

Ветвления

Очень часто встречаются ситуации, когда требуется выбрать между двумя или более вариантами действий в зависимости от заданного условия. Такая алгоритмическая конструкция называется ветвлением.

Операторы условия и перехода

Строки программы на Паскале не нумеруются. Отдельные строки в программе могут иметь метки, к которым можно переходить.

Метки должны быть описаны в программе с помощью ключевого слова

```
label
```

```
N1, N2 ... ;
```

где N1, N2, ... - метки.

Метками могут быть идентификаторы или целые числа (положительные) в диапазоне 0 ÷ 9999.

Оператор перехода по метке:

```
goto N;
```

Где N - метка.

Фрагмент программы с использованием меток:

```
program pr;
```

```
label 3;
```

```
var
```

```
x, y: real;
```

```
begin
```

```
3: readln (x, y);
```

```
...
```

```
go to 3;
```

```
...
```

```
end.
```

Логический оператор

```
if < условия > then P1 [else P2 ];
```

(если) (то) (иначе)

[else P2] - не обязательная часть оператора.

Где <условие> - логическое выражение;

P1, P2 - простые или составные операторы.

По этому оператору:

если <условие> - "истинно" (true), то выполняется P1, если - "ложно" (false), то выполняется P2. Перед структурой else «;» - не ставится.

Если else - отсутствует и <условие> - "ложно", то управление передается следующему оператору.

Приведем примеры использования оператора:

1. наибольшее из значений двух переменных вывести на печать:

```
if a>b then writeln('a=',a) else writeln(' b=',b);
```

2. в зависимости от значения переменной, вычислить значение функции по одной из формул:

```
If x>=0 then y: = sin(x) else y: = - sin(x);
```

3. в зависимости от значения переменной, выполнить определенные действия:

```
if a<0 then  
begin  
b:=2*a;  
writeln(b);  
end  
else  
begin  
b:=a/2;  
goto 3;  
end;
```

Операции отношения

Операция	Название
=	Равно
<>	Не равно
>	Больше
<	Меньше
>=	Больше или равно
<=	Меньше или равно

Логические операции

Операция	Название
and	И
or	Или

В операторе if можно одновременно проверять несколько условий.

Например:

```
if (a>b)and(a>c) then writeln('a - max');
```

При проверке нескольких условий в одном операторе, каждое условие записывается в скобках.

Паскаль допускает вложенность операторов if.

Например:

```
if n>0 then
```

```
  if (m div n)>n then m:= m - n
```

```
  else m:= m + n;
```

Структура else - всегда относится к ближайшему оператору if.

Если $n>0$ и $(m \text{ div } n)>n$ - будет выполнено $m:= m-n$.

Если $n>0$, но $(m \text{ div } n) \leq n$ - будет выполнено $m:= m + n$.

Если $n \leq 0$ - переход к следующему оператору.

Задача.

Найти максимальное число из трех вещественных чисел.

```
program max;
var
a, b, c: real;
begin
writeln ('введите a, b, c'); readln (a, b, c);
if a>=b then
if a>=c then writeln ('max - a')
else writeln ('max - c')
else
if b<=c then writeln ('max - b')
else writeln ('max - c') ;
readln;
end.
```

Второй способ решения этой задачи:

```
program max_1;
var
a, b, c: real;
begin
writeln ('введите a, b, c');
readln (a, b, c);
if (a>=b) and (a>=c) then writeln ('max-a');
if (b>=a) and (b>=c) then writeln ('max-b');
if (c>=a) and (c>=b) then writeln ('max-c');
```

readln;

end.

Если $a=b=c$ - то все три оператора if будут выполнены.

Оператор выбора

Этот оператор предназначен для замены конструкций из вложенных if.

Структура:

case N of

N1: P1;

N2: P2;

NN: PN;

[else P;]

end;

[else P;] - необязательная часть оператора.

где N - целочисленная переменная, или выражение целого типа;

N1, N2, ..., NN - возможные значения переменной N;

P, P1, P2, ..., PN - простые или составные операторы.

По этому оператору:

если значение $N = N1$, то выполняется P1 (после чего управление передается оператору, следующему за оператором case...of);

если значение $N = N2$, то выполняется P2, (после чего управление передается оператору, следующему за оператором case...of);

...

если N не принимает ни одного из перечисленных значений, управление передается оператору P;

если структура else - отсутствует и N - не принимает ни одного из перечисленных значений, управление передается следующему за

case...of оператору.

Например:

case N of

1, 2, 5: writeln ('a');

7 .. 10: writeln ('b');

end;

1, 2, 5 - перечисляемые значения разделяются точкой.

7 .. 10 - интервал задается начальными и конечными значениями через двоеточие.

Пример программы разветвленной структуры

Задача.

1. Вычислить значение функции:

$$y = \begin{cases} \sin x, & 0 < x < 90 \\ 0, & x \leq 0 \end{cases}$$

Значение аргумента x в градусах ввести с клавиатуры.

2. Таблица идентификаторов

№	Наименование переменной	Обозначения в программе
1	Функция	y
2	Аргумент	x

3. Листинг программы.

```

program lab2;
uses
crt;
var
x, y: real;
begin
clrscr;
writeln ('Введите x');
readln (x);
if x>90 then writeln('функция не определена')
else
{составной оператор}
begin
if x<0 then y:= 0
else y:= sin (x*pi/180);
writeln ('y = ',y:8:3);
end;
readln;
end.

```