

Операторы языка Pascal

read

begin

writeln

if - then - else

for - to - do

```
X:= 4;
```

```
y:=3;
```

```
writeln('Значение выражения =',x+sqr(y))
```

Оператор присваивания

Оператор присваивания

предназначен для присваивания переменной значения выражения.

Оператор обозначается символом

:=

в левой части указывается имя переменной,

в правой – вычисляемое выражение.

Примеры использования оператора присваивания:

С переменной целого типа

`x := 6;`

Вещественного

`y := sin(Pi/3);`

Символьного

`c := '7';`

Булевского

`test := false;`

Операторы ввода-вывода

Read(<Список ввода>);

Readln(<Список ввода>);

ОПЕРАТОРЫ
ВВОДА

Write(<Список вывода>);

Writeln(<Список вывода>);

ОПЕРАТОРЫ
ВЫВОДА

Составной оператор

Операторы, входящие в составной оператор, разделяются символом «;», в начале ставится служебное слово *begin*, в конце – *end*.

Составной оператор применяется в тех случаях, когда синтаксис языка допускает использование только одного оператора, а смысл требует выполнения нескольких действий.

Пример составного оператора

Begin

X:=10; y:=5; z:=1;

***Writeln('g=', sqr(x) + sqr(y) +
sqr(z)***

end;

Фактически весь раздел операторов, обрамленный словами ***begin ... end*** представляет собой один составной оператор.

ВНИМАНИЕ! Перед служебным словом ***end*** разделитель (символ «;») не ставится.

Объявление массива

Var Имя переменной-массива : **Array** [
Диапазон индексов] **Of** Тип
элементов;

Одномерный массив

**Var T : Array [1..40] Of
Real;**

Двумерный массив

**Var A : Array [1..20, 1..30]
Of Integer;**

Трёхмерный массив

Var A : Array

[1..20, 1..30, 1..20] Of

Integer;

Оператор безусловного перехода

Goto метка

СМЫСЛОВОЕ значение в
программе

«перейти к ...»

Условный оператор

1. **if <условие>
then <оператор1>
else <оператор2>;**

Полная
форма
условного
оператора

2. **if <условие> then
<оператор1>;**

Примеры

Рассмотрим определение функции у:

$$y = \begin{cases} x^2 - 1, & \text{если } x < 0, \\ x + 5, & \text{если } x \geq 0 \end{cases}$$

1 сп.) *if* $x < 0$
 then $y := x * x - 1$
 else $y := x + 5;$

2 сп.) $b := x < 0;$
 if b
 then $y := x * x - 1$
 else $y := x + 5;$

Во 2-м способе описывается логическая переменная b , значение которой будет равно *true*, если $x < 0$, и *false* – иначе.

Пример сокращенной формы условного оператора:

```
If not ((x<3)) then y:= sqr(x);
```

При использовании вложенных условных операторов может возникнуть неоднозначность, например:

```
if <условие1>  
  then if <условие2>  
        then <оператор1>  
        else <оператор2>
```

В данном случае необходимо помнить, что служебное слово **else** всегда ассоциируется с ближайшим по тексту программы **if**, которое еще не связано с **else**. А чтобы такая двусмысленность не возникала, лучше всего пользоваться составным оператором:

```
if <условие1>  
  then  
    begin  
      if <условие2>  
        then <оператор1>  
        else <оператор2>  
      end;
```

ИЛИ

```
if <условие1>  
  then  
    begin  
      if <условие2>  
        then <оператор1>  
      end;  
    else <оператор2>
```

Задача 1. Написать программу, проверяющую принадлежит ли число, введенное с клавиатуры, интервалу (1;5)

```
program Z1;  
var x : integer;  
begin  
    writeln ('Введите число x');  
    readln (x);  
    if (x>0) and (x<5)  
        then writeln (x, 'принадлежит (1;5)')  
        else writeln (x, 'не принадлежит (1;5)')  
end.
```

Операторы цикла

Оператор цикла

- с предусловием;
- с постусловием;
- с параметром.

Оператор цикла с предусловием

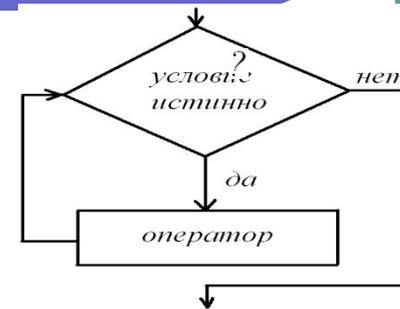
while <условие> **do** <оператор>;

где **while** (пока), **do** (делать) – зарезервированные слова

Задача 1. Вычисление факториала из числа n. (2 способа)

```
Program Factorial1;  
Const n=5;  
Var i,fact : integer;  
Begin  
  i:=1; fact:=1;  
  while i<=n do {заголовок цикла}  
    begin {начало цикла}  
      fact := fact * i;  
      i := i+1  
    end; {конец цикла}  
  writeln ('факториал', n, '=', fact:10);  
End.
```

```
Program Factorial2;  
Const n=5;  
Var fact : integer;  
Begin  
  fact:=1;  
  while n>0 do {заголовок цикла}  
    begin {начало цикла}  
      fact := fact * n;  
      n:=n-1  
    end; {конец цикла}  
  writeln ('факториал', n, '=', fact:10);  
End.
```



Задача 2. Дано натуральное число. Найти количество четных цифр числа.

```
Program Z2;  
Uses crt;  
Var m,n : longint; k,p: integer;  
Begin  
  clrscr;  
  write('n='); readln(n);  
  m:=n; k:=0;  
  while m<>0 do begin  
    p:=m mod 10;  
    if p mod 2=0 then  
      k:=k+1; m:= div 10;  
    end;  
  writeln('Количество четных цифр K=',K);  
  readln;  
end.
```

Оператор цикла с постусловием

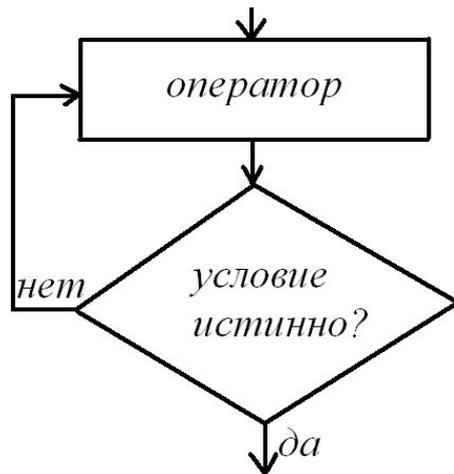
Repeat

<оператор1>;

<оператор2>;

Until <условие>;

где Repeat (повторять до тех пор), Until (пока не будет выполнено условие) – зарезервированные слова



Задача. Написать программу для игры «Угадайка».

Program Uгадайка;

Var x,y,n : word;

Begin

randomize; {инициализация датчика}

x:= random(100);

n:=0;

repeat

writeln ('Введите число');

read (y);

n:=n+1;

if x>y then writeln ('больше');

if x<y then writeln ('меньше');

until x=y;

writeln('Вы угадали число за ', n, ' попыток');

end/.

Оператор цикла с параметром

Используют этот оператор, когда количество повторений заранее известно.

for <параметр> := <начальное значение> **to** <конечное значение> **do** <оператор>;

где **for** (для), **to** (до) **do** (выполнить) – зарезервированные слова.

Задача. Вычислить n-ую степень числа 2.

```
Program stepen _2_n;  
Var n,x,k:word;  
Begin  
  writeln ('Введите n:');  
  read (n);  
  for k:=1 to n do  
    x:=x*2;  
  Writeln ('2 в степени n равно ',x);  
end.
```

Вложенные циклы

Вложенный цикл – цикл, содержащий в себе несколько простых циклов. Различают внешний и внутренние циклы. Внутренние циклы должны записываться таким образом, чтобы они полностью находились внутри других циклов.

Пример 1. (Печатает 10 строк пар чисел)

```
Program Primer1;  
Var l,k : integer;  
Begin  
  for i:=1 to 10 do  
  begin  
    for k:=1 to 3 do  
      writeln ('i=', l , 'k=' , k);  
    writeln;  
  end;  
  readln;  
end.
```

Пример 2. Найти все такие тройки натуральных чисел x, y, z из интервала от 1 до 20, для которых выполняется равенство: $x^2 - y = z^2$.

```
Program Primer2;  
Uses crt;  
Var x, y, z : integer;  
Begin  
  clrscr;  
  for x:=1 to 20 do  
    for y:=1 to 20 do  
      for z:=1 to 20 do  
        if x*x - y = z*z then  
          writeln (x,' ', y, ' ', z);  
        readln;  
      end.  
    end.  
  end.
```