

Л. Л. Босова

# ИНФОРМАТИКА И ИКТ

6



ИЗДАТЕЛЬСТВО

**БИНОМ**

Л. Л. Босова

# ИНФОРМАТИКА И ИКТ

*Учебник для 6 класса*

4-е издание



Москва  
БИНОМ. Лаборатория знаний  
2012

# Оглавление

<b>Ваш учебник</b> . . . . .	7
<b>Глава 1. Компьютер и информация</b> . . . . .	9
§ 1.1. Компьютер — универсальная машина для работы с информацией . . . . .	9
§ 1.2. Файлы и папки . . . . .	11
§ 1.3. Как информация представляется в компьютере, или Цифровые данные. . . . .	16
Двоичное кодирование числовой информации . . . . .	17
Тексты в памяти компьютера . . . . .	21
Изображения в памяти компьютера . . . . .	23
§ 1.4. Единицы измерения информации . . . . .	29
<b>Глава 2. Человек и информация</b> . . . . .	31
§ 2.1. Информация и знания . . . . .	31
§ 2.2. Чувственное познание окружающего мира. . . . .	35
§ 2.3. Понятие как форма мышления. . . . .	38
Как образуются понятия . . . . .	40
Содержание и объем понятия . . . . .	45
Отношения между понятиями . . . . .	47
Определение понятия . . . . .	54
Классификация. . . . .	57
§ 2.4. Суждение как форма мышления . . . . .	59
§ 2.5. Умозаключение как форма мышления . . . . .	62

<b>Глава 3. Алгоритмы и исполнители . . . . .</b>	<b>65</b>
§ 3.1. Что такое алгоритм . . . . .	65
§ 3.2. Исполнители вокруг нас . . . . .	69
§ 3.3. Формы записи алгоритмов . . . . .	73
§ 3.4. Типы алгоритмов . . . . .	76
Линейные алгоритмы . . . . .	76
Алгоритмы с ветвлениями . . . . .	77
Алгоритмы с повторениями . . . . .	79
<b>Глава 4. Материал для любознательных . . . . .</b>	<b>84</b>
§ 4.1. Путешествие в историю чисел . . . . .	84
§ 4.2. Обозначение чисел и счет в Древнем Египте . . . . .	86
§ 4.3. Римская система счисления . . . . .	89
§ 4.4. Алфавитные системы счисления . . . . .	92
§ 4.5. Славянский цифровой алфавит . . . . .	93
§ 4.6. Ясачные грамоты . . . . .	96
§ 4.7. Позиционные системы счисления . . . . .	98
§ 4.8. Вавилонская система счисления . . . . .	99
§ 4.9. Десятичная система счисления . . . . .	100
§ 4.10. Другие позиционные системы счисления . .	102
§ 4.11. О происхождении слова «алгоритм» . . . .	103
§ 4.12. Ханойская башня, или Один замечательный алгоритм . . . . .	104
§ 4.13. Руки — первый инструмент для счета . . .	107
§ 4.14. Абак и счеты . . . . .	108
§ 4.15. Арифмометр . . . . .	110
§ 4.16. Машина Бэббиджа . . . . .	112
§ 4.17. Счетная машина Холлерита . . . . .	114
§ 4.18. Поколения ЭВМ . . . . .	116



§ 4.19. Графические исполнители. . . . .	118
Исполнитель DRAW . . . . .	118
Исполнитель LINE . . . . .	122
Исполнитель CIRCLE. . . . .	124
Случайные числа . . . . .	125
<b>Глава 5. Компьютерный практикум . . . . .</b>	<b>127</b>
Работа 1. Работаем с файлами и папками.	
Часть 1 . . . . .	127
Работа 2. Знакомимся с текстовым процессором . . . . .	128
Работа 3. Редактируем и форматируем текст. Создаем надписи . . . . .	132
Работа 4. Нумерованные списки. . . . .	135
Работа 5. Маркированные списки . . . . .	137
Работа 6. Создаем таблицы . . . . .	139
Работа 7. Размещаем текст и графику в таблице . . . . .	144
Работа 8. Строим диаграммы. . . . .	146
Работа 9. Изучаем графический редактор Paint . .	150
Работа 10. Планируем работу в графическом редакторе . . . . .	156
Работа 11. Знакомимся с векторной графикой . . .	163
Работа 12. Рисунок на свободную тему . . . . .	170
Работа 13. Создаем презентацию «Часы» . . . . .	170
Работа 14. Создаем презентацию «Времена года» .	171
Работа 15. Создаем презентацию «Скакалочка» . .	173
Работа 16. Работаем с файлами и папками.	
Часть 2 . . . . .	174
Работа 17. Создаем слайд-шоу . . . . .	175

---

<b>*Работа 18. Знакомимся со средой программирования QBasic . . . . .</b>	<b>177</b>
<b>*Работа 19. Исполнитель DRAW . . . . .</b>	<b>179</b>
<b>*Работа 20. Исполнитель LINE . . . . .</b>	<b>184</b>
<b>*Работа 21. Исполнитель CIRCLE . . . . .</b>	<b>189</b>
<b>Терминологический словарь . . . . .</b>	<b>193</b>
<b>Справочные материалы . . . . .</b>	<b>196</b>


# Ваш учебник

Одним из основных видов информационной деятельности человека является чтение. Успешное овладение большим количеством учебных материалов как по информатике, так и по другим школьным предметам во многом зависит от того, как вы читаете. Чтобы повысить скорость и эффективность чтения, желательно соблюдать следующие правила:

- Текст параграфа читайте от начала до конца целиком. Только после окончания чтения и попытки ответить на вопросы после параграфа можно читать его снова.
- Читайте целенаправленно, в процессе чтения ищите ответы на вопросы:
  1. О чем текст?
  2. Какие факты в нем приведены?
  3. Какие события, явления, процессы описаны?
  4. Что нового вы узнали?
  5. С какими мыслями автора вы не согласны?
  6. Что вам после чтения осталось непонятным?
- Старайтесь при чтении производить вертикальные движения глаз по странице.
- Старайтесь сосредоточиться на чтении. Учитесь понимать прочитанное.
- При чтении старайтесь увидеть ключевые, главные слова, выражающие смысл текста.
- Запоминайте не отдельные слова и фразы, а мысли и идеи.
- Постоянно тренируйтесь, ежедневно читая не только учебную, но и научно-популярную литературу.


В учебнике для 6 класса, как и в учебнике для 5 класса, применяются условные обозначения. Напомним, что они означают:

 — самое главное;

 — вопросы и задания для самоконтроля;

 — материал для любознательных;

 — упражнение на компьютере;

 — этим значком отмечены те умения, которые вы получите при выполнении компьютерных заданий.

Эти значки придуманы специально для того, чтобы вы могли быстрее ориентироваться в учебнике и лучше усваивать содержащуюся в нем информацию.

# Компьютер и информация

---

### § 1.1. Компьютер — универсальная машина для работы с информацией

Компьютеры совершили революцию в нашей жизни. Они изменили стиль работы, образования, торговли и развлечений людей. Их используют врачи и фермеры, учителя и банковские работники, инженеры и дизайнеры. Без компьютеров не обойтись при подготовке к изданию книг и журналов, в научных и инженерных расчетах, при создании спецэффектов в кино и на телевидении и во многих других случаях. Некоторые профессии полностью связаны с компьютерами.

Объект, пригодный для многих целей, с разнообразным назначением, выполняющий разнообразные функции, называют **универсальным**.

Компьютер — универсальная машина для работы с информацией. Слово «универсальная» подчеркивает, что компьютер может применяться для многих целей: обрабатывать, хранить и передавать самую разнообразную информацию, использоваться в самых разных видах человеческой деятельности.

Но что бы ни делал человек с помощью компьютера, это всегда работа с информацией — числами, текстами, звуками или изображениями.

Самую разнообразную информацию, представленную в форме, пригодной для обработки компьютером, называют **данными**. Обработку данных компьютер осуществляет с помощью установленных на нем программ.

Чем шире спектр программного обеспечения, тем большее число задач можно решать на компьютере.



Прочитав главу 4 «Материал для любознательных», вы узнаете много интересных сведений из истории вычислительной техники.



### **Самое главное**

Компьютер — универсальная машина для работы с информацией. Он может применяться для многих целей: обрабатывать, хранить и передавать самую разнообразную информацию, использоваться в самых разных видах человеческой деятельности.



### **Вопросы и задания**

1. Что означает слово «универсальный»?
2. Почему компьютер является универсальной машиной для работы с информацией?
3. Приведите примеры использования компьютеров.
4. Вспомните, какие программы предназначены для обработки числовой, текстовой и графической информации.



## § 1.2. Файлы и папки

Все программы и данные в памяти компьютера и на дисках хранятся в виде файлов. **Файл** — это информация, хранящаяся в долговременной памяти как единое целое и обозначенная именем.

Имя файла состоит из двух частей: собственно **имени** и **расширения**.

Имя файлу придумывает тот, кто его создает. В операционной системе Windows имя файла может иметь до 255 символов, в нем могут использоваться буквы латинского и русского алфавитов и некоторые другие символы. При этом использование русских букв и пробелов в именах файлов хотя и допустимо, но может создать серьезные проблемы (при публикации в Интернете и переносе в другие операционные системы). Имя файла не должно включать следующие символы:

\ / : \* ? " < > |

Расширение обычно автоматически задается программой, в которой вы работаете, и указывает на тип файла (хотя пользователь может задавать и нетрадиционные расширения). Оно говорит пользователю и компьютеру о том, какая информация хранится в файле и какой программой был создан этот файл. Почти всегда расширение состоит из трех букв латинского алфавита. От имени расширение отделяется точкой.

Существует большое количество типов файлов. Вот некоторые из них:



- исполнимые — файлы, содержащие готовые к исполнению программы; их можно узнать по расширениям com, exe;



- текстовые документы — имеют расширения txt, doc, rtf;



- графические — файлы, содержащие изображения; их расширения — bmp, jpg и др.;



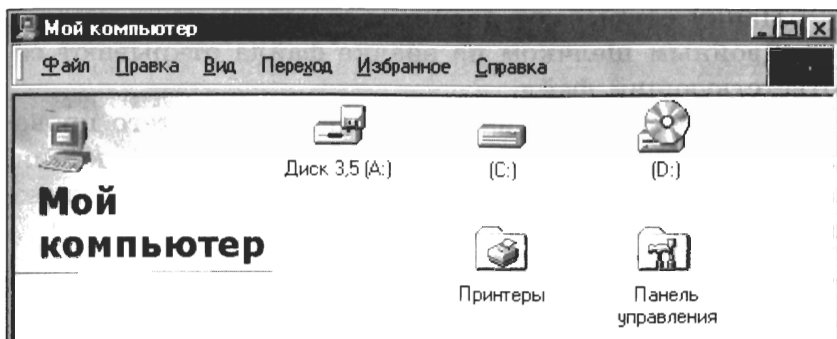
- звуковые — файлы, содержащие голоса и музыку; их расширения — wav, mid.

На жестком диске одного компьютера может храниться огромное количество файлов — десятки и даже сотни тысяч. Чтобы не возникло путаницы, все файлы хранятся в определенной системе: в папках, которые, в свою очередь, могут содержаться в других папках (быть вложенными в них) и так далее. Пусть на жестком диске компьютера записано несколько игр. Игра представляет собой набор файлов. Каждая игра хранится в отдельной папке, при этом все папки с играми для удобства могут быть вложены в одну общую папку с именем Игры.

Система хранения файлов напоминает хранение большого количества книг в библиотеке:

Библиотека	↔	Диск
Шкаф	↔	Папка
Полка	↔	Вложенная папка
Книга	↔	Файл
Название книги	↔	Имя файла

Рассмотрим одно из средств, позволяющих увидеть, какие файлы хранятся в компьютере.



Окно *Мой компьютер* открывается двойным щелчком на одноименном значке, расположенном на Рабочем столе. Окно *Мой компьютер* содержит значки всех устройств компьютера, на которых можно хранить файлы. Оно может также содержать значок *Панель управления* и значок *Принтеры*.

Чтобы узнать, что хранится на диске C:, дважды щелкните на его значке — в окне отобразится содержимое диска. Это окно может содержать несколько типов значков:



мурка



Звук

одни — для представления файлов с данными;



Mspaint



Qb

другие — для представления файлов-программ;



Common



Games

третьи — для представления папок;



Paint

кроме того, можно увидеть и ярлыки, обеспечивающие быстрый доступ к какой-либо папке, программе или документу.

Папку открывают двойным щелчком на ее значке. Двойным щелчком на значке файла открывают соответствующий файл.

Во время работы на компьютере чаще всего проводятся такие операции с файлами, как модификация, копирование, удаление и перемещение. Здесь также можно провести аналогию с книгами в библиотеке:

Библиотека	Файловая система
Реставрация книги (снять книгу с полки, заменить поврежденные страницы, поставить на место)	Модификация файла (открыть имеющийся файл, внести в него изменения и сохранить под тем же именем)
Сделать копию книги при помощи ксерокса, переплести ее и поставить в другой шкаф или на другую книжную полку. Теперь в библиотеке две одинаковые книги	Скопировать файл и сохранить его в другой папке
Убрать из библиотеки подлежащую списанию книгу	Удалить файл
Взять книгу из шкафа и поставить ее в другой шкаф или на другую книжную полку	Переместить файл

При работе с файлами не следует:

- удалять файл, точно не выяснив, обязательно ли это следует делать;
- давать файлу имя, которое не поясняет его содержание;
- сохранять файл в той папке, где его потом будет трудно найти;

- удалять или перемещать файлы, находящиеся в папках прикладных программ — это может привести к тому, что программы перестанут работать.



### **Самое главное**

Файл — это информация, хранящаяся в долговременной памяти как единое целое и обозначенная именем.

Имя файла состоит из двух частей. Имя файлу придумывает тот, кто его создает. Расширение обычно автоматически задается программой, в которой создается файл, и указывает на тип файла.

Чтобы не возникло путаницы, все файлы хранятся в определенной системе: в папках, которые, в свою очередь, могут быть вложенными в другие папки и так далее.



### **Вопросы и задания**

1. Что такое файл?
2. Из каких частей состоит имя файла?
3. Какие правила записи имени файла следует соблюдать?
4. Опишите систему хранения файлов на диске.
5. Какие операции можно совершать с файлами?
6. Каких действий следует избегать при работе с файлами?

### § 1.3. Как информация представляется в компьютере, или Цифровые данные

Для того чтобы понять, как самая разнообразная информация представлена в компьютере, «заглянем» внутрь машинной памяти. Ее удобно представить в виде листа в клетку. В каждой такой «клетке» хранится только одно из двух значений: нуль или единица. Две цифры удобны для электронного хранения данных, поскольку они требуют только двух состояний электронной схемы — «включено» (это соответствует цифре 1) и «выключено» (это соответствует цифре 0). Каждая «клетка» памяти компьютера называется **битом**. Цифры 0 и 1, хранящиеся в «клетках» памяти компьютера, называют значениями битов.

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



С помощью последовательности битов можно представить самую разную информацию. Такое представление информации называется **двоичным** или **цифровым кодированием**.

Преимуществом цифровых данных является то, что их относительно просто копировать и изменять. Их можно хранить и передавать с использованием одних и тех же методов, независимо от типа данных.

Способы цифрового кодирования текстов, звуков (голоса, музыка), изображений (фотографии, иллюст-



рации) и последовательностей изображений (кино и видео), а также трехмерных объектов были придуманы в 80-х годах прошлого века.

### Двоичное кодирование числовой информации

Известно множество способов записи чисел. Мы пользуемся десятичной позиционной системой счисления.

**Десятичной** она называется потому, что в этой системе счисления десять единиц одного разряда составляют одну единицу следующего старшего разряда. Число 10 называется основанием десятичной системы счисления. Для записи чисел в десятичной системе счисления используются десять цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8 и 9.

**Позиционной** эта система счисления называется потому, что одна и та же цифра получает различные количественные значения в зависимости от места, или позиции, которую она занимает в записи числа. Например, в записи числа 555 цифра 5, стоящая на первом месте справа, обозначает 5 единиц, на втором — 5 десятков, на третьем — 5 сотен.

Рассмотрим два числовых ряда:

1, 10, 100, 1000, 10 000, 100 000 ...

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048 ...

Оба этих ряда начинаются с единицы. Каждое следующее число первого ряда получается путем умножения предыдущего числа на 10. Каждое следующее число второго ряда получается путем умножения предыдущего числа на 2.

Любое целое число можно представить в виде суммы разрядных слагаемых — единиц, десятков, сотен, тысяч и так далее, записанных в первом ряду. При этом каждый член этого ряда может либо не входить в сумму, либо входить в нее от 1 до 9 раз. Пример:

$$1409 = 1 \cdot 1000 + 4 \cdot 100 + 0 \cdot 10 + 9 \cdot 1.$$

Числа 1, 4, 0, 9, на которые умножаются члены первого ряда, составляют исходное число 1409.

## Перевод целых десятичных чисел в двоичный код

### Способ 1

Попробуем представить число 1409 в виде суммы членов второго ряда.

Воспользуемся методом разностей. Возьмем ближайший к исходному числу, но не превосходящий его член второго ряда и составим разность:

$$1409 - 1024 = 385.$$

Возьмем ближайший к полученной разности, но не превосходящий ее член второго ряда и составим разность:

$$385 - 256 = 129.$$

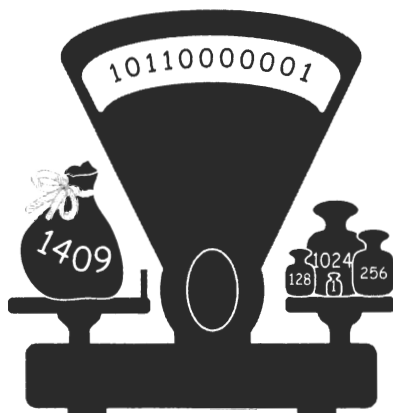
Аналогично составим разность:

$$129 - 128 = 1.$$

В итоге получим:

$$1409 = 1024 + 256 + 128 + 1 = 1 \cdot 1024 + 0 \cdot 512 + \\ + 1 \cdot 256 + 1 \cdot 128 + 0 \cdot 64 + 0 \cdot 32 + 0 \cdot 16 + \\ + 0 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 1 \cdot 1.$$

Мы видим, что каждый член второго ряда может либо не входить в сумму, либо входить в нее только один раз.



Числа 1 и 0, на которые умножаются члены второго ряда, также составляют исходное число 1409, но в его другой, двоичной записи: **10110000001**.

Результат записывают так:

$$1409_{10} = 10110000001_2.$$

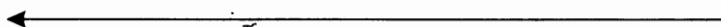
Исходное число мы записали с помощью 0 и 1, другими словами, получили двоичный код этого числа, или представили число в двоичной системе счисления.

## Способ 2

Этот способ получения двоичного кода десятичного числа основан на записи остатков от деления исходного числа и получаемых частных на 2, продолжаемого до тех пор, пока очередное частное не окажется равным 0.

Пример:

1409	704	352	176	88	44	22	11	5	2	1	0
1	0	0	0	0	0	0	1	1	0	1	



В первую ячейку верхней строки записано исходное число, а в каждую следующую — результат целочисленного деления предыдущего числа на 2.

В ячейках нижней строки записаны остатки от деления стоящих в верхней строке чисел на 2.

Последняя ячейка нижней строки остается пустой. Двоичный код исходного десятичного числа получается при последовательной записи всех остатков, начиная с последнего:  $1409_{10} = 10110000001_2$ .

Первые 20 членов натурального ряда в двоичной системе счисления записываются так: 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111, 10000, 10001, 10010, 10011, 10100.

Перевод целых чисел  
из двоичной системы счисления в десятичную

### Способ 1

Пусть имеется число  $111101_2$ . Его можно представить так:

$$\underbrace{111101_2}_{\leftarrow} = 1 \cdot 1 + 0 \cdot 2 + 1 \cdot 4 + 1 \cdot 8 + 1 \cdot 16 + 1 \cdot 32 = 61_{10} \rightarrow$$

### Способ 2

Возьмем то же число  $111101_2$ . Переведем единицу 6-го разряда (первая слева в записи числа) в единицы 5-го разряда, для чего 1 умножим на 2, ибо единица 6-го разряда в двоичной системе содержит 2 единицы 5-го разряда.

К полученным 2 единицам 5-го разряда прибавим имеющуюся единицу 5-го разряда. Переведем эти 3 единицы 5-го разряда в 4-й разряд и прибавим имеющуюся единицу 4-го разряда:  $3 \cdot 2 + 1 = 7$ .

Переведем 7 единиц 4-го разряда в 3-й разряд и прибавим имеющуюся единицу 3-го разряда:  $7 \cdot 2 + 1 = 15$ .

Переведем 15 единиц 3-го разряда во 2-й разряд:  $15 \cdot 2 = 30$ . В исходном числе во 2-м разряде единиц нет.

Переведем 30 единиц 2-го разряда в 1-й разряд и прибавим имеющуюся там единицу:  $30 \cdot 2 + 1 = 61$ . Мы получили, что исходное число содержит 61 единицу 1-го разряда.

Письменные вычисления удобно располагать так:

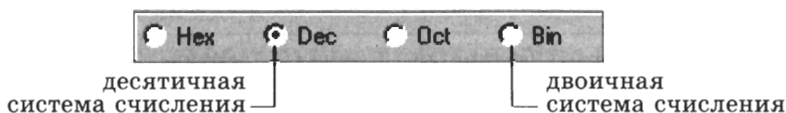
$$((((1 \cdot 2 + 1) \cdot 2 + 1) \cdot 2 + 1) \cdot 2 + 0) \cdot 2 + 1 = 61.$$

Переводить целые числа из десятичной системы счисления в двоичную систему счисления и обратно можно с помощью приложения Калькулятор.



Проведем небольшой эксперимент.

1. Запустите приложение Калькулятор и выполните команду [*Вид-Инженерный*]. Обратите внимание на группу переключателей, определяющих систему счисления:



2. Убедитесь, что Калькулятор настроен на работу в десятичной системе счисления. С помощью клавиатуры или мыши введите в поле ввода произвольное двузначное число. Активизируйте переключатель *Bin* и проследите за изменениями в окне ввода. Вернитесь в десятичную систему счисления. Очистите поле ввода.
3. Повторите пункт 2 несколько раз для других десятичных чисел.
4. Настройте Калькулятор на работу в двоичной системе счисления. Обратите внимание на то, какие кнопки Калькулятора и цифровые клавиши клавиатуры вам доступны. Поочередно введите двоичные коды 5-го, 10-го и 15-го членов натурального ряда и с помощью переключателя *Dec* переведите их в десятичную систему счисления.



В главе 4 «Материал для любознательных» вы можете узнать много интересных сведений из истории счета и систем счисления.

### Тексты в памяти компьютера

При двоичном кодировании текстовой информации чаще всего каждому символу ставится в соответствие уникальная цепочка из 8 нулей и единиц, называемая байтом. Всего существует 256 разных цепочек из 8 ну-

лей и единиц. Это позволяет закодировать 256 разных символов. Например, прописные и строчные буквы русского и латинского алфавитов, цифры, знаки препинания, другие символы. Соответствие символов и кодов задается с помощью специальной кодовой таблицы. Ниже приведен фрагмент таблицы, используемой в системе Windows:

Символ	Десятичный код	Двоичный код	Символ	Десятичный код	Двоичный код
Пробел	32	00100000	0	48	00110000
!	33	00100001	1	49	00110001
*	42	00101010	2	50	00110010
+	43	00101011	3	51	00110011
,	44	00101100	4	52	00110100
-	45	00101101	5	53	00110101
.	46	00101110	6	54	00110110
/	47	00101111	7	55	00110111
=	61	00111101	8	56	00111000
?	63	00111111	9	57	00111001
А	192	11000000	Р	208	11010000
Б	193	11000001	С	209	11010001
В	194	11000010	Т	210	11010010
Г	195	11000011	У	211	11010011
Д	196	11000100	Ф	212	11010100
Е	197	11000101	Х	213	11010101
Ж	198	11000110	Ц	214	11010110
З	199	11000111	Ч	215	11010111
И	200	11001000	Ш	216	11011000
Й	201	11001001	Щ	217	11011001
К	202	11001010	Ъ	218	11011010
Л	203	11001011	Ы	219	11011011
М	204	11001100	Ь	220	11011100
Н	205	11001101	Э	221	11011101
О	206	11001110	Ю	222	11011110
П	207	11001111	Я	223	11011111

Например, слово «ЛУНА» кодируется четырьмя десятичными числами



или двоичной последовательностью

11001011 11010011 11001101 11000000.

Чтобы узнать, какое слово закодировано двоичной последовательностью, ее нужно разбить на 8-символьные цепочки, каждой из которых поставить в соответствие некоторый символ кодовой таблицы. Например, последовательность

1100101011001101110010001100001111000000

разбиваем так:

11001010.11001101.11001000.11000011.11000000

Это соответствует слову «КНИГА».

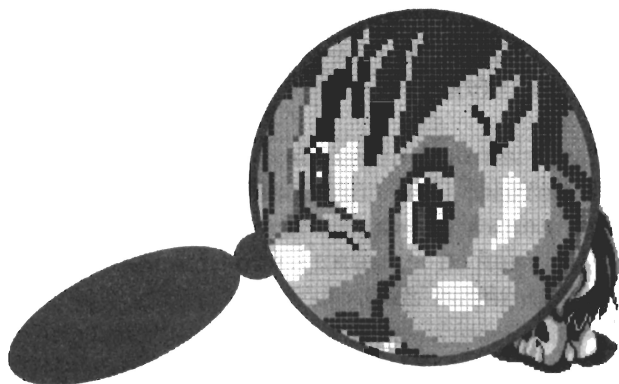
### Изображения в памяти компьютера

Последовательностями нулей и единиц можно закодировать и графическую информацию.

Существует два способа представления изображений в цифровом виде.

#### Способ 1

Графический объект, подлежащий представлению в цифровом виде, делится вертикальными и горизонтальными линиями на крошечные фрагменты — пиксели. Цвет каждого пикселя кодируется двоичным числом. Такой способ называется **растровым кодированием**.

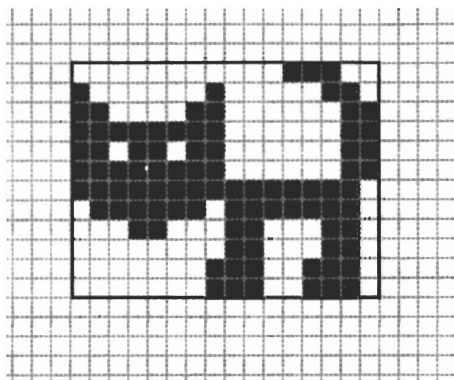


Рассмотрим простую черно-белую картинку:

```

0000000000011100
1000000100000110
1100001100000011
1111111100000011
1101101100000011
1111111100000011
1111111111111110
0111111011111110
0001100011000110
0000000011000110
0000000111001110
0000000111001110

```



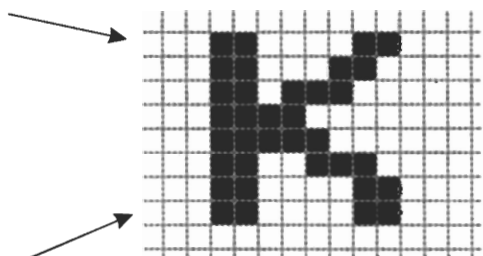
Каждую пустую (белую) клеточку рисунка, заключенного в рамку, мы закодировали нулем, а закрашенную (черную) — единицей.

Попробуем решить обратную задачу — восстановить рисунок по его коду, причем код будет десятичным. Представим имеющиеся десятичные числа в двоичном коде и закрасим клеточки, соответствующие 1:

```

195  11000011
198  11000110
220  11011100
240  11110000
248  11111000
206  11001110
195  11000011
195  11000011

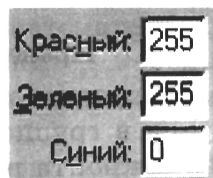
```



В рассмотренных примерах каждый пиксель кодировался 1 битом. При цифровом представлении цветных изображений каждый пиксель кодируется цепочкой из 24 нулей и единиц, что позволяет различать более 16 миллионов цветовых оттенков.

Необычайно богатая цветовая палитра современных компьютеров получается смешением взятых в опреде-

ленной пропорции трех основных цветов: красного, синего и зеленого. На кодирование каждого из них чаще всего отводится по 8 битов, в которых можно записать двоичные коды 256 различных оттенков основного цвета.



Проведем небольшой эксперимент.

1. Запустите графический редактор Paint и выполните команду [*Палитра-Изменить палитру*].
2. В открывшемся диалоговом окне *Изменение палитры* щелкните на кнопке *Определить цвет*; обратите внимание на информацию в правой нижней части экрана.
3. Задайте несколько раз по своему усмотрению значения в полях ввода для основных цветов и проследите за изменениями в окне *Цвет|Заливка*.
4. Установите, какие цвета получатся при следующих значениях основных цветов:

Красный	Зеленый	Синий	Цвет
0	0	0	
0	0	255	
0	255	0	
190	190	190	
255	0	0	
0	255	255	
255	0	255	
255	255	0	
255	255	255	

Точное число различных оттенков вы можете получить, если с помощью приложения Калькулятор вычислите значение произведения  $256 \cdot 256 \cdot 256$ .



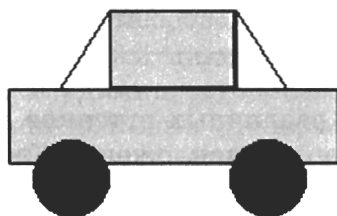
Проведем еще один эксперимент.

1. Запустите графический редактор Paint, находящийся в группе программ *Стандартные*.
2. Откройте рисунок Образец (Мои документы\бкласс\Заготовки).
3. Выполните команду [*Вид-Масштаб-Другой*], в группе *Варианты* установите переключатель 400%, дающий увеличение исходной картинку в 4 раза.
4. Самостоятельно увеличьте исходную картинку в 8 раз (переключатель 800%).
5. Выполните команду [*Вид-Масштаб-Показать сетку*]. Обратите внимание на то, что весь исходный рисунок оказался состоящим из маленьких квадратиков.
6. Выберите инструмент *Заливка* и с его помощью попытайтесь внести изменения в рисунок, перекрашивая отдельные области.
7. Выполните команду [*Вид-Масштаб-Обычный*] и проследите за сделанными изменениями.
8. Выйдите из программы (команда [*Файл-Выход*]), не внося изменений в исходный файл (кнопка *Нет* в окне *Внести изменения*).

## Способ 2

Некоторый графический объект записывается как закодированная в цифровом виде последовательность команд для его создания.

Например, чтобы выполнить следующий рисунок, необходимо изобразить два закрашенных прямоугольника, два прямоугольных треугольника и два круга:



Каждая из этих фигур может быть математически описана: прямоугольники и треугольники — координатами своих вершин, круги — координатами центров и радиусами.

Такой способ называется **векторным кодированием**.



### **Самое главное**

С помощью последовательности нулей и единиц можно представить самую разнообразную информацию. Такое представление информации называется двоичным или цифровым кодированием.

Мы пользуемся десятичной позиционной системой счисления. Существуют специальные правила, позволяющие получить двоичный код любого десятичного числа.

При двоичном кодировании текстовой информации чаще всего каждому символу ставится в соответствие уникальная цепочка из 8 нулей и единиц, называемая байтом. Соответствие символов и кодов задается с помощью специальной кодовой таблицы.

Существует два способа представления изображений в цифровом виде.

Первый способ состоит в том, чтобы графический объект, подлежащий представлению в цифровом виде, разделить вертикальными и горизонтальными линиями на крошечные фрагменты — пиксели, и закодировать цвет каждого пикселя в виде двоичного числа. Такой способ называется растровым кодированием.

Второй способ состоит в том, что некоторый графический объект записывается как закодированная в цифровом виде последовательность команд для его создания. Этот способ называется векторным кодированием.

## ? Вопросы и задания

1. Какие данные называют цифровыми?
2. Почему возникла потребность в цифровом представлении информации?
3. Как получить двоичный код целого десятичного числа? Приведите пример.
4. Как по двоичному коду восстановить соответствующее десятичное число? Приведите пример.
5. Каким образом осуществляется двоичное кодирование текстовой информации? Приведите пример.
6. Как представить в цифровом виде графическую информацию?
7. Как могут быть расшифрованы двоичные цепочки

1	1	0	1	0	0	1	0
---	---	---	---	---	---	---	---

1	1	0	0	1	1	1	0
---	---	---	---	---	---	---	---

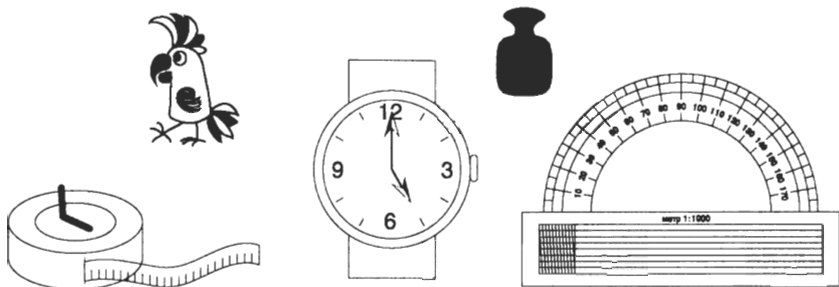
1	1	0	0	1	0	1	0
---	---	---	---	---	---	---	---

- а) калькулятором;
- б) текстовым редактором;
- в) графическим редактором?



## § 1.4. Единицы измерения информации

Вам известны единицы измерения длины. Это миллиметры, сантиметры, метры и километры. Масса измеряется в граммах, килограммах, центнерах и тоннах. Углы измеряются в градусах. Время — в секундах, минутах и часах.



Представленная в цифровом виде информация тоже может быть измерена. Единицами измерения информации являются **биты** (0 или 1) и **байты** (1 байт = 8 битов). Например, сообщение «ИНФОРМАТИКА» состоит из 11 символов, каждый из которых кодируется цепочкой из 8 нулей и единиц. Следовательно, это сообщение имеет информационный объем 88 битов, или 11 байтов.

Более крупными единицами измерения информации являются килобайты, мегабайты и гигабайты:

1 Кбайт (один килобайт) = 1024 байт;

1 Мбайт (один мегабайт) = 1024 Кбайт;

1 Гбайт (один гигабайт) = 1024 Мбайт.

На каждой странице вашего учебника помещается чуть меньше 40 строк, в каждой строке — примерно 60 символов. Следовательно, страница учебника имеет информационный объем около 2400 байтов, а весь учебник, состоящий из 176 страниц, — 422 400 байтов информации. Чтобы перейти от байтов к килобайтам, разделим это число на 1024. Получим:

422 400 байт  $\approx$  413 Кбайт. Для перехода к мегабайтам выполним деление числа 413 на 1024. Получим: 413 Кбайт  $\approx$  0,4 Мбайт.

Значительно больший информационный объем имеют графические файлы. Так изображение, состоящее из  $600 \times 800$  пикселей, каждый из которых кодируется цепочкой из 24 нулей и единиц (3 байта), имеет информационный объем  $600 \times 800 \times 3 = 1\,440\,000$  байт  $\approx$   $\approx$  1406 Кбайт  $\approx$  1,37 Мбайт.

Объем компьютерных информационных носителей также измеряется в мегабайтах и гигабайтах. Так, трехдюймовая дискета имеет объем (емкость) 1,44 Мбайт, стандартный жесткий диск — 40–120 Гбайт, компакт-диск (CD) — 650 Мбайт, цифровой диск (DVD) — от 4,7 до 17 Гбайт.

### Самое главное

Единицами измерения информации являются биты (0 или 1) и байты (1 байт = 8 битов). Более крупными единицами измерения информации являются килобайты, мегабайты и гигабайты:

- 1 Кбайт (один килобайт) = 1024 байт;
- 1 Мбайт (один мегабайт) = 1024 Кбайт;
- 1 Гбайт (один гигабайт) = 1024 Мбайт.

Емкость компьютерных информационных носителей также измеряется в мегабайтах и гигабайтах.

### Вопросы и задания

1. Что такое байт, килобайт, мегабайт и гигабайт?
2. Как байт, килобайт, мегабайт и гигабайт связаны между собой?
3. Сколько битов составляют  $\frac{1}{2}$  килобайта?
4. Сколько школьных учебников емкостью 350 Кбайт можно разместить на трехдюймовой дискете?

# Человек и информация

### § 2.1. Информация и знания

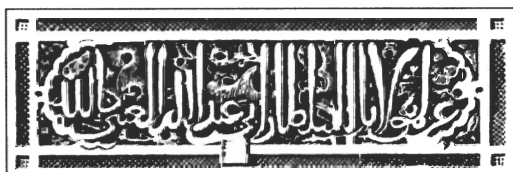
Информация играет в жизни людей огромную роль. Мы получаем ее с помощью наших органов чувств из окружающего мира, посредством разговорного языка от других людей или из книг, добываем в процессе деятельности.

Информация для человека — это знания, которые он получает из различных источников.

Человеческое общество по мере своего развития исследовало окружающий мир, накапливая о нем информацию. Знания об окружающем мире и о самом себе позволяют человеку лучше приспособиться к жизни.

Сообщение, полученное человеком, может пополнить его знания, если содержащиеся в нем сведения являются для человека понятными и новыми.

Например, вряд ли кому-то из ваших одноклассников будет понятен вот такой арабский текст:



Не зная кода, вы не сможете понять и смысл простого, на первый взгляд, сообщения: 55 288 1 498<sup>1</sup>.

<sup>1</sup> Прочитать об этом коде можно в материале для любознательных.

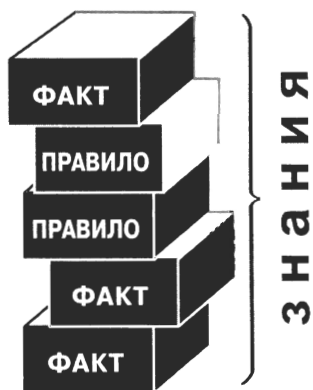
Вряд ли пополнит ваши знания, к примеру, такой вот текст на родном языке, взятый из математической энциклопедии: «Резонанс — явление увеличения амплитуды вынужденных колебаний при приближении частоты внешнего воздействия к одной из частот собственных колебаний динамической системы».

Таким образом, информация понятна, если она выражена на языке, доступном для получателя.

Вместе с тем, совершенно понятное сообщение «Треугольник — фигура, имеющая три угла и три стороны» не расширит ваши знания, так как не является новым для учеников 6 класса. Оно для вас неинформативно, хотя это же сообщение будет нести информацию для учеников 1 класса.

Таким образом, информативность конкретного сообщения зависит от жизненного опыта, знаний получателя информации.

Все знания, которыми обладает каждый из нас, условно можно разделить на две группы: факты и правила.



К фактам относятся знания об определенных явлениях (солнечные и лунные затмения происходят тогда, когда Солнце, Луна и Земля в своем видимом взаимном перемещении занимают положение на одной линии), событиях (первый персональный компьютер фирмы IBM появился в 1981 году), свойствах объектов (вес первой электронной вычислительной машины равнялся 30 тоннам) и зависимостях между объектами (у квадрата все углы прямые и все стороны равны). Описание фактов можно начинать со слов: «Я знаю, что ...».

К правилам относятся знания о последовательностях действий, направленных на достижение некоторой цели (выполнить фонетический разбор слова, вычислить периметр многоугольника, вылечиться от простуды, убрать свою комнату и так далее). Описание правил можно начинать со слов: «Я знаю, как ...» или «Для того чтобы ...».

Получение знаний, или познание реальной действительности — это ее отражение в сознании человека. Существует две формы познания: чувственное (с помощью органов чувств) и логическое, которое называют еще абстрактным мышлением.



### **Самое главное**

Информация для человека — это знания, которые он получает из различных источников.

Сообщение, полученное человеком, может пополнить его знания, если содержащиеся в нем сведения являются для человека понятными и новыми.

Все знания можно разделить на две части: факты и правила.

К фактам относятся знания об определенных явлениях, событиях, свойствах объектов и зависимостях между объектами.

К правилам относятся знания о последовательностях действий, направленных на достижение некоторой цели.

Существует две формы получения знаний о реальной действительности: чувственное и логическое познание.



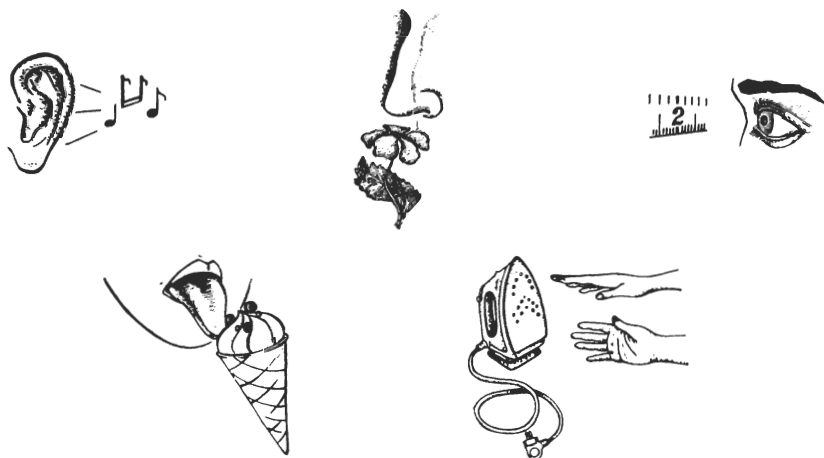
### **Вопросы и задания**

1. Чем является информация для человека?
2. Каким должно быть сообщение, чтобы онополнило ваши знания?

3. Информативны ли для вас следующие сообщения: «Сколько в слове гласных звуков, столько в слове и слогов» и «Термин «мультимедиа» означает объединение текста, звука, графики и видео в одном документе»? Ответ обоснуйте.
4. Приведите 2–3 примера фактов из области истории, географии или биологии.
5. Приведите 2–3 примера правил, изученных на уроках русского языка или математики.

## § 2.2. Чувственное познание окружающего мира

С первых минут жизни человек воспринимает реальную действительность, или окружающий мир (тепло, холод, звуки, свет и другие его проявления). Все окружающие нас объекты воздействуют на наши органы чувств. Органы чувств человека — средства приема сигналов из внешнего мира для передачи их в мозг.



Отдельные свойства объектов окружающего мира вызывают у нас ощущения. Например, мы ощущаем запах цветка, вкус земляники, тепло или холод.

Каждый объект имеет не одно, а множество свойств. Поэтому объект мы узнаем не по одному ощущению (сладкий или гладкий), а по их совокупности. Примеры:

- желтый, овальный, кислый, сочный, ароматный — это лимон;
- зеленая, колючая, пахнет смолой — елка.

**Восприятие** — это целостное отражение объекта, непосредственно воздействующего на наши органы чувств. Оно складывается из отдельных ощущений. Предположим, вас угостили яблоком. С помощью глаз вы воспринимаете его форму и цвет; с помощью языка определяете, кислое оно или сладкое; с помощью носа ощущаете его аромат. В результате создается целостное отражение этого объекта. Но восприятие одного и того же объекта разными людьми может быть различным. Так, скорее всего, взглянув на зеленый луг, художник восхитится его красотой, формой и красками, биолог увидит различные виды растений, а крестьянин прикинет, сколько с него можно получить сена. Из этого примера видно, что восприятие каждого человека очень сильно переплетается с его прежним опытом и знаниями.

Услышав слово «снег», каждый из нас может мысленно его представить, описать его свойства. **Представление** — это чувственный образ объекта, в данный момент нами не воспринимаемого, но воспринятого ранее.

С помощью органов чувств мы познаем отдельные предметы и явления окружающего мира. Такое познание называют чувственным познанием. Возможности чувственного познания ограничены.

Законы мира, сущность предметов и явлений, общее в них мы познаем с помощью мышления.



### **Самое главное**

Все окружающие нас объекты воздействуют на наши органы чувств и вызывают ощущения, восприятия и представления. С помощью органов чувств мы познаем отдельные предметы и явления окружающего мира.

Законы мира, сущность предметов и явлений, общее в них мы познаем с помощью мышления.



## **?** *Вопросы и задания*

1. Что такое реальная действительность?
2. Какими средствами приема сигналов из внешнего мира обладает человек?
3. Какие ощущения вы испытывали, купаясь летом в реке?
4. Из чего складываются восприятия? Приведите пример.
5. Каково ваше представление о носителях информации? Опишите их.

## § 2.3. Понятие как форма мышления

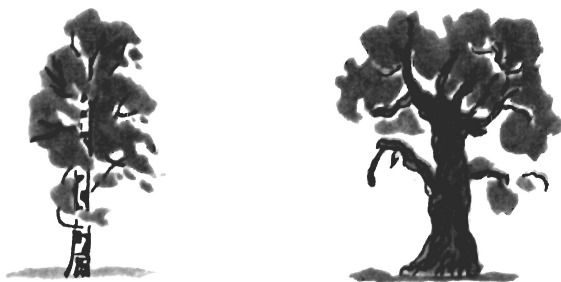
Познать законы окружающего мира, построить планы на будущее, сделать прогноз погоды или научное открытие нам помогает мышление. Во все времена умение правильно мыслить играло важную роль в жизни каждого человека. В глубокой древности возникла наука **логика**, изучающая законы и формы мышления, способы рассуждений и доказательств.

Нас окружает множество **объектов**. Это вещи, процессы, явления, люди, животные и так далее. Каждый из объектов можно описать с помощью признаков. Признаками являются свойства (зеленый, круглый, сочный, горький) и отношения (больше, меньше, длиннее, легче, громче).

Признаки объекта могут быть существенными и несущественными. Чтобы понять различие между ними, рассмотрим следующий пример.

В лесу около озера растет дерево береза (название объекта). Береза имеет возраст (10 лет), высоту (5 метров), место расположения (у озера), цвет и форму ствола, веток, листьев. Все это признаки объекта «дерево береза».

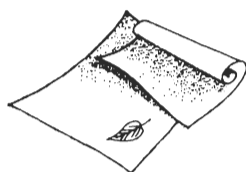
Предположим, что наша задача — найти любую березу среди других деревьев. В этой ситуации высота, возраст и место расположения березы не будут теми признаками, по которым мы отличаем березу, например, от дуба. Здесь существенными признаками будут цвет и форма ствола, цвет и форма листьев.



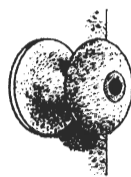
Совокупность существенных признаков отдельного объекта или некоторого множества однородных объектов отражается в **понятии**. Например, понятие «компьютер» отражает в себе существенные признаки всех существующих компьютеров как универсальных электронных устройств для хранения, обработки и передачи данных.

Понятие в языке выражается одним или несколькими словами, например: «мальчик», «мороз», «человек», «информация», «компьютер», «персональный компьютер», «звуковая информация», «двоичное кодирование», «учебник информатики», «рабочая тетрадь».

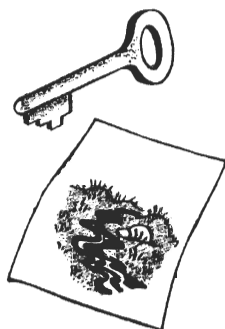
Существуют слова-омонимы, имеющие различные значения, выражающие разные понятия, но одинаково звучащие. Так слово «лист» имеет следующие значения: лист бумаги, лист растения.



ЛИСТ



ручка



КЛЮЧ

Слова-синонимы выражают одно и то же понятие (имеют одинаковые значения), но звучат различно. Так, слова «масса», «уйма», «бездна», «тьма», «пропасть» являются синонимами для понятия «множество» (в смысле «много»).

Словами мы пользуемся, когда говорим. Понятиями мы пользуемся, когда думаем, поэтому говорят, что понятие — это форма мышления.

## **ⓘ** *Самое главное*

Все окружающие нас объекты обладают некоторыми признаками. В понятии отражается совокупность существенных признаков отдельного объекта или класса однородных объектов.

В языке понятие выражается одним или несколькими словами.

Словами мы пользуемся, когда говорим. Понятиями мы пользуемся, когда думаем. Понятие — это форма мышления.

## **?** *Вопросы и задания*

1. С помощью чего можно описать любой объект реальной действительности?
2. Какие признаки объекта считаются существенными? Приведите пример.
3. Что такое понятие? Приведите примеры понятий, выраженных: а) одним существительным; б) словосочетанием.
4. Приведите 2–3 примера слов-омонимов, выражающих различные понятия.
5. Приведите 2–3 примера слов-синонимов, выражающих одно понятие.
6. Можно ли выражать понятие существительным во множественном числе?
7. Чем отличается реально существующий объект от понятия?
8. Приведите примеры понятий из повседневной жизни, а также из курсов математики, русского языка, истории, географии и информатики.

## **Как образуются понятия**

Почти все окружающие нас объекты состоят из отдельных частей.

**Анализ** — мысленное разделение объекта на составные части или выделение признаков объекта.

**Синтез** — мысленное соединение в единое целое частей объекта или его признаков, полученных в процессе анализа.

Так, ребенок, «разбирающий» игрушку, проводит своеобразный анализ (ему интересно, как устроена игрушка); ребенок, собирающий игрушку из ее частей, проводит своеобразный синтез.



«Все познается в сравнении», — гласит народная мудрость. **Сравнение** — мысленное установление сходства или различия объектов по существенным или несущественным признакам.

**Абстрагирование** — мысленное выделение одних признаков объекта и отвлечение от других. Чаще всего мы выделяем существенные признаки и отвлекаемся от несущественных.

**Обобщение** — мысленное объединение однородных объектов. Совокупность объектов, выделенных по какому-либо признаку, называют классом. Примеры: класс носителей информации, класс устройств ввода информации, класс растений и пр.

Анализ, синтез, сравнение, абстрагирование и обобщение являются основными логическими приемами формирования понятий.

Для выделения отдельных признаков требуется произвести анализ, то есть мысленно расчлнить це-

мый предмет на его составные части, отдельные признаки, а затем осуществить обратную операцию — синтез (мысленное объединение) частей предмета, отдельных его признаков в единое целое. Для выделения существенных признаков требуется отвлечься (абстрагироваться) от несущественных признаков, которых в любом предмете очень много. Этому способствует сравнение или сопоставление предметов.

Понятие формируется на основе обобщения существенных признаков (свойств, отношений), присущих классу однородных объектов.

### **Ⓢ** *Самое главное*

Анализ, синтез, сравнение, абстрагирование и обобщение являются основными логическими приемами формирования понятий.

Анализ — мысленное разделение объекта на составные части или выделение признаков объекта.

Синтез — мысленное соединение в единое целое частей объекта или его признаков, полученных в процессе анализа.

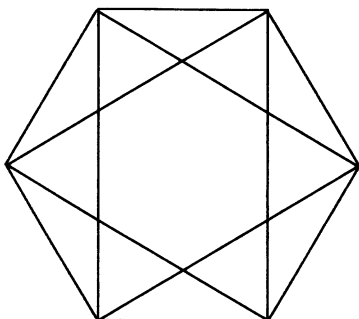
Сравнение — мысленное установление сходства или различия объектов по существенным или несущественным признакам.

Абстрагирование — мысленное выделение одних признаков объекта и отвлечение от других.

Обобщение — мысленное объединение однородных объектов в некоторый класс.

### **?** *Вопросы и задания*

1. Вспомните, в каких ситуациях вы сталкивались с анализом на уроках русского языка, математики.
2. Какое из понятий «последовательность» и «множество» аналогично понятию «класс»?
3. Сосчитайте, сколько треугольников в фигуре, изображенной на рисунке.



4. Внимательно проанализируйте условия следующих задач и дайте ответы.
- На столе стояло 3 стакана с вишней. Оксана съела один стакан вишни. Сколько стаканов осталось?
  - Зажгли 7 свечей, 2 из них погасли. Сколько свечей осталось?
  - Чем кончается день? Ночь? Чем кончается лето и начинается осень?
5. Отгадайте следующие загадки, мысленно представив объект как единое целое по его отдельным признакам.
- Без языка живет, не ест и не пьет, а говорит и поет.
  - На что ни взглянет этот глаз — все на картинке передаст.
  - Не куст, а с листочками, не рубашка, а сшита, не человек, а рассказывает.
  - Моря есть — плавать нельзя, дороги есть — ехать нельзя, земля есть — пахать нельзя. Что это?
  - Выходили двенадцать молодцев, выносили пятьдесят два сокола, выпускали триста шестьдесят пять лебедей. Что это?

6. Катя, Соня, Галя и Тамара родились 2 марта, 17 мая, 2 июля и 20 марта. Соня и Галя родились в одном месяце, а дни рождения Гали и Кати обозначаются одинаковыми числами. Назовите дату рождения каждой девочки.
7. Сравните между собой приведенные последовательности и найдите среди них такие, которые образованы при помощи одного и того же общего для них свойства. Что это за свойство?
- а) 2, 4, 6, 8, 10, ... ;  
 б) 2, 4, 8, 16, 32, ... ;  
 в) 1, 2, 3, 4, 5, ... ;  
 г) 2, 5, 8, 11, 14, ... .
8. Найдите закономерности и продолжите последовательности.
- а) а, б, в, г, ... ;  
 б) а, в, д, ё, ... ;  
 в) Анна, Борис, Валентина, Геннадий, ... ;  
 г) о, д, т, ч, п, ... ;  
 д) Москва, Абакан, Норильск, Киев, ... .
9. Ученик собирался на вечер, когда погас свет в комнате, где в ящике шкафа лежали его коричневые и синие носки. Какое наименьшее число носков он должен взять из ящика, чтобы обеспечить себя парой одного цвета?
10. Какие числа закодированы кругом, треугольником, квадратом и прямоугольником, если верны приведенные ниже равенства?

$$\square : 9 = \triangle$$

$$\square + \triangle = 8$$

$$3 \cdot \square = 162$$

$$9 - \bigcirc = \square$$



## Содержание и объем понятия

Если объект реальной действительности характеризуется формой, размером, цветом и другими признаками, то понятие характеризуется содержанием.

Содержание понятия — это все существенные признаки объекта или класса объектов, отраженные в понятии.

Например, содержание понятия «квадрат» составляют два существенных признака: «быть прямоугольником» и «иметь равные стороны». «Круглый», «упругий», «прыгучий», «используется в игре» — содержание понятия «мяч».

Объем понятия — множество объектов, каждому из которых присущи признаки, составляющие содержание понятия. Множество может, в частности, состоять из трех, двух, одного объекта или быть пустым (например, множество отличников в классе).

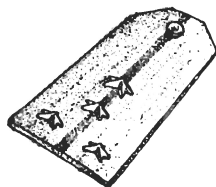
Множество может быть конечным (например, множество цифр или множество букв русского алфавита). Множество может быть бесконечным (например, множество натуральных чисел).

Рассмотрим одно из важнейших понятий информатики — «источник информации». Источник информации — это такой объект реальной действительности, который дает возможность получить информацию о чем-либо. В объем понятия «источник информации» входят: все книги, все газеты и журналы, все передачи и доклады, все люди, которые что-то знают и могут рассказать, и многое другое.

Рассмотрим объем понятия «знак». Знаком может быть любой чувственно воспринимаемый объект (предмет, явление или действие), выступающий как представитель (заменитель) предмета, явления, действия, свойства или отношения. Бывают знаки дорожные, знаки математические, знаки астрономические, знаки препинания, знаки химические, знаки товарные, знаки различия военнослужащих и другие.



**БИНОМ**



Выделяют единичные и общие понятия. За единичным понятием скрывается конкретный объект, тот самый, о котором идет речь. Примеры единичных понятий: «мой письменный стол», «дедушкины очки», «наша школьная доска», «папин компьютер».

За общим понятием скрывается множество однородных объектов, которое иначе называют классом объектов. Например, понятие «компьютер» включает в себя все компьютеры: старые и новые, хорошие и плохие, домашние и школьные и так далее.

К общим понятиям информатики относятся такие понятия, как «объект», «источник информации», «приемник информации», «канал связи», «знак», «исполнитель», «команда» и пр.



### **Самое главное**

Содержание понятия — это все существенные признаки объекта или класса объектов, отраженные в понятии.

Объем понятия — множество объектов, каждому из которых присущи признаки, составляющие содержание понятия.

Выделяют единичные и общие понятия. За единичным понятием скрывается конкретный объект, тот самый, о котором идет речь.

За общим понятием скрывается множество однородных объектов, которое иначе называют классом объектов.

## **?** *Вопросы и задания*

1. Чем обычно характеризуются объекты реальной действительности? Приведите пример.
2. Чем характеризуются понятия?
3. Что такое содержание понятия?
4. Раскройте содержание (перечислите основные признаки) понятий «клавиатура», «редактирование текста», «операционная система».
5. Что такое объем понятия?
6. Приведите примеры понятий, объемы которых являются:
  - а) пустым множеством;
  - б) множеством, состоящим из 1 элемента;
  - в) множеством из 10 элементов;
  - г) бесконечным множеством.
7. В чем различие между единичными и общими понятиями? Приведите пример.
8. Коля рассказывает Пете: «Я вчера смотрел кинофильм. Кинофильм мне понравился». Объясните, общим или единичным в данной ситуации является понятие «кинофильм».

### **Отношения между понятиями**

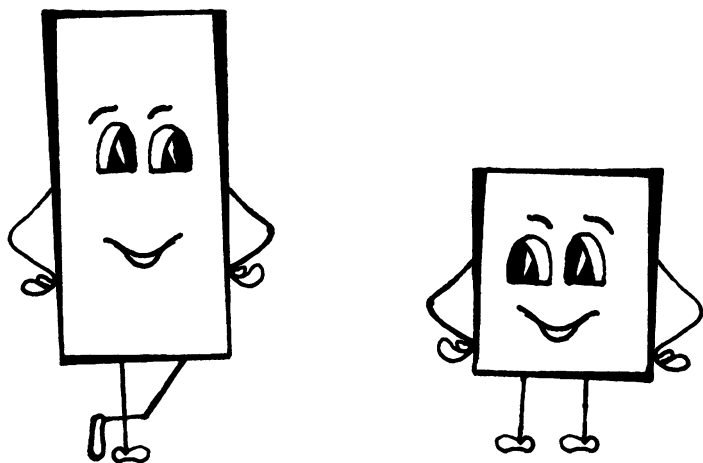
При сравнении реальных объектов мы сравниваем их размеры, цвет, форму и прочее. Отношения между реально существующими объектами описываются словами: больше — меньше; длиннее — короче; ближе — дальше; выше — ниже; брат — сестра и так далее.

Понятия тоже можно сравнивать между собой. Но, в отличие от объектов реальной действительности, понятия не имеют ни цвета, ни запаха, ни размера. Понятия — это наши представления, наши мысли об объектах. При сравнении понятий сравнивают их содержания и их объемы.

Рассмотрим два понятия — «квадрат» и «прямоугольник».

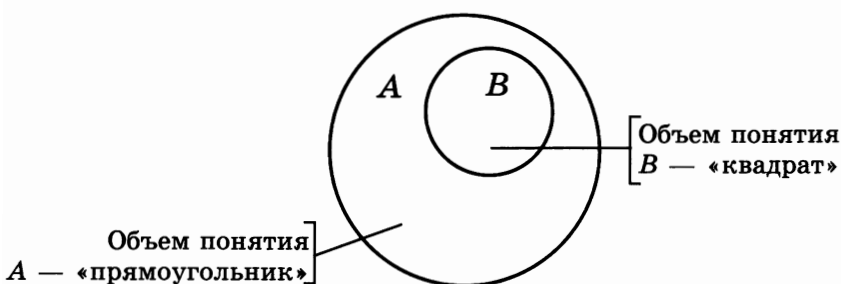
Понятие	Содержание
Квадрат	<ul style="list-style-type: none"> <li>• Четырехугольник;</li> <li>• все углы прямые;</li> <li>• все стороны равны</li> </ul>
Прямоугольник	<ul style="list-style-type: none"> <li>• Четырехугольник;</li> <li>• все углы прямые;</li> <li>• длины противоположных сторон попарно равны</li> </ul>

Как видно из таблицы, содержания понятий отличаются одним признаком: у квадрата *длины всех сторон равны*, а у прямоугольника *длины противоположных сторон попарно равны*.



Объем понятия «прямоугольник» *больше* объема понятия «квадрат», так как все квадраты — тоже прямоугольники.

Отношения между понятиями удобно представлять кругами.



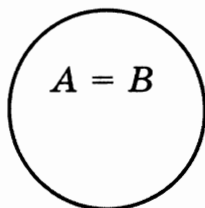
Далекие друг от друга по своему содержанию понятия, не имеющие общих признаков, называются несравнимыми (например, «компьютер» и «молоко», «карандаш» и «ледник»); остальные понятия называются сравнимыми.

Рассмотрим некоторые отношения между сравнимыми понятиями.

### Отношение «тождество»

Если объемы понятий совпадают, другими словами, объем одного понятия равен объему другого, то отношение между этими понятиями называют тождеством.

Например, понятия «равносторонний прямоугольник» (A) и «квадрат» (B) — тождественные понятия. Графическое представление отношения между ними будет выглядеть так:

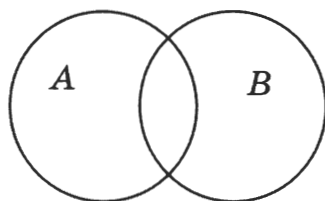


Если совпадают объемы единичных понятий (например, «столица России» ( $A$ ) и «город Москва» ( $B$ ), то такие понятия называют равными.



### Отношение «пересечение»

Пересечением называют отношение между понятиями, объемы которых совпадают частично, то есть содержат общие элементы. Пусть понятие  $A$  — «электронное письмо»,  $B$  — «письмо на русском языке». Тогда:

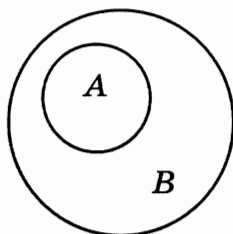


В пересечение двух кругов попадают все электронные письма на русском языке.



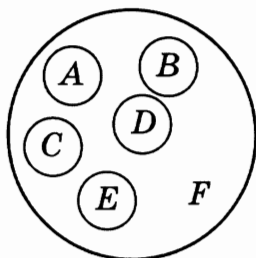
### Отношение «подчинение»

Подчинением называют отношение между понятиями, когда объем одного из них полностью входит в объем другого понятия, но не исчерпывает его. Пусть понятие *A* — «клавиатура», понятие *B* — «устройство ввода». Тогда:



### Отношение «соподчинение»

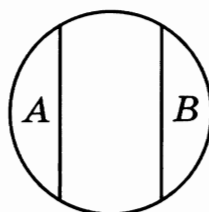
Соподчинением называется отношение между несколькими понятиями, объемы которых не пересекаются, но которые принадлежат некоторому более общему (родовому) понятию. Это виды одного и того же рода. Пусть понятие *A* — «береста», *B* — «папирус», *C* — «глиняная дощечка», *D* — «бумага», *E* — «магнитный диск», *F* — «носитель информации». Здесь *A*, *B*, *C*, *D* и *E* соподчинены *F*:



### Отношение «противоположность»

Слова, выражающие противоположные понятия, называются антонимами. Пусть понятие *A* — «ком-

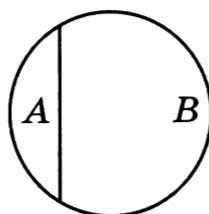
пьютер с маленькой памятью», понятие  $B$  — «компьютер с большой памятью». Тогда:



Объемы этих двух понятий разделены объемом некоторого третьего понятия, например, понятия «компьютер со средней памятью».

### Отношение «противоречие»

Пусть понятие  $A$  — «новый компьютер». Тогда другое понятие, находящееся с ним в отношении противоречия, следует обозначить  $не-A$  (т. е. «неновый компьютер»). Круг, выражающий это отношение, делится на две части: третьего понятия между ними нет:



### **i** Самое главное

Реальные объекты сравнивают по размеру, цвету, форме и так далее.

При сравнении понятий сравнивают их содержания и их объемы.



Далекие друг от друга по своему содержанию понятия, не имеющие общих признаков, называются несравнимыми.

Сравнимые понятия могут находиться в отношениях тождества, пересечения, подчинения, соподчинения, противоположности и противоречия.



## **?** *Вопросы и задания*

1. На основании каких признаков мы сравниваем реальные объекты?
2. Какими словами выражаются отношения между объектами реальной действительности?
3. С помощью чего удобно представлять отношения между понятиями?
4. Какие понятия считаются несравнимыми? Приведите пример.
5. Приведите примеры понятий:
  - а) тождественных;
  - б) пересекающихся;
  - в) подчиняющих и подчиненных;
  - г) соподчиненных.
6. Приведите примеры понятий:
  - а) противоположных;
  - б) противоречащих.Используйте рисунок.

7. В одном множестве 40 элементов, а в другом — 30. Какое максимальное количество элементов может быть в их:
- пересечении;
  - объединении?
8. В детском саду 52 ребенка. Каждый из них любит пирожное или мороженое. Половина детей любит пирожное, а 20 человек — пирожное и мороженое. Сколько детей любит мороженое?
9. Внимательно рассмотрите примеры отношений:

Отношение	Пример
Целое — часть	Окунь — плавник
Вид — род	Ландыш — цветок
Вид — вид	Ромашка — василек
Последовательность	Понедельник — вторник
Причина — следствие	Жара — жажда

Для каждого отношения придумайте 2–3 собственных примера.

### Определение понятия

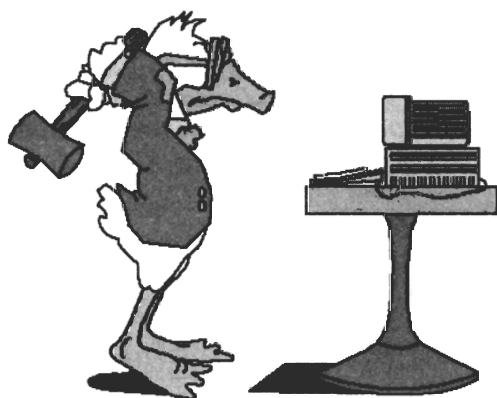
Определение понятия — это перечисление всех существенных признаков объекта (класса однородных объектов) в связном предложении. Каждый из признаков, входящих в определение, должен быть необходим, а все вместе — достаточны для установления данного понятия. Это означает, что в определении должно раскрываться основное содержание понятия, в нем не должно быть лишних слов, но не должно быть и недосказанностей.

Некоторые первоначальные понятия не определяются. Такие понятия есть в каждой науке. Так, в математике это понятия «точка» и «множество», в информатике — «информация».

Очень часто определение строится через ближайший род и видовое отличие. Объем видового понятия более узок и полностью входит в объем родового понятия:

$$\underbrace{\text{видовое понятие}}_{\text{определяемое понятие}} = \underbrace{\text{родовое понятие} + \text{видовое отличие}}_{\text{определяющее понятие}}$$

Например, в определении «Пользователь — это человек, применяющий компьютер для получения информации или решения задачи», понятие «пользователь» — видовое, понятие «человек» — родовое, «применяющий компьютер для получения информации или решения задачи» — видовое отличие.



При составлении определений следует избегать логических ошибок. Рассмотрим такое предложение: «Компьютер — это помощник человека при работе с информацией». Можно ли считать это предложение определением? В определении объемы определяемого и определяющего понятий равны. Если считать объем определяемого понятия «компьютер» равным объему определяющего понятия «помощник человека при работе с информацией», то компьютером можно считать любого человека, помогающего вам работать с инфор-

мацией. Значит, приведенное предложение нельзя считать определением. Верным будет следующее определение: «Компьютер — универсальное электронное устройство, предназначенное для хранения, обработки и передачи данных».

### **ⓘ Самое главное**

Определение понятия — это перечисление всех существенных признаков объекта (класса однородных объектов) в связном предложении. Каждый из признаков, входящих в определение, должен быть необходим, а все вместе — достаточны для установления данного понятия.

Распространенным является определение понятия через ближайший род и видовое отличие:

$$\underbrace{\text{видовое понятие}}_{\text{определяемое понятие}} = \underbrace{\text{родовое понятие} + \text{видовое отличие}}_{\text{определяющее понятие}}$$

### **? Вопросы и задания**

1. Как определить понятие?
2. В. В. Маяковский писал:  
«Если тебе «корова» имя,  
У тебя должно быть молоко и вымя».  
Скажите, достаточно ли признаков, указанных поэтом, для определения понятия «корова»?
3. Каким способом чаще всего определяются понятия?
4. Найдите в терминологическом словаре определения двух-трех понятий через ближайший род и видовое отличие.
5. Почему фразу «Книга — источник знаний» нельзя считать определением понятия «книга»? Как можно определить данное понятие?

## Классификация

Каждое общее понятие может быть представлено как совокупность нескольких понятий.

Так, понятие «живая природа» включает в себя понятия «человек», «животное», «растение». В свою очередь, понятие «животное» включает в себя понятия «птица», «зверь», «рыба», «насекомое» и так далее.

Распределение объема некоторого понятия по избранному основанию на ряд частей называется классификацией.

Очень важен выбор основания классификации. Классификация может производиться по существенным признакам (естественная) или по несущественным признакам (вспомогательная).

Ни один школьный предмет не обходится без классификаций. Так, на уроках русского языка вы сталкиваетесь с естественной классификацией частей речи. Графически ее можно представить так:



О классификации систем счисления вы можете прочитать в главе 4 «Материал для любознательных».

Место, занимаемое объектом в естественной классификации, позволяет судить о его свойствах. Вспомогательная классификация не позволяет судить о свойст-

вах объектов; она служит для более легкого отыскания объектов. Примеры вспомогательных классификаций: список фамилий, расположенных по алфавиту; каталог книг и так далее.

### **ⓘ** *Самое главное*

Классификация — это распределение объема некоторого понятия по избранному основанию на ряд частей.

Классификация может производиться по существенным признакам (естественная) или по несущественным признакам (вспомогательная).

Место, занимаемое объектом в естественной классификации, позволяет судить о его свойствах. Вспомогательная классификация служит для более легкого отыскания объектов.

### **?** *Вопросы и задания*

1. Как вы считаете, для чего нужна классификация?
2. Чем различаются естественная и вспомогательная классификации?
3. Приведите примеры классификаций, с которыми вы познакомились на уроках русского языка, математики, биологии и географии.

## § 2.4. Суждение как форма мышления

В мышлении понятия не выступают разрозненно, они определенным образом связываются между собой. Формой связи понятий друг с другом является суждение.

**Суждение** — это форма мышления, в которой что-либо утверждается или отрицается об объектах, признаках или отношениях объектов.

Суждение имеет свою языковую форму — повествовательное предложение.

Примеры суждений:

- Москва больше Санкт-Петербурга.
- Все мальчики любят играть в футбол.
- Некоторые ученики нашего класса поют в школьном хоре.
- Маша Радугина — лучшая ученица 6 класса.

Суждения бывают простыми и сложными.

Выделяют четыре вида простых суждений.

### 1. Общеутвердительные суждения.

Примеры: «Все квадраты — прямоугольники»; «Все дороги ведут в Рим».

### 2. Общеотрицательные суждения.

Примеры: «Ни один арифмометр не является компьютером»; «Ни один треугольник не является квадратом».

### 3. Частноутвердительные суждения.

Примеры: «Некоторые школьники — спортсмены»; «Некоторые мониторы — жидкокристаллические».

### 4. Частноотрицательные суждения.

Примеры: «Некоторые компьютеры не являются современными»; «Некоторые компьютерные программы не являются источниками информации».

Сложные суждения образуются из простых с помощью связок, называемых логическими: «и», «или», «неверно, что ...», «если ..., то ...». Такие конструкции для нас привычны и часто встречаются в обыденной речи.

Примеры: «Каждый четырехугольник имеет четыре угла и четыре стороны»; «Каждый компьютер комплектуется акустическими колонками или наушниками»; «Неверно, что джойстик является устройством вывода информации»; «Если ласточки летают низко, то скоро будет дождь».

Часто применяются и такие слова-связки, как: «для того чтобы ..., необходимо ...», «для того чтобы ..., достаточно ...», «для того чтобы ..., необходимо и достаточно ...» и пр.

Если некоторое событие не может иметь место без определенного условия, то это условие является **необходимым**.

Пример: «Для того чтобы выводить созданные на компьютере документы на бумагу, необходим принтер».

Если некоторое событие обязательно имеет место при определенном условии, то это условие является **достаточным**.

Пример: «Для того чтобы число делилось на 5, достаточно, чтобы оно оканчивалось нулем».

Если без определенного условия не может быть некоторого события, а из наличия этого условия следует данное событие, то условие является **необходимым и достаточным**.

Пример: «Для того чтобы число делилось на 10, необходимо и достаточно, чтобы оно оканчивалось нулем».



### **Самое главное**

Суждение — это форма мышления, в которой что-либо утверждается или отрицается об объектах, свойствах или отношениях объектов.

Суждения бывают простыми и сложными.

Выделяют четыре вида простых суждений: общеутвердительные, общеотрицательные, частноутвердительные, частноотрицательные.

Сложные суждения образуются из простых с помощью логических связок.



## ? Вопросы и задания

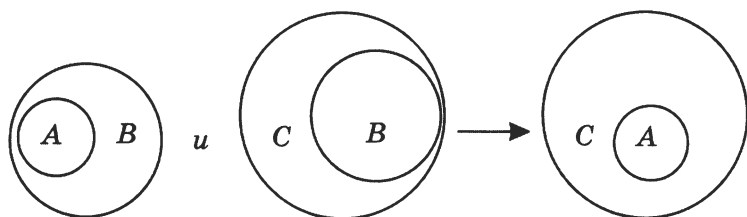
1. Употребляя слова «все», «некоторый», «каждый», «ни один», установите отношения между следующими понятиями:
  - а) «прямоугольный треугольник» и «равнобедренный треугольник»;
  - б) «равносторонний треугольник» и «равнобедренный треугольник»;
  - в) «прямоугольник» и «ромб»;
  - г) «прямоугольник» и «квадрат»;
  - д) «квадрат» и «прямоугольник»;
  - е) «ромб» и «квадрат»;
  - ж) «прямоугольник» и «окружность»;
  - з) «квадрат» и «ромб».
2. Приведите по 2–3 примера простых общеутвердительных, общеотрицательных, частноутвердительных и частноотрицательных суждений.
3. Приведите по одному примеру сложных высказываний, образованных с помощью связок «и», «или», «неверно, что» и «если ..., то».
4. Дополните следующие суждения подходящими по смыслу логическими связками «необходимо», «достаточно», «необходимо и достаточно».
  - а) Чтобы сварить щи, ... иметь воду.
  - б) Чтобы земля на грядках была мокрой, ... чтобы прошел дождь.
  - в) Для того чтобы число делилось на 4, ... чтобы оно было четным.
  - г) Чтобы число делилось на 3, ... чтобы оно делилось на 9.
  - д) Чтобы купить в магазине книгу, ... иметь деньги.
5. Предложите свою классификацию для суждений.

## § 2.5. Умозаключение как форма мышления

**Умозаключение** — форма мышления, посредством которой из одного или нескольких суждений, называемых посылками, мы по определенным правилам вывода получаем суждение-заключение.

Существует очень много правил построения умозаключений (рассуждений). Рассмотрим два из них.

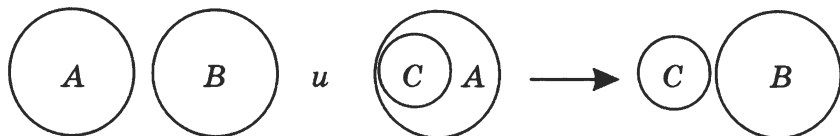
1. Если все  $A$  являются  $B$ , а все  $B$  являются  $C$ , то все  $A$  являются  $C$ .



Пример: «Если все воробьи — птицы, а все птицы — животные, то все воробьи являются животными».



2. Если ни одно  $A$  не является  $B$ , а все  $C$  являются  $A$ , то ни одно  $C$  не является  $B$ .



Пример: «Если ни у одного четного числа десятичная запись не заканчивается цифрой 5 и все числа, делящиеся на 4, четны, то ни у одного числа, делящегося на 4, десятичная запись не заканчивается цифрой 5».

## **ⓘ** Самое главное

Умозаключение — форма мышления, посредством которой из одного или нескольких суждений, называемых посылками, мы по определенным правилам вывода получаем суждение-заключение.

## **?** Вопросы и задания

### 1. Постройте правильные заключения.

- 1) Все имена существительные изменяются по падежам и числам. Слово «урок» — имя существительное. Следовательно, ... .
- 2) Все ученики, которые учатся в первую смену, приходят в школу в 8 часов. Сережа учится в первую смену. Когда приходит в школу Сережа?
- 3) Названия городов — имена собственные. Имена собственные пишутся с большой буквы. Значит, ... .
- 4) Если числитель дроби меньше знаменателя, то дробь правильная. У дроби  $\frac{3}{8}$  числитель меньше знаменателя. Следовательно, ... .
- 5) Все растения на свету поглощают углекислый газ. Тополь — растение. Следовательно, ... .
- 6) Если число оканчивается цифрой 0, то оно делится на 5. Известно, что данное число не делится на 5. Следовательно, ... .
- 7) Все ученики 6 класса занимаются спортом. Саша учится в 6 классе. Значит, ... .
- 8) Если ни один слон не может летать, и все птицы летают, то ... .
- 9) Год состоит из 365 или 366 дней. 1995 год не состоит из 366 дней. Сколько дней в 1995 году?

### 2. Мама купила 4 шара красного и голубого цветов. Красных шаров было больше, чем голубых. Сколько мама купила красных шаров и сколько голубых?

3. У сестер Юли и Тони было три платка: один розовый и два голубых. Увидев на Юле один из этих платков, Тоня поняла, что она может надеть только голубой платок. Какой платок был на Юле?
4. В пакете лежат конфеты двух сортов. Какое наименьшее число конфет (не видя их) надо вытащить из пакета, чтобы среди них были хотя бы:
  - а) две конфеты одного сорта;
  - б) три конфеты одного сорта?
5. Работая в колхозном саду, школьники собрали 22 ящика фруктов; в одних были яблоки, в других — груши, в третьих — сливы. Можно ли утверждать, что имеется по крайней мере 8 ящиков с одним из указанных видов фруктов?
6. Пяти гномам показали 3 красных и 4 синих капюшона. Затем в темноте на них надели 3 красных и 2 синих капюшона, а остальные капюшоны спрятали. Кто из гномов может определить цвет надетого на него капюшона?
7. Турист идет к озеру. У развилки дорог сидят два брата, каждый из которых знает, какая дорога ведет к озеру. На вопросы братья отвечают только «да» и «нет». Один из них всегда говорит только правду. Другой всегда лжет. Все это известно туристу. Помогите туристу сформулировать такие вопросы, чтобы узнать, какая дорога ведет к озеру:
  - а) два вопроса одному из братьев;
  - б) один вопрос одному из братьев.

---

# Алгоритмы и исполнители

---

### § 3.1. Что такое алгоритм

Каждый человек в повседневной жизни, во время учебы или на работе решает огромное количество задач самой разной сложности. Некоторые из этих задач столь просты и привычны, что мы решаем их не задумываясь, автоматически, и даже не считаем задачами. К ним можно отнести такие задачи, как «Купить хлеб», «Собратся в школу», «Закрывать дверь на ключ» и пр. Другие же задачи, напротив, так трудны, что требуют длительных размышлений и усилий для поиска решения и достижения поставленной цели. Например, решения задач «Написать контрольную работу на 5» или «Свободно разговаривать на иностранном языке» требуют выполнения гораздо большего количества сложных действий, чем решение задачи «Купить мороженое». Но решение даже самой простой задачи обычно осуществляется за несколько последовательных шагов.

Например, процесс покупки хлеба можно представить так:

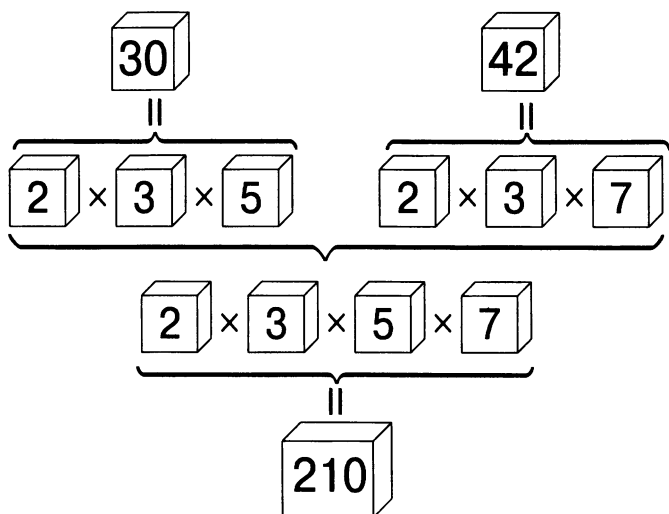
- 1) взять у мамы деньги;
- 2) пойти в магазин;
- 3) выбрать нужные хлебобулочные изделия;
- 4) оплатить стоимость покупки;
- 5) принести хлеб домой.

Аналогично, в виде последовательности действий можно описать процессы решения многих задач, с ко-

торыми вы имеете дело в школе: «Вычислить периметр многоугольника», «Найти наибольший общий делитель двух натуральных чисел», «Определить часть речи», «Провести фонетический разбор слова». Такая последовательность шагов в решении задачи называется алгоритмом. При этом для алгоритма важен не только набор действий, но и то, в каком порядке они выполняются. Например, попробуем переставить в известном вам алгоритме нахождения наименьшего общего кратного (НОК) нескольких натуральных чисел четвертое действие на второе место:

- 1) разложить исходные числа на простые множители;
- 2) найти произведения получившихся множителей;
- 3) выписать множители, входящие в разложение одного из чисел;
- 4) дописать к ним недостающие множители из разложений остальных чисел.

Эту последовательность действий также можно исполнить, но к достижению поставленной цели (нахождению НОК) она не приведет!



Выполним последовательность действий для нахождения значения следующего выражения:

$$(5,88 + 5,52) - 2,8 \cdot (5 \cdot 0,103 - 0,015).$$

- 1)  $5,88 + 5,52 = 11,4$ ;
- 2)  $5 \cdot 0,103 = 0,515$ ;
- 3)  $0,515 - 0,015 = 0,5$ ;
- 4)  $2,8 \cdot 0,5 = 1,4$ ;
- 5)  $11,4 - 1,4 = 10$ .

Можно ли изменить порядок действий в этом случае? С чем, по вашему мнению, это связано?

Алгоритм может представлять собой описание некоторой последовательности вычислений, а может — описание последовательности действий нематематического характера. Но, в любом случае, перед его составлением должны быть четко определены начальные условия и то, что предстоит получить.

**Алгоритм** — описание конечной последовательности шагов в решении задачи, приводящей от исходных данных к требуемому результату.

Первенство в разработке алгоритмов принадлежит человеку. Исполняют алгоритмы люди и всевозможные устройства — компьютеры, роботы, станки, спутники, сложная бытовая техника и даже некоторые детские игрушки.



В главе 4 «Материал для любознательных» вы можете узнать о происхождении слова «алгоритм».



### **Самое главное**

Алгоритм — описание конечной последовательности шагов в решении задачи, приводящей от исходных данных к требуемому результату.

Первенство в разработке алгоритмов принадлежит человеку. Исполняют алгоритмы люди и всевозможные технические устройства.

## **?** *Вопросы и задания*

1. Приведите два-три примера правил или предписаний, которым вы следуете в повседневной жизни.
2. Задача может быть успешно решена только тогда, когда она очень четко описана. Из приведенного списка задач выберите те, которые сформулированы четко. Свой выбор обоснуйте.
  - а) «Сосчитать звезды»;
  - б) «сосчитать число окон в своем доме»;
  - в) «купить книги»;
  - г) «одеться»;
  - д) «дать подробное описание дороги от двери своего дома до школы (пешком, на автобусе или другом транспорте)».
3. Можно ли считать хорошо поставленной задачу в одной из русских народных сказок: «Иди туда, не знаю куда. Принеси то, не знаю что»?
4. Туристы (отец, мать и два брата-близнеца) должны переправиться через реку. В их распоряжении есть маленькая лодка, вмещающая только одного взрослого или двоих детей. Как организовать переправу, если и взрослые, и дети умеют грести?
5. Что такое алгоритм?
6. Приведите два-три примера алгоритмов, изученных вами в школе.

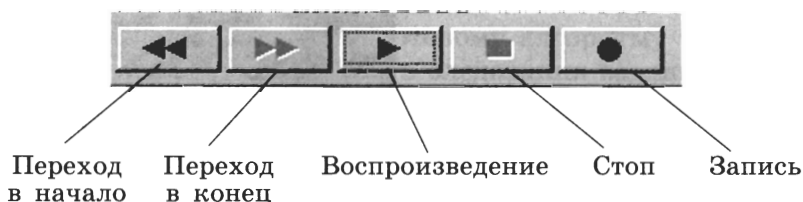


## § 3.2. Исполнители вокруг нас

Современного человека окружает множество разнообразных технических устройств: телевизор, магнитофон, фотоаппарат, телефон, стиральная машина, автомобиль и пр. Каждое из этих устройств предназначено для решения своей задачи и способно выполнять некоторый ограниченный набор действий, или команд.

Устройство, способное выполнять определенный набор команд, мы будем называть исполнителем. Команды, которые может выполнить конкретный исполнитель, образуют систему команд исполнителя (СКИ). Исполнители бывают разные. Одним из самых простых исполнителей можно считать кнопку включения/выключения электропитания на корпусе монитора.

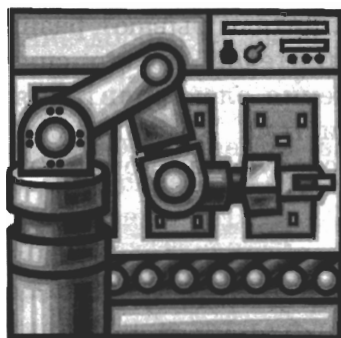
Система команд исполнителя-магнитофона содержит следующие команды:



Более сложным исполнителем является современная стиральная машина, в электронную память которой заложены разработанные инженерами различные программы стирки белья. Весь процесс стирки (замачивание, отстирывание, полоскание, отжим, сушка) машина выполняет автоматически, без участия человека, но по программе, выбранной человеком.

Среди автоматических устройств наиболее совершенными исполнителями являются роботы. Едва ли человек сможет так быстро, безошибочно и качественно собрать сложнейшую электронную плату, как это делает робот-манипулятор на автоматизированном производстве. В наше время созданы человекоподоб-

ные роботы и роботы-игрушки, напоминающие домашних животных.



Самый впечатляющий пример исполнителя — компьютер. Его отличительная черта — универсальность. Вы знакомы с компьютерными программами, предназначенными для обработки текстовой, числовой и графической информации, с обучающими программами и компьютерными играми. Кроме того, существуют программы, с помощью которых компьютер управляет работой других связанных с ним устройств (исполнителей).

Во многих случаях и сам человек является исполнителем алгоритмов. Например, каждый из нас при переходе улицы является исполнителем следующего алгоритма:

- 1) остановись на тротуаре;
- 2) посмотри налево;

- 3) если транспорта нет, то иди до середины улицы и остановись, иначе выполняй п. 2;
- 4) посмотри направо;
- 5) если транспорта нет, то иди до противоположного тротуара, иначе выполняй п. 4.

С большим количеством алгоритмов имеют дело школьники при выполнении письменных и устных заданий.

Итак, **исполнитель** — это человек, группа людей, животное или техническое устройство, способные выполнять заданные команды.

Любой исполнитель имеет ограниченную систему команд. Даже среди людей вряд ли найдется исполнитель, умеющий делать все. Если в качестве исполнителей рассматривать школьников, то шестиклассник способен выполнять значительно больше предписаний, чем ученик первого класса.

Разработка алгоритма — трудоемкая задача, требующая от человека глубоких знаний и больших затрат времени. Решение задачи по готовому алгоритму требует от исполнителя только строгого следования заданным предписаниям. Исполнитель не вникает в смысл того, что он делает, и не рассуждает, почему он поступает так, а не иначе, — он действует формально. С этим связана возможность автоматизации деятельности человека:

- процесс решения задачи представляется в виде последовательности простейших операций;
- создается машина (автоматическое устройство), способная выполнять эти операции в последовательности, заданной в алгоритме;
- человек освобождается от рутинной деятельности, выполнение алгоритма поручается автоматическому устройству.



### **Самое главное**

**Исполнитель** — человек, группа людей, животное или техническое устройство, способные выполнять заданные команды.

Команды, которые может выполнить конкретный исполнитель, образуют систему команд исполнителя (СКИ).

Исполнитель формально выполняет алгоритмы, составленные из команд, входящих в его СКИ.

Человек разрабатывает алгоритмы, управляет работой других исполнителей по выполнению алгоритмов, сам исполняет алгоритмы.

Компьютер управляет работой связанных с ним технических устройств по выполнению алгоритмов; сам исполняет алгоритмы (программы).

## **?** *Вопросы и задания*

1. Кого или что называют исполнителем алгоритма?
2. С какой задачей может справиться только человек?
3. Как вы понимаете высказывание Норберта Винера: «Любая машина стоит лишь столько, сколько стоит человек, который на ней работает»?
4. Опишите достоинства и недостатки исполнителя «Робот» и исполнителя «Человек», выполняющих перевод текста с иностранного языка на русский по следующему алгоритму:
  - 1) прочитать первое предложение на иностранном языке;
  - 2) поочередно перевести все слова предложения на русский язык;
  - 3) составить из полученных слов фразу на русском языке и записать ее;
  - 4) если не все предложения переведены, то взять первое непереведенное предложение и перейти к п. 2; иначе выполнить п. 5;
  - 5) закончить работу.



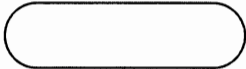
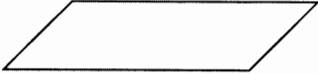
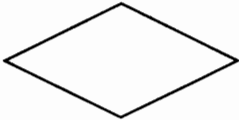

В главе 4 «Материал для любознательных» вы можете познакомиться с графическими исполнителями.

### § 3.3. Формы записи алгоритмов

Чтобы составить алгоритм, необходимо знать систему команд предполагаемого исполнителя, правила записи отдельных команд и всего алгоритма в целом.

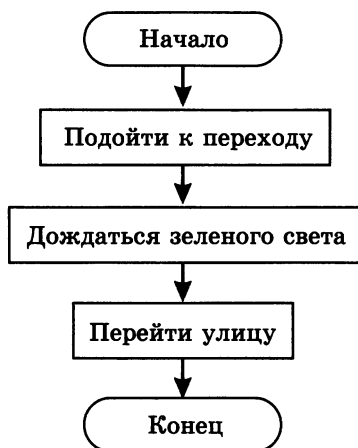
Будем рассматривать алгоритмы, исполнителем которых является человек.

Последовательность шагов, которые выполняются человеком при решении некоторой задачи, удобно записывать в виде нумерованного списка (словесная форма), таблицы или изображать с помощью блок-схемы. В последнем случае для обозначения шагов алгоритма используются следующие геометрические фигуры:

Название фигуры	Изображение	Обозначаемый шаг алгоритма
Овал		Начало или конец
Параллелограмм		Ввод или вывод
Ромб		Принятие решения
Прямоугольник		Выполнение действия

Последовательность действий указывается с помощью стрелок, соединяющих фигуры, обозначающие шаги алгоритма.

Вот так, например, с помощью блок-схемы можно представить алгоритм действий человека при переходе улицы:



### **Самое главное**

Алгоритмы, исполнителем которых является человек, удобно записывать в словесной форме, в табличной форме или изображать с помощью блок-схем.

Для обозначения шагов алгоритма в блок-схемах используются геометрические фигуры: овал (начало и конец), параллелограмм (ввод/вывод), ромб (принятие решения) и прямоугольник (выполнение действия). Стрелки, связывающие эти фигуры, задают порядок выполнения соответствующих шагов.



### **Вопросы и задания**

1. Что необходимо знать при разработке алгоритма?
2. Перечислите известные вам формы записи алгоритмов.

3. Злая мачеха отправила падчерицу к роднику за водой. «Вот тебе 2 ведра, в одно из них входит 9 литров воды, а в другое — 5 литров. Но ты должна принести домой ровно 3 литра воды», — сказала она бедной девушке. Как должна действовать падчерица, чтобы выполнить это поручение?

Будьте готовы представить этот алгоритм в словесной или табличной форме.

4. Какие геометрические фигуры используются в блок-схеме?
5. Каким образом задается порядок выполнения отдельных действий в блок-схеме?

## § 3.4. Типы алгоритмов

В алгоритмах команды записываются друг за другом в определенном порядке. Выполняются они не обязательно в записанной последовательности. В зависимости от порядка выполнения команд можно выделить три типа алгоритмов:

- линейные алгоритмы;
- алгоритмы с ветвлениями;
- алгоритмы с повторениями.

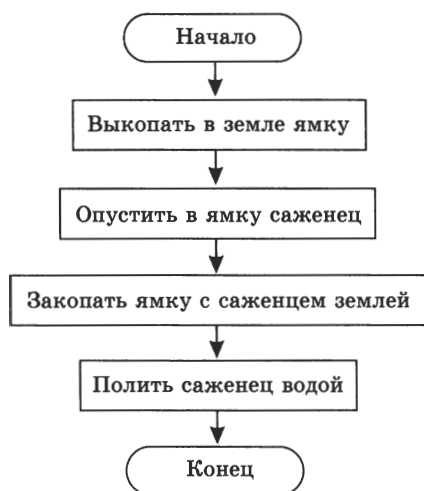
### Линейные алгоритмы

Алгоритм, в котором команды выполняются в порядке их записи, то есть последовательно друг за другом, называется **линейным**.

Например, линейным является следующий алгоритм посадки дерева:

- 1) выкопать в земле ямку;
- 2) опустить в ямку саженец;
- 3) засыпать ямку с саженцем землей;
- 4) полить саженец водой.

С помощью блок-схемы данный алгоритм можно изобразить так:





## Алгоритмы с ветвлениями

Ситуации, когда заранее известна последовательность требуемых действий, встречаются крайне редко. В жизни часто приходится принимать решение в зависимости от сложившейся обстановки. Если идет дождь, мы берем зонт и надеваем плащ; если жарко, надеваем легкую одежду. Встречаются и более сложные условия выбора. В некоторых случаях от выбранного решения зависит дальнейшая судьба человека.

Логику принятия решения можно описать так:

ЕСЛИ <условие> ТО <действия 1>  
ИНАЧЕ <действия 2>

Примеры:

- ЕСЛИ хочешь быть здоров, ТО закаляйся, ИНАЧЕ валяйся весь день на диване;
- ЕСЛИ низко ласточки летают, ТО будет дождь, ИНАЧЕ дождя не будет;
- ЕСЛИ уроки выучены, ТО иди гулять, ИНАЧЕ учи уроки.

В некоторых случаях <действия 2> могут отсутствовать:

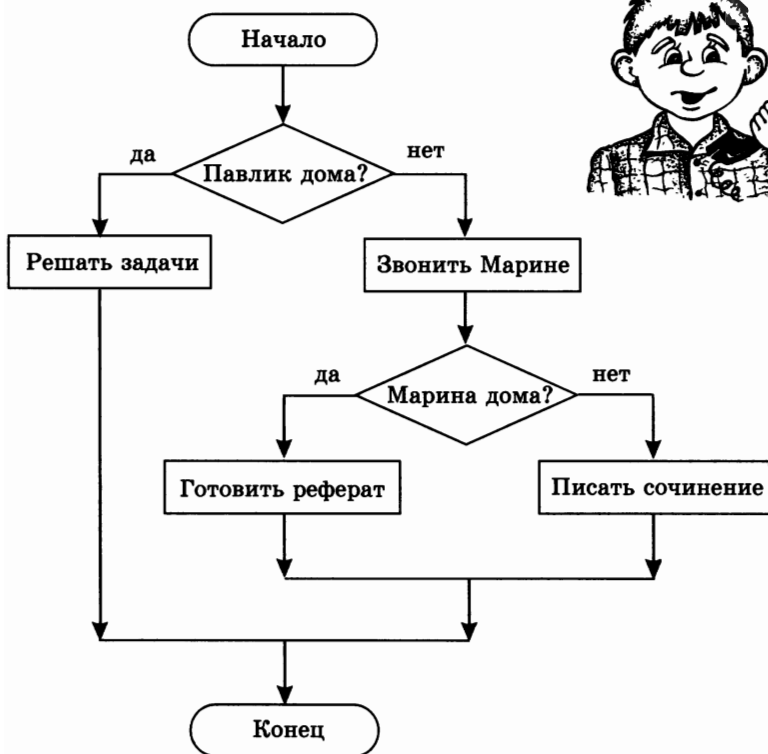
ЕСЛИ <условие> ТО <действия 1>

Пример:

- ЕСЛИ назвался груздем, ТО полезай в кузов.

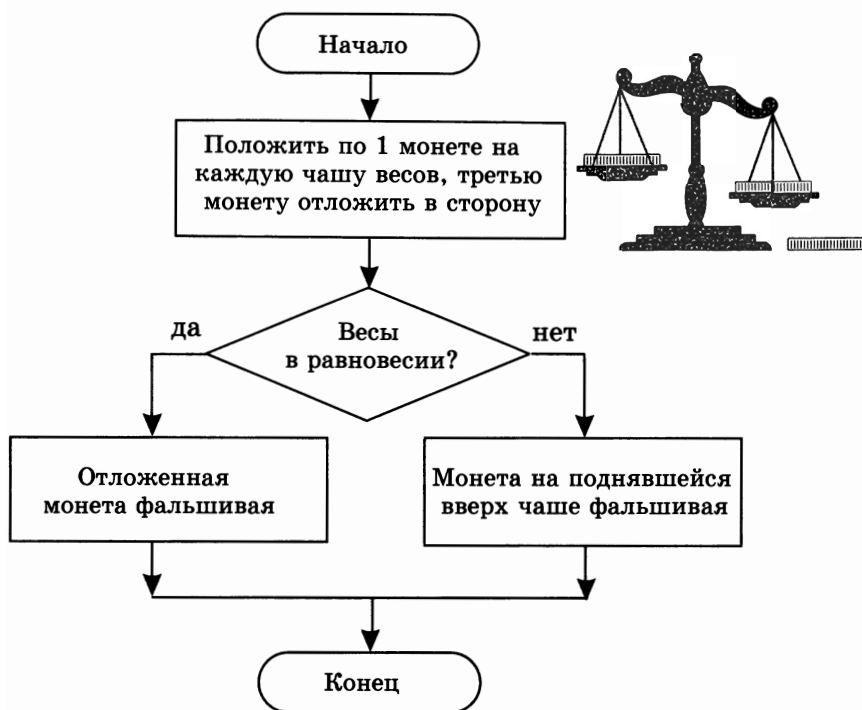
Форма организации действий, при которой в зависимости от выполнения или невыполнения некоторого условия совершается либо одна, либо другая последовательность действий, называется **ветвлением**.

Изобразим в виде блок-схемы последовательность действий ученика 6 класса Мухина Васи, которую он представляет себе так: «Если Павлик дома, будем решать задачи по математике. В противном случае следует позвонить Марине и вместе готовить доклад по биологии. Если же Марины нет дома, то надо сесть за сочинение».



А вот так, с помощью блок-схемы можно очень наглядно представить рассуждения при решении следующей задачи.

Из трех монет одинакового достоинства одна фальшивая (более легкая). Как ее найти с помощью одного взвешивания на чашечных весах без гирь?



### Алгоритмы с повторениями

На практике часто встречаются задачи, в которых одно или несколько действий бывает необходимо повторить несколько раз, пока соблюдается некоторое заранее установленное условие.

Форма организации действий, при которой выполнение одной и той же последовательности действий повторяется, пока выполняется некоторое заранее установленное условие, называется циклом (повторением).

Алгоритм, содержащий циклы, называется **циклическим алгоритмом** или **алгоритмом с повторениями**.

Ситуация, при которой выполнение цикла никогда не заканчивается, называется **зацикливанием**. Следует разрабатывать алгоритмы, не допускающие таких ситуаций.

Рассмотрим пример из математики.

Натуральное число называют простым, если оно имеет только два делителя: единицу и само это число<sup>1</sup>.

2, 3, 5, 7 — простые числа; 4, 6, 8 — нет. В III веке до нашей эры греческий математик Эратосфен предложил следующий алгоритм для нахождения всех простых чисел, меньших заданного числа  $n$ :

- 1) выписать все натуральные числа от 1 до  $n$ ;
- 2) вычеркнуть 1;
- 3) подчеркнуть наименьшее из неотмеченных чисел;
- 4) вычеркнуть все числа, кратные подчеркнутому на предыдущем шаге;
- 5) если в списке имеются неотмеченные числа, то перейти к шагу 3, в противном случае все подчеркнутые числа — простые.

Это циклический алгоритм. При его выполнении повторение шагов 3–5 происходит, пока в исходном списке остаются неотмеченные числа.

Вот так выглядит блок-схема действий школьника, которому перед вечерней прогулкой следует выполнить домашнее задание по математике:



<sup>1</sup> Напомним, что число 1 не относят ни к составным (имеющим более двух делителей), ни к простым числам.

## **Самое главное**

В зависимости от порядка выполнения команд можно выделить три типа алгоритмов:

- линейные алгоритмы;
- алгоритмы с ветвлениями;
- алгоритмы с повторениями.

Алгоритм, в котором команды выполняются в порядке их записи, то есть последовательно друг за другом, называется линейным.

Форма организации действий, при которой в зависимости от выполнения или невыполнения некоторого условия совершается либо одна, либо другая последовательность действий, называется ветвлением.

Форма организации действий, при которой выполнение одной и той же последовательности действий повторяется, пока выполняется некоторое заранее установленное условие, называется циклом (повторением).

## **Вопросы и задания**

1. Какие алгоритмы называют линейными?
2. Приведите пример линейного алгоритма.
3. Исполнитель «Вычислитель» умеет выполнять только две команды: умножать на 2 и прибавлять 1. Придумайте для него наиболее короткий план получения из 0 числа 50.
4. Какая форма организации действий называется ветвлением?
5. Какие условия должна была выполнить героиня сказки «Гуси-лебеди»?
6. Приведите пример алгоритма, содержащего ветвление.
7. Прочитайте отрывок из стихотворения Дж. Родари «Чем пахнут ремесла?»:

У каждого дела запах особый:  
 В булочной пахнет тестом и сдобой.  
 Мимо столярной идешь мастерской —  
 Стружкой пахнет и свежей доской.  
 Пахнет маляр скипидаром и краской.  
 Пахнет стекольщик оконной замазкой.  
 Куртка шофера пахнет бензином,  
 Блуза рабочего — маслом машинным.

Перефразируйте информацию о профессиях с помощью слов «ЕСЛИ ... ТО».

8. Вспомните, герои каких русских народных сказок совершают выбор, определяющий их судьбу.
9. Из 9 монет одинакового достоинства одна фальшивая (более легкая). За сколько взвешиваний на чашечных весах без гирь вы можете ее определить?
10. Какая форма организации действий называется повторением?
11. Приведите пример алгоритма, содержащего повторение.
12. В каких известных вам литературных произведениях имеет место циклическая форма организации действий?
13. Где окажется исполнитель, выполнивший 16 раз подряд следующую группу команд?  
**пройти 10 метров вперед**  
**повернуть на 90° по часовой стрелке**
14. Какую группу действий и сколько раз следует повторить при решении следующей задачи?  
 Сорок солдат подошли к реке, по которой на лодке катаются двое мальчиков. Как солдатам переправиться на другой берег, если лодка вмещает только одного солдата либо двух мальчиков, а солдата и мальчика уже не вмещает?

15. Вспомните задачу о Вычислителе, умеющем только умножать на 2 и прибавлять 1. Разрабатывать для него рациональные алгоритмы будет значительно проще, если воспользоваться следующей блок-схемой:



Используя эту блок-схему, разработайте рациональные алгоритмы получения из числа 0 чисел 1024 и 500.

# Материал для любознательных

---

### § 4.1. Путешествие в историю чисел

«Все есть число», — говорили мудрецы, подчеркивая необычайно важную роль чисел в жизни людей.

Известно множество способов представления чисел. В любом случае число изображается символом или группой символов (словом) некоторого алфавита. Такие символы называют цифрами.

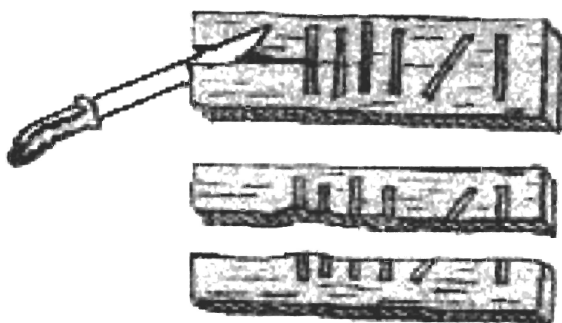
**Система счисления** — это совокупность приемов и правил для обозначения и именования чисел.

Люди научились считать очень давно, еще в каменном веке. Сначала они просто различали, один предмет перед ними или больше. Через некоторое время появилось слово для обозначения двух предметов. У некоторых племен Австралии и Полинезии до самого последнего времени было только два числительных: «один» и «два». А все числа, большие двух, получали названия в виде сочетаний этих двух числительных. Например, три — это «два, один», четыре — «два, два», пять — «два, два, один».

Как только люди начали считать, у них появилась потребность в записи чисел. Находки археологов на стоянках первобытных людей свидетельствуют о том, что первоначально количество предметов отображали равным количеством каких-либо значков: зарубок, черточек, точек.



Чтобы два человека могли точно сохранить некоторую числовую информацию, они брали деревянную бирку, делали на ней нужное число зарубок, а потом раскалывали бирку пополам.



Каждый уносил свою половинку и хранил ее. Этот прием позволял избегать «подделки документов». Ведь при возникновении спорной ситуации половинки можно было сложить и сравнить совпадение и число зарубок.

Такая система записи чисел называется **единичной (унарной)**, так как любое число в ней образуется путем повторения одного знака, символизирующего единицу.

Отголоски единичной системы счисления встречаются и сегодня. Так, чтобы узнать, на каком курсе учится курсант военного училища, нужно сосчитать, сколько полосок нашито на его рукаве. Того не осознавая, этим кодом пользуются малыши, показывая на пальцах свой возраст. Именно унарная система лежит в фундаменте арифметики, и именно она до сих пор вводит школьников в мир счета.








Единичная система — не самый удобный способ записи чисел. Записывать таким образом большие количества утомительно, да и сами записи при этом получаются очень длинными. С течением времени возникли иные, более экономичные системы счисления.

## § 4.2. Обозначение чисел и счет в Древнем Египте

Примерно в третьем тысячелетии до нашей эры египтяне придумали свою числовую систему, в которой для обозначения ключевых чисел 1, 10, 100 и так далее использовались специальные значки — иероглифы. Вот они:

						
1	10	100	1000	10 000	100 000	1 000 000

С течением времени эти знаки изменились и приобрели более простой вид:

						
1	10	100	1000	10 000	100 000	1 000 000

Все остальные числа составлялись из этих ключевых символов при помощи операции сложения. Например, чтобы изобразить 3252, рисовали три цветка лотоса (три тысячи), два свернутых пальмовых листа (две сотни), пять дуг (пять десятков) и два шеста (две единицы):



Величина числа не зависела от того, в каком порядке располагались составляющие его знаки: их можно было записывать сверху вниз, справа налево или вперемешку.

Система счисления называется **непозиционной**, если в ней количественные значения символов, используе-

мых для записи чисел, не зависят от их положения (места, позиции) в коде числа.

Система счисления Древнего Египта является непоозиционной. Особую роль у египтян играло число 2 и его степени. Умножение и деление они проводили путем последовательного удвоения и сложения чисел. Выглядели такие расчеты довольно громоздко.

Например, чтобы умножить 15 на 24, составляли следующую таблицу:

1	24
2 (1 · 2)	48
4 (2 · 2)	96
8 (4 · 2)	192
16 (8 · 2)	384

Здесь в левом столбце записаны результаты удвоений единицы, а в правом — числа 24. Записи делались до тех пор, пока не оказывалось, что из чисел левого столбца можно составить множитель 15:  $1 + 2 + 4 + 8 = 15$ . После этого складывались соответствующие числа правого столбца:  $24 + 48 + 96 + 192 = 360$ .

При делении египтяне многократно удваивали в правом столбце делитель и, соответственно, в левом столбце — число 1, пока числа правого столбца оставались не большими делимого. Далее из чисел правого столбца пытались составить делимое, и если это удавалось, то сумма соответствующих чисел в левом столбце давала искомое частное. Если делимое не делилось нацело на делитель, то получали частное и остаток.

Например, чтобы разделить 541 на 12, надо было составить таблицу:

1	12
2	24
4	48
8	96
16	192
32	384

$$541 = 384 + 96 + 48 + 12 + 1;$$

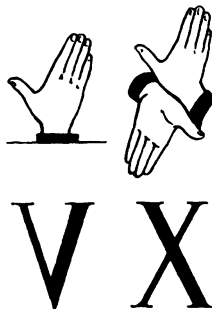
$$32 + 8 + 4 + 1 = 45.$$

Таким образом, при делении 541 на 12 получаем частное 45 и остаток 1.

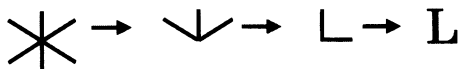
## § 4.3. Римская система счисления

Примером непозиционной системы счисления, которая сохранилась до наших дней, может служить система счисления, применявшаяся более двух с половиной тысяч лет назад в Древнем Риме.

В основе римской системы счисления лежат знаки I (один палец) для числа 1, V (раскрытая ладонь) для числа 5, X (две сложенные ладони) для 10, а также специальные знаки для обозначения чисел 50, 100, 500 и 1000.



Обозначения для последних четырех чисел с течением времени претерпели значительные изменения. Ученые предполагают, что первоначально знак для числа 100 имел вид пучка из трех черточек наподобие русской буквы Ж, а для числа 50 — вид верхней половинки этой буквы, которая в дальнейшем трансформировалась в знак L:



Для обозначения чисел 100, 500 и 1000 стали применять первые буквы соответствующих латинских слов (Centum — сто, Demimille — половина тысячи, Mille — тысяча).

Чтобы записать число, римляне использовали не только сложение, но и вычитание ключевых чисел. При этом применялось следующее правило.

Значение каждого меньшего знака, поставленного слева от большего, вычитается из значения большего знака.

Например, запись IX обозначает число 9, а запись XI — число 11. Десятичное число 28 представляется следующим образом:

$$\text{XXVIII} = 10 + 10 + 5 + 1 + 1 + 1.$$

Десятичное число 99 имеет такое представление:

$$\text{XCIX} = \underbrace{-10 + 100}_{\text{}} - \underbrace{1 + 10}_{\text{}}.$$

То, что при записи новых чисел ключевые числа могут не только складываться, но и вычитаться, имеет существенный недостаток: запись римскими цифрами лишает число единственности представления. Действительно, в соответствии с приведенным выше правилом, число 1995 можно записать, например, следующими способами:

$$\text{MCMXCV} = 1000 + (1000 - 100) + (100 - 10) + 5,$$

$$\text{MDCCCCLXXXV} = 1000 + 500 + 100 + 100 + 100 + 100 + 50 + 10 + 10 + 10 + 10 + 5,$$

$$\text{MVM} = 1000 + (1000 - 5),$$

$$\text{MDVD} = 1000 + 500 + (500 - 5) \text{ и так далее.}$$

Единых правил записи римских чисел до сих пор нет, но существуют предложения о принятии для них международного стандарта.

В наши дни любую из римских цифр предлагается записывать в одном числе не более трех раз подряд. На основании этого построена таблица, которой удобно пользоваться для обозначения чисел римскими цифрами:

Единицы	Десятки	Сотни	Тысячи
1 I	10 X	100 C	1000 M
2 II	20 XX	200 CC	2000 MM
3 III	30 XXX	300 CCC	3000 MMM
4 IV	40 XL	400 CD	
5 V	50 L	500 D	
6 VI	60 LX	600 DC	
7 VII	70 LXX	700 DCC	
8 VIII	80 LXXX	800 DCCC	
9 IX	90 XC	900 CM	

Эта таблица позволяет записать любое целое число от 1 до 3999. Чтобы это сделать, сначала запишите свое число, как обычно (в десятичной системе). Затем для цифр, стоящих в разрядах тысяч, сотен, десятков и единиц, по таблице подберите соответствующие кодовые группы.

Для того чтобы записать числа, большие 3999, применяют специальные правила, но знакомство с ними выходит за рамки нашего курса.

Римскими цифрами пользовались очень долго. Еще 200 лет назад в деловых бумагах числа должны были обозначаться римскими цифрами (считалось, что обычные арабские цифры легко подделать).

Римская система счисления сегодня используется в основном для наименования знаменательных дат, томов, разделов и глав в книгах.

## § 4.4. Алфавитные системы счисления

Наряду с иероглифическими в древности широко применялись алфавитные системы счисления, в которых числа изображались буквами алфавита. Так, в Древней Греции числа 1, 2, ..., 9 обозначали первыми девятью буквами греческого алфавита:  $\alpha = 1$ ,  $\beta = 2$ ,  $\gamma = 3$  и так далее. Для обозначения десятков применялись следующие девять букв:  $\iota = 10$ ,  $\kappa = 20$ ,  $\lambda = 30$ ,  $\mu = 40$  и так далее. Для обозначения сотен использовались последние девять букв:  $\rho = 100$ ,  $\sigma = 200$ ,  $\tau = 300$  и так далее.

Ниже приведен греческий алфавит с числовыми значениями входящих в него букв.

Буква	Название	Числовой эквивалент	Буква	Название	Числовой эквивалент	Буква	Название	Числовой эквивалент
Aα	Альфа	1	Ιι	Йота	10	Ρρ	Ро	100
Bβ	Бета	2	Κκ	Каппа	20	Σσ	Сигма	200
Γγ	Гамма	3	Λλ	Ламбда	30	Ττ	Тау	300
Δδ	Дельта	4	Μμ	Мю	40	Υυ	Ипсилон	400
Eε	Эпсилон	5	Νν	Ню	50	Φφ	Фи	500
—	—	6	Ξξ	Кси	60	Χχ	Хи	600
Zζ	Дзета	7	Οο	Омикрон	70	Ψψ	Пси	700
Ηη	Эта	8	Ππ	Пи	80	Ωω	Омега	800
Θθ	Тэта	9	—	—	90	—	—	900

Пропуск некоторых записей означает, что в древности алфавит содержал еще несколько букв.



## § 4.5. Славянский цифровой алфавит

Алфавитной нумерацией пользовались также южные и восточные славянские народы. У одних славянских народов числовые значения букв установились в порядке славянского алфавита, у других же (в том числе у русских) роль цифр играли не все буквы славянского алфавита, а только те из них, которые имелись и в греческом алфавите. Над буквой, обозначающей цифру, ставился специальный значок «̑» («титло»). При этом числовые значения букв возрастали в том же порядке, в каком следовали буквы в греческом алфавите (порядок букв славянского алфавита был несколько иным). Например, если записать в славянской нумерации числа 55, 288, 1 и 498, то получится фраза: не спи, а учи.

В России славянская нумерация сохранялась до конца XVII века. При Петре I возобладала так называемая арабская нумерация, которой мы пользуемся и сейчас. Славянская нумерация сохранилась только в богослужебных книгах.

Ниже приведен славянский цифровой алфавит:

Буква	Название	Числовой эквивалент	Буква	Название	Числовой эквивалент	Буква	Название	Числовой эквивалент
А̑	Аз	1	И̑	И	10	Р̑	Рцы	100
В̑	Веди	2	К̑	Како	20	С̑	Слово	200
Г̑	Глаголь	3	Л̑	Люди	30	Т̑	Твердо	300
Д̑	Добро	4	М̑	Мыслете	40	У̑	Ук	400
Е̑	Есть	5	Н̑	Наш	50	Ф̑	Ферт	500
З̑	Зело	6	Х̑	Кси	60	Х̑	Хер	600
З̑	Земля	7	О̑	Он	70	Ψ̑	Пси	700
И̑	Иже	8	П̑	Покой	80	Ω̑	Омега	800
Ф̑	Фита	9	Ч̑	Червь	90	Ц̑	Цы	900

Тысячи обозначались теми же буквами с «титлами», что и первые девять цифр, но слева внизу у них ставился специальный знак, например:

$$* \bar{a} = 1000 \quad * \bar{b} = 2000 \quad * \bar{z} = 7000$$

Десятки тысяч назывались «тьмами», их обозначали, обводя знаки единиц кружками. Например, числа 10 000, 20 000, 50 000 записывались так:



Сотни тысяч назывались «легионами», их обозначали, обводя знаки единиц кружками из точек. Например, числа 100 000 и 200 000 обозначались так:



Миллионы назывались «леордами», их обозначали, обводя знаки единиц кружками из лучей или запятых. Например, 1 000 000 и 2 000 000 записывались так:



Десятки миллионов назывались «воронами» или «вранами», их обозначали, обводя знаки единиц кружками из крестиков или ставя по обе стороны знака единиц букву К. Например, числа 10 000 000 и 20 000 000 обозначались так:



Сотни миллионов назывались «колодами». Для их обозначения над и под буквой, обозначающей единицы, ставились квадратные скобки. Например, числа 100 000 000 и 300 000 000 записывались в виде:

[А]                      [Г]

Интересно, что числа от 11 до 19 записывались так, как и говорились: сначала число единиц, потом число десятков. Например, 11 и 14 выглядели так:

[А]І                      [Д]І

Остальные числа записывались буквами слева направо. Например, числа 244 и 1993 изображались так:

[С][М][Д]                      \* [А][Ц][Ч][Г]

Мы рассмотрели так называемый «малый счет». В одной из славянских рукописей рассматривается и «великий счет», доходивший до числа  $10^{50}$ . Далее говорилось: «И более сего несть человеческому уму разумевати».

## § 4.6. Ясачные грамоты

Так как запись чисел с помощью алфавитной системы счисления была достаточно сложна, то в старину на Руси среди простого народа широко применялись системы счисления, отдаленно напоминающие римскую. С их помощью сборщики податей заполняли квитанции об уплате подати — ясака (ясачные грамоты) и делали записи в податной тетради.

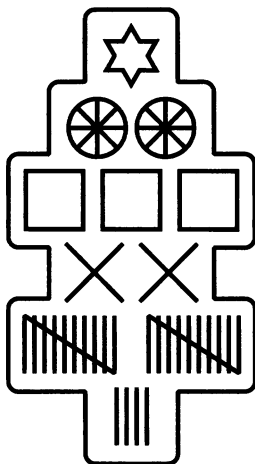
Вот текст закона об этих так называемых ясачных знаках:

«Чтобы на каждой квитанции, выдаваемой Родовитому Старосте, от которого внесен будет ясак, кроме изложения словами, было показано особыми знаками число внесенных рублей и копеек так, чтобы сдающие простым счетом сего числа могли быть уверены в справедливости показания. Употребляемые в квитанции знаки означают:

	— тысяча рублей,
	— сто рублей,
	— десять рублей,
	— один рубль,
	— десять копеек,
	— копейка.

Дабы не можно было сделать здесь никаких прибавлений, все таковые знаки очерчивать кругом прямыми линиями».

Например, 1232 рубля 24 копейки изображались так:



## § 4.7. Позиционные системы счисления

Рассмотренные нами иероглифические и алфавитные системы счисления имели один существенный недостаток — в них было очень трудно выполнять арифметические операции. Этому неудобства нет у позиционных систем.

Система счисления называется **позиционной**, если количественные значения символов, используемых для записи чисел, зависят от их положения (места, позиции) в коде числа.

Французский математик Пьер Симон Лаплас (1749–1827) такими словами оценил «открытие» позиционной системы счисления: «Мысль — выражать все числа немногими знаками, придавая им значение по форме, еще значение по месту, настолько проста, что именно из-за этой простоты трудно оценить, насколько она удивительна».

Основные достоинства любой позиционной системы счисления — простота выполнения арифметических операций и ограниченное количество символов, необходимых для записи любых чисел.

## § 4.8. Вавилонская система счисления

Идея приписывать цифрам разные величины в зависимости от того, какую позицию они занимают в записи числа, впервые появилась в Древнем Вавилоне примерно в III тысячелетии до нашей эры.

До нашего времени дошли многие глиняные таблички Древнего Вавилона, на которых решены сложнейшие задачи, такие как вычисление корней, отыскание объема пирамиды и др. Для записи чисел вавилоняне использовали всего два знака: клин вертикальный (единицы) и клин горизонтальный (десятки). Все числа от 1 до 59 записывались с помощью этих знаков, как в обычной иероглифической системе.

Пример:

$$\llcorner \llcorner = 12, \quad \llcorner \llcorner \llcorner \llcorner = 31, \quad \llcorner \llcorner \llcorner \llcorner \llcorner \llcorner \llcorner \llcorner = 45.$$

Все число в целом записывалось в позиционной системе счисления с основанием 60. Поясним это на примерах.

Запись  $\llcorner \llcorner \llcorner \llcorner \llcorner \llcorner \llcorner \llcorner$  обозначала  $6 \cdot 60 + 3 = 363$ , подобно тому как наша запись 63 обозначает  $6 \cdot 10 + 3$ .

Запись  $\llcorner \llcorner \llcorner \llcorner \llcorner \llcorner \llcorner \llcorner \llcorner \llcorner \llcorner \llcorner \llcorner \llcorner$  обозначала  $32 \cdot 60 + 52 = 1972$ ; запись  $\llcorner \llcorner \llcorner \llcorner \llcorner \llcorner \llcorner \llcorner \llcorner \llcorner \llcorner \llcorner \llcorner \llcorner$  обозначала  $1 \cdot 60 \cdot 60 + 2 \cdot 60 + 4 = 3724$ .

Был у вавилонян и знак, игравший роль нуля. Им обозначали отсутствие промежуточных разрядов. Но отсутствие младших разрядов не обозначалось никак. Так, число  $\llcorner \llcorner$  могло обозначать и 3, и  $180 = 3 \cdot 60$  и  $10\,800 = 3 \cdot 60 \cdot 60$  и так далее. Различать такие числа можно было только по смыслу.

Отголоски этой системы счисления мы находим в сохранившемся до наших дней обыкновении делить один час на 60 минут, одну минуту на 60 секунд, полный угол — на 360 градусов.

## § 4.9. Десятичная система счисления

Обычная система записи чисел, которой мы привыкли пользоваться в повседневной жизни, с которой мы знакомы с детства, в которой производим все наши вычисления, — пример позиционной системы счисления.

В привычной нам системе счисления для записи чисел используются десять различных знаков (цифры 0, 1, 2, 3, 4, 5, 6, 7, 8 и 9). Поэтому ее называют десятичной. Из двух написанных рядом одинаковых цифр левая в 10 раз больше правой. Не только сама цифра, но и ее место, ее позиция в числе имеют определяющее значение. Поэтому данную систему счисления называют позиционной.

В десятичном числе  $255 = 2 \cdot 100 + 5 \cdot 10 + 5 \cdot 1$  цифры 5, находящиеся на разных позициях, имеют различные количественные значения — 5 десятков и 5 единиц. При перемещении цифры на соседнюю позицию ее «вес» изменяется в 10 раз.

Потребовалось много тысячелетий, чтобы люди научились называть и записывать числа так, как это делаем мы с вами. Начало этому было положено в Древнем Египте и Вавилоне. Дело в основном завершили индийские математики в V–VII веках нашей эры. Важным достижением индийской науки было введение особого обозначения для пропуска разрядов — нуля. Арабы, познакомившись с этой нумерацией первыми, по достоинству ее оценили, усвоили и перенесли в Европу. Получив название **арабской**, эта система в XII веке нашей эры распространилась по всей Европе и, будучи проще и удобнее остальных систем счисления, быстро их вытеснила. Сегодня десятичными цифрами выражаются время, номера домов и телефонов, цены, бюджет, на них базируется метрическая система мер.



Время многократно изменяло облик десятичных цифр, пока они не приобрели привычный для нас вид. Некогда написание цифр было таким:



Такое изображение десятичных цифр не случайно: каждая цифра обозначает число, соответствующее количеству углов в ней. Подсчитайте и убедитесь в этом сами!

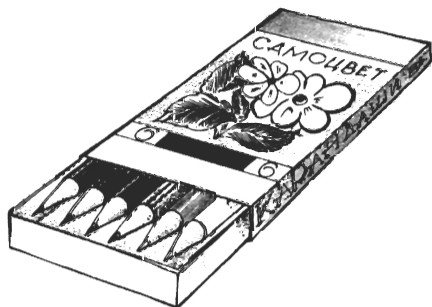
Арифметические действия над десятичными числами производятся с помощью достаточно простых операций, в основе которых лежат известные каждому школьнику таблицы умножения и сложения, а также правило переноса: если в результате сложения двух цифр получается число, которое больше или равно 10, то оно записывается с помощью нескольких цифр, находящихся на соседних позициях.

Изучаемые в самом раннем возрасте, эти правила в результате повседневной практики усваиваются так прочно, что мы оперируем ими уже подсознательно. По этой причине сегодня многие люди даже не догадываются о существовании других систем счисления.

## § 4.10. Другие позиционные системы счисления

Широкое распространение до первой трети XX века имели элементы двенадцатеричной системы счисления. Число двенадцать (дюжина) даже составляло конкуренцию десятке в борьбе за почетный пост основания общеупотребительной системы счисления. Дело в том, что число 12 имеет больше делителей (2, 3, 4, 6), чем 10 (2 и 5). Поэтому в двенадцатеричной системе счисления более удобно производить расчеты, чем в десятичной. Неудивительно, что в XIX веке среди математиков раздавались голоса за полный переход на эту систему счисления. И только возможность счета по пальцам рук склонила чашу весов на сторону числа 10. Тем не менее дюжина достаточно прочно вошла в нашу жизнь: в сутках две дюжины часов, час делится на пять дюжин минут, круг содержит тридцать дюжин градусов, фут делится на двенадцать дюймов. Влияние двенадцатеричной системы счисления ощущается сегодня хотя бы в том, что карандашей или фломастеров в наборе обычно бывает 6, 12 или 24; чайные и столовые сервизы бывают на 6 или на 12 персон; комплект носовых платков — 12 штук.

А вот шведский король Карл XII в 1717 году увлекался восьмеричной системой, считал ее более удобной, чем десятичная, и намеревался королевским указом ввести ее как общегосударственную. Только неожиданная смерть короля помешала осуществлению столь необычного намерения.



## § 4.11. О происхождении слова «алгоритм»



Правила выполнения арифметических действий над целыми числами и простыми дробями в десятичной системе счисления впервые были сформулированы выдающимся средневековым ученым по имени Мухаммед ибн Муса ал-Хорезми (в переводе с арабского это означает «Мухаммед, сын Мусы из Хорезма»), сокращенно Ал-Хорезми.

Ал-Хорезми жил и творил в IX веке. Арабский оригинал его арифметического труда утерян, но имеется латинский перевод XII века, по которому Западная Европа ознакомилась с десятичной позиционной системой счисления и правилами выполнения в ней арифметических действий.

Ал-Хорезми стремился к тому, чтобы сформулированные им правила были понятны для всех грамотных людей. Достичь этого в веке, когда еще не была разработана математическая символика (знаки операций, скобки, буквенные обозначения и т. п.), было очень трудно. Но Ал-Хорезми удалось выработать в своих трудах такой стиль четкого, строгого словесного предписания, который не давал читателю никакой возможности уклониться от предписанного или пропустить какие-нибудь действия.

В латинском переводе книги Ал-Хорезми правила начинались словами «Алгоризми сказал». С течением времени люди забыли, что «Алгоризми» — это автор правил, и стали сами эти правила называть алгоритмами. Постепенно «Алгоризми сказал» преобразовалось в «алгоритм гласит».

Таким образом, слово «алгоритм» происходит от имени ученого Ал-Хорезми. Как научный термин первоначально оно обозначало лишь правила выполнения действий в десятичной системе счисления. С течением времени это слово приобрело более широкий смысл и стало обозначать любые точные правила действий. В настоящее время слово «алгоритм» является одним из важнейших понятий науки информатики.

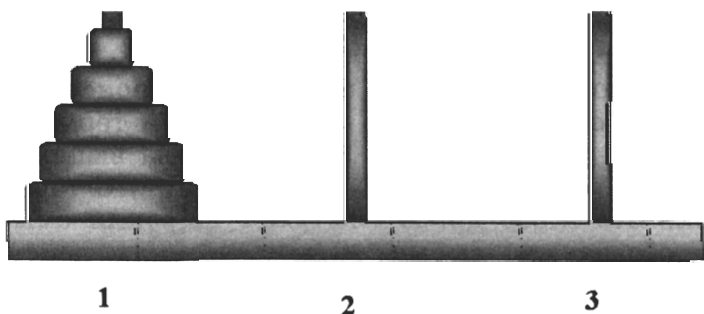
## § 4.12. Ханойская башня, или Один замечательный алгоритм

Одна из древних легенд гласит: «В непроходимых джунглях недалеко от города Ханоя есть храм бога Браммы. В нем находится бронзовая плита с тремя алмазными стержнями. На один из стержней бог при сотворении мира нанизал 64 диска разных диаметров из чистого золота. Наибольший диск лежит на бронзовой плите, а остальные образуют пирамиду, сужающуюся кверху. Это башня Браммы. Работая день и ночь, жрецы храма переносят диски с одного стержня на другой, следуя законам Браммы:

- 1) диски можно перемещать с одного стержня на другой только по одному;
- 2) нельзя класть больший диск на меньший;
- 3) нельзя откладывать диски в сторону, при переносе дисков с одного стержня на другой можно использовать промежуточный третий стержень, на котором диски должны находиться тоже только в виде пирамиды, сужающейся кверху.

Когда все 64 диска будут перенесены с одного стержня на другой, наступит конец света».

Эта древняя легенда положена в основу задачи о Ханойской башне: переместить  $n$  дисков со стержня 1 на стержень 3, используя промежуточный стержень 2 и соблюдая законы Браммы.



Если башня состоит из одного диска, то она переносится за один ход:  $1 \rightarrow 3$ .

Башня из двух дисков переносится за три хода:  $1 \rightarrow 2$ ,  $1 \rightarrow 3$ ,  $2 \rightarrow 3$ .

Для переноса башни из трех дисков потребуется уже семь ходов:  $1 \rightarrow 3$ ,  $1 \rightarrow 2$ ,  $3 \rightarrow 2$ ,  $1 \rightarrow 3$ ,  $2 \rightarrow 1$ ,  $2 \rightarrow 3$ ,  $1 \rightarrow 3$ . Обратите внимание, за первые три хода мы переносим башню из двух верхних дисков на второй промежуточный стержень. Затем переносим самый большой диск с первого стержня на третий и еще раз проделываем хорошо знакомую нам операцию: переносим башню из двух дисков на третий диск.

Следовательно, чтобы перенести башню из четырех дисков с первого стержня на третий, необходимо действовать по плану:

- 1) перенести башню из трех верхних дисков с первого стержня на второй (7 ходов);
- 2) самый большой диск перенести с первого стержня на третий (1 ход);
- 3) перенести башню из трех дисков со второго стержня на третий (7 ходов).

Всего на перенос потребуется 15 ходов.

Рассуждая аналогичным образом, сосчитаем число ходов, необходимых для переноса башни из пяти дисков:  $15 + 1 + 15 = 2 \cdot 15 + 1 = 31$ .

Для башни из 6 дисков получаем:  $2 \cdot 31 + 1 = 63$  и т. д.

Рассмотренный нами алгоритм решения задачи «Ханойская башня» обладает одним удивительным свойством: в ходе его выполнения для башни, состоящей из  $n$  колец, мы используем алгоритм для чуть более простой ситуации — переноса башни, состоящей из  $n - 1$  кольца. В свою очередь, в алгоритме для башни из  $n - 1$  кольца используется этот же алгоритм для  $n - 2$  колец и т. д.

Прием, когда некоторый процесс описывается через самого себя, называется рекурсией. Алгоритм решения задачи «Ханойская башня» — пример рекурсивного алгоритма.

## ? Вопросы и задания

1. Проведите необходимые вычисления и заполните следующую таблицу:

Число дисков	Число ходов
7	
8	
9	
10	
20	1 048 575
21	
30	1 073 741 823
31	
39	
40	1 099 511 627 775

2. Назовите числа: 1 048 575, 1 073 741 823, 1 099 511 627 775. Что это за числа?
3. Для того чтобы переместить башню из 64 дисков, при безошибочной работе потребуется 18 446 744 073 709 551 615 перекладываний. Сколько уйдет на это времени, если считать, что на одно перекладывание уходит одна секунда? Для выполнения вычислений используйте приложение Калькулятор.
4. Переместите башню из пяти дисков, действуя по следующим правилам: на первом, третьем и других нечетных шагах переносите самый маленький диск «по кругу» (1–2, 2–3, 3–1); на четных шагах переносите диск между теми стержнями, где нет самого маленького кольца. Что вы можете сказать об этом алгоритме?

## § 4.13. Руки — первый инструмент для счета

Каждый человек постоянно сталкивается с необходимостью обработки числовой информации.

Уже в каменном веке, когда люди собирали плоды, ловили рыбу и охотились на животных, возникла потребность в счете. На местах стоянок первобытных людей ученые находили кости с зарубками — так наши далекие предки фиксировали количество предметов.

Но количество предметов то увеличивалось, то уменьшалось, поэтому важно было уметь складывать и вычитать. Помогал в этом нашим далеким предкам их первобытный «компьютер» — десять пальцев на руках. Загибал человек пальцы — складывал, разгибал — вычитал. Точно так же, как делает это каждый маленький ребенок, когда учится считать.

В Западной Европе существовала целая система, позволявшая представлять на пальцах числа до 9999.

Существует множество приемов, позволяющих с помощью пальцев производить различные арифметические операции. Вот прием, позволяющий запомнить таблицу умножения на 9.

Положите обе руки рядом, ладонями на стол. Мысленно пронумеруйте все пальцы обеих рук слева направо. Приподнимите вверх палец, соответствующий числу, на которое требуется умножить 9. Число пальцев, расположенных слева от поднятого, дает число десятков, а расположенных справа — единиц искомого результата.

## § 4.14. Абак и счёты

На пальцах считать удобно, только результат счёта хранить нельзя. Не станешь же целый день ходить с загнутыми пальцами. И человек догадался — для счёта можно использовать все, что попадет под руку. Камешки, палочки, косточки, веревки и так далее. Пасет пастух стадо овец. На поясе у него веревка, а на веревке столько узелков, сколько овец в стаде. Родился ягненок — пастух завязал еще один узелок. Утащили волки двух овец — развязал два узелка.

С развитием цивилизации появлялись различные приемы счёта. Они были необходимы и сборщикам налогов, и купцам, и ремесленникам, и ростовщикам. Искусством счёта владели немногие специально обученные люди — счетчики. Они использовали счетные инструменты — абак.



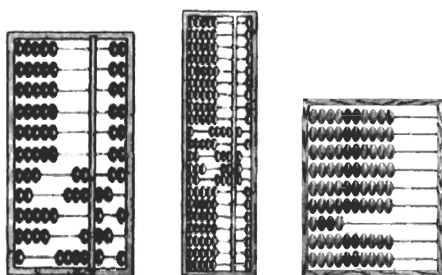
Простейший абак — это доска с прорезанными в ней желобами. Чтобы найти сумму двух чисел (например, 258 и 125), счетчик сначала обозначал на абаке первое слагаемое. Для этого он укладывал в нижнем желобе 8 камешков, в следующем желобе — 5 камешков и 2 камешка — в третьем желобе. Если в каком-то разряде в числе стоял нуль, то соответствующий желоб оставался пустым. Дальше счетчик добавлял в последний желоб к имеющимся там 8 камешкам еще 5, затем вынимал оттуда 10 (там оставалось 3) и 1 камешек добавлял во второй желоб. Потом добавлял во второй желоб еще 2 камешка и 1 камешек — в третий желоб. После этого камешки на доске показывали число 383.



Абаки использовались уже в V–IV веках до нашей эры. Их изготавливали из бронзы, камня, слоновой кости, цветного стекла. Слово «абак» имеет греческое происхождение и буквально означает «пыль», хотя его смысловое значение — «счетная доска». В чем тут дело? Ответ прост: изначально камешки раскладывали на совершенно ровной доске, а чтобы они не скатывались со своего первоначального положения, доска покрывалась тонким слоем песка или пыли. А от слова «камешек» (по латыни — «calculus») произошло название современного счетного прибора — «калькулятор».

Абак использовался и в Древней Греции, и в Древнем Риме, а затем и в Западной Европе вплоть до XVIII века. Он похож на знакомые вам счеты — косточки на вставленных в рамку металлических спицах.

Счеты использовали разные народы, и у каждого народа они имели свои особенности. Так, в русских счетах по десять косточек в каждом ряду, а в западноевропейских — по девять. У китайских счетов суанпан на каждой проволоке по семь шариков, причем два отделены от остальных пяти. Каждый из этих двух шариков означает пять единиц данного разряда. Такое усовершенствование позволяет уменьшить число шариков в счетах.



В Японии и в наши дни проводятся соревнования по скорости счета между людьми, вооруженными японскими счетами соробан, и операторами вычислительных машин. Причем, как правило, побеждают вычислители на счетах. Ведь чтобы машина начала считать, для нее надо составить программу.

## § 4.15. Арифмометр

Шло время, и потребности людей в обработке числовой информации возрастали. Первые идеи механизации вычислительного процесса появились в конце XV — начале XVI века. Об этом свидетельствует найденный в конце 60-х годов прошлого века эскиз суммирующего устройства, разработанный еще Леонардо да Винчи.

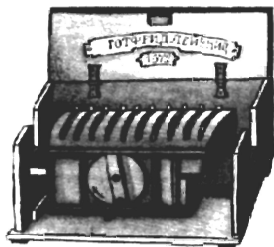
В XVII веке физики и астрономы столкнулись с необходимостью произведения сложных и громоздких вычислений. Им требовались машины, способные выполнять большой объем вычислений за малое время и с высокой точностью.

В 1642 году молодым французом Блезом Паскалем, ставшим в будущем знаменитым физиком и математиком, была создана и завоевала огромную популярность первая механическая счетная машина — арифмометр. Счетная машина Паскаля была похожа на маленькую шкатулку, на крышке которой, как на часах, были расположены циферблаты. На них и устанавливали числа. Для цифр разных разрядов были отведены различные зубчатые колеса. Каждое предыдущее колесо соединялось с последующим с помощью одного зубца. Этот зубец вступал в зацепление с очередным колесом только после того, как были пройдены все девять цифр данного разряда. Пусть, например, к шести прибавляется пять, тогда колесо единиц совершит в сумме 11 шагов; в положении «0», следующем после положения «9», сцепится с колесом десятков и повернет его на один зубец. В результате колеса покажут число 11.



За три века, прошедшие с момента создания первого арифмометра, было создано около четырехсот видов разнообразных механических счетчиков и счетных машин. Большинство из этих изобретений уже забыто. Но были и такие изобретения, которые явились важными событиями в истории вычислительных машин.

В 1677 году великий немецкий математик и философ Готфрид Вильгельм Лейбниц сконструировал свою счетную машину, позволявшую не только складывать и вычитать, но также умножать и делить многозначные числа. В своем арифмометре Лейбниц использовал вместо колес цилиндры. На цилиндры были нанесены цифры. Каждый цилиндр имел девять рядов выступов: один выступ — в первом ряду, два — во втором и так до девятого, содержащего девять выступов. Эти цилиндры были подвижными и устанавливались в определенные положения оператором.



Большой вклад в усовершенствование счетных машин внесли русские ученые и инженеры. Так арифмометр, созданный в 1874 году русским инженером Однером, успешно конкурировал с лучшими арифмометрами европейских фирм и нашел применение во всем мире. Его модификация «Феликс» выпускалась в нашей стране до 50-х годов XX века.

Арифмометры долгое время обладали серьезным недостатком: каждый результат вычислений вручную записывался на листке бумаги. Пора было позаботиться о том, чтобы счетная машина сама печатала на бумаге ответ, тем более что пишущая машинка уже была изобретена. И вот в 1889 году появилась первая счетная машина, снабженная печатающим устройством.

## § 4.16. Машина Бэббиджа



В середине XIX века английский математик, профессор Кембриджского университета Чарльз Бэббидж разработал несколько проектов механических вычислительных машин, в том числе проект первой программируемой вычислительной машины.

Ворота, через которые завозятся сырье и инструкции по его обработке.  
(Клавиатура, мышь и пр.)



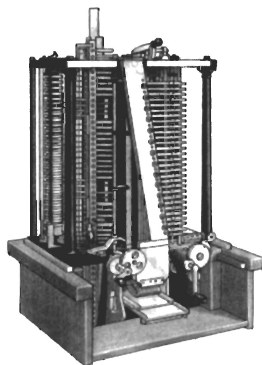
Машина, придуманная Чарльзом Бэббиджем, была похожа на настоящую фабрику по производству вычислений. На любой фабрике есть склад, где хранятся сырье и готовая продукция. Есть цех, где эта продукция производится. Есть контора, которая управляет производством. Машина Бэббиджа имела подобную конструкцию. Набор специальных колес — «склад» чисел. Здесь запоминаются исходные данные и результаты вычисления. Механизм из шестеренок, рычагов и пружин — «цех». Тут производятся вычисления. Есть и «контора», которая управляет всем вычислительным процессом с помощью заранее подготовляемых вычис-

лителем картонных лент с отверстиями — перфокарт<sup>1</sup>. Машина считает сама — работает по программе. Результаты вычислений она пробивает на металлических пластинках. С таких пластинок их можно печатать.

Несмотря на то что машина Бэббиджа представляла собой важный шаг вперед в технике вычислений, полностью реализована она не была. После 25 лет труда и огромных издержек изобретатель был вынужден отказаться от ее завершения.

Проекты огромных механических вычислительных устройств Бэббиджа стоят в ряду наиболее поразительных достижений человеческой мысли XIX века, хотя ему и не удалось воплотить свои планы в жизнь.

В конце прошлого века англичане решили построить хотя бы одну из машин, спроектированных Чарльзом Бэббиджем. Началась напряженная работа по созданию автоматического калькулятора (Разностной машины). И в год 200-летия со дня рождения знаменитого англичанина (1991) машина была построена и произвела серьезные вычисления. Этот успех доказал, что неудачи изобретателя были вызваны упущениями в реализации замысла, а не ошибками в самом проекте.



<sup>1</sup> Идею использования перфокарт Бэббидж позаимствовал у французского изобретателя Жозефа Жаккара, который в 1801–1804 годах применил карточки с пробитыми отверстиями для контроля ткацких операций. Карточки с разным расположением отверстий давали различные узоры при переплетении нитей.

## § 4.17. Счетная машина Холлерита



Если Чарльз Бэббидж был первым, кому пришла идея использовать перфокарты применительно к вычислительной технике, то первым, кто практически реализовал эту идею, был американский инженер Герман Холлерит, разработавший машину для обработки результатов переписи населения.

Сотни людей занимались этой огромной работой. Надо было обойти все улицы во всех городах и поселках. Зайти в каждый дом и в каждую квартиру. Записать сведения о каждой семье и каждом человеке. Наконец все данные собраны. И тут, оказывается, начинались главные трудности. Как обработать результаты переписи? Как сосчитать всех жителей страны? Да не просто сосчитать, а ответить на самые разные вопросы. Сколько в стране мужчин и женщин? Детей и стариков? Школьников и студентов? Горожан и сельских жителей? Рабочих, инженеров, врачей, учителей? На эту работу уходило до восьми лет. Если учесть, что в США перепись населения проводилась каждые 10 лет, то получается, что, едва закончив обработку данных одной переписи, нужно было сразу приступить к новой.



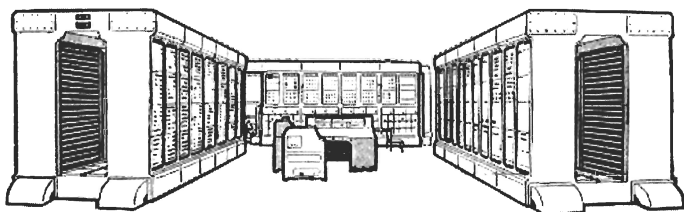
Вот как, по рассказу самого Холлерита, пришла ему в голову идея создания нового счетчика. Однажды на железнодорожной станции он наблюдал за работой кондуктора, когда тот пробивал дырочки в билетах. Так обозначалась станция, до которой ехал пассажир. И Холлерит решил изготовить такие же карты для проводимой переписи населения. Он распределил вопросы так, чтобы ответ можно было обозначать дырочкой в строке. Пол и возраст, работа и место жительства — все обозначалось отверстиями. Все эти данные потом «прочитывались» машиной, которая прощупывала перфокарту системой игл. Если напротив иглы оказывалось отверстие, то игла, пройдя сквозь него, касалась металлической поверхности, расположенной под картой и замыкала контакт. К показаниям соответствующего счетчика автоматически добавлялась единица.

В 1890 году счетно-аналитическая машина Холлерита впервые использовалась при обработке результатов очередной переписи и сократила период обработки с восьми до трех лет.

## § 4.18. Поколения ЭВМ

Первая полностью электронная вычислительная машина «Эниак» была построена в США в 1946 году. Ее размеры были громадны: более 30 м в длину и 85 м<sup>3</sup> по занимаемому объему. Вес машины равнялся весу четырех африканских слонов — примерно 30 тоннам. Хранение и обработка данных в ней осуществлялись с помощью 18 тысяч электронных ламп. В нашей стране первая ЭВМ была построена в 1951 году.

В 1953 году наша промышленность стала выпускать электронную вычислительную машину «Стрела». Она состояла из десятков больших металлических шкафов, в которых находились сотни электронных ламп (применявшихся также в радиоаппаратуре). Рядом стояли мощные трансформаторы, обеспечивавшие нужное напряжение для ламп. Лампы при работе сильно нагреваются. Чтобы охлаждать тысячи ламп первых компьютеров, требовались мощные вентиляционные установки. Вычислительная машина «Стрела» вместе со вспомогательным оборудованием занимала площадь в 500 квадратных метров. Этого хватило бы на 10 квартир.



Гигантские компьютеры на электронных лампах 50-х годов XX века составили первое поколение вычислительных машин.

Спустя десятилетие, в начале 60-х годов на смену электронным лампам пришли транзисторы. С ними связано появление второго поколения ЭВМ.

Металлы — серебро, медь, алюминий — проводят электрический ток. Их называют проводниками. Стекло, фарфор, пластмасса ток не проводят. Это изоляторы. А вот некоторые редкие вещества — кремний, гер-





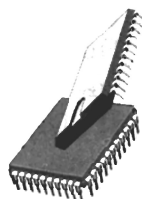
Электронная лампа



Транзисторы

маний, селен — то проводят электрический ток, то не проводят, в зависимости от его направления. Эти вещества наполовину изоляторы, наполовину проводники. Их называют полупроводниками. Они и стали основой для транзисторов — маленьких кристалликов полупроводника с двумя металлическими усиками-проводками. Транзистор был значительно меньше лампы, весил несколько граммов и практически не грелся. К тому же один транзистор был способен заменить 40 ламп. Машины стали значительно меньше, надежнее, их быстроедействие возросло.

Рождение машин **третьего поколения** связывают с появлением интегральных схем — кремниевых кристаллов с миниатюрной электронной схемой. Слово «интегральный» означает «цельный, единый». Размер такой схемы — не больше горошины, а транзисторов в ней упакованы тысячи. Машины уменьшились настолько, что уже могли размещаться на письменном столе.

Интегральные  
схемы

С развитием микроэлектроники появилась возможность размещать на кристалле не одну, а тысячи интегральных схем. В 1980 году на кристалле площадью около  $1,5 \text{ см}^2$  удалось разместить центральный процессор небольшой ЭВМ. Началась эпоха **микрокомпьютеров**.

Процесс развития вычислительной техники продолжается и сегодня. Можно предположить, что в ХХI веке компьютеры станут еще более мощными, еще меньшими по размерам и еще более простыми в использовании.

## § 4.19. Графические исполнители

Графические исполнители, с которыми мы будем знакомиться, могут вычерчивать различные фигуры на поле размером 639 на 463 единицы и закрашивать их любым из 16 цветов. Они выполняют чертежи в собственной системе координат, которая несколько отличается от той, что принята в математике и рассматривалась вами ранее: здесь начало координат расположено в левом верхнем углу, ось  $OX$  направлена вправо, а  $OY$  — сверху вниз. Среда для графических исполнителей задается командой SCREEN 12.

### Исполнитель DRAW

Художник по имени DRAW (произносится «дро-о») умеет рисовать картинки, состоящие из ломаных линий (в ломаной линии конец предыдущего отрезка является началом следующего).



DRAW умеет многое. Но, как и любой хороший исполнитель, он совершенно безынициативен: DRAW ничего не может делать сам, без наших указаний. Наша задача — научиться им управлять.

Художник DRAW невидим; по экрану он перемещается бесследно («прыгает») или оставляет за собой след. В начале работы DRAW обычно находится в центре экрана, а далее — в той точке, координаты которой задавались ему последний раз.

Точку, координаты которой задавались последними, будем называть текущей.

Чтобы нарисовать ломаную линию, следует отдать команду:

**DRAW" <командная строка> "**

Здесь:

**DRAW** — имя исполнителя;  
 <командная строка> — строка, состоящая из указаний, понятных исполнителю **DRAW**. Отдельные указания (команды) можно отделять друг от друга с помощью пробела или не отделять вообще.

Часть СКИ исполнителя **DRAW** приведена в следующей таблице:

№	Команда	Описание
1	<i>cn</i>	Установка цвета рисования <i>n</i> (1–16)
2	<i>mx, y</i>	Движение из текущей точки в точку с координатами ( <i>x, y</i> )
3	<i>b</i>	Движение, но не рисование («прыжок»)
4	<i>up</i>	Вверх на <i>n</i>
5	<i>dn</i>	Вниз на <i>n</i>
6	<i>ln</i>	Влево на <i>n</i>
7	<i>rn</i>	Вправо на <i>n</i>
8	<i>en</i>	По диагонали вверх и вправо на <i>n</i>
9	<i>fn</i>	По диагонали вниз и вправо на <i>n</i>
10	<i>gn</i>	По диагонали вниз и влево на <i>n</i>
11	<i>hn</i>	По диагонали вверх и влево на <i>n</i>
12	<i>n</i>	Двигаться, рисовать, затем вернуться в исходную точку

По команде *mx, y* **DRAW** прочертит отрезок от текущей точки до точки с координатами (*x, y*) и будет считать текущей точку с координатами (*x, y*).

Если использовать команду *bm<sub>x, y</sub>*, то **DRAW** «прыгнет» в точку с указанными координатами. Где бы ни был художник перед этой командой, выполнив ее, он окажется в точке с координатами (*x, y*), но ни-

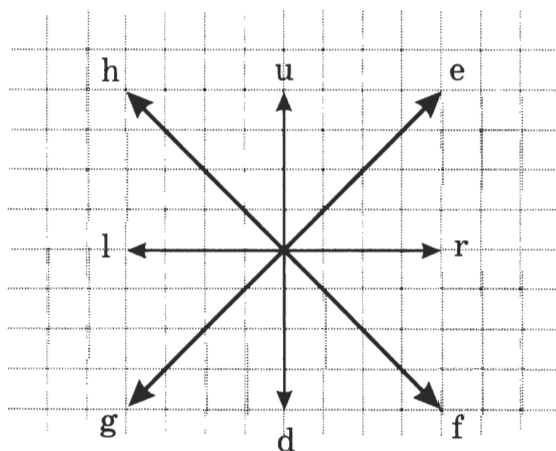
каких видимых изменений на экране не произведет. Другими словами, эта команда только изменит текущую точку.

Исполнитель DRAW может перемещаться по экрану вверх, вниз, влево и вправо на заданное число шагов. Другими словами, он может выполнять команды:

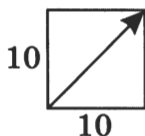
- вверх — u (up);
- вниз — d (down);
- влево — l (left);
- вправо — r (right).

Буквы u, d, l, r задают направление. Число шагов, которые необходимо выполнить в заданном направлении, указывается после соответствующей буквы.

С помощью команд e, f, g, h исполнитель DRAW может передвигаться по диагональным направлениям, показанным на рисунке:



**Внимание!** Команда e10 означает, что DRAW передвинется из исходной точки по диагонали в правый верхний угол квадрата со стороной 10 единиц.

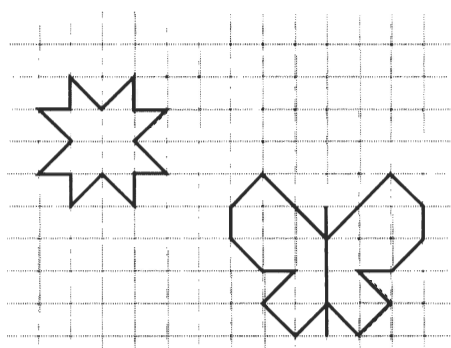


Следовательно, шаг исполнителя, определяемый командами e, f, g, h, длиннее шага, определяемого командами u, d, l, r.

Например, следующие командные строки позволяют вывести на экран рисунки цветка и бабочки:

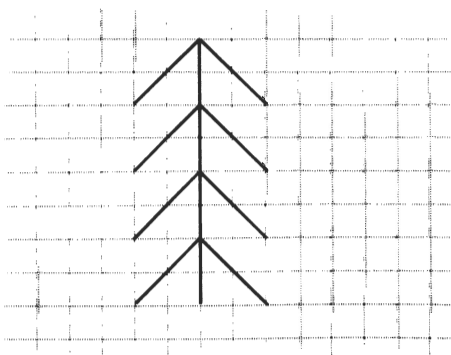
```
DRAW"e10 d10 r10 g10 f10 l10 d10 h10
g10 u10 l10 e10 h10 r10 u10 f10"
```

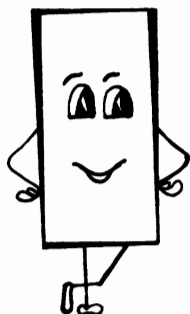
```
DRAW"u40 d10 e20 f10 d10 g10 l10 f10
g10 h10 g10 h10 e10 l10 h10 u10 e10 f20"
```



Чтобы координатам текущей точки возвращались их предыдущие значения, необходимо перед командой e, r, l, h, g, f, d, u и m использовать команду n.

Например, повторив 4 раза фрагмент командной строки ng20 nf20 d20, можно нарисовать вот такую ель:





## Исполнитель LINE

Нарисовать отрезок, квадрат, прямоугольник или какую-нибудь другую сложную линию можно поручить исполнителю по имени LINE.

### Отрезок

Чтобы изобразить отрезок, исполнителю необходимо отдать следующую команду:

LINE ( $x_n, y_n$ ) — ( $x_k, y_k$ ),  $c$

Здесь:

LINE — имя исполнителя;

( $x_n, y_n$ ) — координаты начала отрезка;

( $x_k, y_k$ ) — координаты конца отрезка;

$c$  — номер цвета рисования контура. Его можно не указывать. По умолчанию цвет будет текущим, то есть таким, каким был назначен ранее, или белым, если он не назначался вами вообще.

### Ломаная линия

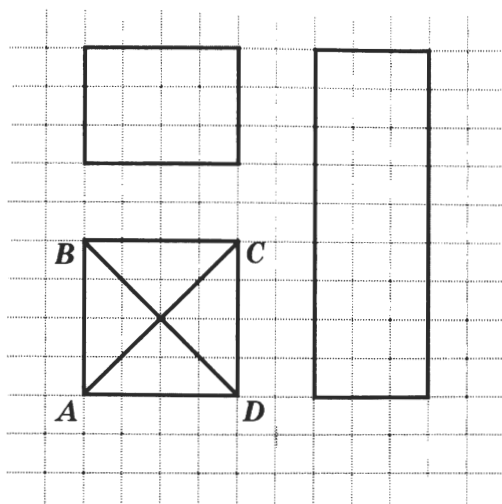
Чтобы изобразить звено ломаной линии, можно применить сокращенную форму записи команды LINE:

LINE — ( $x_k, y_k$ )

Здесь указывается только конец отрезка, а его началом служит текущая точка (координаты которой вводились последними).

### Прямоугольник

Рассмотрим прямоугольники, стороны которых параллельны осям координат:



Отрезки, соединяющие противоположные вершины прямоугольника, называются его диагоналями.  $AC$  и  $BD$  — диагонали прямоугольника  $ABCD$ .

Можно использовать команду **LINE** для того, чтобы нарисовать закрашенный прямоугольник, задав только его диагональ:

**LINE** ( $x_a, y_a$ ) — ( $x_c, y_c$ ),  $c$ ,  $bf$

Здесь:

( $x_a, y_a$ ), ( $x_c, y_c$ ) — координаты точек  $A$  и  $C$  соответственно;

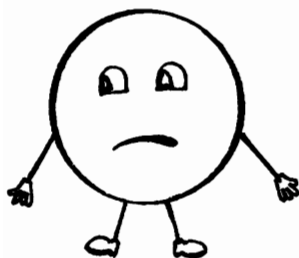
$c$  — цвет контура и внутренней области прямоугольника.

К обычной команде мы добавили буквы  $b$  и  $f$ :  $b$  — первая буква английского слова «box» — «коробка»;  $f$  — первая буква английского слова «full» — «полный». Таким образом, мы отдаем исполнителю команду нарисовать не линию, а «коробку» — прямоугольник с указанной диагональю, и заполнить его краской  $c$ .

Если нет необходимости закрашивать прямоугольник, то следует воспользоваться командой:

$$\text{LINE } (x_a, y_a) - (x_c, y_c), c, b$$

### Исполнитель CIRCLE



Чтобы изобразить окружность на листе бумаги, мы используем циркуль. Ножку с иглой устанавливаем неподвижно в нужном месте листа бумаги и вращаем вокруг нее ножку с грифелем. Конец ножки с грифелем описывает замкнутую линию, которая называется окружностью.

Каждая окружность имеет свой центр и радиус. Центр окружности — это точка, в которой располагалась ножка с иглой. Радиус — отрезок, соединяющий центр окружности и любую ее точку.

Вычерчивать окружности на экране нам поможет исполнитель по имени CIRCLE.

Этот исполнитель умеет рисовать окружности любых размеров и в любом месте экрана. Надо только правильно сообщать ему все необходимые данные — координаты центра и длину радиуса.

Вот общий вид приглашения к работе этого исполнителя:

$$\text{CIRCLE } (x, y), r, c$$

Здесь:

CIRCLE — имя исполнителя (оно должно быть написано без ошибок);

$(x, y)$  — координаты центра окружности;

$r$  — длина радиуса;

$c$  — номер цвета окружности.



Если вы не укажете номер цвета рисования (умолчите об этом), то по умолчанию окружность будет нарисована белым цветом.

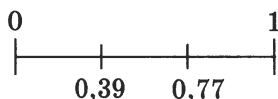
### Случайные числа

Ваши программы будут гораздо интереснее, если внести в них элемент случайности.

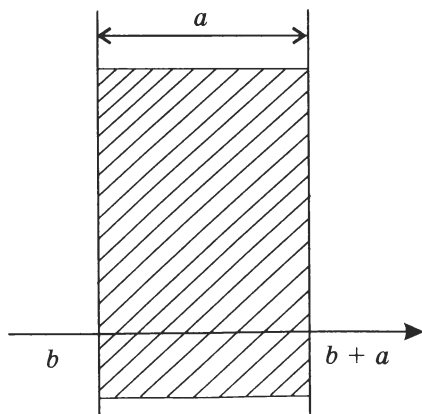
Дело в том, что у компьютера есть специальная функция, которая выдает случайные числа. Ее обозначение RND получено сокращением английского слова «random», которое означает «случайный».

Когда бросается игральная кость, то заранее не известно, какое число выпадет. Но в любом случае это будет число от 1 до 6.

При помощи функции RND получается произвольное (случайное) число от 0 до 1.



Чтобы получить случайные числа из произвольного промежутка, используют выражение  $\text{RND} * a + b$ , где  $a$  — ширина промежутка,  $b$  — значение левой границы.



Для того чтобы компьютер «задумывал» произвольные координаты  $x$  и  $y$ , принадлежащие нашему графическому экрану, воспользуемся следующими программными строками:

```
x = 639*RND
y = 463*RND
```

Здесь числа 639 и 463 определяют верхние границы выдаваемых значений.

Рассмотрим пример. Пусть  $(x, y)$  — координаты некоторой точки на экране, «задуманные» компьютером.  $(x+5, y+5)$  — координаты точки, находящейся неподалеку. Нарисуем «косой дождик»:

```
SCREEN 12
10 x = RND*639
   y = RND*463
   LINE (x, y) - (x+5, y+5)
   GOTO 10
```

В результате выполнения этой программы весь экран покрывается «косыми дождинками». В последней строке программы мы отдаем приказ вернуться на строку с меткой 10. Компьютер снова и снова «задумывает» координаты точки и чертит отрезок-«дождинку» на экране. Рисование будет продолжаться до тех пор, пока вы не прервете выполнение программы.

---

# Компьютерный практикум<sup>1</sup>


---

### Работа 1. Работаем с файлами и папками. Часть 1

1. Займите свое рабочее место за компьютером.
2. Откройте двойным щелчком левой кнопкой мыши окно *Мой компьютер*.
3. Убедитесь, что в меню *Вид* включен флажок *Крупные значки*.
4. Дважды щелкните на значке жесткого диска C: и рассмотрите его содержимое.
5. Найдите папку Мои документы и откройте ее двойным щелчком на соответствующем значке.
6. В папке Мои документы найдите файл Начало.doc. Откройте его двойным щелчком. Прочтите и выполните содержащееся в файле указание (закройте окно щелчком на закрывающей кнопке).
7. Найдите на Рабочем столе папку Мои документы и откройте ее.
8. Откройте папку бкласс, вложенную в папку Мои документы. Рассмотрите ее содержимое.

---

<sup>1</sup> Компьютерный практикум может проводиться в операционных системах Windows, Linux и MacOS с использованием пакетов Microsoft Office и OpenOffice.org. Подробное описание работ для OpenOffice.org содержится на CD с программно-методической поддержкой и размещено на сайте методической поддержки издательства.

9. Откройте папку *Заготовки*, вложенную в папку *бкласс*. Выполните команду [*Вид-Таблица*]. Ознакомьтесь с информацией, приведенной для каждого файла.
10. Расположите значки файлов в алфавитном порядке. Для этого установите указатель мыши в чистую область окна *Заготовки* и щелкните правой кнопкой мыши (вызов контекстного меню). Выполните команду [*Упорядочить значки-по имени*].
11. Расположите значки файлов в порядке возрастания размеров файлов ([*Упорядочить значки-по размеру*]).
12. Расположите значки по типам файлов.
13. В папке *бкласс* создайте собственную папку, в которой будут храниться все ваши работы. Для этого переведите указатель мыши в чистую область окна *бкласс* и щелкните правой кнопкой мыши (вызов контекстного меню). Выполните команду [*Создать-Папку*] и в качестве имени новой папки введите свою фамилию. Убедитесь, что фамилия написана правильно, и нажмите клавишу {Enter}.
14. Поднимитесь в папку *Мои документы* (кнопка ). Закройте окно папки *Мои документы*.



### Теперь мы умеем


- открывать и закрывать папки;
- упорядочивать содержимое папки — файлы и вложенные папки;
- создавать папки.

## Работа 2. Знакомимся с текстовым процессором

### Задание 1

1. Откройте текстовый процессор.

2. Найдите строку заголовка, строку меню, строку состояния. С помощью меню *Вид* узнайте, какие панели инструментов установлены. Уберите все лишние панели, оставив только панели *Стандартная* и *Форматирование*. Вспомните назначение уже знакомых вам кнопок и выясните назначение новых кнопок, списков и полей этих панелей.
3. Откройте документ Ошибка.doc (Мои документы\бкласс\Заготовки). Для этого:
  - 1) в меню *Файл* выберите пункт *Открыть* (или щелкните на кнопке *Открыть* на панели *Стандартная*);
  - 2) в появившемся окне *Открытие документа*, последовательно открывая папки, выберите ту, в которой находится документ;
  - 3) двойным щелчком мышью откройте нужный документ.
4. Обратите внимание на то, что некоторые слова и предложения подчеркнуты красными и зелеными волнистыми линиями. Красная линия говорит о том, что в слове, скорее всего, допущена ошибка или же его нет в компьютерном словаре. Зеленая линия говорит о том, что в предложении неверно расставлены знаки препинания. Попробуйте самостоятельно устранить выявленные текстовым процессором ошибки.
5. Сохраните исправленный документ в собственной папке под именем Проверка1. Для этого:
  - 1) в меню *Файл* выберите пункт *Сохранить как ...*;
  - 2) в появившемся окне *Сохранение документа*, последовательно открывая папки, выберите ту, в которой следует сохранить документ;
  - 3) двойным щелчком мыши откройте нужную папку;
  - 4) введите имя документа в поле *Имя файла*;
  - 5) щелкните на кнопке *Сохранить*.
6. Повторно откройте документ Ошибка.doc.

7. Запустите проверку документа с помощью команды *Правописание* меню *Сервис* или кнопки *Правописание*  панели *Стандартная*.

Внимательно анализируйте информацию, появляющуюся в диалоговом окне *Правописание*, и с помощью соответствующих командных кнопок вносите изменения или пропускайте помеченные слова.


8. Сохраните исправленный документ в собственной папке под именем Проверка2 и закройте программу.

## Задание 2

1. Откройте текстовый процессор.
2. На горизонтальной линейке установите маркер отступа первой строки на 1 см.
3. Установите шрифт Arial, размер шрифта 14, выравнивание по ширине. Введите следующий текст:

Текст — это связанное по смыслу и грамматически высказывание в устной или письменной форме. Основными признаками текста являются связность и цельность. Предложения, входящие в текст, расположены в определенном порядке. Это создает связность текста. Цельность тексту придает смысловое единство. Каждый текст включает в себе определенное содержание, то есть имеет свою тему. Предложения в тексте объединяются не только темой, но и основной идеей. Несколько предложений, выражающих одну мысль, оформляют отдельным абзацем.

4. С помощью кнопки *Правописание* проверьте, нет ли ошибок в набранном вами тексте.
5. Разбейте текст на абзацы. Для этого с помощью клавиш управления курсором (клавиши со стрелками) или мыши помещайте курсор в нужное место (конец абзаца) и нажимайте клавишу {Enter}.

С помощью кнопки *Непечатаемые символы*  отобразите на экране знаки форматирования и убедитесь, что символы, отмечающие концы абзацев, стоят в нужных местах. Убедитесь, что все абзацы начинаются с красной строки.

Текст — это связанное по смыслу и грамматически высказывание в устной или письменной форме. Основными признаками текста являются связность и цельность.

Предложения, входящие в текст, расположены в определенном порядке. Это создает связность текста.

Цельность тексту придает смысловое единство. Каждый текст заключает в себе определенное содержание, то есть имеет свою тему.

Предложения в тексте объединяются не только темой, но и основной идеей. Несколько предложений, выражающих одну мысль, оформляют отдельным абзацем.

6. Вспомните, как выделяются отдельные слова и строки. Завершите форматирование текста по следующему образцу:

Текст – это связанное по смыслу и грамматически высказывание в устной или письменной форме. Основными признаками текста являются **связность** и **цельность**.

Предложения, входящие в текст, расположены в определенном порядке. Это создает связность текста.

Цельность тексту придает *смысловое единство*. Каждый текст заключает в себе определенное **содержание**, то есть имеет свою тему.

Предложения в тексте объединяются не только темой, но и основной идеей. Несколько предложений, выражающих одну мысль, оформляют отдельным абзацем.

7. Сохраните файл в собственной папке под именем Текст и закройте программу.




### Теперь мы умеем

- запускать текстовый процессор;
- открывать, изменять и сохранять документы в текстовом процессоре;
- использовать команду проверки правописания в текстовом процессоре;
- устанавливать абзацный отступ и разбивать текст на абзацы в текстовом процессоре;
- выделять фрагмент текста (произвольный участок, строку, слово, абзац) и изменять начертание шрифта в текстовом процессоре;
- завершать работу с текстовым процессором.

## Работа 3. Редактируем и форматируем текст. Создаем надписи

### Задание 1

1. Откройте текстовый процессор MS Word.
2. С помощью меню *Вид* установите панель *Рисование*.
3. На панели *Рисование* найдите кнопку *Вставить объект WordArt*  и с ее помощью вызовите коллекцию *WordArt*, содержащую различные варианты надписей.
4. Щелкните мышью на понравившейся вам надписи, затем — на кнопке *OK*.
5. Рассмотрите раскрывшееся диалоговое окно *Изменение текста WordArt*. В нем можно выбирать шрифт, его размер и начертание. Нажав клавишу {Delete}, очистите рабочее поле.



6. Введите текст надписи «ШКОЛА» и щелкните на кнопке *ОК*. На экране появится созданная вами красочная надпись.



7. Самостоятельно создайте еще три варианта надписей.
8. Созданные надписи можно перемещать, удалять, изменять. Для этого надпись нужно выделить. Выделите одну из надписей — поместите на нее указатель мыши и выполните щелчок левой кнопкой мыши. Надпись выделена, если вокруг нее появилась рамочка с квадратиками. Чтобы снять выделение, можно щелкнуть в любом месте вне надписи.
9. Измените расположение надписей на экране. Для этого:
- 1) выделите произвольную надпись;
  - 2) удерживая нажатой левую кнопку мыши, перетащите надпись в другое место.
10. Оставьте на экране самую удачную надпись, а все другие удалите. Для этого:
- 1) выделяйте надписи;
  - 2) нажимайте клавишу {Delete}.
11. Выделите надпись. Обратите внимание на появившуюся панель *WordArt*. С ее помощью можно полностью изменить исходную надпись. Попробуйте это сделать самостоятельно.



12. Сохраните файл в собственной папке под именем Школа и закройте программу.

## **Задание 2**

1. Откройте текстовый процессор.
2. Откройте файл Заготовка.doc (Мои документы\бкласс\Заготовки):

Жили у бабуси  
Два веселых гуся.  
Один серый, другой белый —  
Два веселых гуся.  
Мыли гуси лапки  
В луже у канавки.  
Вытянули шеи —  
У кого длиннее?  
Вот кричит бабуся:  
«Ой, пропали гуси!»  
Выходили гуси,  
Кланялись бабусе.

3. Используя кнопки *Копировать* и *Вставить*, поместите фрагмент «Один серый, другой белый — два веселых гуся» в нужные места текста так, чтобы текст стихотворения стал полным.
4. Выделите весь текст. Задайте для него следующие параметры форматирования: отступ первой строки 5 см; шрифт Arial; размер шрифта 12; выравнивание по левому краю. Отмените выделение текста.
5. Создайте красочную надпись «ГУСИ» и поместите ее над текстом стихотворения.
6. Сохраните файл в собственной папке под именем Гуси и закройте программу.




## Теперь мы умеем

- создавать, изменять и перемещать красочные надписи в текстовом процессоре;
- копировать и вставлять фрагменты текста в текстовом процессоре;
- форматировать фрагмент текста в текстовом процессоре.

## Работа 4. Нумерованные списки

### Задание 1

1. Откройте текстовый процессор MS Word.
2. Включите нумерацию, щелкнув на кнопке *Нумерация*  .
3. Наберите текст (для перехода к каждому следующему пункту надо нажимать клавишу {Enter}):

1. Записать слово, разделив его на фонетические слоги.
2. Указать количество слогов, выделить ударный.
3. Дать характеристику звуков — гласных и согласных.
4. Указать количество букв и звуков в слове.

4. Отмените продолжение списка, два раза нажав клавишу {Enter}.
5. Создайте красочную надпись «ФОНЕТИЧЕСКИЙ РАЗБОР» и поместите ее над текстом.
6. Сохраните файл в собственной папке под именем Фонетика и закройте программу.

## **Задание 2**

1. Откройте текстовый процессор.
2. Откройте файл Слова.doc (Мои документы\6класс\Заготовки):

Число; столица России; учебник; наука; житель Москвы; столица; самая высокая гора; А. С. Пушкин; сказочный персонаж; Змей Горыныч.

3. Из приведенных слов сформируйте два нумерованных списка — общие понятия и единичные понятия.
4. Заголовок каждого списка оформите в виде красочной надписи.
5. Сохраните файл в собственной папке под именем Понятия и закройте программу.

## **Задание 3**

1. Откройте текстовый процессор.
2. Откройте файл Кувшин.doc (Мои документы\6класс\Заготовки):

Есть 2 кувшина емкостью 3 и 8 литров. Как с помощью только этих кувшинов набрать из реки 7 литров воды?

3. Внимательно изучите условие и продумайте решение задачи. Возможный план действий оформите в виде нумерованного списка.
4. Сохраните файл в собственной папке под именем Переливания и закройте программу.



**Теперь мы умеем**


- создавать нумерованные списки в текстовом процессоре.

## Работа 5. Маркированные списки

### Задание 1

1. Откройте текстовый процессор MS Word и наберите текст:

Признаки культурной речи:

2. Перейдите на новую строку. Щелкните на кнопке *Маркеры*  и наберите текст (для перехода к каждому следующему пункту надо нажимать клавишу {Enter}):

- правильность;
- чистота;
- точность;
- выразительность;
- логичность;
- уместность;
- богатство.

3. Отмените продолжение списка, два раза нажав клавишу {Enter}.
4. Отформатируйте заголовок маркированного списка, придав ему полужирное начертание.
5. Выделите весь маркированный список (кроме заголовка), щелкните на кнопке *Нумерация* и проследите за изменениями, произошедшими со списком. Верните списку первоначальный вид.
6. Сохраните файл в собственной папке под именем Речь и закройте программу.

## Задание 2

1. Откройте текстовый процессор.
2. Откройте текстовый файл Чудо (Мои документы\бкласс\Заготовки):

Семь чудес света: Великая пирамида в Гизе; Висячие сады Вавилона; Храм Артемиды в Эфесе; Статуя Зевса в Олимпии; Мавзолей в Галикарнасе; Колосс Родосский; Александрийский маяк.

3. Оформите имеющийся текст в виде маркированного списка.
4. Выделите весь маркированный список (кроме заголовка). Выберите команду *Список* меню *Формат*. На вкладке *Маркированный* диалогового окна *Список* выберите наиболее подходящий, по вашему мнению, вид маркера и щелкните на кнопке *ОК*. Проследите за изменениями, произошедшими со списком.
5. Сохраните файл в собственной папке под тем же именем и закройте программу.

## Задание 3

1. Откройте текстовый процессор.
2. Откройте текстовый файл Природа (Мои документы\бкласс\Заготовки).
3. Выберите в тексте информацию о том, какие семейства входят в отряд хищных. Лишнюю информацию удалите.
4. Оформите оставшуюся информацию в виде маркированного списка.
5. Сохраните файл в собственной папке под именем Хищники и закройте программу.

## Задание 4

1. Откройте текстовый процессор.
2. Откройте текстовый файл Делитель (Мои документы)\бкласс\Заготовки):

Чтобы найти наибольший общий делитель нескольких натуральных чисел, надо: разложить их на простые множители; из множителей, входящих в разложение одного из этих чисел, вычеркнуть те, которые не входят в разложение других чисел; найти произведение оставшихся множителей.

3. По своему усмотрению оформите имеющийся текст в виде маркированного или нумерованного списка. Будьте готовы обосновать свой выбор.
4. Сохраните файл в собственной папке под тем же именем и закройте программу.




### Теперь мы умеем

- создавать маркированные списки в текстовом процессоре;
- изменять вид маркера;
- изменять вид списка.

## Работа 6. Создаем таблицы

### Задание 1

1. Откройте текстовый процессор MS Word.
2. С помощью команды [*Файл-Параметры страницы-Размер бумаги*] выберите альбомную ориентацию страницы.
3. С помощью кнопки *Добавить таблицу*  задайте таблицу, состоящую из двух строк и пяти столбцов.

4. В ячейки верхней строки внесите названия первых пяти дней недели: понедельник, вторник, среда, четверг, пятница.
5. В соответствующие ячейки второй строки впишите ваше расписание уроков. Расписание на каждый день оформите в виде нумерованного списка.
6. Выделите таблицу и оформите ее, используя автоформат *Сетка 8* ([*Таблица-Автоформат*]).
7. В верхней части страницы создайте красочную надпись «РАСПИСАНИЕ УРОКОВ».
8. Сохраните файл в собственной папке под именем Расписание и закройте программу.

## Задание 2

1. Откройте текстовый процессор.
2. Откройте текстовый файл Пары.doc (Мои документы\бкласс\Заготовки):

Цветок, фрукт, овощ, буква, цифра, орудие, прибор, школьник, дом, день, лодка, животное, насекомое, металл, орган, олово, подросток, почки, морковь, микроскоп, два, военный, фиалки, сержант, топор, груши, «У», тюлень, резиновая лодка, оса, деревянный дом, зимний день.

3. Под текстом с помощью команды [*Таблица-Добавить-Таблица*] задайте таблицу, состоящую из 2 столбцов и 17 строк.
4. В ячейки верхней строки внесите словосочетания «Родовое понятие» и «Видовое понятие».
5. Из перечня имеющихся слов выделите попарно родовые и видовые понятия и запишите их в соответствующие столбцы таблицы.
6. Выделите таблицу и оформите ее, используя автоформат *Современный* ([*Таблица-Автоформат*]).
7. В верхней части страницы создайте красочную надпись «РОД — ВИД».



8. Сохраните файл в собственной папке под именем **Отношение** и закройте программу.

### **Задание 3**

1. Откройте текстовый процессор.
2. Откройте текстовый файл **Семь чудес света.doc** (Мои документы\6класс\Заготовки):

Кто не слышал о Египетских пирамидах, Висячих садах Вавилона, Колоссе Родосском, Статуе Зевса в Олимпии, Мавзолее в Галикарнасе, Александрийском маяке или Храме Артемиды в Эфесе? Эти замечательные памятники древности, известные всем как семь чудес света, приводили в восторг современников. Но не только в те далекие времена зодчие, скульпторы и художники создавали рукотворные чудеса.

В Средние века были созданы такие памятники, как Пещера Десяти тысяч Будд (Китай), Большое Зимбабве, храм Ангкор Ват (Камбоджа), замок Крак де Шевалье (Сирия), Солсберийский собор (Англия), крепость Альгамбра (Испания), столица ацтеков Теночтитлан. Неустанно работая и делая новые открытия, человек создал много уникальных творений, которые можно назвать чудесами нашего времени.

3. Под текстом создайте таблицу, состоящую из 3 столбцов и 8 строк.
4. В ячейки верхней строки внесите словосочетания «Древний мир», «Средние века», «Наше время».
5. Выберите в тексте и внесите в соответствующие ячейки таблицы названия сооружений, считающихся чудесами света для каждого исторического периода (чудеса нашего времени напишите по своему усмотрению).

6. Выделите таблицу и оформите ее, используя автоформатирование по своему усмотрению.
7. В верхней части страницы создайте красочную надпись «СЕМЬ ЧУДЕС СВЕТА».
8. Сохраните файл в собственной папке под именем Чудеса и закройте программу.

#### **Задание 4**

1. Откройте текстовый файл Солнечная система.doc (Мои документы\6класс\Заготовки):

#### **Планеты Солнечной системы**

Расстояние от Юпитера до Солнца – 778 млн км. Расстояние от Урана до Солнца – 2870 млн км. Диаметр планеты Юпитер – 142 800 км. Диаметр планеты Сатурн – 120 860 км. Расстояние от Сатурна до Солнца – 1 427 млн км. Диаметр планеты Уран – 52 000 км. Расстояние от Земли до Солнца – 150 млн км. Расстояние от Плутона до Солнца – 5950 млн км. Диаметр планеты Меркурий – 4880 км. Расстояние от Нептуна до Солнца – 4497 млн км. Время обращения Сатурна вокруг Солнца – 29,5 лет. Диаметр планеты Плутон – 3000 км. Расстояние от Марса до Солнца – 228 млн км. Диаметр планеты Нептун – 48 400 км. Время обращения Урана вокруг Солнца – 84 года. Время обращения Нептуна вокруг Солнца – 165 лет. Время обращения Юпитера вокруг Солнца – 12 лет. Расстояние от Меркурия до Солнца – 58 млн км. Время обращения Земли вокруг Солнца – 365 дней. Время обращения Меркурия вокруг Солнца – 88 дней. Диаметр планеты Марс – 6790 км. Время обращения Венеры вокруг Солнца – 225 дней. Диаметр планеты Земля – 12 756 км. Диаметр планеты Венера – 12 100 км. Время обращения Плутона вокруг Солнца – 248 лет. Расстояние от Венеры до Солнца – 108 млн км. Время обращения Марса вокруг Солнца – 687 дней.

2. Под текстом создайте таблицу, состоящую из 4 столбцов и 10 строк следующего вида:

Планета	Диаметр (км)	Расстояние от Солнца (млн км)	Время обращения вокруг Солнца

3. Заполните таблицу на основании информации из текста.
4. Выделите таблицу и оформите ее, используя автоформатирование по своему усмотрению.
5. В меню *Вид* с помощью команды *Панели инструментов* выберите панель *Таблицы и границы*. Найдите на этой панели кнопки *Сортировка по возрастанию* и *Сортировка по убыванию*.
6. Отсортируйте строки по возрастанию диаметров планет. Для этого переместите указатель мыши в ячейку с надписью «Диаметр (км)» и щелкните на кнопке *Сортировка по возрастанию*. Проследите за изменениями в таблице.
7. Отсортируйте строки по убыванию расстояния от планет до Солнца.
8. Выстройте строки в алфавитном порядке названий планет.
9. С помощью приложения *Калькулятор* (*[Пуск-Программы-Стандартные-Калькулятор]*) выразите данные последнего столбца в днях и выполните их сортировку по возрастанию. Считайте 1 год равным 365 дням.
10. Сохраните файл в собственной папке под именем *Планеты* и закройте программу.








**Теперь мы умеем**

- изменять ориентацию страницы;
- создавать таблицу, состоящую из требуемого числа столбцов и строк;
- оформлять таблицу, используя автоформат.

## Работа 7. Размещаем текст и графику в таблице

1. Откройте текстовый процессор MS Word.
2. Откройте текстовый файл Загадки.doc (Мои документы\бкласс\Заготовки).
3. Под текстом создайте таблицу, состоящую из 2 столбцов и 6 строк.
4. Загадки разместите в ячейках таблицы «лесенкой». Переносите нужные фрагменты текста следующим способом:
  - 1) выделите нужный фрагмент;
  - 2) нажмите левую кнопку мыши внутри выделенной области и перетащите фрагмент в нужное место;
  - 3) отпустите кнопку мыши.
5. В ячейку таблицы, рядом с первой загадкой, поместите картинку-разгадку из коллекции *ClipArt*. Для этого:
  - 1) поместите указатель мыши в нужную ячейку и выполните команду [*Вставка-Рисунок-Картинки*];
  - 2) в поле ввода *Искать (Найти клипы)* введите ключевое слово *книги*. Для запуска поиска щелкните на кнопке *Начать* (нажмите клавишу *Enter*);
  - 3) выберите щелчком подходящий рисунок (щелкните на появившейся кнопке *Вставить клип*).
6. Щелкните левой кнопкой мыши внутри области рисунка — вокруг рисунка появится восемь черных прямоугольников — маркеров. Перетаскивая маркеры, придайте рисунку размеры по вашему усмотрению.
7. Повторите п. 5–6 для оставшихся загадок (соответствующие рисунки находятся по ключевым словам *Транспортные средства*).
8. В верхней части страницы создайте красочную надпись «ЗАГАДКИ». Получившийся документ должен иметь примерно такой вид:

# Загадки

<p>Страну чудес откроем мы И встретимся с героями В строчках, на листочках, Где станции на точках.</p>	
	<p>Три глаза — три приказа, Красный — самый опасный.</p>
<p>Я в любое время года И в любую непогоду Очень быстро в час любой Провезу вас под землей.</p>	
	<p>Едет конь стальной, рычит, Сзади плуги волочит.</p>
<p>Что за чудо синий дом, Окна светлые кругом, Носит обувь из резины И питается бензином.</p>	
	<p>В поле лестница лежит, Дом по лестнице бежит.</p>

9. Сохраните файл в собственной папке под именем Отгадки и закройте программу.



## Теперь мы умеем

- перемещать фрагмент текста в заданную ячейку таблицы;
- находить нужный рисунок в коллекции ClipArt;
- вставлять рисунок из коллекции ClipArt в требуемое место;
- придавать рисунку из коллекции ClipArt размеры по своему усмотрению.

## Работа 8. Строим диаграммы

### Задание 1

1. Откройте текстовый процессор.
2. Откройте текстовый файл Задача1.doc (Мои документы\бкласс\Заготовки):

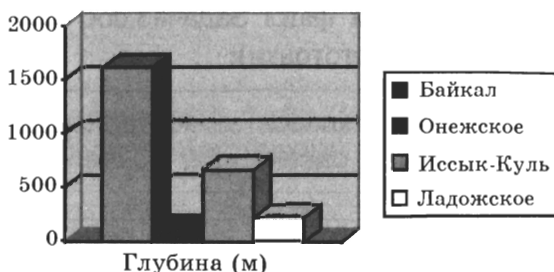
Наибольшая глубина озера Байкал – 1620 м, Онежского озера – 127 м, озера Иссык-Куль – 668 м, Ладожского озера – 225 м.

3. На основании этой информации создайте и заполните следующую таблицу:

Озеро	Глубина (м)
Байкал	1620
Онежское	127
Иссык-Куль	668
Ладожское	225

4. Выделите таблицу и щелкните на кнопке *Добавить диаграмму*. Появится столбчатая диаграмма, показывающая глубину перечисленных озер, а также еще одна таблица с исходными данными.

### Глубина озер



5. Выполните команду [Диаграмма-Параметры диаграммы]. В открывшемся диалоговом окне выберите вкладку *Заголовки*, в соответствующее поле ввода введите название диаграммы «Глубина озер»; на вкладке *Легенда* установите флажок *Добавить легенду* и активизируйте по своему усмотрению один из переключателей, задающий ее расположение. Щелкните на кнопке *ОК*.
6. Установите указатель мыши вне области новых объектов и выполните щелчок левой кнопкой мыши.
7. Сохраните файл в собственной папке под именем Глубина и закройте программу.

### Задание 2

1. Откройте текстовый файл Задача2.doc (Мои документы\бкласс\Заготовки):

Площадь России равна 17,1 млн км<sup>2</sup>, площадь Китая – 9,6 млн км<sup>2</sup>, площадь Индии – 3,3 млн км<sup>2</sup> и площадь США – 9,4 млн км<sup>2</sup>.

2. Оформите приведенные данные в виде таблицы (см. п. 3 задания 1).
3. По таблице постройте столбчатую диаграмму (см. п. 4–6 задания 1).
4. Сохраните файл в собственной папке под именем Площадь и закройте программу.

### Задание 3

- Откройте текстовый файл Задача3.doc (Мои документы\бкласс\Заготовки):

Из 27 учащихся класса за контрольную работу 9 человек получили оценку «5», 15 человек – «4» и 3 человека – «3».

- На основании этой информации создайте и заполните следующую таблицу:

Оценка	«5»	«4»	«3»
Количество учеников	9	15	3

- По таблице постройте столбчатую диаграмму:



- Повторно выделите таблицу, щелкните на кнопке *Добавить диаграмму* и в меню *Диаграмма* выполните команду [*Тип диаграммы-Круговая*].
- В меню *Диаграмма* выберите команду *Параметры диаграммы*. В открывшемся окне выберите вкладку *Подписи данных*, в группе *Подписи значений* активизируйте переключатель *Доля*.





6. Сохраните файл в собственной папке под именем Оценки и закройте программу.

#### Задание 4

1. Откройте текстовый файл Задача4.doc (Мои документы\6класс\Заготовки):

Тихий океан имеет площадь 179 млн км<sup>2</sup>, Атлантический – 93 млн км<sup>2</sup>, Индийский – 75 млн км<sup>2</sup> и Северный Ледовитый – 13 млн км<sup>2</sup>.

2. На основании этой информации создайте и заполните следующую таблицу:

Океан	Тихий	Атлантический	Индийский	Северный Ледовитый
Площадь (млн км <sup>2</sup> )	179	93	75	13

3. По таблице постройте круговую диаграмму «Площади океанов». Предусмотрите вывод названия диаграммы, легенды и выраженный в процентах вклад каждого океана в воды Мирового океана (команда *Параметры диаграммы*).
4. Сохраните файл в собственной папке под именем Океаны и закройте программу.

#### Задание 5

1. Откройте текстовый файл Задача5.doc (Мои документы\6класс\Заготовки):

Суша на Земле распределена следующим образом: леса занимают площадь 57 млн км<sup>2</sup>, степи – 24 млн км<sup>2</sup>, тундры, пустыни и болота – 54 млн км<sup>2</sup> и пашня – 15 млн км<sup>2</sup>.

2. На основании этой информации постройте диаграмму. Тип диаграммы выберите самостоятельно.
3. Сохраните файл в собственной папке под именем Суша и закройте программу.



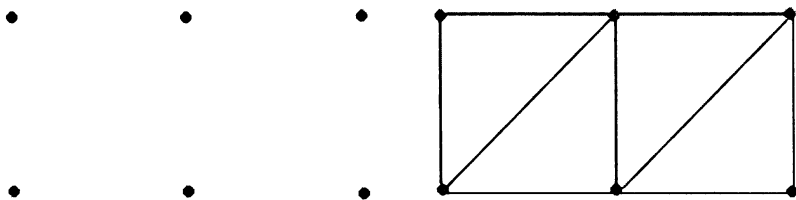
### Теперь мы умеем

- создавать диаграмму по таблице, верхняя строка и левый столбец которой содержат подписи, а остальные ячейки заполнены числами;
- устанавливать параметры диаграммы в диалоговом окне;
- изменять тип диаграммы.

## Работа 9. Изучаем графический редактор Paint

### Задание 1

1. Запустите графический редактор Paint ([*Пуск-Программы-Стандартные-Paint*]).
2. Откройте файл Головоломка.bmp (Мои документы\бкласс\Заготовки).
3. С помощью инструмента *Линия* при нажатой клавише {Shift} попытайтесь «без отрыва руки» соединить все точки так, как это показано на рисунке:




При необходимости используйте команду [*Правка-Отменить*].

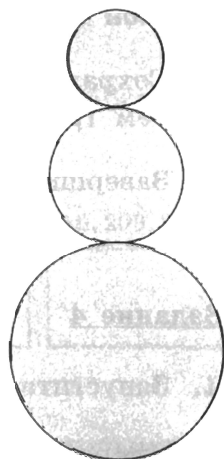
4. Сохраните результат работы под тем же именем, но в собственной папке.
5. Завершите работу с графическим редактором.

## Задание 2

1. Запустите графический редактор.
2. С помощью пункта *Атрибуты* меню *Рисунок* задайте рабочую область шириной 20 и высотой 15 см.
3. Изобразите снеговика, состоящего из трех разных по размеру кругов (инструмент *Эллипс*, с нажатой клавишей {Shift}).

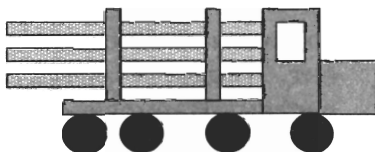
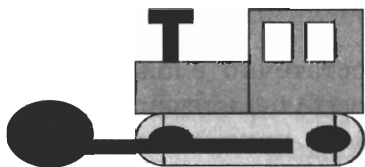
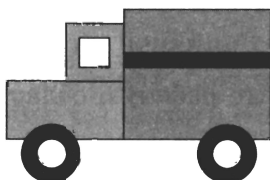
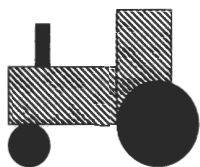
*Внимание!* Снеговик — достаточно сложный объект. Сложные объекты желательно изображать по частям. Нарисуйте каждый из кругов для снеговика отдельно. Поочередно выделите второй и третий круги (инструмент *Выделение*, режим *Прозрачный фрагмент* ) и перетащите их в нужные места.

4. Сделайте столько копий полученного рисунка, чтобы после заливки кругов голубым и синим цветами все рисунки были раскрашены по-разному.
5. С помощью инструмента *Надпись* в свободной части рабочей области укажите, сколько различных вариантов окрашивания снеговиков вам удалось придумать.
6. При наличии времени дополните рисунок по своему усмотрению.
7. Сохраните рисунок в собственной папке под именем *Снеговика*.
8. Завершите работу с графическим редактором.



### Задание 3

1. Запустите графический редактор.
2. Изобразите один из следующих рисунков (вспомните, как следует рисовать сложные объекты).




3. При наличии времени придумайте и изобразите свой вариант транспортного средства.
4. Сохраните рисунок в собственной папке под именем Транспорт.
5. Завершите работу с графическим редактором.

### Задание 4

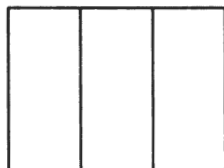
1. Запустите графический редактор.
2. Откройте файл Клоуны.bmp (Мои документы\бкласс\Заготовки).



3. Раскрасьте черно-белого клоуна так, как раскрашен разноцветный клоун. Рекомендуемая последовательность действий:
- 1) активизируйте инструмент *Выбор цвета*  ;
  - 2) щелкните на объекте, цвет которого следует скопировать;
  - 3) с помощью инструмента *Заливка* закрасьте нужную область.
4. Сохраните результат работы под тем же именем, но в собственной папке.

### Задание 5

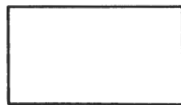
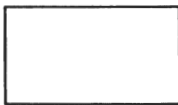
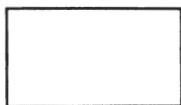
1. Откройте файл Флаги.bmp (Мои документы\6класс\Заготовки).



250, 120, 180

125, 250, 120

10, 200, 250

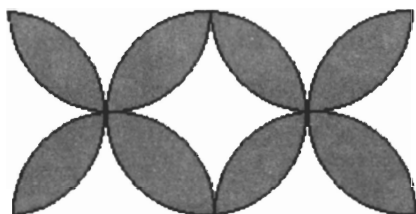


2. Каждый из трех горизонтальных прямоугольников заполните цветом, красная, зеленая и синяя составляющие которого имеют указанные числовые значения (см. эксперимент на стр. 25 учебника).

3. Подумайте, сколько разных трехцветных флагов можно составить, используя данные цвета. Размножьте заготовку флага и изобразите все придуманные вами варианты.
4. Сохраните результат работы под тем же именем, но в собственной папке.

### Задание 6

1. Откройте файл Лепестки.bmp (Мои документы\бкласс\Заготовки).

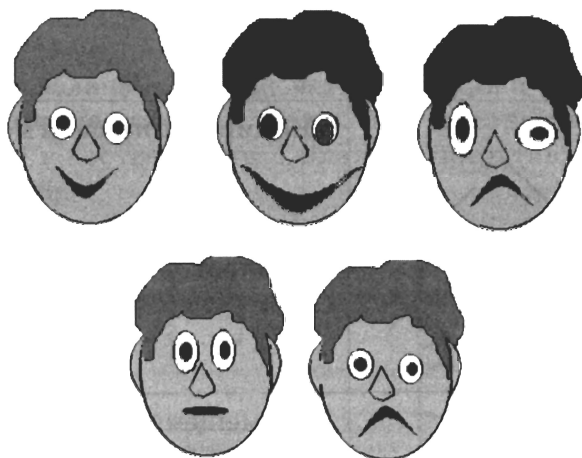


2. Цвет контура четырех центральных лепестков замените на синий. Используйте возможности «цветного ластика». Для этого:
  - 1) с помощью инструмента *Выбор цвета* сделайте цвет контура основным;
  - 2) сделайте синий цвет фоновым;
  - 3) выберите инструмент *Ластик*; нажмите правую кнопку мыши и, не отпуская ее, пройдите *Ластиком* линии рисунка, цвет которых следует заменить.
3. Сделайте цвет контура четырех боковых лепестков зеленым.
4. Сохраните результат работы под тем же именем, но в собственной папке.
5. Завершите работу с графическим редактором.

### Задание 7

1. Запустите графический редактор.

- Используя все известные вам инструменты, нарисуйте черным контуром на белом фоне забавную физиономию.
- Сделайте несколько копий полученного рисунка.
- Внесите изменения в отдельные детали рисунка, так чтобы каждая физиономия имела свое выражение. Для этого выделяйте нужные фрагменты (инструмент *Выделение*) и применяйте команды *Отразить/повернуть* и *Растянуть/наклонить* меню *Рисунок*. Мелкие изменения можно вносить, закрашивая отдельные пиксели после увеличения рисунка (инструмент *Масштаб*).



- Раскрасьте все рисунки и сохраните их в собственной папке под именем *Шутка*.
- Завершите работу с графическим редактором.




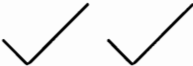


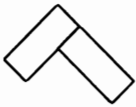

### Теперь мы умеем

- создавать сложные объекты из фрагментов;
- копировать цвет заданной области;
- получать цвет по кодам составляющих его цветов;
- изменять цвет части существующей линии.

## Работа 10. Планируем работу в графическом редакторе

### Задание 1

1. Запустите графический редактор.
2. Задайте рабочую область шириной 20 и высотой 15 см.
3. Заполните всю рабочую область экрана паркетом «елочка», работая по следующему плану:

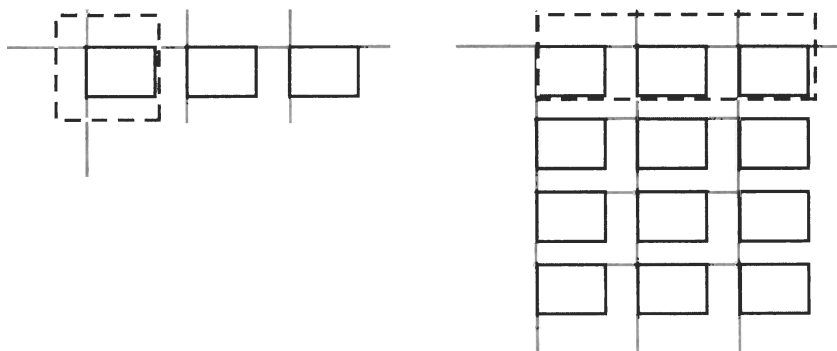
<p>Шаг 1:</p> 	<p>Инструмент <i>Линия</i></p>
<p>Шаг 2:</p> 	<p>Копирование фрагмента</p>
<p>Шаг 3:</p> 	<p>Поворот фрагмента на 180°</p>
<p>Шаг 4:</p> 	<p>Совмещение двух фрагментов (прозрачный фрагмент), заливка замкнутой области</p>
<p>Шаг 5:</p> 	<p>Копирование, отражение слева направо, совмещение фрагментов</p>
<p>Шаг 6:</p> 	<p>Размножение фрагментов по всей рабочей области</p>



4. Сохраните результат работы в собственной папке под именем Паркет.
5. Завершите работу с графическим редактором.

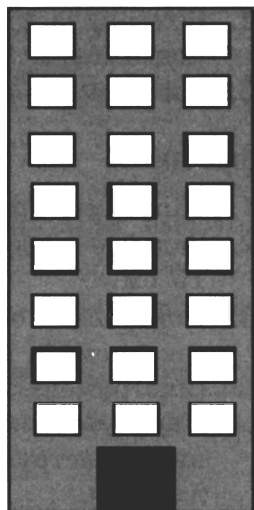
## Задание 2

1. Запустите графический редактор.
2. Задайте рабочую область шириной 20 и высотой 15 см.
3. Установите основной цвет серый и проведите пересекающиеся вертикальную и горизонтальную линии (инструмент *Линия*, с нажатой клавишей {Shift}). Нарисуйте окно-прямоугольник (основной цвет черный, инструмент *Прямоугольник*, режим «без заливки»).
4. Наша задача — нарисовать 8-этажный дом с тремя окнами на каждом этаже. Его построение выполняйте в соответствии с планом, представленным на рисунке.



**Внимание!** Работая с фрагментами, используйте следующий прием. Инструментом *Выделение* выделите указанный фрагмент. Удерживая нажатой левую кнопку мыши и клавишу {Ctrl}, перетаскивайте фрагмент в нужное место; всякий раз при отпускании кнопки мыши на экране будет возникать очередная копия фрагмента.

5. Когда все 8 этажей дома будут готовы, заключите их в большой прямоугольник и залейте его тем цветом, которым выполнялись вспомогательные линии. После этого стену дома можете перекрасить по своему вкусу.
6. Считая, что окна дома могут быть темными (черный цвет) или светиться (желтый цвет), окрасьте их так, чтобы на всех этажах сочетания темных и светлых окон были различными.
7. При наличии времени дополните рисунок по своему усмотрению.
8. Сохраните рисунок в собственной папке под именем Дом.
9. Завершите работу с графическим редактором.



### Задание 3

1. Запустите графический редактор.
2. С помощью инструментов *Эллипс* и *Кривая* изобразите цепочку из четырех одинаковых по размеру бусин.



3. Считая, что бусины могут быть только красного и синего цветов, попытайтесь придумать и изобразите на экране все возможные различные цепочки из 4 таких бусин.
4. С помощью инструмента *Надпись* в свободной части рабочей области укажите, сколько различных цепочек вам удалось придумать.

5. Сохраните рисунок в собственной папке под именем Бусины.
6. Завершите работу с графическим редактором.

#### Задание 4

1. Запустите графический редактор.
2. Проведите вспомогательный вертикальный отрезок серого цвета — ось симметрии будущей бабочки.
3. Установите основной цвет черный, фоновый — белый. С помощью инструмента *Многоугольник* (без заливки) изобразите контур крыла бабочки (начинайте и заканчивайте рисование на оси симметрии).
4. Разместите на экране четыре копии правого крыла бабочки.
5. Для каждого крыла придумайте оригинальный вариант окраски и воплотите свой замысел с помощью инструментов графического редактора.
6. Выделите прямоугольный фрагмент (режим *Прозрачный фрагмент*), содержащий одно из крыльев, строго по оси симметрии будущей бабочки. Скопируйте его (*[Правка-Копировать]*) и вставьте (*[Правка-Вставить]*). Отрадите вставленный фрагмент слева направо (команда *Отразить/Повернуть ...* из меню *Рисунок*). Совместите два крыла как можно точнее по оси симметрии.
7. Повторите пункт 6 для других крыльев.
8. В свободной части рабочего поля изобразите туловище бабочки (инструменты *Скругленный прямоугольник* и *Линия*). Выделите фрагмент, содержащий туловище бабочки, и разместите его вдоль оси симметрии каждой из бабочек.







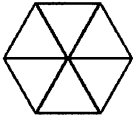

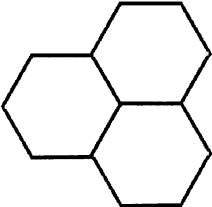
9. Разместите бабочек так, как показано на рисунке. Для этого выделяйте бабочку и отражайте ее слева направо или сверху вниз (команда *Отразить/Повернуть ...* из меню *Рисунок*).



10. При наличии времени изобразите в центре рисунка цветок.
11. Сохраните рисунок в собственной папке под именем *Бабочки*.
12. Завершите работу с графическим редактором.

### **Задание 5**

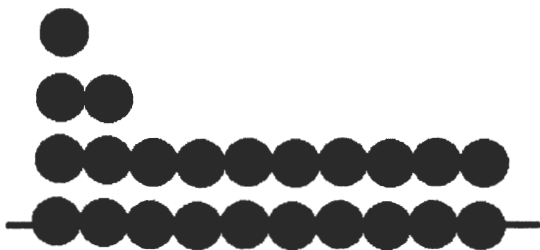
1. Запустите графический редактор.
2. Задайте рабочую область, чтобы она занимала всю видимую часть экрана.
3. Заполните рабочую область экрана «сотами», работая по следующему плану:

<p>Шаг 1:</p> 	<p>Инструмент <i>Линия</i></p>
<p>Шаг 2:</p> 	<p>Наклон по горизонтали на 30°</p>
<p>Шаг 3:</p> 	<p>Копирование и отражение фрагмента слева направо</p>
<p>Шаг 4:</p> 	<p>Совмещение двух фрагментов (<i>Прозрачный фрагмент</i>), удаление <i>Ластиком</i> лишних деталей</p>
<p>Шаг 5:</p> 	<p>Копирование, отражение сверху вниз, совмещение фрагментов</p>
<p>Шаг 6:</p> 	<p>Удаление <i>Ластиком</i> лишних деталей, заливка замкнутой области</p>
<p>Шаг 7:</p> 	<p>Размножение фрагмента по всей рабочей области</p>

4. Сохраните результат работы в собственной папке под именем Соты.
5. Завершите работу с графическим редактором.

### Задание 6

1. Рассмотрите рисунки китайских, японских и русских счетов, приведенных на странице 109 вашего учебника. Подумайте, что общего в этих счетах и чем они отличаются друг от друга.
2. Если предварительно обдумать последовательность своих действий, то в графическом редакторе достаточно быстро можно построить самые сложные объекты. Выполните рисунок русских счетов (обратите внимание, что в них на одном из стержней всего 4 косточки). Рациональный план рисования вам подскажут следующие картинки:



3. Рамку для счетов изобразите с помощью инструмента *Прямоугольник* (без заливки). Чтобы стороны прямоугольника были более широкими, предварительно установите самую большую ширину линии для инструмента *Линия*.
4. Сохраните рисунок в собственной папке под именем Счеты.
5. Завершите работу с графическим редактором.



### Теперь мы умеем

- отражать, поворачивать, наклонять и совмещать фрагменты рисунка;
- использовать вспомогательные линии;
- планировать работу по созданию рисунка.

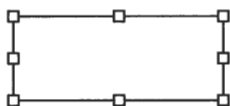
## Работа 11. Знакомимся с векторной графикой

### Задание 1

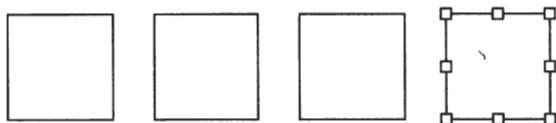
1. Откройте текстовый процессор MS Word.
2. С помощью команды [*Вид-Панели инструментов-Рисование*] выведите на экран панель *Рисование*.
3. По всплывающим подсказкам узнайте назначение каждого элемента этой панели.



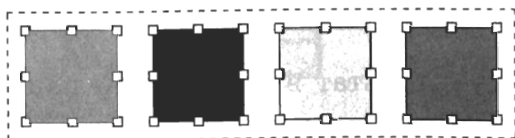
4. С помощью инструмента *Прямоугольник* нарисуйте прямоугольник:



5. Измените размеры прямоугольника; удалите прямоугольник; попытайтесь изобразить небольшой квадрат (вспомните, как строили квадраты в графическом редакторе Paint).
6. Скопируйте квадрат в буфер обмена. Разместите три копии квадрата рядом с оригиналом:



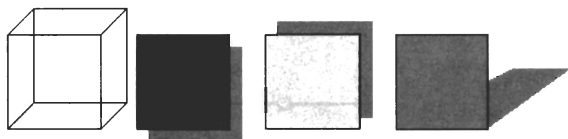
7. Поочередно выделяя каждый из квадратов, подбирайте по своему усмотрению цвет заливки и закрашивайте его.
8. Инструментом *Выбор объектов* выделите все четыре квадрата:



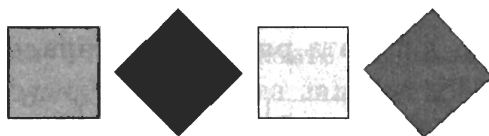
9. Скопируйте выделенный фрагмент в буфер обмена. Четыре копии фрагмента (рядов квадратов) разместите под оригиналом.
10. Измените тип и цвет линии границы каждого из четырех квадратов первого ряда так:



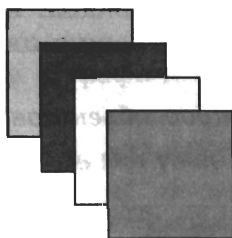
11. Поэкспериментируйте с инструментами *Тень* и *Объем* на втором ряду квадратов:



12. Опробуйте инструмент *Свободное вращение* на третьем ряду квадратов:



13. Измените положение и наложение квадратов четвертого ряда по образцу:



14. Сохраните результат в собственной папке под именем *Квадраты*.



## Задание 2

1. На чистой странице с помощью инструмента *Овал* выполните рисунок:



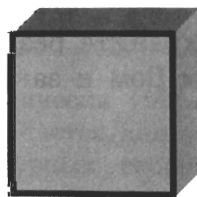
2. Сохраните результат в собственной папке под именем *Овалы*.

## Задание 3

1. Откройте файл *Домик.doc* (Мои документы\бкласс\Заготовки). На экране появится рисунок:




2. Выделите рисунок — щелкните левой кнопкой внутри области рисунка. Выполните команду [*Действия-Разгруппировать*]. Рисунок «рассыплется» на несколько составных частей, каждая из которых будет окружена белыми маркерами.
3. Щелкните вне области рисунка — все маркеры исчезнут. Поочередно щелкните на крыше, окне и стенах домика и, удерживая нажатой левую кнопку мыши, растащите их в разные стороны.



4. Из полученных частей, как из деталей конструктора, соберите большой дом.

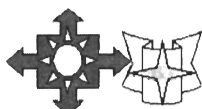


Для этого:

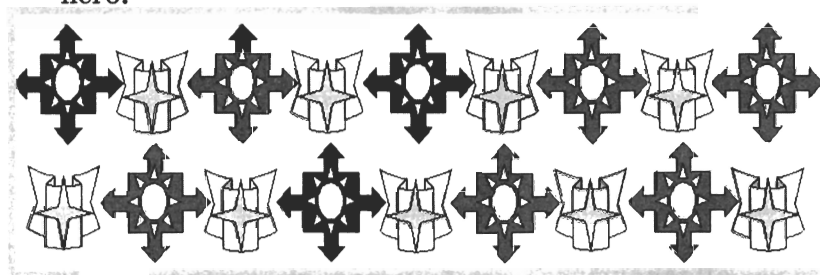
- 1) скопируйте фрагмент «стена» в буфер обмена и разместите на экране еще 9 таких копий;
  - 2) разместите в нужных местах 9 копий фрагмента «окно» и 5 копий фрагмента «крыша»;
  - 3) с помощью инструментов *Прямоугольник* и *Цвет заливки* нарисуйте закрашенный прямоугольник — дверь;
  - 4) для правильной компоновки фрагментов используйте команду [*Действия-Порядок-На передний план*].
5. Активизируйте кнопку *Выбор объекта*  и с ее помощью очертите прямоугольник, полностью вмещающий выполненный вами рисунок. Выполните команду [*Действия-Группировать*].
6. По своему усмотрению измените размеры рисунка.
7. Сохраните результат в собственной папке под именем Дом и закройте программу.

**Задание 4**

1. Откройте текстовый процессор.
2. Рассмотрите имеющиеся варианты автофигур (панель *Рисование*). Из них вы будете «собирать» орнамент<sup>1</sup>.



3. Для создания орнамента продумайте, соберите и раскрасьте повторяющийся фрагмент, например, такой: Чтобы закрасить автофигуру, ее следует выделить (щелчок левой кнопкой мыши) и указать цвет заливки.
4. Сгруппируйте созданный фрагмент.
5. В случае необходимости уменьшите созданный вами фрагмент так, чтобы его можно было 3–4 раза разместить по ширине страницы.
6. Скопируйте фрагмент в буфер и вставьте нужное количество его копий так, чтобы получилось 2–3 ряда по 3–4 фрагмента в каждом ряду.
7. С помощью инструмента *Прямоугольник* изобразите большой прямоугольник (*Нет заливки*, *Тип линии* — 6 пт), так чтобы орнамент оказался внутри него:

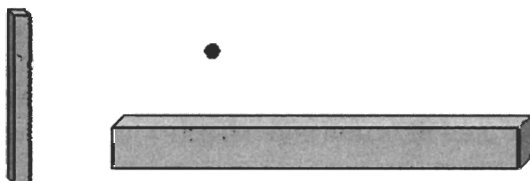


<sup>1</sup> Орнамент — узор, состоящий из ритмически упорядоченных одинаковых элементов; предназначается для украшения различных предметов, архитектурных сооружений, произведений декоративно-прикладного искусства и т. п.

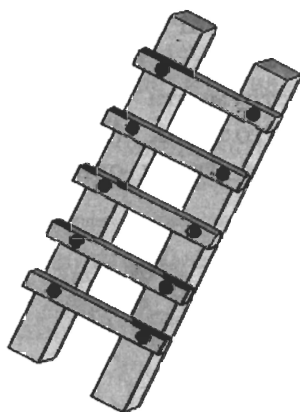
8. Сохраните результат в собственной папке под именем Орнамент и закройте программу.

### Задание 5

1. Откройте текстовый процессор.
2. Откройте файл Конструктор.doc (Мои документы\бкласс\Заготовки). На экране появятся детали конструктора:



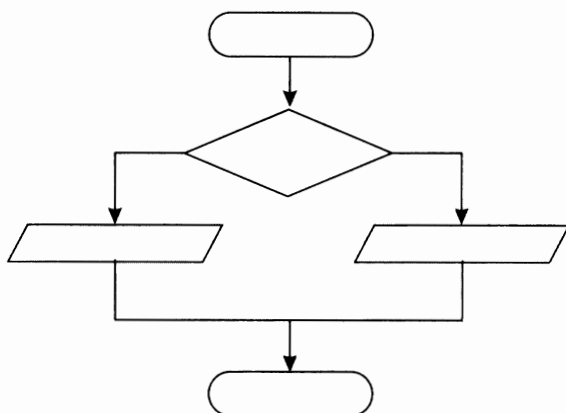
3. С помощью операций поворота ([*Действия-Повернуть/отразить*]), копирования и вставки соберите из имеющихся деталей лестницу:



4. Сохраните результат в собственной папке под именем Лестница и закройте программу.

## Задание 6

1. Откройте текстовый процессор.
2. Среди вариантов автофигур (панель *Рисование*) есть группа *Блок-схемы*. Найдите те из них, о которых шла речь на уроке.
3. Используя автофигуры, начертите следующую блок-схему:



4. Сохраните результат в собственной папке под именем *Схема*.



### Теперь мы умеем

- создавать простые графические объекты (фигуры) в текстовом процессоре;
- выделять графические объекты (фрагменты), перемещать и удалять их;
- редактировать графические объекты: изменять размеры и поворачивать, изменять цвет заливки, тип и цвет линии границы;
- копировать и размножать графические фрагменты;
- собирать сложные объекты из простых: устанавливать порядок следования, группировать, разделять сложные объекты на составные части.

## Работа 12. Рисунок на свободную тему

1. Продумайте идею своего рисунка.
2. Выполните его средствами любого знакомого вам графического редактора.
3. Сохраните результат в собственной папке под именем *Фантазия*.

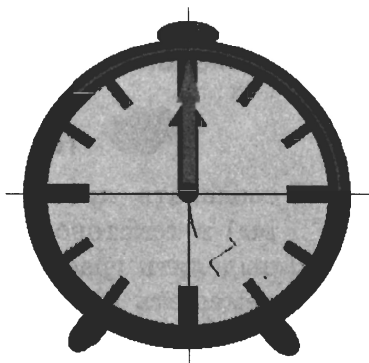


### Теперь мы умеем

- создавать рисунки по собственному замыслу средствами растровой и векторной графики.

## Работа 13. Создаем презентацию «Часы»

1. Через пункт *Программы* главного меню запустите программу PowerPoint. Создайте новую презентацию, содержащую пустой слайд.
2. С помощью инструментов панели *Рисование* изобразите циферблат с двумя стрелками. Чтобы циферблат был симметричным, сначала изобразите в центре экрана пересекающиеся горизонтальную и вертикальную прямые — оси симметрии. Пытайтесь рисовать рационально — копируйте повторяющиеся фрагменты, при необходимости изменяйте их с помощью команд группы *Повернуть/отразить*.
3. Скопируйте слайд с часами и сделайте еще 4 копии этого слайда (всего 5 слайдов).



4. Внесите изменения в положение стрелок на слайдах, так чтобы на них последовательно отмечалось время: 12.00, 12.15, 12.30, 12.45 и 13.00.
5. Выполните команду [*Показ слайдов-Смена слайдов*]. В открывшемся окне в группе *Продвижение (Смена слайда)* отметьте флажком режим *Автоматически после* и установите время *1 сек (00:01)*. Щелкните на кнопке *Применить ко всем*.
6. Запустите презентацию, нажав клавишу {F5}.
7. При наличии времени внесите в презентацию промежуточные слайды так, чтобы на них отмечалось время 12.05, 12.10 и т. д.
8. Сохраните работу в собственной папке под именем Часы.



### Теперь мы умеем

- пользоваться инструментами рисования в программе для создания презентаций;
- копировать и редактировать слайды;
- создавать линейную презентацию из нескольких слайдов.

## Работа 14. Создаем презентацию «Времена года»

1. Запустите программу PowerPoint. Создайте новую презентацию.
2. Выберите слайд *Маркированный список*. В поле *Заголовок слайда* введите текст *Времена года*, в поле *Текст слайда* перечислите все времена года.
3. В меню *Вставка* выберите пункт *Новый слайд*. Выберите слайд *Текст и графика (Заголовок, текст и графика)*. Озаглавьте слайд *Зима*. В текстовое поле введите любое известное вам четверостишие о зиме. В графическом поле разместите подходящий рисунок из коллекции *ClipArt*.

4. Аналогичным образом создайте слайды Весна, Лето и Осень.
5. Перейдите на первый слайд. Необходимо связать слово «зима» с соответствующим слайдом, иначе говоря, создать гиперссылку. Выделите слово «зима» и выполните команду [*Показ слайда — Настройка действия*]. На вкладке *По щелчку мыши* установите переключатель *Перейти по гиперссылке*. В раскрывающемся списке выберите пункт *Слайд...*, а затем в перечне слайдов — слайд Зима. Подтвердите свой выбор щелчком на кнопке *ОК*. Обратите внимание, как изменилось слово зима на первом слайде.
6. Аналогичным образом создайте гиперссылки из слов весна, лето, осень.
7. Создайте на каждом из слайдов Зима, Весна, Лето и Осень гиперссылку В начало, обеспечивающую переход на первый слайд. Область для ввода текста задайте с помощью инструмента *Надпись*.
8. Убедитесь, что смена слайдов будет осуществляться по щелчку.
9. Запустите презентацию, нажав клавишу {F5}. Просмотрите слайды презентации с использованием гиперссылок.
10. Сохраните работу в собственной папке под именем *Времена года*.



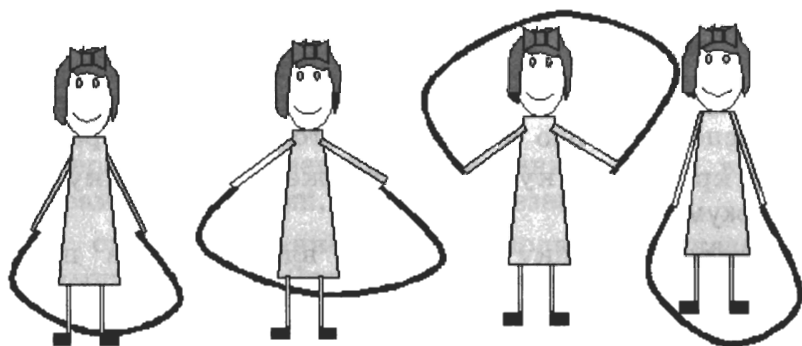
### Теперь мы умеем

- использовать шаблоны слайдов разных типов в программе для создания презентаций;
- создавать гиперссылки;
- создавать презентацию из нескольких слайдов, имеющую разветвленную структуру.



## Работа 15. Создаем презентацию «Скакалочка»

1. Запустите программу PowerPoint. Создайте новую презентацию.
2. С помощью инструментов панели *Рисование* на четырех пустых слайдах изобразите девочку, прыгающую через скакалку. Это может выглядеть, например, так:



При создании каждого следующего слайда старайтесь как можно больше использовать изображение на предыдущем слайде.

3. Когда все четыре слайда будут готовы, в меню *Показ слайдов* выберите команду *Настройка презентации*. Установите флажок *непрерывный цикл до нажатия клавиши {Esc}* и щелкните на кнопке *ОК*. В этом режиме происходит циклическое повторение показа слайдов.
4. Установите смену слайдов по щелчку и запустите презентацию.
5. Запустите презентацию в режиме автоматической смены слайдов.
6. При наличии времени дополните слайды по своему усмотрению. Например, можно использовать в качестве фонового рисунок *Дом.bmp*, созданный вами ранее в графическом редакторе *Paint*.


7. Сохраните работу в собственной папке под именем Скакалочка.



### Теперь мы умеем

- ☑ организовывать непрерывную циклическую демонстрацию презентации.

## Работа 16. Работаем с файлами и папками. Часть 2

1. Откройте двойным щелчком папку Мои документы, расположенную на Рабочем столе компьютера.
2. Откройте папку бкласс, вложенную в папку Мои документы.
3. Откройте папку Заготовки, вложенную в папку бкласс.
4. Расположите значки файлов в алфавитном порядке. Для этого установите указатель мыши в чистую область окна *Заготовки* и щелкните правой кнопкой мыши (вызов контекстного меню). Выполните команду [*Упорядочить значки-по имени*].
5. Вернитесь в папку бкласс (кнопка ).
6. Откройте собственную папку, вложенную в папку бкласс.
7. Выполните команду [*Вид-Крупные значки*].
8. Создайте папки Документы, Рисунки и Презентации.
9. Расположите значки по типам файлов ([*Упорядочить значки-по типу*]).
10. Измените размеры окна собственной папки так, чтобы ее можно было разместить в правой части экрана.
11. Еще раз откройте собственную папку. В ней откройте папку Документы и расположите ее окно в левой части экрана.

12. Поочередно переместите файлы, созданные в текстовом процессоре Word, в папку Документы. Для этого выполняйте щелчки на значках нужных файлов и, удерживая нажатой левую кнопку мыши, перетаскивайте значки в папку Документы.
13. Закройте папку Документы.
14. Еще раз откройте собственную папку. В ней откройте папку Рисунки и расположите окно этой папки в левой части экрана.
15. Поочередно переместите файлы, созданные в графическом редакторе Paint, в папку Рисунки.
16. Закройте папку Рисунки.
17. Перенесите презентации, созданные вами в программе PowerPoint, в папку Презентации.
18. В окне собственной папки подведите указатель мыши к папке Документы и вызовите контекстное меню (щелчок правой кнопкой мыши). Выберите пункт *Свойства* и изучите информацию о папке. Закройте окно *Свойства: Документы*.
19. По аналогии получите информацию о папках Презентации и Рисунки.
20. Закройте все открытые вами окна.



**Теперь мы умеем**

- размещать файлы в нужных папках;
- получать информацию о свойствах папки.

## Работа 17. Создаем слайд-шоу

С помощью программы MS PowerPoint создадим слайд-шоу для демонстрации рисунков, созданных в течение учебного года.

1. Запустите программу PowerPoint. Создайте новую презентацию.

2. Выберите *Титульный слайд*. В поле *Заголовок слайда* введите текст «Мои рисунки». В поле *Подзаголовок слайда* введите свои фамилию и имя.
3. В меню *Вставка* выберите пункт *Новый слайд*. Выберите *Пустой слайд*.
4. Вставьте в пустой слайд один из своих рисунков, созданных в программе Paint. Для этого выполните команду [*Вставка-Рисунок-Из файла*]. Разыщите рисунок и щелкните на кнопке *Вставить*. Рисунок появится на втором слайде. Поместите рисунок в центр слайда и, если это необходимо, измените размеры рисунка.
5. Повторите процедуру (п. 3–4) для других рисунков, созданных в графическом редакторе Paint.
6. Создайте пустой слайд.
7. Через пункт *Программы* главного меню запустите программу Word. Откройте один из документов, созданных вами в этой программе, содержащий рисунок. Скопируйте рисунок в буфер обмена.
8. Перейдите в окно программы PowerPoint, щелкнув на соответствующей кнопке на панели задач. Выполните команду [*Правка-Вставить*]. Рисунок появится на слайде. Поместите рисунок в центр слайда и, если это необходимо, измените размеры рисунка.
9. Повторите процедуру (п. 6–8) для других рисунков, созданных в текстовом процессоре Word.
10. Разместив все рисунки, выполните команду [*Вид-Показ слайдов*]. Для перехода к очередному слайду просто щелкайте мышью.
11. Сохраните свою работу в собственной папке Презентации под именем Рисунки.



### Теперь мы умеем

- оформлять и представлять результаты своей работы.

## \*Работа 18<sup>1</sup>. Знакомимся со средой программирования QBasic

1. Найдите на Рабочем столе ярлык среды программирования QBasic и дважды щелкните на нем.
2. Перед вами приглашение в QBasic. Если нажать на клавиатуре клавишу {Enter}, то на экране появится руководство для начинающих. В нем содержатся сведения о системе программирования и много примеров небольших программ. Для начала работы нажмите клавишу {Esc}.
3. Перед вами окно редактора QBasic. Рассмотрите его основные элементы: это строка меню, окно текстового редактора, окно для немедленного выполнения отдельных команд, полосы прокрутки, строка состояния с подсказками.
4. Щелкните левой кнопкой мыши в окне выполнения отдельных команд. Наберите в одной строке знак «?» и арифметическое выражение  $(25 - 2 * 10) / (17 - 14)$ . Здесь «\*» обозначает умножение, «/» — деление, а «?» — команду «напечатать на экране значение выражения». Чтобы вычислить значение этого выражения, нажмите клавишу {Enter}.
5. Самостоятельно вычислите в окне немедленного выполнения команд выражения  $\frac{97 + 43}{10 - 3}$  и  $\frac{(24 + 48) \cdot 5}{36}$ .
6. Выполните команду [*Файл-Открыть программу*]. В открывшемся окне найдите файл Рис1.bas, щелкните на нем (он должен выделиться), затем щелкните на кнопке *Открыть*.
7. Проследите за изменениями, произошедшими на экране: в самой верхней части окна текстового редактора появится название вызванной (открытой) вами программы, а ниже — ее текст. Запустите программу на выполнение с помощью команды *Запуск*

<sup>1</sup> Работы, отмеченные значком «\*», не обязательны для выполнения.

2. Выберите *Титульный слайд*. В поле *Заголовок слайда* введите текст «Мои рисунки». В поле *Подзаголовок слайда* введите свои фамилию и имя.
3. В меню *Вставка* выберите пункт *Новый слайд*. Выберите *Пустой слайд*.
4. Вставьте в пустой слайд один из своих рисунков, созданных в программе Paint. Для этого выполните команду [*Вставка-Рисунок-Из файла*]. Разыщите рисунок и щелкните на кнопке *Вставить*. Рисунок появится на втором слайде. Поместите рисунок в центр слайда и, если это необходимо, измените размеры рисунка.
5. Повторите процедуру (п. 3–4) для других рисунков, созданных в графическом редакторе Paint.
6. Создайте пустой слайд.
7. Через пункт *Программы* главного меню запустите программу Word. Откройте один из документов, созданных вами в этой программе, содержащий рисунок. Скопируйте рисунок в буфер обмена.
8. Перейдите в окно программы PowerPoint, щелкнув на соответствующей кнопке на панели задач. Выполните команду [*Правка-Вставить*]. Рисунок появится на слайде. Поместите рисунок в центр слайда и, если это необходимо, измените размеры рисунка.
9. Повторите процедуру (п. 6–8) для других рисунков, созданных в текстовом процессоре Word.
10. Разместив все рисунки, выполните команду [*Вид-Показ слайдов*]. Для перехода к очередному слайду просто щелкайте мышью.
11. Сохраните свою работу в собственной папке Презентации под именем Рисунки.



### Теперь мы умеем

- оформлять и представлять результаты своей работы.

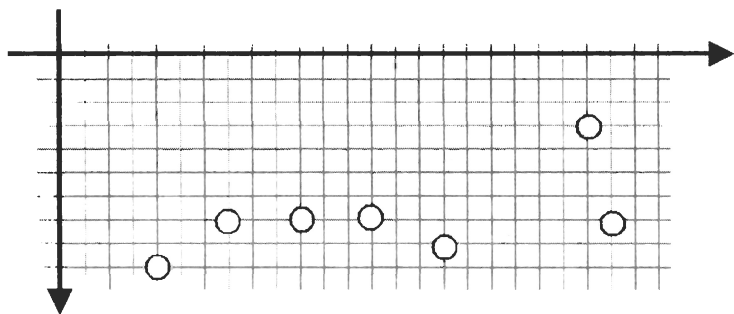
## \*Работа 18<sup>1</sup>. Знакомимся со средой программирования QBasic

1. Найдите на Рабочем столе ярлык среды программирования QBasic и дважды щелкните на нем.
2. Перед вами приглашение в QBasic. Если нажать на клавиатуре клавишу {Enter}, то на экране появится руководство для начинающих. В нем содержатся сведения о системе программирования и много примеров небольших программ. Для начала работы нажмите клавишу {Esc}.
3. Перед вами окно редактора QBasic. Рассмотрите его основные элементы: это строка меню, окно текстового редактора, окно для немедленного выполнения отдельных команд, полосы прокрутки, строка состояния с подсказками.
4. Щелкните левой кнопкой мыши в окне выполнения отдельных команд. Наберите в одной строке знак «?» и арифметическое выражение  $(25 - 2 * 10) / (17 - 14)$ . Здесь «\*» обозначает умножение, «/» — деление, а «?» — команду «напечатать на экране значение выражения». Чтобы вычислить значение этого выражения, нажмите клавишу {Enter}.
5. Самостоятельно вычислите в окне немедленного выполнения команд выражения  $\frac{97 + 43}{10 - 3}$  и  $\frac{(24 + 48) \cdot 5}{36}$ .
6. Выполните команду [*Файл-Открыть программу*]. В открывшемся окне найдите файл Рис1.bas, щелкните на нем (он должен выделиться), затем щелкните на кнопке *Открыть*.
7. Проследите за изменениями, произошедшими на экране: в самой верхней части окна текстового редактора появится название вызванной (открытой) вами программы, а ниже — ее текст. Запустите программу на выполнение с помощью команды *Запуск*

<sup>1</sup> Работы, отмеченные значком «\*», не обязательны для выполнения.

меню *Запуск* или комбинации клавиш {Shift}+{F5}. На экране появится графическая композиция, составленная из отрезков, ломаных линий, прямоугольников и окружностей. Такие рисунки вы скоро сможете выполнять самостоятельно,

8. Для продолжения работы нажмите любую клавишу. Выполните команду [*Файл-Выход*].
9. Запустите повторно среду программирования QBasic. Откройте файл Рис2.bas и запустите соответствующую программу на выполнение.
10. Выполните команду [*Файл-Новая программа*]. В очищенном окне текстового редактора наберите небольшую программу, с помощью которой будут произведены следующие действия:
  - 1) установлен графический режим;
  - 2) разноцветными точками высвечены на экране изображения звезд, образующих созвездие Большой Медведицы.



Программа будет иметь следующий вид:

```
SCREEN 12
PSET (40,90), 1
PSET (70,70), 2
PSET (100,70), 3
PSET (130,70), 4
PSET (160,80), 5
PSET (230,70), 6
PSET (220,30), 7
```



При вводе строк программы помните:

- команды программы набираются строчными латинскими буквами;
  - ввод каждой команды завершается нажатием клавиши {Enter} (обратите внимание: если команда набрана без ошибок, то после нажатия клавиши {Enter} строчные буквы «превращаются» в прописные);
  - если вы заметили ошибку, допущенную при наборе программы, то для ее исправления (удаления ошибочного символа и ввода нового символа) следует применять приемы, известные вам по работе в текстовых процессорах.
11. Запустите набранную вами программу на выполнение.
  12. Если при выполнении вашей программы компьютер обнаружит ошибки, то на экране появится соответствующее сообщение. Щелкните на кнопке *OK* и отредактируйте текст программы. Запустите программу повторно.
  13. Сохраните свою программу под именем *Проба* с помощью команды [*Файл-Сохранить как*].
  14. Завершите работу со средой программирования QBasic.



### Теперь мы умеем

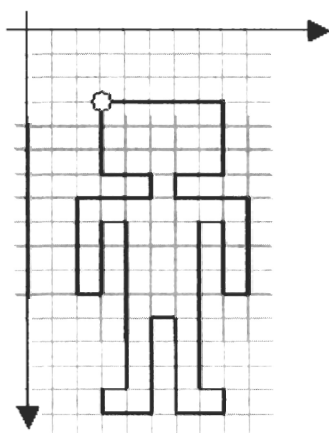
- запускать среду программирования QBasic;
- открывать и запускать на выполнение программы в QBasic;
- набирать и редактировать программы в QBasic;
- завершать работу со средой программирования QBasic.

## \*Работа 19. Исполнитель DRAW

### Задание 1

1. Запустите среду программирования QBasic.

2. Наберите команду `SCREEN 12`, устанавливающую графический режим.
3. Наберите команду `DRAW"с6"`; с ее помощью исполнитель `DRAW` установит цвет рисования 6.
4. Наберите команду `DRAW"bm30,30"`; с ее помощью исполнитель `DRAW` «прыгнет» в точку с координатами (30, 30).
5. Наберите командную строку `DRAW"r50 d30 l20 d10 r30 d40 l10 u30 l10 d70 r10 d10 l20 u40 l10 d40 l20 u10 r10 u70 l10 d30 l10 u40 r30 u10 l20 u30"`; с ее помощью исполнитель `DRAW`, перемещаясь вверх, вниз, влево и вправо, изобразит на экране компьютера следующий замкнутый контур:

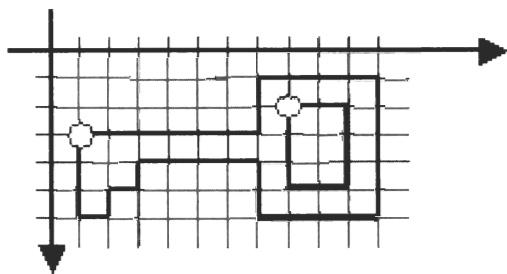


6. Наберите команду `DRAW"bm50,40"`; с ее помощью исполнитель «прыгнет» во внутреннюю точку контура с координатами (50,40).
7. Наберите команду `DRAW"р3,6"`; по ней исполнитель зальет внутреннюю область фигуры цветом 3.
8. Запустите набранную вами программу на выполнение.
9. Если при выполнении вашей программы компьютер обнаружит ошибки, то внесите исправления и запустите программу повторно.

10. Сохраните свою программу под именем Робот с помощью команды [Файл-Сохранить как...].
11. Завершите работу со средой программирования QBasic.

## Задание 2

1. Запустите среду программирования QBasic.
2. Наберите команду, устанавливающую графический режим.
3. Наберите команду, устанавливающую цвет рисования 5.
4. Наберите команду, с помощью которой исполнитель «прыгнет» в точку с координатами (10,30).
5. Наберите командную строку, с помощью которой исполнитель DRAW, перемещаясь вверх, вниз, влево и вправо, изобразит на экране компьютера внешний контур следующей фигуры:

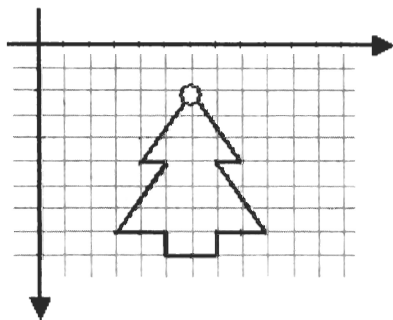


6. Наберите команду, с помощью которой исполнитель «прыгнет» в точку с координатами (80,20).
7. Наберите командную строку, с помощью которой исполнитель DRAW, перемещаясь вверх, вниз, влево и вправо, изобразит на экране компьютера внутренний контур фигуры.
8. Наберите команду, с помощью которой исполнитель «прыгнет» в точку с координатами (20,40).
9. Залейте внутреннюю область фигуры цветом 10.
10. Запустите набранную вами программу на выполнение.

11. Если при выполнении вашей программы компьютер обнаружит ошибки, то внесите исправления и запустите программу повторно.
12. Сохраните свою программу под именем Ключ с помощью команды [*Файл-Сохранить как ...*].
13. Завершите работу со средой программирования QBasic.

### Задание 3

1. Запустите систему программирования QBasic.
2. Наберите команду, устанавливающую графический режим.
3. Наберите команду, по которой исполнитель установит цвет рисования 6.
4. Наберите команду, по которой исполнитель «прыгнет» в точку с координатами (60,20).
5. Наберите командную строку `DRAW"m80,50 110 m90,80 120 d10 120 u10 120 m50,50 110 m60,20"`; с ее помощью исполнитель `DRAW`, перемещаясь не только по вертикали и горизонтали, но и к произвольной точке, изобразит на экране компьютера замкнутый контур, соответствующий нашему рисунку:

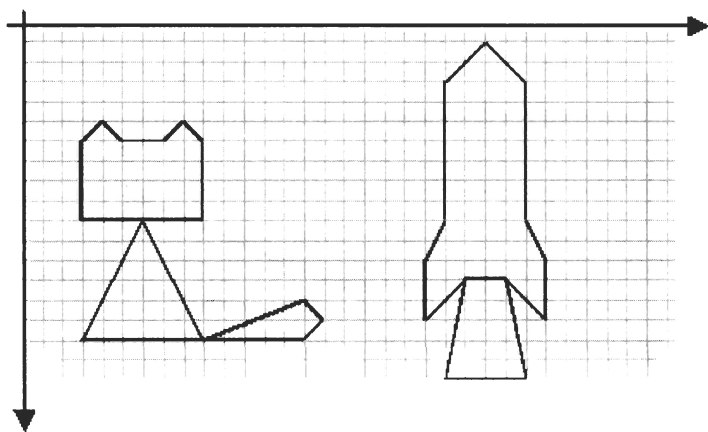


6. Наберите команду, по которой исполнитель «прыгнет» во внутреннюю точку контура с координатами (60,50).

7. Наберите команду, по которой исполнитель зальет внутреннюю область фигуры цветом 12.
8. Запустите набранную вами программу на выполнение.
9. Отладьте вашу программу (внесите исправления, если есть ошибки).
10. Сохраните свою программу под именем Ель.
11. Завершите работу со средой программирования QBasic.

#### Задание 4

1. Запустите систему программирования QBasic.
2. Напишите и отладьте программу для исполнителя DRAW, с помощью которой на экран компьютера будет выведен один из рисунков:



3. Можете дополнить рисунки деталями по своему усмотрению.
4. Отладьте свою программу (внесите исправления, если есть ошибки).
5. Добавьте в программу команды, с помощью которых рисунки будут раскрашены.
6. Сохраните свою программу под именем соответственно Кот или Ракета.
7. Завершите работу со средой программирования QBasic.



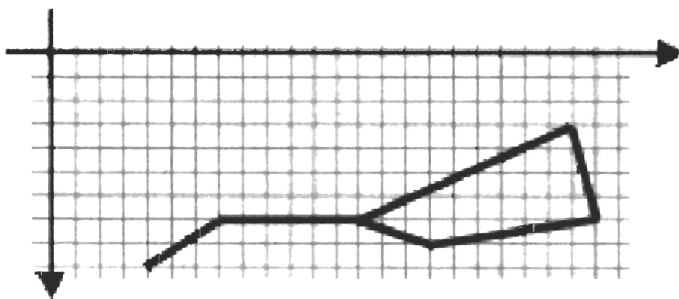
## Теперь мы умеем

- ☑ записывать отдельные команды для исполнителя DRAW в среде программирования QBasic;
- ☑ планировать последовательность действий исполнителя DRAW по созданию заданного рисунка из отрезков;
- ☑ создавать графические линейные программы для исполнителя DRAW.

### \*Работа 20. Исполнитель LINE

#### Задание 1

**Задача:** написать программу для исполнителя LINE, с помощью которой на экран компьютера будет выведен следующий рисунок:

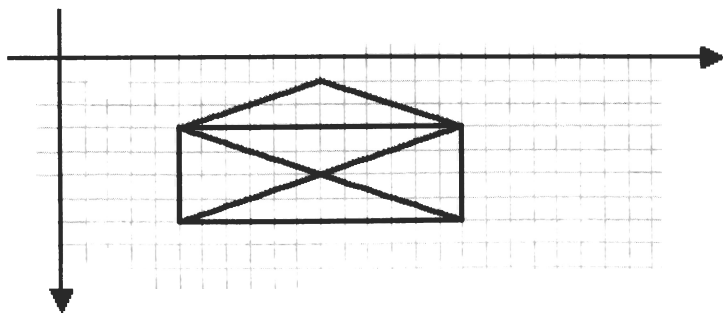


1. Запустите среду программирования QBasic.
2. Наберите команду `SCREEN 12`, устанавливающую графический режим.
3. Наберите команду `LINE (40,90) - (70,70), 5`; с ее помощью исполнитель LINE начертит на экране отрезок цвета 5 от точки с координатами (40,90) до точки с координатами (70, 70).
4. Наберите команду `LINE - (100,70), 5`; с ее помощью исполнитель LINE начертит на экране отрезок цвета 5 от последней введенной точки (ее координаты (70,70)) до точки с координатами (130,70).

5. Наберите команду `LINE - (160,80), 5`; с ее помощью исполнитель `LINE` начертит на экране отрезок цвета 5 от последней введенной точки до точки с координатами (160,80).
6. Наберите команду, с помощью которой исполнитель `LINE` начертит на экране отрезок цвета 5 от последней введенной точки до точки с координатами (230,70).
7. Самостоятельно наберите две последние команды программы.
8. Отладьте набранную вами программу.
9. Сохраните свою программу под именем Звезды1 с помощью команды [*Файл-Сохранить как...*].
10. При наличии времени напишите новую программу для исполнителя `DRAW`, с помощью которой на экран компьютера будет выведен этот же рисунок. Сохраните программу под именем Звезды2.
11. Завершите работу со средой программирования QBasic.

## Задание 2

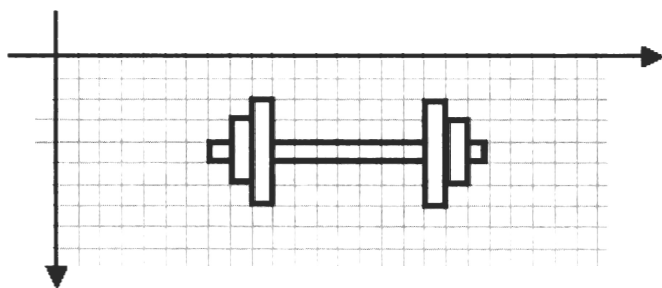
**Задача:** написать программу для исполнителя `LINE`, с помощью которой на экран компьютера (желательно, «одним росчерком») будет выведен следующий рисунок:



1. Запустите среду программирования QBasic.
2. Наберите команду, устанавливающую графический режим.
3. Продумайте план вычерчивания фигуры и наберите соответствующие команды.
4. Отладьте свою программу.
5. Сохраните свою программу под именем Конверт1.
6. При наличии времени напишите новую программу для исполнителя DRAW, с помощью которой на экран компьютера будет выведен этот же рисунок. Сохраните программу под именем Конверт2.
7. Завершите работу со средой программирования QBasic.

### Задание 3

**Задача:** написать программу для исполнителя LINE, с помощью которой на экран компьютера будет выведен следующий рисунок:



Все семь прямоугольников, образующих этот рисунок, будем строить последовательно, слева направо.

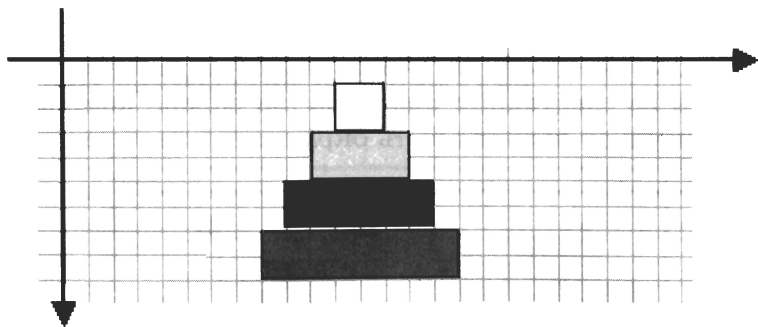
1. Запустите среду программирования QBasic.
2. Наберите команду, устанавливающую графический режим.
3. Наберите команду `LINE (70,40) - (80,50), 4, b;` с ее помощью исполнитель LINE начертит на экране прямоугольник цвета 4, у которого (70,40) — координаты левой верхней вершины, а (80,50) — правой нижней.



4. Наберите команду `LINE (80,30) - (90,60), 4, b;` с ее помощью исполнитель `LINE` начертит на экране прямоугольник цвета 4, у которого  $(80,30)$  — координаты левой верхней вершины, а  $(90,60)$  — правой нижней.
5. Наберите команду `LINE (90,20) - (100,70), 4, b;` с ее помощью исполнитель `LINE` начертит на экране 3-й прямоугольник, образующий заданную фигуру.
6. Наберите команду, с помощью которой исполнитель `LINE` начертит на экране 4-й прямоугольник, образующий заданную фигуру.
7. Самостоятельно допишите три команды, с помощью которых будут начерчены 5-й, 6-й и 7-й прямоугольники.
8. Отладьте набранную вами программу.
9. Сохраните свою программу под именем `Штанга1`.
10. Завершите работу со средой программирования `QBasic`.

#### Задание 4

**Задача:** написать программу для исполнителя `LINE`, с помощью которой на экран компьютера будет выведен следующий рисунок (на экране компьютера прямоугольники должны быть окрашены в желтый, зеленый, синий и красный цвета (сверху вниз)):

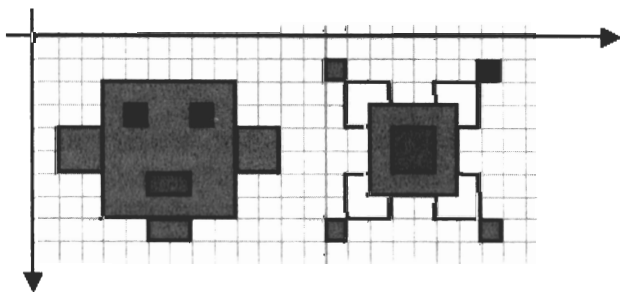


Все четыре закрашенных прямоугольника, образующих этот рисунок, будем строить последовательно, снизу вверх.

1. Запустите среду программирования QBasic.
2. Наберите команду, устанавливающую графический режим.
3. Наберите команду `LINE (90,70) - (170,90), 4, bf;` с ее помощью исполнитель `LINE` начертит на экране прямоугольник, закрашенный красным цветом, у которого  $(90,70)$  — координаты левой верхней, а  $(170,90)$  — правой нижней вершины.
4. Наберите команду `LINE (100,50) - (160,70), 8, bf;` с ее помощью исполнитель `LINE` начертит на экране прямоугольник, закрашенный синим цветом.
5. Самостоятельно допишите команды, с помощью которых будут начерчены зеленый и желтый прямоугольники.
6. Отладьте набранную вами программу.
7. Сохраните свою программу под именем Пирамида.
8. При наличии свободного времени откройте программу `Штанга1` и измените ее так, чтобы перекладина и диски штанги имели синий цвет. Сохраните новую программу под именем `Штанга2` с помощью команды [*Файл-Сохранить как...*].
9. Завершите работу со средой программирования QBasic.

### Задание 5

**Задача:** написать программу для исполнителя `LINE`, с помощью которой на экран компьютера будет выведен один из следующих рисунков (на экране компьютера части рисунков должны быть окрашены в разные цвета):



1. Запустите среду программирования QBasic.
2. Продумайте и напишите программу, с помощью которой на экран компьютера будет выведен выбранный вами рисунок. (Раскрасьте рисунок по собственному усмотрению.)
3. Отладьте свою программу.
4. Сохраните свою программу под именем соответственно Голова или Узор.
5. Завершите работу со средой программирования QBasic.



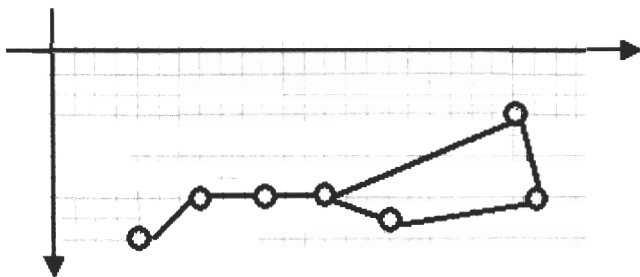
### Теперь мы умеем

- записывать отдельные команды для исполнителя LINE в среде программирования QBasic;
- планировать последовательность действий исполнителя LINE по созданию заданного рисунка из отрезков и прямоугольников;
- создавать графические линейные программы для исполнителя LINE.

## \*Работа 21. Исполнитель CIRCLE

### Задание 1

**Задача:** добавить в программу Звезда1 (или Звезды2) команды для исполнителя CIRCLE так, чтобы звезды, образующие созвездие Большой Медведицы, были изображены небольшими окружностями:



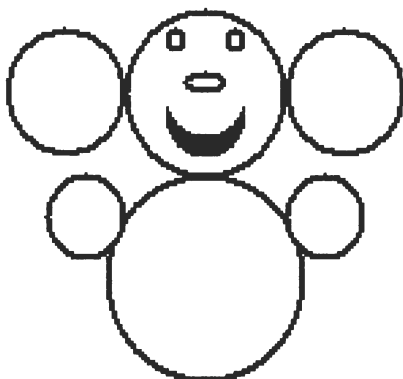
1. Запустите среду программирования QBasic.

2. С помощью команды [*Файл-Открыть программу*] откройте программу Звезды1.
3. В конец программы добавьте команду CIRCLE (40,90), 3, 8; с ее помощью исполнитель CIRCLE начертит на экране окружность синего цвета с центром в точке (40,90) и радиусом 3.
4. Наберите команду CIRCLE (70,70), 3, 8; с ее помощью исполнитель CIRCLE начертит на экране окружность синего цвета с центром в точке (70,70) и радиусом 3.
5. Наберите команду, с помощью которой исполнитель CIRCLE начертит на экране окружность синего цвета с центром в точке (100,70) и радиусом 3.
6. Самостоятельно наберите команды для изображения оставшихся звезд.
7. Отладьте набранную вами программу.
8. Сохраните свою программу под именем Звезды3 с помощью команды [*Файл-Сохранить как...*].
9. Завершите работу со средой программирования QBasic.

## Задание 2

**Задача:** на основании информации из таблицы получить представленный ниже рисунок.

№	Координата x	Координата y	Радиус	Цвет окружности	Цвет круга
1	100	150	50	15	7
2	200	150	50	15	7
3	300	150	50	15	7
4	200	150	30	4	4
5	200	146	30	7	7
6	180	130	4	1	1
7	220	130	4	1	1
8	200	270	70	15	7
9	125	220	20	15	7
10	275	220	20	15	7
11	200	150	5	6	4



1. Запустите среду программирования QBasic.
2. Установите графический режим.
3. Для обработки каждой строки таблицы вам потребуются две команды. Например, команда `CIRCLE (100,150), 50, 15` предназначена для изображения окружности белого цвета с центром в точке  $(100,150)$  и радиусом 50. Команда `PAINT (100,150), 7, 15` предназначена для окрашивания внутренней области данной окружности в цвет с номером 7; здесь  $(100,150)$  — точка, лежащая внутри окружности, 7 — цвет заливки внутренней области окружности (круга), 15 — цвет самой окружности (границы области).
4. Самостоятельно наберите команды, соответствующие строкам 2–11 таблицы.
5. Отладьте набранную вами программу.
6. Сохраните свою программу под именем Чебураш.
7. Завершите работу со средой программирования QBasic.

### Задание 3

**Задача:** написать программу на основании информации, представленной в таблице.

№	Координата x	Координата y	Радиус	Цвет окружности	Цвет круга
1	300	230	100	15	7
2	300	180	50	15	7
3	300	200	10	15	4
4	320	170	3	15	1
5	280	170	3	15	1
6	305	200	2	15	15
7	295	200	2	15	15
8	320	120	10	15	7
9	280	120	10	15	7
10	240	320	10	15	7
11	360	320	10	15	7

1. Запустите среду программирования QBasic.
2. Установите графический режим.
3. Самостоятельно наберите команды, соответствующие строкам 1–11 таблицы.
4. Отладьте набранную вами программу.
5. Сохраните свою программу под именем Свинка.
6. Завершите работу со средой программирования QBasic.



### Теперь мы умеем

- записывать отдельные команды для исполнителя CIRCLE в среде программирования QBasic;
- планировать последовательность действий исполнителя CIRCLE по созданию заданного рисунка из окружностей;
- создавать графические линейные программы для исполнителя CIRCLE.

# Терминологический словарь

**Абстрагирование** — мысленное выделение одних признаков объекта и отвлечение от других.

**Алгоритм** — конечная последовательность шагов в решении задачи, приводящая от исходных данных к требуемому результату.

**Анализ** — мысленное разделение объекта на составные части или мысленное выделение признаков объекта.

**Блок-схема** — форма записи алгоритма, при которой для обозначения различных шагов алгоритма используются геометрические фигуры: овал (начало и конец), параллелограмм (ввод/вывод), ромб (принятие решения) и прямоугольник (выполнение действия). Стрелки, связывающие эти фигуры, задают порядок выполнения соответствующих шагов.

**Векторный способ представления изображений** — способ представления изображений в цифровом виде, состоящий в том, что некоторый графический объект записывается как закодированная последовательность команд для его создания.

**Ветвление** — форма организации действий, при которой в зависимости от выполнения или невыполнения некоторого условия совершается либо одна, либо другая последовательность действий.

**Единицы измерения информации** — бит (0, 1), байт (8 битов), килобайт (1024 байта), мегабайт (1024 килобайта), гигабайт (1024 мегабайта).

**Информация для человека** — знания, которые он получает из различных источников. Сообщение, полученное человеком, может пополнить его знания, если содержащиеся в нем сведения являются для человека понятными и новыми.

- Исполнитель** — человек, группа людей, животное или техническое устройство, способные выполнять заданные команды.
- Класс** — совокупность объектов, выделенных по какому-либо признаку.
- Кодовая таблица** — таблица, задающая соответствие символов и их кодов.
- Компьютер** — универсальное электронное устройство для работы с информацией; он может применяться для многих целей: обрабатывать, хранить и передавать самую разнообразную информацию, использоваться в самых разных видах человеческой деятельности.
- Линейный алгоритм** — алгоритм, в котором команды выполняются в порядке, в котором они записаны, то есть последовательно друг за другом.
- Логика** — наука, изучающая законы и формы мышления, способы рассуждений и доказательств.
- Обобщение** — мысленное объединение однородных объектов в некоторый класс.
- Объем понятия** — множество объектов, каждому из которых принадлежат признаки, составляющие содержание понятия.
- Определение понятия** — перечисление всех существенных признаков объекта (класса однородных объектов) в связном предложении.
- Позиционная система счисления** — система счисления, в которой одна и та же цифра получает различные количественные значения в зависимости от места, или позиции, которое она занимает в записи данного числа.
- Понятие** — форма мышления, отражающая совокупность существенных признаков отдельного объекта или класса объектов.
- Правила** — знания о последовательности действий, направленных на достижение некоторой цели.



- Растровый способ представления изображений** — способ представления изображений в цифровом виде, состоящий в том, чтобы разделить графический объект вертикальными и горизонтальными линиями на крошечные фрагменты — пиксели и закодировать цвет каждого пикселя в виде двоичного числа.
- Синтез** — мысленное соединение в единое целое частей объекта или его признаков, полученных в процессе анализа.
- Система команд исполнителя (СКИ)** — перечень всех команд, которые может выполнить конкретный исполнитель.
- Система счисления** — совокупность приемов и правил для обозначения и именованя чисел.
- Содержание понятия** — все существенные признаки объекта или класса объектов, отраженные в понятии.
- Сравнение** — мысленное установление сходства или различия объектов по существенным или несущественным признакам.
- Суждение** — форма мышления, в которой что-либо утверждается или отрицается об объектах, признаках или отношениях объектов.
- Умозаключение** — форма мышления, посредством которой из одного или нескольких суждений, называемых посылками, мы по определенным правилам вывода получаем суждение-заключение.
- Файл** — информация, хранящаяся в долговременной памяти компьютера как единое целое и обозначенная именем.
- Факты** — знания об определенных явлениях, событиях, свойствах объектов и зависимостях между ними.
- Цикл (повторение)** — форма организации действий, при которой выполнение одной и той же последовательности действий повторяется, пока выполняется некоторое заранее установленное условие.
- Цифровое (двоичное) кодирование** — представление самой разнообразной информации с помощью последовательности битов (0 и 1).

# Справочные материалы

## Техника безопасности и организация рабочего места

**ЗАПОМНИТЕ!** К каждому рабочему месту подведено опасное для жизни напряжение.

Во время работы следует быть предельно внимательным.

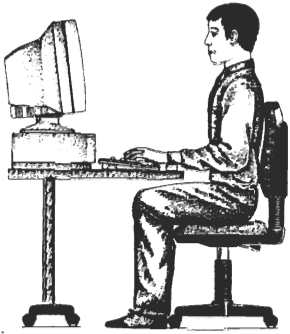
Во избежание несчастного случая, поражения электрическим током, поломки оборудования рекомендуется выполнять следующие правила:

- Входите в компьютерный класс спокойно, не торопясь, не толкаясь, не задевая мебель и оборудование и только с разрешения преподавателя.
- Не включайте и не выключайте компьютеры без разрешения преподавателя.
- Не трогайте питающие провода и разъемы соединительных кабелей.
- Не прикасайтесь к экрану и тыльной стороне монитора.
- Не размещайте на рабочем месте посторонние предметы.
- Не вставляйте со своих мест, когда в кабинет входят посетители.
- Не пытайтесь самостоятельно устранять неисправности в работе аппаратуры; при неполадках и сбоях в работе компьютера немедленно прекратите работу и сообщите об этом преподавателю.

- Работайте на клавиатуре чистыми, сухими руками; легко нажимайте на клавиши, не допуская резких ударов и не задерживая клавиши в нажатом положении.

**ЗАПОМНИТЕ!** Если не принимать мер предосторожности, работа за компьютером может оказаться вредной для здоровья.

Чтобы не навредить своему здоровью, необходимо соблюдать ряд простых рекомендаций:

- Неправильная посадка за компьютером может стать причиной боли в плечах и пояснице. Поэтому садитесь свободно, без напряжения, не сутулясь, не наклоняясь и не наваливаясь на спинку стула. Ноги ставьте прямо на пол, одна возле другой, не вытягивайте их и не подгибайте.
- 
- A black and white illustration of a man sitting at a desk, working on a computer. He is shown in profile, facing left. He is sitting on an office chair with a backrest. His feet are flat on the floor. The desk is at a height that allows him to sit upright without slouching. The computer monitor is positioned at eye level. The keyboard is on the desk, and his hands are on it. The illustration demonstrates the correct posture for working at a computer.
- Если стул с регулируемой высотой, то ее следует отрегулировать так, чтобы угол между плечом и предплечьем был чуть больше прямого. Туловище должно находиться от стола на расстоянии 15–16 см. Линия зрения должна быть направлена в центр экрана. Если вы имеете очки для постоянного ношения, работайте в очках.
  - Плечи при работе должны быть расслаблены. Предплечья должны находиться на той же высоте, что и клавиатура.
  - При напряженной длительной работе глаза переутомляются, поэтому каждые 5 минут отрывайте взгляд от экрана и смотрите на что-нибудь, находящееся вдали.

## Текстовый процессор MS Word

### Клавиши быстрого перемещения по тексту

{Home}	— в начало строки
{End}	— в конец строки
{Ctrl} + {→}	— на слово вправо
{Ctrl} + {←}	— на слово влево
{Page Up}	— на экранную страницу вверх
{Page Down}	— на экранную страницу вниз
{Ctrl} + {Page Up}	— на страницу вверх
{Ctrl} + {Page Down}	— на страницу вниз
{Ctrl} + {Home}	— в начало текста
{Ctrl} + {End}	— в конец текста

### Функции клавиш {Enter}, {Delete} и {Backspace}

*Функции клавиши {Enter} при работе с текстом:*

- указание конца абзаца;
- разрезание строки на две в том месте, где стоит курсор;
- вставка пустой строки:  
перед текущей, если нажать {Enter} в начале строки,  
после текущей, если нажать {Enter} в конце строки.

*Функции клавиши {Delete} при работе с текстом:*

- удаление символа справа от курсора;
- склеивание строк, если нажать {Delete} в конце строки.

*Функции клавиши {Backspace} при работе с текстом:*

- удаление символа слева от курсора;
- склеивание строк, если нажать {Backspace} в начале строки.

### Выделение фрагмента текста

#### Выделение произвольного участка

##### Способ 1 (с помощью мыши)

Установить указатель мыши в начало выделяемого участка, нажать левую кнопку мыши и, не отпуская ее, переместить мышь в конец выделяемого участка.

### **Способ 2** (с помощью клавиатуры)

Установить курсор в начало выделяемого участка, нажать клавишу {Shift} и, не отпуская ее, с помощью клавиш управления курсором выделить нужный участок.

### **Выделение отдельного слова**

Два раза щелкнуть на нужном слове левой кнопкой мыши.

### **Выделение строки (блока строк)**

Подвести указатель мыши к левой границе строки так, чтобы он принял вид наклонной стрелки, нажать левую кнопку мыши и, не отпуская ее, переместить мышь вниз на нужное количество строк.

### **Выделение абзаца**

#### **Способ 1**

Подвести указатель мыши к левой границе абзаца так, чтобы он изменил свой вид, выполнить двойной щелчок левой кнопкой мыши.

#### **Способ 2**

Поставить указатель мыши на нужный абзац и выполнить тройной щелчок левой кнопкой мыши.

### **Выделение всего текста**

#### **Способ 1**

Подвести указатель мыши к левой границе текста так, чтобы он изменил свой вид, и выполнить тройной щелчок левой кнопкой мыши.

#### **Способ 2**



В меню *Правка* выбрать команду *Выделить все*.

## **Работа со списками**

### **Создание нумерованного (маркированного) списка**

#### **Способ 1**

1. Установить курсор там, где будет начинаться список.

- Щелкнуть на кнопке *Нумерация*  (Маркеры ) на панели инструментов *Стандартная*. Ввести нужный текст.
- По окончании ввода текста каждого пункта списка следует нажимать клавишу {Enter}.

### Способ 2

- Установить курсор там, где будет начинаться список.
- Ввести первый номер, поставить после него точку и пробел (поставить знак \* и пробел). Ввести текст.
- По окончании ввода текста каждого пункта списка следует нажимать клавишу {Enter}.

### Удаление пунктов списка

- Выделить удаляемый пункт.
- Нажать клавишу {Delete}.

### Добавление пунктов в список

- Поставить курсор в конец пункта, после которого будет добавлен новый пункт.
- Нажать клавишу {Enter}.
- После того как появится пустая строка с соответствующим номером (маркером), ввести нужный текст.

### Изменение вида номеров (маркеров)

- Выделить изменяемый список.
- В меню *Вид* выбрать команду *Список*, в появившемся диалоговом окне указать требуемый вид номеров (маркеров).


## Работа с таблицами

### Создание таблицы


#### Способ 1

- В меню *Таблица* выбрать команду *Добавить*, а затем — команду *Таблица*.
- В появившемся диалоговом окне указать необходимое количество столбцов и строк в таблице, а также способ *Автоподбора ширины столбцов* (например, по ширине окна).

## Способ 2

1. На панели инструментов *Стандартная* щелкнуть на кнопке *Добавить таблицу*  .
2. Нажав и не отпуская левую кнопку мыши, выделить необходимое количество строк и столбцов, после чего отпустить кнопку мыши.

## Выделение элементов таблицы

Элемент	Способ выделения
Вся таблица	<p><i>Способ 1</i></p> <ol style="list-style-type: none"> <li>1. Щелкнуть мышью внутри таблицы.</li> <li>2. Выбрать команду [Таблица-Выделить-Таблица].</li> </ol> <p><i>Способ 2</i></p> <ol style="list-style-type: none"> <li>1. Щелкнуть мышью внутри таблицы.</li> <li>2. Щелкнуть на кнопке  в левом верхнем углу таблицы</li> </ol>
Строка	<p><i>Способ 1</i></p> <ol style="list-style-type: none"> <li>1. Щелкнуть мышью в нужной строке.</li> <li>2. Выбрать команду [Таблица-Выделить-Строка].</li> </ol> <p><i>Способ 2</i></p> <p>Щелкнуть мышью в поле левее выделяемой строки</p>
Столбец	<p><i>Способ 1</i></p> <ol style="list-style-type: none"> <li>1. Щелкнуть мышью в нужной строке.</li> <li>2. Выбрать команду [Таблица-Выделить-Столбец].</li> </ol> <p><i>Способ 2</i></p> <p>Щелкнуть мышью в поле выше выделяемого столбца</p>
Ячейка	<p><i>Способ 1</i></p> <ol style="list-style-type: none"> <li>1. Щелкнуть мышью в нужной ячейке.</li> <li>2. Выбрать команду [Таблица-Выделить-Ячейка].</li> </ol> <p><i>Способ 2</i></p> <p>Щелкнуть мышью в левой части выделяемой ячейки</p>

## Области выделения:

ячеек:

Наименование
Процессор
Жесткий диск
Оперативная память
Дисковод

строки:

Наименование
Процессор
Жесткий диск
Оперативная память
Дисковод

столбца:


Наименование
Процессор
Жесткий диск
Оперативная память
Дисковод

## Вставка дополнительных строк в таблицу

## Способ 1

1. Выделить строку, **перед** (или **после**) которой будет добавляться новая строка.
2. В меню *Таблица* выбрать команду *Добавить*, затем — команду *Строки выше* (или *Строки ниже*).

## Способ 2


1. Выделить строку, **перед** которой будет добавляться новая строка.
2. На панели инструментов *Стандартная* щелкнуть на кнопке *Добавить строки*  .

## Удаление строк из таблицы

## Способ 1

1. Выделить удаляемые строки.
2. В меню *Таблица* выбрать команду *Удалить*, затем — команду *Строки*.

## Способ 2

1. Выделить удаляемые строки.
2. Щелкнуть на кнопке *Вырезать*  на панели инструментов *Стандартная*.


## Вставка дополнительных столбцов в таблицу

## Способ 1

1. Выделить столбец, **слева** (или **справа**) от которого будет добавляться новый столбец.
2. В меню *Таблица* выбрать команду *Добавить*, затем — команду *Столбцы слева* (или *Столбцы справа*).



## Способ 2


1. Выделить столбец, слева от которого будет добавляться новый столбец.
2. На панели инструментов *Стандартная* щелкнуть на кнопке *Добавить столбцы* .

## Удаление столбцов из таблицы




### Способ 1

1. Выделить удаляемые столбцы.
2. В меню *Таблица* выбрать команду *Удалить*, затем — команду *Столбцы*.

### Способ 2

1. Выделить удаляемые столбцы.
2. Щелкнуть на кнопке *Вырезать*  на панели инструментов *Стандартная*.

## Сортировка информации в таблице

1. Включить панель *Таблицы и границы*, щелкнув на кнопке *Таблицы и границы*  на панели инструментов *Стандартная*.
2. Щелкнуть на заголовке столбца, по которому будет выполняться сортировка.
3. На панели *Таблицы и границы* щелкнуть на кнопке *Сортировка по возрастанию*  (или *Сортировка по убыванию* ).

## Оформление таблицы с использованием автоформата

Для того чтобы задать нужный формат, следует:

1. Выделить таблицу.
2. В меню *Таблица* выбрать команду *Автоформат* и задать нужные параметры.

## Графический редактор Paint

### Размер области для рисования

В меню *Рисунок* можно выбрать пункт *Атрибуты* и с его помощью установить нужные размеры белого «холста» — области для рисования.

### Выбор цвета

По умолчанию используются черный основной и белый фоновый цвета.

Основной цвет — тот, которым выполняется рисунок; он устанавливается щелчком левой кнопкой мыши на палитре цветов.

Фоновый цвет — этот цвет как бы подкладывается под белый холст и проявляется при использовании инструмента *Ластик*, а также в некоторых других ситуациях; он устанавливается щелчком правой кнопкой мыши на палитре цветов.

### Инструменты



— *Карандаш* (его цвет выбирается вами на палитре);



— *Кисть* (один из двенадцати вариантов кисти выбирается вами в дополнительном меню настройки);




— *Распылитель* (можно выбрать один из трех вариантов).


Инструмент выбирается левым щелчком на его кнопке. Перемещением указателя мыши его устанавливают в нужную точку рабочей области. Применяют инструмент протягиванием мыши при нажатой кнопке (левой или правой).




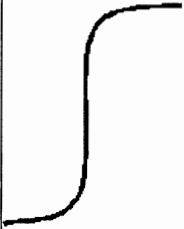

— *Заливка*. Используется для закраски замкнутой области. Его выбирают, устанавливают указатель мыши в выбранную область и выполняют щелчок (левой или правой кнопкой мыши).

Если при работе с инструментами *Карандаш*, *Кисть*, *Распылитель* и *Заливка* вы нажимаете левую кнопку мыши, то изображение выполняется основным цветом. Если нажимаете правую кнопку — фоновым.

 — *Линия*. С помощью этого инструмента удобно проводить прямые линии (отрезки). Для этого следует предварительно выбрать ширину линии (один из пяти вариантов меню настройки), щелкнуть кнопкой в начальной точке и перетащить указатель мыши в нужном направлении. Для изображения вертикальной линии, горизонтальной линии и линии с наклоном  $45^\circ$  при их создании следует удерживать нажатой клавишу <Shift>.

 — *Кривая*. С помощью этого инструмента можно изображать кривые линии, состоящие из одной или двух дуг, а также замкнутую кривую (петлю).

Кривая	Как это сделать
	<ol style="list-style-type: none"> <li>1) Активизировать инструмент <i>Кривая</i>;</li> <li>2) выбрать толщину линии;</li> <li>3) изобразить прямую линию — щелкнуть кнопкой в начальной точке и, перетащив указатель мыши в нужном направлении, отпустить его;</li> <li>4) установить указатель мыши в вершину будущей дуги и, нажав кнопку мыши, при необходимости изменить кривизну дуги, перетаскивая указатель; достигнув нужного результата, отпустить кнопку мыши;</li> <li>5) выполнить щелчок левой кнопкой мыши</li> </ol>

Кривая	Как это сделать
	<ol style="list-style-type: none"> <li>1) Активизировать инструмент <i>Кривая</i>;</li> <li>2) выбрать толщину линии;</li> <li>3) изобразить прямую линию — щелкнуть кнопкой в начальной точке и, перетаскив указатель мыши в нужном направлении, отпустить его;</li> <li>4) установить указатель мыши в вершину будущей дуги и, нажав кнопку мыши, при необходимости изменить кривизну дуги, перетаскивая указатель; достигнув нужного результата, отпустить кнопку мыши;</li> <li>5) установить указатель мыши в вершину второй дуги и выполнить те же действия, что и в п. 4</li> </ol>
	<ol style="list-style-type: none"> <li>1) Активизировать инструмент <i>Кривая</i>;</li> <li>2) выбрать толщину линии;</li> <li>3) щелкнуть левой кнопкой мыши сначала в одном, а затем в другом месте экрана — появится прямая линия;</li> <li>4) установить указатель мыши недалеко от получившейся линии, нажать левую кнопку — прямая преобразуется в замкнутую кривую (петлю);</li> <li>5) перетаскивать указатель мыши в разных направлениях, пока петля не примет нужную форму; отпустить кнопку мыши</li> </ol>

Для создания геометрических фигур применяют инструменты:



— *Прямоугольник*;



— *Эллипс*;



— *Скругленный прямоугольник*;



— *Многоугольник*.

Прямоугольник, эллипс и скругленный прямоугольник строят так:

- 1) активизируют нужный инструмент;
- 2) помещают указатель мыши в ту часть экрана, где должна быть построена фигура;
- 3) нажимают левую кнопку мыши и перетаскивают указатель мыши по диагонали.

Чтобы нарисовать квадрат и окружность при использовании инструментов *Прямоугольник* и *Эллипс*, следует удерживать нажатой клавишу <Shift>.

С помощью инструмента *Многоугольник* можно изображать замкнутые ломаные линии причудливой формы. Делают это так:

- 1) активизируют инструмент *Многоугольник*;
- 2) изображают одно звено ломаной (устанавливают указатель в определенном месте рабочей области и перетаскивают его в нужном направлении), отпускают кнопку мыши;
- 3) поочередно выполняют щелчки в других вершинах ломаной — каждая следующая вершина соединяется отрезком с предыдущей;
- 4) в последней вершине выполняют двойной щелчок — последняя вершина соединяется с первой.

Чтобы многоугольник содержал только углы по  $45^\circ$  и  $90^\circ$ , при перемещении указателя мыши следует удерживать нажатой клавишу <Shift>.



Когда активизирован любой из инструментов *Прямоугольник*, *Эллипс*, *Скругленный прямоугольник* или *Многоугольник*, под панелью инструментов появляются три возможных способа заполнения. Ниже показан эффект от применения каждого из них.



Чтобы установить ширину контура для фигур, получаемых с помощью инструментов *Прямоугольник*, *Эллипс*, *Многоугольник* и *Скругленный прямоугольник*, следует предварительно активизировать инструмент *Линия* и в его меню настройки указать требуемую ширину.

### Работа с фрагментами

Для выделения фрагмента предназначены два инструмента:

-  — *Выделение прямоугольной области* (указатель мыши перетаскивают по диагонали через область выделения);
-  — *Выделение произвольной области* (указатель перетаскивают вокруг нужной части рисунка).

Снимается выделение щелчком в любом месте вне выделенной области.

Любой из инструментов выделения фрагментов можно использовать в двух вариантах:



— *Непрозрачный фрагмент* (захватывается выделенная область, имеющая цвет фона);




— *Прозрачный фрагмент* (не захватываются части выделенной области, имеющие цвет фона).



Выделенный фрагмент можно:

- удалить — выделенный фрагмент исчезнет, а его место заполнится цветом фона;
- переместить методом перетаскивания в любое место холста, при этом прежнее место расположения фрагмента заполнится цветом фона;
- вырезать (команда *Вырезать* из меню *Правка*) — выделенный фрагмент исчезает с экрана и помещается в буфер обмена, а его место на холсте заполняется цветом фона;
- копировать (команда *Копировать* из меню *Правка*) — выделенный фрагмент остается на прежнем месте, а его точная копия помещается в буфер обмена;
- размножить перетаскиванием при нажатой клавише {Ctrl} — очередная копия фрагмента будет появляться всякий раз, когда кнопка мыши будет отпущена;
- преобразовать (повернуть, растянуть, наклонить) — в диалоговых окнах команд из меню *Рисунок* точно указывается, как именно будет преобразован выделенный фрагмент.

### Исправление ошибок

*Вариант 1.* Инструмент *Ластик*  применяют, если область для внесения изменений небольшая. Чем мельче недочеты, тем миниатюрнее должен быть *Ластик* для их устранения. Изображение стирается протягиванием. Очищаемая область принимает цвет, заданный в качестве фонового.

**Вариант 2.** Если требуется перерисовать большую область рисунка, то поступают так:

- 1) с помощью инструментов  и  выделяют произвольную или прямоугольную область;
- 2) в меню *Правка* выбирают команду *Очистить выделение*. При этом очищенная область заливается цветом фона.


**Вариант 3.** Если рисунок не удался совсем, то можно начать работу заново, предварительно выполнив команду *Очистить* из меню *Рисунок*.

**Вариант 4.** В графическом редакторе Paint можно отменить три последних действия. Для этого служит команда *Отменить*. Команда *Повторить* меню *Правка* позволяет вернуться к исправленному варианту.

### Дополнительные возможности

Инструмент *Масштаб*  позволяет увеличивать изображение. Работают с ним так:

- 1) активизируют инструмент *Масштаб*;
- 2) появившийся прямоугольник помещают над той областью, которую необходимо увеличить;
- 3) увеличенный рисунок появляется после щелчка левой кнопкой мыши;
- 4) возвращают изображение в нормальный вид повторным выбором инструмента *Масштаб* и щелчком в произвольном месте рабочей области.

Дополнить рисунок текстом можно с помощью инструмента *Надпись* . Делают это так:

- 1) активизируют инструмент *Надпись*;
- 2) щелчком отмечают место размещения текста — открывается поле ввода с мигающим курсором;




- 3) при необходимости шрифт, его размер и начертание изменяют на панели атрибутов текста (если панель атрибутов текста не отображается, ее включают в меню *Вид*);
- 4) при необходимости размер поля ввода изменяется путем перетаскивания его границ.

Пока курсор находится внутри рамки, можно:

- изменить текст по тем же правилам, что и в текстовом редакторе;
- щелчком левой кнопкой мыши выбрать на палитре цвет надписи;
- установить непрозрачный фон и щелчком правой кнопкой выбрать на палитре цвет фона для текста;
- сдвинуть надпись или изменить ее размер.

В отличие от других инструментов инструмент *Надпись* не работает с увеличенным с помощью инструмента *Масштаб* изображением.

Инструмент *Выбор цвета*  позволяет копировать цвет любого объекта. Для этого следует:

- 1) активизировать инструмент *Выбор цвета*;
- 2) щелкнуть на объекте, цвет которого следует скопировать;
- 3) с помощью инструмента *Заливка* закрасить нужную область.

Для изменения цвета существующей линии или ее части можно использовать *Цветной ластик*. Для этого следует:

- 1) с помощью инструмента *Выбор цвета* сделать цвет контура основным;
- 2) сделать новый цвет фоновым;
- 3) выбрать инструмент *Ластик*;
- 4) нажать правую кнопку мыши и, не отпуская ее, пройти линии рисунка, цвет которых следует заменить.

## Система программирования QBasic

### Среда графических исполнителей

SCREEN 12 — команда, задающая на экране область размером 639 единиц по горизонтали на 463 единицы по вертикали, в которой можно строить различные фигуры и окрашивать их в любой из 16 цветов (0–15).

### Цветовая палитра

№ цвета	Описание
0	Черный
1	Синий
2	Зеленый
3	Голубой
4	Красный
5	Коричневый
6	Лиловый
7	Светло-серый
8	Темно-серый
9	Светло-синий
10	Светло-зеленый
11	Светло-голубой
12	Розовый
13	Светло-лиловый
14	Желтый

## Система команд исполнителя DRAW

Команда	Описание
<i>cn</i>	Установка цвета рисования <i>n</i> (0–15)
<i>mx, y</i>	Движение из текущей точки в точку с координатами ( <i>x, y</i> )
<i>b</i>	Двигаться, но не рисовать («прыгать»)
<i>un</i>	Вверх на <i>n</i>
<i>dn</i>	Вниз на <i>n</i>
<i>ln</i>	Влево на <i>n</i>
<i>rn</i>	Вправо на <i>n</i>
<i>en</i>	По диагонали вверх и вправо на <i>n</i>
<i>fn</i>	По диагонали вниз и вправо на <i>n</i>
<i>gn</i>	По диагонали вниз и влево на <i>n</i>
<i>bn</i>	По диагонали вверх и вправо на <i>n</i>
<i>p</i>	Заливка замкнутого контура
<i>n</i>	Двигаться, рисовать, затем вернуться в исходную точку

## Система команд исполнителя LINE

### Отрезок

LINE (*xn, yn*) — (*xk, yk*), *c*

Здесь: (*xn, yn*) — координаты начала отрезка;

(*xk, yk*) — координаты конца отрезка;

*c* — номер цвета рисования контура.

### Ломаная линия

Чтобы изобразить ломаную линию, следует несколько раз применить сокращенную форму записи оператора LINE:

LINE — (*xk, yk*), *c*

Здесь указывается только конец отрезка, а его началом служит текущая точка (координаты которой вводились последними).

### Прямоугольник

#### Закрашенный прямоугольник

LINE ( $x_a, y_a$ ) — ( $x_c, y_c$ ),  $c$ ,  $bf$

Здесь: ( $x_a, y_a$ ) и ( $x_c, y_c$ ) — координаты диагонали прямоугольника;  
 $c$  — цвет контура и внутренней области прямоугольника.

#### Незакрашенный прямоугольник

LINE ( $x_a, y_a$ ) — ( $x_c, y_c$ ),  $c$ ,  $b$

Здесь: ( $x_a, y_a$ ) и ( $x_c, y_c$ ) — координаты диагонали прямоугольника;  
 $c$  — цвет контура.

### Система команд исполнителя CIRCLE

#### Окружность

CIRCLE ( $x, y$ ),  $r$ ,  $c$

Здесь: ( $x, y$ ) — координаты центра окружности;  
 $r$  — радиус;  
 $c$  — цвет окружности.

#### Круг

CIRCLE ( $x, y$ ),  $r$ ,  $c$

PAINT ( $x, y$ ),  $c1$ ,  $c2$

Здесь:  $(x, y)$  — координаты центра окружности;  
 $r$  — радиус;  
 $c1$  — цвет заливки окружности;  
 $c2$  — цвет контура окружности.

### Овал

CIRCLE  $(x, y), r, c, k$

Здесь:  $(x, y)$  — координаты центра окружности,  
 $r$  — радиус;  
 $c$  — цвет окружности;  
 $k$  — число, определяющее форму овала:  
если  $k < 1$ , то овал вытянут по горизонтали,  
если  $k > 1$ , то — по вертикали.

*Учебное издание*

**Босова Людмила Леонидовна**

**ИНФОРМАТИКА И ИКТ**

**Учебник для 6 класса**

Ведущий редактор *О. Полежаева*

Иллюстрации: *С. Белаш*

Художник *С. Инфантэ*

Технический редактор *Е. Денюкова*

Корректор *Е. Клитина*

Компьютерная верстка: *В. Носенко, С. Янковая*

Подписано в печать 12.01.12. Формат 60×90/16.  
Усл. печ. л. 13,5. Доп. тираж 50 000 экз. Заказ 1503.

Издательство «**БИНОМ. Лаборатория знаний**»

125167, Москва, проезд Аэропорта, д. 3

Телефон: (499) 157-5272

e-mail: [binom@Lbz.ru](mailto:binom@Lbz.ru), <http://www.Lbz.ru>

Отпечатано с готового оригинал-макета в ООО ПФ «Полиграфист»,

160001, г. Вологда, ул. Челюскинцев, 3.

Тел.: 8(817-2) 72-61-75; 8(817-2) 72-60-63.

УДК 004.9  
ББК 32.97  
Б85

**Босова Л. Л.**

**Б85 Информатика и ИКТ : учебник для 6 класса / Л. Л. Босова. — 4-е изд. — М. : БИНОМ. Лаборатория знаний, 2012. — 215 с. : ил.**

**ISBN 978-5-9963-0972-6**

Учебник для 6 класса входит в состав учебно-методического комплекта по информатике и ИКТ для 5–7 классов. В состав УМК входят учебники, рабочие тетради, методическое пособие для учителей и набор цифровых образовательных ресурсов на диске «Информатика 5–7».

Материал учебника структурирован по пяти главам, первые три из которых посвящены рассмотрению теоретических вопросов («Компьютер и информация», «Человек и информация», «Алгоритмы и исполнители»), четвертая содержит материал для дополнительного изучения, пятая — компьютерный практикум.

Файлы, необходимые для выполнения работ компьютерного практикума, размещены на сайте издательства.

**УДК 004.9  
ББК 32.97**

**По вопросам приобретения обращаться:  
«БИНОМ. Лаборатория знаний»  
Телефон: (499) 157-5272  
e-mail: binom@Lbz.ru, <http://www.Lbz.ru>**

**ISBN 978-5-9963-0972-6**

**© БИНОМ. Лаборатория знаний,  
2012**

ISBN 978-5-9963-0972-6



9 785996 309726