание, где появляются слова МОЯ СТРАНИЦА, а где предложение «Эта страница расскажет обо мне.»

<H1></H1>	Контейнер, определяющий размер шрифта (H1 — самый крупный шрифт, H6 — самый мелкий шрифт) + атрибут ALIGN="___", выравнивание ("right", "left", "center")
 	Контейнер, который задаёт параметры форматирования шрифта + атрибуты FONT — гарнитура шрифта (FONT = "Arial") SIZE — размер шрифта (SIZE=4) COLOR — цвет шрифта (COLOR = "blue". Может принимать значения "blue", "red", "green" и т.д. или соответствующий 16-теричный код цвета ("#RRGGBB"). Например, "blue" = "#0000FF")

4) Внесите в файл изменения:

```
<HTML>
<HEAD>
<TITLE> МОЯ СТРАНИЦА </TITLE>
</HEAD>
<BODY><FONT face="Franklin Gothic Medium Cond" COLOR="#003366"><H1
ALIGN="center"> Эта страница расскажет обо мне.</H1> </FONT>
</BODY>
</HTML>
```

5) Файл – Сохранить. Файл, открытый в Браузере – обновите.

!!! Обратите внимание: все изменения по содержанию страницы «опускаем» в контейнер <BODY></BODY>. Все дальнейшие изменения и дополнения для содержимого страницы добавляем туда же.

6) Вставьте после закрывающего тэга с новой строки:

```
<table border="0" width="100%" cellpadding="0" style="border-collapse: collapse">
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
</table>
```

Тэг <table></table> отвечает за внешний вид и расположение таблицы;

Тэг <tr></tr> отвечает строку, <td></td> отвечает за ячейки.

Таким образом мы вставляем таблицу в одну строку с двумя ячейками.

Вставка означает, что в данный момент ячейка пуста.

7) Заполните вторую ячейку (нижний <td> </td>): вместо вставьте:

```
<font face="Franklin Gothic Medium Cond" size="4"
color="#003366"></font>
```

Затем **перед** закрывающим тэгом вставьте свой текст.

3. Очень краткий справочник тэгов

<HR>	Одиночный тэг «горизонтальная линия»
<P></P>	Контейнер, разделяющий текст на абзацы
	Одиночный тэг вставки изображения + атрибуты: SRC= “__” — указание пути к файлу (если файл лежит в той же папке, то указывается только имя файла) ALT= “__” — поясняющий текст ALIGN= “__” — расположение рисунка относительно текста (TOP (верх), MIDDLE (середина), BOTTOM (вниз), LEFT, (слева), RIGHT(справа))
<A>	Контейнер гиперссылки + атрибут HREF = “__”, где указывается адрес перехода [Классификация]
<ADDRESS></ADDRESS>	Контейнер, в котором указывается адрес электронной почты: <ADDRESS> E-mail: mmm@mail.ru </ADDRESS>
	Контейнер для нумерованного списка
	Тэг, определяющий элемент списка + атрибут TYPE=“__” , определяющий тип нумерации (по умолчанию (без атрибута) — арабские цифры, I — римские, а — прописные буквы.)
	Контейнер для маркированного списка
	Тэг, определяющий элемент списка + атрибут TYPE=“__” , “disc” (диск), “square” (квадрат), “circle” (окружность)
<DL></DL>	Контейнер для списка определений
<DT>	Тэг для выделения термина
<DD>	Тэг для определения



Приложение MS FrontPage

MS FrontPage – редактор, предназначенный для создания HTML-документов, сайтов, Web-узлов.

HTML-документы (Web-страницы) - отдельные страницы, создаваемые специально для размещения на сайтах в Интернете.

Сайты – несколько HTML-документов и папок, объединённых определённой тематикой, общим дизайном. Страницы сайта, как правило, связаны системой **гиперссылок**, называемой **НАВИГАЦИЕЙ**.

Web-узлы – разновидность сайта, создаваемого в MS FrontPage, страницы и папки которого изначально объединены. Для Web-узла проще осуществлять различные операции - вставку объектов (анимации, рисунков, файлов), создание гиперссылок и т.д.

1. Основные правила при создании HTML-страниц сайта.

Все имена файлов должны быть заданы латинскими буквами (лучше задавать «говорящие» имена);

Главная страница сайта именуется **index.htm**.

Все изображения, используемые в проекте, необходимо **сохранять в папке Image**, хранящейся вместе со всеми страницами сайта;

При внедрении изображений на страницу необходимо пользоваться меню **Вставка - Рисунок - Из файла - Image**;

Сайт лучше выглядит, если все страницы имеют **одинаковое** цветовое и шрифтовое решение, а также **одинаковое строение**;

При создании сайта **тестирование (просмотр сделанного)** лучше производить через **Обозреватель (Браузер)**.

2. Виды позиционирования объектов

Свободное позиционирование

При **Свободном позиционировании** все размещаемые в HTML-документе объекты (тексты, изображения, кнопки и т.д.) выравниваются произвольным образом: вставкой пробелов или интервалов).

+ Не требует особенных усилий при конструировании.

- При просмотре документа через браузер можно заметить, что расположение объектов совсем не такое, какое было в конструкторе FrontPage. Пропорции "искажаются", объекты смещаются. Трудно расположить объекты так, как задумано. Сайт, как правило, получается пёстрым, неудобным для просмотра.

Табличное позиционирование

При **Табличном позиционировании** все объекты сайта размещаются в ячейках таблицы.

+ Есть возможность расположить объекты каждый со своим выравниванием и форматированием. Удобно при создании Web-страниц, содержащих разнообразную информацию, которую необходимо просматривать одновременно. Например: страницы, содержащие новости - необходима "главная" колонка (содержит развёрнутое сообщение), необходимы колонки (ячейки) новостных анонсов-ссылок.

- Если все страницы вашего сайта имеют табличное позиционирование, то необходимо тщательно продумать дизайн страницы (расположение ячеек текста, изображений, навигации, выравнивание, фон и т.д.). Далее такая страница создаётся в конструкторе. Остальные страницы получают копированием и переименовыванием

Таблица единиц измерения информации

Бит	наименьшая единица измерения количества информации. Принимает значение 0 или 1 (bit – binary digit — двоичный знак)
1 байт	$= 2^3 \text{ бит} = 8 \text{ бит}$
1 кбайт	$= 2^{10} \text{ байт} = 1024 \text{ байт}$
1 Мбайт	$= 2^{10} \text{ Кбайт} = 1024 \text{ Кбайт} = 2^{20} \text{ байт}$
1 Гбайт	$= 2^{10} \text{ Мбайт} = 1024 \text{ Мбайт} = 2^{30} \text{ байт}$

Таблица соответствия десятичной, шестнадцатеричной, восьмеричной и двоичной систем счисления

Десятичная	0	1	2	3	4	5	6	7
Восьмеричная	0	1	2	3	4	5	6	7
Шестнадцатеричная	0	1	2	3	4	5	6	7
Двоичная	0000	0001	0010	0011	0100	0101	0110	0111
Десятичная	8	9	10	11	12	13	14	15
Восьмеричная	10	11	12	13	14	15	16	17
Шестнадцатеричная	8	9	A	B	C	D	E	F
Двоичная	1000	1001	1010	1011	1100	1101	1110	1111

Таблица степеней числа 2

2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1024	512	256	128	64	32	16	8	4	2	1