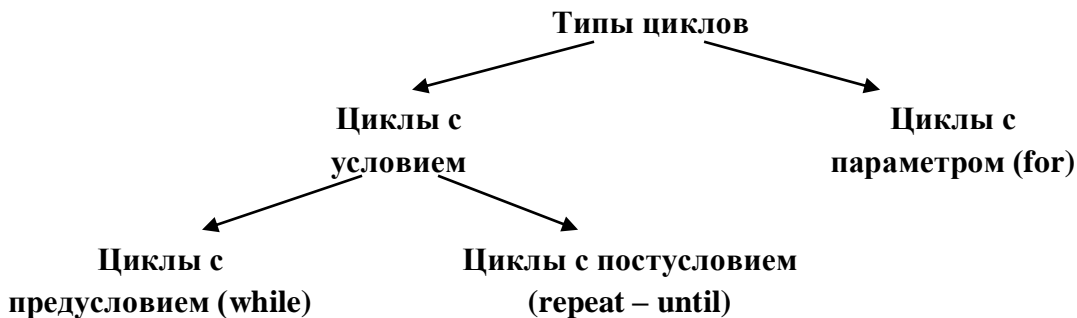


# Циклы в Pascal

**Цикл** – разновидность управляющей конструкции, которая предназначена для организации многократного повторения (исполнения) набора операторов (команд).

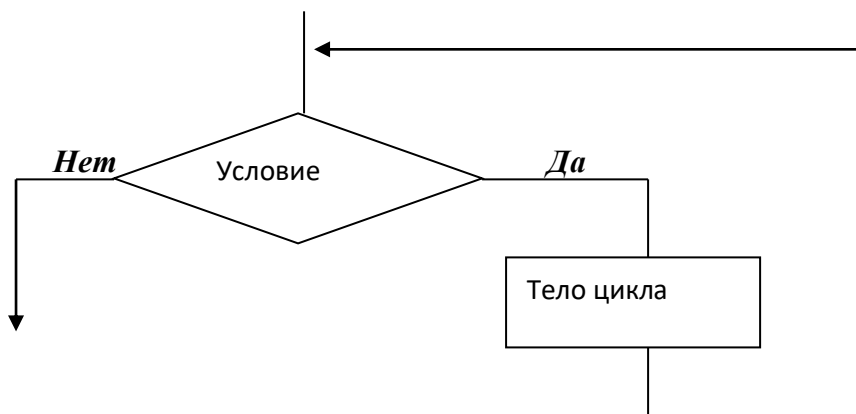
Все циклы делятся на две группы: **циклы с условием** и **циклы с параметром (for)**. Циклы с условием также можно разбить на две группы: **циклы с предусловием (while)** и **циклы с постусловием (repeat – until)**. Схематично разбиение на типы циклов можно изобразить следующим образом:



Рассмотрим **цикл с предусловием while**. Оператор цикла с предусловием выполняет действия заранее неизвестное число раз. В заголовке цикла находится некоторое логическое выражение (условие). Если условие истинно, то тело цикла выполняется, если ложно – то нет. Так как верность логического выражения проверяется в начале, тело цикла может не выполниться ни одного разу.

Когда тело цикла было выполнено, то ход программы снова возвращается в начало цикла. Условие выполнения тела снова проверяется (находится значение логического выражения). Тело цикла выполнится столько раз, сколько раз логическое выражение истинно. Поэтому очень важно в теле цикла предусмотреть изменение переменной, фигурирующей в условии цикла, таким образом, чтобы когда-нибудь обязательно наступала ситуация, когда логическое выражение окажется ложным. Иначе произойдет **зацикливание**.

Схематическое изображение работы цикла while (блок – схема) выглядит следующим образом:



**Структура цикла while:**

```
While <условие> do
  Begin
    <тело цикла>;
  End;
```

**Пример:** Вычислить сумму всех черных чисел до 50.

Program Number;

```

Var sum: integer;
    n: integer;
begin
    sum:=0;
    n:=2;
    while n <= 50 do
        begin
            sum:= sum + n;
            n:= n + 2;
        end;
    writeln ('Сумма равна: ',sum);
end.

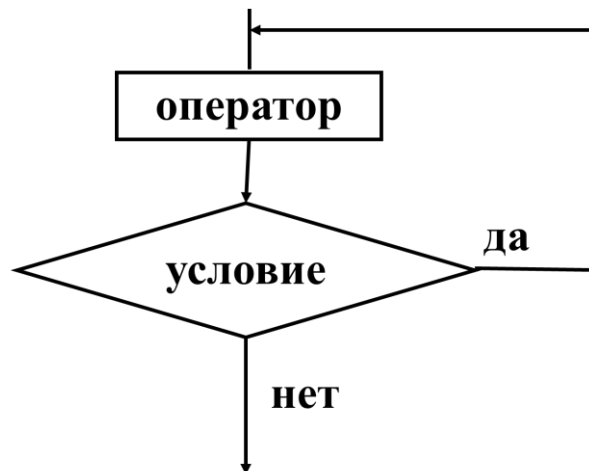
```

Следующий тип циклов, который мы рассмотрим, будет **цикл с постусловием repeat – until**. Этот оператор аналогичен оператору цикла с предусловием **while**, но отличается от него тем, что проверка условия производится после выполнения тела (операторов) цикла. Это обеспечивает его выполнение хотя бы один раз в отличие от других типов циклов. Причем, в отличие от цикла **while**, цикл выполняется до тех пор, пока логическое выражение (условие) ложно, а когда оно станет истинным, то произойдет выход из цикла.

Как и в цикле **while**, тело цикла выполнится столько раз, сколько раз логическое выражение ложно. Поэтому очень важно в теле цикла предусмотреть изменение переменной, фигурирующей в условии, таким образом, чтобы когда-нибудь обязательно наступала ситуация, когда логическое выражение окажется истинным. Иначе произойдет **зацикливание**.

Обратите внимание на то, что данный оператор цикла предполагает наличие нескольких операторов в теле цикла, то есть можно выполнять несколько действий, поэтому операторные скобки Begin и End не нужны.

Схематическое изображение работы цикла repeat – until (блок – схема) выглядит следующим образом:



**Структура цикла repeat – until:**

```

Repet
    <тело цикла>;
Until
    <условие>;

```

**Пример:** Определить сумму первой и последней цифр в числе.

Program Number;

```

Var a,b,c,d: integer;
begin
  writeln('введите число');
  readln(a);
  d:=a mod 10;
  repeat
    b:=a mod 10;
    a:=a div 10;
  until a=0;
  c:=d+b;
  writeln('Сумма первой и последней цифры равна: 'c);
end.

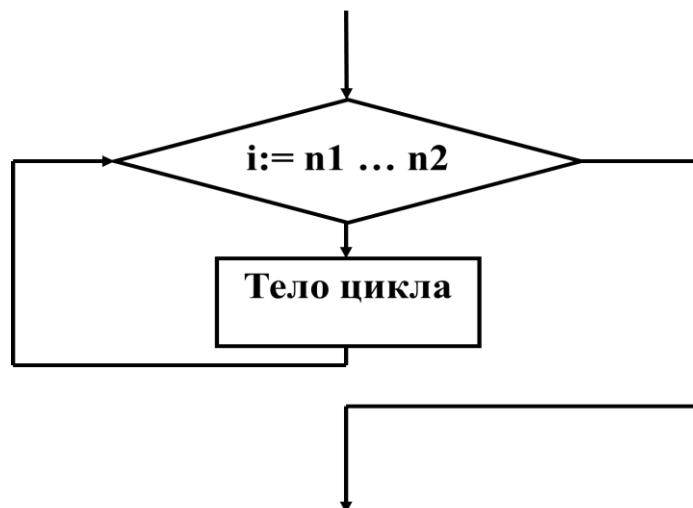
```

Последний тип циклов: **цикл с параметром 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.

**Счетчик** – это переменная любого типа (integer, real, и т. д.). 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.

В заголовке цикла указываются два значения. Первое значение присваивается переменной-счетчику (обычно *i* или *j*), 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 данном цикле, в отличие от циклов **while** и **repeat – until** (в них изменялось логическое выражение (условие) в цикле), в теле цикла не должны содержаться операторы, изменяющие счетчик цикла.

Схематическое изображение работы цикла с параметром **for** (блок – схема) выглядит следующим образом:



Структура цикла **for** (2 формы):

```

For i := n1 to n2 do
  Begin
    <тело цикла>;
  End;

```

```

For i := n1 downto n2 do
  Begin
    <тело цикла>;
  End;

```

**Пример:** Написать программу, которая вычисляет n-ую степень заданного целого числа.

*Program Number;*

*Var a, n, i, pr: integer;*

*begin*

*writeln ('Введите число');*

*readln (a);*

*writeln ('Введите степень числа');*

*readln (n);*

*pr:= 1;*

*for i:= 1 to n do*

*begin*

*pr:= pr \* a;*

*end;*

*writeln ('Степень числа равна ',pr);*

*end.*