

Тема урока:

**Организация
циклов на языке
Pascal**

Цели урока:



1. Рассмотреть особенности организации циклов на языке Pascal;
2. Познакомится с правилами организации различных циклов на языке Pascal.

Организация циклов на языке Pascal

Часто в программе требуется повторить какой-то оператор или группу операторов несколько раз. В Pascal существует три вида циклов, согласно трем циклическим алгоритмическим структурам



Цикл с предварительным условием (цикл «Тока»)

```
i:=<начальное условие цикла>;  
while <конечное условие цикла > do  
begin  
<тело цикла>;  
i:=i+h;  
end;
```

конечное условие цикла – логическое
выражение,

i – параметр цикла;

h – шаг цикла.

Пример:

```
i:=10;  
while i<=100 do  
begin  
  writeln (i);  
  i:=i+10;  
end;
```

Правила использования этого цикла

1. При каждом, в том числе и первом выполнении цикла, условие выхода из цикла проверяется, поэтому оно должно иметь конкретное значение.
2. Тело цикла должно содержать хотя бы один оператор, влияющий на окончание цикла.
3. Параметр цикла (i) – переменная порядкового типа.

Пример: Написать программу, которая позволяет вывести все цифры в порядке возрастания.

```
program cifri2;  
var i:integer;  
begin  
i:=0;  
while i<10 do (i<=9)  
begin  
writeln (i);  
i:=i+1;  
end;  
end.
```



Цикл с постусловием (цикл «До тех пор»)

i := <начальное условие цикла>;

repeat

<тело цикла>

i := **i** + **h**

until <условие выхода из цикла>;

конечное условие цикла –

логическое выражение,

i – параметр цикла;

h – шаг цикла.

Пример:

```
i := 15;
```

```
repeat
```

```
    writeln
```

```
    (sqr(i));
```

```
    i := i + 1;
```

```
until
```

Правила использования цикла с постусловием

1. После каждого выполнения тела цикла условие выхода из цикла проверяется, поэтому оно должно иметь конкретное значение.
2. Тело цикла должно содержать хотя бы один оператор, влияющий на окончание цикла.
3. Параметр цикла (*i*) – переменная порядкового типа.
4. Ключевые слова `repeat` и `until` сами по себе являются операционными скобками, поэтому `begin` и `end` не нужны.

Пример: Написать программу, которая позволяет вывести все цифры в порядке убывания

```
program cifri3;  
var i:integer;  
begin  
  i:=9;  
  repeat  
    writeln (i);  
    i:=i-1;  
  until i<0;  
end.
```



Цикл с параметром («цикл «Для каждого»)

downto

```
for <параметр>:=<начальное значение цикла> to  
<конечное значение цикла> do  
begin  
<тело цикла>;  
end;
```

Параметром может быть любой порядковый тип
(**integer, char, boolean**).

Пример: **for c:='Я' downto 'А' do writeln(c);** - на экран будет выведен русский алфавит в обратном порядке.

Правила организации данного цикла

1. Параметр цикла, его начальное и конечное значения должны быть одного и того же порядкового типа.
2. Если начальное значение цикла меньше, чем конечное, то нужно использовать слово **to**, в данном случае шаг цикла плюс единица (+1), если же конечное значение цикла меньше начального, то необходимо использовать слово **downto** и шаг цикла будет минус единица (-1).
3. Нельзя в теле цикла изменять значение параметра, начальное и конечное значения цикла, так же нельзя менять шаг цикла. Это может привести к зависанию программы.

Пример: Написать программу, которая позволяет вывести все целые числа на отрезке **[a,b]**.

```
program cifri4;
```

```
var i,a,b:integer;
```

```
begin
```

```
writeln('Введите a и b (причем a<b): ');
```

```
readln(a,b);
```

```
for i:=a to b do writeln(i);
```

```
end.
```



Домашнее задание



1. Запишите в тетради тему занятия и сделайте краткий конспект (слайды 1, 4, 7 и 10).
2. Используя возможности программы PascalABC.NET протестируйте примеры программ, которые представлены в презентации (слайды 6, 9 и 12).
3. Выполните тестовое задание, пройдя по ссылке:

Примечание: Конспект и скриншоты проверки работоспособности программ высылать учителю не надо.