



А. Г. ГЕЙН Н. А. ЮНЕРМАН А. А. ГЕЙН

ИНФОРМАТИКА




ПРОСВЕЩЕНИЕ
ИЗДАТЕЛЬСТВО

8



А. Г. ГЕЙН Н. А. ЮНЕРМАН А. А. ГЕЙН

ИНФОРМАТИКА



8 **КЛАСС**

**Учебник
для общеобразовательных
учреждений**

*Рекомендовано
Министерством образования и науки
Российской Федерации*

Москва «Просвещение» 2013

Перед вами продолжение учебника для 7 класса «Информатика», по которому, как мы надеемся, вы занимались в прошлом году. Вам приоткрылась дверь в одну из самых современных наук, и теперь вы будете углублять и расширять свои знания. Начнём мы с повторения — ведь нельзя двигаться дальше, не вспомнив того, что уже пройдено.

Работая с этим учебником, вы продолжите знакомство с алгоритмами. Как и прежде, помощником в этом вам будет учебный исполнитель Паркетчик. Осваивая технологии обработки информации, вы войдёте в мир глобальных компьютерных сетей, в том числе в мир Интернета.

Как работать с учебником

Учебник разделён на главы, а главы разделены на параграфы. В начале каждого параграфа сформулированы *проблемные вопросы*, ответам на которые посвящён этот параграф, и приведены *ключевые слова* — они служат главным ориентиром в материале каждого параграфа. Ключевые слова помогут вам найти дополнительную информацию по заинтересовавшему вас вопросу.

Новые термины напечатаны жирным шрифтом. Определения, свойства и правила выделены так:

Информация — это отражение разнообразия в существующем мире.

Рассказать об информатике всё невозможно, но некоторые сведения о том, что лежит за страницами учебника, всё же помещены в эту книгу. Такой материал располагается на розовом фоне.

По ходу объяснительного текста время от времени даются вопросы и небольшие задания. Старайтесь тут же ответить на заданный вопрос или выполнить предложенное задание — это облегчит вам понимание последующего текста. Обычно в тексте после вопроса приведён ответ на него. Но не спешите сразу смотреть дальше в текст — ведь намного интереснее самому догадаться и получить ответ, сверив его потом с тем ответом, который предлагаем вам мы.

В каждом параграфе есть рубрика *Подведём итоги*. В ней размещены выводы и основные сведения, которые вы должны были усвоить, изучив материал данного параграфа.

Учебник содержит разнообразные вопросы и задания. Выполняя задания, вы проверите, насколько хорошо вы умеете применять полученные вами знания. Самые трудные задания отмечены знаком *. В конце каждой главы помещены тестовые задания, позволяющие проверить, как вами усвоен материал главы в целом.

Номер задания, который повторяется в рабочей тетради к учебнику, помещён в синюю рамку, например: **22**. Почему же рабочая тетрадь напечатана отдельной книгой? Потому что в ней помещены задания, где вам будет предложено заполнить какие-либо пропуски в схемах, пустые клетки в таблицах и т. п. А в самом учебнике никаких записей делать нельзя! В случае отсутствия у вас рабочей тетради вам придётся перерисовать схему или таблицу в обычную тетрадь и уже только после этого выполнить задание.

Практические навыки в обработке информации и в решении задач с помощью компьютера вы будете осваивать, выполняя лабораторные работы в компьютерном классе. Описание этих работ также приведено в учебнике.

А теперь в путь! За новыми знаниями в страну Информатика.



Глава 1

Переменные в алгоритмах и структуры данных

Главное требование к формальному исполнителю — точно выполнять предписанную ему инструкцию. Но и к самой инструкции есть непреложные требования: она должна быть написана на понятном исполнителю языке и содержать команды на выполнение только таких действий, которые допустимы для исполнителя. Данная глава начинается с напоминания того, как составляются подобные инструкции.

Но этим не ограничится ваше знакомство с возможностями формальных исполнителей. Мы расскажем о том, как организуются вычисления, как передаётся информация, полученная при исполнении одного алгоритма, в другой алгоритм и т. п.

Вы уже знакомы с формальным исполнителем Паркетчиком, который перемещается по клетчатому полю и выкладывает на нём плитки красного и зелёного цвета. Он умеет проверять условия: является ли граница клетки стеной, ограничивающей поле, и какой цвет имеет плитка, на которой он стоит. В 8 классе вам предстоит продолжить это увлекательное знакомство, решая разные задачи, иногда весьма непростые.

§ 1. Алгоритм и алгоритмические конструкции. Сложные условия



Как из простых условий составляются сложные?

Как проверяют истинность сложных условий?

Прежде всего напомним некоторые понятия и правила, которые вы изучили в 7 классе.

Исполнитель — это субъект или объект, выполняющий действия согласно предписанной ему инструкции.

Исполнитель может быть **формальным**; такой исполнитель выполняет инструкцию, не вникая в её смысл и не принимая во внимание цель её составления.

- Приведите пример формального исполнителя.

В роли инструкции для формального исполнителя нередко выступает алгоритм.

Ключевые слова:

алгоритм; оператор цикла; алгоритмические конструкции; оператор; оператор ветвления; операторные скобки; формальный исполнитель; система команд исполнителя; таблица истинности

Алгоритм — это организованная последовательность действий, допустимых для некоторого исполнителя.

Запись алгоритма состоит из **команд**, каждая из которых указывает исполнителю, какое допустимое действие он должен совершить. Набор всех команд, понимаемых исполнителем, называется **системой команд исполнителя** (сокращённо **СКИ**). По-другому команды называют **операторами**. Последовательность операторов, составляющих тело программы или какой-либо алгоритмической конструкции, заключают в **операторные скобки**. В записи алгоритма могут присутствовать **комментарии**, поясняющие, для чего предназначено действие в алгоритме.

- Вспомните, какие алгоритмические конструкции были вами изучены в 7 классе.

Таблица 1.1

Форма записи цикла	Форма записи ветвления
Делать пока (условие) { оператор; оператор; оператор; ... }	Если (условие) то { оператор; оператор; ... } иначе { оператор; ... }

Вы наверняка назовёте такие конструкции, как ветвление и цикл. Что в них общего? Они позволяют сообщить исполнителю, нужно ли ему совершать ту или иную последовательность действий. Чтобы в этом убедиться, достаточно одного взгляда на запись этих конструкций (таблица 1.1). В заголовке цикла и в заголовке ветвления записывается условие, справедливость которого исполнителю надлежит проверить, прежде чем начать выполнять соответствующую последовательность операторов.

В алгоритмических конструкциях можно записывать только такое условие, истинность которого умеет проверять формальный исполнитель. Для исполнителя не будет проблемой проверить истинность отрицания того же условия. Ведь ясно, что

не истина = ложь

не ложь = истина

А как следует поступить формальному исполнителю в вашем лице при исполнении алгоритма «**Если** (на улице дождь или ветер) **то** надеть куртку»? Здесь фигурируют два условия: на улице дождь, на улице ветер. Надеемся, что вы умеете проверять истинность каждого условия. А что означает истинность высказывания «на улице дождь или ветер»? Для истинности такого высказывания достаточно истинности одного из условий.

Подобным правилом определения истинности высказывания, составленного из двух других с помощью союза «или», владеет каждый формальный исполнитель. Он всегда помнит, что:

истина **или** истина = истина

истина **или** ложь = истина

ложь **или** истина = истина

ложь **или** ложь = ложь

Нетрудно догадаться, как распространить это правило на большее число высказываний, если для их соединения используется только союз **или**.



Высказывание, составленное из двух высказываний или большего числа высказываний с помощью союза **или**, истинно в том и только в том случае, когда истинно хотя бы одно из составляющих его высказываний.

Кроме союза **или**, высказывания могут быть соединены союзом **и**.

- Сформулируйте самостоятельно правило определения истинности высказывания, составленного из двух других с помощью союза **и**.

Вот как можно записать это правило в форме равенств:

истина **и** истина = истина
 истина **и** ложь = ложь
 ложь **и** истина = ложь
 ложь **и** ложь = ложь

Легко догадаться, как это правило распространить на произвольное число высказываний, соединённых союзом **и**.

Высказывание, составленное из двух высказываний или большего числа высказываний с помощью союза **и**, истинно в том и только в том случае, когда истинны все составляющие его высказывания.

Конечно, в сложных высказываниях может быть не один союз, а много. И частица **не** может переплетаться с ними довольно причудливыми способами. Для указания порядка соединения высказываний используют скобки.

Написанные выше равенства нередко оформляют в виде таблиц (таблицы 1.2—1.3), которые называются **таблицами истинности**. В них буквами **A** и **B** мы будем обозначать высказывания, а вместо слов «Истина» и «Ложь» писать только первые буквы **И** и **Л**.

Таблица 1.2

A	B	A и B
И	И	И
И	Л	Л
Л	И	Л
Л	Л	Л

Таблица 1.3

A	B	A и B
И	И	И
И	Л	Л
Л	И	Л
Л	Л	Л

Выполняя задания к этому параграфу, вам предстоит вспомнить то, что вы изучили в 7 классе. Если у кого-то возникнут трудности, мы советуем перечитать соответствующие параграфы нашего учебника для 7 класса.

Подведём итоги

- 1 Если имеется несколько высказываний, истинность которых формальный исполнитель умеет проверять, то он умеет проверять истинность высказываний, полученных с помощью отрицания и соединения их союзами **и** и **или**.
- 2 Чтобы проверить истинность высказываний, полученных соединением других высказываний с помощью союзов **и** и **или**, достаточно знать формальные правила, определённые таблицами истинности.

Вопросы и задания

- 1 Какая форма организации действий в алгоритме называется циклом? Приведите пример алгоритма, в котором используется эта конструкция.
- 2 Какая форма организации действий в алгоритме называется ветвлением? Приведите пример алгоритма, в котором используется эта конструкция.
- 3 Что такое ветвление в полной форме и ветвление в неполной форме?
- 4 Зачем нужны операторные скобки?
- 5 а) Прочитайте стихотворение-алгоритм Г. Остера.

Если ты попал в больницу
И не хочешь там валяться,
Жди, пока к тебе в палату
Самый главный врач придёт.
Укуси его, и сразу
Кончится твоё лечение,
В тот же вечер из больницы
Заберут тебя домой.

Перепишите его в форме, принятой для записи алгоритмов.

б) Аналогично пункту а творчески переработайте ещё одно произведение Г. Остера, обратив внимание на употребление союзов.

Если вы по коридору
Мчитесь на велосипеде,
А навстречу вам из ванной

Вышел папа погулять,
Не сворачивайте в кухню,
В кухне твёрдый холодильник,
Тормозите лучше в папу,
Папа мягкий. Он простит.

- 6** Прочитайте высказывание одного школьника:
«Я был сегодня на всех уроках и получил пятёрку».
Определите, истинно оно или ложно в следующих ситуациях:
а) школьник не пропустил ни одного урока и на уроке информатики получил пятёрку;
б) школьник пропустил один из уроков, а на уроке информатики получил пятёрку;
в) школьника спросили на каждом из шести уроков, которые были в этот день, и он получил три пятёрки и три четвёрки.
- 7** Исполнитель умеет заменять в слове ровно одну букву, причём из осмысленного слова должно получаться снова осмысленное слово (иначе исполнитель ломается). Составьте алгоритмы преобразования:
а) слова МЯЧ в слово ГОЛ;
б) слова ТИШЬ в слово БУРЯ;
в)* слова ВРАГ в слово ДРУГ;
г)* слова ТЬМА в слово СВЕТ.
- 8** Алгоритмические конструкции можно изображать с помощью схем. Какой схемой изображают ветвление? Какой схемой изображают цикл?
- 9** Для разных значений a , b , c исполните алгоритм, изображённый на рисунке 1.1.
Определите, для чего предназначен этот алгоритм.
- 10** Имеется несколько одинаковых по виду монет, среди которых ровно одна фальшивая, отличающаяся от настоящей по весу. Составьте алгоритм нахождения фальшивой монеты с помощью чашечных весов без гирь, если:
а) известно, что фальшивая монета тяжелее настоящей;
б) неизвестно, фальшивая монета легче настоящей или тяжелее.
- 11** Имеется несколько одинаковых по виду монет, среди которых, возможно, встречаются фальшивые. Известно, что все фальшивые монеты имеют один и тот же вес и он меньше, чем у настоящей монеты. Составьте алгоритм нахождения всех фальшивых монет с помощью чашечных весов без гирь (по условию предполагается, что в наборе обязательно есть хотя бы одна настоящая монета).

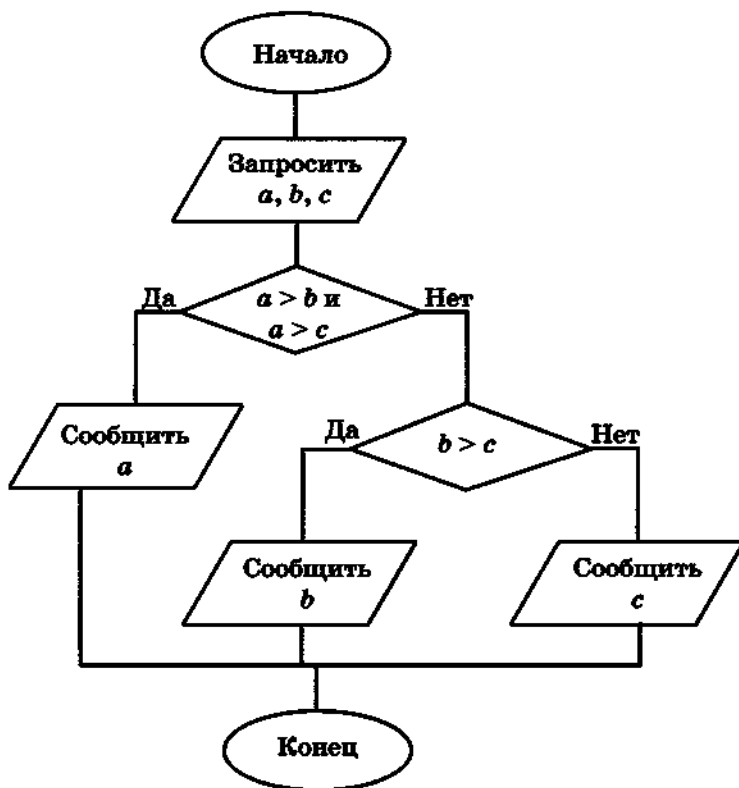


Рис. 1.1

- 12** Для ситуаций, указанных на рисунке 1.2, в роли Паркетчика исполните следующий алгоритм:

Алгоритм

```

{ Перейти на (2; 2);
  Если (зелёная плитка) то
  { Снять плитку;
    Положить красную;
  }
  Если (не красная плитка) то
  { Перейти на (3; 3);
  }
  иначе
  { Положить красную плитку; }
  Если (не красная плитка) и (не зелёная плитка) то

```

```

{ Положить зелёную плитку;
}
Перейти на (2; 2);
Если (красная плитка) или (зелёная плитка) то
{ Перейти на (4; 4);
}
Если (красная плитка) то
{ Снять плитку;
}
}

```

Объясните, почему в некоторых ситуациях Паркетчик отказывается работать, выдавая сообщение «Ошибка во время исполнения».

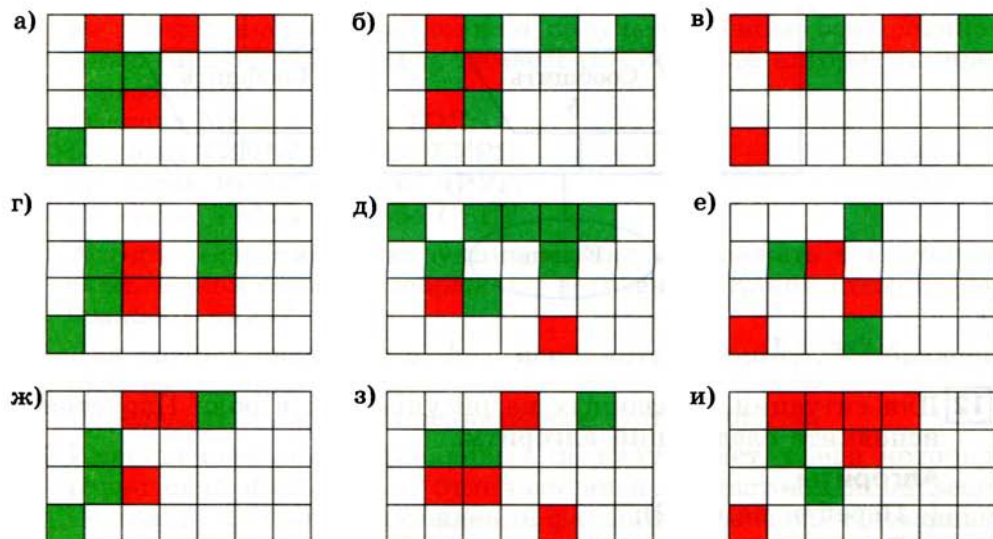


Рис. 1.2

13 а) На прямоугольном поле, свободном от плиток, требуется выложить ряд красных плиток, начиная с левого нижнего угла (рис. 1.3, а). Составьте соответствующий алгоритм для Паркетчика.

б) На прямоугольном поле, свободном от плиток, требуется выложить лестницу из красных плиток, ведущую из левого верхнего угла поля до нижней границы (рис. 1.3, б). Составьте соответствующий алгоритм для Паркетчика.

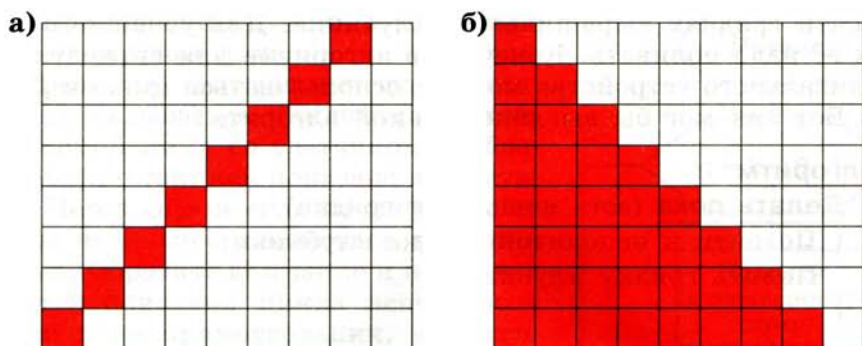


Рис. 1.3

- 14** На прямоугольном поле, свободном от плиток, требуется выложить зелёными плитками квадрат наибольших размеров. Составьте соответствующий алгоритм для Паркетчика.
- 15*** На поле располагается несколько фигур, составленных из трёх красных плиток. Ясно, что это или полоска (расположенная либо вертикально, либо горизонтально), или уголок (расположенный одним из четырёх возможных вариантов). Никакие две фигуры не соприкасаются.
- а) Составьте алгоритм, выполнив который Паркетчик поменяет цвет каждой полоски, а все уголки оставит красными.
- б) Составьте алгоритм, выполнив который Паркетчик поменяет цвет каждого уголка, а все полоски оставит красными.

§ 2. Ещё раз о циклах.

Оператор цикла со счётчиком



Как в алгоритмах записывается повторение действий?

- Как организуется цикл, если заранее известно количество повторений тела цикла?

Стремление переложить на плечи автоматических устройств как можно больше рутинной работы естественно и похвально. Пусть, например, у нас имеется автоматизированная теплица, в которой

Ключевые слова:

оператор цикла;
цикл со счётчиком

на десяти грядках выращивается клубника. Для успешного созревания её надо поливать. Конечно, в алгоритме для автоматического полива устройства можно воспользоваться циклом **Делать пока**. Вот как мог бы выглядеть такой алгоритм:

Алгоритм

```
{ Делать пока (есть неполитая грядка)
  { Подойти к неполивной грядке клубники;
    Полить грядку клубники;
  }
}
```

Но про исполнение данного цикла мы знаем гораздо больше, чем обычно: нам известно, сколько раз будет исполняться тело цикла и какая по счёту грядка будет поливаться при очередном исполнении цикла. Это знание отнюдь не лишнее. Оказывается, тогда можно воспользоваться другой конструкцией цикла:

Алгоритм

```
{ Делать от  $k := 1$  до 10
  { Подойти к  $k$ -й грядке клубники;
    Полить  $k$ -ю грядку клубники;
  }
}
```

Для простоты понимания будем пока считать, что буквой k обозначена некоторая коробочка, в которую при каждом выполнении цикла кидается по одной бусинке. Эта коробочка называется счётчиком цикла. Как только в коробочке станет 10 бусинок, цикл исполнится последний раз.

Бывает, что в счётчик цикла при каждом исполнении тела цикла попадает не одна бусинка, а две, три или даже больше. Например, если мы хотим занумеровать пять домов на чётной стороне только что отстроенной улицы, можно исполнить такой цикл:

```
Делать от  $k := 2$  до 10 с шагом 2
{ Подойти к очередному дому;
  Присвоить ему номер  $k$ ;
}
```

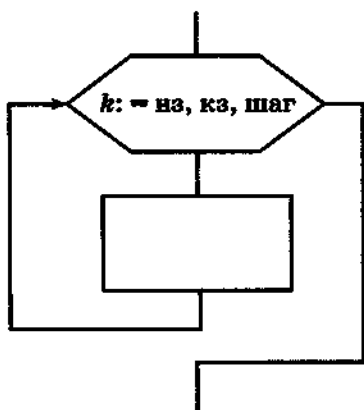


Рис. 2.1. Схема цикла со счётчиком

В результате выполнения этого цикла переменная k последовательно будет

принимать значения 2, 4, 6, 8, 10. Такие номера и получают дома.

Описанная алгоритмическая конструкция называется **циклом со счётчиком**. Схемой цикл со счётчиком изображают обычно так, как показано на рисунке 2.1. Через «нз» и «кз» на рисунке обозначены начальное и конечное значения счётчика. Обратите внимание и на то, что здесь для описания цикла использован не ромб и не прямоугольник, а шестиугольник. Дело в том, что данный блок не только отвечает за проверку условия, но и содержит в себе команду изменения счётчика цикла. Вот и получилась «сумма» прямоугольника с ромбом.

На рисунке 2.2 представлена схема алгоритма нумерации домов.

Если значение счётчика цикла после очередного исполнения тела цикла увеличивается на 1, то в заголовке цикла шаг можно не указывать. В этом случае и в схеме алгоритма величину шага тоже не указывают.



Рис. 2.2. Схема алгоритма нумерации домов на чётной стороне

Подведём итоги

- 1 Если заранее известно, сколько раз должно повторяться тело цикла, то полезно использовать алгоритмическую конструкцию «цикл со счётчиком».
- 2 В операторе цикла со счётчиком обязательно указывается начальное значение счётчика цикла и конечное и может указываться шаг изменения счётчика. В их качестве может выступать любое целое число или значение арифметического выражения.
- 3 Исполнение тела цикла со счётчиком прекращается, как только значение счётчика цикла стало больше конечного значения. Изменение счётчика цикла происходит сразу после окончания очередного исполнения тела цикла на величину, указанную шагом. Если шаг в заголовке цикла не указан, то значение счётчика увеличивается на 1.

Вопросы и задания

- 1 Когда целесообразно использовать цикл Делать от ... до ...?
- 2 Определите, сколько раз исполняется тело цикла в каждом из случаев.
 - а) Делать от $K := -1$ до 7 с шагом 2
{ действие;
...
действие;
}
 - б) Делать от $K := -5$ до -7 с шагом -1
{ действие;
...
действие;
}
 - в) Делать от $K := 8*2 - 5*3$ до $(-3)*4 + 7*2$
{ действие;
...
действие;
}
- 3 Может ли случиться так, что тело цикла со счётчиком не исполнится ни разу? Если может, то укажите, когда такое может быть; если нет, то объясните почему.
- 4 Придумайте 2—3 алгоритма для себя, в качестве формального исполнителя, с использованием операторов Делать пока, Делать от ... до
- 5 Формальный исполнитель умеет складывать любое число с числом, записанным на доске, и результат записывать на доске. Кроме того, он понимает все определённые нами конструкции. Напишите для него программу:
 - а) сложения всех чисел от 1 до 20;
 - б) сложения всех чётных чисел от 2 до 50.
- 6 В восьмидесятиквартирном доме 4 подъезда, по 20 квартир в каждом. Почтовые ящики всех квартир одного подъезда располагаются на первом этаже. Изобразите схему алгоритма раскладки почты в этом доме. (Совет: воспользуйтесь двойным циклом в форме Делать от ... до)
- 7 а) Сколько раз исполняется цикл в данном алгоритме?

Алгоритм

```
{ Положить  $k$  равным 1;  
  Делать пока  $k^3 - 3k^2 < 1$   
    { Сообщить  $\sqrt{k^2 + k + 1}$ ;  
      Увеличить значение  $k$  на 1;  
    }  
}
```

б) используя пункт а), составьте алгоритм, решающий ту же задачу, но с циклом Делать от ... до

- 8 Исполнитель записывает цепочку символов по следующему алгоритму:

Алгоритм

```
{ Делать от  $k = 1$  до 4  
  { Приписать справа к цепочке её первый символ;  
    Приписать слева получившуюся цепочку;  
    Стереть последний символ в полученной цепочке;  
  }  
}
```

Первоначально записана цепочка АБВ. Каким будет символ, стоящий в полученной цепочке на 33-м месте?

- 9 Какой орнамент выложит Паркетчик на квадратном поле из 36 клеток, исполнив данный ниже алгоритм?

Алгоритм

```
{ Делать от  $n := 2$  до 6  
  { Положить (к);  
    Перейти на (n; n);  
  }  
  Положить (к);  
  Перейти на (1; 6);  
  Делать от  $n := 2$  до 6  
  { Положить (з);  
    Перейти на (n; 7 - n);  
  }  
  Положить (з);  
  Перейти на (1; 4);  
  Делать пока (не з)  
  { Положить (к);  
    Шаг вправо;  
  }  
}
```

10 Исполнитель РОБОТ перемещается по клетчатому полю, на котором между клетками могут стоять перегородки. Выходить за края поля и проходить сквозь перегородки РОБОТ не может. Пусть стрелка $\uparrow$ означает команду «переместиться на одну клетку вверх», стрелка $\downarrow$ означает команду «переместиться на одну клетку вниз», стрелка $\leftarrow$ — «переместиться на одну клетку влево», стрелка $\rightarrow$ — «переместиться на одну клетку вправо»; других команд у РОБОТа нет. На поле, изображённом на рисунке 2.3, РОБОТ исполняет следующий алгоритм:

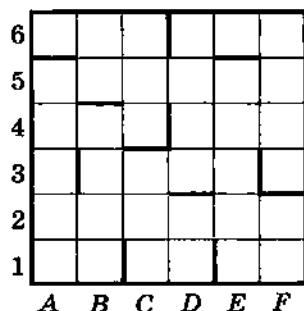


Рис. 2.3

Алгоритм

```
{ Делать от  $k := 1$  до  $t$ 
  {  $\uparrow$ ;
  }
  Делать от  $k := 1$  до  $n$ 
  {  $\rightarrow$ ;
  }
  Делать от  $k := 1$  до  $p$ 
  {  $\downarrow$ ;
  }
  Делать от  $k := 1$  до  $q$ 
  {  $\leftarrow$ ;
  }
}
```

а) Какое наибольшее значение может иметь сумма $t + n + p + q$, для которой РОБОТ, начав исполнение *предложенного* алгоритма с клетки A1, успешно его закончит?

б) Выполните то же задание, если РОБОТ начинает исполнение алгоритма с клетки B1.

11 На поле, изображённом на рисунке 2.3, РОБОТ исполняет следующий алгоритм:

Алгоритм

```
Делать от  $k := 1$  до  $t$ 
{  $\rightarrow$ ;
  Делать от  $p := 1$  до  $n$ 
  {  $\uparrow$ ;
  }
}
```

- а) Какое наибольшее значение может иметь сумма $m + n$, для которой РОБОТ, начав исполнение предложенного алгоритма с клетки A1, успешно его закончит?
- б) Выполните то же задание, если РОБОТ начинает исполнение алгоритма с клетки A2.

12* Исполнитель записывает цепочку символов, исполняя следующий алгоритм:

Алгоритм

```
{ Делать от  $k := 1$  до 9
  { Приписать к имеющейся цепочке слева число  $k$ ;
    Приписать к имеющейся цепочке справа число  $k + 1$ ;
    Приписать к имеющейся цепочке её же ещё раз;
  }
}
```

Первоначально записана цепочка 135.

- а) Какой будет цифра, стоящая в полученной цепочке на 100-м месте?
- б) Какой будет цифра, стоящая в полученной цепочке на 1000-м месте?
- в) Какой будет цифра, стоящая в полученной цепочке на последнем месте?
- 13** На поле Паркетчика располагаются плитки красного и зелёного цвета. Размер поля: 20×17 клеток. Составьте алгоритм, выполнив который Паркетчик очистит поле от всех плиток.
- 14** Имеется квадратное поле размером 20×20 клеток.
- а) Паркетчику нужно на этом поле выложить орнамент из чередующихся красных и зелёных полос, изогнутых под прямым углом (его левый нижний угол изображён на рисунке 2.4, а). Составьте соответствующий алгоритм.

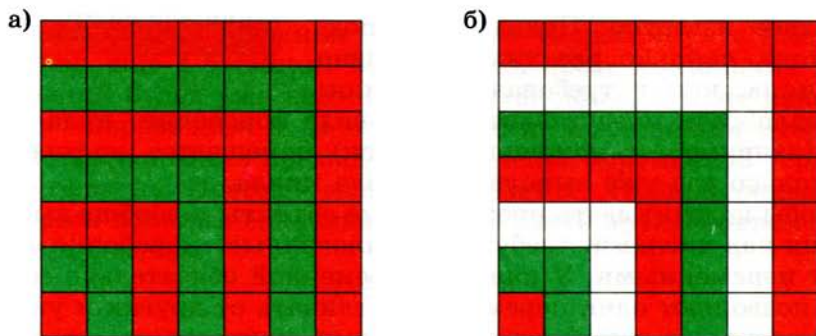


Рис. 2.4

б) Составьте алгоритм для выкладывания орнамента, левый нижний угол которого изображён на рисунке 2.4, б, на первоначально чистом поле.

15 Разделите мысленно поле размером 24×16 клеток на четыре равные части. В левой нижней четверти (размером 12×8 клеток) выложен некоторый орнамент из красных и зелёных плиток.

а) Напишите программу для Паркетчика, выполнив которую он выложит точно такой же орнамент в правой верхней четверти.

б) Напишите программу для Паркетчика, выполнив которую он выложит точно такие же орнаменты в остальных трёх четвертях.

§ 3. Переменные в алгоритмах



Для чего нужны переменные в алгоритме?

Как формальный исполнитель различает переменные?

Как переменная получает значение?

Какие операции можно выполнять над переменными?

В предыдущем параграфе вы познакомились с такой формой цикла, при которой исполнитель просто подсчитывает, сколько раз тело цикла уже выполнено. Оказалось очень удобным, когда количество исполнений тела цикла заранее известно. Правда, для подсчёта того, сколько раз уже исполнено тело цикла, нам потребовался счётчик:

мы предложили представлять его в виде коробочки, куда так или иначе (например, с помощью бусинок) помещается информация о том, сколько раз уже выполнилось тело цикла.

Чтобы подсчитывать произвольные объекты и вообще выполнять действия над числами, требуются аналогичные «коробочки». Их называют переменными. У каждой переменной обязательно есть имя. Имена позволяют одни переменные отличать от других и указывать исполнителю, из какой переменной ему нужно взять информацию для обработки и в какую поместить результат.

Ключевые слова:

переменная;
имя переменной;
значение переменной;
оператор
присваивания

Какие имена можно употреблять, зависит от исполнителя. Большинство компьютерных исполнителей понимает имена, состоящие из букв и цифр (так что никаких фамилий и отчеств у переменных нет).

Информация, которая хранится в переменной, называется её значением.

Самое важное действие над переменной состоит в том, что ей можно присвоить то или иное значение, например 1. Чтобы это сделать, используют оператор присваивания. Мы будем обозначать его символом $:=$. Например, если S — имя некоторой переменной, то указанное действие записывается так:

$$S := 1;$$

Слева от знака присваивания всегда указывается имя переменной, а справа пишется выражение, состоящее из констант и имён переменных, связанных знаками операций, допустимых для данного исполнителя. В выражении могут присутствовать скобки, указывающие на изменение естественного порядка действий.

Если переменная уже имеет какое-то значение, то его можно поменять. Для этого есть несколько путей. Можно просто присвоить переменной новое значение:

$$S := 7;$$

А можно указать, на сколько надо изменить исходное значение:

$$S := S + 4;$$

Последняя запись означает, что к прежнему значению переменной надо прибавить 4 и результат снова присвоить переменной S . Или, говоря нашим прежним языком, в «коробочку» с именем S надо бросить ещё 4 бусинки. Естественно, после нескольких подобных операций будет уже невозможно установить, что же изначально было в «коробочке» S . И вообще, после выполнения очередной операции присваивания предыдущее значение переменной утрачивается безвозвратно! Поэтому если нам хочется, чтобы переменная S помнила своё прежнее значение, то надо взять другую переменную, например, с именем T :

$$T := S + 4;$$

Тогда в S сохранится её исходное значение, а переменная T будет иметь значение, на 4 большее, чем значение S .



Формальные исполнители, реализованные на компьютере, обычно понимают все арифметические действия над числами и умеют их выполнять, причём не только над целыми, но и над дробными числами. Но об этом мы поговорим несколько позже.

Кроме того, среди допустимых действий исполнителя всегда есть сравнение значений переменных (больше, меньше, равно, не равно, не больше, не меньше). Наконец, если нам потребуется, исполнитель сообщит значение интересующей нас переменной. Соответствующее действие мы в алгоритмах будем записывать так:

Сообщить имя переменной;

Можно попросить исполнителя сообщить и некоторый текст. Соответствующее действие записывается так:

Сообщить "текст";

(кавычки подсказывают исполнителю, что здесь записано не имя переменной, а некоторое сообщение, которое надо просто выдать на экран).

Получение информации формальным исполнителем от пользователя мы будем обозначать допустимым действием

Запросить имя переменной;

По этой команде переменной с указанным именем будет присвоено значение, вводимое пользователем.

А теперь рассмотрим задачу, решить которую без переменных весьма затруднительно (впрочем, можете попробовать).

■ Сколько натуральных делителей имеет натуральное число n ?

Идея алгоритма состоит в последовательной проверке, на какие числа, меньшие, чем n , делится n . Нам потребуется переменная, в которой будет подсчитываться количество делителей числа n . Дадим ей имя k .

Алгоритм Число делителей

```
{ Запросить  $n$ ;  
  Если ( $n = 1$ ) то  
  {  $k := 1$ ;  
  }  
  иначе  
  {  $k := 2$ ; (*Ведь два делителя всегда есть: 1 и само число  $n$ *)  
    Делать от  $m := 2$  до  $n - 1$   
    { Если ( $n$  делится на  $m$ ) то  
    {  $k := k + 1$ ;  
    }  
  } (*Конец цикла*)  
}
```

Сообщить k ;

}

Будем надеяться, алгоритм достаточно понятен для вас.

Подведём итоги

- 1 Для работы с информацией исполнитель использует переменные.
- 2 Каждая переменная имеет имя и значение. Имена переменных позволяют отличать их друг от друга, а значение переменной — это та информация, которая подлежит обработке данным исполнителем.
- 3 Чтобы переменная получила или изменила своё значение, используется операция присваивания. Она обычно обозначается $:=$.
- 4 Чтобы использовать переменную, ей прежде всего надо присвоить начальное значение. Это может быть сделано либо с помощью операции присваивания, либо по запросу к пользователю.

Вопросы и задания

- 1 Для чего нужно имя переменной? Что называют значением переменной?
- 2 С помощью какой операции можно изменить значение переменной?
- 3 Чему будут равны a и b после выполнения каждого из следующих алгоритмов? Сколько раз будет выполняться тело цикла в каждом из них?

а) **Алгоритм 1**

```
{ a := -5;  
  b := -2;  
  Делать пока ( $a + b < a * b$ )  
  { a := 2 * a;  
    b := b + 1;  
  }  
}
```

б) Алгоритм 2

```
{ a := 25;  
  b := 1;  
  Делать пока (a - b > a * b)  
  { a := a + 10;  
    b := -3 * b;  
  }  
}
```

в) Алгоритм 3

```
{ a := -3;  
  b := -1;  
  Делать пока (a < b)  
  { Если (b < 3) то  
    { a := a - b;  
      b := b + 2;  
    }  
    иначе  
    { a := a + 2;  
      b := b - a; }  
  }  
}
```

- 4** Коля в алгоритме «Число_делителей» заменил одну строку: вместо

Делать от $m := 2$ до $n - 1$

написал

Делать от $m := 2$ до $n / 2$

Он утверждает, что и его алгоритм тоже находит число делителей числа n . Прав ли Коля?

- 5** В некоторых клетках прямоугольника размером 9×8 клеток проставлены стрелки $\uparrow$ и $\rightarrow$. Исполнитель обрабатывает информацию о стрелках.

Определите, какие значения переменных A и B сообщит исполнитель, выполнив следующий алгоритм на поле, изображённом:

- а) на рисунке 3.1, а;
- б) на рисунке 3.1, б;
- в) на рисунке 3.1, в.

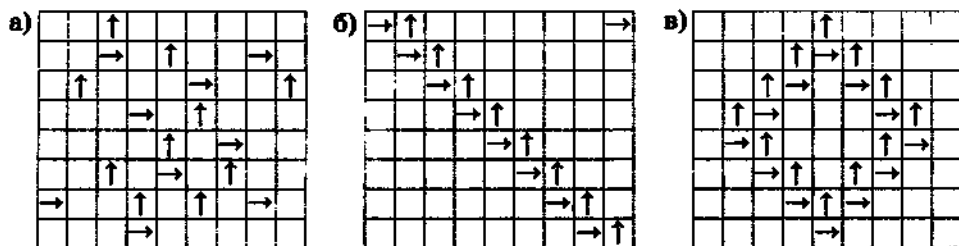


Рис. 3.1

Алгоритм

{ $A := 0$;

$B := 0$;

Делать от $K := 1$ до 8

{ **Делать от $N := 1$ до 9**

{ Перейти на клетку, расположенную в K -й строке и N -м столбце;

Если (в клетке $\uparrow$) **то** { $A := A + 1$; }

Если (в клетке $\rightarrow$) **то** { $B := B + 1$; }

} (*Конец цикла*)

} (*Конец цикла*)

Если ($A > B$) **то** { $B := A + B$; }

Если ($A < B$) **то** { $B := A + B$; }

Сообщить A ;

Сообщить B ;

}

- 6 На поле Паркетчика располагаются плитки красного и зелёного цвета. Требуется составить алгоритм, выполнив который Паркетчик сообщит:

а) количество зелёных клеток на поле;

б) количество непокрытых клеток на поле.

Изобразите схему каждого из алгоритмов и напишите соответствующие программы.

- 7 На поле Паркетчика располагаются плитки красного и зелёного цвета. Размеры поля: 20×17 клеток. Известно, что плиток каждого цвета не более 15. Составьте схему алгоритма и напишите программу, выполнив которую Паркетчик переместит все зелёные плитки в первый ряд, расположив их одна за другой без просветов, а красные — так же во второй ряд.

- 8* На левой половине поля Паркетчика имеется некоторый орнамент из красных и зелёных плиток. Составьте алгоритм,

позволяющий на правой половине поля создать орнамент, симметричный данному относительно вертикальной оси симметрии.

§ 4. Вспомогательный алгоритм



Какую пользу приносит составление вспомогательного алгоритма?

Чем вспомогательный алгоритм отличается от основного?

Что такое подпрограмма?

Ключевые слова:

вспомогательный алгоритм;
подпрограмма;
аргументы вспомогательного алгоритма;
результаты вспомогательного алгоритма

Борьба за экономию времени пронизывает всю историю человеческой деятельности. Для этого придумывались самые разнообразные вспомогательные средства. Есть такое средство и для разработки алгоритмов. Это вспомогательные алгоритмы. Циклов и ветвлений вполне достаточно, чтобы записать любой алгоритм.

Поэтому вспомогательными алгоритмами пользоваться необязательно (ведь и в Москву из Владивостока можно прийти пешком, не пользуясь транспортом). Но они экономят время и силы, уменьшают количество ошибок.

Представьте себе, что вы находитесь в летнем лагере труда и отдыха, вас назначили дежурным по столовой и вы хотите продумать порядок своих действий. Некоторые обязанности дежурного придётся в течение дня исполнять несколько раз. Например:

Убрать со столов посуду.

Вытереть столы.

Расставить стулья.

Уйти.

Эту последовательность действий предстоит выполнить три раза за дежурство: после завтрака, обеда и ужина. Назовём соответствующий алгоритм «Обязанности», а при написании распорядка дня заменим эти четыре действия одной командой:

Исполнить алгоритм Обязанности.

Тогда расписание дежурства по столовой примет такой вид:

Прийти в столовую в 7.00.

Накрыть столы к завтраку.

Дождаться окончания завтрака.

Исполнить алгоритм Обязанности.

Прийти в столовую в 13.00.

Накрыть столы к обеду.

Дождаться окончания обеда.

Исполнить алгоритм Обязанности.

Прийти в столовую в 18.00.

Накрыть столы к ужину.

Дождаться окончания ужина.

Исполнить алгоритм Обязанности.

1092	2
546	2
273	3
91	7
13	13

Рис. 4.1

Легко понять, что использование команды Исполнить алгоритм... позволило почти в два раза сократить длину нашей инструкции.

Подобным образом поступают часто, когда при составлении алгоритма возникает необходимость многократного использования одного и того же набора действий. Этот набор действий выделяют в качестве самостоятельного алгоритма и дают ему имя. С этого момента появляется возможность его использования в других алгоритмах, т. е. он становится вспомогательным алгоритмом. Набор команд, из которых состоит вспомогательный алгоритм, заменяют одной командой, которая называется командой вызова вспомогательного алгоритма.

Составляя для исполнителя вспомогательные алгоритмы, мы как бы обучаем исполнителя новым действиям. Точно так же происходит и обучение в школе. Вначале школьников обучают сложению и вычитанию, затем, используя эти умения, умножению и делению. Умножение и деление, в свою очередь, используются в дальнейшем при решении более сложных задач.

Раз уж речь зашла о школе, поговорим о разложении натурального числа на простые множители. Вспомните, как вы это делали на уроках математики. Сначала находили первый простой множитель и на него делили исходное число. Потом у получившегося числа снова находили простой множитель и делили на него и т. д., пока не оставалось простое число. На рисунке 4.1 показан этот процесс в том виде, как его обычно оформляют в школьной тетради, — в правом столбце оказываются перечисленными все простые множители исходного числа.

Ясно, что находить простой делитель нам приходится несколько раз, а потому вполне разумно этот процесс описать в виде самостоятельного алгоритма. Чтобы его составить, воспользуемся идеей, которая, по существу, уже применялась в предыдущем параграфе,

когда записывали алгоритм подсчёта числа делителей у натурального числа n . Идея состоит в том, чтобы, перебирая натуральные числа одно за другим, найти самое маленькое из них, которое делит данное нам число. Вот этот алгоритм:

Алгоритм Простой множитель

```
{  $m := 2$ ;  
  Делать пока ( $n$  не делится на  $m$ )  
  {  $m := m + 1$ ;  
  } (*Конец цикла*)  
  Сообщить  $m$ ;  
}
```

- Обдумайте, как применять этот алгоритм, и поясните, чего в нём не хватает.

Давайте проанализируем, как используется этот алгоритм. Мы сообщаем исполнителю исходное число, скажем 1092, и он, работая по представленному алгоритму, разыщет самый маленький делитель этого числа, отличный от 1. Чтобы найти простой множитель для другого числа, исполнителю надо сообщить это другое число. Но как исполнитель узнает, для какого именно числа ему надо исполнить алгоритм?

Как видите, между вспомогательными алгоритмами «Обязанности» и «Простой множитель» есть существенная разница. При обращении к алгоритму «Простой множитель» необходимо указать исходное число, для которого требуется его исполнить. А в алгоритме «Обязанности» нет исходных данных. Поэтому ничего, кроме названия, в команде вызова указывать не надо.

При использовании вспомогательного алгоритма, как правило, никого не интересует, из каких действий он состоит. Важно только, каковы исходные данные (аргументы) этого алгоритма и что является результатом его работы. Точно так же для того, чтобы пользоваться стиральной машиной, не обязательно вникать в её устройство. Достаточно знать, что «аргументами» для неё являются стиральный порошок, чистая вода и грязные вещи, а «результатом» — грязная мыльная вода и чистые вещи.

Что касается алгоритма «Простой множитель», то его аргументом является число, а результатом — найденный простой множитель.

Значит, записывая вспомогательный алгоритм, естественно, кроме названия, указывать переменные, служащие аргументами и результатами. Например, запись вспомогательного алгоритма «Простой множитель» можно начать так:

Алгоритм Простой множитель (arg: n ; рез: m)

Эту запись будем называть заголовком вспомогательного алгоритма. А действия, из которых состоит вспомогательный алгоритм, будем заключать в фигурные скобки. Фигурные скобки показывают, что весь вспомогательный алгоритм может рассматриваться как один большой оператор, состоящий из нескольких операторов. Совокупность операторов вспомогательного алгоритма, заключённая в операторные скобки, называется телом вспомогательного алгоритма.

При вызове вспомогательного алгоритма необходимо указать значения исходных данных. Например, обращение к вспомогательному алгоритму «Простой множитель» можно записать так:

Вызвать Простой множитель (1092, k);

Тогда алгоритм будет исполнен при $n = 1092$, а найденный простой множитель будет помещён в переменную k .

Вернёмся к обсуждению исходной задачи — поиску разложения натурального числа на простые множители. Ясно, что такое разложение существует только для чисел, больших 1. А дальше действуем по «школьной схеме». Получаем следующий алгоритм:

Алгоритм Разложение на простые множители

```
{ Запросить  $n$ ;  
  Если ( $n = 1$ ) то  
  { Сообщить "Разложения не существует";  
  }  
  Делать пока ( $n > 1$ )  
  { Вызвать Простой множитель ( $n, m$ );  
     $n := n / m$ ;  
  }  
}
```

Здесь необходимо упомянуть об одном очень важном обстоятельстве. Представьте себе, что вы подарили своему товарищу вспомогательный алгоритм «Простой множитель» и объяснили, как им пользоваться. Разумеется, вашему товарищу нет никакого дела до того, какие обозначения вы использовали в вашем алгоритме. Поэтому может случиться, что в его основном алгоритме будут использоваться те же обозначения, что и в теле алгоритма «Простой множитель», например m (именно это произошло в только что написанном алгоритме).

Понятно, что надо как-то различать переменные, одинаково обозначенные в основном и вспомогательном алгоритмах. Поэтому договариваются считать, что все обозначения из вспомогательного алгоритма действительны только в пределах этого алгоритма.

Образно говоря, переменные, имеющие одно и то же название в основном и вспомогательном алгоритмах, просто «однофамильцы». При вызове вспомогательного алгоритма значения переменных основного алгоритма как бы замораживаются, а после окончания работы вспомогательного алгоритма они размораживаются.

Чтобы алгоритм был понятен тому или иному формальному исполнителю, его надо записать на языке команд этого исполнителя. Алгоритм превращается в программу. Вспомогательный алгоритм, записанный на языке формального исполнителя, называется **подпрограммой**.

- Как вы думаете, почему возникли именно такие названия: «вспомогательный алгоритм» и «подпрограмма»?

Подведём итоги

- 1 При разработке алгоритмов для решения сложных задач полезно использовать вспомогательные алгоритмы.
- 2 Вспомогательный алгоритм имеет заголовок, в котором указывается его имя и имена переменных, которые служат аргументами и результатами вспомогательного алгоритма.
- 3 Обращение к вспомогательному алгоритму осуществляется с помощью команды вызова. В ней указывается, значения каких переменных должны быть взяты в качестве аргументов вспомогательного алгоритма и в какие переменные следует записать результаты исполнения вспомогательного алгоритма.
- 4 Переменные, которые используются в теле вспомогательного алгоритма, могут иметь те же имена, что переменные в основном алгоритме, но их значения независимы друг от друга.

Вопросы и задания

- 1 Какая информация указывается в команде вызова вспомогательного алгоритма?
- 2 В чём заключается «замораживание» и «размораживание» переменных при использовании вспомогательного алгоритма?

3 Приведите примеры создания вспомогательных алгоритмов для известных вам формальных исполнителей.

4 Формальный исполнитель умеет:

брать бумажку с написанным на ней числом;
записывать одно число с бумажки на доску;
сравнивать два числа (определять, какое больше) — число на бумажке или на доске;
стирать число с доски.

а) Напишите для него алгоритм поиска максимального числа из трёх чисел.

б) Используя алгоритм, созданный при выполнении пункта а, как вспомогательный, напишите алгоритм поиска максимального числа из пяти чисел.

5 Петя составил алгоритм, в котором использовал вспомогательный алгоритм Подсчёт (арг: k, m ; рез: s). Вот этот алгоритм:

Алгоритм

```
{ Запросить  $k$ ;  
  Вызвать Подсчёт ( $k, m, s$ );  
}
```

Укажите ошибку в этом алгоритме.

6 а) Ниже приведён алгоритм мытья посуды после чаепития:

Алгоритм

```
{ Вызвать Мыть (чашки);  
  Вызвать Мыть (блюда);  
  Вызвать Мыть (ложки)  
} (*Конец алгоритма*)
```

Составьте вспомогательный алгоритм Мыть (арг: посуда), если исполнитель понимает такие команды:

открыть кран;
закрыть кран;
взять посуду (по этой команде исполнитель берёт со стола 1 экземпляр посуды);
вымыть посуду под краном (по этой команде исполнитель моет имеющийся у него экземпляр посуды);
положить посуду в сушку (исполнитель кладёт имеющийся у него экземпляр посуды в сушку).

Совет: чтобы исполнитель наверняка перемыл всю посуду, вспомните о циклической форме организации действий.

б) Составьте программу мытья посуды после обеда, использующую подпрограмму Мыть (арг: посуда).



Почему требуется различать переменные по типам хранимой числовой информации?

■ Как в алгоритмах и программах указывается тип переменной?

■ Чем отличаются формальные параметры вспомогательного алгоритма от фактических?

Есть много различных формальных исполнителей алгоритмов, которые умеют работать с числовыми переменными. Все они оперируют с числами примерно одинаково.

Вы, конечно, помните, что каждый исполнитель алгоритмов характеризуется прежде всего набором своих допустимых действий. Надо лишь прояснить одно обстоятельство, связанное с тем, что переменные могут принимать не только значение, равное натуральному числу, но и другие значения.

Вспомним, как записывают числа на уроках математики.

- Небольшие натуральные числа записывают цифрами без всяких дополнительных знаков.
- Дробные числа могут записываться с помощью дробной черты или запятой.
- Большие или, наоборот, маленькие числа часто записывают, используя степень числа 10: например, вместо 20 000 000 000 000 пишут $2 \cdot 10^{13}$, а вместо 0,00000000000015 пишут $1,5 \cdot 10^{-12}$.

Более того, некоторые числа, скажем π или $\sqrt{2}$, цифрами вообще не запишешь.

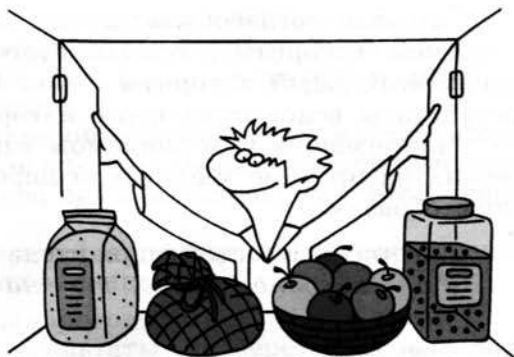
Алгоритмы выполнения действий над числами сильно зависят от того, как эти числа записаны. Например, десятичные дроби складываются совсем не так, как обыкновенные. Поэтому формальный исполнитель должен знать, с числами какого вида он будет иметь дело.

В алгоритме редко присутствует запись самих чисел. Обычно в нём фигурируют переменные, которые, словно пустая тара, всегда готовы принять в себя числа на временное хранение. Но никто не

Ключевые слова:

тип переменной;
объявление переменных;
переменная
вещественного типа;
переменная целого типа;
локальная переменная;
глобальная переменная;
формальные параметры;
фактические параметры;
передача данных
во вспомогательный
алгоритм

хранит подсолнечное масло в бумажных пакетах или сахар в сетках. Каждому виду продукта — свой тип тары. Так же и с переменными: каждому виду чисел — свой тип переменных.



Если формальный исполнитель имеет дело только с одним видом чисел, то можно не указывать тип переменных — и так всё понятно. Обычно исполнители могут работать с несколькими видами чисел. Наиболее распространённые из них — это целые числа и вещественные числа (их ещё называют действительными). Поэтому и переменные бывают нескольких типов. Прежде всего используются целые и вещественные переменные.

Приступая к написанию программы, мы обязаны сообщить исполнителю, каков тип используемых в ней переменных. Эту информацию записывают с помощью специальной конструкции, называемой **объявлением переменных**.

Объявление переменных может, например, выглядеть так:

вещ: *a, b, x, c*, киса;

цел: *n, m, i, j*, вая;

Переменные объявляются до фигурных скобок, ограничивающих тело алгоритма, программы или подпрограммы. Например:

Программа

вещ: *x, c*, киса;

цел: *a, b, i, j*, вая;

{ ...

}

Подпрограмма Нечто (**арг:** **цел:** *c*, вая; **вещ:** *b*; **рез:** **цел:** *x, z*)

вещ: *a, y*;

цел: *n, m, i, j*, киса;

{ ...

}

Стандартные слова **вещ** и **цел**, как обычно, подчёркиваются.

Напомним, что переменные в основном алгоритме и во вспомогательном алгоритме могут иметь одинаковые имена. И они вполне могут быть разного типа. Основной алгоритм и вспомогательный алгоритм могли составлять разные люди, и они не обязаны знать

или заранее согласовывать, как будут называться переменные в основном алгоритме, откуда будет поставляться информация во вспомогательный алгоритм и как будут именоваться аргументы и результаты вспомогательного алгоритма.

Например, в уже знакомом вам алгоритме нахождения разложения на простые множители переменные можно описать следующим образом:

Алгоритм Простой множитель (arg: цел: n ; рез: цел: m)
(* n — исходное число, m — наименьший простой множитель числа n *)
{ $m := 2$;
 Делать пока (n не делится на m)
 { $m := m + 1$;
 } (*Конец цикла*)
 Сообщить m ;
}

Алгоритм Разложение на простые множители

цел: n, m ;
{ **Запросить** m ;
(*Здесь m — исходное число, а n — наименьший простой множитель числа n *)
 Если ($m = 1$) **то**
 { **Сообщить** "Разложения не существует";
 }
 Делать пока ($m > 1$)
 { **Вызвать** Простой множитель (m, n);
 $m := m / n$;
 }
}

Переменные, фигурирующие в заголовке вспомогательного алгоритма, называют **формальными параметрами** — они всего лишь сообщают пользователю вспомогательным алгоритмом, сколько переменных требуется для передачи исходных данных и результатов. Разрабатывая алгоритм, полезно прокомментировать, какая именно информация должна храниться в каждой переменной.

Имена переменных (неважно, аргументы это или результаты), которые фигурируют в операторе **Вызвать**, являются **фактическими параметрами**. Правила присваивания значений формальным параметрам очень просты.

Перед началом выполнения подпрограммы:

если в подпрограмме есть аргументы, то берётся первая переменная в операторе **Вызвать**; её содержимое (т. е. значение) присваивается первой переменной после слова **арг** в заголовке соответствующей подпрограммы;

если в подпрограмме есть второй аргумент, то берётся вторая переменная в операторе **Вызвать**; её содержимое присваивается второй переменной после слова **арг** в заголовке соответствующей подпрограммы;

если в подпрограмме есть третий аргумент, то третья переменная присваивается третьей переменной и т. д.

После окончания работы подпрограммы:

если в подпрограмме есть результаты, то берётся первая переменная после слова **рез** в заголовке соответствующей подпрограммы; её содержимое присваивается следующей переменной, которая осталась после присваивания значений аргументам в операторе **Вызвать**;

если есть ещё результаты, то их значения присваиваются соответствующим переменным.

Количество переменных и их тип в операторе **Вызвать** должны совпадать с количеством переменных и их типом в заголовке соответствующей подпрограммы.

Таким образом, переменные *вася* и *киса*, объявленные дважды в примере описания переменных, именуют совсем разные переменные. Это просто тэски, имеющие (как и у людей) одно имя. Поскольку мы не видим, как выглядит оператор **Вызвать**, то очень может быть, что содержимое *васи* из программы вовсе и не соответствует *васе* из подпрограммы, как могли некоторые из вас ожидать. Ну а раз переменные разные, ничто не мешает им иметь и разные типы.

Но переменные можно объявить в основном алгоритме, работать с ними во вспомогательном алгоритме. Такие переменные называются глобальными переменными. Например, в алгоритме разложения на простые множители переменные можно описать так:

Алгоритм Простой множитель (**рез:** цел: *m*)

```
{ m := 2;  
  Делать пока (n не делится на m)  
  { m := m + 1;  
  } (*Конец цикла*)  
  Сообщить m;  
}
```

Алгоритм Разложение на простые множители

```
цел: n, p;  
{ Запросить n;  
  Если (n = 1) то
```

```

{ Сообщить "Разложения не существует";
}
Делать пока ( $n > 1$ )
{ Вызвать Простой множитель ( $p$ );
   $n := n / p$ ;
}
}

```

Теперь во вспомогательном алгоритме нет аргументов, есть только результаты. Переменная n здесь является глобальной, она используется как в основном, так и во вспомогательном алгоритме. А переменная m — локальная, она может использоваться только во вспомогательном алгоритме.

Мы могли бы отказаться от описания результатов вспомогательного алгоритма и сделать две глобальные переменные — n и m . Может показаться, что так значительно проще — описывать переменные только в основном алгоритме и ни о чём не думать. Но в этом случае вспомогательный алгоритм теряет всю свою привлекательность. Сложный алгоритм может быть собран из различных вспомогательных алгоритмов, как домик из кубиков. Но это можно легко сделать только в том случае, если вспомогательные алгоритмы не используют глобальных переменных. Ведь может случиться так, что в разных вспомогательных алгоритмах используются переменные с одинаковыми именами или вспомогательный алгоритм может содержать очень большое количество глобальных переменных.

Использование параметров во вспомогательном алгоритме помогает не вникать в то, из чего состоит тело этого вспомогательного алгоритма, а использовать его как уже готовый блок для строительства новых алгоритмов.

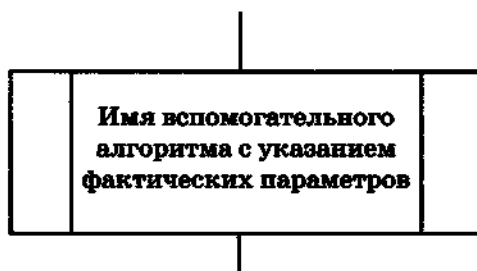


Рис. 5.1

В схеме алгоритма вызов вспомогательного алгоритма обозначается блоком, изображённым на рисунке 5.1. Фактические параметры указываются в этом блоке, а формальные параметры — в начальном блоке вспомогательного алгоритма.

Подведём итоги

- 1 Для работы с данными разного вида требуются переменные разных типов. Для переменных, в которых хранится числовая информация, обычно используются целый и вещественный типы. Тип используемой в алгоритме переменной должен быть объявлен до тела алгоритма.
- 2 Переменные, объявленные во вспомогательном алгоритме, являются локальными — их тип и значения действительны только в пределах исполнения вспомогательного алгоритма. Если переменная объявлена в основном алгоритме, но не объявлена во вспомогательном, хотя там используется, то она является глобальной переменной.
- 3 Переменные в заголовке вспомогательного алгоритма являются формальными параметрами. Они указывают на то, как внутри вспомогательного алгоритма будут называться переменные, в которые передаётся информация из основного алгоритма (аргументы вспомогательного алгоритма), и откуда будет браться информация, полученная после исполнения вспомогательного алгоритма (результаты вспомогательного алгоритма).
- 4 Для передачи информации во вспомогательный алгоритм и получения его результатов в команде вызова вспомогательного алгоритма указываются фактические параметры — имена переменных основного алгоритма или арифметические выражения.

Вопросы и задания

- 1 Что такое тип переменных? Переменные каких типов могут использоваться исполнителями алгоритмов?
- 2 Что такое формальные и фактические параметры?
- 3 Чем отличаются локальные переменные от глобальных?
- 4 Как оформляется конструкция объявления переменных?
- 5 В записи алгоритма вычисления значения выражения

$$(x - 2)/(x^3 - x + 7)$$

Петя Торопыжкин, как всегда, напутал с операторными скобками. Вот как выглядит его алгоритм:

Алгоритм

вещ: x, y ;

```
{ Запросить  $x$ ;  
  Если  $(x^3 - x + 7 = 0)$  то { Сообщить "Не вычисляется!"; }  
  иначе {  $y := (x - 2)/(x^3 - x + 7)$ ; }  
  Сообщить  $y$ ;  
}
```

Исправьте программу, поставив правильно операторные скобки.

- 6 а) Напишите программу определения максимального числа из двух чисел.
б) Используя результат пункта а как подпрограмму, напишите программу определения максимального числа из трёх чисел.
- 7 Многочлен $x^3 + 3x - 1$ на отрезке $[0; 1]$ имеет корень (попытайтесь объяснить почему). Для его вычисления Коля составил следующую программу:

Программа

```
вещ:  $x$ ;  
{  $x := 0$ ;  
  Делать пока  $(x^3 + 3 * x - 1 < 0)$   
  {  $x := x + 0.01$ ;  
    } (*Конец цикла*)  
  Сообщить "Корень равен",  $x$ ;  
}
```

- а) Определите, с какой точностью вычисляется корень по Колиной программе.
- б) Петя, списывая у Коли этот алгоритм, по невнимательности переставил местами две строки. Тем не менее и по Петинему алгоритму компьютер вычислил корень. Какие две строки оказались переставленными?
- 8 Составьте алгоритм, исполняя который компьютер запросит три числа и сообщит то из них, которое на числовой оси расположено между двумя другими.
- 9 Составь алгоритм, в котором будут запрошены три положительных числа — длины отрезков и сообщено, можно ли из этих отрезков построить треугольник. (Совет: полезно воспользоваться программой нахождения максимального числа из трёх заданных чисел, которую вы написали, выполняя задание 6б).

§ 6. Подпрограммы в работе Паркетчика



Однажды пришёл заказчик и заказал паркет (рис. 6.1). Фантазия у заказчика, прямо скажем, небогатая, но сколько придётся попотеть, пока запрограммируешь выкладывание каждого цветного квадрата! А не выделить ли нам выкладывание квадрата в подпрограмму и каждый раз, обращаясь к подпрограмме, указывать, с какого места рисовать квадрат и каким цветом?

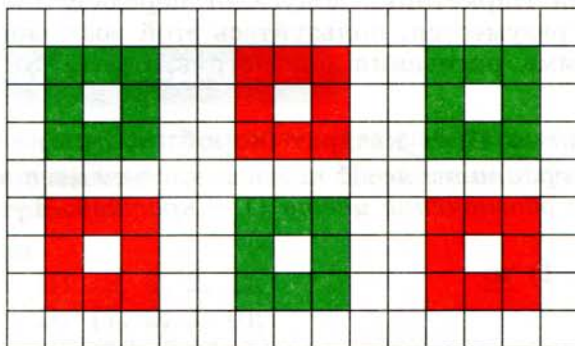


Рис. 6.1

Значит, наш исполнитель Паркетчик должен уметь работать с переменными по крайней мере двух типов: обозначающих целое число и цвет. Мы уже договорились, что для описания целочисленных переменных мы будем применять сокращение цел. Для переменных, содержащих информацию о цвете, будем в описаниях применять слово цвет (и, как всякое зарезервированное слово, будем его подчёркивать). Для имён переменных будем использовать латинские буквы.

Паркетчик — очень молодой исполнитель, он и считать-то научился только в этой главе. Для него слова «аргументы» и «результаты» звучат пугающе. Гораздо понятнее, что исходные данные ему надо складывать в коробочку (вспомните: мы уже говорили, что переменную удобно представлять себе как коробочку, в которую складывают бусинки). И чтобы не мучиться с окончаниями — то ли будет использоваться одна коробочка, то ли их будет несколько, — в программах для Паркетчика будем вместо сокращения арг писать короб. А вот результатов в подпрограммах для Паркетчика вообще нет, ведь результатом его работы является орнамент, выложенный на поле.

Паркетчик был придуман одним из авторов этого учебника в 1993 г. Компьютеры тогда были не чета нынешним. Конечно, по мере совершенствования компьютеров менялись и программные реализации Паркетчика. Его интерфейс становился более удобным, расширялись возможности. Тем не менее вполне вероятно, что в версии, которая используется у вас в классе, Паркетчик умеет передавать в подпрограмму значения только числовых переменных. Как же тогда передать информацию о цвете? Очень просто: надо закодировать цвет целым числом. Договоримся, к примеру, что зелёный цвет кодируется числом 0, а красный — числом 1. В нашем учебнике мы будем писать программы, придерживаясь этой договорённости. Если же ваша версия Паркетчика допускает передачу цвета в качестве параметра, то, разумеется, пользуйтесь этой возможностью.

Подпрограмма рисования цветного квадрата тогда может выглядеть так:

Подпрограмма Цвет_квадрат (короб: k, m, s)

(* k, m — координаты левой нижней клетки цветного квадрата;
 s — кодовое обозначение цвета (1 — красный, 0 — зелёный)*)

цвет: s ;

{ **Если** ($s = 1$) **то**

{ $c := k$;

}

иначе

{ $c := s$;

} (*Декодирование цвета*)

Перейти на (k, m);

Положить (c);

Шаг вверх;

Положить (c);

Шаг вверх;

Положить (c);

Шаг вправо;

Положить (c);

Шаг вправо;

Положить (c);

Шаг вниз;

Положить (c);

Шаг вниз;

Положить (c);

Шаг влево;

Положить (c);

}

Основную программу теперь написать совсем нетрудно:

Программа

```
{ Вызвать Цвет_квадрат (2, 2, 1);  
  Вызвать Цвет_квадрат (7, 2, 0);  
  Вызвать Цвет_квадрат (12, 2, 1);  
  Вызвать Цвет_квадрат (2, 6, 0);  
  Вызвать Цвет_квадрат (7, 6, 1);  
  Вызвать Цвет_квадрат (12, 6, 0);  
}
```

Вот и всё!

Вопросы и задания

1 С переменными каких типов умеет работать Паркетчик?

2 а) Какой орнамент выложит Паркетчик, исполнив данную ниже программу?

Программа

```
{ Вызвать ЦГ (2, 2, 2, 1);  
  Вызвать ЦГ (7, 2, 2, 0);  
  Вызвать ЦГ (12, 2, 2, 1);  
  Вызвать ЦГ (2, 6, 2, 0);  
  Вызвать ЦГ (7, 6, 2, 1);  
  Вызвать ЦГ (12, 6, 2, 0);  
}
```

Подпрограмма ЦГ (короб: k, m, n, s)

цвет: c ;

цел: i ;

```
{ Если ( $s = 1$ ) то  
  {  $c := k$ ;  
  }  
  иначе  
  {  $c := 3$ ;  
    (*Декодирование цвета*)  
  }  
  Перейти на ( $k, m$ );  
  Положить ( $c$ );  
  Делать от  $i := 1$  до  $n$   
  { Перейти на ( $k, m + i$ );  
    Положить ( $c$ );  
  }  
}
```

```

Перейти на  $(k, m)$ ;
Делать от  $i := 1$  до  $n$ 
{ Перейти на  $(k + i, m)$ ;
  Положить  $(c)$ ;
  Перейти на  $(k + i, m + n)$ ;
  Положить  $(c)$ ;
}
Шаг вниз;
Делать пока (не  $c$ )
{ Положить  $(c)$ ;
  Шаг вниз;
}
}

```

б) Сравните получившийся орнамент с тем, который изображён на рисунке 6.1.

в) Напишите программу, которая с использованием той же подпрограммы ЦГ позволит Паркетчику выложить орнамент, изображённый на рисунке 6.2.

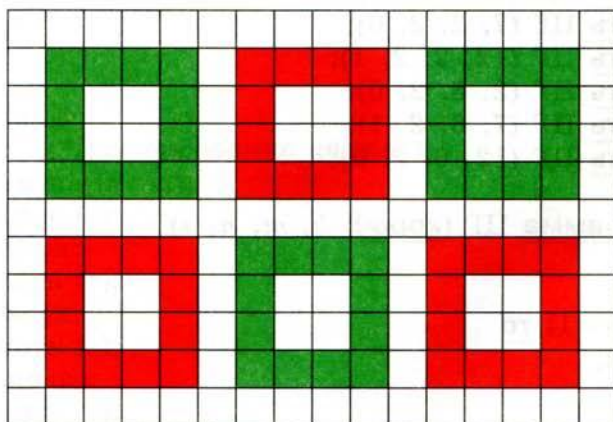


Рис. 6.2

г) В чём, на ваш взгляд, преимущества подпрограммы ЦГ перед подпрограммой Цвет_квадрат?

3 Напишите программу, выполнив которую Паркетчик выложит орнамент, изображённый на рисунке 6.3.

Какую подпрограмму (и может быть, не одну!) целесообразно выделить при написании этой программы?

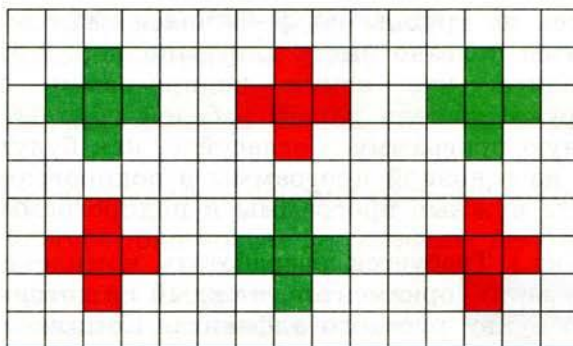


Рис. 6.3

- 4** Какой орнамент выложит Паркетчик, исполнив данную ниже программу?

Программа

```
{ Вызвать Линия (1);
}
```

Подпрограмма Линия (короб: k)

цел: i ;

```
{ Делать от  $i := 1$  до 5
  { Перейти на ( $i$ ,  $k$ );
    Положить ( $z$ );
  }
}
```

Если (не сверху стена) то

```
{ Вызвать Линия ( $k + 1$ );
}
```

```
}
```

- 5** На рисунке 6.4 показан орнамент из красных плиток, который изображает слово **РОБОТ**. Требуется составить программу для Паркетчика, исполнив которую он выложит этот орнамент.

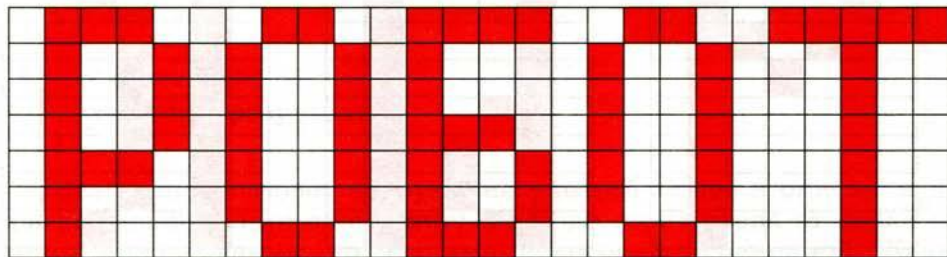


Рис. 6.4

- а) Разбейтесь на группы по 2—3 человека и обсудите, какие подпрограммы полезно иметь для выполнения этого задания.
- б) Распределите, кто какую подпрограмму будет писать. Выберите руководителя вашей рабочей группы, он будет писать основную программу. Согласуйте, как будут передаваться параметры из основной программы в подпрограмму.
- в) Напишите нужные программы и подпрограммы.

6 (Мини-проект.) Требуется разработать комплекс подпрограмм по выкладыванию орнаментов, каждый из которых изображает какую-либо букву русского алфавита. Комплекс должен быть таким, чтобы с его помощью можно было писать программу выкладывания орнамента в виде слова.

Для выполнения этого проекта разбейтесь на группы по 3—4 человека.

а) Для букв **М**, **Ы** и **Ю** вы можете воспользоваться изображениями, приведёнными на рисунке 6.5. Для других букв придумайте изображения самостоятельно и обсудите предложенные варианты в группе. Возможно, вы решите, что не все буквы должны быть одного цвета.

б) Распределите, кто какую подпрограмму будет писать, и согласуйте, как будут передаваться параметры из основной программы в подпрограмму. Напишите соответствующие подпрограммы.

в) Напишите программы для рисования орнаментов, изображающих слова **ТОПОТ**, **ШОРТЫ**, **КОМИК**.

г) Придумайте ещё 2—3 слова, состоящие из 5—7 букв, и напишите для них программы рисования этих слов.

д) Сравните результаты работы разных групп.

(Совет: предварительно сформулируйте критерии, по которым будет происходить сравнение, например эстетический вид получившихся орнаментов, рациональность и экономичность комплекса подпрограмм и т. п.)

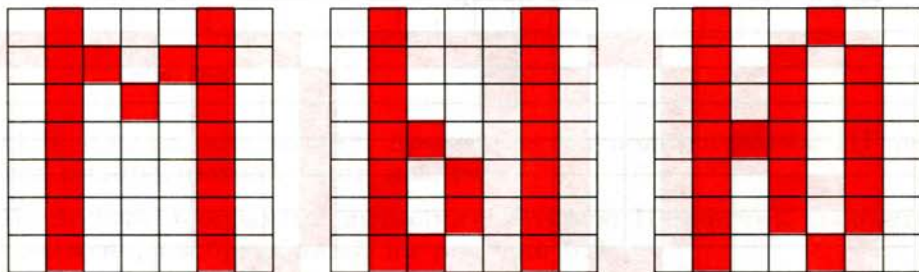


Рис. 6.5

§ 7. Подпрограмма-функция

(для дополнительного чтения)



Как более удобно оформить вспомогательный алгоритм?

Как научить исполнителя новым допустимым действиям?

Вспомогательный алгоритм и подпрограмма — мощные средства, облегчающие решение задач. Но всегда хочется иметь более комфортный способ решения. Когда нам требуется найти значение функции $\sqrt{x}$, мы пишем $y := \text{sqrt}(x)$, а не вызываем подпрограмму вычисления этой функции. На самом деле подпрограмма вызывается (потому что компьютер умеет выполнять только основные арифметические операции; всё остальное — это набор стандартных подпрограмм, встроенных в язык программирования), но мы этого не ощущаем и воспринимаем вычисление такой функции как просто ещё одно допустимое действие исполнителя. Но хочется и самому уметь создавать подобные функции и пользоваться ими на уровне команды исполнителю. Инструментом для реализации такой возможности служит подпрограмма-функция.

Во-первых, подпрограмма-функция, как и любая подпрограмма, имеет имя, посредством которого к ней можно обратиться. Во-вторых, она имеет набор аргументов, которые описываются именами переменных. А вот переменной для результата у неё нет, поскольку вычисляемое ею значение всегда тут же присваивается переменной, указанной пользователем.

Договоримся объявлять подпрограмму-функцию следующим образом:

```
Функция Имя: <тип> (арг: <типы аргументов>: <список переменных>): <тип> <список локальных переменных>
{ <действие>;
  ...
  <действие>;
  знач := <результат>;
}
```

Возможно, вызовет недоумение употреблённое в описании подпрограммы-функции слово **знач**. Раз оно стоит слева от знака операции присваивания, то это имя некоторой переменной. Данная переменная — тёмная лошадка, она служит сугубо внутренним це-

Ключевые слова:

подпрограмма-
функция

лям подпрограммы-функции, но внутри тела подпрограммы-функции тоже может быть использована как полноправная переменная. Именно её тип указывается после имени процедуры.

Чтобы воспользоваться подпрограммой-функцией, достаточно написать команду

$$z := \text{Имя}(a_1, a_2, \dots, a_n);$$

где $a_1, a_2, \dots, a_n$ — значения аргументов, для которых требуется вычислить значение функции.

Иногда для двух целых чисел, отличных от 0, требуется найти их наибольший общий делитель, сокращённо НОД. Напомним, что **наибольшим общим делителем** двух ненулевых целых чисел называется такое целое число, которое делит каждое из этих чисел (т. е. является их общим делителем) и среди всех общих делителей является наибольшим. Легко понять, что наибольший общий делитель всегда является натуральным числом.

Составим подпрограмму-функцию, вычисляющую НОД для двух целых чисел.

Функция НОД: цел (арг: цел: x, y)

цел: m, n ;

{ $m := \text{ABS}(x)$;

(*Функция ABS находит абсолютное значение числа x *)

$n := \text{ABS}(y)$;

Делать пока (не $(m = n)$)

{ **Если** $(m > n)$ **то** { $m := m - n$; }

иначе { $n := n - m$; }

}

знач $:= m$;

}

Сам алгоритм нахождения НОД двух целых чисел, который мы оформили в виде подпрограммы-функции, был предложен великим математиком древности Евклидом и носит его имя.

Подведём итоги

- 1 Подпрограмма-функция является разновидностью вспомогательного алгоритма. Она удобна, когда результатом является одна величина, значение которой присваивается нужной переменной из основного алгоритма.
- 2 Подпрограмму-функцию можно рассматривать как новое допустимое действие исполнителя.

- 1 В чём состоит отличие подпрограммы-функции от обычной подпрограммы? В каких случаях удобно её применять?
- 2 Формальный исполнитель не должен вникать в предложенный ему алгоритм. В тексте параграфа мы предложили алгоритм вычисления НОД двух целых чисел, не обосновывая его работоспособность.
 - а) В роли формального исполнителя выполните этот алгоритм для пар чисел 6 и 4, 9 и 12, 8 и 24.
 - б)* Нам вовсе не хочется, чтобы вы ощущали себя только формальными исполнителями. Поэтому постарайтесь обосновать, что алгоритм Евклида действительно вычисляет НОД. (Совет: полезно воспользоваться следующим свойством для натуральных чисел a и b : если $a > b$, то $\text{НОД}(a, b) = \text{НОД}(a - b, b)$. Конечно, это свойство тоже нуждается в обосновании.)
 - в)* Постарайтесь объяснить, почему алгоритм Евклида конечен при любых допустимых начальных значениях аргументов. (Совет: заметьте, что при каждом исполнении тела цикла уменьшается значение по крайней мере одной из локальных переменных. Может ли это происходить бесконечное число раз?)
- 3 а) Дана бесконечная десятичная дробь $0,123456789101112\dots$, которая образована натуральными числами, выписанными подряд. Составьте подпрограмму-функцию, которая по натуральному числу n вычисляет цифру, стоящую на n -м месте после запятой.
 - б) Решите такую же задачу, как в пункте а, для дроби $0,14916253649\dots$, образованной квадратами последовательных натуральных чисел, выписанными подряд.
- 4 Напомним, что простым называется натуральное число, большее 1, и имеющее ровно два делителя: 1 и самого себя. Первое простое число — это 2, второе — 3, третье — 5, четвёртое — 7, пятое — 11 и т. д.
 - а) Составьте подпрограмму-функцию, которая по натуральному числу n вычисляет n -е простое число.
 - б)* Дана бесконечная десятичная дробь $0,23571113\dots$, которая образована простыми числами, выписанными подряд. Составьте алгоритм, который по натуральному числу n вычисляет цифру, стоящую в этой дроби на n -м месте после запятой.
- 5 Используя подпрограмму-функцию, вычисляющую НОД двух чисел, составьте подпрограмму-функцию для вычисления НОД трёх целых чисел.

- 6 а) Дробь $\frac{a}{b}$ задана парой натуральных чисел (a, b) . Используя подпрограмму-функцию, вычисляющую НОД двух чисел, составьте подпрограмму-функцию, вычисляющую числитель m и знаменатель n несократимой дроби $\frac{m}{n}$, равной $\frac{a}{b}$.
- б) Дроби $\frac{a}{b}$ и $\frac{c}{d}$ заданы парами натуральных чисел (a, b) и (c, d) . Используя подпрограмму-функцию, вычисляющую НОД двух чисел, составьте подпрограмму-функцию, вычисляющую числитель m и знаменатель n несократимой дроби $\frac{m}{n}$, равной $\frac{a}{b} + \frac{c}{d}$.
- 7 Напомним, что два целых числа называются взаимно простыми, если их НОД равен 1.
- а) Функцией Эйлера от натурального числа n называют количество натуральных чисел, не превосходящих n и с ним взаимно простых. Используя подпрограмму-функцию, вычисляющую НОД двух чисел, составьте подпрограмму-функцию для вычисления функции Эйлера.
- б) Используя подпрограмму-функцию, вычисляющую НОД двух чисел, составьте подпрограмму-функцию для вычисления суммы всех натуральных чисел, не превосходящих натурального числа n и взаимно простых с ним.

§ 8. Массивы



Как исполнители работают с таблицами?

Как поступают, если «плоской» таблицы недостаточно?

Мы уже обсуждали, что нередко ту или иную информацию удобно представлять и обрабатывать в табличной форме. И компьютер в этом всегда готов помочь человеку. Одним из популярных формальных исполнителей, реализованных в виде компьютерной программы, является уже хорошо знакомая вам электронная таблица.

Ключевые слова:

табличная величина;
массив; имя массива;
тип массива;
размерность массива;
элемент массива;
индекс элемента массива

- Вспомните, что такое электронная таблица и как ей давать указания на проведение вычислений.

Можно сказать, что вы использовали исполнителя «Электронная таблица» в режиме непосредственного управления — записали формулу в ячейку, нажали клавишу Enter, и компьютер воспринял это как команду на вычисление значения по данной формуле.

Но можно обрабатывать информацию, представленную в табличной форме, и в режиме программного управления. В этом случае подобные таблицы принято называть **массивами**. Есть ещё одно отличие — в массиве могут храниться данные только одного типа. Можно, например, говорить о массиве, в котором хранятся дни рождения, или массиве, в котором хранятся имена учеников класса. В массив можно поместить таблицу розыгрыша первенства по теннису или расстояния между аэропортами какой-либо страны. Номер строки и номер столбца в таблице называют **индексами** того элемента массива, который стоит в клетке на пересечении данной строки и данного столбца.

Обычно таблица имеет несколько строк и несколько столбцов. Массивы, соответствующие таким таблицам, называют **двумерными**. На рисунке 8.1 показано, как в таблице располагаются элементы двумерного массива $A[1:3; 1:4]$. Если массив представляет таблицу, состоящую из одной строки или одного столбца, то такой массив называют **линейным** или, по-другому, **одномерным**.

Впрочем, массивы могут быть и трёхмерными, и четырёхмерными, и ещё большей размерности — всё зависит от задачи, которую приходится решать.

Итак, **массив** — это набор однотипных данных, снабжённых системой из одного или нескольких индексов, каждый из которых принимает последовательные целые значения. Количество используемых индексов называют **размерностью** массива, тип входящих в него элементов — **типом** данного массива.

Чтобы задать массив, надо указать его **имя**, **тип**, **диапазон** изменения каждого индекса, например **вещ:** $M[1:14; 3:25; 5:70]$. Элемент массива также имеет имя: оно состоит из имени массива и значений индексов, определяющих данный элемент. Договоримся значения индексов указывать после имени массива в квадратных скобках через запятую.

При создании алгоритмов каждый элемент массива можно рассматривать как самостоятельную переменную, именем которой является имя массива с указанием значений индексов данного элемента. Например, в массиве $A[1:3; 1:4]$ элемент, стоящий во второй строке и третьем столбце, — это переменная с именем $A[2, 3]$.

A (1, 1)	A (1, 2)	A (1, 3)	A (1, 4)
A (2, 1)	A (2, 2)	A (2, 3)	A (2, 4)
A (3, 1)	A (3, 2)	A (3, 3)	A (3, 4)

Рис. 8.1. Расположение элементов двумерного массива в таблице

Чтобы заполнить массив нужными значениями, обычно применяют двойной цикл со счётчиком. Например, для заполнения массива $A[1:3; 1:4]$ можно написать:

```
Делать от  $i := 1$  до 3  
{ Делать от  $j := 1$  до 4  
  { Запросить  $A[i, j]$ ;  
  }  
}
```

- Для указанного фрагмента алгоритма определите, какой элемент массива получит значение в первую очередь, какой — во вторую, какой — в третью и т. д.

Для краткости, чтобы каждый раз в алгоритмах не записывать такой двойной цикл, договоримся заполнение массива задавать одним оператором. Например:

Запросить $A[1:3; 1:4]$;

Не забывайте, что после слова **Запросить** надо указать имя и размеры того массива, который нужен.

Заполнение массива с помощью вычислений тоже записывается с помощью двойного цикла и операции присваивания. Например:

```
Делать от  $i := 1$  до 3  
{ Делать от  $j := 1$  до 4  
  {  $A[i, j] := i * i - j + 1$ ;  
  }  
}
```

- По приведённой записи определите, какое значение получит каждый элемент массива $A[1:3; 1:4]$.

Подведём итоги

- 1 Для обработки информации, представленной в табличном виде, используются массивы.
- 2 Массив — это набор однотипных данных, снабжённых некоторым количеством индексов. Количество индексов называют размерностью массива.
- 3 Элемент массива удобно представлять себе как отдельную переменную. Имя такой переменной состоит из имени массива и значений тех индексов, которые указывают положение элемента в массиве.

Вопросы и задания

- 1 Что такое массив? Что называют типом массива? Что такое размерность массива?
- 2 Как связаны имена элементов массива с именем самого массива?
- 3 Какие значения переменных A и B сообщит компьютер, выполнив следующий алгоритм для двумерного массива M , заданного: а) таблицей 8.1; б) таблицей 8.2; в) таблицей 8.3?

Алгоритм

цел: $A, B, i, j, M[1:5; 1:4]$;

{ Запросить $M[1:5; 1:4]$;

$A := 0$;

$B := 0$;

Делать от $i := 1$ до 5

{ Делать от $j := 1$ до 4

{ Если $(M[i, j] = 1)$ то { $A := A + 1$; }

Если $(M[i, j] = -1)$ то { $B := B + 1$; }

}

(*Конец цикла*)

}

(*Конец цикла*)

Если $(A > B)$ то { $B := A + B$; }

Если $(A < B)$ то { $B := A + B$; }

Сообщить A ;

Сообщить B ;

}

Таблица 8.1

0	1	0	-1
1	0	-1	1
2	-1	2	1
-1	0	-1	2
0	1	1	1

Таблица 8.2

-1	0	2	1
1	-1	0	2
0	1	-1	2
2	0	1	-1
-1	0	3	1

Таблица 8.3

1	0	1	2
3	-1	-1	-1
1	-1	1	-1
0	1	2	-1
-1	0	1	2

- 4 Требуется заполнить массив $A[1:6; 1:6]$ целых чисел по правилу $A(k, n) = k^n$. Изобразите схемой алгоритм, позволяющий заполнить массив нужным образом.
- 5 Составьте алгоритм, позволяющий в заданном двумерном массиве найти:

- а) сумму всех элементов массива;
 - б) количество положительных и отрицательных чисел в массиве;
 - в) индексы и величины наименьшего и наибольшего элементов массива.
- 6** Дан трёхмерный числовой массив $A [1:10; 1:10; 1:10]$.
- а) Составьте алгоритм поиска максимального элемента в этом массиве.
 - б) Составьте алгоритм, сообщающий индексы всех наименьших элементов данного массива.
- 7** Дан трёхмерный массив $M [1:100; 1:20; 1:75]$ вещественного типа.
- а) Составьте алгоритм вычисления суммы всех положительных элементов массива.
 - б) Составьте алгоритм, позволяющий выяснить, положительных или отрицательных элементов в массиве больше и на сколько.
- 8** На клетчатой бумаге начерчен прямоугольник. Внутри прямоугольника в каждой клетке написано одно из чисел: -1 , 0 и 1 .
- а) Измените на противоположное то ненулевое число, которое находится в клетке, наиболее удалённой от левой верхней клетки прямоугольника. Составьте соответствующий алгоритм.
 - б) Измените на противоположные числа в тех парах клеток, расстояние между которыми наибольшее. Составьте соответствующий алгоритм.
- 9** Дан одномерный массив, заполненный целыми числами. Требуется построить массив, содержащий только чётные числа из исходного массива, или сообщить, что все числа в исходном массиве нечётны. Составьте алгоритм, позволяющий решить такую задачу.
- 10** Имеется два массива: $A [1:4; 1:5]$ и $B [1:4; 1:5]$.
- а) Заполните массив $C [1:4; 1:5]$, каждый элемент которого равен сумме элементов массивов A и B , стоящих на том же месте, что и элемент массива C . Составьте нужный алгоритм.
 - б) Заполните массив $D [1:4; 1:5]$, каждый элемент которого равен большему из элементов массивов A и B , стоящих на том же месте, что и элемент массива D . Составьте нужный алгоритм.
- 11** Представьте, что одномерный массив «склеен» в кольцо, т. е. после последнего элемента снова идёт первый и т. д. Требуется в таком «кольцевом» массиве каждый элемент заменить полусуммой его двух соседей. Составьте алгоритм, позволяющий решить эту задачу.

§ 9. Знакомство с языками программирования



На каком языке пишут программы для компьютера?

С чего начать изучение языка программирования?

Мы неустанно твердили, твердим и будем твердить, что любой компьютер — это формальный исполнитель со своей специфичной системой команд. При этом подавляющее большинство компьютеров имеют совершенно нечеловеческую систему команд и общаться с ними на их родном языке весьма затруднительно. Поэтому для составления программ используется какой-либо промежуточный язык, более понятный человеку. Таких языков много, но мы расскажем только о двух. Это языки КуМир и Pascal (Паскаль). Если два языка для вас многовато — изучайте один!

Ключевые слова:

язык
программирования

Что значит изучить язык? Для обычного иностранного языка это значит выучить перевод слов и правила их соединения в предложения. То же самое и для языков программирования. К счастью, в языках программирования используется очень мало слов (как правило, чуть больше двух десятков). Правила построения алгоритмических конструкций ветвления, цикла часто совпадают с теми, с которыми вы познакомились в 7 классе. Посмотрим, как они переводятся на языки КуМир и Pascal.

Удобно себе представлять, что каждый из языков программирования — это язык некоторого формального исполнителя, который понимает команды, отдаваемые ему на данном языке. Тогда, как обычно, для исполнителя есть среда, состоящая из тех объектов, над которыми он может выполнять допустимые действия. Если вы ещё раз прочтёте заголовок параграфа, то вам станет ясно, что речь у нас пойдёт об исполнителях, которые должны вычислять. Значит, их объекты — это числа, а допустимые действия — это действия над числами.

Конечно, вычисления — это всего лишь одна из многих разновидностей обработки информации (хотя именно ей компьютер обязан своим названием). Поэтому настоящие языки программирования, как и управляемые с их помощью исполнители, способны отнюдь не только к вычислениям.

Первый язык программирования

Инициатором работы по созданию языков программирования стала знаменитая фирма IBM (International Business Machine Company). С опозданием выйдя на рынок компьютеров, фирма испытывала трудности с их продажей. Для неё было крайне заманчиво предложить компьютер, доступный не только профессиональным программистам, но и научным работникам и инженерам, привыкшим к алгебраическим формулам.

В 1953 г. Джон Бэкус предложил разработать язык, позволяющий резко упростить программирование. Система, названная FORTRAN (FORmula TRANslator — переводчик формул), была готова в апреле 1957 г. и позволяла не только переводить формулы на машинный язык, но и, самое главное, автоматизировала организацию циклов, которые в этом языке записываются почти по-человечески:

```
DO 3 I = 1, 10
```

```
...
```

```
3 CONTINUE
```

Здесь тройкой отмечен последний оператор цикла, поскольку в заголовке имеется ссылка на эту метку.

Успех системы превзошёл все ожидания: уже к 1958 г. более половины программ на компьютерах IBM-704 было создано с помощью языка FORTRAN.

К сегодняшнему дню создано и используется несколько сотен различных языков программирования.

Чтобы описать все возможности какого-либо языка программирования, пишут специальное и довольно большое руководство. Мы же в основном ограничимся лишь простейшими вычислительными «способностями» языков КуМир и Pascal.

Итак, давайте вместе разберёмся, что должен уметь делать вычисляющий исполнитель. Разумеется, ему необходимо уметь получать от нас числовую информацию, сохранять её в памяти, выполнять над ней арифметические операции и сообщать нам о получившихся результатах. Арифметические операции в языках программирования обозначаются, как правило, стандартными математическими знаками: + (сложение), - (вычитание), * (умножение), / (деление). В § 5 вы познакомились с типами числовых переменных.

Но компьютерные программы обрабатывают не только числа. Значит, языки программирования должны допускать и другие типы переменных. Один из них — логический тип. По-другому его называют булевым. Переменная такого типа принимает только два значения: Истина и Ложь. Над логическими переменными можно выполнять логические операции: соединять их союзами и и или, а также ставить перед ними частицу не.

Результат применения этих операций определяется таблицами истинности. В таблице 9.1 приведены обозначения логиче-

ских операций и логических значений, принятые в языках Pascal и КуМир.

Таблица 9.1

Операция или логическое значение	Pascal	КуМир
и	and	и
или	or	или
не	not	не
истина	true	истина
ложь	false	ложь

Логические значения получаются при сравнении значений переменных. К примеру, по команде (записанной по правилам языка Pascal)

$$q := (x > -2) \text{ and } (x < 6)$$

логической переменной q будет присвоено значение true, если числовая переменная x имеет, например, значение 4. Если же x имеет значение -3 , то в q запишется значение false.

Основные операторы языков программирования КуМир и Pascal приведены в таблице 9.2.

И в заключение приведём запись одного и того же алгоритма на обоих языках. Это вычисление члена ряда Фибоначчи по его номеру, который задаётся пользователем программы. Название для вас, безусловно, новое, но, если вы проанализируете результаты, всё станет ясно. Вот сам алгоритм:

Алгоритм

цел: $a1, a2$, Раб, Номер, Счётчик;

{ Запросить Номер;

$a1 := 0; a2 := 1;$

Делать от Счётчик := 2 до Номер

{ Раб := $a1 + a2$;

$a1 := a2$;

$a2 := \text{Раб}$;

}

Сообщить "Член с номером", Номер;

Сообщить "равен", $a2$;

}

Язык записи алгоритмов	КуМир	Pascal
1. Начальные операторы		
Алгоритм {...} (или Программа {...})	алг <имя> нач ... кон	Program <имя>; begin ... end.
2. Описание переменных		
цел: <список переменных> — целочисленные; вещ: <список переменных> — вещественные;	цел: <список переменных> — целочисленные вещ: <список переменных> — вещественные	var <список переменных>: integer; — целочисленные; <список переменных>: real; — вещественные;
лог: <список переменных> — логические	лог: <список переменных> — логические	<список переменных>: boolean; — логические
3. Оператор условия		
Если (<условие>) то { <оператор>; ... } иначе { <оператор>; ... }	Если (<условие>) то <оператор> ... иначе <оператор> ... все	if (<условие>) then begin <оператор>; ... end else begin <оператор>; ... end;
Во всех трёх языках конструкция иначе (else) может отсутствовать. В этом случае в языке Pascal необходимо после оператора end поставить точку с запятой.		
4. Оператор цикла по условию		
Делать пока (<условие>) { <оператор>; ... }	нц пока <условие> <оператор> ... кц	while (<условие>) do begin <оператор> ... end;

Язык записи алгоритмов	КуМир	Pascal
5. Оператор цикла с параметром		
Делать от <переменная> := <арифм. выражение> до <арифм. выражение> с шагом <арифм. выражение> { <оператор>; ... }	нц для <переменная> от <арифм. выраже- ние> до <арифм. выра- жение> <оператор> ... кц	for <переменная> := <арифм. выражение> to <арифм. выраже- ние> do begin <оператор> ... end;
		Если нужен шаг не 1, а - 1, вместо <i>to</i> пишут <i>downto</i>.
6. Подпрограмма		
Подпрограмма <имя> (арг: <список параме- тров>; рез: <список параме- тров>) <описание локальных переменных> { <оператор>; ... }	алг <имя> (арг: <список параме- тров>, рез: <список пара- метров>) нач <описание ло- кальных перемен- ных> <оператор> ... кон	procedure <имя> (<список параме- тров>; <описание локальных переменных>; begin <оператор> ... end;
Обращение к подпрограмме:		
Вызвать <имя подпро- граммы> (<список аргументов и результатов>);	<имя алгоритма> (<список аргументов и результатов>)	<имя> (<список па- раметров>);

Язык записи алгоритмов	КуМир	Pascal
7. Подпрограмма-функция		
Функция <имя>; <тип> (arg: <типы аргументов>; <список переменных> <тип> <список локальных переменных> { <оператор>; ... <u>знач</u> := <результат>; }	алг <тип> <имя> (arg: <тип> <список параметров> нач <описание локальных переменных> <оператор> ... <u>знач</u> := <результат> кон	function <имя> (<список параметров>): <тип значения функции>; <описание локальных переменных>; begin <оператор> ... имя функции := <результат>; end ;
Обращение к подпрограмме-функции:		
z := Имя (<список параметров>);	z := Имя (<список параметров>)	z := Имя (<список параметров>);
8. Операторы ввода и вывода информации		
Запросить <имя переменной>; Сообщить <имя переменной или/и сообщение в кавычках>;	Ввод <список переменных> Вывод <список переменных или/и сообщение в кавычках>	read (<список переменных>); write (<список переменных или/и сообщение в кавычках>);
9. Описание массивов		
<тип>; <имя>[<минимальное значение индекса> : <максимальное значение индекса>, ..., <минимальное значение индекса>: <максимальное значение индекса>];	<тип> таб <имя> [<минимальное значение индекса> : <максимальное значение индекса>, ..., <минимальное значение индекса> : <максимальное значение индекса>]	<имя>: array [<минимальное значение индекса> ... <максимальное значение индекса>, ..., <минимальное значение индекса> ... <максимальное значение индекса>] of <тип>;

Программа на языке КуМир:

алг Фиб

нач цел a1, a2, Раб, Номер, Счётчик

ввод Номер

a1 := 0

a2 := 1

нц для Счётчик от 2 до Номер

Раб := a1 + a2

a1 := a2

a2 := Раб

кц

вывод Номер, "-й член равен ", a2

кон

Программа на языке Pascal:

program fib;

var a1, a2, n, r, i : integer;

begin

readln(n);

a1 := 0;

a2 := 1;

for i := 2 *to* n *do*

begin

r := a1 + a2;

a1 := a2;

a2 := r;

end;

write (n, '-й член равен ', a2);

end.

Ни один из языков в написании своих операторов не различает маленькие и большие буквы, поэтому пишите так, чтобы программа для вас выглядела красиво.

Не ленитесь в составляемой программе писать комментарии. Вам они облегчат проверку и тестирование, а другим людям, использующим эту программу, — её понимание.

Подчеркнём ещё раз, что оба языка (КуМир и Pascal) отнюдь не исчерпываются вышеприведёнными конструкциями. Но мы и не ставим перед вами цель овладеть ими в совершенстве. Того «ядра», которое описано, вам с лихвой хватит для старта в освоении языков программирования.

Вопросы и задания

- 1 Что значит изучить язык программирования? Перечислите основные алгоритмические конструкции и запишите их на том языке, который вы изучаете.
- 2 Почему бывают переменные разных типов? Зачем нужно указывать тип переменной? Какие типы числовых переменных вам известны? О каком нечисловом типе рассказано в этом параграфе?
- 3 Напишите какую-нибудь несложную программу на одном из доступных вам языков программирования.
- 4 Выполнив задание 8 из § 5, вы составили алгоритм, по которому из трёх запрошенных у пользователя чисел будет определено среднее из них. Напишите программу, реализующую этот алгоритм, на одном из доступных вам языков программирования.
- 5 Напишите программу, реализующую алгоритм Евклида для нахождения наибольшего общего делителя двух натуральных чисел. (Описание этого алгоритма приведено в § 7.)
- 6 Напишите программы, реализующие алгоритмы, составленные вами при выполнении заданий 5, а—в из § 8.

Возможные проекты

- 1 По подсчётам Института систем информатики им. А. П. Ершова СО РАН, в мире сейчас используется более 1000 различных языков программирования. Некоторые из них универсальны, т. е. могут использоваться для решения очень широкого круга задач, другие специализированы под задачи какой-то определённой области, например управления роботами. Попробуйте выяснить, какими бывают специализации у языков программирования.
- 2 Один из языков программирования называется Whitespace — дословно «Белое пространство», в переносном смысле — «Чистый лист». Найдите объяснение этому названию. Для каких целей был разработан этот язык? Подготовьте небольшое сообщение об этом языке. Найдите какую-нибудь программу, написанную на нём, и продемонстрируйте её исполнение.

Проверь себя

Часть 1. При выполнении предложенных ниже заданий запишите в тетради шифр задания и номер правильного ответа. По окончании работы сверьте получившуюся у вас запись с ключами, приведёнными в конце учебника.

А1. Некоторый исполнитель из одной цепочки символов получает новую цепочку по следующему правилу: сначала записывается исходная цепочка символов, к ней справа приписывается исходная цепочка символов в обратном порядке, затем последний символ стирается и вместо него записывается какой-нибудь символ, которого не было в исходной цепочке. Например, из цепочки СИЛА может получиться цепочка СИЛААЛИМ.

Дана цепочка символов КИТ. Она преобразована указанным образом, и к полученной цепочке снова применили то же правило. На 11-м месте получившейся после этого цепочки стоит символ

- 1) К;
- 2) И;
- 3) Т;
- 4) не встречавшийся в исходной цепочке.

А2. Имеются две кучки камней, в одной 15 штук, в другой 17 штук. Исполнитель имеет два допустимых действия:

Взять из первой кучки 2 камня и переложить их во вторую.

Взять из второй кучки 5 камней и переложить их в первую. И умеет проверять, остались ли в кучке камни.

Исполнитель выполняет алгоритм:

Делать пока ((в первой кучке есть камни) и (во второй кучке есть камни))

```
{ Взять из первой кучки 2 камня и переложить их во вторую;  
  Взять из второй кучки 5 камней и переложить их в первую;  
}
```

Через несколько шагов исполнитель прекратил работу, потому что:

- 1) кончились камни в первой кучке;
- 2) кончились камни во второй кучке;
- 3) не может исполнить первое допустимое действие;
- 4) не может исполнить второе допустимое действие.

А3. Имеется две кучки камней. Допустимое действие исполнителя: если в кучке чётное число камней, то половину этих камней переложить в другую кучку.

Исполнитель выполняет алгоритм:

Делать пока ((в первой кучке чётное число камней) или (во второй кучке чётное число камней))

```

{ Если (в первой кучке чётное число камней) то
  { Половину камней из первой кучки переложить во вто-
    рую кучку;
  }
  Если (во второй кучке чётное число камней) то
  { Половину камней из второй кучки переложить в первую
    кучку;
  }
}

```

Через несколько шагов исполнитель закончил выполнение ал-
горитма. Первоначально могло быть:

- 1) могло быть в первой кучке 12 камней, во второй кучке 6 камней;
- 2) могло быть в первой кучке 5 камней, во второй кучке 10 камней;
- 3) могло быть в первой кучке 9 камней, во второй кучке 10 камней;
- 4) должно быть иное количество камней в кучках, нежели ука-
занное в пунктах 1—3.

А4. Высказывание $(B > A + B)$ и не $(A/B = A - B)$ истинно для:

- 1) $A = -0,5$; $B = 0,5$;
- 2) $A = 0,5$; $B = 1,5$;
- 3) $A = -1,5$; $B = -0,5$;
- 4) ни один из вариантов пунктов 1—3 не годится.

А5. Высказывание

не (первая буква имени = третья буква имени *или*
вторая буква имени согласная)

истинно для имени:

- 1) Нина; 2) Владимир; 3) Павел; 4) Елена.

А6. Сложное условие выражено через высказывания A и B так,
что имеет место следующая таблица истинности

A	B	$?$
истина	истина	истина
ложь	ложь	ложь
ложь	истина	истина
истина	ложь	истина

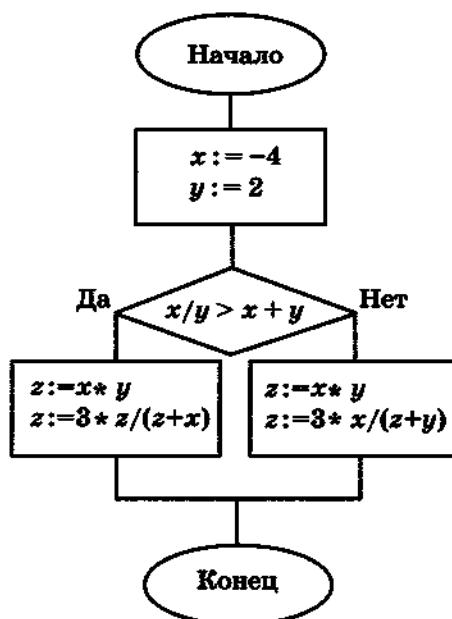
Тогда в этом сложном условии высказывания A и B соединены
 1) союзом *и*; 2) союзом *или*; 3) частицей *не*;
 4) не верен ни один из ответов, указанных в пунктах 1—3.

А7. Логическое выражение *не (не A или B)* равносильно выражению:

- 1) A или *не* B ; 2) A и *не* B ;
 3) *не* A и B ; 4) *не* A или *не* B .

А8. Значения переменных y и z после исполнения алгоритма, изображённого на рисунке, равны:

- 1) $y = 2$; $z = 1$; 2) $y = -8$; $z = 1$;
 3) $y = 2$; $z = 2$; 4) $y = -8$; $z = 6$.



А9. Дан фрагмент алгоритма:

$a := 1$;

$b := -1$;

Делать пока $(a + b > a * b)$

{ $a := b + 1$;

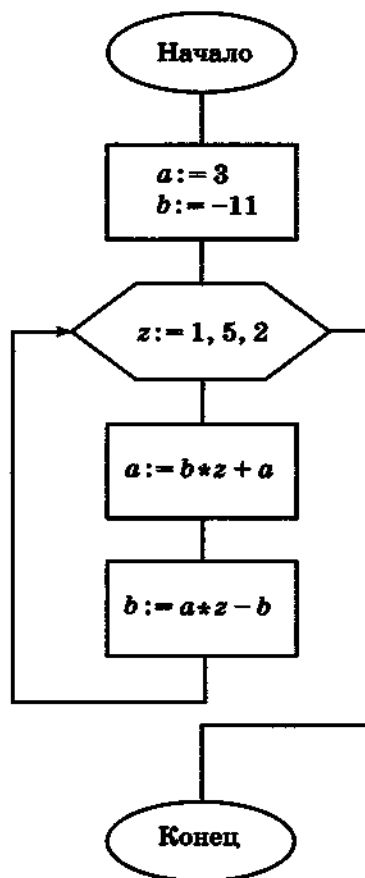
$b := 2 * a$;

}

Тело цикла будет исполнено:

- 1) 0 раз;
- 2) 1 раз;
- 3) 3 раза;
- 4) более 3 раз.

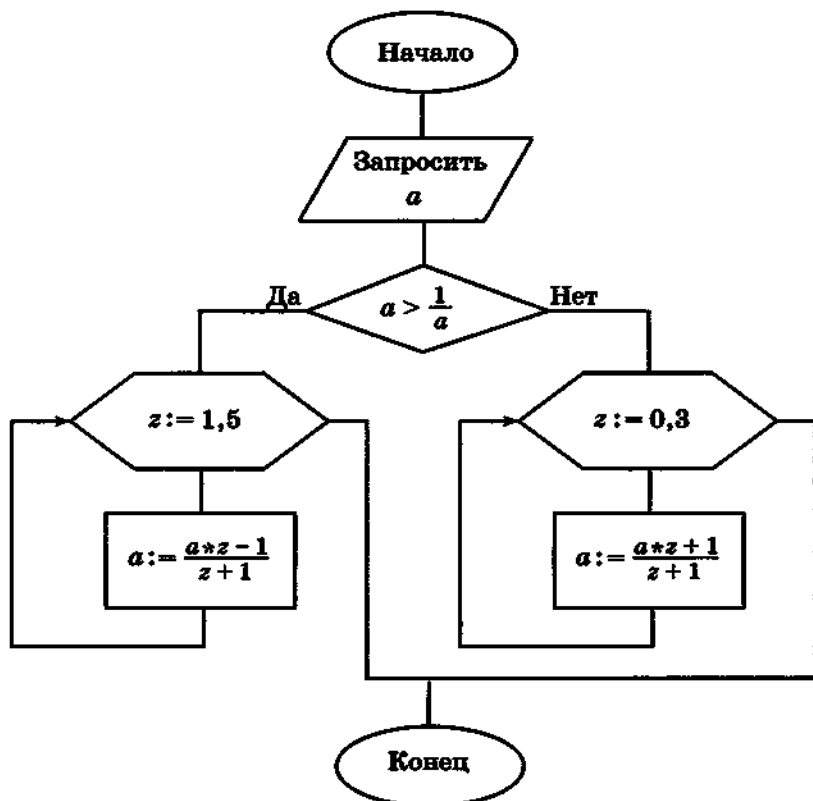
A10. На рисунке изображена схема алгоритма.



Значения переменных a и b после исполнения этого алгоритма равны:

- 1) $a = -144$, $b = 112$;
- 2) $a = 1$, $b = 5$;
- 3) $a = -29$, $b = 11$;
- 4) не совпадают ни с одной парой значений, указанных в пунктах 1—3.

A11. После исполнения алгоритма, изображённого на рисунке, переменная a получила значение 1.



Значение, присвоенное переменной a в результате исполнения первого оператора алгоритма,

- 1) обязательно было равно 1;
- 2) могло быть равно -1;
- 3) обязательно было равно 2;
- 4) нельзя указать однозначно.

A12. Формальные параметры вспомогательного алгоритма — это:

- 1) переменные, используемые во вспомогательном алгоритме;
- 2) аргументы вспомогательного алгоритма;
- 3) количество переменных в заголовке вспомогательного алгоритма;
- 4) переменные, указанные в заголовке вспомогательного алгоритма.

A13. Значения целочисленных переменных x , y и z после исполнения фрагмента программы

КуМир	Паскаль
<pre> x := -3 y := 4 Если x * y < x - y то z := x * x + y * y y := z - y x := x * y + z иначе z := x * 2 + y * 3 y := z - y * y x := x * y + z все </pre>	<pre> x := -3; y := 4; if x * y < x - y then begin z := x * x + y * y; y := z - y; x := x * y + z; end else begin z := x * 2 + y * 3; y := z - y * y; x := x * y + z; end; </pre>

равны:

- 1) $x = -38$; $y = 21$; $z = 25$;
- 2) $x = 36$; $y = -10$; $z = 6$;
- 3) $x = 13$; $y = 21$; $z = 25$;
- 4) не совпадают ни с одной тройкой значений, указанных в пунктах 1—3.

A14. Значения целочисленных переменных x и y после исполнения фрагмента программы

КуМир	Паскаль
<pre> x := 3 y := 5 нц пока (x / y + y / x < y) y := y + 1 x := 2 * x кц </pre>	<pre> x := 3; y := 5; while (x / y + y / x < y) do begin y := y + 1; x := 2 * x; end; </pre>

равны:

- 1) $x = 96$; $y = 10$;
- 2) $x = 192$; $y = 11$;
- 3) $x = 48$; $y = 9$;
- 4) не совпадают ни с одной парой значений, указанных в пунктах 1—3.

A15. Значения целочисленных переменных x , y и z после исполнения фрагмента программы

КуМир	Паскаль
$x := 7$ $y := 3$ нц для z от 1 до 4 $x := x + y * z$ $y := y - z$ кц	$x := 7;$ $y := 3;$ for $z := 1$ to 4 do begin $x := x + y * z;$ $y := y - z;$ end;

равны:

- 1) $x = 2, \quad y = -7, \quad z = 5;$ 2) $x = 37, \quad y = -1, \quad z = 4;$
- 3) $x = 2, \quad y = -7, \quad z = 4;$
- 4) не совпадают ни с одной тройкой значений, указанных в пунктах 1—3.

A16. Массив — это:

- 1) последовательность переменных;
- 2) набор однотипных переменных, снабжённых системой целочисленных индексов;
- 3) расположение элементов в виде одной или нескольких строк;
- 4) в пунктах 1—3 нет правильной формулировки.

A17. Размерность массива — это:

- 1) количество индексов, используемых для описания массива;
- 2) наибольшее значение какого-либо индекса массива;
- 3) количество элементов в массиве;
- 4) в пунктах 1—3 нет правильной формулировки.

A18. Дан алгоритм:

Алгоритм

вещ: $X, Y, M[1:10];$

цел: $K;$

{ $X := (M(1) + M(10) + \text{ABS}(M(1) - M(10))) / 2;$

$Y := (M(1) + M(10) - \text{ABS}(M(1) - M(10))) / 2;$

Делать от $K := 1$ до 10

{ **Если** $(Y < M(K) < X)$ **то**

{ **Сообщить** $M(K);$

} **(*Конец ветвления*)**

} **(*Конец цикла*)**

}

Этот алгоритм был исполнен для массива M :

-1,2	-0,5	3,6	-1,2	1	2,1	1,9	0	-1,5	3
------	------	-----	------	---	-----	-----	---	------	---

После исполнения сообщено:

- 1) 1 число; 2) 3 числа; 3) 5 чисел; 4) 6 чисел.

A19. Дан алгоритм:

Алгоритм

вещ: $X, Y, M[1:10]$;

цел: K ;

{ $X := M(1)$;

$Y := M(1)$;

Делать от $K := 2$ до 10

{ Если $(M(K) < X)$ то

{ $X := M(K)$;

} (*Конец ветвления*)

Если $(M(K) > Y)$ то

{ $Y := M(K)$;

} (*Конец ветвления*)

} (*Конец цикла*)

Сообщить $Y - X$;

}

Этот алгоритм был исполнен для массива M , заданного таблицей:

1,2	-0,5	3,6	0,8	-1	2,1	1,9	0	-1,5	3
-----	------	-----	-----	----	-----	-----	---	------	---

Во время исполнения алгоритма было сообщено число:

- 1) 2,1; 2) 5,1; 3) -1,5;
4) другое, нежели указанное в пунктах 1—3.

A20. Дан алгоритм:

Алгоритм

цел: M ;

{ Запросить M ;

Если $(M < 0)$ то { $M := -M$; }

Делать пока $((M - 1) * (M - 8) > 0)$

{ $M := M + 1$;

}

}

Количество целочисленных значений M , при которых данный алгоритм конечен, равно

- 1) 1;
- 2) 9;
- 3) 17;
- 4) другому числу, нежели указаны в пунктах 1—3.

Часть 2. При выполнении предложенных ниже заданий запишите в тетради шифр задания и рядом ответ в виде последовательности символов. По окончании работы сверьте получившуюся у вас запись с ключами, приведёнными в конце учебника.

B1. У исполнителя Утроитель две команды, которым присвоены номера:

1. Вычти 2.
2. Умножь на 3.

Первая из них уменьшает число на экране на 2, вторая — утраивает его. Запишите порядок команд для получения числа 26 из числа 2 в алгоритме, содержащем не более 8 команд. В ответе указываются только номера команд. Например, запись 21121 означает алгоритм, который преобразует число 3 в число 13:

Умножь на 3
Вычти 2
Вычти 2
Умножь на 3
Вычти 2

Если алгоритмов больше одного, запишите любой из них.

B2. Исполнитель РОБОТ перемещается по клетчатому полю, на котором между клетками могут стоять перегородки. Выходить за края поля и проходить сквозь перегородки РОБОТ не может. Стрелка $\uparrow$ означает команду «переместиться на одну клетку вверх», стрелка $\downarrow$ означает команду «переместиться на одну клетку вниз», стрелка $\leftarrow$ — «переместиться на одну клетку влево», стрелка $\rightarrow$ — «переместиться на одну клетку вправо»; других команд у РОБОТа нет.

На поле, изображённом на рисунке, РОБОТ исполняет следующий алгоритм:

```

Делать от  $k := 1$  до  $m$ 
{
  ↑;
  Делать от  $k := 1$  до  $n$ 
  {
    →;
  }
}
↑
Делать от  $k := 1$  до  $p$ 
{
  ←;
  ↓;
  ←;
}

```

5						
4						
3						
2						
1						
	A	B	C	D	E	F

Первоначально РОБОТ находится на клетке A1, а значения m , n и p таковы, что после исполнения алгоритма РОБОТ снова оказывается на клетке A1. Укажите значения m , n и p , перечислив их в указанном порядке без пробелов и знаков препинания (например, 121). Если вариантов ответа несколько, укажите любой один из них.

В3. Дан алгоритм:

Алгоритм

```

{
  Задумать число;
  Обозначить его буквой X;
  Если (не  $((X > 150)$  или  $(X < 75))$ ) то
  {
    Делать пока  $(X \text{ чётно})$ 
    {
      Разделить  $X$  на 2 и обозначить полученное число снова
      буквой  $X$ ;
    }
  }
  иначе
  {
    К  $X$  прибавить 2 и обозначить полученное число снова
    буквой  $X$ ;
    Если  $(X < 10)$  то
    {
      От  $X$  отнять 3 и обозначить полученное число снова
      буквой  $X$ ;
    }
  }
  Сообщить  $X$ ;
}

```

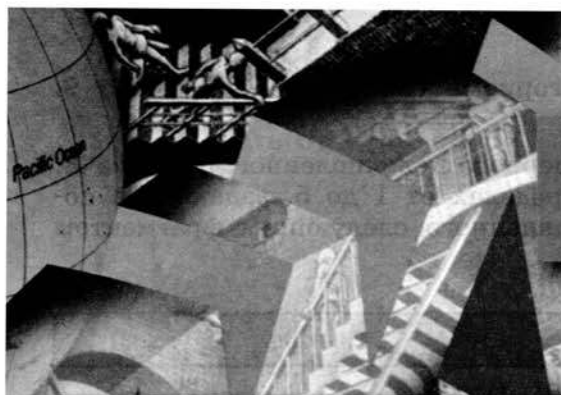
После исполнения этого алгоритма было сообщено число 9. Какое число было задумано?

В4. Значения элементов двумерного целочисленного массива B , индексы которого меняются в пределах от 1 до 5, задаются с помощью двойного цикла, представленного следующим фрагментом программы:

КуМир	Паскаль
<pre>В[1, 1] := 1 нц для k от 1 до 5 нц для n от k до 5 Если (n > 1) то В[k, n] := В[k, n - 1] + n * k В[n, k] := -В[k, n] все кц кц</pre>	<pre>B(1,1) := 1; for k := 1 to 5 do for n := k to 5 do begin if (n > 1) then B(k, n) := B(k, n - 1) + n * k; B(n, k) := -B(k, n); end</pre>

Укажите значение элемента $B(4, 3)$.

В5. Закончите предложение: «Чтобы изменить значение переменной, нужно использовать операцию...»



Глава 2

Человек и информация

Сегодня жизнь каждого человека обусловлена окружающим его быстро меняющимся информационным пространством. Для полноценной жизни в современном обществе каждому приходится принимать решения, учитывая информацию, относящуюся далеко не только к его ближайшему окружению. И наши решения зачастую уже имеют последствия не только для того узкого круга, в котором мы вращаемся, но и для многих людей, удалённых от нас географически или социально. Например, благодаря Интернету тысячи людей с разных концов нашей страны сегодня имеют возможность вступить в прямое общение с Президентом России, и это непосредственно сказывается на ситуации в стране. Конечно, не каждый из вас станет президентом или губернатором, но круг информационной деятельности, которую вы ведёте сегодня и тем более будете вести в будущем, только расширяется. Сегодня взаимодействие человека и информации определяет направление развития каждого из вас и общества в целом. Об особенностях этого взаимодействия и пойдёт речь в данной главе.

§ 10. Информация и её свойства



- Какие свойства информации важны для её использования?
- Какое общество называют информационным?
- В чём состоит принцип информационной открытости?
- Что принято считать информационной революцией?

В 7 классе вы познакомились с тем, как понимают слово «информация» в различных областях науки и человеческой деятельности.

- Вспомните наиболее общее, как говорят, философское понимание слова «информация». Как понимают этот термин в кибернетике? Что понимают под информацией в науках об обществе и человеке?

Прочитав ещё раз название главы, вы поймёте, что чаще слово «информация» мы будем рассматривать как сведения, знания об окружающем человека мире и о самом человеке. Любая целенаправленная деятельность человека неосуществима без информации, и её использование — обязательный элемент этой деятельности. Именно при использовании информации выявляются такие её важные свойства, как новизна, актуальность, достоверность, объективность, ценность, полнота и т. п. И хотя многие из вас, наверно, не раз слышали, а быть может, и сами употребляли указанные эпитеты в сочетании со словом «информация», мы разъясним смысл каждого из этих свойств.

Информация обладает новизной, если её смысловое содержание отличается от смыслового содержания ранее имевшейся информации.

Информация актуальна (иными словами, своевременна), если она оказывает влияние на формирование целенаправленной деятельности именно в данный момент времени.

Информация достоверна, если принимается, что она отражает реальное положение дел, в частности не вступает в противоречие с уже имеющейся информацией, также признаваемой в качестве достоверной. Вовсе не исключается, что с поступлением новой информации данная информация уже перестанет быть достоверной.

Информация объективна, если она не зависит от свойств источника информации. Надо понимать, что абсолютно не зависеть от свойств источника информация не может, однако при тех или иных условиях можно считать, что такое влияние пренебрежимо мало.

Достоверность и объективность информации нередко оцениваются путём сравнения её при получении из разных источников. Обычно информацию считают достоверной, если она подтверждена не менее чем тремя по возможности независимыми источниками. Никакое научное открытие не будет признано достоверным, если оно не подтверждено экспериментами разных исследователей. Важно учитывать возможные цели, с которыми данная информация предоставлена источником, а также различать факт (т. е. объективное изложение информации) и личное мнение (т. е. субъективное

Ключевые слова:

информация;
информационное общество;
принцип информационной открытости;
информационная революция;
информационная культура

Развитие информационного общества в России

7 февраля 2008 г. Президент России В. В. Путин утвердил «Стратегию развития информационного общества в Российской Федерации». Вот некоторые её положения:

«...Информационное общество характеризуется высоким уровнем развития информационных и телекоммуникационных технологий и их интенсивным использованием гражданами, бизнесом и органами государственной власти.

...Развитие информационного общества в Российской Федерации базируется на следующих принципах:

- партнёрство государства, бизнеса и гражданского общества;
- свобода и равенство доступа к информации и знаниям;
- поддержка отечественных производителей продукции и услуг в сфере информационных и телекоммуникационных технологий;
- содействие развитию международного сотрудничества в сфере информационных и телекоммуникационных технологий;
- обеспечение национальной безопасности в информационной сфере.

отношение к нему). К примеру, если команда на соревнованиях заняла четвёртое место, то в одной газете можно прочитать о провале наших спортсменов, не попавших в тройку призёров, а в другой — об их успехе, позволившем им войти в пятёрку сильнейших.

Информация обладает ценностью (или, по-другому, полезностью), если она повышает вероятность достижения цели в целенаправленной деятельности того, кто использует эту информацию.

Информация полна, если её достаточно для достижения цели. Полная информация может быть избыточной, если для достижения цели достаточно только части данной информации. К примеру, в словах русского языка гласные буквы, как правило, несут избыточную информацию, поскольку могут быть однозначно восстановлены по оставшимся согласным буквам (например, слово «ра-дость» однозначно восстанавливается по слову «рдсть»). Это, однако, не означает, что надо стремиться избавиться от избыточности — это свойство часто используется для защиты информации от возможных искажений в процессе передачи информации. Оценить полноту

информации можно, только указав цель, для достижения которой она будет использоваться. В приведённом примере с русским языком целью является восстановление смысла текста. Если же целью была проверка грамотности пишущего (например, во время диктанта), то пропуск буквы «о» делает имеющуюся информацию неполной.

Важно понимать, что перечисленными свойствами информация обладает в рамках конкретно протекающего информационного процесса. Следовательно, все они носят временный, можно даже сказать, сиюминутный характер. Разумеется, какие-то свойства могут сохраняться и длительный промежуток времени, но надо всегда принимать во внимание относительный по времени характер указанных свойств.

Напомним, что информация может использоваться бессознательно (для живых организмов в этом случае говорят — рефлекторно) либо осознанно. Во втором случае информация используется для выработки решения, как следует действовать в имеющейся ситуации.

Вот несколько примеров осознанного использования информации.

Каждый из вас наверняка слышал слово «биржа». Биржи бывают разные. На товарной бирже заключаются сделки на продажу партий товара, на валютной происходит покупка и продажа валют разных стран, наконец, на бирже труда происходит распределение вакансий рабочих мест. Но при этом никто не грузит товар, никто не отсчитывает из бумажника деньги, никто не становится тут же к верстаку или за чертёжный кульман. Что же происходит на бирже? На основе предоставляемой информации принимаются решения на покупку и продажу, на занятие рабочих мест. И информация о произошедших изменениях — товар уже продан, рабочее место уже занято и т. д. — тут же предоставляется всем, кто пришёл на биржу. Обновлённая информация снова используется для принятия решений, например, о повышении или понижении цены.

Информация, добываемая в научных исследованиях, используется для разработки технологий, позволяющих производить товары с меньшими затратами сырья, энергии и трудовых ресурсов. Развитие естественных наук позволяет подчинить воле человека всё большие и большие энергетические ресурсы. Воздействие человека на окружающий мир зачастую носит уже планетарный, а не узконаправленный характер. А значит, чтобы принимать правильные решения, которые не грозили бы общечеловеческой катастрофой, требуется и овладение всё большими информационными ресурсами, и умение их использовать. Вследствие этого всё больше людей и средств вовлекается в сферу информационной деятельности, всё востребованнее становятся информационные технологии. Поэтому, говоря о современном обществе, его часто называют информационным. При этом одним из важнейших принципов становится его информационная открытость. Принцип информационной открытости означает право каждого человека на получение любой информации, кроме той, распространение которой нарушает права личности или приводит к утрате безопасности существования общества. Важную роль в реализации принципа информационной открытости играют Интернет и другие телекоммуникационные технологии.

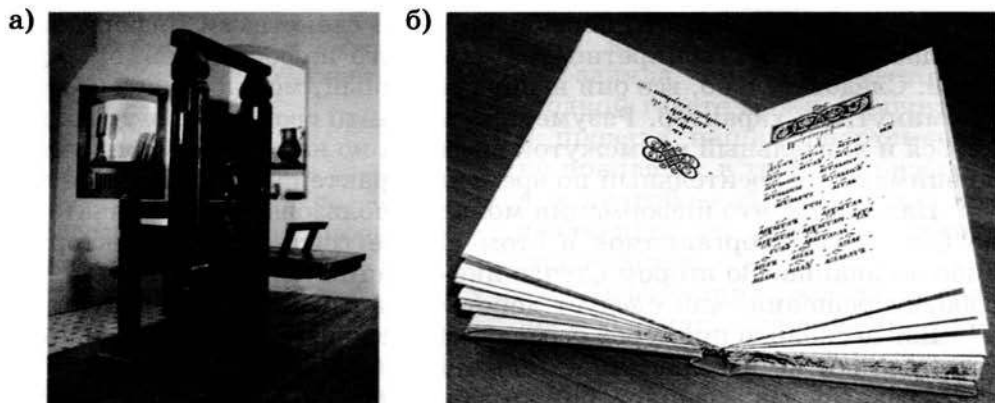


Рис. 10.1. а) Первый печатный станок; б) «Азбука» И. Фёдорова

Способы, посредством которых фиксируется и распространяется информация, во многом определяют информационный «лик» цивилизации. А принципиальные изменения в технологии хранения, передачи и обработки информации нередко называют информационной революцией.

Первой информационной революцией можно назвать изобретение письменности. Именно письменность сыграла решающую роль в организации передачи информации из поколения в поколение: устный пересказ, которым пользовались в дописьменный период, опирался на такое ненадёжное устройство хранения информации, как человеческая память. Гибель человека, весьма нередкая в те времена, и просто забывчивость приводили к безвозвратной потере информации.

Второй информационной революцией по праву считают изобретение книгопечатания. На рисунке 11.1 изображены первый печатный станок и «Азбука» Ивана Фёдорова. Ведь теперь накопленная человечеством информация, представленная в виде текстов, становилась доступной каждому грамотному человеку. Обучение грамоте обрело массовый характер, поскольку появилась возможность создать и издать учебник для обучения не единиц, а тысяч людей.

Во второй половине XX в. выпуск научно-технической печатной продукции стал подобен нарастающей лавине. Ни отдельно взятый человек, ни целые научные коллективы оказались не способны оперативно осваивать весь информационный поток, находя в нём то, что требуется для осуществления тех или иных работ.

И тут на помощь пришёл компьютер. Правда, только компьютеров здесь недостаточно — нужны ещё средства коммуника-

ции, способные доставить информацию от одного компьютера к другому.

Иными словами, нужны компьютерная сеть и программные средства навигации в ней. Создание сообщества сетей, именуемого Интернетом, знаменует собой третью информационную революцию. Сегодня практически вся накопленная человечеством информация оказалась переведённой на электронные носители, а мощные компьютерные станции, объединённые в глобальные сети и снабжённые эффективными средствами поиска информации, способны оперативно доставлять эту информацию пользователю из любого уголка планеты.

Для каждого уровня развития общества ему присуща определённая культура. Это многогранное понятие, отражающее уровень материального производства в обществе, уровень его интеллектуального и духовного развития, способы и формы общения людей. Частью культуры информационного общества является информационная культура каждого человека.

Информационная культура личности — это готовность человека к жизни и деятельности в высокоразвитой информационной среде, умение эффективно использовать её возможности и защищаться от её негативных воздействий.

Поэтому в понятие информационной культуры человека входит:

- понимание закономерностей информационных процессов;
- умение организовать поиск и отбор информации, необходимой для решения стоящей задачи;
- умение оценивать достоверность, полноту, объективность и другие характеристики поступающей информации;
- умение представлять информацию в различных видах, обрабатывать её посредством подходящих информационных (в том числе компьютерных) технологий;
- умение использовать полученную информацию для принятия решений;
- этичное поведение при использовании информации.

Этические нормы в использовании информации существовали всегда. Но в современных условиях, когда информации стало больше, а доступ к ней благодаря глобальным компьютерным сетям стал намного легче, информационная этика обретает новые черты. Подробнее мы это обсудим в последующих параграфах.

Подведём итоги

- 1 Важно понимать степень достоверности, объективности и полноты информации, которую предполагается использовать.
- 2 Достоверность информации заключается в её соответствии реальности.
- 3 Объективность информации — это независимость её содержания от источника информации.
- 4 Информация полна, если её достаточно для достижения поставленной цели.
- 5 Ценность информации определяется тем, насколько она способствует достижению поставленной цели, принятию правильного решения.
- 6 Информационное общество — современный этап развития цивилизации с доминирующей ролью знаний и информации, воздействием информационно-коммуникационных технологий на все сферы человеческой деятельности и общество в целом.
- 7 Жизнь в информационном обществе требует от человека обладания высокой информационной культурой.

Вопросы и задания

- 1 Какие свойства информации необходимо принимать во внимание при её использовании?
- 2 Какая информация называется достоверной? А какая объективной?
- 3 а) Всякая реклама несёт информацию для покупателя. Найдите в газете какую-нибудь рекламу и укажите, какими из перечисленных в объяснительном тексте свойств обладает информация в выбранной вами рекламе.
б) Какими свойствами, на ваш взгляд, должна обладать информация в любой рекламе?
- 4 Может ли быть недостоверной информация, используемая рефлекторно?

- 5** Информация может быть получена непосредственно из наблюдений или из какого-либо хранилища информации. Приведите примеры, когда осознаваемая информация, получаемая из наблюдений, оказывается недостоверной.
- 6** В приведённых ниже примерах определите, полна ли информация для принятия требуемого решения. Если, на ваш взгляд, она не полна, то какую ещё информацию вы бы хотели иметь?
- а) Вашему классу предлагают в ближайшее воскресенье поехать на экскурсию в соседний город. Вам сообщили стоимость и продолжительность экскурсии. Требуется принять решение, заказывать ли эту экскурсию.
- б) Ваша семья собирается приобрести автомобиль. Вы выяснили, автомобили каких марок продаются в доступных вам магазинах и для каждой марки вместимость автомобиля, расход горючего на 100 км пути, мощность двигателя, вид потребляемого топлива, цену автомобиля. Требуется принять решение, какой автомобиль купить.
- 7** В чём суть информационных революций, имевших место в истории развития человечества? Укажите отличительные признаки каждой из них.
- 8** Перечислите составляющие информационной культуры человека.
- 9** В чём заключается принцип информационной открытости общества?

Темы для размышления и обсуждения

- 1** Можно ли использовать недостоверную информацию при планировании будущей деятельности? Если можно, то как уменьшить возможные негативные последствия?
- 2** Является ли компьютерная грамотность, т. е. умение использовать компьютер в повседневной информационной деятельности, компонентом информационной культуры человека современного общества?
- 3** Информация может оказаться бесполезной. А может ли информация быть вредной?
- 4** Почему надо различать свойства «объективность информации» и «достоверность информации»? Может ли достоверная информация быть необъективной? Может ли объективная информация быть недостоверной?

§ 11. Информационная грамотность — базовый элемент культуры человека в современном обществе



Как грамотно работать с информацией?

Что значит быть информационно грамотной личностью?

Как почувствовать себя уверенным в информационном пространстве современного общества? Выход один — овладеть методами работы с информацией.

Работу человека с информацией можно условно разделить на три этапа.

На первом этапе — назовём его стартовым — важно задать самому себе вопрос, для чего мне нужна информация, и сформулировать чёткий ответ. Возможно, вы готовитесь к сдаче экзаменов; возможно, предполагаете участвовать в конкурсе исследовательских проектов; возможно, вы хотите определиться с будущей профессией, а может быть, вам требуется разрешить какие-то личные проблемы. На этом этапе вы активизируете все те знания, которые уже имеются у вас по интересующей проблеме. Именно в этот момент у вас появляется возможность сформулировать информационную потребность, и чем чётче вы её сформулируете, тем легче вам будет найти необходимую информацию.

Второй этап — осмысление полученной информации. И он тоже осуществляется пошагово. На первом шаге происходит восприятие информации, т. е. своеобразная её расшифровка, когда из отдельных частей складывается общее содержание. Затем осуществляется интерпретация полученной информации, при которой происходит упорядочение и классификация, объяснение найденных фактов и их суммирование, различение, сравнение и сопоставление, группировка, анализ и обобщение, соотнесение с собственным опытом, размышление и формулировка выводов. Можно сказать, что на этом этапе происходит извлечение смысла из полученной информации. Третий шаг в осмыслении информации — это выдвижение гипотез и высказывание предположений, формулирование суждений, моделирование ситуаций и обобщение. Здесь человек создаёт новый смысл индивидуальным, только ему одному присущим способом. В ответ на полученную информацию у него возникают соб-

Ключевые слова:

информационная грамотность;
информационная потребность

ственные чувства, мысли, образы. В этой индивидуальности одно из объяснений сосуществования различных точек зрения по многим вопросам.

Перечислим ещё раз шаги, по которым происходит осмысление информации:

- восприятие;
- извлечение смысла;
- создание собственного смысла.

Те, кто на этом этапе останавливается на первом шаге освоения информации, как правило, способны лишь механически воспроизвести содержание. Когда-то этого было достаточно для получения образования (стоит лишь вспомнить классическую зубрёжку). Сегодня, когда необходимо постоянное обновление знаний и образование продолжается в течение всей жизни, на первый план выдвигается творческое освоение информации, которое требует её интерпретации, оценки и создания на этой основе собственных смыслов.

Третий этап — рефлексивный. На этом этапе новый опыт, новое знание встраиваются в систему личностных смыслов, становясь для человека своими.

На каждом этапе работы с информацией человек использует различные операции, приёмы, механизмы работы, объединённые определённой последовательностью и правилами. Вместе они составляют то, что можно назвать стратегией обработки информации. У каждого человека такая стратегия своя, но можно выделить некоторые общие приёмы, которые полезно использовать при построении собственной стратегии.

О методах поиска информации, т. е. о первом этапе работы с информацией, мы расскажем в следующем параграфе. А сейчас рассмотрим методы, которые помогают выполнить второй этап — осмысление информации. Осмысление всегда сопровождается процессом рассуждений. Но что значит рассуждать? Если вы попытаетесь ответить на этот вопрос, то у вас может получиться примерно следующее. Рассуждать — это как бы беседовать с самим собой, задавать себе вопросы и, обдумывая ответы на них, делать выводы и формулировать новые вопросы. Не случайно говорят, что по тому, как человек умеет задавать вопросы, можно определить, как он умеет думать. Большинство людей при работе с информацией ограничиваются лишь простыми вопросами, ответами



Таблица 11.1

Название группы	Характеристика вопросов	Возможные формулировки вопросов
Простые	Относятся к фактографической информации	Кто совершил ...? Когда произошло ...? Где случилось ...? Кем наблюдалось ...? Что это было ...?
Уточняющие	Служат для установления обратной связи с источником информации	Верно ли я понял, что ...?
Интерпретационные	Служат для выяснения причины и раскрытия смысла	Почему случилось ...? Как происходит ...? В чём заключается связь между ...?
Творческие	Содержат элементы условности, фантазии, прогноза	Как могло бы быть, если бы ...? Можно ли изменить так, чтобы ...? Следует ли ожидать ...?
Оценочные	Служат для определения качества получаемой информации	Что привлекательного в ...? Почему есть сомнения в правильности ...?
Практические	Направлены на установление возможности использовать информацию	Где может пригодиться ...? Как использовать ...?

на которые служат некоторые факты, имена, даты и другие сведения. Американский педагог и психолог Бенджамин Блум показал, что, помимо простых вопросов, есть ещё пять групп вопросов. В таблице 11.1 приведены названия этих групп, общая характеристика входящих в них вопросов и примеры.

То, как человек умеет работать с информацией, определяет уровень его информационной грамотности. Под обычной грамотностью мы понимаем умения человека читать, писать, воспринимать графические образы и наличие навыков использования этих умений в бытовой и профессиональной деятельности. А что такое информационная грамотность?

Кратко можно сказать так: информационная грамотность — это способность сформулировать, какая именно информация требуется, умение запрашивать, искать, отбирать, оценивать и по-своему осмысливать, или, как говорят, интерпретировать информацию, в каком бы виде она ни была представлена.

В этом определении присутствуют два очень важных момента.

Первый момент — умение формулировать информационную потребность. Это означает, что вы способны чётко осознать, а затем и выразить словами, где проходит граница между вашим знанием и незнанием, не позволяющая добиться нужного результата. Только после того, как будет осознана и сформулирована информационная потребность, вы сможете построить запрос на поиск необходимой вам информации.

Второй момент — умение интерпретировать полученную информацию. Это означает, что вы не просто законспектировали или даже усвоили добытую вами информацию, а сумели соотнести её со своими знаниями, оценить её соответствие вашим собственным взглядам, сделать из неё выводы и, быть может, измениться сами.

Теперь мы можем более развёрнуто ответить на вопрос, что такое информационная грамотность. Это умение человека:

- осознать и сформулировать потребность в информации для решения той или иной проблемы;
- выработать стратегию поиска информации;
- найти соответствующую информацию;
- оценить качество информации: полноту, достоверность, актуальность, объективность;
- сформировать собственное отношение к этой информации;
- представить (аудитории или самому себе) свою точку зрения, новые знания и понимание или решение проблемы;
- оценить эффективность проделанной работы по следующим параметрам: полученные знания, приобретённые навыки и успешность в решении поставленной задачи;
- осознать, что знания и навыки, полученные в процессе решения данной проблемы (или учебной задачи), можно распространить на другие задачи и даже другие сферы деятельности человека;
- осознать влияние тех знаний, которые были получены в ходе решения задачи, на свои личные позиции и поведение.

Будьте информационно грамотными, научитесь работать с информацией, и это будет залогом принятия правильных решений в самых разных жизненных ситуациях.

Подведём итоги

- 1 По мере развития информационного общества возрастает роль овладения информационной грамотностью каждым человеком, живущим в современных условиях.
- 2 Информационная грамотность — это умение сформулировать информационную потребность, определить направления поиска нужной информации и найти её, осмыслить информацию и выработать на этой основе собственную точку зрения, принять обоснованные решения.

Вопросы и задания

- 1 Почему информационную грамотность следует считать базовым элементом информационной культуры личности?
- 2 Раскройте содержание понятия «информационная грамотность».
- 3 Что такое интерпретация информации?
- 4 Каковы этапы работы с информацией? Раскройте содержание каждого из этапов.
- 5 Ниже процитированы тексты разных авторов и для каждого из них приведены суждения читателей о прочитанном. По этим высказываниям попытайтесь определить, на каком шаге осмысления информации они были сделаны. Обоснуйте свой ответ.
 - а) «Скажи-ка, дядя, ведь не даром
Москва, спалённая пожаром,
Французу отдана?»

(М. Ю. Лермонтов)

Суждения:

- 1) Здесь говорится о том, какой ценой была взята Москва войсками Наполеона.
- 2) Выражена просьба рассказать о событиях Отечественной войны 1812 года.
- 3) Здесь говорится о самоотверженности всего русского народа — как тех, кто сражался на Бородинском поле, так и мирных жителей, которые, пожертвовав всем, фактически не отдали Москву.
- б) «Не только вы собрали книги, но и книги собрали вас».
(В. Шкловский)

Суждения:

- 1) Книги обладают свойством объединять людей.
 - 2) С помощью книг можно найти единомышленников.
 - 3) По книжным собраниям можно судить о внутреннем мире людей.
 - 4) Людей всегда объединяют общие интересы, книги — лучший фундамент для такого объединения.
- 6 В чём, на ваш взгляд, заключается полезность знания групп вопросов, предложенных Б. Блумом?
- 7 Ниже приведён отрывок из книги Шона Кови «7 навыков высокоэффективных тинейджеров: как стать крутым и продвинутым» (М.: Добрая книга, 2006). Прочитав его, попытайтесь сформулировать к содержанию этого текста хотя бы по одному вопросу из каждой группы вопросов, фигурирующих в таблице 11.1.

«Я написал эту книгу, потому что жизнь подростков перестала быть лёгкой и беззаботной. Сегодня это — джунгли. Если я сделал свою работу как следует, эта книга может стать компасом, который поможет тебе выбраться из них.

От реальной жизни не спрячешься, как ни пытайся. Я и не пытаюсь: вместо этого я предлагаю тебе набор инструментов, которые прекрасно помогают справляться с реальной жизнью. Что это за инструменты? Это — семь навыков эффективных тинейджеров, или, другими словами, семь качеств, которыми отличаются все успешные тинейджеры в мире.

Сейчас ты, наверное, гадаешь, что это за навыки... Пожалуй, я перестану держать тебя в напряжении и назову их.

Навык 1. *Будь проактивен*¹.

Возьми на себя ответственность за свою жизнь.

Навык 2. *Начинай, представляя конечную цель.*

Определи свою миссию и цели в жизни.

Навык 3. *Начинай с самого важного.*

Расставляй приоритеты и в первую очередь делай самое важное.

Навык 4. *Действуй по принципу „выиграть — выиграть“.*

Поддерживай установку „выиграть может каждый“.

Навык 5. *Сначала стремись понять, потом — быть понятым.*

Искренне слушай других людей.

¹ Проактивное поведение — сознательный выбор, основанный на внутренних ценностях человека.

Навык 6. *Синергия*¹.

Сотрудничай с другими людьми, чтобы достичь большего.

Навык 7. *„Затачивай пилу“*.

Регулярно обновляй себя».

Тема для размышления и обсуждения

Французский учёный Т. Адорно считает, что под влиянием телевидения сложился новый тип личности: «Знает много, понимает мало, не способен к критической оценке, не имеет собственной точки зрения». Как вы думаете, существует ли защита от такого влияния?

§ 12. Поиск информации



Каковы методы поиска информации?

- Чем поиск информации отличается от других информационных процессов?

Прежде всего давайте вспомним, о каких основных видах информационных процессов шла речь в курсе информатики 7 класса.

- Перечислите основные виды информационных процессов. К какому из них, по вашему мнению, относится поиск информации?

Ключевые слова:

информационный запрос;
адресный поиск;
семантический поиск;
фактографический поиск

Возможно, у вас возникла дискуссия, потому что одни считают, что поиск — это один из вариантов получения информации, другие — что это передача информации, третьи вообще сомневаются, что поиск может рассматриваться как частный случай какого-либо из основных видов информационных процессов. Обратимся к справочной литературе. И вот что мы найдём:

¹ Синергия — эффект от взаимодействия двух и более элементов, результат которого представляет собой больше, чем сумма результатов деятельности отдельных элементов.

«Поиск информации представляет собой процесс выявления в некотором множестве документов (текстов) всех тех, которые посвящены указанной теме (предмету), удовлетворяют заранее определённой условию поиска (запросу) или содержат необходимые (соответствующие информационной потребности) факты, сведения, данные».

Давайте осмыслим информацию, содержащуюся в этом определении.

- Найдите в таблице 11.1 примеры вопросов, которые помогают понять смысл полученной информации. Сформулируйте вопрос, подходящий для данного случая.

Вот один из таких вопросов: «В чём заключается связь между поиском информации и другими видами информационных процессов?» Ответить можно так: поскольку поиск осуществляется в том массиве информации, которая уже получена, зафиксирована на некотором носителе и передана на хранение, то этот информационный процесс — промежуточное звено между информационными процессами хранения и передачи информации.

Следующий вопрос: «Как осуществить поиск информации?» Ответ находим в том же определении: формируется запрос, в котором указывается тема, в чём состоит информационная потребность и каким условиям должна удовлетворять разыскиваемая информация.

Казалось бы, всё просто. Но вот, к примеру, потребовалась информация о Крымской войне в России в 1853—1856 гг. Вы идёте в школьную библиотеку и просите литературу про эту войну. Тут-то и выясняется, что в своём запросе вы пока сформулировали только тему. А для чего вам нужна эта литерату-

**Информационный поиск —
от истоков
к сегодняшнему дню**

Термин «информационный поиск» был введён в 1948 г. Кельвином Муром в связи с осознанием проблемы лавинообразного увеличения количества научной информации. Сложилась парадоксальная ситуация, когда оказывалось быстрее и дешевле повторить то или иное исследование, чем найти о нём информацию в научной литературе.

Интенсивная разработка теории и методов информационного поиска, особенно в связи с проникновением Интернета во все виды человеческой деятельности, привела к созданию целой индустрии в информационной сфере.

Сегодня методами поиска информации овладевают не только специалисты в этой области, но и школьники и студенты. Для них даже проводятся соревнования «Открытый Кубок России по поиску в Интернете».

ра? Вы готовитесь к викторине по истории? Или пишете реферат о том, как эти события отражены в мемуарах современников? Или вам нужен ответ на какой-нибудь совсем конкретный вопрос, скажем, о соотношении вооружённых сил, участвовавших в этой войне? Без точного понимания информационной потребности вы окажетесь перед горой из книг и других источников. Хотя, скорее всего, школьный библиотекарь вас пожалеет и постарается вместе с вами понять, что же вам нужно на самом деле. Ещё необходимо определить, в какой форме вам потребуется информация — в текстовой или в графической, а может быть, вам нужны видеофрагменты или аудиоматериалы. Это позволит сформулировать дополнительные условия, которым должна удовлетворять разыскиваемая информация.

Но в современном обществе нужную информацию чаще разыскивают в Интернете, а не в ближайшей библиотеке. И в таком поиске вам помочь уже некому. Значит, надо уметь самостоятельно правильно формулировать запрос на поиск информации.

Сегодня глобальные компьютерные сети — это не только средство коммуникации, но и почти бездонное хранилище самой разнообразной информации. Объединение же глобальных сетей в Интернет знаменует собой третью информационную революцию, когда практически вся накопленная человечеством информация оказалась переведённой на электронные носители, а мощные компьютеры, объединённые в глобальные сети и снабжённые эффективными средствами поиска информации, способны оперативно доставить эту информацию пользователю из любого уголка планеты.

Но для того чтобы начать поиск информации, нужно сформулировать запрос. Принято выделять три основных типа поиска информации и, соответственно, три типа запросов.

Самый простой, к тому же известный с докомпьютерных времён — это адресный поиск. Это означает, что о разыскиваемом вами документе точно известны какие-то его параметры, например автор, или название, или место публикации и т. п. В этом случае вы просто указываете в запросе эти параметры. Именно для такого поиска предназначены каталоги в библиотеках. Подобные каталоги имеются и для информации, хранимой в Интернете.

Второй тип поиска носит название семантический. Это название происходит от слова «семантика» — смысл, значение. Конечно, смысл того, что вы разыскиваете, надо передать в запросе достаточно точно и кратко. Для этого служат так называемые ключевые слова. Сейчас практически каждую публикацию принято сопровождать указанием набора ключевых слов, отражающих основные вопросы, которым она посвящена. Это существенно облегчает поиск информации. И это важно не только для того, кто разыскивает ин-

формацию, но и для того, кто её разместил, — ведь он хочет, чтобы его информация была востребована, чтобы её нашли те, кому она полезна. Впрочем, даже если ключевые слова не выделены специальным образом, современные автоматизированные поисковые системы (о некоторых из них мы расскажем позже) умеют выделять тексты, в которых указанные в вашем запросе слова могли бы являться ключевыми.

Третий тип поиска — фактографический. Это тот случай, когда вам нужны не столько источники информации, сколько те или иные фактические данные, где бы они ни содержались. Но при этом разыскиваемый вами «просто факт» всё равно где-то содержится, и поэтому вы получите как результат поиска некоторое множество ссылок на документы, в которых он содержится.

Иногда вам может понадобиться найти документ, из текста которого вам известна лишь некоторая цитата. Тогда в запросе вы указываете данную цитату. И это тоже может рассматриваться как некий вариант фактографического поиска.

Далеко не всегда на сформулированный вами запрос вы получите именно ту информацию, которую хотели. Тогда запрос надо уточнить, более точно формулируя тему, или информационную потребность, или дополнительные ограничения поиска. К последним относится, в частности, указание, где именно производить поиск, скажем, на сайтах определённой группы или среди файлов определённого типа. Вполне возможно, что уточнять запрос вам придётся не один раз, пока вы не получите интересующую вас информацию.

Впрочем, может случиться так, что вам потребуется, наоборот, расширить тему запроса. Обычно это происходит тогда, когда требуется найти ответ на вопрос, о котором вы знаете совсем мало. В таком случае полезно начинать с уточнения значения малознакомых вам слов или сферы их применения. Для этого можно обратиться к толковому словарю — в бумажном или электронном виде. В бумажном словаре вы просто находите нужное вам слово — там слова располагаются в алфавитном порядке. А в поисковой системе Интернета вы можете просто написать запрос «Что такое ...». Затем, исходя из выясненного смысла слова, вы сможете более точно сформулировать запрос на поиск нужной вам информации.

В рассмотренных нами типах запросов предполагается, что в них явно указан предмет поиска. Поэтому такие запросы называют явными. Но могут быть и неявные запросы. Например: «Какая сегодня погода?», или «Происходит ли сейчас что-то важное?», или «Что делал Иван-царевич в тридесятom царстве?». Для ответа на подобные запросы разрабатываются специальные инструменты поиска информации. Скажем, текущую информацию о погоде можно найти на сайтах метеорологической службы, а информацию

о последних событиях — на новостных лентах информационных агентств, вот про Ивана-царевича — несколько сложнее... Хотя и в этом случае, переформулировав запрос и сделав его явным, можно добыть нужную информацию.

Как бы точно вы ни пытались составить запрос, почти наверняка вам будут представлены документы, не удовлетворяющие вашу информационную потребность. Информация, соответствующая информационной потребности, называется **релевантной**. Эффективность поиска обычно оценивают по двум параметрам — **полноте** поиска и **точности** поиска.

Полнота поиска — это отношение числа выданных релевантных документов к общему числу релевантных документов, имеющих в распоряжении поисковой системы. В идеале это число должно было бы равняться 1. На практике такое значение почти никогда не достигается (как и положено любому идеалу). В реальных поисковых системах коэффициент полноты поиска может достигать значений 0,7—0,9.

Точность поиска — это отношение числа выданных релевантных документов к общему числу документов, выданных системой по данному запросу. Значение этого параметра колеблется от 0,1 до 1. Как вы понимаете, здесь многое зависит от того, насколько хорошо был составлен запрос пользователем.

И не забывайте, что полученную вами информацию прежде, чем её использовать, нужно проверить на...

- Закончите фразу, перечислив свойства, которыми должна обладать информация, чтобы её использование было эффективным.

Подведём итоги

- 1 Поиск информации — это информационный процесс, направленный на выделение совокупности документов, отвечающих требованиям, сформулированным в запросе на поиск.
- 2 Запрос на поиск отражает информационную потребность пользователя. Успешность поиска зависит от того, насколько точно выражена информационная потребность в формулировке запроса.
- 3 Поисковый запрос может быть адресным (содержать указания точных параметров разыскиваемых документов), семантическим (указывать основной смысл искомой инфор-

Подведём итоги

мации) или фактографическим (направленным на выявление документов, содержащих конкретный факт). Все такие запросы являются явными.

- 4 В неявных запросах указывается лишь тема или область интересов. Обработка неявных запросов, как правило, требует применения специальных средств поиска.

Вопросы и задания

- 1 Чем поиск информации отличается от других информационных процессов — получения информации, её хранения, передачи и обработки?
- 2 В чём отличие семантического поиска от адресного?
- 3 Для чего используется фактографический поиск?
- 4 Представьте, что на день рождения вам подарили хомячка. Вам нужно знать, как обеспечить ему нормальную жизнь в вашем доме.
Разбейтесь на группы по 2—3 человека и в каждой группе:
а) Обсудите, какие темы должны быть представлены в запросах, позволяющих получить требуемую информацию.
б) Для каждой темы запишите ключевые слова, полезные, на ваш взгляд, для получения информации по данной теме.
Сравните результаты работы групп и выберите лучшие варианты запросов.
- 5 О некотором учёном сказано, что одна из наиболее интересных его статей называется «Иррациональность антиномий чистого разума». Как определить, в какой научной области ведёт исследования этот учёный?
- 6 На XI Кубке России по поиску в Интернете, проходившем в 2010 г., был задан вопрос: «Какова ширина (в сантиметрах) портрета из основной экспозиции Третьяковской галереи, на котором изображена мать нобелевского лауреата?» Получить на него ответ, сформулировав ровно один запрос, вряд ли возможно. Попробуйте сформулировать последовательность уточняющих запросов, которые позволят, по вашему мнению, получить нужную информацию. (Один из участников Кубка сумел получить ответ за 66 с.)

- 7 В «Сказке о мёртвой царевне и о семи богатырях» А. С. Пушкина царица обращается к чудо-зеркальцу с запросом:

«Свет мой, зеркальце! Скажи
Да всю правду доложи:
Я ль на свете всех милее,
Всех румяней и белее?»

Какова тема сформулированного царицей запроса? Является ли этот запрос явным?

- 8 Какая информация называется релевантной?
- 9 Сформулируйте, какое качество выполненного поиска характеризуется параметром «точность». А какое параметром «полнота»?

§ 13. Базы данных



Где хранятся данные?

Какие действия можно выполнять с данными в базах данных?

Когда в предыдущем параграфе мы обсуждали вопросы поиска информации, то молчаливо предполагали, что эта информация где-то хранится. Важно также, чтобы хранилище представляло собой не «свалку», найти в которой что-либо почти невозможно, а специально организованную совокупность данных, в которых и аккумулирована востребованная информация.

Вовсе не обязательно, чтобы в таких хранилищах содержались большие объёмы информации. К примеру, сегодня практически в любом магазине каждый кассовый аппарат связан с информационным хранилищем, в котором для каждого товара имеется информация о его цене и других характеристиках. Кассиру достаточно всего лишь считать сканером штрихкод товара, вся остальная информация о нём поступит и будет обработана автоматически.

Ключевые слова:

данные; база данных;
система управления
базой данных (СУБД);
информационно-
поисковые системы (ИПС);
иерархическая база данных;
сетевая база данных;
реляционная база данных;
сортировка данных

В отделе кадров предприятия будет востребована уже другая информация: фамилии, имена и отчества сотрудников, занимаемые ими должности, даты рождения, образование и т. д. И запросы могут быть весьма разными — не только о данных конкретного работника, но и о том, у кого есть высшее образование или кто в этом году достигнет пенсионного возраста.

В справочном бюро железнодорожной станции интересуются временем прибытия и отправления поездов, наличием мест, возможностью доехать с одной станции до другой и т. д.

Несмотря на весьма большое различие в характере информации, описанные ситуации имеют много общего: в большом объёме информации разыскивается та, которая необходима на данный момент. И здесь снова надёжным помощником становится компьютер. Ведь всё, что вы уже изучили в курсе информатики, не может не убедить вас, что компьютер — это мощный инструмент для переработки информации. Разумеется, сама информация должна быть представлена в таком виде, чтобы обрабатывать её можно было автоматически. В этом случае говорят, что информация представлена в виде данных.

Данные — это представление фактов и идей в формализованном виде, пригодном для передачи и обработки в некотором информационном процессе.

Теперь нам требуется, чтобы компьютер сыграл роль рыбака с сетью, который выловит из громадной стаи рыб только ту добычу, которая нас интересует. Или, если это вам больше понравится, компьютер похож на обоганительную фабрику, которая из сотен тонн породы добывает нужный минерал. Роль косяка рыбы (или обогащённой руды) выполняют базы данных. А чтобы накапливать, изменять, хранить и искать нужную информацию, необходимо использовать специальные программы — системы управления базами данных (СУБД) и информационно-поисковые системы (ИПС).

Различие между ними весьма условное. Пояснить его можно так. Если речь идёт о каталоге библиотеки, то никто не разрешит простому посетителю изменять в каталоге информацию о книгах, хранящихся в фондах библиотеки. А просмотреть и выбрать то, что необходимо, — пожалуйста. Можно сказать, что специалистам — сотрудникам библиотеки — доступны все возможности работы с базой данных, а посетители работают только с информационно-поисковой системой.

Как же организована база данных? В ней содержатся сведения о большом количестве однотипных объектов. При этом для каж-

дого из объектов существенными являются значения лишь некоторых признаков. Такие признаки в теории баз данных принято называть атрибутами. Иногда некоторые из атрибутов объявляют ключевыми. Некоторые из признаков могут быть объявлены обязательно присутствующими. Например, любая книга всегда имеет какое-нибудь название.

Теперь, когда у вас есть представление о том, как и для чего может использоваться база данных, сформулируем, что это такое.

База данных — это систематизированная совокупность данных, организованная так, чтобы для них было обеспечено хранение, первичная обработка и поиск нужной информации с помощью компьютерных технологий.

Такое определение исключает из понятия баз данных те или иные внекомпьютерные хранилища информации (архивы бумажных документов, бумажные каталоги библиотек, бумажные карты и т. п.). Но к базам данных традиционно не относят и некоторые компьютеризированные хранилища информации, например интернет-порталы, которые, разумеется, содержат разнообразные материалы в электронном виде, но они слишком уж разнородны по своему содержанию. Впрочем, нередко интернет-портал содержит в себе какую-либо базу данных или организует доступ пользователя к нужным ему базам данных.

Каждая конкретная база данных предназначена для решения своего класса задач. В свою очередь, для каждого класса задач характерен свой набор объектов и их признаков. Поэтому и соответствующие базы будут различными: одна нужна, чтобы разыскивать книгу по каталогу, и совсем другая — для облегчения работы директору школы. Впрочем, для разных классов задач объекты могут быть одинаковыми, а наборы нужных признаков — разными. Например, овощевода интересуют урожайность, время посева и сроки уборки овощей, повара — калорийность овощей и рецепты овощных блюд, а директора овощехранилища — условия хранения этих овощей.

Для того чтобы заставить компьютер найти интересующие нас сведения, нужно составить запрос. Правила записи запросов для каждой ИПС свои. Эти правила устанавливаются теми, кто создаёт ИПС. Обычно самые распространённые запросы к базе данных уже заранее составлены. Чтобы получить информацию по такому запросу, надо просто выбрать соответствующий пункт в меню, расположенном на экране. Более того, в большинстве персональных баз данных вообще все запросы заранее составлены, и работа

с ними доступна даже человеку, далёкому от компьютерных премудростей.

Сейчас в мире созданы сотни тысяч баз данных. Они используются в библиотеках и больницах, в гидрометцентрах и на заводах, в магазинах и планирующих организациях, в банках и частных фирмах. Тематические базы данных для узких специалистов называют банками данных. Это может быть банк данных по микропроцессорам, банк по лекарственным средствам, банк публикаций в области ядерной физики и т. п.

В нашей стране действуют десятки банков данных. Пожалуй, самый большой из них — банк данных Российского института научной и технической информации. В нём содержится более 6 млн библиографических сведений о книгах и статьях практически по всем отраслям знаний. Тот, кто покупает авиабилеты, пользуется услугами другого крупного банка данных нашей страны — системой «Сирена». С помощью компьютера кассир связывается с центральным компьютером, находящимся за тысячи километров от него, почти мгновенно получает сведения о наличии мест на данный рейс и печатает билет.

Глобальная сеть Интернет также содержит в себе большое количество самых разнообразных баз данных. Это и общедоступные базы данных, и базы данных сугубо специального назначения. А поиск в сети Интернет осуществляется с помощью размещённых в ней ИПС. Среди русскоязычных поисковых систем лидерами можно назвать Rambler (rambler.ru, рамблер.рф), Яндекс (yandex.ru, яндекс.рф) и Google (google.ru, гугл.рф).

По способу организации связей между атрибутами различают три вида баз данных: иерархические, реляционные и сетевые.

В иерархических базах данных атрибуты соподчинены между собой так, что данные образуют древовидную структуру. Эта структура заимствована из бескомпьютерных аналогов — различных каталогов, справочников и т. п. Например, авторский каталог книг. В нём главный атрибут — фамилия автора, следующий атрибут по значимости — название книги, затем, к примеру, издательство, далее — год издания и т. д. Тем самым записи в таком каталоге будут упорядочены сначала в алфавитном порядке авторов, для одного автора — в алфавитном порядке названий, затем для одного названия — в алфавитном порядке издательств, наконец, для одного издательства — по годам издания. На рисунке 13.1 показано, как выглядит соответствующая структура записей.

Если требуется найти все произведения И. П. Соколова или любого другого писателя, то это сделать несложно. Однако найти в такой структуре, например, все издания, выпущенные издательством «Альфа», будет уже не так легко. Для подобных целей составляют

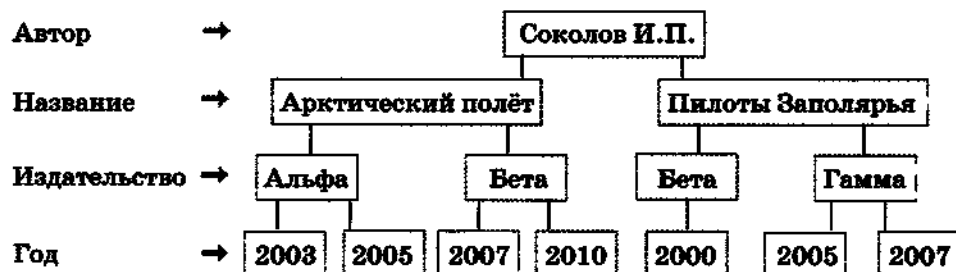


Рис. 13.1

другой каталог. Однако хранить много различных каталогов, т. е. иерархически организованных списков, накладно даже для компьютеров с большой памятью.

Сетевая база данных представляет собой более общую структуру, нежели иерархическая база: в ней элементы связаны произвольным образом. Каждый элемент может быть связан с любым количеством элементов из разных уровней.

Реляционная база данных — это совокупность связанных между собою таблиц. Название столбцов в таких таблицах — атрибуты данных, строка содержит значения всех атрибутов для конкретного объекта. Например, таблицей 13.1 можно представить информацию о книгах, которые фигурируют в примере иерархической базы данных (рис. 13.1).

Таблица 13.1

Атрибут	Автор	Название	Изда- тельство	Год
Значения атрибута	Соколов И. П.	Арктический полёт	Альфа	2003
	Соколов И. П.	Арктический полёт	Альфа	2005
	Соколов И. П.	Арктический полёт	Бета	2007
	Соколов И. П.	Арктический полёт	Бета	2010
	Соколов И. П.	Пилоты Заполярья	Бета	2000
	Соколов И. П.	Пилоты Заполярья	Гамма	2005
	Соколов И. П.	Пилоты Заполярья	Гамма	2007
	...	...	...	...

Реляционная база данных — это совокупность подобных таблиц. Важно при этом, что между таблицами могут быть установлены связи, указывающие на совпадение множества значений для разных атрибутов. Это позволяет при поиске нужной информации переходить от одной таблицы к другой, получая ответы на такие запросы, которые нельзя получить с использованием только одной таблицы.

В настоящее время наибольшее распространение получили именно реляционные базы данных. Чтобы получить из такой базы нужную информацию, составляют запрос, в котором указывают известные пользователю значения атрибутов и отмечают, какую информацию требуется получить, т. е. значения каких атрибутов вывести на экран компьютера. Например, если нужно узнать, какие произведения писателя Соколова вышли в издательстве «Альфа», то для атрибутов *Автор* и *Издательство* задаются значения Соколов и Альфа соответственно, а в качестве запрашиваемого атрибута надо отметить *Название*. Может случиться так, что вы разыскиваете произведение, про которое вы лишь помните, что то ли автор у него Соколов, то ли издавалось оно в «Альфе». Тогда составляется запрос, в котором значения атрибутов *Автор* и *Издательство* соединяются союзом *или* (можно считать, что в предыдущем случае значения этих атрибутов были соединены союзом *и*).

Над данными, хранящимися в базе данных, можно выполнять некоторые операции. Одной из таких операций является *сортировка*. Можно, например, дать указание СУБД расположить строки в таблице по возрастанию или убыванию значений заданного числового атрибута или в алфавитном порядке, если атрибут текстового типа.

Правила составления запросов и выполнения операций над данными у каждой базы данных, разумеется, свои; обычно они описываются в инструкции пользователю.

Подведём итоги

- 1 Для хранения и дальнейшей обработки информация преобразуется в данные.
- 2 Для хранения информации в электронном виде и осуществления её оперативного поиска создаются базы данных.
- 3 По способам организации связей между атрибутами базы данных подразделяются на иерархические, реляционные и сетевые.

Подведём итоги

- 4 Для работы с базой данных используется программная оболочка — система управления базой данных (СУБД).
- 5 Пользователю базой данных обычно предоставляется ограниченный доступ к функциям, реализуемым СУБД: обычно разрешается осуществлять лишь поиск информации без права её изменения и размещения собственной информации. База данных с такими ограничениями называется информационно-поисковой системой (ИПС).

Вопросы и задания

- 1 Что такое база данных?
- 2 Что такое СУБД? Что такое ИПС?
- 3 Какие операции можно производить с данными, хранящимися в базе данных?
- 4 Что является объектом в базе данных «Почтовые марки»? Какие атрибуты объектов в этой базе данных целесообразно иметь?
- 5 Какие существуют основные виды баз данных?
- 6 Какие атрибуты вы считаете нужным выделить, если организуется база данных:
 - а) описывающая фонд библиотеки;
 - б) содержащая кулинарные рецепты;
 - в) описывающая движение поездов по данной железнодорожной станции;
 - г) содержащая сведения о сотрудниках данного предприятия;
 - д) содержащая сведения о больных, посещающих данную поликлинику;
 - е) содержащая сведения о погоде в данной местности за год;
 - ж) описывающая фонд путёвок данного туристического агентства;
 - з) описывающая историю олимпийского движения (в целом или по данному виду спорта);
 - и) содержащая сведения об учениках одной школы;
 - к) содержащая сведения о государствах Европы;
 - л) содержащая сведения о свойствах химических элементов («Электронная таблица Менделеева»)?На основе каких из перечисленных баз данных возможно, на ваш взгляд, организовать ИПС?

7 В базе данных «Города России» хранятся, в частности, следующие сведения о городах:

- название — атрибут *Название*;
 - площадь (кв. км) — атрибут *Площадь*;
 - население (тыс. чел.) — атрибут *Население*;
 - год основания — атрибут *Год*;
- к какому субъекту Федерации (краю, области, автономной республике) относится данный город — атрибут *Субъект*;
- является ли административным центром субъекта Федерации — атрибут *Центр*.

В запросе к этой базе данных указывается имя атрибута и значения, по которым требуется провести поиск. Отношения между атрибутом и значением описываются знаками =, ≠, а для числовых атрибутов могут также использоваться знаки >, <, ≥ (не меньше) и ≤ (не больше). Например, чтобы найти все города площадью больше 10 кв. км, составляется запрос

Площадь > 10,

а для городов, существовавших ещё в XVIII в. и являющихся сейчас административными центрами, потребуется запрос

Год < 1900 и Центр = да.

Составьте запросы, позволяющие найти:

- а) все города-миллионники (с населением более 1 млн чел.);
 - б) все города с площадью более 50 кв. км и населением меньше 300 тыс. чел.;
 - в) все административные центры, кроме Москвы и Санкт-Петербурга;
 - г) все крупные (с населением не менее 500 тыс. чел.) города Поволжья (т. е. города тех субъектов Федерации, по территории которых протекает р. Волга).
- 8** Какой запрос надо сформулировать базе данных «Города России» (см. задание 7), чтобы выяснить:
- а) все ли административные центры имеют население более 600 тыс. чел.;
 - б) все ли города с населением более 600 тыс. чел. являются административными центрами?
- 9** Однажды школьник решил воспользоваться базой данных «Города России» (см. задание 7), чтобы найти все города Тульской и Новгородской областей. Тут подошёл злоумышленник и быстро составил следующий запрос:

Субъект = Тульская и Субъект = Новгородская.

В базе данных не оказалось городов, удовлетворяющих этому запросу. Объясните почему. Какой запрос надо составить, чтобы получить нужную школьнику информацию?

10 а) Спроектируйте базу данных, содержащую информацию о вашем классе. Для этого надо определить атрибуты, по значениям которых будут разыскиваться ответы на различные запросы — о домашнем адресе, дате рождения, об успеваемости, интересах и т. п. Разбейтесь на группы по 3—4 человека и составьте подходящий, на ваш взгляд, список атрибутов.

б) В таблице 13.2 представлен результат работы одной из таких групп. Оцените, насколько удачно определены ею атрибуты для проектируемой базы данных «Класс». (Совет: продумайте возможные запросы и то, удобно ли получать на них ответы при таком описании атрибутов.)

Таблица 13.2

Фамилия, имя	Адрес	Дата рождения	...
Алексеев Андрей	Ул. Серова, д. 25, кв. 7	25 июля 1999 г.	...
Белых Белла	Пр. Синий, д. 3, кв. 29	9 ноября 1998 г.	...
Васильев Валерий	Ул. Ванина, д. 6, кв. 105	29 февраля 1999 г.	...
...	...	...	...

11 Выполнив задание 10, вы спроектировали базу данных «Класс». Ниже сформулированы вопросы, на которые должна уметь отвечать эта база данных:

- У кого день рождения в мае?
- Кто живёт на той же улице, где расположена школа?
- Кто самый младший (по возрасту) ученик в классе?
- Кто самый старший (по возрасту) ученик в классе?
- Кто из мальчиков класса самый старший (по возрасту)?
- Кто из девочек класса самая младшая (по возрасту)?
- У кого первого в текущем учебном году был день рождения?
- Кто из мальчиков занимается в первой подгруппе иностранного языка?
- Есть ли в классе однофамильцы?
- Есть ли в классе братья и сёстры?
- У кого в классе есть братья или сёстры?

Подумайте, какие атрибуты должны быть у базы данных, чтобы можно было сформулировать к ней запросы, позволяющие получить ответы на перечисленные вопросы. Обсудите получившиеся у вас списки атрибутов и выберите наилучший, по вашему мнению, вариант.

§ 14. Интернет: сеть глобальных коммуникаций и распределённая база данных



Что такое Всемирная информационная паутина?

Какие программы обеспечивают навигацию в Интернете?

Как именуются компьютеры, являющиеся узлами глобальной компьютерной сети?

Кто предоставляет услуги в телекоммуникационных сетях?

Влияние Интернета на мир не имеет исторических аналогов. Изобретение телеграфа, телефона, радио и компьютеров подготовило почву для происходящей сейчас беспрецедентной интеграции. Интернет одновременно является и средством общемирового вещания, и механизмом накопления и распространения информации, и средой для сотрудничества и общения, охватывающей весь земной шар.

Впрочем, Интернет так и остался бы компьютерной сетью для специалистов, если бы не два поистине революционных изобретения, сделанные в области программного обеспечения.

Первым изобретением стала электронная почта, вторым — Всемирная информационная паутина, которая в сокращении известна как WWW. Эти изобретения, во-первых, сделали Интернет предметом первой необходимости для огромного числа людей, а во-вторых, обеспечили необыкновенную лёгкость работы с сетевым программным обеспечением.

История WWW (World Wide Web) началась в марте 1989 г., когда Тим Бернерс-Ли из Европейской лаборатории по ядерным исследованиям предложил новый способ обмена результатами исследований и идеями между участниками коллектива исследователей-физиков, работавших в разных странах. Для передачи документов и установления связи предлагалось использовать специально созданную систему распространения текстовых документов, содер-

Ключевые слова:

Интернет;
веб-страница;
Всемирная
информационная
паутина; сайт;
веб-сайт; браузер;
провайдер;
доменная система
имён; адресация в
сети; универсальный
указатель ресурса

История Интернета

- Как ни прискорбно об этом сообщать, но Интернет стал таким же детищем военных технологий, как и сам компьютер. В ответ на запуск в 1957 г. советского спутника президент США Д. Эйзенхауэр создал в рамках Министерства обороны США (Department of Defence, DOD) лабораторию DARPA. Первая буква D в этом сокращении, как и последняя в аббревиатуре DOD, означает «оборона», т. е. лаборатория была военная.
- В 1960-х гг. основные работы DARPA были посвящены разработке метода соединений компьютеров друг с другом. Возглавлял исследовательскую программу, посвящённую системе глобальной коммуникации, начатую DARPA в октябре 1962 г., Дж. Ликлайдер, который тогда же опубликовал работу «Galactic Network». В ней он предсказывал возможность существования в будущем глобальной компьютерной связи между людьми, имеющими мгновенный доступ к программам и базам данных из любой точки земного шара. Его предвидение отражает современное устройство международной сети Интернет.
- К концу 1969 г. в одну компьютерную сеть были объединены четыре исследовательских центра. Эта сеть получила название ARPANET. Буква D исчезла.

жащих ссылки на фрагменты других, таких же текстовых документов, но располагающихся на удалённых компьютерах, подключённых в единую сеть. Сегодня World Wide Web — это совокупность огромного числа связанных между собой информационных материалов.

Вот основные понятия, относящиеся к Всемирной информационной паутине:

WWW — множество веб-страниц, размещённых на узлах Интернета и связанных между собой гиперссылками.

Веб-страница — структурная единица WWW, которая включает в себя собственно информацию (текстовую и графическую) и ссылки на другие страницы или ресурсы Интернета.

Систему веб-страниц, управляемую одним человеком или одной организацией, обычно называют веб-сайтом или просто сайтом (от английского site — местоположение, участок под застройку).

В настоящее время Интернет превратился в единое информационное поле планетарных масштабов. И его воздействие на развитие цивилизации грандиозно, хотя до сих пор и не осознано в полной мере. К концу 2010 г. число пользователей Интернета было более 2 млрд человек, из них в России зарегистрировано более 60 млн.

Во все современные операционные системы встроено всё

необходимое для начала работы в Интернете, включая мастера для установки и настройки программного обеспечения. Но для доступа

к Интернету необходимо подключение в какой-либо форме к поставщику услуг Интернета, так называемому провайдеру. Это можно осуществить двумя способами:

непосредственно обратившись к поставщику услуг Интернета; подключившись к локальной сети, имеющей, в свою очередь, постоянный выход в Интернет.

В первом случае снова возможны два варианта подключения: постоянное подключение по специально выделенному каналу связи и сеансовое подключение по обычному телефонному каналу с использованием модема. Как правило, провайдер предоставляет и бесплатный электронный почтовый ящик. Услуги за пользование сетью оплачиваются пользователем в зависимости от так называемого тарифного плана компании-провайдера. Тарифный план может учитывать время пребывания в сети, скорость обмена информацией и объём предоставляемой информации.

Что касается подключения к локальной сети, имеющей постоянный выход в Интернет, то в настоящее время такие локальные сети могут объединять микрорайон, отдельный дом или даже подъезд.

Чтобы получить информацию, хранящуюся в Интернете, надо обратиться к компьютеру, который управляет соответствующим хранилищем данных. А чтобы найти этот компьютер, нужно знать его место в сети, или, как говорят, его сетевой адрес. Реально эти адреса представляют собой наборы цифр. Но это крайне неудобно для человека! Поэтому была предложена система имён, принцип которой похож на обычную почту. Ведь, вообще говоря, все жилища тоже можно было бы просто перенумеровать последовательными числами, но попробуй тогда быстро найти нужный дом. Совсем другое дело, когда сначала указана страна, область, потом населённый пункт, затем улица и номер дома и, наконец, номер квартиры (если дом многоквартирный). Обратите внимание — в такой системе адресации действует иерархический принцип: сначала самое крупное географическое подразделение, затем следующее по степени охвата адресатов и т. д.

Адресация в компьютерной сети тоже устроена по иерархическому принципу: каждый адрес в компьютерной сети состоит из нескольких частей, разделённых точкой. Каждая часть — её называют доменом — указывает на то, в какой части сети располагается данный компьютер. Поскольку Интернет развивался в первую очередь в США за счёт американских налогоплательщиков, это вызвало некоторый крен при формировании доменов первого уровня: Интернет как бы оказался поделённым между США и всем остальным миром.

Наиболее известные домены первого уровня:

com — коммерческие организации (главным образом в США);
edu — учебные заведения США;

gov — правительственные учреждения США;
mil — военные учреждения США;
net — различные сетевые агентства и интернет-провайдеры;
int — международные организации;
org — некоммерческие учреждения;
info — средства массовой информации.

Позднее, когда сеть перешагнула национальные границы США, появились национальные домены типа: uk, jp, au, ch и т. п. Для СССР тоже был выделен домен — su. После 1991 г., когда республики Союза стали суверенными, многие из них получили свои собственные домены: ru, by, ua, la, li и т. п. Однако Интернет — не СССР, и просто так выбросить домен su из сервера имён нельзя: на основе доменных имён строятся адреса электронной почты и осуществляется доступ ко многим другим информационным ресурсам Интернета. Поэтому гораздо проще оказалось добавить новый домен к существующему, чем заменить его. Таким образом, в сети до сих пор существуют организации с доменными именами, оканчивающимися на su.

Доменов верхнего уровня очень немного — всего около 250. Большая часть из них — географические домены. Если бы вам захотелось зарегистрировать ещё один домен верхнего уровня, то для этого потребовалось бы представить такие серьёзные обоснования, что гораздо проще было бы организовать своё маленькое государство и для него уже получить географический домен.

Вслед за доменами верхнего уровня следуют домены, определяющие либо регионы, либо организации. Например, домен `msk` для домена верхнего уровня `ru` или домен `1сентября` для кириллического домена верхнего уровня `рф`. Далее идут следующие уровни иерархии, которые могут быть закреплены либо за небольшими организациями, либо за подразделениями больших организаций.

Иерархия доменных имён строится следующим образом: дерево начинается с корня, обозначаемого точкой. Затем следует старшая символическая часть имени, вторая по старшинству символическая часть имени и т. д. Младшая часть имени соответствует конечному узлу сети. Запись доменного имени начинается с самой младшей составляющей, а заканчивается самой старшей. Составные части доменного имени отделяются друг от друга точкой. Например, в имени `w2000.microsoft.com` составляющая `w2000` является именем одного из компьютеров в домене `microsoft.com`.

Разделение административной ответственности позволяет решить проблему образования уникальных имён без взаимных консультаций между организациями, отвечающими за имена одного уровня иерархии. Очевидно, что должна существовать одна организация, отвечающая за назначение имён верхнего уровня иерархии. В Интернете корневой домен управляется центром ICANN,

в состав которого входят по пять представителей от каждого континента.

Процедура получения имени, например, в зоне *ru* или *com* называется регистрацией домена. Конечно, каждая компания, подключающаяся к Интернету, стремится получить как можно более естественное и лёгкое для запоминания имя. Так, для компании Microsoft логично зарезервировать домен *microsoft.com*.

Каждый домен администрируется отдельной организацией, которая обычно разбивает свой домен на поддомены и передаёт функции администрирования этих поддоменов другим организациям. Чтобы получить доменное имя, необходимо зарегистрироваться в какой-либо организации, которой ICANN делегировал свои полномочия по распределению имён доменов. В России такой организацией является NIC.ru, которая отвечает за делегирование имён поддоменов в домене *ru* и кириллическом домене *рф*.

Наконец, имеется ещё одно важное для Интернета понятие — URL (Uniform Resource Locator) — универсальный указатель ресурса. Он представляет собой точное описание ресурса, включающее его местонахождение в Интернете, и имеет следующую структуру:

`service://host:port/path/file.ext`

(вид сервиса://имя узла:номер порта/путь/имя файла.расширение).

Например:

`http://www.ustu.ru/...`

Сервис *http* означает, что мы намерены иметь дело со Всемирной паутиной; *www.ustu.ru* — это имя того узлового компьютера, с информационными ресурсами которого мы собираемся работать; в данном случае речь идёт об узловом компьютере сети WWW, обслуживающем Уральский федеральный университет имени первого Президента России Б. Н. Ельцина.

Знания адреса сайта недостаточно. Нужна компьютерная программа, позволяющая обратиться к нужному ресурсу с помощью универсального указателя ресурса. Такие программы называются браузерами (от англ. *browse* — пролистывать, просматривать). Эти программы осуществляют следующие важные функции: просмотр, сохранение и печать документа, имеющегося в сети, формирование закладок, хранение истории навигации в Интернете, настройку на различные форматы документов. Из современных браузеров наиболее популярными являются Microsoft Internet Explorer, Mozilla Firefox и Opera. Внешний вид окон этих браузеров изображён на рисунках 14.1—14.3.

В окне каждого из браузеров, изображённых на рисунках 14.1—14.3, вы видите ту или иную поисковую систему. О двух из них — Google и Rambler — мы упоминали в § 13; о системе *mail.ru* ничего не говорилось, поскольку она была создана (в 1998 г.) в первую



Рис. 14.1. Браузер Microsoft Internet Explorer



Рис. 14.2. Браузер Mozilla Firefox



Рис. 14.3. Браузер Opera

очередь как бесплатная электронная почта для российских пользователей. Но позже (с конца 2006 г.) и в этой системе появилась возможность поиска в сети Интернет.

Пока в сети было не так много информации и список доменных имён уместился на нескольких страницах, поисковые системы не требовались. А сегодня с каждым днём и даже часом в гигантской паутине гипертекстовых связей появляются всё новые и новые документы, и теперь каждый исследователь, прокладывая себе путь в необъятных просторах киберпространства, с первых же минут ощущает потребность в надёжном путеводителе. Дэвид Фило и Джерри Янг из Стэнфордского университета в апреле 1994 г. пришли к выводу, что у каждого из них накопилось такое большое количество ссылок на самые разные информационные источники, что пришла пора каким-то образом их упорядочить. Так родилась идея специализированной базы данных, которая весьма скоро стала популярной у тысяч пользователей, разыскивающих информацию в сети. Эта поисковая система получила имя Yahoo! и, разумеется, была англоязычной.

Первой русскоязычной поисковой системой стала система Яндекс. Официально она была представлена 23 сентября 1997 г. на выставке Softool.

На первой странице любой поисковой системы имеется специальное окно, в которое вы вписываете запрос на поиск. Сам запрос состоит из ключевых слов, которые выражают то, что со-

ставляет суть разыскиваемой информации. Пусть, к примеру, вам для подготовки реферата по истории Интернета нужно найти фотографию человека, которого нередко называют «отцом» Интернета. Это Винтон Серф. Тогда в строке поиска надо набрать слово Серф. А ещё лучше набрать два слова: Винтон Серф. Пробел между словами большинство поисковых систем воспринимает как союз **и**. Впрочем, если вы попробуете так поступить, то количество найденных документов составит несколько тысяч. Среди них и статьи самого В. Серфа, и статьи о нём, и книги, в которых упоминается его имя (в частности, и этот учебник). Это вовсе не значит, что поисковая система сработала плохо, это вы неудачно составили запрос. Вы же хотите фотографию, но ничего об этом поисковой системе не сказали. Составив запрос: Винтон Серф фото, — вы резко увеличите шансы на успех. Правила составления запросов с союзом **или** весьма сильно зависят от поисковой системы; вы можете их посмотреть, обратившись к режиму *Помощь*. И вообще, мы советуем, прежде чем начать работать в поисковой системе, познакомиться с правилами составления запросов в ней — это сэкономит ваши силы и время.

Подведём итоги

- 1 Телекоммуникационная сеть Интернет выполняет несколько функций. Одной из важнейших является хранение и доставка пользователю информации различного содержания и назначения.
- 2 Информация хранится распределённым образом, т. е. в системе электронных хранилищ под управлением компьютеров, соединённых в единую сеть.
- 3 Каждый компьютер сети имеет уникальный адрес. Для удобства пользователя компьютерной сетью эти адреса организованы в иерархическую структуру, называемую доменной системой имён.
- 4 Для навигации в сети существуют специальные программы, называемые браузерами.
- 5 Для обращения к компьютеру сети, который управляет нужным пользователю информационным ресурсом, необходимо указать адрес этого компьютера и имя ресурса. В совокупности адрес компьютера и имя ресурса называется универсальным указателем ресурса.
- 6 Пользование сетью — платная услуга, предоставляемая компанией — провайдером сети.

- 1 Что называют Всемирной паутиной?
- 2 Как называются программы просмотра страниц Всемирной паутины?
- 3 Для чего служит доменное имя?
- 4 Как устроен универсальный указатель ресурса в Интернете? Для чего он предназначен?
- 5 Что такое веб-сервер?
- 6 Какие существуют варианты подключения индивидуального пользователя к Интернету?
- 7 Если ваш компьютерный класс подключён к Интернету, выясните, какой из вариантов для этого используется.
- 8 Какова роль провайдера?
 - а) Выясните, есть ли в вашем населённом пункте провайдеры.
 - б) Изучите тарифные планы, которые они предлагают. Какой из них кажется вам наиболее подходящим? Обсудите своё решение с одноклассниками и с кем-либо из старших.

§ 15. Что ещё можно делать в Интернете



Как можно общаться в Интернете?

В чём преимущества коммерческой деятельности, ведущейся через Интернет?

Чем полезны интернет-магазины?

Хотя мы уже довольно подробно обсуждали достоинства Интернета как бездонного хранилища информации, исторически первым сервисом, предоставлявшим ещё сетью ARPANET, была электронная почта. Она позволяет обмениваться текстовыми сообщениями с любым пользователем, имеющим выход в глобальную компьютерную сеть и зарегистрированным на каком-либо почтовом сервере.

Ключевые слова:

электронная почта;
ftp-сервис; ICQ;
«Живой журнал»;
социальная
интернет-сеть;
Твиттер; блог;
интернет-магазин

При регистрации абонент получает электронный почтовый адрес, имя пользователя для входа в сеть — так называемый **login** — и пароль, позволяющий избежать несанкционированного доступа посторонних лиц к вашей почте.

Адрес в электронной почте образован из доменных имён, которые стоят после символа **@**. Например: **Billt@tenet.edu**, или **Shaynes@monroe.lib.mi.us**, или **2170@dialup.mplik.ru**. Слева от значка **@**, как правило, указывается идентификатор конечного пользователя, т. е. присвоенный вам позывной на том сервере, где вы заказали электронный почтовый ящик.

Преимущества электронной почты по сравнению с обычной весьма значительны. Она позволяет:

- посылать сообщения сразу нескольким абонентам;
- посылать не только собственно текст письма, но и файловые вложения с графикой, звуком, программным обеспечением и т. д.;
- пересылать полученные сообщения другим адресатам;
- уведомлять отправившего письмо, что оно получено.

Это далеко не всё, что умеют делать современные программы почтового сервиса. Одной из таких программ является, например, **Outlook Express**. Впрочем, в последнее время всё большее распространение в работе с электронной почтой получают веб-технологии. Появились веб-сайты, которые предлагают всем желающим зарегистрировать бесплатный почтовый ящик. Например, это можно сделать по адресу **mail.ru**. Но можно такой ящик получить и на сайте любой другой поисковой системы.

Надеемся, что вы адекватно оценили Интернет как информационное хранилище и средство коммуникации с любым уголком нашей планеты. Но информация информации рознь, и ваш интерес к ней тоже может быть весьма различным. Одно дело просмотреть новости в той или иной газете и совсем другое — изучить большую научную статью. Для такого изучения потребуется время, и возвращаться к её тексту, возможно, потребуется не один раз. Гораздо эффективнее получить нужную информацию к себе на компьютер в виде соответствующего файла. Кроме того, Интернет не только наполнен текстовой, звуковой или видеоинформацией, но в нём имеется немало самого разнообразного программного обеспечения, распространяемого как солидными программистскими фирмами, так и просто программистами-любителями. Получить файлы по сети позволяет **ftp-сервис** (от **File Transfer Protocol** — протокол передачи файлов). Именно он предоставляет доступ к файловым системам и выполняет передачу файлов в сеть (например, при публикации собственного веб-сайта). Одним из распространённых видов **ftp-серверов** является анонимный **ftp-сервер**. На таком сервере вы используете в качестве своего имени слово «**anonymous**», а в качестве пароля — свой **e-mail**. Если же вы

хотите установить соединение с компьютером, не предоставляющим анонимного сервиса, вы должны иметь право доступа к системе, т. е. зарегистрироваться на нём, выбрав для себя имя и пароль.

Для работы с ftp-серверами можно воспользоваться браузером. Только в начале адреса надо написать ftp://, а затем URL нужного вам ресурса. Удобнее, однако, для работы с большими по объёму файлами использовать специализированные программы, которые называются менеджерами загрузки файлов (например, FlashGet, Go!Zilla и др.). Они разбивают файловый архив на части, загружают эти части одновременно, что уменьшает время загрузки файла на ваш компьютер, позволяют прервать загрузку файла после обрыва соединения с сервером и выполняют ряд других работ, облегчающих жизнь пользователю.

Но конечно, польза от ftp-сервиса была бы не так значительна, если для получения информации (чем можно пожить благодаря этому сервису) пришлось бы просматривать все имеющиеся в сети анонимные серверы. Интернет и здесь идёт на встречу пользователю. Для этого существуют ИПС, с помощью которых можно проводить поиск информации, расположенной на анонимных ftp-серверах. Система, поддерживающая этот вид услуг, регулярно собирает с таких серверов информацию о содержащихся на них файлах. К таким системам относится, например, ИПС Archie.

Перестроив на электронный манер почтовую связь, разработчики глобальных сетей не успокоились. Получать письма приятно, но ещё приятнее диалог, когда на каждый вопрос или суждение ты тут же получаешь реакцию собеседника. Простейший вариант такого общения — письменное сообщение: ты строку,

«Живой журнал» — хроника развития

День рождения «Живого журнала» — 19 марта 1999 г. К концу 2005 г. зарегистрированными пользователями этого сервиса стали 8,9 млн человек. К середине 2007 г. их стало более 13 млн человек, а в июне 2010 г. в «Живом журнале» было зарегистрировано более 26 млн пользователей.

Вот шестёрка лидирующих стран (на 17.06.2010):



США — 4 940 561 чел.;



Россия — 1 637 819 чел.;



Великобритания —
457 793 чел.;



Канада — 440 460 чел.;



Украина — 264 843 чел.;



Сингапур — 213 453 чел.

Если записи в блоге не обновлялись в течение двух лет, то его удаляют из «Живого журнала». Впрочем, с июля 2010 г. неактивный блог, если он представляет интерес для истории, стало возможным выкупить у «Живого журнала».

тебе в ответ строчку, снова ты и т. д. В последнее время большой популярностью пользуется интерактивное общение с помощью сервиса ICQ. Каждый участник такого общения имеет уникальный идентифицирующий его номер, который он получает при регистрации. Чтобы стать участником ICQ, достаточно скачать эту программу, например, с сервера www.freeware.ru и во время установки зарегистрироваться. После запуска программы пользователь может начать общение с любым подключённым в этот момент зарегистрированным членом этого сообщества, которое сейчас насчитывает более 150 млн чел. По меткому выражению одного блоггера, Интернет заполнен болтающими подростками. Мог ли всё это представить себе президент США Эйзенхауэр, когда отдавал распоряжение об организации DARPA, с которого начался Интернет?

Если к компьютеру подключён микрофон и наушники, то общение станет не только письменным, но и звуковым. А если ещё подключена и веб-камера, то ваш собеседник увидит вас на экране своего компьютера. Такие возможности предоставляет Skype.

Личный дневник... В самом названии заложено общение с самим собой, изложение собственного восприятия происходящих вокруг событий и тех эмоциональных переживаний, которые они вызвали. Раньше такие дневники становились достоянием общества как мемуарная литература знаменитых людей. Интернет изменил и эту сторону жизни. Теперь каждый желающий может вести такой дневник в Интернете, причём не только на домашних страницах собственного сайта, который известен только узкому кругу. «Живой журнал» («Live Journal») — один из сайтов, предназначенный для ведения частных дневников. В «Живом журнале» можно делать свои записи, комментировать дневники других, вставлять фотографии и иллюстрации, делать гиперссылки, создавать настроение (словесно или смайликом). В «Живом журнале» можно искать друзей по интересам. Особенно много объединений, посвящённых книгам или образованных вокруг книг. Размышления о книгах есть во многих интернет-дневниках. Особую атмосферу интернет-разговоров о книгах создаёт отсутствие какой бы то ни было цензуры и усиленное сдерживание рекламных атак со стороны книжного бизнеса. Вот несколько адресов таких книжных сообществ:

<http://community.livejournal.com/knigi/>
http://community.livejournal.com/4itaem_zh_vas/
http://community.livejournal.com/fantasy_ru/
<http://community.livejournal.com/lovebooks/>
http://community.livejournal.com/booktalk_spb/

Широкое общение предоставляют так называемые социальные сети. Они предназначены для создания сообществ в Интернете из

людей со схожими интересами или деятельностью. По числу пользователей в мире лидирует сеть Facebook (около 500 млн человек). В нашей стране наиболее популярными являются vkontakte.ru и odnoklassniki.ru. В этих сетях вы можете найти тех, с кем давно утеряна связь, и тех, кто сейчас тесно с вами общается, но вас разделяет расстояние.

Сервисом, в котором есть черты от ICQ, «Живого журнала» и социальной сети, является Твиттер (от англ. *tweet* — чирикать, щебетать, болтать). Там тоже публикуются текстовые сообщения, но не превышающие 140 символов. Поэтому Твиттер называют системой микроблогов. Это скорее средство общения и оперативного получения новой информации. В то же время с помощью Твиттера вы можете познакомиться со многими интересными людьми, знакомство с которыми при других обстоятельствах могло бы и не произойти. Ведь для общения в Твиттере не нужно добавлять друг друга в друзья или в лист контактов, как это принято в социальных сетях и ICQ. Даже если человек не подписан на ваши обновления, вы можете отвечать на его сообщения, размещая свой ответ после знака @, и он обязательно его получит.

Интернет всё активнее вторгается в коммерческую сферу. С помощью Интернета проводятся деловые встречи и совещания, заключаются сделки, проводятся аукционы. В течение одного дня, не выходя из офиса (а то и просто из домашнего кабинета), руководитель фирмы может провести переговоры с фирмами-компаньонами, находящимися на расстоянии десятков, сотен, а то и тысяч километров. Это в несколько раз повышает возможности осуществления деловой активности, снижает непроизводственные расходы, экономит силы людей и средства фирмы.

Но и для рядового пользователя Интернет сегодня предоставляет значительное количество коммерческих услуг, например в торговой сфере. Всё большее распространение получают интернет-магазины, для которых процесс общения с покупателем происходит не



в специально оборудованном помещении с витринами и стеллажами, с большим количеством продавцов и кассиров. Товар представлен виртуально — на сайте магазина вы найдёте фотографии, видеоролик и описание интересующей вас вещи, продукта или услуги (например, туристической поездки). Вы можете сделать заказ, и выбранный товар будет вам доставлен. Нередко цены в интернет-магазине оказываются ниже, чем в магазине на соседней улице, — ведь торгующей организации уже нет нужды арендовать помещение и содержать столь значительный персонал.

Многие имеют мобильные телефоны. Оперативная, или, как говорят, мобильная, телефонная связь не только удобна, но часто в нашей насыщенной событиями жизни просто необходима. Но можно установить мобильную связь не только между двумя телефонами, но и между телефоном и компьютером. Прежде всего речь идёт о возможности отправлять с телефона на компьютер и обратно так называемые SMS (сокращение от Short Message Service — доставка коротких сообщений). Для беспроводного доступа с мобильных телефонов к ресурсам Интернета используется специальный протокол WAP (Wireless Application Protocol — протокол беспроводного применения). Отметим, что этот протокол поддерживает связь не с каким-то конкретным компьютером, а с Интернетом и его сервисами в целом. Более точно, связь поддерживается с так называемыми WAP-сайтами, которые располагаются на веб-серверах и представлены в специальном формате WML (от Wireless Markup Language). Это специальный язык разметки, ориентированный на возможности мобильного телефона — его графику, маленький экран и небольшую память.

На WAP-сайтах вы найдёте новости политики, экономики, спорта, прогноз погоды и многое другое. Вы можете отправить электронную почту и поучаствовать в WAP-чате.

Мобильный телефон может быть подключён к компьютеру. В этом случае телефон превращается во внешнее устройство, осуществляющее полномасштабную связь компьютера с Интернетом.

Жизнь человека в обществе невозможна без информации о происходящих в нём экономических и политических процессах, без приобщения к культурным событиям своей страны. Ещё полвека назад потребность в этой информации удовлетворяли газеты, журналы, радио, телевидение. Все эти источники существуют и сейчас. Но теперь Интернет предоставляет возможность обратиться практически к любому из них в виде интернет-версии. Нередко в них имеются гиперссылки на другие источники, так что, пользуясь такой версией, каждый из вас получает возможность самостоятельно выстроить маршрут для получения разносторонней информации по заинтересовавшему вас вопросу. Ведь глубина освещения вопроса

и ракурс его рассмотрения могут весьма различаться в разных изданиях. Отметим также, что все современные поисковые системы включают в набор своих основных сервисов группу «Новости». К примеру, в 2002 г. разработчиками поисковика Google была создана специальная служба Google News, которая регулярно и оперативно собирает информацию более чем с 4500 сайтов.

Однако современный Интернет не просто выполняет роль средств массовой информации, предоставляя пространство для размещения соответствующих сайтов. Широкое распространение получила так называемая технология передачи потокового звука и видео. Прослушивание и просмотр таких файлов осуществляется с помощью специальных программ, например RealPlayer или Windows Media Player. Просмотр и прослушивание передач можно осуществлять в фоновом режиме, работая в это время с другими приложениями. Одним из распространённых вариантов применения этой технологии стала непрерывная трансляция с видеокамер LiveCam, установленных на улицах городов, в музеях, заповедниках и т. д.

В настоящее время достаточно много радио и телевизионных станций осуществляют вещание через Интернет, используя потоковую технологию. Более того, некоторые компании предлагают самому пользователю сформировать из предлагаемого меню индивидуальную программу передач на его персональный компьютер, так что слушать новости и смотреть любимые фильмы вы сможете именно тогда, когда вам это будет удобно.

Вопросы и задания

- 1 Почему Интернет можно назвать бездонным хранилищем информации?
- 2 Как устроен адрес электронной почты?
- 3 В чём суть ftp-сервиса?
- 4 Как узнать, имеется ли на ftp-сервере нужная вам информация?
- 5 Какие возможности предоставляет ICQ-сервис?
- 6 В чём различия сервиса «Живой журнал» и социальных сетей?
- 7 В чём суть сервиса Твиттер?
- 8 Для чего используется протокол WAP?
- 9 Какие преимущества имеют интернет-издания по сравнению с обычными?
- 10 Какие выгоды даёт применение интернет-технологий в бизнесе?
- 11 Подготовьте сообщение (или компьютерную презентацию) об использовании интернет-технологий в коммерческой деятельности.

§ 16. Информационные системы



Какие информационные системы наиболее широко используются?

Согласно законодательству Российской Федерации информационной системой называется организационно упорядоченная совокупность документов (массивов документов) и информационных технологий, в том числе с использованием средств вычислительной техники и связи, реализующих информационные процессы. Создаются же информационные системы для хранения, обработки, поиска, передачи, предоставления, а нередко и для распространения информации. Информационная система может обслуживать одно предприятие (например, содержать сведения о сотрудниках), а может быть общегосударственной (например, метеорологическая информационная система).

Ключевые слова:
информационная система;
библиотечная
информационная система;
геоинформационная
система; информационные
системы офисной
автоматизации

Универсальных информационных систем не бывает. Каждая из них существует в единственном числе, опирается на конкретные информационные ресурсы, связанные с вполне определёнными моментами или периодами времени. Кроме того, ограничен круг лиц, имеющих возможность пользоваться данной информационной системой (хотя он может быть и весьма широким).

Информационные системы обычно классифицируют по тому, как в них представлены информационные ресурсы. Вот основные виды информационных систем:

- структурированные фактографические системы, основанные на технологии баз данных;
- текстовые системы (поисковые), оперирующие документами на естественных языках;
- гипермедийные информационные системы WWW.

Из этого перечисления видно, что мы уже познакомили вас с теми информационными технологиями, на основе которых создаются и функционируют эти виды информационных систем.

Именно развитие информационных технологий определяло облик кардинальных изменений в архитектуре и функционировании информационных систем.

Рассмотрим, к примеру, библиотеку. Ещё в 70-е годы прошлого века основу её информационной системы составляли систематический и авторский каталоги, а также система читательских карточек абонентов. Информационная подсистема «Авторский каталог» представляла собой несколько десятков (а в больших библиотеках, как, например, в Российской государственной библиотеке, несколько сотен) ящиков со стандартными карточками, расположенными в алфавитном порядке, каждая из которых начиналась фамилией автора, а затем шло полное библиографическое описание изданного произведения, включающее краткую его аннотацию. Аналогично устроен и систематический каталог, только там карточки распределены по отраслям наук и упорядочены по алфавиту названий. В современной библиотеке эти каталоги сведены в единую компьютерную информационную систему, туда же включена информация о том, находится ли эта книга в фонде или выдана читателю.

Глобальные сети позволяют объединить информационные системы разных библиотек. Теперь посетитель одной библиотеки может узнать, есть ли нужная ему книга в другой библиотеке города. Про межбиблиотечную информационную систему говорят, что она является распределённой. Вот несколько примеров самых крупных, распределённых библиотечных информационных систем:

1. <http://arbicon.ru/services> — Ресурсы российских корпоративных библиотечных систем;

2. <http://www.arbicon.ru> — Ассоциация региональных библиотечных консорциумов;

3. <http://mars.udsu.ru> — Межрегиональная аналитическая роспись статей;

4. <http://corporate.gpntb.ru> — Корпоративная сеть московских библиотек.

Также значительно количество специализированных информационных систем. К таковым относятся правовые информационные системы Кодекс или Консультант, система ИРиД (Информационные Ресурсы и Данные) для физиков, различные образовательные системы, включающие разнообразные справочники и энциклопедии.

В этих распределённых системах в разных их узлах хранится прежде всего информация, которая средствами телекоммуникационной связи может быть оперативно доставлена пользователю. Но естественна идея распределять не только данные, но и средства её обработки. Предположим, задача, которую вы решаете, требует намного большей производительности, чем способен предоставить ваш компьютер, — вы хотели бы иметь ответ, скажем, через час, а на своём компьютере сможете получить его только через 10 ч.

Если имеется мощный компьютер с требуемым программным обеспечением, то ему и следует поручить решение задачи. Вместо того чтобы в каждом университете был свой суперкомпьютер, можно иметь один на несколько учебных заведений — далеко не всем сотрудникам сразу требуется решать задачи, требующие сверхмощных ресурсов.

Распределённой называется информационная система, которая базируется на сети мощных компьютерных станций и обладает программным обеспечением, делающим возможным удалённую обработку и доставку данных по запросу пользователя.

Взаимодействие компонентов распределённой информационной системы, а также удалённый доступ пользователей к её ресурсам обеспечивает коммуникационное оборудование. К нему относятся выделенные или коммутируемые проводные и беспроводные каналы связи, различное сетевое оборудование, а также устройства приёма-передачи информации, например телефонные или радиомодемы и антенные устройства.

Неотъемлемой частью информационной системы является системное программное обеспечение, включающее операционные системы, различные операционные оболочки, повышающие уровень пользовательского интерфейса, системы программирования, служебные программы для поддержки деятельности системного администратора, сетевое программное обеспечение.

Информационные системы используют также разнообразное прикладное программное обеспечение, как типовое, так и специализированное. Типовое программное обеспечение ориентировано на классы задач, решаемых большинством информационных систем. В качестве такого обеспечения могут выступать СУБД, веб-серверы, системы текстового поиска, системы управления документами, текстовые процессоры, конверторы данных, программы распознавания текста и речи, системы электронных таблиц, генераторы отчётов для систем баз данных и др.

Специализированное прикладное программное обеспечение разрабатывается для конкретной информационной системы, например информационная система, предназначенная для поддержки деятельности руководителя конкретного предприятия, бухгалтера или библиотекаря.

Важную роль играют лингвистические средства информационных систем, которые служат для представления информационных ресурсов в системе и обеспечивают взаимодействие пользователей

с системой. К ним относятся естественные и искусственные языки, а также средства их поддержки — словари естественных языков, толковые словари предметных областей, переводные словари и др. Всё это повышает эффективность поиска требуемых документов за счёт расширения возможностей в обработке запросов пользователя.

Наконец, каждая информационная система снабжена средствами собственной безопасности, защищающими её от проникновения вирусов, от несанкционированного изменения и уничтожения информации и других нападений. На обеспечение безопасности направлены также меры разграничения прав доступа пользователей системы.

В некоторых информационных системах контингент пользователей не фиксируется. Информационные ресурсы таких систем свободно предоставляются любому пользователю. Так обстоит дело, например, со многими веб-сайтами. В других системах, для того чтобы стать пользователем, необходимо получить от системного администратора требуемые полномочия доступа к системе в целом или к некоторым её информационным ресурсам, другими словами, пройти процедуру персонификации, или регистрации. Доступ пользователей к ресурсам такой системы возможен только в пределах предоставленных им полномочий, которые обычно проверяются автоматически системными механизмами при попытках доступа. Соответствующая процедура называется идентификацией пользователя.

Одним из примеров информационной системы является автоматизированная библиотечная информационная система (сокращённо АБИС). Умение работать в такой системе — залог успеха в учебной и профессиональной деятельности. Рассмотрим основные характеристики АБИС.

1. Предметной областью АБИС являются библиографические записи, фиксирующие сведения о документе, позволяющие его идентифицировать, раскрыть его состав и содержание в целях библиографического поиска (ГОСТ 7.76—96).

2. Функции обработки информационных ресурсов составляют единый технологический процесс — от комплектования источников информации и создания документов на автоматизированном рабочем месте (АРМ) «Каталогизатор» до выполнения многофакторного поиска документов и составления различного рода отчётов.

3. Средой хранения информационных ресурсов в АБИС может выступать файловый сервер, находящийся непосредственно в библиотеке, или распределённая система серверов информационно-библиотечных комплексов в сети Интернет (WWW-адреса некоторых таких комплексов приведены в этом параграфе).

полагается нужное вам учреждение или дом, а также выяснить, каким транспортом до него добраться. Для общенациональных геоинформационных систем такие базы данных содержат различные статистические сведения (о численности населения в целом и отдельных регионах, о площади территории и т. д.). Одна из таких систем располагается по интернет-адресу nakarte.ru.

Для повышения эффективности работы и упрощения труда в делопроизводстве используются информационные системы офисной автоматизации.

Основное предназначение таких систем — обработка данных и создание новых информационных ресурсов. Коммуникационные возможности систем офисной автоматизации обеспечивают эффективное взаимодействие партнёров по профессиональной деятельности: управление документооборотом, коммуникации, составление расписаний и т. д. Вот некоторые функции систем офисной автоматизации:

- обработка текстов на компьютерах с помощью различных текстовых процессоров;
- производство высококачественной печатной продукции;
- архивация документов;
- электронные календари и записные книжки для внесения деловой информации;
- электронная почта, видео- и телеконференции.

В российской и зарубежной практике проектирования информационных систем офисного делопроизводства имеются существенные различия в технологии производства, их можно наблюдать, ознакомившись с некоторыми примерами таких систем:

- система автоматизации делопроизводства и документооборота (САДД) «Дело-96», предназначенная для автоматизации делопроизводства, основанного на традиционных отечественных технологиях и закреплённого соответствующими стандартами, и документационного обеспечения управленческой деятельности государственных организаций;
- система «Евфрат 97» фирмы «Cognitive Technologies»;
- система «Босс-Референт» компании «Ай-Ти», на базе продукта Lotus Notes фирмы Lotus Development Corporation, который представляет собой средство проектирования систем поддержки групповой работы и может рассматриваться как стандарт в этой области.

А теперь поговорим о вашем будущем. Сейчас вы учитесь и вам предстоит определиться со своей будущей профессией. В своём общественном развитии каждый человек проходит различные стадии: получение профессионального образования, поступление на работу, профессиональный рост, поддержка индивидуальных про-

фессиональных способностей, уход на пенсию. Профессиональная карьера человека во многом зависит от его способности к самоопределению в каждый конкретный момент жизни. Постоянное развитие технологий производства и информационной сферы деятельности требуют от человека получения образования на протяжении всей жизни.

Своевременная и качественная информация о рынке труда и системе профессионального образования необходима как на этапе первичного самоопределения (это возраст 15—17 лет), так и для последующей профессиональной ориентации, возникающей при необходимости изменить профессию. Но если даже вы и не собираетесь менять профессию, для карьерного роста необходима информация о системе дополнительного профессионального образования в форме повышения квалификации.

Для первичной профессиональной ориентации вам полезно воспользоваться информационно-программным комплексом «Карьера», содержащим предметную информационную систему «Профессиональное образование». Чтобы выбрать перспективную профессию, важно знать состояние рынка труда, который формируется в результате взаимодействия спроса и предложения рабочей силы. Мониторинг рынка труда (сокращённо м.р.т.) — процесс системного статистического исследования состояния рынка труда и действий органов службы занятости. С помощью мониторинга рынка труда формируется система информации, которая позволяет быстро и гибко реагировать на изменения в соотношении спроса и предложения в сфере занятости, прогнозировать возникновение критических ситуаций, планировать и осуществлять активную политику на рынке труда. К примеру, посетителям портала www.uhr.ru/index/news «Человеческие ресурсы Урала» информационная система м.р.т. предоставляет следующие материалы:

- рейтинги наиболее востребованных профессий;
- информацию о размерах заработной платы по 262 должностям в компаниях Омской, Пермской, Свердловской, Тюменской и Челябинской областей;
- динамику изменения зарплат;
- различные прогнозы и др.

Например, динамика изменения потребности в квалифицированных программистах в течение 2001—2005 гг. (по полугодиям) изображена на рисунке 16.2. Для той же категории работников данные по заработной плате можно увидеть на рисунке 16.3.

Аналогичные системы имеются в каждом регионе нашей страны. Но прежде всего важно для самого себя определить, какая профессия тебе по душе. Получить ответ на этот вопрос помогут сайты:

proekt-pro.ru — сайт, помогающий школьникам в выборе будущей профессии;

profcenter.educom.ru — сайт центра профессиональной ориентации.

Второй из них ориентирован в первую очередь на школьников, проживающих в г. Москве, но аналогичные сайты есть во многих других регионах России.

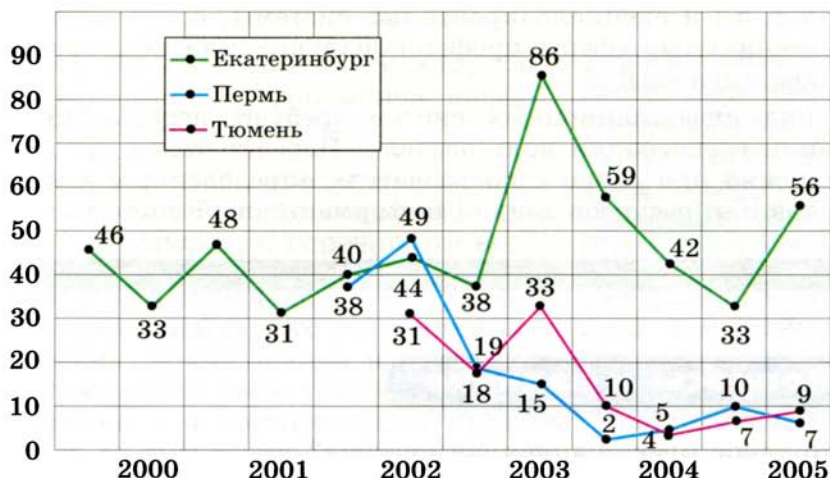


Рис. 16.2

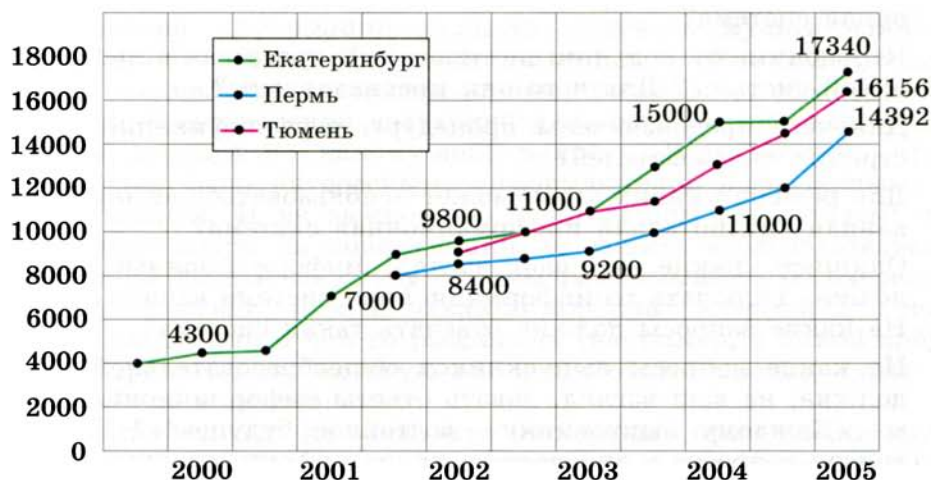


Рис. 16.3

Подведём итоги

- 1 Для обеспечения информацией значительного числа пользователей создаются информационные системы. Среди них есть системы широкого назначения (геоинформационные системы, метеорологические информационные системы, библиографические информационные системы и т. п.) и специализированные системы, обеспечивающие конкретные сферы профессиональной или общественной деятельности.
- 2 Ряд информационных систем требуют персонификации или регистрации пользователя. Нередко такая процедура нужна при запросе пользователя определённых документов или ресурсов данной информационной системы.

Вопросы и задания

- 1 Что такое информационная система?
- 2 Назовите виды информационных систем.
- 3 Какая информационная система называется распределённой?
- 4 Для решения каких задач может использоваться информационная система?
- 5 Что понимают под лингвистическими средствами информационной системы? Для чего они предназначены?
- 6 Для чего предназначены процедуры персонификации и регистрации пользователей?
- 7 Для решения каких задач может использоваться автоматизированная библиотечная информационная система?
- 8 Опишите, какие, на ваш взгляд, информационные ресурсы должна включать геоинформационная система вашего региона. На какие запросы должна отвечать такая система?
- 9 На какие вопросы выпускников общеобразовательной школы должна, на ваш взгляд, давать ответы информационная система «Каждому выпускнику — достойное будущее!»? Составьте список вопросов и определите, какие информационные ресурсы требуются для получения ответов.
- 10 Каковы цели мониторинга рынка труда?

§ 17. Электронные презентации



Как представить результаты своей работы?

■ Какие компьютерные средства позволяют создавать мультимедийные презентации?

В § 11 мы подробно обсуждали, как должен работать с информацией информационно грамотный человек.

■ Найдите в § 11 описание того, что должен уметь информационно грамотный человек, и перечитайте его.

Ключевые слова:

электронная презентация; слайд; слайд-шоу; анимация объектов

Если теперь проанализировать, что вы уже узнали в этой главе, то можно увидеть, что мы дошли до пункта: представить свою точку зрения, новые знания и понимание или решение проблемы.

Содержательно свою точку зрения вы будете формулировать самостоятельно, а о компьютерных средствах, позволяющих лучше донести её до аудитории, мы и поговорим в этом параграфе.

Электронные презентации предназначены для того, чтобы представить интересующую информацию в как можно более доступном виде. Эффективность восприятия информации возрастает, если она представлена в разных видах — видео, звуковой, текстовой. Такое представление информации принято называть мультимедийным (от лат. *multi* — много). Для создания мультимедийных документов имеется специальное инструментальное программное обеспечение.

Нередко презентации создаются для сопровождения выступления или доклада по какому-либо вопросу. Ведь хочется, чтобы выступление было интересным, информационно насыщенным, запоминающимся. И не важно, кто будет вашим слушателем: члены строгого жюри на конференции по защите исследовательских работ, ваши друзья по клубу, где вы рассказываете об очередном своём путешествии, или школьники младших классов, для которых вас попросили о чём-то рассказать (например, о какой-нибудь книге). Презентация, которую вы подготовите заранее, будет вашим надёжным помощником.

Ни один учитель не откажется от хорошей презентации изучаемого материала. Это могут быть знаменитые памятники истории, некоторые доказательства теоремы Пифагора, редкие животные нашей страны или вашего родного края, история компьютерной тех-

ники, биография знаменитого писателя и т. п. Любая тема может стать основой для создания учебной презентации.

На выставке презентация может освободить вас от необходимости много раз повторять одно и то же разным посетителям. Компьютер в отличие от человека никогда не забудет сказать все нужные слова, показать все необходимые диаграммы, продемонстрировать нужные изображения. Презентацию, выполняемую компьютером автоматически, т. е. без участия человека, обычно называют слайд-фильмом.

Основным структурным элементом презентации является слайд. Во время демонстрации один слайд сменяется другим. Работу по созданию презентации можно разбить на следующие этапы:

- подготовка: выбор темы, определение цели, круга слушателей, содержания и структуры презентации в целом и каждого слайда;
- создание слайдов: размещение объектов на слайде, выбор фона, применение эффектов, подключение звука;
- создание последовательности слайдов: определение переходов от слайда к слайду, создание меню и т. п.;
- репетиция выступления с презентацией;
- редактирование презентации: внесение изменений по результатам репетиции.

Первый этап полностью ложится на вас. А вот про второй мы сейчас расскажем.

Любая презентация начинается с так называемого титульного слайда. На нём приводятся название или тема презентации и авторы того проекта, который она представляет. Такой слайд нередко содержит логотип того учреждения, сотрудники которого осуществляли проект. К примеру, если вы на районной (городской, областной) научно-практической конференции защищаете свою творческую работу, то очень хорошо на титульном слайде разместить логотип школы, в которой вы учитесь. И всем сразу понятно — вы уважаете труд ваших наставников, которые помогли вам достичь определённых высот в знаниях и умениях.

Неплохо, если на следующем после титульного слайде вы представите основную цель и задачи вашего проекта.

Не забудьте в заключительных слайдах презентации указать источники информации, которые использовались при её создании.

Если вам захочется получить отклики на вашу работу, то на заключительном слайде укажите, как с вами связаться.

Поблагодарите слушателей за внимание к вашей работе.

А теперь поговорим о создании презентации.

На поле слайда вы можете добавлять самые разнообразные объекты: стандартные фигуры, надписи, объекты из библиотек гото-

вых рисунков, фотографии, изображения, найденные в Интернете, ваш личный рисунок и т. д. Всё это делается так же, как в текстовом редакторе Microsoft Word или Open Office Writer.

Необходимо позаботиться и о фоне слайда. Его можно создавать самостоятельно, а можно воспользоваться *Шаблонами оформления*.

Как правило, любой размещённый на слайде объект можно заставить двигаться. В этом случае говорят, что на объекте настроена анимация. Само слово «анимация» означает «оживление» (от англ. *animal* — животное). В первую очередь определяют порядок и время анимации. Можно выбрать два режима для запуска объекта в движение: по щелчку мыши или автоматически. В последнем случае надо указать время. Обычно это время составляет 2—3 с. Затем следует назначить эффекты анимации. Среди эффектов есть визуальные — возникновение, вращение, вспышка и т. п., а есть звуковые — барабан, аплодисменты и т. п. Впрочем, звук вы можете вставить из любого звукового файла.

Создав слайд, вы можете тут же просмотреть его.

Перейдём к третьему этапу. Прежде всего заметим, что предусмотрено несколько способов отображения созданной презентации (таблица 17.1).

Таблица 17.1

Режим отображения	Назначение режима
Слайд	Панель слайда занимает весь экран презентации
Обычный	Именно в этом режиме создаются файлы. Могут отображаться панели Структура, Слайды, Заметки к слайду
Страницы заметок	Каждый слайд на экране сопровождается пустой страницей ниже основного кадра. Здесь можно разместить комментарий
Сортировщик слайдов	Организовывает порядок следования слайдов
Показ слайдов	Показ слайдов в том виде, в каком они будут демонстрироваться на презентации

Режим Сортировщика слайдов позволяет целиком увидеть содержание презентации в миниатюре. Каждый слайд можно снабдить дополнительными эффектами, например указать, каким образом слайд будет появляться на экране («наплыв вверх», «выцветание» и т. п.). В режиме Сортировщика удобно перемещать, удалять, копировать и вставлять новые слайды.

Подведём итоги

- 1 Для представления результатов работы над проектом используют электронные презентации. Они позволяют передать слушателям информацию, представленную в разнообразной форме, т. е. являются мультимедийным продуктом.
- 2 Для достижения наибольшего эффекта используются различные формы подачи материала в презентации. Разумеется, здесь важно соблюдать меру — слишком большое разнообразие эффектов может отвлекать слушателя от излагаемой темы.

Вопросы и задания

- 1 В чём преимущество мультимедийных продуктов перед другими видами информации?
- 2 Для каких целей может создаваться презентация?
- 3 Каковы основные этапы разработки электронной презентации?
- 4 Подготовьте создание презентации художественного произведения, которое вы недавно прочитали.
 - а) Выберите произведение, презентацию которого вы намерены сделать. Продумайте, какая информация должна быть сообщена в первую очередь. Составьте план, как эта информация будет размещена на первом слайде.
 - б) Продумайте, как вы можете представить главных героев произведения, в каком порядке они будут появляться в вашей презентации. Составьте план, как должен выглядеть соответствующий слайд (или последовательность слайдов), какие эффекты анимации в этом случае были бы полезны.
 - в) Продумайте, какими событиями могут быть представлены основные сюжетные линии, как обрисовать взаимоотношения героев. Если вам нужны графические образы, продумайте, как они могут быть созданы.
 - г) Продумайте, как выразить ваше отношение к произведению.
- 5 Подготовьте создание рекламы какого-либо любимого вами продукта, например мороженого.
- 6 Создайте электронное учебное пособие для первоклассников — «Живая азбука». Выполнение этого задания распределите

те между учениками класса. Каждый участник проекта должен создать несколько слайдов.

а) Каждой букве алфавита отводится один слайд. Продумайте, какие слова и какие соответствующие им изображения животных должны иллюстрировать употребление данной буквы.

б) Продумайте структуру каждого из слайдов — она должна быть единой. Определите, как будет осуществляться навигация. Какие в связи с этим требования должны быть объявлены разработчикам слайдов?

в) Придумайте, как можно организовать проверку того, как первоклассник — пользователь вашего продукта — выучил употребление данной буквы. К примеру, можно показать букву и набор изображений предметов, из которых он должен выбрать те, в названиях которых встречается данная буква. Более сложный вариант — показывается набор изображений предметов, в названиях которых встречается одна и та же буква; обучаемый должен найти эту букву.

Темы для размышления и обсуждения

- 1** Какова роль устного выступления, если результаты вашей работы оформлены в виде презентации?
- 2** «Живая азбука», проект создания которой описан в задании 6, является, по сути, электронным учебным пособием. Как лучше использовать это пособие — в индивидуальной работе учащихся или при работе класса целиком?

Часть 1. При выполнении предложенных ниже заданий запишите в тетради шифр задания и номер правильного ответа. По окончании работы сверьте получившуюся у вас запись с ключами, приведёнными в конце учебника.

A1. Информация « $2 + 2 = 4$ » для выполнения этого задания:

- 1) достоверна, актуальна, объективна, полезна, полна;
- 2) достоверна, актуальна, объективна, бесполезна, полна;
- 3) достоверна, неактуальна, объективна, полезна, полна;
- 4) достоверна, неактуальна, объективна, бесполезна, полна.

A2. В комедии Н. В. Гоголя «Ревизор» П. И. Бобчинский и П. И. Добчинский сообщают городничему о приезде ревизора из Санкт-Петербурга. Эта информация:

- 1) достоверна, актуальна, объективна, неполна;
- 2) недостоверна, актуальна, полезна, полна;
- 3) актуальна, объективна, полезна, полна;
- 4) недостоверна, неактуальна, объективна, неполна.

A3. Полезность полученной информации означает, что эта информация:

- 1) соответствует реальности;
- 2) достаточна для достижения поставленной цели;
- 3) способствует принятию правильного решения;
- 4) оказывает влияние на понимание ситуации, имеющейся в данный момент времени.

A4. Адресный поиск информации — это поиск

- 1) по ключевым словам;
- 2) путём указания факта, информация о котором разыскивается;
- 3) по адресу электронной почты;
- 4) путём указания точных параметров источника, содержащего разыскиваемую информацию.

A5. Желая узнать, кто автор стихотворения «Как хороши и как свежи были розы», пользователь в поисковой строке браузера набрал эту фразу. Данный запрос является:

- 1) адресным;
- 2) семантическим;
- 3) фактографическим;
- 4) неявным.

А6. Форму адреса электронной почты имеет запись:

- 1) <http://ya.ru>
- 2) some@proba.ru
- 3) <ftp://ege.com>
- 4) <http://solv@mail.ru>

А7. Форму адреса во Всемирной паутине имеет запись:

- 1) <http://www.el.ru>
- 2) <ftp://fakt.ru>
- 3) one@norht.ru
- 4) <http://night@mail.ru>

А8. В адресе электронной почты слева от символа @ указывается:

- 1) вид сервиса глобальной сети;
- 2) доменное имя хост-компьютера;
- 3) идентификатор конечного пользователя;
- 4) иная, нежели указанная в пунктах 1—3, информация об адресате.

А9. Дан фрагмент однотабличной базы данных (в первой строке указаны имена атрибутов):

Таблица: Озёра

Название	Глубина, м	Площадь, кв. км	Часть света	Высота над уровнем моря, м
Аральское	68	33 640	Азия	53
Байкал	1741	31 500	Азия	455
Гурон	228	59 700	Америка	177
Маракайбо	250	13 512	Америка	0
Виктория	80	69 485	Африка	1134
Чад	4	11 000	Африка	240
Ладожское	225	17 700	Европа	5
Севан	99	1400	Европа	1914

Количество записей в этом фрагменте, удовлетворяющих запросу
Площадь > 11 000 и (Высота < 200 или Глубина < 250),

равно:

- 1) 1; 2) 3; 3) 5;
4) другому числу, нежели указаны в пунктах 1—3.

А10. Дан фрагмент базы данных:

Таблица: Учащиеся

№	Фамилия	Имя	Класс	Шко- ла
1	Смирнов	Николай	9	15
2	Иванов	Вадим	9	28
3	Белов	Сергей	10	135
4	Петров	Павел	11	36
5	Сидоров	Андрей	11	72

Таблица: Школы

№	Шко- ла	Округ
1	135	Центральный
2	72	Юго-Западный
3	36	Юго-Западный
4	28	Центральный
5	15	Юго-Западный

Согласно запросу по признаку

Округ = Юго-Западный

формировалась таблица-результат с атрибутами *Фамилия*, *Имя* и *Класс*, причём записи в ней были отсортированы по номерам школ в порядке возрастания:

№	Фамилия	Имя	Класс
---	---------	-----	-------

После исполнения запроса в строке с № 1 таблицы-результата будет стоять фамилия

- 1) Иванов; 2) Петров; 3) Сидоров; 4) другого ученика.

Часть 2. При выполнении предложенных ниже заданий запишите в тетради шифр задания и рядом ответ в виде последовательности символов. По окончании работы сверьте получившуюся у вас запись с ключами, приведёнными в конце учебника.

В1. Доступ к файлу doc.com, расположенному на российском сервере, осуществляется по протоколу http. Фрагменты адреса

закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети Интернет.

А	Б	В	Г	Д	Е	Ж
/	мысль	.com	://	.рф	http	doc

В2. Ниже приведены запросы к поисковому серверу, обозначенные буквами от А до Г. Расположите обозначения запросов в порядке невозрастания количества страниц, которые найдёт поисковый сервер по каждому из запросов.

Для обозначения логической операции *или* в запросе используется символ |, а для обозначения логической операции *и* — символ &.

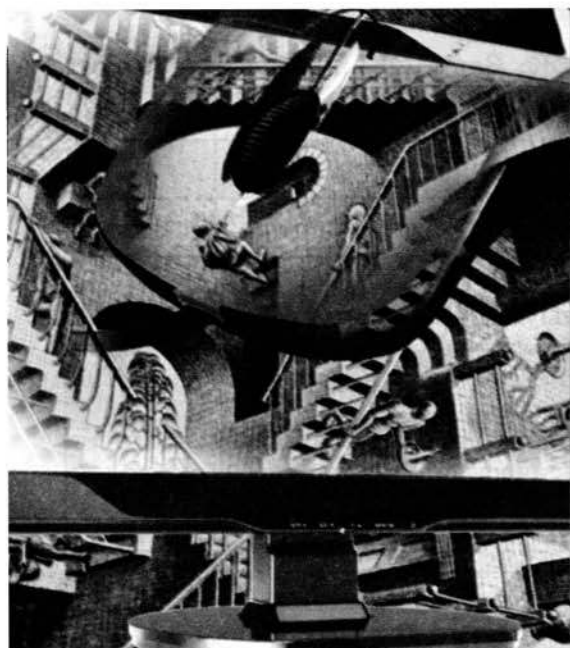
А	Упоров
Б	Семёнов & Упоров
В	Семёнов Упоров
Г	Семёнов & Королёв Упоров

В3. В базе данных отдела кадров небольшой фирмы имеется 50 записей о рабочих трёх категорий: токарях, слесарях и разнорабочих. Среди атрибутов в этих записях, кроме специальности, фигурирует атрибут *Пол* со значениями «м» или «ж». Количество записей *N*, удовлетворяющих различным запросам, приведено в таблице:

Запрос	<i>N</i>
<i>Пол</i> = м и <i>Специальность</i> ≠ разнорабочий	28
<i>Специальность</i> ≠ слесарь	35
<i>Пол</i> = ж и <i>Специальность</i> ≠ токарь	12

Укажите количество записей, удовлетворяющих запросу

Пол = м и *Специальность* = токарь или
Пол = ж и *Специальность* = разнорабочий.



Компьютерный практикум

Лабораторная работа № 1

В КОМПЬЮТЕРНОМ КЛАССЕ ПОСЛЕ ЛЕТНИХ КАНИКУЛ

Прошло лето, и мы считаем нелишним напомнить правила поведения в компьютерном классе.

1. Если вы обнаружили какую-либо неисправность, немедленно сообщите об этом преподавателю. Не работайте на неисправном оборудовании!

2. Не включайте и не выключайте компьютеры самостоятельно.

3. Не дергайте и вообще не трогайте различные провода.

4. Не стучите по клавиатуре и мыши.

5. Не садитесь за клавиатуру с грязными руками.

А теперь приступайте к работе за компьютером.

1 Запустите клавиатурный тренажёр и проверьте свои навыки по набору текста.

2 Наберите в текстовом редакторе следующее стихотворение:

«У простого сторожа
Непросторный дом:
Часто в нём стоножка
Бродит под столом.
Дорожит стоножка
Чистотю ног

И столичной ваксой
Чистит сто сапог.
Вместо двух непросто
Вычистить все сто,
Сразу столько обуви
Не носил никто!
У простой стножки
Стоит постоять
И у той стножки
Опыт перенять!!!»

А теперь замените все слова «сто», которые встречаются этом тексте, на число 100. Разумеется, сделайте это не в ручном, а автоматическом режиме редактора!

В конце 7 класса вы начали составлять программы для довольно простого исполнителя — Паркетчика.

- 3** Пусть на поле Паркетчика выложен какой-нибудь орнамент из плиток красного и зелёного цвета. Напишите программу, исполнив которую Паркетчик заменит все красные плитки на зелёные, а все зелёные — на красные. (Напомним, что все слова, которые понимает Паркетчик, занесены в меню, расположенное в верхней строке окна, и выбор слова в меню обеспечивает его автоматическое появление в тексте программы.)
- 4** Напишите программу, исполнив которую Паркетчик выложит сначала рамку из красных плиток вдоль границ поля, а затем рамку из зелёных плиток вдоль границ красной рамки.

Лабораторная работа № 2

СЛОЖНЫЕ УСЛОВИЯ В ПРОГРАММАХ ДЛЯ ПАРКЕТЧИКА

-
- 1** Наберите программу, которую вы составили, выполнив задание 13, а из § 1, и проверьте, как её исполняет Паркетчик для поля исходных размеров. А теперь уменьшите ширину поля, превратив его в вытянутый по вертикали прямоугольник. Справляется ли Паркетчик со своей работой на таком поле? Если нет, сделайте необходимые исправления в программе. Ещё раз поменяйте размеры поля так, чтобы теперь оно стало вытянутым по горизонтали, и проверьте, как Паркетчик справляется с работой на таком поле.
 - 2** Прodelайте такую же работу, как в предыдущем задании для программы, которую вы составили, выполнив задание 13, б из § 1. После окончания экспериментов верните полю исходные размеры.

- 3** Установите размеры поля такими, чтобы получился прямоугольник, вытянутый по горизонтали. Наберите программу, которую вы составили, выполнив задание 14 из § 1, и проверьте, как её исполняет Паркетчик для этого поля.
- 4** Расположите на поле несколько трёхклеточных красных полосок (часть вертикально, часть горизонтально; полоски не должны перекрывать друг друга), а также трёхклеточных уголков (здесь возможны четыре варианта расположения, и пусть все они присутствуют на поле). Наберите программу, которую вы составили, выполнив задание 15, а из § 1, и проверьте, как её исполняет Паркетчик.
- 5** Прodelайте такую же работу, как в предыдущем задании для программы, которую вы составили, выполнив задание 15, б из § 1.

Лабораторная работа № 3

ЦИКЛЫ СО СЧЁТЧИКОМ

- 1** Пусть на поле Паркетчика выложен какой-нибудь орнамент из плиток красного и зелёного цвета. Выполнив задание 3 из лабораторной работы № 1, вы составили программу изменения цвета у всех плиток, лежащих на поле. Но её можно значительно упростить, зная, что размеры поля Паркетчика 27×27 клеток, и используя алгоритмическую конструкцию цикла со счётчиком. Вот как может выглядеть эта программа:

Программа

```

{ Делать от СТОЛБЕЦ := 1 до 27
  { Делать от СТРОКА := 1 до 27
    { Перейти на (СТОЛБЕЦ, СТРОКА);
      Если (з) то
      { Снять плитку;
        Положить (к);
      }
      Если (к) то
      { Снять плитку;
        Положить (з);
      }
    }
  }
}

```

Чтобы превратить этот алгоритм в программу для Паркетчика, ему надо объяснить, что скрывается за словами **СТОЛБЕЦ** и **СТРОКА**. Для этого надо написать оператор:

цел: **СТОЛБЕЦ**, **СТРОКА**;

Окончательно получится:

Программа

цел: **СТОЛБЕЦ**, **СТРОКА**;

```
{ Делать от СТОЛБЕЦ := 1 до 27
  { Делать от СТРОКА:= 1 до 27
    { Перейти на (СТОЛБЕЦ, СТРОКА);
      Если (з) то
      { Снять плитку;
        Положить (к);
      }
      Если (к) то
      { Снять плитку;
        Положить (з);
      }
    }
  }
}
```

- 2** Создайте на поле Паркетчика какой-нибудь орнамент из красных и зелёных плиток. Теперь наберите программу и запустите её на исполнение. Работает ли она так, как требуется в задаче? Если да, то переходите к выполнению следующего задания, иначе займитесь отладкой программы.
- 3** Выполнив задание 13 из § 2, вы составили программу, очищающую поле от каких бы то ни было плиток. Создайте поле указанных в этом задании размеров и разбросайте по нему красные и зелёные плитки. Запустите вашу программу и проверьте, как Паркетчик справляется с уборкой мусора.
- 4** Проверьте, как Паркетчик исполняет вашу программу, написанную по заданию 14 из § 2.
- 5** Проверьте, как Паркетчик исполняет ваши программы, написанные по заданию 15 из § 3.

Лабораторная работа № 4

ПЕРЕМЕННЫЕ В ЖИЗНИ ПАРКЕТЧИКА

- 1** Выполнив задание 6 из § 3, вы написали две программы для Паркетчика (размеры поля неизвестны!). Проверьте, как он исполняет эти программы на полях различных размеров.
- 2** Проверьте, как Паркетчик исполняет ваши программы, написанные по заданию 7 из § 3.
- 3** В заданиях предыдущей лабораторной работы мы пользовались тем, что изначально известны размеры поля для Паркетчика. Но ведь это не всегда так! А знать размеры поля при составлении программ весьма удобно. Составьте программу, исполнив которую Паркетчик узнает и сообщит вам размеры поля, на котором он работает.
- 4** Если у вас осталось время, попытайтесь научить Паркетчика выполнять отражение относительно оси (задание 8 из § 3). Размеры поля заранее неизвестны, поэтому ваша программа должна правильно работать как при чётном числе столбцов на поле, так и при нечётном. (Совет: подумайте, может быть, удобно начать с того, что выяснить размеры поля и запомнить их в каких-либо переменных.)

Лабораторная работа № 5

ПОДПРОГРАММЫ ДЛЯ ПАРКЕТЧИКА

Версии Паркетчика, как и любого программного обеспечения, время от времени обновляются и становятся более удобными. Одно из неудобств некоторых старых версий Паркетчика (работающих под ОС Windows) состоит в том, что тексты подпрограмм должны располагаться обязательно до текста основной программы. Поэтому мы советуем придерживаться этого правила.

- 1** Наберите подпрограмму и программу рисования квадратных цветных рамок, которые приведены в объяснительном тексте § 6. Проследите, как Паркетчик их исполняет. Измените программу так, чтобы Паркетчик выкладывал по горизонтали 4 рамки, а по вертикали 3 (всего 12 рамок). Потребовалось ли для этого что-то менять в подпрограмме?
- 2** Наберите подпрограмму и программу рисования квадратных цветных рамок, которые приведены в задании 2, а из § 6. Проследите, как Паркетчик их исполняет. Измените программу так, чтобы Паркетчик выкладывал орнамент, изображённый на рисунке 6.2.

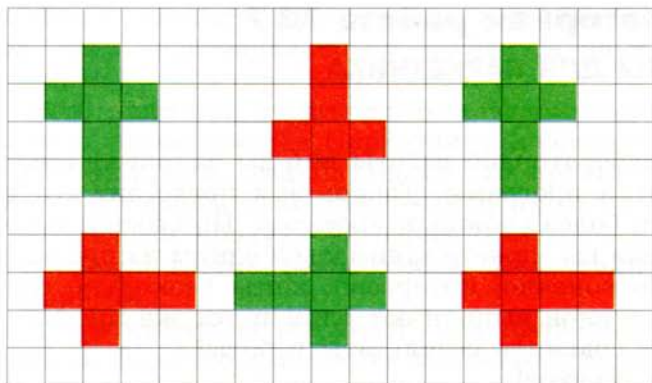


Рис. КП. 1

- 3 Выполнив задание 3 из § 6, вы составили программу и подпрограммы, обеспечивающие рисование узора, изображённого на рисунке 6.3. Проследите, как справляется с ними Паркетчик. Годится ли составленный вами комплект подпрограмм, если потребуется составить программу для рисования узоров, показанных на рисунке КП.1? Продумайте, какие подпрограммы были бы полезны для этого, и составьте их.
- 4 Наберите подпрограмму и программу, которые приведены в задании 4 из § 6. Пронаблюдайте, как Паркетчик их исполняет. Совпал ли орнамент, выложенный Паркетчиком, с орнаментом, который нарисовали вы, выполняя это задание вручную?

Лабораторная работа № 6

КОЛЛЕКТИВНАЯ РАЗРАБОТКА ПРОГРАММЫ

Выполняя задание 5 из § 6, вы для решения задачи создали небольшие коллективы разработчиков программ. Пришла пора проверить, как эти коллективы справились с поставленной задачей.

- 1 Под контролем руководителя группы соберите составленные вами подпрограммы в единый комплекс так, чтобы основная программа могла с ними работать. Проверьте работоспособность созданного вами комплекса.
- 2 Проведите аналогичную работу по выполнению задания 6 из § 6.

Лабораторная работа № 7

ПРОЕКТЫ ДЛЯ ПАРКЕТЧИКА

В ходе лабораторной работы № 6 вы накопили опыт коллективной разработки программ. Теперь вам предстоит его использовать для создания более сложных проектов. По своему выбору или указанию учителя вы можете заниматься одним из предлагаемых нами проектов. Но конечно, интересно, когда несколько групп (хотя бы две) разработчиков выполняют один и тот же проект — ведь тогда можно будет сравнить результаты работы!

Желаем успехов!

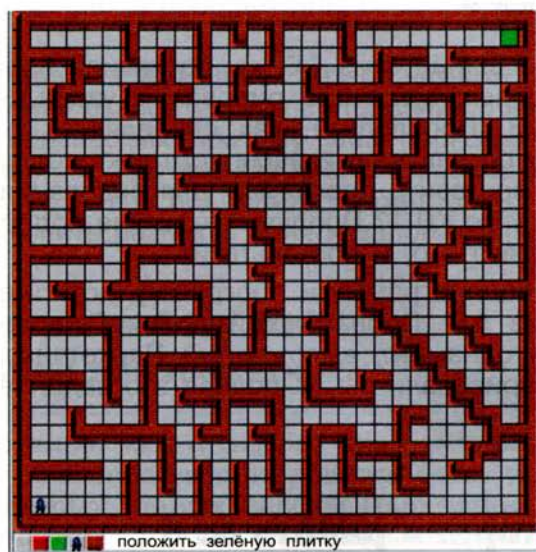


Рис. КП. 2

Проект № 1. Обход лабиринта

Многие из вас, возможно, обратили внимание на ещё один квадратик, расположенный под полем Паркетчика. А может быть, некоторые и пробовали положить его на поле. На самом деле это строительный материал, из которого на поле можно выкладывать стены. Сквозь стену Паркетчик, естественно, пройти не может, как не может он встать и на ту клетку, где построена стена. Таким образом, поле можно превратить в настоящий лабиринт (рис. КП. 2)! Зелёная плитка обозначает выход из лабиринта. Вы можете установить её на любом другом поле, свободном от стен. Да и сам лабиринт вы можете построить по своему вкусу.

Ваша задача — научить Паркетчика добираться до выхода из лабиринта, если первоначально он находится на клетке (1, 1). Поэтому не занимайте клетку (1, 1) стеной.

Прежде всего надо придумать идею и сформулировать какое-то общее правило, позволяющее Паркетчику добраться до выхода. Затем составить алгоритм, реализующий это правило. Потом написать комплекс программ и подпрограмм и их отладить. А после можно предлагать свой продукт на общее обсуждение — пусть все убедятся, что, следуя вашим разработкам, Паркетчик выбирается из любого лабиринта.

Возможно, что кто-то из вас знает, как правильно искать выход из лабиринта. Если по лабиринту двигаться, держась всё время правой рукой за стенку, то в конце концов доберётесь до выхода. Эта идея поиска выхода называется правилом правой руки. Разумеется, можно с таким же успехом пользоваться правилом левой руки — сформулируйте его самостоятельно.

Но Паркетчику не скажешь: «Руководствуйся правилом правой руки»! Надо это правило превратить в набор формальных инструкций. Одна из таких инструкций может звучать так: пока снизу стена, двигайся вправо, если справа нет стены. Эта инструкция реализует правило правой руки при движении вдоль стены, расположенной снизу от Паркетчика.

- 1** Сформулируйте аналогичные инструкции по перемещению Паркетчика вдоль правой, верхней и левой стен.
- 2** Запишите каждую из инструкций в виде подпрограммы.
- 3** Используя составленные подпрограммы, разработайте программу обхода лабиринта.
- 4** Отладьте получившийся комплекс программ.

Разработанный вами алгоритм годится, если лабиринт «честный», т. е., путешествуя по нему, действительно можно добраться до выхода, а сам выход располагается около внешней стены лабиринта. А если лабиринт устроен так, как показано на рисунках КП. 3 и КП. 4?

- 5** Что произойдёт с Паркетчиком, если предложить ему исполнить составленную вами программу в этих лабиринтах?

Вы видите, что исполнение программы заикливается — Паркетчик снова и снова возвращается туда, где он уже был. Значит, надо сделать так, чтобы он узнавал об этом!

- 6** Придумайте, как Паркетчик может информировать себя о том, что он уже был на этой клетке.

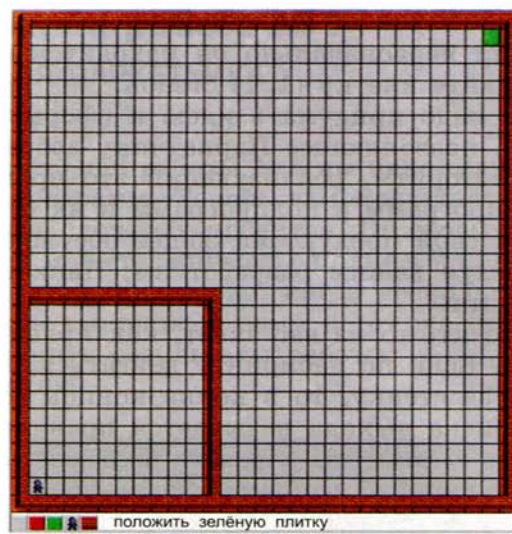


Рис. КП. 3

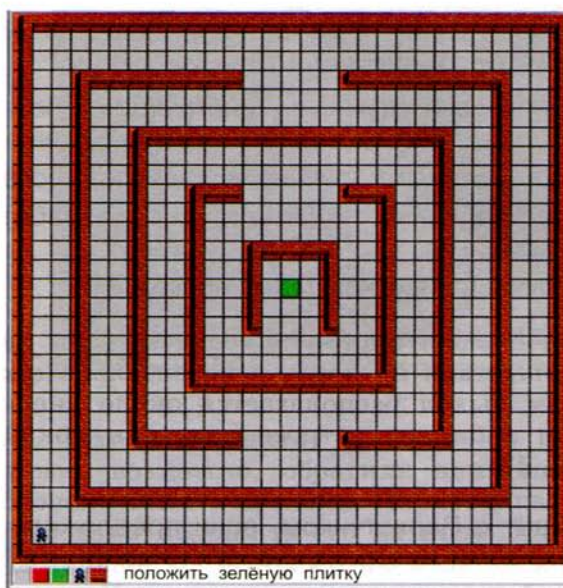


Рис. КП. 4

- 7** Разработайте комплекс программ, обеспечивающих Паркетчику выход из лабиринта, если это возможно, и передачу сообщения SOS, если до выхода он добраться не может.

Проект № 2. Жизнь

Эту игру придумал Д. Конуэй. На бесконечном листе бумаги живут колонии покрашенных клеток. Эти клетки рождаются и умирают (по определённым правилам), а их колония эволюционирует во времени. Вот правила игры:

а) Если покрашенная клетка имеет четырёх или более покрашенных соседей, то она умирает (т. е. перестаёт быть покрашенной) от перенаселённости (всего у каждой клетки 8 соседей).

б) Если покрашенная клетка не имеет покрашенных соседей или имеет ровно одного покрашенного соседа, то она умирает от нехватки общения.

в) Если непокрашенная клетка имеет ровно трёх покрашенных соседей, то она окрашивается (в этом случае говорят, что родилась новая покрашенная клетка).

Время в этой игре течёт дискретно (т. е. по шагам), и за один шаг с каждой клеткой поля происходит (или не происходит) изменение, описанное в вышеуказанных правилах. Договоримся называть старыми те клетки, которые на очередном шаге должны будут умереть.

Играют в эту игру так: рисуют какую-нибудь колонию покрашенных клеток и затем шаг за шагом прослеживают её эволюцию.

1 Прodelайте это для колоний, изображённых на рисунках КП. 5.

Вручную стирать и окрашивать клетки не так уж интересно. Это работа для формального исполнителя. И Паркетчик, наверно, вполне может справиться с такой работой. Давайте обучим Паркетчика правилам игры, и пусть он показывает нам эволюцию колоний.

Фактически Паркетчика надо научить двум вещам: класть новые плитки и снимать старые. Для каждой отдельной клетки Паркетчик легко сосчитает, сколько у неё соседей, и решит, должна ли родиться новая клетка или умереть старая. Проблема, однако, в том, что если к существующей колонии добавить покрашенную

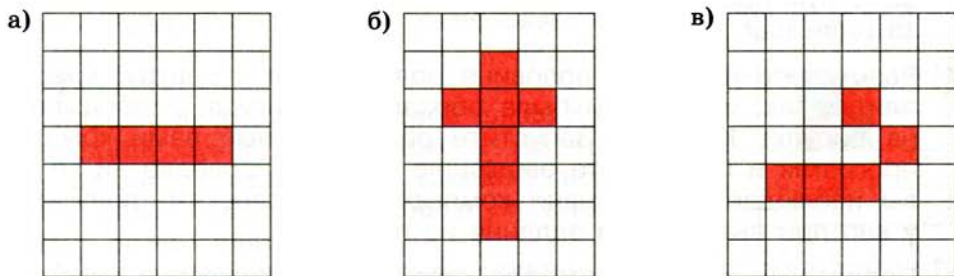


Рис. КП. 5

клетку или удалить какую-либо закрашенную клетку, то для других клеток ситуация может оказаться уже не той, какая была перед этим, хотя процесс определения рождающихся и умирающих на данном шаге клеток ещё не завершён.

Как же разрешить эту проблему? Как разделить во времени два процесса — получение информации о том, где плитки должны исчезнуть и где появиться, и само преобразование колонии закрашенных клеток?

Казалось бы, мы стоим перед неразрешимой задачей. Но ещё древние люди советовали: хочешь выиграть время — пожертвуй пространством. В технологическом мире это правило тоже хорошо известно и звучит так: если не можешь разделить два процесса во времени, раздели их в пространстве. А в компьютерном мире звучит ещё конкретнее: хочешь, чтобы программа работала быстрее, придётся выделить ей больше оперативной памяти.

Вернёмся к нашей проблеме. Раз нет возможности разделить эти процессы во времени, придётся пожертвовать пространством, т. е. частью поля, играющего для Паркетчика роль оперативной памяти. А именно мысленно разделим поле на две равные части — с первого столбца по тринадцатый и с пятнадцатого по двадцать седьмой. На левой части будет жить и эволюционировать наша колония клеток, а на правой фиксироваться информация, где должна появиться новая закрашенная клетка, какая клетка перестанет быть закрашенной, а какая останется без изменений. Фактически на правой половине поля будет постепенно — клетка за клеткой — создаваться копия колонии, которая должна появиться на следующем шаге. Затем её можно «перерисовать» на левую половину поля, правую очистить от закрашенных клеток и приступить к построению колонии на следующем шаге.

- 2** Следуя этой идее, напишите подпрограммы для создания копии колонии клеток после очередного шага и очистки правой половины поля от плиток. Отладьте эти подпрограммы и напишите основную программу для изображения эволюции колонии клеток за один шаг.
- 3** Разместите на левой половине поля (ближе к центру) красные плитки так, чтобы ими была образована фигура, изображённая на рисунке КП. 5, а. Запустите разработанный вами комплекс программ и проследите эволюцию колонии. Совпало ли то, что вы наблюдаете на экране компьютера, с тем, что получалось у вас при выполнении задания из пункта 1?
- 4** Проделайте то же самое для колоний, изображённых на рисунках КП. 5, б и КП. 5, в.

Если созданный вами программный комплекс работает безошибочно, обсудим следующее.

Возможно, что через несколько шагов какая-нибудь клетка колонии окажется на границе поля. Но тогда у неё уже не 8 соседей! Чтобы справиться с этой проблемой, давайте будем считать, что наше поле склеено противоположными сторонами. Тогда соседними для клетки (1, 3) будут не только клетки, которые расположены правее её и выше, но и клетки (13, 2), (13, 3) и (13, 4). А для клетки (5, 27) соседями будут не только клетки снизу и с боков, но и клетки (4, 1), (5, 1) и (6, 1). Каковы соседи клетки (1, 1)? Это не только клетки (1, 2), (2, 2) и (2, 1), но и клетки (13, 1), (13, 2), (1, 27), (2, 27), а самое удивительное, что ещё и клетка (13, 27). Обдумайте это.

- 5** Перечислите соседей у каждой из клеток (1, 27), (13, 27) и (13, 1).
- 6** Напишите комплекс программ, позволяющих следить за эволюцией клеток на таком поле. Ясно, что правую половину поля Паркетчика надо тоже понимать склеенной противоположными сторонами.
- 7** Выложите на левой половине поля ещё какие-нибудь колонии клеток и проследите за их эволюцией.
- 8** Попытайтесь найти все колонии, состоящие не более чем из 8 закрашенных клеток, которые будут жить вечно.

Проект № 3. Игра «Реверси»

Игра, о которой идёт речь в этом проекте, проводится на клетчатом поле. В неё играют два игрока, которые делают ходы по очереди, и каждый игрок имеет комплект фишек (ограниченный или неограниченный) своего цвета или формы. К таким играм относятся «Крестики-нолики», «Уголки», «Го» и т. п. А вот шашки такой игрой не являются — там у каждого игрока может быть два вида фишек: простые и дамки.

Возможно, что не все знают правила игры «Реверси». Поэтому расскажем их, причём уже в том виде, который подойдёт для Паркетчика.

Играют двое на квадратной доске 8 × 8. В середине лежат две зелёные и две красные плитки (рис. КП. 6). Каждый игрок своим ходом кладёт на свободную клетку плитку своего цвета.

Если при очередном ходе только что положенная плитка и какая-то плитка того

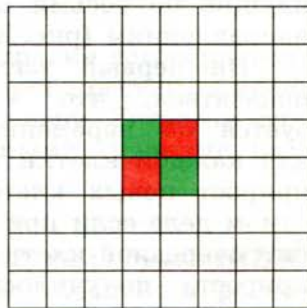


Рис. КП. 6

же цвета ограничивают ряд плиток противника (этот ряд не должен содержать пробелов из непокрытых клеток), то все плитки этого ряда заменяются на плитки игрока, сделавшего ход (ряд может быть горизонтальным, вертикальным или диагональным). Если таких рядов одновременно возникает несколько, то замена производится во всех таких рядах.

Игра заканчивается, когда на поле не осталось непокрытых клеток. Выигрывает тот игрок, чьих плиток в этот момент оказалось на поле больше.

Чтобы лучше освоиться с этой игрой, можете сначала поиграть друг с другом. Для этого не обязательно изготавливать плитки красного и зелёного цвета — можно играть обыкновенными шашками.

Поиграли? А теперь давайте научим играть в неё Паркетчика. Собственно говоря, нам надо научить его делать ход — класть плитку своего цвета и, если возникает необходимость, заменять плитки противника своими плитками.

Правда, хороший игрок, прежде чем сделать ход, думает, куда лучше ходить. Бывают игроки, которые сначала ходят, а потом думают, но они нам не пример.

Как же выбрать, куда ходить? Первое, что приходит в голову, — выбрать такую клетку, чтобы число своих плиток после хода на эту клетку стало максимальным по сравнению с количеством своих плиток после хода на другую клетку.

Значит, перебирая по очереди все клетки поля, находим ту, для которой прирост своих клеток максимален. Этот прирост можно запомнить в некоторой переменной. А чтобы подсчитать его для клетки (I, J) , надо обследовать ряды клеток, начинающиеся с клетки (I, J) и идущие по восьми возможным направлениям (рис. КП. 7).

На первый взгляд может показаться, что нам потребуется 64 переменные, чтобы для каждой клетки подсчитать прирост новых клеток. Но на самом деле если при обследовании очередной клетки значение прироста получилось больше, то зачем нам помнить меньшее значение?

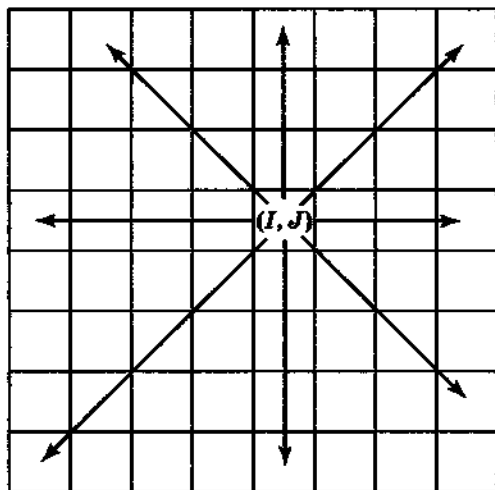


Рис. КП. 7

- 1** Составьте подпрограмму поиска клетки, после хода на которую прирост ваших клеток будет максимальным.
- 2** Можно заранее договориться, плитками какого цвета играет Паркетчик. Пусть в основной программе Паркетчик предлагает своему противнику выбрать цвет плиток, а Паркетчик тогда будет играть плитками другого цвета. Кроме того, пусть Паркетчик вежливо предложит сопернику выбрать, кто сделает первый ход. Составьте основную программу, отладьте её и поиграйте с Паркетчиком, первый ход сделайте сами, а затем предложите сделать ход Паркетчику.
- 3** Составленная программа показывает, что Паркетчик не очень-то опытный и дальновидный игрок. Ведь вполне может оказаться, что, пожадничав, он после ответного хода противника проиграет куда больше. Хорошо бы научить Паркетчика анализировать возможные ответы противника и принимать своё решение с учётом результата этого анализа. Составьте соответствующую программу. Стал ли после этого Паркетчик играть лучше?
- 4** Можно ещё больше «нарастить интеллект» Паркетчика, запрограммировав анализ большего числа ходов — как своих, так и противника. Составьте программу, в которой по желанию можно было бы настраивать глубину такого анализа. Чем более глубокий анализ вы предложите осуществлять Паркетчику, тем на более длительное время он будет задумываться, прежде чем сделать очередной ход.
- 5** До этого момента мы предполагали, что противником Паркетчика является человек. Но вполне можно устроить турнир составленных вами программ. Для этого программу, составленную каждой группой разработчиков, надо оформить в виде подпрограммы, а основная программа будет только по очереди передавать ход этим подпрограммам. Организуйте такой турнир и определите программу-победительницу.

Проект № 4. Игра «Крестики-нолики»

Это всем достаточно известная игра. Надо только договориться, что, например, зелёная плитка на поле обозначает крестик, а красная — нолик. Научите Паркетчика играть в эту игру. Устройте турнир между программами, составленными разными группами разработчиков.

Проект № 5. Игра «Уголки»

Начальная позиция игры «Уголки» показана на рисунке КП. 8. Цель игры — перевести свои фишки на места начального расположения фишек партнёра. Выигрывает тот, кто это сделает первым. За один ход можно передвинуть одну свою фишку на любое

свободное соседнее поле (по вертикали, горизонтали или диагонали) или совершить своей фишкой один или несколько прыжков, каждый из которых производится на свободное поле через одно поле, обязательно занятое фишкой (неважно, своей или чужой). Во время совершения серии прыжков направление можно менять (например, первый прыжок сделать по вертикали, а второй — по диагонали).

Научите Паркетчика играть за одного из игроков. Вторым игроком может быть человек, а может быть программа «конкурирующей фирмы» — другой группы разработчиков такого же проекта. Проведите турнир и определите программу-победительницу.

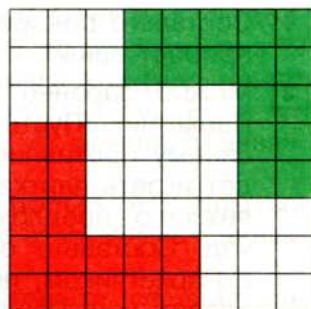


Рис. КП. 8

Лабораторная работа № 8

ЗНАКОМСТВО С ЯЗЫКОМ ПРОГРАММИРОВАНИЯ

Выполняя задания 3—6 из § 9, вы написали несколько программ на изучаемом вами языке программирования. Теперь для каждой из них проведите компьютерный эксперимент с различными исходными данными.

Лабораторная работа № 9

ПОИСК ИНФОРМАЦИИ В ИНТЕРНЕТЕ

Начнём с простого: пусть вам известно, где взять нужную информацию. Это значит, что вы знаете, на каком сервере хранится данная информация, в какой папке (или в каком каталоге) она располагается и, быть может, каково имя файла. Иными словами, вы знаете универсальный указатель ресурса, предоставляющего нужную информацию.

- 1 Возможно, у вас есть на примете какой-нибудь интересный URL, но мы предлагаем воспользоваться следующим:
<http://www.hermitage.ru>.

- а) Вызовите имеющийся на вашем компьютере браузер и запишите в строку вызова указанный URL.
- б) Перед вами главная страница одного из самых известных музеев мира. Пользуясь гиперссылками, просмотрите страницы, на

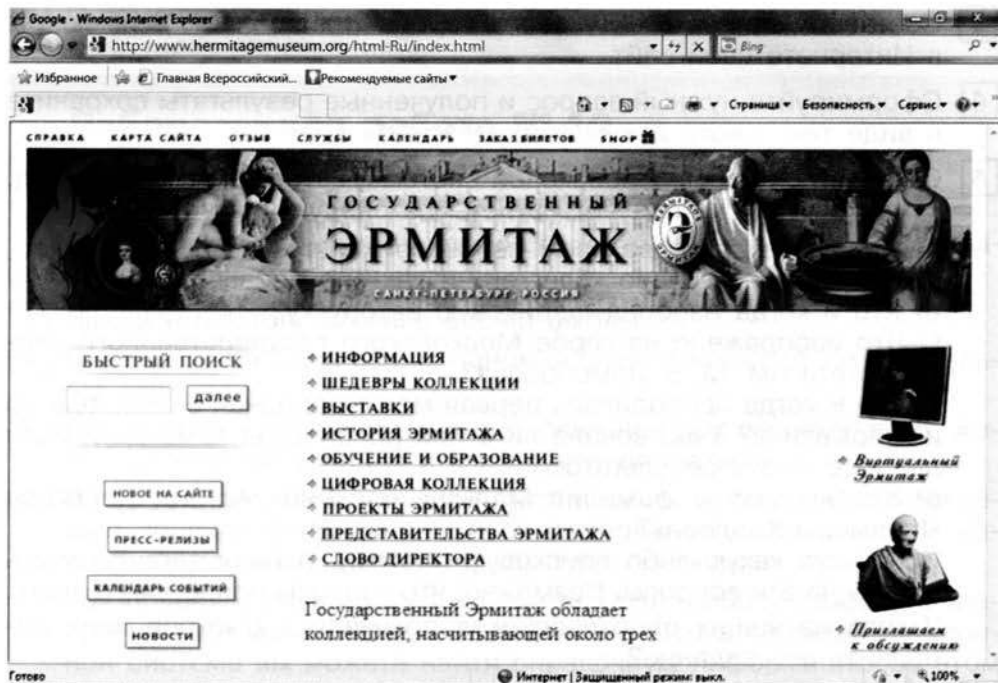


Рис. КП. 9

которых рассказывается об истории этого уникального хранилища (рис. КП. 9).

в) С помощью буфера обмена скопируйте текст нескольких страниц, посвящённых истории музея, и вставьте его в текстовый редактор.

г) Отформатируйте получившийся документ так, чтобы он смотрелся как реферат.

д) Сохраните документ с подходящим именем.

Если вы чувствуете, что полностью разобрались с первым заданием, то переходите ко второму. Пусть вам, к примеру, понадобится узнать, как выглядит статья 14 закона «Об образовании» в России.

2 Перейдите на сайт какой-нибудь поисковой системы и сформируйте соответствующий запрос. Найденную статью сохраните в виде отдельного документа.

Мы надеемся, что и с этим заданием вы справитесь легко. Следующие задания сложнее.

- 3** Составьте список школ своего города (или области), имеющих в Интернете свой сайт.
- 4** Сформируйте нужный запрос и полученные результаты сохраните в виде текстового документа.
- 5** Составьте запросы к поисковой системе, которые, на ваш взгляд, могли бы помочь получить ответы на следующие вопросы:
- а) Кто является создателем поисковой системы Yandex?
 - б) Что означает слово «блоггер» и как оно образовалось?
 - в) Кто и когда изобрёл шариковую авторучку?
 - г) Что изображено на гербе Московского государственного университета им. М. В. Ломоносова?
 - д) Где и когда проводилась первая международная олимпиада по информатике? Участвовала ли в ней российская команда и если да, то с каким результатом?
 - е) Каковы имя и фамилия Малыша из книги Астрид Линдгрен «Малыш и Карлсон»?
- Используя какую-либо поисковую систему, попытайтесь получить ответы на эти вопросы. Возможно, что запросы придётся уточнять. У кого из ваших одноклассников получилось быстрее всех выполнить это задание?
- 6** Найдите фотографию, на которой изображён памятник огурцу.
- 7** Наша страна располагается и в Европе, и в Азии. В некоторых местах границы между Европой и Азией установлены обелиски, указывающие на это разделение. На рисунках КП. 10 и КП. 11 изображены два из них. Определите, где и когда были установлены эти обелиски. Сколько всего в нашей стране установлено знаков, обозначающих границу между Европой и Азией?



Рис. КП. 10



Рис. КП. 11

- 8** С помощью толкового словаря из Интернета ответьте на вопрос: «Что такое индексация?»

Лабораторная работа № 10

РАБОТА С ГЕОИНФОРМАЦИОННОЙ СИСТЕМОЙ

Откройте геоинформационную систему, в которой есть ваш населённый пункт или ваш регион.

- 1** Найдите местоположение вашей школы.
- 2** Определите, кто из ваших одноклассников живёт от школы дальше всех.
- 3** Определите, где располагается ближайшая к вашему дому железнодорожная станция и как до неё добраться.
- 4** Предположим, вам поручили организовать экскурсию в ближайший зоопарк (возможно, что он располагается не в вашем населённом пункте). Выясните, как туда добраться.

Как правило, если вы разыскиваете какое-либо учреждение (например, тот же зоопарк или кинотеатр), то в геоинформационной системе вы можете найти справочную информацию об этом учреждении, в частности его официальный сайт.

- 5** Попытайтесь узнать график работы зоопарка, стоимость билета, какие интересные мероприятия в нём планируют провести в ближайшее время и т. п.
- 6** Выберите ещё какое-либо интересное для вас учреждение (музей, кинотеатр и т. п.) и выясните его местоположение, как до него добраться и его график работы.

Лабораторная работа № 11

СОЗДАНИЕ ЭЛЕКТРОННЫХ ПРЕЗЕНТАЦИЙ

Если на вашем компьютере установлено приложение Microsoft Office, то основным средством создания презентаций может стать PowerPoint. Если же у вас установлен Open Office, то вы можете пользоваться программой Impress.

- 1** Запустите программу создания электронных презентаций и с помощью справочной системы ознакомьтесь с её возможностями. Возможно, вы сочтёте полезным наиболее часто используемые приёмы выписать в виде отдельного текстового документа и распечатать его.

- 2** Выполнив задание 4 из § 17, вы разработали план создания презентации. Реализуйте его с помощью программы создания электронных презентаций, имеющейся в вашем распоряжении.
- 3** Выполнив задание 5 из § 17, вы разработали план создания презентации-рекламы. Реализуйте его с помощью программы создания электронных презентаций.
- 4** Проведите конкурс среди созданных в вашем классе реклам. Для этого разработайте систему критериев оценки. Критериями могут быть полнота и достоверность предоставляемой информации, эстетические стороны представления информации, разумное использование эффектов и т. п.

Лабораторная работа № 12

ПРЕЗЕНТАЦИОННЫЕ ПРОЕКТЫ

Проект № 1. «Живая азбука»

К разработке этого проекта вы приступили, выполняя задание 6 из § 17. Завершите его, создав при помощи компьютерных средств соответствующую презентацию.

Проект № 2. «Памятные места моего родного края»

Место, где живёт каждый из нас, часто называют малой родиной. Она действительно мала по сравнению со всей Россией. Но именно малая родина — это то, что вспоминаем мы, находясь вдали от неё.

Каковы памятные места малой родины? Это могут быть какие-то памятники или архитектурные строения, парки, созданные человеком, или, наоборот, какие-то природные образования... Разработайте проект, представляющий наиболее интересно те места, где вы живёте. Оформите результаты вашей работы в виде презентации.

Проект № 3. «История вычислительной техники»

Компьютер настолько вошёл в нашу повседневную жизнь, что уже трудно представить, как раньше мы жили без него. А ведь появился компьютер по историческим меркам совсем недавно и предназначался в первую очередь для проведения трудоёмких вычислений. Чему на смену пришёл компьютер? Какова история развития техники, облегчающей человеку работу с числами? Разработайте соответствующий проект и оформите результаты вашей работы в виде презентации.

Проект № 4. ...

Нам вовсе не хочется навязывать вам темы проектов. У вас может возникнуть желание самим предложить какой-либо проект. Обсудите его тему с учителем — он поможет точнее сформулировать задачи проекта. Создайте команду разработчиков, и — за работу!

Желаем успехов!

1. Гейн А. Г. Информатика и информационные технологии: задачник-практикум: учеб. пособие для учащихся 8—9 кл. общеобразоват. учреждений / А. Г. Гейн, Н. А. Юнерман. — М.: Просвещение, 2008. — 127 с.

2. Гейн А. Г. Информатика и информационные технологии. Тематические тесты. 8 класс / А. Г. Гейн, Н. А. Юнерман. — М.: Просвещение, 2009. — 96 с.

3. Гейн А. Г. Информатика и информационные технологии. Тематические тесты. 9 класс / А. Г. Гейн, Н. А. Юнерман. — М.: Просвещение, 2009. — 144 с.

4. Звонкин А. К. Информатика: алгоритмика: учеб. для 6 кл. общеобразоват. учреждений / А. К. Звонкин, С. К. Ландо, А. Л. Семёнов. — М.: Просвещение, 2006. — 239 с.

5. Кирюхин В. М. Информатика. Всероссийские олимпиады. Выпуск 1 / В. М. Кирюхин. — М.: Просвещение, 2008. — 220 с.

6. Кирюхин В. М. Информатика. Всероссийские олимпиады. Выпуск 2 / В. М. Кирюхин. — М.: Просвещение, 2009. — 222 с.

7. Кирюхин В. М. Информатика. Всероссийские олимпиады. Выпуск 3 / В. М. Кирюхин. — М.: Просвещение, 2011. — 222 с.

8. Козлова Т. В. Как найти любую информацию в Internet / Т. В. Козлова. — М.: ООО «НТ ПРЕСС», 2007. — 144 с.

9. Колисниченко Д. Н. Linux от новичка к профессионалу / Д. Н. Колисниченко. — СПб.: БХВ-Петербург, 2010. — 764 с.; [Электронный ресурс] <http://free-docs.ru/os/linux/248-linux-ot-novichka-k-professionalu.html>

10. Ландо С. К. Информатика: алгоритмика: учеб. для 7 кл. общеобразоват. учреждений / С. К. Ландо, А. Л. Семёнов, М. Н. Вялый. — М.: Просвещение, 2008. — 208 с.

11. Соловьёва Л. Ф. Информатика и ИКТ. Работаем в Windows и Linux. Практикум для 8 класса / Л. Ф. Соловьёва. — СПб.: БХВ-Петербург, 2010. — 64 с.

12. Энциклопедия школьной информатики / под ред. И. Г. Семкина. — М.: Бином, 2011. — 400 с.

Ключи к тестовым заданиям

Глава 1

Задание	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
Ответ	2	4	4	3	3	2	2	2	4	2

Задание	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
Ответ	3	4	3	2	1	2	1	3	2	3

Задание	B1	B2	B3	B4	B5
Ответ	2121211	233	144	-6	присваивания

Глава 2

Задание	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
Ответ	3	2	3	4	4	2	1	3	3	4

Задание	B1	B2	B3
Ответ	ЕГБДАЖВ	ВГАВ	25

А

- Актуальность информации 73
- Алгоритм 6
 - вспомогательный 26
 - Евклида 46
- Аргументы вспомогательного алгоритма 28
- Атрибут базы данных 94
 - — — ключевой 94
 - — — обязательно присутствующий 94

Б

- База данных 94
 - иерархическая 95
 - реляционная 95
 - сетевая 95
- Блог 113
- Браузер 105

В

- Веб-сайт 102
- Веб-страница 102
- Всемирная информационная паутина 101
- Высказывание 7

Д

- Данные 93
 - исходные 28
- Домен 103
- Доменная система имён 103
- Достоверность информации 73

Ж

- «Живой журнал» 112

З

- Заголовок ветвления 7
 - вспомогательного алгоритма 29
 - подпрограммы 29
 - цикла 7
- Запрос информационный 87
- Значение переменной 21

И

- Идентификатор пользователя 110
- Избыточность информации 74
- Имя массива 49
 - переменной 20
- Индекс элемента массива 49
- Интернет 101
- Информационная грамотность 83
 - культура 76
 - потребность 80
 - революция 76
- Информационное общество 75
- Информация релевантная 90
- ИПС 93
- Исполнитель 5
 - формальный 6

К

- Команда 6
 - вызова вспомогательного алгоритма 27
- Комментарий 6

Л

Логотип 126

М

Массив 49

- двумерный 49
- линейный 49
- одномерный 49

Н

Наибольший общий делитель 46

Новизна информации 73

О

Объективность информации 73

Объявление переменной 33

Оператор 6

- присваивания 21
- цикла 6
- — со счётчиком 14

Операторные скобки 6

Отрицание высказывания 7

П

Параметр фактический 34

- формальный 34

Переменная 20

- целого типа 33
- цветного типа 39
- вещественного типа 33

Подпрограмма 30

Подпрограмма-функция 45

Поиск адресный 88

- семантический 88
- фактографический 89

Полнота информации 73

- поиска 90

Принцип информационной открытости 75

Провайдер 103

Процедура персонификации 119

- регистрации 119

Р

Размерность массива 49

Результаты вспомогательного алгоритма 28

Релевантность информации 90

С

Сеть социальная 112

Система геоинформационная 120

- информационная 116
- — распределённая 118
- информационно-поисковая 93
- команд исполнителя 6
- управления базами данных 93

Слайд 126

- титульный 126

Сортировка данных 97

СУБД 93

Схема цикла со счётчиком 14

Т

Таблица истинности 8

Твиттер 113

Тело вспомогательного алгоритма 29

Тип массива 49

Точность поиска 90

У

Универсальный указатель ресурса 105

Условие 7

— сложное 7

Ф

Функция Эйлера 48

Ц

Ценность информации 73

Цикл 6

— со счётчиком 15

Ч

Числа взаимно простые 48

Число простое 47

Э

Электронная почта 101, 109

Элемент массива 49

F

ftp-сервис 110

I

ICQ 112

L

login пользователя 110

S

Skype 112

SMS 114

U

URL 105

W

WAP-сайт 114

World Wide Web 101

WWW 101

Предисловие.....	3
Глава 1. Переменные в алгоритмах и структуры данных	5
§ 1. Алгоритм и алгоритмические конструкции. Сложные условия	—
§ 2. Ещё раз о циклах. Оператор цикла со счётчиком.....	13
§ 3. Переменные в алгоритмах	20
§ 4. Вспомогательный алгоритм	26
§ 5. Типы переменных	32
§ 6. Подпрограммы в работе Паркетчика	39
§ 7. Подпрограмма-функция (для дополнительного чтения).....	45
§ 8. Массивы	48
§ 9. Знакомство с языками программирования	53
Проверь себя	61
Глава 2. Человек и информация	72
§ 10. Информация и её свойства	—
§ 11. Информационная грамотность — базовый элемент культуры человека в современном обществе	80
§ 12. Поиск информации	86
§ 13. Базы данных	92
§ 14. Интернет: сеть глобальных коммуникаций и распределённая база данных	101
§ 15. Что ещё можно делать в Интернете	109
§ 16. Информационные системы	116
§ 17. Электронные презентации	125
Проверь себя	129
Компьютерный практикум	134
Лабораторная работа № 1. В компьютерном классе после летних каникул	—
Лабораторная работа № 2. Сложные условия в программах для Паркетчика	135
Лабораторная работа № 3. Циклы со счётчиком	136

Лабораторная работа № 4. Переменные в жизни Паркетчика ...	138
Лабораторная работа № 5. Подпрограммы для Паркетчика	—
Лабораторная работа № 6. Коллективная разработка программы	139
Лабораторная работа № 7. Проекты для Паркетчика	140
Лабораторная работа № 8. Знакомство с языком программирования	148
Лабораторная работа № 9. Поиск информации в Интернете	—
Лабораторная работа № 10. Работа с геоинформационной системой	151
Лабораторная работа № 11. Создание электронных презентаций	—
Лабораторная работа № 12. Презентационные проекты	152
Дополнительная литература	153
Ключи к тестовым заданиям	154
Предметный указатель	155

Учебное издание

**Гейн Александр Георгиевич
Юерман Нина Ароновна
Гейн Андрей Александрович**

ИНФОРМАТИКА

8 класс

Учебник для общеобразовательных учреждений

Зав. редакцией Т. А. Бурмистрова

Редактор О. В. Платонова

Художественный редактор О. П. Богомолова

Художник А. В. Василькова

Фотографии «РИА-Новости»

Компьютерная графика: М. Е. Савельевой, А. Г. Вьюниковской

Технический редактор Н. Н. Репьева

Корректоры Ф. А. Шагиданова, А. К. Райхчин

Налоговая льгота — Общероссийский классификатор продукции ОК 005-93—953000. Изд. лиц. Серия ИД № 05824 от 12.09.01. Подписано в печать с оригинал-макета 04.02.13. Формат 70×90¹/₁₆. Бумага офсетная. Гарнитура Школьная. Печать офсетная. Уч.-изд. л. 8,85 + 0,52 форз. Тираж 3000 экз. Заказ № 34064 (89-13).

Открытое акционерное общество «Издательство «Просвещение».
127521, Москва, 3-й проезд Марьиной рощи, 41.

Отпечатано в филиале «Смоленский полиграфический комбинат»
ОАО «Издательство «Высшая школа». 214020, Смоленск, ул. Смольянинова, 1
Тел. +7 (4812) 31-11-96. Факс: +7 (4812) 31-31-70
E-mail: spk@smolpk.ru <http://www.smolpk.ru>

На учебник получены положительные заключения Российской академии наук (№ 10106-5215/427 от 14.10.11) и Российской академии образования (№ 01-5/7д-351 от 17.10.11).

Гейн А. Г.

Г29 Информатика. 8 класс : учеб. для общеобразоват. учреждений / А. Г. Гейн, Н. А. Юнерман, А. А. Гейн. — М. : Просвещение, 2013. — 159 с. : ил. — ISBN 978-5-09-019992-6.

В учебнике, кроме объяснительного текста, содержится большое количество вопросов и заданий, в том числе выполняемых при помощи компьютера. Приводятся темы для обсуждения в группах, а также темы возможных проектов. Для закрепления пройденного материала в конце каждой главы помещены тематические тесты в форме ЕГЭ. Компьютерный практикум вынесен в отдельный раздел учебника, что позволяет учителю более гибко планировать учебное время. На сайтах <http://kadm.math.usu.ru> и <http://prosv.ru> можно найти некоторые учебные программы («Паркетчик», «Машина Тьюринга», «Машина Поста» и др.), которые будут полезны при изучении информатики.

УДК 373.167.1:004
ББК 32.81я72

ISBN 978-5-09-019992-6

© Издательство «Просвещение», 2013
© Художественное оформление.
Издательство «Просвещение», 2013
Все права защищены

Язык записи алгоритмов

Pascal

1. Заголовок программы

Алгоритм {...}

program <имя>;

2. Описание переменных

var

(*Обязательно перед описанием переменных*)

вещественные переменные

вещ: <список переменных>;

<список переменных>: **real**;

целочисленные переменные

цел: <список переменных>;

<список переменных>: **integer**;

символьные переменные

сим: <список переменных>;

<список переменных>: **string**;

массивы

вещ: <имя массива> [1:100; 1:100];

<имя массива>: **array** [1..100, 1..100] **of** **real**;

3. Оператор условия

Если <условие> **то**

{ <оператор>;

...

}

иначе

{<оператор>;

...

}

if <условие> **then**

begin

<оператор>; ...

end

else

begin

<оператор>; ...

end;

Конструкция **иначе (else)** может отсутствовать

4. Оператор цикла по условию

Делать пока <условие>

{ <оператор>;

...

}

while <условие> **do**

begin

<оператор>; ...

end;

5. Оператор цикла со счётчиком

Делать от <переменная> := <выражение>
до <выражение> **с шагом** <выражение>

{ <оператор>; ... }

for <переменная> := <выражение>
to <выражение> **do**
 begin
 <оператор>; ...
 end;

Если нужен шаг -1, то вместо **to** пишут **downto**

6. Операторы ввода и вывода информации

Запросить <имя переменной>;

Сообщить <имя переменной или/и
сообщение в кавычках>;

readln (<список переменных>;

writeln ('<сообщение>', <список
переменных>;

7. Подпрограммы и функции

Подпрограмма <имя подпрограммы>;
(**arg**: <список параметров>,
рез: <список параметров>)
{ <оператор>; ... }

Функция <имя>: (**arg**: <типы аргументов>:
<список переменных>): <тип>
{ <оператор>;
 ...
 знач := <результат>;
}

Procedure <имя> (<параметры>: <тип>;
var <результаты>: <тип>);
begin
 <оператор>; ...
end;

function <имя> (<параметры>: <тип>):
<тип>;
 begin
 <оператор>; ...
 <имя функции> := <результат>;
 end;

8. Обращение к подпрограмме

Вызвать <имя подпрограммы>
(**arg**: <список параметров>;
рез: <список параметров>);

<имя> (<список параметров>);

9. Обращение к функции

<переменная> := <имя функции>
(<список фактических параметров>);

<переменная> := <имя функции>
(<список фактических параметров>);

1. Заголовок программы

Алгоритм

{...}

алг <имя>нач

...

кон

2. Описание переменных

вещественные переменные

вещ: <список переменных>;вещ: <список переменных>

целочисленные переменные

цел: <список переменных>;цел: <список переменных>

символьные переменные

сим: <список переменных>;сим: <список переменных>

массивы

вещ: <имя массива> [1:100; 1:100];вещ таб <имя массива> [1:100, 1:100]

3. Оператор условия

Если (<условие>) то

{ <оператор>;

... }

иначе

{ <оператор>;

... }

Если <условие>то

<оператор> ...

иначе

<оператор> ...

всеКонструкция иначе может отсутствовать

4. Оператор цикла по условию

Делать пока (<условие>)

{ <оператор>;

... }

нцпока <условие>

<оператор>

...

кц

5. Оператор цикла со счётчиком

Делать от <переменная> := <выражение>
до <переменная> **с шагом** <выражение>
 { <оператор>;
 ...
 }

нц для <переменная> **от** <выражение>
до <выражение>
 <оператор>
 ...
кц

6. Операторы ввода и вывода информации

Запросить <имя переменной>;

Ввод <список переменных через запятую >

Сообщить <имя переменной или/и
 сообщение в кавычках>;

Вывод <список переменных или/и
 сообщение в кавычках>

7. Подпрограммы и функции

Подпрограмма <имя подпрограммы>;
 (**арг**: <список параметров>,
рез: <список параметров>)
 { <оператор>; ... }

алг <имя> (**арг**: <параметры>,
рез: <параметры>)
нач <описание переменных>
 <оператор> ...
кон

Функция <имя>: (**арг**: <типы аргументов>:
 <список переменных>): <тип>
 { <действие>;
 ...
знач := <результат>;
 }

алг <тип> <имя> (**арг**: <тип> <параметры>)
нач <описание переменных>
 <оператор> ...
знач := <результат>
кон

8. Обращение к подпрограмме

Вызвать <имя подпрограммы>
 (**арг**: <список параметров>;
рез: <список параметров>);

<имя алгоритма> (<список параметров и
 результатов>)

9. Обращение к функции

<переменная> := <имя функции>
 (<список фактических параметров>);

<переменная> := <имя функции> (<список
 параметров>);

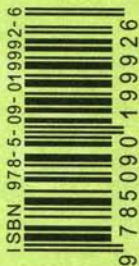


ИНФОРМАТИКА

Учебно-методический комплект авторского коллектива под руководством А. Г. Гейна:

- А. Г. Гейн, Н. А. Юнорман, А. А. Гейн.
Информатика. 8 класс
- А. Г. Гейн, А. А. Гейн.
Информатика. Рабочая тетрадь. 8 класс
- А. Г. Гейн, Н. А. Юнорман.
Информатика и информационные технологии.
Задачник-практикум. 8—9 классы
- А. Г. Гейн.
Информатика. Методические рекомендации.
8 класс
- А. Г. Гейн.
Информатика. Рабочие программы.
7—9 классы

Программное обеспечение для проведения практических занятий и методические рекомендации размещены на сайтах:
www.prosv.ru
<http://kadm.math.usu.ru> (на страничке А.Г. Гейна)



ПРОСВЕЩЕНИЕ
ИЗДАТЕЛЬСТВО