Министерство образования и науки Российской Федерации

Государственное образовательное учреждение

высшего профессионального образования
«Тихоокеанский государственный университет»

Т.А. Жданова
Ю.С. Бузыкова

ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

Утверждено издательско-библиотечным советом университета
в качестве учебного пособия

Хабаровск
Издательство ТОГУ
2011

УДК 004.43(075.8)
ББК З 973.2-018я7

Рецензенты:

кафедра информационные технологии ХИИК СибГУТИ (зав. кафедрой, к. п. н. Е. В. Ковнерова)

канд. техн. наук, доцент А. Н. Вишневский

Научный редактор: к. ф.-м. н. В. В. Стригунов

Ж 422 **Жданова Т.А.**

Основы алгоритмизации и программирования: учеб. пособие / Т.А. Жданова, Ю.С. Бузыкова. – Хабаровск : Изд-во Тихоокеан. гос.ун-та, 2011. – 56 с.

ISBN

Учебное пособие является частью учебно-методического комплекса по информатике и знакомит с основами алгоритмизации и программированию на Qbasic студентов-иностранцев, обучающихся на всех специальностях университета.

Содержание пособия соответствует государственному образовательному стандарту по информатике, а также требованиям, предъявляемым на экзамене по информатике по разделам «алгоритмизация» и «программирование».

Цель создания пособия продиктована отсутствием адаптированных курсов по информатике для иностранцев. Учебное пособие является учебным средством, которое предусматривает самостоятельную работу с текстом для проработки теоретического и практического материала.

Авторы предполагают вариативность использования данного пособия в учебном процессе для всех форм обучения студентов-иностранцев информатике.

Собранный в учебное пособие материал неоднократно использовался на учебных занятиях.

УДК 004.43(075.8)
ББК З 973.2-018я7

SBN

Учебное издание

ЖДАНОВА Татьяна Аркадьевна
БУЗЫКОВА Юлия Сергеевна

ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

Учебное пособие

Главный редактор *Л. А. Суевалова*
Редактор *Н.Г. Петряева*
Дизайнер обложки
Операторы компьютерной верстки *Т. А. Жданова, Ю. С. Бузыкова*

Подписано в печать 00.00.20011. Формат 60 х 84 $^{1}/_{16.}$
Бумага писчая. Гарнитура «Таймс». Печать цифровая.
Усл. печ. л. 3,35. Тираж 150 экз. Заказ

Издательство Тихоокеанского государственного университета.
680035, Хабаровск, ул. Тихоокеанская, 136.

Отдел оперативной полиграфии издательства
Тихоокеанского государственного университета.
680035, Хабаровск, ул. Тихоокеанская, 136

ВВЕДЕНИЕ

Основы алгоритмизации и программирование являются фундаментальными основами теоретической информатики, но по учебному плану дисциплины имеют весьма ограниченное время для изучения. Поэтому некоторые важные вопросы не рассматриваются на лекциях вообще, или рассматриваются недостаточно глубоко (например, решение задач на составление алгоритмов, основы программирования, запись выражений на языке программирования и др.). Предлагаемое пособие является учебным средством, предусматривающим ориентацию иностранных студентов на самостоятельную работу и самообразование.

Пособие состоит из шести глав.

В первой главе даются основные понятия алгоритмизации, способы записи алгоритмов, составление блок-схем и базовые управляющие конструкции алгоритмов. Во второй главе излагаются основные конструкции языка программирования Qbasic: алфавит, константы, переменные, функции, логические и арифметические выражения. В третьей – операторы языка QBasic для реализации линейных алгоритмов. В четвертой – операторы передачи управления для реализации разветвляющихся алгоритмов. В пятой – операторы циклов для реализации циклических алгоритмов. В шестой главе рассматриваются одномерные массивы.

В конце каждой главы даются контрольные вопросы для самоконтроля, упражнения для закрепления, контрольная работа и тест. Подобные задания включены в экзаменационные билеты. Для каждой главы составлен терминологический словарь, он размещен в приложении.

Язык учебного пособия адаптирован, так как оно предназначается для лиц, слабо владеющих сложными грамматическими конструкциями. В этом важное отличие этого пособия от других учебников по информатике. Теория, упражнения и тестовые задания составлены с учетом обеспечения формирования навыков речи и чтения специальной литературы по курсу «информатика».

ГЛАВА 1. ОСНОВЫ АЛГОРИТМИЗАЦИИ

1.1. Понятие алгоритма

Алгоритмом называется строго определенная последовательность действий, определяющих процесс перехода от исходных данных к искомому результату.

Примером алгоритма может служить алгоритм «Переправа».

На левом берегу реки находятся два молодых человека со своими девушками. Всем нужно перебраться на правый берег, но в лодке только два места. Каждая девушка не хочет оставаться на берегу без своего молодого человека, если на этом же берегу находится другой молодой человек. Как всем переплыть на другой берег? (рис. 1.1)

Р е ш е н и е : Обозначим девушек и молодых людей Д1, Д2, М1, М2, переезд на правый берег $\rightarrow$, переезд на левый берег $\leftarrow$. Можно записать алгоритм:

1) Д1, Д2 $\rightarrow$
2) Д1 $\leftarrow$
3) М1, М2 $\rightarrow$
4) М1 $\leftarrow$;
5) Д1, М1 $\rightarrow$

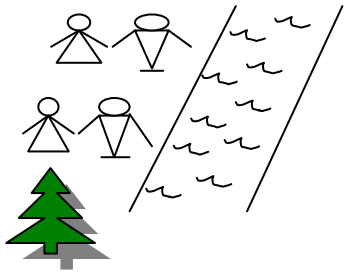


Рис. 1.1

Порядок действий считается алгоритмом в том случае, если он обладает определенными свойствами.

1.2. Свойства алгоритма

- *Дискретность*. Алгоритм должен представлять процесс решения задачи как последовательность выполнения простых действий (шагов, этапов). При этом для выполнения каждого действия алгоритма требуется время.

- *Детерминированность (Однозначность).* Каждое действие (шаг, этап) должно быть четким, однозначным, исключающим произвольное толкование и не оставляющим места для двусмысленности. Выполнение алгоритма носит, по сути, механический характер и не требует никаких дополнительных указаний.
- *Результативность***.** Алгоритм должен приводить к решению задачи или сообщению, что задача решений не имеет за конечное число шагов.
- *Конечность***.** Каждое отдельное действие, как и весь алгоритм должны иметь возможность реального исполнения. Поэтому алгоритм имеет придел, т. е. конечен.
- *Массовость***.** Алгоритм разрабатывается в общем виде так, чтобы его можно было применять для класса задач, различающихся только исходными данными. При этом исходные данные выбираются из некоторой области, которая называется областью применяемости алгоритма. Например, для решения квадратного уравнения $ax^2+bx+c=0$, коэффициенты действительные числа, $a \neq 0$, и а, в, с – различные.

1.3. Способы записи алгоритмов

Существуют разные способы записи алгоритмов – *словесно-формульный, графический, операторный* (программа на алгоритмическом языке).

а) *Словесно-формульный способ*. Например, требуется решить квадратное уравнение $ax^2+bx+c=0$ в области действительных чисел. Математической моделью этой задачи является известная формула корней квадратного уравнения:

$$y_{1,2}= \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

На основании этой формулы запишем алгоритм:

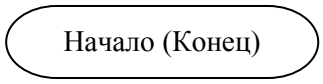
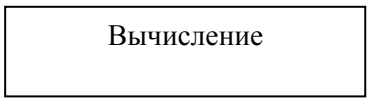
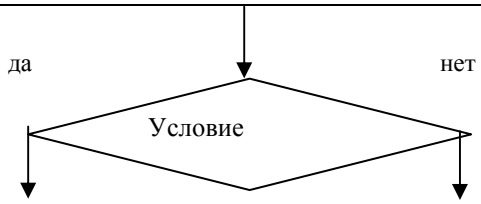
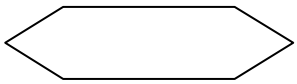
1. Задать значения a, b, c.
2. Вычислить дискриминант $d = b^2 – 4ac$.

3. Сравнить дискриминант с нулем, если он больше нуля, то вычислить корни по формуле $y_{1,2}= \frac{-b \pm \sqrt{d}}{2a}$ и перейти к п. 4, иначе сообщить: «В области действительных чисел уравнение решений не имеет».
4. Записать результат: «Корни уравнения y_1 и y_2».

б*) Графический способ* описания алгоритма иначе называют блок - схемой. В блок-схемах используются геометрические фигуры, каждая из которых изображает какую-либо операцию или действие, а также этап процесса решения задачи. Каждая фигура называется блоком. Порядок выполнения этапов показывается стрелками, соединяющими блоки. Блоки необходимо размещать сверху вниз или слева направо в порядке их выполнения.

В табл. 1.1 приведены наиболее часто употребляемые блоки.

Таблица 1.1

Обозначение блока	Выполняемая функция
	Начало или Конец алгоритма
	Вычисляемые действия
	Проверка условия: выбор одного из двух направлений
	Ввод или Вывод данных
	Организация циклических процессов
	Направление линий потока – стрелки

Правила построения алгоритмов на языке блок-схем

1. Блок-схема строится сверху вниз.
2. В любой блок-схеме имеется один элемент, соответствующий началу, и один элемент, соответствующий концу.
3. Должен быть хотя бы один путь из начала блок-схемы к любому элементу.
4. Должен быть хотя бы один путь от каждого элемента блок-схемы в конец блок-схемы.

в) Операторный способ (алгоритмический язык). Алгоритм – это задание для исполнителя. Исполнитель выполняет алгоритм, т. е. делает то, что написано в алгоритме. Если исполнитель точно выполнит то, что написано в алгоритме, то он получит результат.

Человек, автоматическое устройство, компьютер – это разные исполнители алгоритмов. Для того чтобы компьютер мог выполнить алгоритм, его надо написать на понятном компьютеру языке. Компьютер понимает машинный язык. Например, равенство x = y на машинном языке имеет вид: 111101110011110111110101.

Понятно, что человеку трудно писать и читать алгоритмы на машинном языке. Человек легко может писать и читать на естественном языке. Но нельзя научить компьютер понимать естественный язык потому, что в естественном языке много слов и нет строгих правил записи предложений.

Для того чтобы человек и компьютер понимали друг друга, разработаны специальные языки для записей алгоритмов – алгоритмические языки. Самые известные алгоритмические языки – это Бейсик (Basic), Паскаль (Pascal), Фортран (Fortran).

Алгоритмический язык отличается от машинного языка тем, что состоит из слов и символов, как естественный язык. Алгоритмический язык отличается от естественного языка тем, что в нем мало основных слов (обычно 30-40) и очень строгие правила составления предложений. Основные слова алгоритмического языка называют служебными словами. В алгоритмических языках используют слова английского алфавита. Алгоритмический язык легко понимает и человек и компьютер.

Алгоритм, который записан на алгоритмическом языке, – это программа для компьютера. Каждое предложение в программе – это оператор.

Например, можно написать программу решения квадратного уравнения $ax^2+bx+c=0$ на компьютере. На алгоритмическом языке Бейсик эта программа будет выглядеть так:

```
REM РЕШЕНИЕ КВАДРАТНОГО УРАВНЕНИЯ
INPUT “введите a,b,c“; A,B,C
D=B-2 - 4*A*C
IF D<0 THEN “РЕШЕНИЙ НЕТ”
Y1=(-B+SQR(D))/(2*A):Y2=(-B-SQR(D))/(2*A)
PRINT “КОРНИ УРАВНЕНИЯ”;Y1,Y2
```

1.4. Типы алгоритмов

Алгоритмы бывают *линейные*, *разветвляющиеся* и *циклические*.

Линейный алгоритм – это алгоритм, в котором действия выполняются только один раз и строго в том порядке, в котором они записаны.

Линейные алгоритмы в математике – это, например, вычисление площадей фигур.

П р и м е р . Составить алгоритм вычисления площади трапеции с основаниями a,b и высотой h.

Р е ш е н и е . Известно, что площадь трапеции вычисляется по формуле $S=\frac{a+b}{2}h$. Поэтому можно записать алгоритм:

1. Задать численное значение a, b, h.
2. Вычислить выражение $S=\frac{a+b}{2}h$.
3. Записать ответ S. (рис. 1.2)

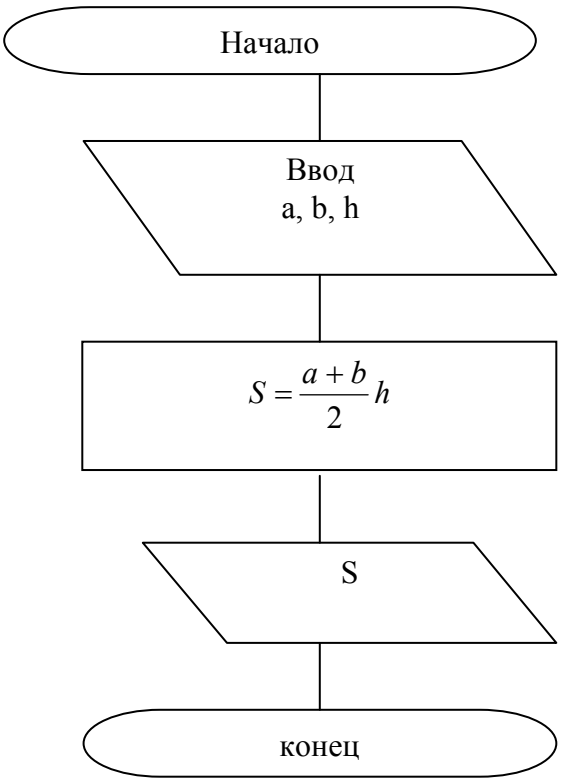


Рис. 1.2

Разветвляющийся алгоритм – это алгоритм, в котором то или иное действие выполняется после анализа условия. Процесс анализа условия и выбора одной из ветвей на блок-схеме показывают с помощью логического блока. Пример разветвляющейся структуры показан на рис. 1.3.

Процесс анализа условия и выбора одной из ветвей на блок-схеме показывают с помощью логического блока. Логический блок имеет один вход и два выхода (ветвь «да» и ветвь «нет»).

В блок-схемах разветвляющихся алгоритмов всегда есть логический блок.

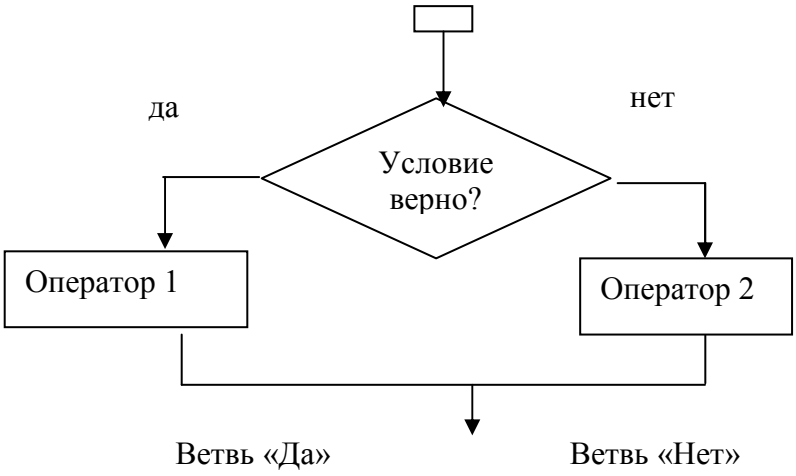


Рис. 1.3

На рис. 1.4 приведен пример блок-схемы решения квадратного уравнения $a x^2 + b x + c = 0$.

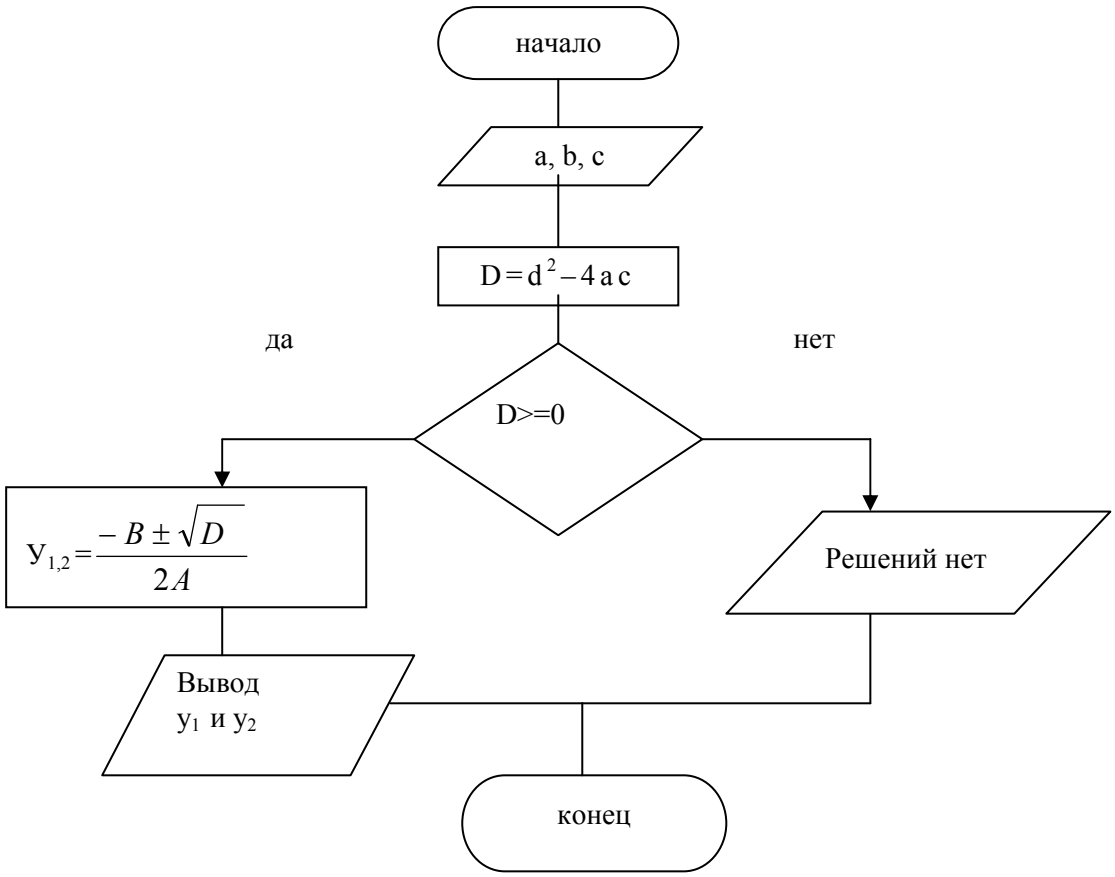


Рис. 1.4

Циклический алгоритм (цикл)- это алгоритм, в котором группа операторов выполняется несколько раз подряд. Блок-схема цикла обязательно содержит логический блок, ее структура показана на рис 1.5.

Выполняется циклический алгоритм так: сначала проверяется условие, если условие верно (истина), то выполняется тело цикла (действия или группа операторов) и, далее, изменяются значения параметра цикла и снова проверяется условие и т. д. На каком-то шаге условие не выполнится (ложь) и тогда происходит выход из цикла и продолжается выполнение программы.

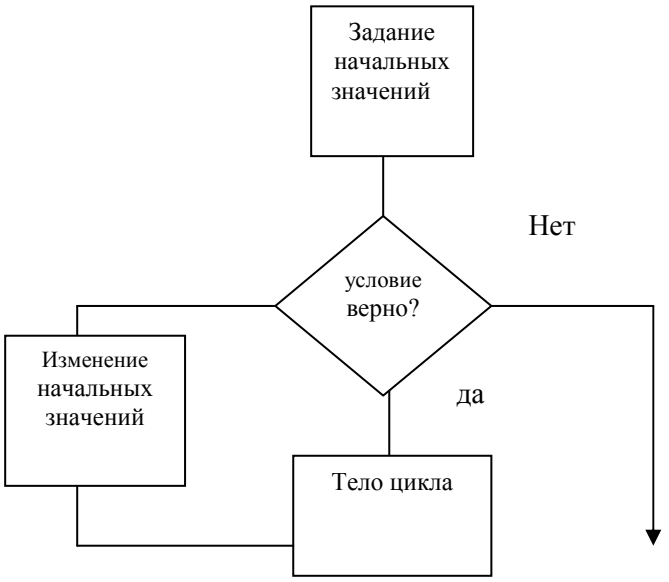


Рис. 1.5

Приведем примеры построения блок-схем алгоритма циклической структуры:

Вычислить сумму N чисел, последовательно вводимых с клавиатур.

На рис. 1.6 для реализации циклического процесса использованы комбинации блоков присваивания и ветвления. На рис. 1.7 показан блок цикла, который реализует те же функции.

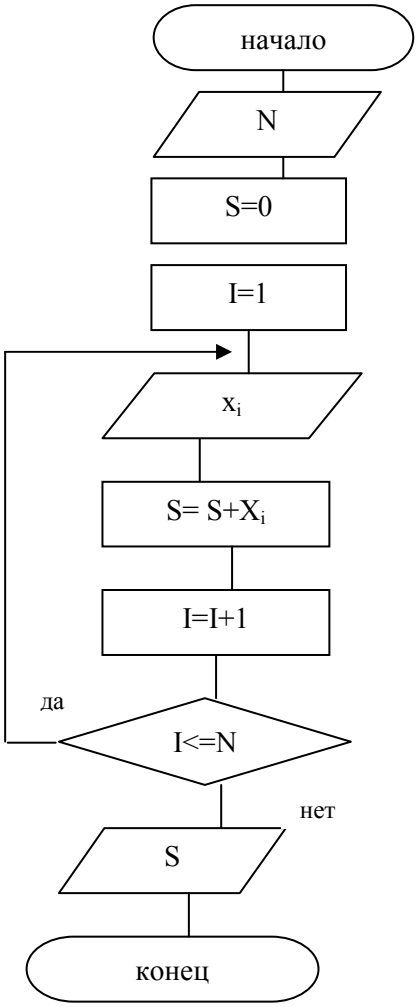


Рис. 1.6

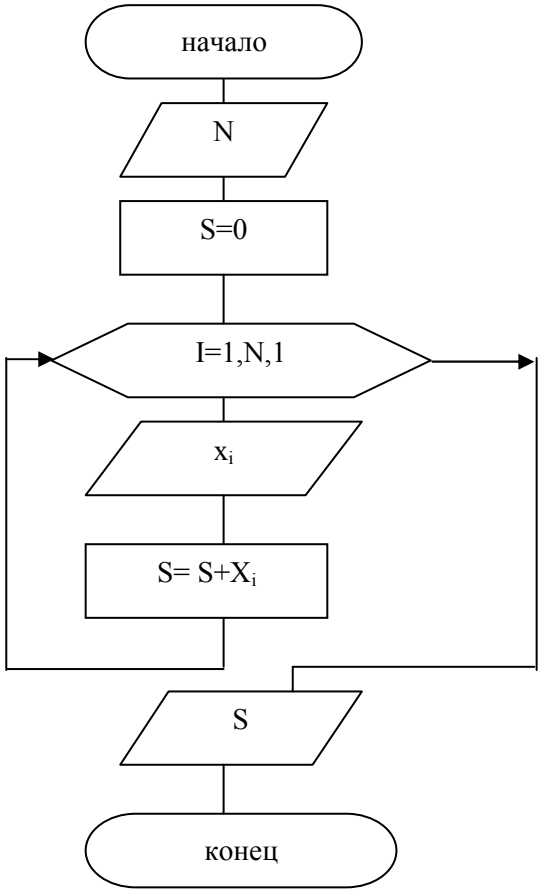


Рис. 1.7

ВОПРОСЫ И ЗАДАЧИ

1. Что такое алгоритм?
2. Какие способы записи алгоритма вы знаете? Приведите примеры.
3. Какие типы алгоритмов бывают? Подберите пример алгоритма для каждого типа.
4. Предложите алгоритм решения задачи «Переправа», если на левом берегу реки находятся три пары.
5. Есть 27 монет. Известно, что одна монета фальшивая (ее вес меньше). На чашечных весах (рис. 1.8) можно сравнивать вес монет (весы показывают, какие монеты весят больше, меньше, или вес одинаковый). Най-

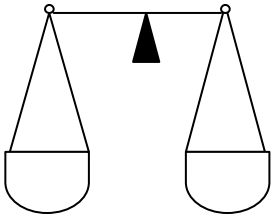


Рис. 1.8

ти фальшивую монету. Составить алгоритм решения этой задачи, если использовать весы можно только три раза.

6. Как изменится алгоритм решения задачи 5, если известно, больше или меньше весит фальшивая монета? Какое наименьшее число взвешиваний при этом необходимо?
7. Нарисовать блок-схему алгоритма вычисления выражения $4x^3 + 3x^2 + 2x + 1$ по заданному значению x.
8. Выражение $4x^3 + 3x^2 + 2x + 1$ можно записать в виде $x(x(4x + 3) + 2) + 1$. Нарисуйте блок-схему алгоритма.
9. В упражнениях 7 и 8 использованы разные алгоритмы вычисления тождественных выражений. Почему алгоритм из упражнения 8 более рациональный?
10. Нарисовать блок-схему алгоритма вычисления функции:
11. $Y = \begin{cases} x^2 + 4x + 5 \ \textit{при} \ x \le -1 \\ \dfrac{1}{x^2 + 4x + 5} \ \textit{при} \ x > 1 \end{cases}$.
12. Нарисовать блок-схему алгоритма вычисления функции:
13. $y = \begin{cases} \dfrac{1}{x^3} & \textit{при} \ x \le -2 \\ x^{\frac{3}{4}} & \textit{при} \ -2 < x < 0 \\ \sqrt{6x + 4} & \textit{при} \ x \ge 0 \end{cases}$.
14. Запишите алгоритм Евклида. Найти наибольший общий делитель (НОД) двух целых положительных чисел.

КОНТРОЛЬНЫЙ ТЕСТ ПО АЛГОРИТМИЗАЦИИ

1. Строго определенная последовательность действий, необходимых для решения поставленной задачи, – это …

a) метод решения;
b) алгоритм;
c) блок-схема.

2. Ниже перечислены основные свойства алгоритма. Некоторые из этих понятий *не* относятся к основным свойствам алгоритма. Укажите, какие именно.

a) дискретность;
b) определенность;
c) актуальность;
d) результативность;
e) массовость
f) строгость;
g) секретность.

3. Свойство, означающее, что решение задачи, записанное в виде алгоритма, разбито на отдельные простейшие команды, которые расположены в порядке их выполнения, – это…

a) дискретность;
b) определенность;
c) результативность.

4. Массовость алгоритма – это свойство заключается в том, что каждый алгоритм, разработанный для решения некоторой задачи, должен быть применен для решения задач данного типа при всех допустимых значениях исходных данных. Верно ли данное высказывание? Все ли способы здесь перечислены?

5. Существуют несколько способов записей алгоритмов:

a) Описание с помощью слов и формул;
b) Описание с помощью графических схем.

6. Графическое описание алгоритмов как последовательности действий называется … Вставить пропущенное словосочетание.

7. Команда алгоритма, в которой делается выбор: выполнять или не выполнять какую-либо группу команд, называется ….
Вставьте слово.

8. Приведены две блок-схемы некоторых алгоритмов (рис. 1.9, 1.10). Какая из схем ошибочна?

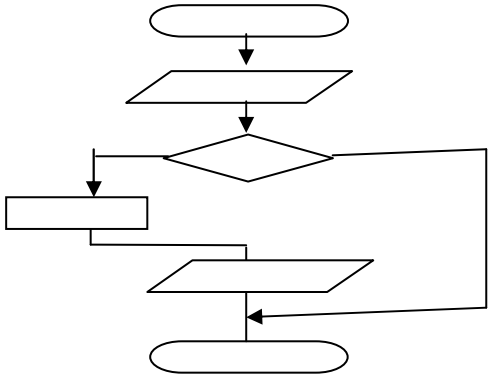


Рис. 1.9

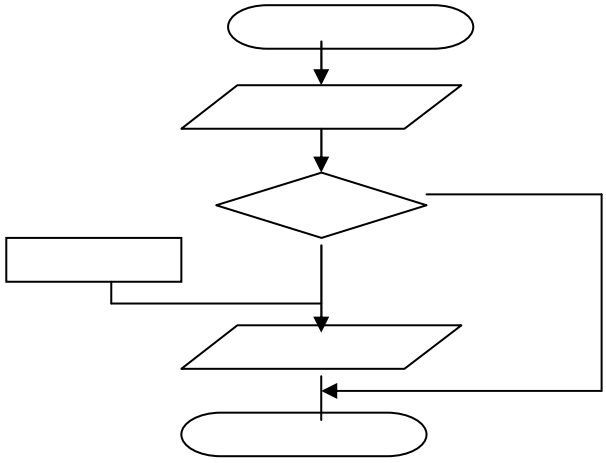


Рис. 1.10

9. В зависимости от особенностей своего построения алгоритмы делятся на несколько основных групп:

a) линейные;
b) разветвляющиеся;
c) структурные;
d) циклические.

Некоторые из этих понятий не относятся к основным группам алгоритмов. Укажите, какие именно.

10. «Линейным называется алгоритм, в котором все этапы выполняются строго последовательно». Верно ли данное высказывание?

11. Укажите правильный вариант ответа. Циклом называется:

a) Этап решения задачи, выполняемый строго последовательно;
b) Последовательность действий, выполняемых многократно, каждый раз при новых значениях параметров;
c) Выбор одного из нескольких возможных вариантов вычислительного процесса.

12. Программа, представленная блок-схемой, начинается с блока … .
Вставьте нужное слово.

13. Ниже приведены блок-схемы некоторых алгоритмов (рис. 1.11, 1.12). Укажите, какая из нижеприведенных блок-схем является блок-схемой линейной структуры?

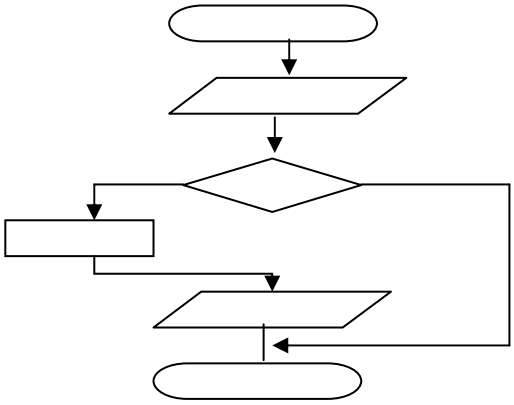


Рис. 1.11

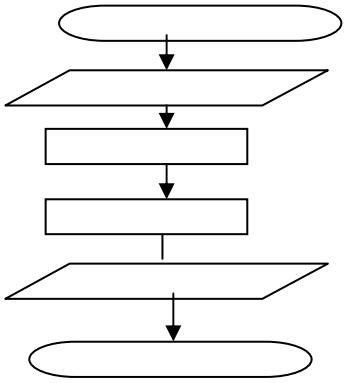


Рис. 1.12

14. Ниже приведены блок-схемы некоторых алгоритмов (рис. 1.13, 1.14). Укажите, какая из нижеприведенных блок-схем является блок-схемой циклической структуры?

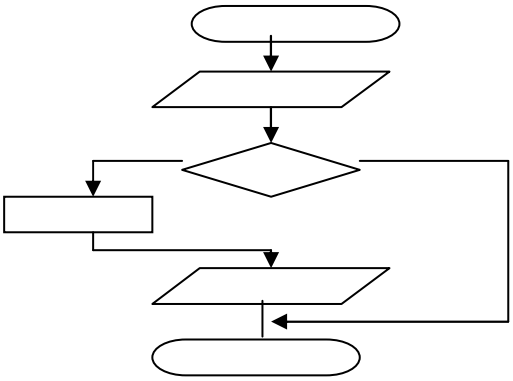


Рис. 1.13

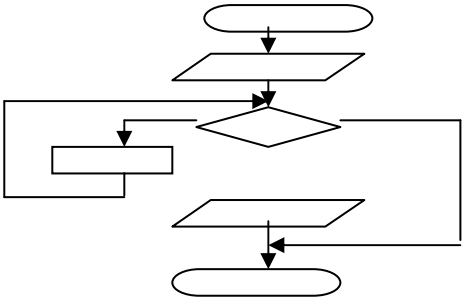


Рис. 1.14

Глава 2. ПРОСТЕЙШИЕ КОНСТРУКЦИИ ЯЗЫКА ПРОГРАММИРОВАНИЯ QBASIC

2.1. Алфавит языка

Язык программирования Qbasic (как и любой другой язык) образуют три его составляющие: алфавит, синтаксис и семантика.

Алфавит – это фиксированный для данного языка набор основных символов, т. е. «букв», из которых должен состоять любой текст на этом языке – никакие другие символы в тексте не допускаются.

Синтаксис – это правила построения фраз, позволяющие определить, правильно или неправильно написана та или иная фраза.

Семантика определяет смысловое значение предложений языка.

Основными понятиями в языках программирования обычно являются алфавит языка, константы, переменные, встроенные функции, логические и арифметические выражения.

Алфавит языка включает:

- буквы латинского алфавита от **A** до **Z** (строчные и прописные)
- арабские цифры: **0 1 2 … 9**
- знаки арифметических операций: **+ – * / ^ **
- специальные символы объявления типа: **% # ! $ &**
- круглые скобки **( )** кавычки **“ ”** апостроф **’** , подчеркивание **_**
- знаки отношений: **< > = <> <= >=**
- буквы русского алфавита используются только для записи текстовых констант и комментариев к программе.

2.2. Константы

Константы – это данные, не изменяющиеся в процессе решения задачи. В Qbasic используются числовые и текстовые константы.

Числовые константы записываются в программе в виде конкретного числа и бывают двух типов: целые и вещественные.

Целые константы – это последовательность цифр, перед которой может стоять знак + или – . Например: –567, +29, 29.

Вещественные константы – это числа, имеющие целую и дробную части. Бывают две формы записи вещественных констант: основная и экспоненциальная формы записи. В *основной форме* записи с фиксированной точкой целая часть числа отделяется от дробной части десятичной точкой, например, -1.012, 3.14159. В *экспоненциальной форме* записи с «плавающей» точкой число записывается в виде: $mE \pm p$, где m – мантисса (число в основной форме), E – основание 10, p – порядок числа (целая константа, содержащая не более двух цифр).

Например: 2.4E-05, -2.9E+07.

Чтобы перейти от экспоненциальной формы к основной, нужно m (мантиссу) умножить на 10 в степени p (порядок).

Например: 2.4E-5 = $2.4 \cdot 10^{-5}$ = 0.000024, -2,9E+07 = $2.9 \cdot 10^{7}$ = 29000000.

Строковые константы – это набор любых символов длиной, не превышающей 255 символов. Строковая константа заключается в двойные кавычки. Например: “ТАБЛИЦА”, “ALFA_2”,“ Решений нет”.

2.3. Переменные

Переменные – это величины, значения которых могут изменяться в процессе выполнения программы, обозначаются *именем* (*идентификатором*). *Имя переменной* представляется последовательностью не более чем 40 латинских букв и цифр, начинающейся обязательно с буквы и заканчивающейся суффиксом, определяющим тип переменной. Символы типов: **%** – целый; **!** – вещественный; **$** – строковый. Если символ типа отсутствует, то переменная по умолчанию считается вещественной. Например: переменные SUM, a, B12, Max, S!, P!, B! – вещественного типа; Z%, D%, A% – целого типа; F$, S$, G$ – строкового типа.

Различают *простые переменные* и *переменные с индексом*. Простая переменная определяется только именем. Переменная с индексом является элементом массива, определяется именем и индексами, которые задают

местоположение элемента в массиве. Индексы записываются в скобках через запятую после имени переменной. Например: X(10), A(5,6).

Массивом называется упорядоченная последовательность величин одного типа. Массив характеризуется именем, размером и размерностью. Имена массивов образуются по тем же правилам, что и имена простых переменных. Размер массива определяет число элементов в массиве. Размерность массива – это число индексов, определяющих местоположение элементов в массиве. Индексы записываются после имени массива в скобках через запятую. Если размерность массива равна единице, то такой массив называют одномерным, если двум – двумерным.

Примером одномерного массива является вектор $\bar{a}(a_1, a_2, \ldots, a_n)$. На языке Basic данный вектор (одномерный массив) записывается в виде A(1), A(2),…,A(n).

Примером двумерного массива может служить матрица

$$A\begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1m} \\ a_{21} & a_{22} & \ldots & a_{2m} \\ & \ldots. & & \\ a_{n1} & a_{n2} & \ldots & a_{nm} \end{pmatrix},$$

каждый элемент которой имеет два индекса – номер строки и номер столбца. На языке Basic элементы данного двумерного массива записываются в виде A(1,1), A(1,2), …, A(n,m).

2.4. Функции

При программировании часто бывает необходимо вычислить значение функции (например, логарифм числа, корень квадратный и т. п.). Вычисление функций осуществляется с помощью подпрограмм, которые заранее запрограммированы. Для обращения к подпрограмме (вычисления значения функции) необходимо указать имя функции и в круглых скобках аргумент. В табл. 2.1 приведены наиболее часто употребляемые функции.

Таблица 2.1

Название функции	Математическое обозначение функции	Запись функции на языке Basic	Примечание
Синус	sin(x)	SIN(X)	x – в радианах
Косинус	cos(x)	COS(X)	x – в радианах
Тангенс	Tg(x)	TAN(X)	x – в радианах
Арктангенс	arctg(x)	ATN(X)	Вычисляется значение в интервале $(-\pi/2; +\pi/2)$
Логарифм натуральный	Ln(x)	LOG(X)	X > 0
Показательная	e^x	EXP(X)	
Корень квадратный	$\sqrt{x}$	SQR(X)	x≥0
Абсолютное значение	$\lvert x \rvert$	ABS(X)	
Целочисленная функция	]x[	INT(X)	Наибольшее целое, не превосходящее X
Отбрасывание дробной части		FIX(X)	Отбрасывает дробную часть
Генерирование случайного числа в интервале от 0 до 1		RND[(X)]	Аргумент может быть опущен

2.5. Выражения

Различают выражения *арифметические*, *логические*, *строковые*.

Арифметические выражения – это выражения, записанные с помощью констант, переменных, функций, знаков арифметических операций и круглых скобок. Результатом вычисления арифметического выражения является числовая константа. Порядок выполнения операций задается установленным приоритетом:

1) вычисление функции;
2) возведение в степень;

3) деление и умножение;
4) деление нацело и определение остатка (операция по модулю MOD);
5) сложение и вычитание.

Операции одного приоритета (умножение и деление; сложение и вычитание) выполняются слева направо в порядке следования. Для задания нужной последовательности выполнения операций используются круглые скобки.

В языке Qbasic кроме знакомых из арифметики операций существуют взаимообратные операции деления нацело и нахождение остатка от деления нацело. Покажем их.

Деление нацело (*целочисленное деление*). Целочисленное деление обозначается наклонной чертой \ . Перед делением компоненты действий округляются до целых чисел, а в частном отбрасывается остаток.

Например: 14 \ 2 = 2, 25.68 \ 7 = 3.

Операция по модулю MOD. Результат вычисления по модулю – целое число, являющееся остатком от деления нацело.

Например: 11 MOD 4 = 3 (11\ 4 = 2 остаток 3)
25.68 MOD 6.87 = 5 (26 \ 7 = 3 остаток 5)

При записи арифметических выражений необходимо придерживаться следующих правил и ограничений:

1. Все символы выражения записываются в одну сторону. Запрещены многоэтажные выражения и верхние и нижние индексы.
2. Два знака арифметических операций не должны располагаться рядом. Знак умножения опускать нельзя.
3. Операции в арифметическом выражении выполняются в порядке старшинства, т. е.:
   1) операции внутри скобок;
   2) вычисление встроенных функций;
   3) возведение в степень;
   4) умножение и деление;
   5) целочисленное деление;
   6) деление по модулю;

7) сложение и вычитание.

Операции равного старшинства выполняются по порядку слева направо. Исключение: A ^B^C = A^(B^C).

Тип арифметического выражения определяется типом ее результата. Примеры записи арифметических выражений приведены в табл. 2.2.

Таблица 2.2

Математическая запись	Запись на языке Basic
$\frac{\sqrt{2x^3+6{,}3}}{A-5{,}7}B-D$	SQR(2*X^3+6.3)/(A-5.7)*B-D
$\frac{\sqrt[3]{4x-\cos x^2}}{2{,}5+\lvert z\rvert}$	(4*X-COS(X^2)^(1/3)/(2.5+ABS(Z))
$\frac{\pi \sin x-\cos^2 x}{e^2 \ln\lvert 2x-x^3\rvert}$	(3.1415*SIN(x)-COS(X)^2)/(EXP(X)^2*LOG(ABS(2*X-X^3)))
$\frac{tgx+a}{\lg(x+a)}e^{-ax}$	(TAN(X)+A)/(LOG(X+A)/LOG(10))*EXP(-A*X)
$\left\lvert\frac{\ln(x+2{,}2)-4}{e^{-x}+e^{x}}\right\rvert$	ABS(LOG(X+2.2)-4)/(EXP(-X)+EXP(X)))

Логическое выражение служит для установления отношения между двумя числовыми или строковыми значениями. Результатом вычисления логического выражения является значение «Истина» или «Ложь». Для записи логического выражения используются операции отношения и логические операции (табл. 2.3).

Таблица 2.3

Операции отношения	Логические операции
= – равно	NOT – отрицание
<> – не равно	AND – логическое умножение (логическое И)
< – меньше	OR – логическое сложение (логическое ИЛИ)
> – больше	EQV – эквивалентность
<= – меньше или равно	–
>= – больше или равно	–

Порядок выполнения операций задается установленным приоритетом операций: операции отношения (выполняются слева направо в порядке следования), NOT, AND, OR, EQV. Для задания нужной последовательности выполнения операций используются круглые скобки (табл. 2.4).

Таблица 2.4

Математическая запись	Запись на языке Basic
a≥b	a>=b
-5 < x ≤ 4	x > -5 AND x <= 4
x ≤ -3 или x > 7	x <= -3 OR x > 7
число N кратно 5	N MOD 5= 0

Строковые выражения – это текст, заключенный в кавычки. Двойные кавычки называют ограничителями. Они служат для определения начала и конца текстового выражения. Строковое выражение состоит из строковых констант, строковых переменных и строковых функций. Результатом вычисления строкового выражения является строка символов. Например: “ДОБРОЕ УТРО”, “3 АВГУСТА 2006”, “КЛАССИЧЕСКАЯ МУЗЫКА”

ВОПРОСЫ И ЗАДАНИЯ

Простейшие конструкции языка программирования QBasic

1. Буквы какого алфавита используются в языке Qbasic?
2. Какие типы данных вам известны?
3. Что такое переменная? Что такое константа?
4. Что может быть именем переменной?
5. Как обозначается константа?
6. Какие типы переменных существуют? Как обозначается переменные разных типов? О чем говорит тип переменной?
7. Какие из групп символов являются числами в алгоритмическом языке:
   0 -5 ½ 3,14 +7.7 0.66… 0.(6) -0.85 2+2.5
8. Запишите числа так, как их пишут в математике:
   1.23E-04 0.056E3 -12.3E-11 6.54E+20 -0.12E02 0.12E-05 -3.2E+05

9. Запишите числа на Бейсике в форме с плавающей точкой:
   1200000. 0.000463 -110000. -0.03042 $-3.62\cdot10^{-14}$ $0.0034\cdot10^{+25}$
10. Что называется арифметическим выражением? Какой порядок выполнения операций в арифметическом выражении?
11. Записать формулы по правилам линейной записи:

$D=\sqrt{x_1^2+x_2^2}$ $Q=I^2R$ $J+2\pi R$ $S=v_0t+\frac{at^2}{2}$ $R=\frac{1}{\frac{1}{R_1}+\frac{1}{R_2}}$

12. Записать арифметические выражения на Qbasic:

а) $\frac{1}{x+\sqrt{x}}+\frac{\cos x}{x-\sqrt{x}}$

б) $\frac{\lg x}{\sqrt[3]{2+\sin^2 x^2}}$

с) $\frac{e^{-x}+e^{x}}{2a}$

д) $\frac{|4xtg3x|}{1+tg^3 3x}$

е) $\ln(\sin x)+\sqrt{x^{3+2}}+2$

ж) $2\sqrt[5]{x^2}+\sqrt{y^2}-|x+1|$

з) $\cos^2 ax+\sin^2 ax^3$

13. Восстановите математическую запись выражений:

а) Z=3*x^5-x^4+6*x^3-2*x^2

б) Y=1+x/2+x/3+x/4+x/5

в) $V=1/3*h_i*(g+SQR(g^r))$

г) Y=(3*ABS(X)+c^(1/3)+TAN(x))/(2*D-3*B)

КОНТРОЛЬНАЯ РАБОТА

1. Какие значения можно присвоить следующим переменным? Выбрать соответствующие ответы для данных переменных.

1) A	1) 5
2) A#	2) –5.7
3) A%	3) “IBM”
4) A$	4) 8E03
5) A!	5) –13.5D-100
6) A&	6) 5891089

2. Записать на Qbasic числа.

1) 6,38 2) $-24,*10^{-3}$ 3) $7,8*10^{103}$ 4) π

3. Указать неправильные записи чисел. Объяснить свой выбор.

1) 7,03 2) 7E8 3) 7. 4) .578 5) 7.3E-1.3

4. Записать по правилам языка программирования Qbasic следующие выражения.

1) $(\log_2 \frac{x}{3})^{\frac{3}{4}}$

2) $\cos^2 4x^3$

3) $\sqrt[5]{x+y} + \sqrt{arctg(x-1)^3}$

4) $6,673 \cdot 10^{-8} - \frac{m_1 \cdot m_2}{m_1 + m_2}$

5. Записать выражения в общепринятой форме.

1) 1.2D102*(X^(2/5)-ABS(X))/LOG(X^2+1)

2) 1.98E3*(EXP(X^3+1)^4+X)/ATN(X*2/1E-3)

3) X*(X^3+X^2-X)/SQR(X+1)+1D-110

Глава 3. ОСНОВНЫЕ ОПЕРАТОРЫ ЯЗЫКА QBASIC ДЛЯ РЕАЛИЗАЦИИ ЛИНЕЙНЫХ АЛГОРИТМОВ

3.1. Структура программы

Программа представляется в виде последовательности строк, описывающих алгоритм решения задачи. В Qbasic операторные строки не нумеруются. В зависимости от логики программы в строке может содержаться более одного оператора. В этом случае между операторами одной строки ставится разделительный знак **:** (двоеточие).

В Qbasic каждая операторная строка не метится, но она может содержать метку, которая используется для передачи управления в программе. В качестве метки можно использовать число или идентификатор, состоящий от 1 до 40 символов, начинающийся с буквы, а завершающийся двоеточием (**:**)**.** Например, ALFA:

В программе операторы бывают двух типов: исполняемые и неисполняемые. Исполняемые операторы реализуют действия, предусмотренные алгоритмом решения задачи. Неисполняемые операторы описывают свойства данных, комментируют текст программы. Неисполняемые операторы могут располагаться в любом месте программы до оператора конца. Программа может заканчиваться любым оператором. При явном указании на конец вычислений последним оператором программы должен быть оператор конца.

При описании операторов в квадратных скобках [] будем записывать необязательные параметры, а в фигурных {} – альтернативные параметры.

3.2. Оператор присваивания

Оператор присваивания служит для вычисления значения выражения и присваивания этого значения переменной.

В общем виде оператор представляется так: **имя =** *a***,** где имя – имя переменной; *a* – арифметическое выражение или константа.

Работа оператора: вычисляет значение арифметического выражения и присваивает его переменной, указанной слева. Для того чтобы опе-

ратор присваивания выполнялся правильно, необходимо, чтобы значения всех переменных, входящих в выражение, были определены. При записи оператора необходимо, чтобы имя переменной и выражение были одного типа, например, оба числовые или оба строковые.

П р и м е р ы : *Y = 2.236*

X=X+1

*Z= COS(2*X^3) – EXP(–X+3)/(2*C)*

F$ = “Фамилия”

A=B

Особенности оператора присваивания:

1) Не следует отождествлять присваивание со знаком математического равенства. Так, выражения A=B и B=A не тождественны.

2) Выражение A + 9 – B = X в программировании лишено смысла, т.к. имя переменной не может содержать знаков действий.

3.3. Операторы ввода

Операторы ввода используются для ввода значений переменных. Ввод значений переменных можно осуществить двумя способами: в диалоговом режиме и из блока данных. При диалоговом режиме ввода данные вводятся с клавиатуры непосредственно при выполнении программы.

Оператор ввода данных с клавиатуры (диалоговый)

Значения переменных вводятся в программу с клавиатуры по мере из запроса самой программой и размещаются в ячейках памяти, выделенных для этих величин. Оператор ввода дает возможность решать одну и ту же задачу с разными значениями исходных величин без изменения программы. В общем виде оператор представляется так:

INPUT [“Сообщение”] {,|;} список переменных

где INPUT – ключевое слово «ввести»; Сообщение – строковая константа, которая используется в качестве подсказки о том, какие данные и в каком порядке следует ввести пользователю с клавиатуры; Список переменных – имена переменных, перечисленных через запятую, значения которых предполагается вводить с клавиатуры.

Работа оператора: При выполнении оператора INPUT на экран выводится:

1) сообщение или знак «?» (при отсутствии сообщения в операторе) и выполнение программы приостанавливается.

2) Пользователь должен в ответ с клавиатуры ввести через запятую значения переменных, указанных в списке переменных. При вводе значений строковая константа заключается в кавычки, если она начинается со значащих пробелов или заканчивается ими, а также, если содержит запятые или двоеточия.

3) После окончания набора пользователь должен нажать клавишу Enter. Данные с экрана вводятся в компьютер. Пока клавиша Enter не нажата, вводимые символы можно редактировать.

З а м е ч а н и я :

1) Если сообщение отделяется от списка переменных точкой с запятой, то на экран выводится знак «?», если запятой, то знак «?» не выводится на экран;

2) Точка с запятой позволяет сохранять позицию курсора на той же строке после ввода данных;

3) При вводе слишком большого или недостающего количества данных появляется сообщение: "*Redo from start*" – повторить сначала.

П р и м е р : INPUT "Введите значения X, Y, Z", X, Y, Z

После выполнения этого оператора на экране появится сообщение:

Введите значения X, Y, Z

и выполнение программы приостанавливается. Пользователь с клавиатуры должен ввести числовые значения X, Y, Z через запятую, например: 3, 4, 5. После окончания ввода нажать клавишу Enter. Это действие будет равносильно операторам присваивания: X=3: Y=4: Z=5

Блоковый оператор ввода данных

Совместное действие операторов READ и DATA дают возможность вводит данные в программу из блока данных.

Ф о р м а т о п е р а т о р о в :

READ список переменных

DATA список констант

где READ, DATA – ключевые слова «читать», «данные» соответственно;

список переменных – имена переменных, перечисляемые через запятую; список констант – числовые или строковые константы, перечисляемые через запятую. Строковая константа заключается в кавычки, если она начинается со значащих пробелов или заканчивается ими, а также если содержит запятые или двоеточия.

Операторы READ и DATA не обязательно должны быть парными. Например, двум операторам READ может соответствовать один оператор DATA, и наоборот.

Перед выполнением программы просматриваются все операторы DATA в порядке следования в программе и формируется единый блок данных. Если в программе встречается оператор READ, то считывается текущее значение констант из блока данных в переменную и запоминается положение последнего считанного данного с помощью указателя позиции. Следующий оператор READ начинает выбирать данные с той позиции, которая была отмечена указателем позиции. Количество переменных в операторе READ должно быть не больше количества констант в операторе DATA. Неиспользованные константы из блока данных игнорируются. Но если блок данных исчерпан, то при попытке продолжить чтение, выдается сообщение об ошибке.

Можно возвратить указатель позиции в начало блока данных с помощью оператора RESTORE. Первый оператор READ, следующий за оператором RESTORE, будет считывать первое значение из блока данных.

Оператор DATA относится к невыполняемым операторам.

П р и м е р ы :

DATA –3.2, 12.3, 125

DATA Аргумент, 3.14

READ A, B, C, F$, P

После выполнения приведенных операторов переменные получат значения:

A = -3.2 B = 12.3 C = 125 F$= “Аргумент” P = 3.14

DATA 41, -12, 5, 32

READ A, B, C

RESTORE

READ X, Y, Z, F

После выполнения приведенных операторов переменные примут значения:

A = 41 B = -12 C = 5 X = 41 Y = -12 Z = 5 F = 32

3.4. Оператор вывода

Оператор вывода служит для вывода значений величин на экран монитора или принтер в процессе выполнения программы.

Общий вид оператора:

PRINT [список вывода]

где PRINT – ключевое слово «печать»; список вывода – константы, переменные, выражения, функции, разделенные запятыми или точками с запятой.

Работа оператора: оператор просматривает последовательно список вывода, и если элемент является константой, то выводит ее на экран, если переменной, то выводит на экран ее значение. Расположение на экране выводимых значений определяется разделителем элементов списка – точкой с запятой или запятой. Вывод данных можно организовать в двух форматах – зонном и компактном.

При *зонном формате* (печать по зонам) каждое значение выводится в своей зоне. Каждая строка содержит 5 зон по 14 позиций каждая. Для вывода данных в зонном формате элементы списка отделяются друг от друга запятой.

|1_ _1зона_ _ 14|15 _ _2 з о н а_ _28|29_ _3з о н а_ ___42| и т. д.

При выводе данных в *компактном формате* элементы списка отделяются друг от друга точкой с запятой, при этом числовые значения выводятся через пробел, а текстовые размещаются подряд. Необходимо помнить, что при выводе числовых данных первая позиция отводится под знак числа. Например:

X = 10 : Y= - 2

PRINT " При X= "; X, "Y=";Y

PRINT " X + Y = "; X +Y

PRINT " X - Y = ", X -Y

*PRINT " X*Y = "; X * Y*

После выполнения вышеприведенных операторов на экране будет выведено:

При X= 10 Y=-2

X + Y = 8

X - Y = 12

X*Y =-20

Если в конце списка вывода оператора PRINT стоит запятая или точка с запятой, то следующий оператор PRINT выводит данные в той же строке, что и предыдущий. Например:

G=1998: D$="Год рождения - "

PRINT D$;

PRINT G

После выполнения вышеприведенных операторов на экране будет выведено:

Год рождения – 1998

Особенности записи оператора PRINT

1) Оператор PRINT может записываться без списка величин. В этом случае он выводит на экран пустую строку, строку пробелов.
2) В одном операторе PRINT можно использовать разные разделители.
3) Если в программе используется несколько операторов PRINT, то работают они как один с общим списком выводимых величин с единственным исключением: при отсутствии в конце оператора PRINT разделителя (**,** или ;) следующий за ним оператор PRINT выводит величины с начала строки. Например:

PRINT "МАША";

PRINT "ЛЮБИТ",

PRINT "КАШУ"

PRINT "ОЧЕНЬ"

После выполнения выше приведенных операторов на экране будет выведено:

МАША ЛЮБИТ КАШУ
ОЧЕНЬ

Функция Tab в операторе PRINT

Функция Tab устанавливает позицию курсора на определенной позиции экрана. С установленной позиции и будет начинаться вывод информации, содержащейся в операторе PRINT. Общая форма функции выглядит следующим образом: **Tab (N)**, где N – целое число от 1 до 80.

Работа оператора PRINT с функцией Tab: оператор PRINT TAB(n); X выводит значение X с n-й позиции строки.

Ограничение: Если n меньше номера текущей позиции строки, то X выводится с n-й позиции следующей строки.

Например, в результате выполнения операторов

10 C=10

20 B= - 20

30 PRINT TAB(5); C , TAB(10) ; B , TAB(20) ;"DATA"

на экране получим:

```
          10
                -20                 DATA
---------|5--------|10-------------------|20
```

3.5. Оператор комментария

Невыполняемый оператор комментария служит для пояснений к программе. Формат оператора: **REM текст,** где REM – ключевое слово «ремарка»; текст – текст комментария, который может включать набор любых символов.

Ключевое слово REM можно заменить символом «'» (апостроф). Оператор комментария должен быть единственным в операторной строке, либо последним. Например: REM Задание 1

' Описание массивов в программе

ВОПРОСЫ И ЗАДАНИЯ

Программирование линейных алгоритмов

1. Как оформляется оператор ввода?
2. Что можно указывать в качестве элементов списка ввода?
3. Как работает оператор ввода (что происходит при его выполнении)?
4. Как оформляется оператор вывода на экран?
5. Что можно указывать в качестве элементов списка вывода?
6. Какой символ используется для разделения элементов списка вывода?
7. Что будет выведено на экран, если в списке вывода записано:
   a) число?
   b) имя величины?
   c) текст в кавычках?
   d) арифметическое выражение?
8. Как оформляется оператор присваивания? Как он работает (что происходит при его выполнении)?
9. Составить программу вывода на экран числа, вводимого с клавиатуры. Выводимому числу должно предшествовать сообщение «Вы ввели число».
10. Составить программу вывода на экран числа, вводимого с клавиатуры. После выводимого числа должно следовать сообщение «вот какое число Вы ввели».
11. Вывести на одной строчке числа 5, 18 и 67 с одним пробелом между ними.
12. Вывести на одной строчке числа 8, 13 и 111 с двумя пробелами между ними.
13. Вывести на экран числа 50 и 10 одно под другим.
14. Какие значения будут выведены в результате выполнения последовательности операторов? Изобразите, в каком виде выведутся результаты выполнения программы на экран монитора. Составить блок-схему.

Программа 1	Программа 2
Q=5	X=(SIN(SQR(0))+2*ABS(-2))/COS(0)
Z=Q+8	Y=FIX(-3.17)
PRINT Q;Z	Z=10\4
	R=INT (-2.5)
	PRINT X;Y
	PRINT Z,R

15. Указать значение величины S после выполнения следующих операторов присваивания:

a)	b)	c)	d)
S=5	S=6	S=45	K=30
S=57	S=-5.2*S	K=-25	D=K-5
	S=0	S=S+K	K=2*D
			S=K-100

16. Указать значения величины *а* и *в* после выполнения следующих операторов присваивания:

a)	b)
a=5.8	a=0
b=-7.9	b=-9.99
b=a	b=a
a=b	a=b

17. Даны два числа. Найти их сумму, разность, произведение, а также частное от деления первого числа на второе.
18. Даны два числа. Найти среднее арифметическое и среднее геометрическое их модулей.
19. Составить программу решения линейного уравнения ax + b = 0 (a≠0).
20. Составить программу обмена значениями двух переменных величин.
21. Дано вещественное число a. Пользуясь только операцией умножения, получить a^4 за две операции.
22. Составить блок-схему и написать программу вычисления значения функции $y = e^{\cos\frac{\pi - x}{a}} \sqrt[b]{\arcsin\sqrt{0.25645}} + a; \quad a = 11, b = 4, x = -6.1$ (ответ: 12.6638).

Глава 4. ОПЕРАТОРЫ ПЕРЕДАЧИ УПРАВЛЕНИЯ ДЛЯ РЕАЛИЗАЦИИ РАЗВЕТВЛЯЮЩИХСЯ АЛГОРИТМОВ

4.1. Оператор безусловного перехода

Оператор безусловного перехода служит для перехода из одной точки (строки) к другой, указанной операторной строке.

Общий вид оператора: **GOTO nc,** где GOTO – ключевое слово «перейти к …»; nc – метка операторной строки на которую осуществляется переход.

Работа оператора: оператор GO TO обеспечивает переход к строке с меткой **nc.**

Оператор GO TO должен быть единственным в операторной строке либо последним.

П р и м е р ы :

1)

```
30 X=X+1
40 Y=(X+2)*3
GOTO 30
```

Оператор GO TO обеспечивает циклическое выполнение двух операторов 30 и 40.

2)

```
40 X=3*A
50 GO TO 70
60 Y=2*X
70 Z=5*X
```

Оператор GO TO позволяет обойти 60-ю строку.

4.2. Операторы условного перехода

Операторы условного перехода обеспечивают проверку условия и организацию ветвления. Имеют несколько вариантов записей.

Общий вид оператора:

IF условие THEN оператор [ELSE оператор]

где IF, THEN, ELSE – ключевые слова «если» …, «то» … «иначе»;
условие – логическое выражение;

оператор – один или группа операторов, разделенных символом «:» (двоеточие).

Работа оператора: Оператор IF проверяет условие, если логическое выражение принимает значение «истина», то выполняется оператор, следующий за THEN, затем управление передается следующей операторной строке. Если логическое выражение принимает значение «ложь», то выполняется оператор, следующий за оператором ELSE. Если в операторе IF опущена ветвь ELSE, то при значении логического выражения «ложь» управление передается следующей операторной строке.

П р и м е р :

Y=20

IF X<=10 THEN Y=180 ELSE Z=Y+10

Результат работы:

при X=5 Y=180

при X=20 Z=30

Рассмотренный оператор условного перехода называется *линейным*.

Рассмотрим еще одну форму оператора условного перехода – *блочный оператор условного перехода*.

Общий вид оператора:

IF условие_1 THEN
оператор_1
ELSEIF условие_2 THEN
оператор_2
…
ELSEIF условие_n-1 THEN
оператор_n-1
ELSE
оператор_n
END IF

Данный оператор работает следующим образом. Проверяется условие_1, если условие_1 принимает значение «истина», то выполняется оператор_1 и управление передается оператору, следующему за оператором END IF. Если условие_1 принимает значение «ложь», то проверяется усло-

вие _2 и если оно принимает значение «истина», то выполняется оператор_2 и управление передается оператору, следующему за оператором END IF и т.д. Если в операторе все условия принимают значение «ложь», то выполняется оператор_n и управление передается оператору, следующему за оператором END IF.

П р и м е р :

Составить программу вычисления функции

$$Y=\begin{cases}\sin x, & \text{при } x \le 0\\ \cos x, & \text{при } 0 < x < 3.1415\\ \ln x, & \text{при } x \ge 3.1415\end{cases}$$

Программа, использующая линейную форму оператора условного перехода:

```
INPUT "Введите значение x", x
IF x <= 0 THEN Y = SIN(x): GOTO 10
IF x >= 3.1415 THEN Y = LOG(x) ELSE Y = COS(x)
10 PRINT "При x="; x; "Y="; Y
END
```

Программа, использующая блочную форму оператора условного перехода:

```
INPUT "Введите значение x", x
IF x <= 0 THEN
Y = SIN(x)
ELSEIF x >= 3.1415 THEN
Y = LOG(x)
ELSE
Y = COS(x)
END IF
10 PRINT "При x="; x; "Y="; Y
END
```

ВОПРОСЫ И ЗАДАНИЯ

Программирование разветвляющихся алгоритмов

1. Какие виды условных операторов вы знаете?
2. В каких случаях в программе используется полный условный оператор? Как он оформляется? Как он работает (что происходит при его выполнении)?
3. В каких случаях в программе используется неполный условный оператор? Как он работает (что происходит при его выполнении)?
4. Определить максимальное и минимальное значения из двух различных вещественных чисел.
5. Даны вещественные числа *a*, *b*, *c* (a≠0). Выяснить, имеет ли уравнение $ax^2 + bx + c = 0$.
6. Определить, является ли число *a* делителем числа *b*? А наоборот?
7. Без использования компьютера изобразите, в каком виде выведутся результаты выполнения программы на экран монитора.

```
T=1
20 PRINT T, T*T
T=T+2
IF T<=10 THEN GOTO 20
```

8. Составить программу с получением на ПК правильных ответов.

$$y = \begin{cases} e^x & when\ x \le 0 \\ x \ln x & when\ 0 < x \le 1 \\ x + 1 & when\ x > 1 \end{cases}$$

Ответы:

x=-1	x=0	x=0.5	x=1	x=2
y=0.368	y=1.0	y=-.347	y=0	y=3.0

Глава 5. ОПЕРАТОРЫ ЦИКЛА ДЛЯ РЕАЛИЗАЦИИ ЦИКЛИЧЕСКИХ АЛГОРИТМОВ

Для решения задач иногда бывает необходимо многократно выполнить вычисления по одним и тем же зависимостям при различных значениях входящих в них параметров. Такой вычислительный процесс называется циклическим, а многократно повторяющиеся участки этого процесса – циклами (см. блок-схемы на рис 1.6 и 1.7).

Для реализации циклического процесса можно использовать операторы присваивания, безусловного перехода, условного перехода, но есть и специальные операторы.

Оператор цикла с условием WHILE…WEND

Общий вид оператора:

WHILE условие
операторы рабочей части цикла
WEND

где WHILE, WEND – ключевые слова; условие – логическое выражение; операторы рабочей части цикла – многократно повторяющиеся зависимости.

Работа оператора. Проверяется условие, если оно принимает значение «истина», то выполняются операторы рабочей части цикла, затем опять осуществляется проверка условия. Операторы рабочей части цикла выполняются до тех пор, пока условие принимает значение «истина». Если условие принимает значение «ложь», то управление передается оператору, следующему за оператором WEND.

П р и м е р : Составить программу для вычисления наибольшего целого положительного числа n, удовлетворяющее условию $3n^3 - 690\ n \leq 7$.

```
REM Вычисление наибольшего целого
N = 1
WHILE 3 * N ^ 3 - 690 * N <= 7
N = N + 1
WEND
PRINT "Наибольшее целое N ="; N – 1
END
```

Операторы цикла с параметром FOR … NEXT

Вместо оператора перехода в ряде случаев удобно использовать так называемый оператор цикла, который позволяет многократно (заданное число раз) выполнять определенную группу операторов.

О б щ и й в и д :

FOR I=In TO Ik [STEP h]

операторы рабочей части цикла

NEXT

где FOR и NEXT – ключевые слова «для» и «следующий» соответственно; I – переменная (параметр) цикла; In, Ik – начальное и конечное значение переменной цикла I; h – шаг изменения переменной цикла; если h = 1, то его в операторе можно не указывать; операторы рабочей части цикла – многократно повторяющиеся зависимости.

Работа оператора:

1) вычисляются или задаются значения для In, Ik, h;
2) параметру цикла I присваивается начальное значение In;
3) если $h > 0$ и $I \le In$ или $h < 0$ и $I \ge In$, то выполняются операторы рабочей части цикла;
4) параметр цикла I изменяется на величину шага и повторяется п. 3;
5) если в п. 3 указанные условия не выполняются, то осуществляется выход из цикла и управление передается операторной строке, следующей за оператором NEXT.

Например, вычислим, какое значение примет переменная F после выполнения программы:

```
10 F = 1
20 FOR I = 1 TO 7 STEP 2
30 F = F*I
40 NEXT I
50 PRINT F
60 END
```

Для удобства пояснений к выполняемым действиям операторы в программе пронумерованы. Запишем последовательность вычислений, производимых программой, причем в пояснениях номер будет соответствовать выполняемой операторной строке (табл. 5.1).

Таблица 5.1

Номер шага	Номер оператора	Операция	Условие
1	10	F = 1	
2	20	i = 1	1 ≤ 7
3	30	F = 1*1= 1	
4	40	переход к строке 20	
5	20	i = 1+2 =3	3 ≤ 7
6	30	F = 1*3= 3	
7	40	переход к строке 20	
8	20	i = 3+2 =5	5 ≤ 7
9	30	F = 3*5= 15	
10	40	переход к строке 20	
11	20	i = 5+2 =7	7 ≤ 7
12	30	F= 15*7= 105	
13	40	переход к строке 20	
14	20	i = 7+2 =9	9 > 7
15	50	F=105	
16	60		конец

Таким образом, после выполнения программы F = 105.

Особенности работы оператора цикла с параметром:

число выполнений цикла $K=\left[\frac{Ik - In}{h}\right]+1$

П р и м е р . Составить программу для вычисления и печати таблицы умножения на 12.

```
REM Таблица умножения
FOR I = 1 TO 12
p = I * 12
PRINT I; "*12 ="; p
NEXT I
END
```

ВОПРОСЫ И ЗАДАНИЯ

Программирование циклических процессов

1. В каких случаях используются операторы цикла с условием?
2. Что такое «тело оператора цикла с условием»?
3. Может ли тело оператора цикла с условием не выполниться ни разу?
4. Может ли тело оператора цикла с условием выполняться бесконечное число раз?
5. Дано число n. Из чисел 1, 4, 9, 16, 25, … . Напечатать те, которые не превышают n.
6. В каких случаях используется оператор цикла с параметром? Как он оформляется? Как он работает (что происходит при его выполнении)?
7. Может ли тело оператора цикла с параметром не выполниться ни разу?
8. Чему равно количество повторений тела оператора цикла с параметром, если параметр цикла принимает:
    a. все целые значения от 1 до 10?
    b. все целые значения от a до b?
    c. все нечетные значения от 1 до 20?
    d. все значения от 10 до 100 с шагом 7?
9. Можно ли в теле цикла с параметром не использовать величину – параметр цикла?
10. Какое значение примет переменная P в результате выполнения программы?

```
P = 1
FOR I = 1 TO 5 STEP 2
P = P * (P + I)
NEXT I
PRINT P
```

11. Вычислить значение выражения *y*.

$$Y = \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \frac{1}{5^2} + \cdots + \frac{1}{15^2}$$

Ответ: y = 0,58044

Глава 6. ОПЕРАТОРЫ РАБОТЫ С МАССИВАМИ

6.1. Оператор описания массивов

Оператор описания массивов служит для резервирования места в памяти компьютера для элементов массивов. Является неисполняемым.

Общий вид оператора: **DIM список массивов,** где DIM – ключевое слово «размерность»; список массивов – имена массивов с указанием верхних границ каждого индекса (нижние границы равны нулю).

Например: DIM A(19), X(3,4), F$(9)

Оператор DIM резервирует место в памяти для одномерного массива A, состоящего из 20 элементов A(0), A(1), A(2), …, A(19), для двумерного массива X, содержащего 4 строки и 5 столбцов (включая нулевые) X(0,0), X(0,1), …, X(3,4), и для строкового массива F$, содержащего 10 элементов F$(0), F$(1), …, F$(9).

После описания массивов начальные значения для всех числовых массивов равны нулю, для строковых – пустой строке.

6.2. Ввод и вывод элементов массивов

Для ввода и вывода элементов массива используются операторы ввода и вывода INPUT, READ, DATA, PRINT. Ввод и вывод массивов осуществляется с помощью оператора цикла FOR…NEXT. Операторы ввода или вывода помещаются внутри циклического участка, и при каждом выполнении цикла производится ввод или вывод одного значения элемента массива.

Рассмотрим примеры ввода и вывода массива.

П р и м е р : Заданы элементы одномерного массива

A(-3, 6, 7, -3, 4, -5, 0, -2, 1, 12, -9, 9)

1. Записать операторы ввода элементов одномерного массива A, используя оператор INPUT.

2. Записать операторы ввода элементов одномерного массива A, используя операторы READ, DATA.

3. Записать операторы вывода элементов одномерного массива А в строчку.

4. Записать операторы вывода элементов одномерного массива А в столбец.

Р е ш е н и е

1) элементы массива вводятся с клавиатуры:

```
DIM A(12)
PRINT “Введите элементы массива А”
FOR i= 1 TO 12
INPUT A(i)
NEXT i
END
```

2) элементы массива вводятся из блока данных:

```
DIM A(12)
DATA -3, 6, 7, -3, 4, -5, 0, -2, 1, 12, -9,9
FOR i= 1 TO 12
READ A(i)
NEXT i
END
```

3) элементы массива выводятся на экран в столбец

```
FOR i= 1 TO 12
PRINT A(i)
NEXT i
END
```

4) элементы массива выводятся на экран в строку

```
FOR i= 1 TO 12
PRINT A(i);
NEXT i
END
```

П р и м е р : Заданы элементы одномерного массива B(n) (B(1), B(2),…,B(n)). Записать операторы ввода и вывода данного массива.

Р е ш е н и е

```
INPUT “Введите размер массива n”; n
DIM B(n)
PRINT “Введите элементы массива B”
FOR i= 1 TO n
INPUT B(i)
NEXT i
FOR i= 1 TO n
PRINT B(i);
NEXT i
END
```

В данном примере сначала с клавиатуры вводится размер массива n, затем сами элементы массива. Элементы массива будут выведены на экран в строку.

П р и м е р : Заданы элементы двумерного массива C(n, m). Записать операторы ввода и вывода данного массива.

Р е ш е н и е

```
INPUT “Введите число строк массива n”; n
INPUT “Введите число столбцов массива m”; m
DIM C(n, m)
PRINT “Введите элементы массива C”
FOR i= 1 TO n
FOR j= 1 TO m
INPUT C(i, j)
NEXT j
NEXT i
FOR i= 1 TO n
FOR j= 1 TO m
PRINT C(i, j);
NEXT j
PRINT
```

NEXT i

END

В данном примере сначала с клавиатуры вводится размер массива C (число строк и столбцов), затем сами элементы массива. Заметим, что для ввода массива C использовался вложенный цикл и элементы массива вводятся по строкам, т. е. сначала вводятся элементы первой строки, затем второй и т. д.

Элементы массива будут выведены на экран в общепринятом виде, т. е. в виде матрицы. Для вывода массива также используется вложенный цикл.

ВОПРОСЫ И ЗАДАНИЯ

Одномерные и двумерные массивы

1. Что такое одномерные массивы? Как они описываются?
2. Для чего используются одномерные массивы?
3. Как называется номер элемента одномерного массива?
4. Как можно заполнить одномерный массив?
5. Для чего в программах используются двумерные массивы? Как они описываются?
6. Сколько индексов характеризуют конкретный элемент двумерного массива?
7. Заполнить одномерный массив из восьми элементов следующими значениями: первый элемент массива равен 73, второй – 0, третий – 40, четвертый – 36, пятый – 74, шестой – 46, седьмой – 0, восьмой – 13.
8. Дан массив. Составить программу для вычисления суммы всех элементов массива (для вычисления произведения всех элементов массива).
9. Какую работу выполняет данный фрагмент программы?

```
INPUT n
DIM A(n)
S = 0
FOR i = 1 TO n
INPUT A(i)
S = S + A(i)
NEXT i
PRINT S
END
```

ЗАКЛЮЧЕНИЕ

Хочется верить, что, пролистав учебное пособие, учащийся не потеряет к нему интерес и найдет применение изложенному материалу на лабораторных занятиях при решении задач на составление алгоритмов, написании программ, а также при подготовке и сдаче компьютерных тестов и экзамена по рассмотренным темам.

Почему авторы выбрали язык программирования Qbasic, ведь в настоящее время существует большое количество алгоритмических языков, которым присущи как общие, так и отличительные черты. Это и Фортран, и Паскаль и др.

Бейсик (BASIC) – это сокращение английских слов Beginners All-purpouse Symbolic Instraction Code, что в переводе означает «многоцелевой язык символических инструкций для начинающих». Он был разработан профессорами Дартмутского колледжа (США) Т. Куртцем и Дж. Кемени в 1965 году для обучения студентов, незнакомых с вычислительной техникой. Этот язык, благодаря простоте, стал очень популярным. Особенно его популярность повысилась с появлением персональных компьютеров, где он стал одним из основных языков программирования.

Существует множество версий языка Бейсик, более сорока, и все они имеют особенности. Это и Qbasic, и Visual Basic, и др. Но в каждой из них можно выделить общее, в котором отражены характерные (стандартные) грамматика, синтаксис и семантика языка. Наиболее популярной версией для обучения программированию является Qbasic, благодаря удобному интерфейсу и представлению пользователю ряда сервисных возможностей, присущих современным системам программирования. На примере Qbasic можно наглядно увидеть те характерные особенности, которые присущи программированию на алгоритмических языках вообще. Поэтому тексты представленных в учебном пособии программ отлажены именно в нем.

Теоретический материал пособия закрепляется при выполнении упражнений и контрольных тестов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

Основная литература

1. Бобровский С.И. Программирование на языке Qbasic для школьников и студентов / С.И. Бобровский. – М.: изд-во Инфорком-Пресс, ТехБук(Десс), 2000. – 207с.
2. Златопольский Д.М. Сборник задач по программированию / Д.М. Златопольский. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2007. – 240с.: ил.
3. Ляхович В.Ф. Основы информатики / В.Ф. Ляхович. – Ростов н/Д. : «Феникс», 2000. – 608с.
4. Михайлов В.Ю. Современный Бейсик для IBM PC. Среда, язык, программирование / В.Ю. Михайлов, В.М. Степанников. – М.: изд-во МАИ, 1993. – 288с. : ил.
5. Ставнистый Н.Н. Qbasic в математике. Решение задач с помощью компьютера / Н.Н. Ставнистый. – М:. Солон-Р, 2001. – 143с.

Дополнительная литература

1. Задачи по программированию / С.Н Абрамов, Г.Г. Гнездилов, Е.Н. Капустин и др. – М.: Наука, 1988. – 224с.
2. Введение в программирование на языке Microsoft BASIC : учеб. пособие / под. ред. В. Г. Потемкина. – М. : Диалог-МИФИ, 1991. – 160с.

ПРИЛОЖЕНИЕ

ТЕРМИНОЛОГИЧЕСКИЙ СЛОВАРЬ

К главе 1

алгоритм
алгоритмический язык
блок
блок-схема
данные
детерминированность
дискретность
массовость
оператор
разветвляющийся алгоритм
результативность
цикл

К главе 2

арифметическое выражение
идентификатор
индекс
константа
логическое выражение
переменная
переменная с индексом
строковая константа
целая константа
экспоненциальная форма
числа

К главе 3

оператор присваивания
оператор ввода INPUT
оператор ввода данных READ, DATA
оператор вывода (печати) PRINT
оператор комментария REM
Метка операторной строки

К главе 4

оператор безусловного перехода
оператор условного перехода
Ложь
Истина

К главе 5

параметр цикла
цикл с условием
цикл с параметром
WHILE…WEND
FOR…NEXT

К главе 6

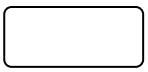
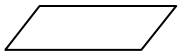
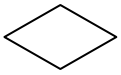
двумерный массив
индекс
массив
одномерный массив
оператор описания массива
DIM
размерность массива
столбец
строка
элемент массива

ПРИМЕР ЭКЗАМЕНАЦИОННОГО ТЕСТА

В КАЖДОМ ВОПРОСЕ УКАЖИТЕ НОМЕР, СООТВЕТСТВУЮЩИЙ ПРАВИЛЬНОМУ ОТВЕТУ, ИЛИ РЕЗУЛЬТАТ ВЫЧИСЛЕНИЙ

1. Алгоритм является...
   1) последовательностью команд, которую может выполнить исполнитель
   2) системой команд исполнителя
   3) математической моделью
   4) информационной моделью

2. Какая фигура в блок-схеме обозначает выполнение операции или группы операции?

1)  2)  3)  4) 

3. В качестве имени переменной не может быть:

1) A_1 2) A.1 3)A+1 4) A$

4. Алфавит языка Qbasic не включает:
   1) буквы латинского алфавита
   2) знаки арифметических операций
   3) цифры
   4) буквы русского алфавита

5. Для написания комментария на языке Qbasic можно использовать:
   1) оператор DIM
   2) оператор REM
   3) символы {} после операторов
   4) символ кавычки в конце строки

6. Дано описание массива DIM S(18). Данный массив содержит
   1) строковые (символьные) значения
   2) любые числовые значения

3) только целые числовые значения

4) только дробные числовые значения

7. Какой оператор выведет на экран значение переменной X

1) PRINT«X»

2) PRINT X

3) INPUT X

8. Объявите переменные, необходимые для вычисления значения функции $y=\ln x^2$?

1) x 2) y 3) x, y 4) ln x^2 5) x, y, ln x2

9. Задано арифметическое выражение. Указать номер выражения, записанного верно на языке Basic.

$$\frac{\sin 2\alpha + e^{2x-1}}{\ln\left|x^2-1\right| + x}$$

1) Y=(SIN(2A)+EXP(2*X- 1))/(LOG(ABS(X^2- 1))+X)
2) Y=(SIN(2*A)+EXP^2*X-1)/(LOG(ABS(X^2-1))+X)
3) Y=(SIN(2*A)+EXP(2*X-1))/LOG(ABS(X^2-1))+X
4) Y=(SIN(2*A)+EXP(2*X-1))/((LOG(ABS(X^2-1))+X)
5) Y=SIN(2* A)+EXP(2*X- 1)/LOG(ABS(X^2-1))+X

10. Чему равны значения переменных и *y* после выполнения операторов?

x=10: y=5: c=x: x=y: y=c

11. Чему равны значения переменных A и B после выполнения данной программы:

```
A=4: B=7
IF A<B THEN A=A+B: GOTO 40
B=A+B
PRINT A; B
```

12. Какое значение примет переменная D в результате выполнения программы?

D=1

```
FOR K = 1 TO 3
D = D + 2* K
NEXT K
PRINT D
```

13. Сколько раз выполнился цикл в предыдущем задании?
14. Какое значение примет величина k при выполнении программы?

```
DATA 3, 4, 5
k = 0
FOR i = 1 TO 3
READ m
k = k + m + 1
PRINT k
END
```

15. Какую работу выполняет данный фрагмент программы?

```
DIM a(15)
b = 1
FOR 1 = 1 TO 15
INPUT a(i)
IF a(i) < O THEN b = b * a(i)
NEXT i
PRINT b
END
```

1) Вычисление произведения элементов массива
2) Вычисление произведения отрицательных элементов массива
3) Вычисление произведения положительных элементов массива
4) Сортировку элементов массива в порядке возрастания
5) Вычисление количества отрицательных элементов массива

ОГЛАВЛЕНИЕ