

“ОПЕРАТОРЫ ЯЗЫКА ЛИНЕЙНЫЕ ПРОГРАММЫ”

Оформление программ.

Главной частью программы является последовательность инструкций, которую должен выполнить компьютер. Эти инструкции принято называть операторами языка, а всю последовательность инструкций — программой.

После служебного слова, означающего название оператора, обязательно ставится пробел.

В отличие от старых версий языка, в которых каждая строка программы нумеровалась, QBASIC не требует обязательной нумерации.

Если на одной строке записывается несколько операторов, то их отделяют друг от друга двоеточием. Набор строки на компьютере заканчивается нажатием клавиши {Enter}. Заканчивается программа оператором конца программы — END.

Элементы команды можно разделять произвольным количеством пробелов. Допускаются также и пустые строки. QBASIC сам форматирует строки — переводит строчные буквы служебных слов в прописные и расставляет пробелы.

Правила хорошего тона в программировании рекомендуют использовать для блоков команд различного уровня вложенности разные отступы от левой границы строки — так программу легче читать.

Например:

```
FOR K=1 TO 10 STEP 1
  PRINT K
  PRINT K^2
NEXT K
```

Любую команду в программе можно снабдить поясняющим текстом — комментарием, который начинается служебным словом REM.

Комментарии используются исключительно для удобств того человека, который будет читать или проверять Вашу программу.

Например: **REM Это моя первая программа.**

Служебное слово REM можно заменить знаком апостроф :

‘ Это моя первая программа.

С другой стороны не стоит злоупотреблять большим количеством комментариев и комментировать очевидные действия.

Для того, чтобы при выводе результатов выполнения программы на экране не было лишних данных, в начале программы целесообразно поставить оператор очистки экрана **CLS**.

Оператор присваивания LET.

Память компьютера можно сравнить с раздевалкой в детском саду, где имеется какое-то количество шкафчиков для одежды — в программе это переменные. На каждом шкафчике наклеена определенная картинка, отличная от других — это имена переменных. Когда дети приходят в детский сад, они вешают одежду в эти шкафчики — это значение переменных. Часто для значения величин отводится сразу несколько ячеек.

Для того, чтобы задать значение какой-нибудь переменной в память, чаще всего пользуются оператором присваивания.

ОБЩИЙ ВИД:

LET имя переменной = выражение

В современных версиях языка разрешается опускать служебное слово LET, таким образом оператор присваивания примет вид:

Имя переменной = выражение

Работа команды делится на два этапа:

1 этап Вычисляется значение выражения, стоящего в правой части команды.

2 этап Полученное значение присваивается переменной, имя которой стоит в левой части команды.

Если такой переменной еще нет, то компьютер заводит ее и помещает значение туда.

ВНИМАНИЕ! В команде присваивания типы переменной и выражения должны быть согласованы: нельзя числовой переменной присваивать символьное значение и наоборот.

Команду присваивания нельзя отождествлять со знаком равенства.

Читать команду **A=5** следует таким образом: “Переменной A присвоить значение 5.” или просто “A присвоить 5”.

Команду **A=A+1** следует понимать следующим образом. Выше описаны два этапа работы команды присваивания: на первом этапе вычисляется значение **A+1**. Для компьютера **A** — это не буква, а переменная с именем **A**, в которой лежит какое-то значение. Таким образом компьютер к этому значению прибавит 1. Полученное число 6 на втором этапе будет помещено в переменную, имя которой записано слева, т.е. в переменную **A**. Старое значение будет заменено новым.

При записи выражений в команде присваивания необходимо помнить о приоритете операций:

- 1) возведение в степень и действия в скобках.
- 2) умножение и деление.
- 3) сложение и вычитание.

Задача 1.

Какие значения последовательно будет получать переменная X при выполнении следующих команд

X=10

X=X+5

X=X/3

X=X*2+1?

Решение:

X=10

X=X+5 → X=10+5=15 → X=15

X=X/3 → X=15/3=5 → X=5

X=X*2+1 → X=5*2+1=26 → X=26

Задача 2.

Определите, какие значения будут присвоены переменным A, B и C в результате выполнения действий.

а) A\$="РАДИО"

B\$="СЛУШАТЕЛЬ"

C\$=A\$+B\$

Решение: C\$="РАДИО"+"СЛУШАТЕЛЬ"="РАДИОСЛУШАТЕЛЬ"

б) A=5

B=A+4

C=A+B

Решение: A=5

B=5+4=9 → B=9

C=5+9=14 → C=14

Задача 3.

Найдите ошибку в программе:

а) A=5

B\$=6 — Ошибка! Символьной переменной присваивается числовое значение.

б) A=13

B\$="число"

C=A+B\$ — Ошибка! Числовой переменной присваивается сумма переменных разных типов.

Задача 4.

Запишите команды в результате выполнения которых значения переменных A и B поменяются.

Решение: Эту задачу можно решить двумя способами — с помощью вспомогательной переменной и с помощью встроенной функции SWAP.

Рассмотрим первый способ:

X=A :REM запомнили значение переменной A

A=B :REM переменной A присвоили значение переменной B

B=X :REM B присвоили X, т.е. значение переменной A.

Второй способ:

Обмен значений переменных с помощью оператора **SWAP переменная 1 ,переменная 2**

Например, SWAP A, B.

Оператор PRINT.

Оператор PRINT осуществляет вывод на экран алфавитно-цифровой информации и результатов вычислений.

ОБЩИЙ ВИД ОПЕРАТОРА PRINT:

PRINT □ список вывода

В списке вывода могут быть как числовые и строковые константы, так и имена переменных.

Например:

О П Е Р А Т О Р	Д Е Й С Т В И Е	Р Е З У Л Ь Т А Т
PRINT • 15	Выводит на экран число 15	15
PRINT • "число"	Выводит на экран текстовую константу "число"	число

PRINT • A	Выводит на экран значение переменной A (при A=5)	5
PRINT • 10*3.1	Выводит значение выражения	31

Список вывода может содержать смешанные данные, т.е. и константы, и величины. Данные в списке могут быть разделены “;” или “,”. В конце списка так же может стоять один из этих знаков.

Чтобы элементы списка вывести подряд в одной строке, в операторе PRINT их разделяют запятыми или точкой с запятой. Если же этот знак стоит в конце списка вывода, то следующая команда PRINT продолжит вывод в этой же строке.

Если элементы списка разделены точкой с запятой, то они выводятся в одной строке через одну позицию. Элементы списка вывода, разделенные запятыми, выводятся каждый с новой зоны строки. Строка разбивается на зоны по 14 символов.

Пример:

A=5

PRINT “КОМПЬЮТЕР”

PRINT “A=”,A

PRINT A*10;A*100;A*1000

PRINT 2,4,6;

PRINT “-- четные числа”

```

КОМПЬЮТЕР
A=5
50 500 5000
2      4      6-- четные числа
  14     14

```

Форматированный вывод чисел на печать PRINT USING.

Знак “#” указывает сколько позиций при печати следует отвести под число.

Например, PRINT USING “###”; 5;34

В этом случае число 5 будет напечатано в четвертой позиции от начала, а число 34 в третьей и четвертой позиции от числа 5, т.е. на вывод каждого числа отводится по четыре позиции.

```

• • • 5 • • 34

```

Таким образом, числа при печати выравниваются по правому краю отведенного места.

Также можно указывать количество знаков после десятичной точки при выводе вещественных чисел.

Например, PRINT USING “##.##”;57.389;2.5

В этом случае на экран будет выведено число 57.39, т.е. исходное число будет округлено, а к дробной части числа 2.5 наоборот, будет приписан еще один знак.

```

57.39 □ 2.50

```

ВНИМАНИЕ! В случае, если будет задан неверный формат числа, машина выдаст перед числом знак “%”.

Например, PRINT USING “###”;34578

PRINT USING “#. #”;15,7

```

%34578
%15.70

```

§ 7.2 Управление выводом на печать LOCATE.

Выше мы рассмотрели возможность выводить информацию как построчно, так и в

одной строке. Оператор LOCATE позволяет вывести данные в любом месте экрана.

ОБЩИЙ ВИД:

LOCATE PY, PX

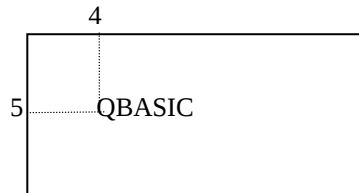
где PY — номер строки ($1 \leq PY \leq 24$),

PX — номер столбца ($1 \leq PX \leq 80$).

Оператор LOCATE устанавливает курсор для печати в строку PX и столбец PY.

Например, LOCATE 5,4: PRINT "QBASIC"

Результат:



Задача 1.

Вывести в третьей, четвертой и пятой строках, в третьем, шестом и девятом столбцах слова “июнь”, “июль”, “август”.

CLS

LOCATE 3,3 :PRINT “ИЮНЬ”

LOCATE 4,6 :PRINT “ИЮЛЬ”

LOCATE 5,9 :PRINT “ АВГУСТ”

END

Ввод данных с клавиатуры INPUT.

Этот оператор служит для ввода числовой и текстовой информации в память машины.

ОБЩИЙ ВИД :

INPUT список ввода

В списке ввода должны быть указаны имена величин, значения которых будут вводиться. Если вводится символьная величина, то она должна быть заключена в кавычки в том случае, если она содержит знаки препинания и пробелы.

Выполняя оператор INPUT, компьютер в качестве приглашения для ввода информации выдает на экран знак вопроса. В ответ пользователь должен набрать на клавиатуре необходимые данные и нажать {ENTER}, после чего набранная информация записывается в переменную памяти с указанным в операторе именем. В одном и том же операторе можно вводить значения нескольких величин.

Например, INPUT A, B

Выполняя этот оператор, компьютер выведет на экран знак “?” и будет ожидать ввода. Введем с клавиатуры, допустим, числа 5 и 7 через запятую и нажмем клавишу ENTER. Число 5 поступит в переменную A, а число 7 — в B.

Если программа содержит несколько операторов INPUT, то лучше если в оператор будет вставлена строковая константа, поясняющая список ввода. Строковая константа должна быть заключена в кавычки и отделена от списка ввода точкой с запятой.

Например, INPUT “Введите два числа”; A, B

После того, как компьютер прочитает этот оператор, он выведет на экран строковую константу и, поставив после нее знак вопроса, будет ожидать ввода:

Задача 1.

Подсчитать среднее арифметическое трех чисел, введенных с клавиатуры.

CLS

REM ввод значений в переменных A, B, C

INPUT “Введите первое число-”; A

INPUT “Введите второе число-”; B

INPUT “Введите третье число-”; C

REM подсчет ср.арифм. и вывод на печать

SA=(A+B+C)/3

PRINT “Среднее арифметическое равно ”;SA

END

При значениях A=1, B=2, C=3 получим следующий результат:

Введите первое число-?1↵
Введите второе число- ?2↵
Введите третье число- ?3↵
Среднее арифметическое равно 2

Задача 2.
Разместите введенное с клавиатуры слово в 20 столбце и в 10 строке текстового экрана.

Решение:

```
INPUT " "; A$  
LOCATE 10,20: PRINT A$  
END
```

Операторы DATA и READ .

Задание значений величин в программе может осуществляться тремя способами:

- 1) присвоение с помощью оператора присваивания,
- 2) ввод значений с клавиатуры с помощью оператора INPUT,
- 3) программное задание блока данных с последующим считыванием.

С первыми двумя способами мы ознакомились выше. Последний способ применяется в тех случаях, когда объем данных достаточно велик, а вводить их с клавиатуры неудобно. Для этого в программе создают блок данных — список констант, записанных в специальном операторе DATA.

ОБЩИЙ ВИД:

DATA список констант

где в списке констант — числовые или символьные константы, отделяемые друг от друга запятыми.

Строковая константа должна быть заключена в кавычки, если она начинается или заканчивается значащими пробелами, либо она содержит какие-либо знаки препинания. При этом в заключенной в кавычки строковой константе запрещено использовать кавычки, т.е. внутри самой константы кавычек быть не должно.

Например, DATA 5, 7.3, QBASIC, " PROGRAM:"

Оператор DATA относится к невыполняемым операторам, т.е. может быть записан в любом месте программы. В одной программе может быть несколько операторов DATA.

Значения констант из блока данных присваиваются переменным с помощью оператора READ.

ОБЩИЙ ВИД :

READ список переменных

где в списке величин перечисляются имена тех переменных, которым должны быть присвоены значения.

Оператор READ присваивает первое значение из блока данных первой переменной списка величин, второе — второй переменной и т.д.

Если в дальнейшем в программе встретится еще один оператор READ, то считывание данных будет происходить со следующего свободного, т.е. не считанного, значения из DATA.

Например, DATA 1,2,3,4,5
READ A, B, C
READ X, Y

В результате в переменным A, B, C, D, X, Y будут помещены значения:

A	B	C	X	Y
1	2	3	4	5

При формировании списка данных в операторе DATA и списка переменных в операторе READ необходимо следить за соответствием типов переменных и данных. Допускается считывание числовой константы как символьной.

Например, DATA 3, "Дом №5", 34.4
READ A\$, B\$, C

В результате в переменным A\$, B\$, C будут помещены значения:

A\$	B\$	C
3	Дом№5	34.4

В этом случае число 3 будет восприниматься как символьное выражение.

ВНИМАНИЕ!

Число переменных в операторах READ не должно превышать количество констант в списке операторов DATA , иначе компьютер выдаст сообщение об ошибке “ВНЕ ДАННЫХ”. Список констант в DATA не может содержать арифметических выражений.

Для того, чтобы начать считывание данных из блока сначала, используют оператор RESTORE. RESTORE возвращает указатель с текущего значения на начало списка.

Например:

ПРОГРАММА.

DATA 10,20,30

DATA 1,2,3

READ A1,A2

RESTORE

READ B1,B2,B3,B4

READ B5,C\$

ПЕРЕМЕННЫЕ

A1

A2

10

20

B1

B2

B3

B4

10

20

30

1

B5

C\$

2

3