# 1. программирование линейных алгоритмов

*Цель работы:* изучение правил записи констант, переменных, типов данных, выражений, операторов присваивания, операторов ввода-вывода и структуры программы.

## Методические указания

При выполнении программы происходит обработка данных. Данные в программировании называют величинами. Величины, значения которых могут изменяться в процессе выполнения программы, называют переменными. Значение каждой переменной хранится в определенном участке памяти компьютера.

Каждая переменная характеризуется именем, типом и значением.

Имя переменной (идентификатор) всегда должно начинаться с латинской буквы, после которой могут следовать несколько латинских букв, цифр либо символ подчеркивания «\_», записанных без пробелов. Например: A, B1, sum, Name, Pr\_3.

Тип переменной определяет диапазон допустимых значений.

Все переменные, используемые в программе, должны быть описаны в разделе описаний. Например:

var

A, B: integer;

X:real;

Text: string;

Переменная не имеет какого-либо конкретного значения до тех пор, пока компьютеру не будет дано точное предписание, поместить что-либо определенное в соответствующую ячейку памяти.

На Паскале такого рода предписание обычно выражается командой присваивания, имеющей вид:

имя\_переменной := выражение;

Оператор присваивания (:=) предписывает выполнить выражение, заданное в его правой части, и присвоить результат переменной, идентификатор которой расположен в левой части. Переменная и выражение должны быть совместимы по типу.

Оператор присваивания выполняется следующим образом: сначала вычисляется выражение в правой части присваивания, а затем его значение присваивается переменной, указанной в левой части оператора.

Например, для оператора

Rezult:=A div В;

сначала выполняется целочисленное деление значения переменной А на значение переменной В, а затем результат присваивается переменной Rezult.

Примеры применения оператора присваивания:

A:=22;

B:=A+6;

X:=a/10;

text:=’Privet!’;

Пример, демонстрирующий работу команды присваивания.

program primer;

var

a:integer;

begin

a:=5; {переменной А присваивается значение 5 – исходное значение}

writeln(‘a=’,a); {вывод на экран монитора значения переменной а}

a:=2\*a; {значение переменной А увеличивается в 2 раза}

writeln(‘a=’,a); {вывод на экран монитора промежуточного значения переменной а}

a:=a+1; {значение переменной А увеличивается на 1}

writeln(‘a=’,a); {вывод на экран монитора значения переменной а - результат}

end.

В результате выполнения программы на экране в «окне вывода» появится следующая информация об изменении значений переменной А:

a=5

a=10

a=11

Важно помнить: в результате выполнения команды присваивания предыдущее значение переменной стирается.

Для выполнения операций ввода-вывода служат четыре процедуры: Read, Readln, Write, WriteLn.

Процедура чтения Read обеспечивает ввод числовых данных, символов, строк и т.д. для последующей их обработки программой.

Формат процедуры Read: Read (x1, x2, …, xn);

где x1, x2, …, xn- переменные допустимых типов данных.

Значения x1, x2, …, xn набираются минимум через один пробел на клавиатуре и высвечиваются на экране. После набора данных для одной процедуры Read нажимается клавиша ввода Enter.

Значения переменных должны вводится в строгом соответствии с синтаксисом языка Паскаль. Если соответствие нарушено (например, х1 имеет тип Integer, а при вводе набирается значение типа Char), то возникают ошибки ввода-вывода. Сообщение об ошибке имеет вид: I/O error XX, где ХХ - код ошибки.

Процедура чтения ReadLn аналогична процедуре Read, единственное отличие заключается в том, что ее выполнения курсор автоматически перейдет на новую строку.

Процедура записи Write производит вывод числовых данных, символов, строк, булевских значений.

Формат процедуры Write: Write (<список вывода>);

где <список вывода>- последовательность переменных, констант, математических выражений, перечисляемых через запятую.

Процедура записи WriteLn аналогична процедуре Write, единственное отличие заключается в том, что после вывода последнего в списке значения для одной процедуры WriteLn данные для следующей процедуры WriteLn будут выводиться с начала новой строки.

Процедуры вывода допускают использование указания о ширине поля, отводимого под значение в явном виде:

WRITE (Y:m:n,X:k:l,...);

WRITELN (Y:m:n:,X:k: I,...);

где m и k- количество позиций, отведенных под запись значения переменных Y и X соответственно;

n и I - количество позиций, отведенных под запись дробной части чисел Y и X.

**Пример 1.** Составить программу расчета значения функции.

Z = |cos х4 – 3 tg х2 |+0.8 sin(yх2)+ 10 при любых значениях х и у.

Результат вывести в виде: при х= и у=… z=…

Используемые переменные: x, y -аргументы, z – значение функции

Program pr1;

var

x,y,z: real;

begin

writeln('введите X Y'); {вывод строки подсказки}

readln (x,y); {ввод аргументов x и y}

z:=abs(cos(sqr(x)\*sqr(x)-3\*sin(sqr(x))/cos(sqr(x))))+0.8\*sin(y\*sqr(x))+10;

writeln('при x=',x:8:2,' y=',y:8:2,' z=',z:8:2); {вывод результата}

End.

**Пример 2**. Найти площадь круга и длину окружности.

Используемые переменные: r - радиус, d – длина окружности, s – площадь круга

program pr2;

var

d,r,s:real;

begin

write('введите радиус окружности ');

readln (r); {ввод радиуса}

d:= 2\*Pi\*r; {вычисление длины окружности}

s:=Pi\*sqr(r); { вычисление площади круга}

writeln('длина окружности= ',d:4:2); {вывод результата}

writeln('площадь окружности= ',s:4:2);

end.

**Пример 3**. Вычисление суммы цифр введенного натурального двузначного числа.

Используемые переменные: n - двузначное число, a, b – цифры числа.

program pr3;

var

n, a, b: integer;

begin

write('n= '); readln(n); {ввод исходного двузначного числа}

a:=n div 10; {1-я цифра}

b:=n mod 10; {2-я цифра}

writeln('сумма = ', a+b); {вывод результата}

end.

**Пример 4**. Введенное натуральное 4-значное число изменить так, чтобы 2 и 3 цифры поменялись местами.

Четырехзначное число N можно представить в виде суммы разрядных слагаемых: N=n1\*1000+n2\*100+n3\*10+n4, где n1, n2, n3, n4 – цифры соответствующих разрядов. Например, 3562=3\*1000+5\*100+6\*10+2

Чтобы во введенном числе N поменять цифры местами, нужно выделить каждую цифру и записать число в виде N=n1\*1000+n3\*100+n2\*10+n4

Используемые переменные: N – вводимое четырехзначное число, n1, n2, n3, n4 – цифры.

program pr5;

var

N, n1, n2, n3, n4:integer;

begin

write('введите n ');

readln (n); {ввод исходного 4-значного числа}

n1=N div 1000; {1-я цифра числа}

n2:=N div 100 mod 10; {2-я цифра числа }

n3:=N div 10 mod 10; {3-я цифра числа }

n4:=N mod 10; {4-я цифра числа}

n:= n1\*1000+n3\*100+n2\*10+n4; {получение числа в виде суммы разрядных слагаемых}

writeln('результат ', n); {вывод результата}

end.

## Упражнения для самостоятельной работы

1. Вводится вещественное число а. Не пользуясь никакими арифметическими операциями, кроме умножения, получить a9 за четыре операции.

2. Найти сумму цифр введенного 4-значного числа.

3. Вводятся два числа: 2-значное и 3-значное. Получить 5-значное число, состоящее из цифр исходных чисел. Например, 25 и 137 -> 25137.

4. Составить программу, которая переводит значение температуры из шкалы Цельсия в шкалу Фаренгейта по формуле   
TF = 1.8TC + 32.

5. Составить программу для перевода суммы из долларов в рубли. Вводится текущий курс доллара и сумма в долларах. Результат должен выводиться в денежном формате, например, 345 руб. 50 коп.

## Задание к лабораторной работе

Выполнение задания состоит из следующих этапов:

1. изучить материал по заданной теме;
2. составить линейный алгоритм и программу, в которой вводятся исходные данные, вычисляются заданные арифметические выражения и выводятся на экран дисплея результаты вычислений.

**Вариант 1**

; 

**Вариант 2**

; 

**Вариант 3**

; 

**Вариант 4**

; 

**Вариант 5**

; ; 

**Вариант 6**

; ; 

**Вариант 7**

; 

**Вариант 8**

; 

**Вариант 9**

; ; ; 

**Вариант 10**

; ; 

**Вариант 11**

; 

**Вариант 12**

; 

**Вариант 13**

; 

**Вариант 14**

; 

**Вариант 15**

; 