

Операторы в языке Паскаль



Условные операторы



- Бывает, что в процессе выполнения программы требуется реализовать разный набор команд в зависимости от произошедших до этого событий. В языках программирования это достигается с помощью специальных конструкций – *условных операторов*.
- Чаще всего в качестве условного оператора в языках программирования используется конструкция **if-else** или ее сокращенный вариант **if**. Также существует оператор выбора **case**, который имеет более специфичное применение.

Полное ветвление



Например, программа должна определять, ввел пользователь четное или нечетное число, и выводить на экран сообщение. Тогда программный код на языке Pascal может быть таким:

```
var n: integer;  
begin write ('Введите целое число: ');  
  readln (n);  
if n mod 2 = 0 then write ('Оно четное.')  
else write ('Оно нечетное.');
```

readln
end.

Неполное ветвление



В примере ниже, если переменная имеет значение меньше нуля, то ее значение изменяется (находится модуль числа). Если же значение переменной изначально больше нуля, то блок кода при операторе **if** вообще не выполняется, т.к. не соблюдено условие ($n < 0$).

```
var n: integer;  
begin write ('Введите целое число: ');  
readln (n);  
  if n < 0 then n := abs (n);  
write (n);  
readln  
end.
```

Циклы в Паскале



При решении задач может возникнуть необходимость повторить одни и те же действия несколько или множество раз. В программировании блоки кода, которые требуется повторять не единожды, оборачиваются в специальные конструкции – *циклы*. У циклов выделяют заголовок и тело. Заголовок определяет, до каких пор или сколько раз тело цикла будет выполняться. Тело содержит выражения, которые выполняются, если в заголовке цикла выражение вернуло логическую истину (True, не ноль). После того как достигнута последняя инструкция тела, поток выполнения снова возвращается к заголовку цикла. Снова проверяется условие выполнения цикла. В зависимости от результата тело цикла либо повторяется, либо поток выполнения переходит к следующему выражению после всего цикла.

В языке программирования Паскаль существует три вида циклических конструкций.

Цикл for



- Часто цикл **for** называют циклом со счетчиком. Этот цикл используется, когда число повторений не связано с тем, что происходит в теле цикла. Т.е. количество повторений может быть вычислено заранее (хотя оно не вычисляется).
- В заголовке цикла указываются два значения. Первое значение присваивается так называемой переменной-счетчику, от этого значения начинается отсчет количества итераций (повторений). Отсчет идет всегда с шагом равным единице. Второе значение указывает, при каком значении счетчика цикл должен остановиться. Другими словами, количество итераций цикла определяется разностью между вторым и первым значением плюс единица. В Pascal тело цикла не должно содержать выражений, изменяющих счетчик.
- Цикл **for** существует в двух формах:
- **for** счетчик:=значение **to** конечное_значение **do** тело_цикла;
- **for** счетчик:=значение **downto** конечное_значение **do** тело_цикла;

- *Счетчик* – это переменная любого из перечисляемых типов (целого, булевого, символьного, диапазонного, перечисления). Начальные и конечные значения могут быть представлены не только значениями, но и выражениями, возвращающими совместимые с типом счетчика типы данных. Если между начальным и конечным выражением указано служебное слово **to**, то на каждом шаге цикла значение параметра будет увеличиваться на единицу. Если же указано **downto**, то значение параметра будет уменьшаться на единицу.
- Количество итераций цикла **for** известно именно до его выполнения, но не до выполнения всей программы. Так в примере ниже, количество выполнений цикла определяется пользователем. Значение присваивается переменной, а затем используется в заголовке цикла. Но когда оно используется, циклу уже точно известно, сколько раз надо выполниться.

```
var i, n: integer;
```

```
begin write ('Количество знаков: ');
```

```
readln (n);
```

```
for i := 1 to n do
```

```
write ('(*) ');
```

```
readln
```

```
end.
```

Цикл `while`



- Цикл **`while`** является циклом с предусловием. В заголовке цикла находится логическое выражение. Если оно возвращает **`true`**, то тело цикла выполняется, если **`false`** – то нет.
- Когда тело цикла было выполнено, то ход программы снова возвращается в заголовок цикла. Условие выполнения тела снова проверяется (находится значение логического выражения). Тело цикла выполнится столько раз, сколько раз логическое выражение верно **`true`**.

```
var i, n: integer;  
begin  
  write ('Количество знаков: ');  
  readln (n);  
  i := 1;  
while i <= n do  
  begin  
    write ('(*) ');  
    i := i + 1  
  end;  
  readln  
end.
```


Операторы **break** и **continue**

Операторы **break** и **continue** используются для того, чтобы прервать ход выполнения цикла.

Оператор **break** выполняет полный выход из цикла, т.е. все возможные итерации цикла прерываются. Оператор **continue** прерывает только текущую итерацию.

В примере у пользователя несколько раз запрашивается число, только в том случае, если он не вводит ноль.

```
var num: real; i: integer;  
begin  
  for i := 1 to 5 do begin  
    write ('Введите число: ');  
    readln (num);  
    if num = 0 then break;  
    writeln (num) end;  
  readln  
end.
```

В примере запрашиваются 5 чисел и суммируются только положительные из них.

```
var num, sum: real; i: integer;  
begin sum := 0;  
for i := 1 to 5 do  
begin  
write ('Введите число: ');  
readln (num);  
if num < 0 then continue;  
sum := sum + num  
end;  
write (sum:10:2);  
readln  
end.
```