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 вернет **true**. Поэтому очень важно в теле цикла предусмотреть изменение переменной, фигурирующей в заголовке цикла, таким образом, чтобы когда-нибудь обязательно наступала ситуация **false**. Иначе произойдет так называемое **зацикливание**, одна из самых неприятных ошибок в программировании.

```
var
    i, n: integer;

begin
    write ('Количество знаков: ');
    readln (n);

    i := 1;
    while i <= n do begin
        write ('(*) ');
        i := i + 1
    end;

readln
end.
```

Цикл repeat

Цикл **while** может не выполниться ни разу, если логическое выражение в заголовке сразу вернуло **false**. Однако такая ситуация не всегда может быть приемлемой. Бывает, что тело цикла должно выполниться хотя бы один раз, не зависимо оттого, что вернет логическое выражение. В таком случае используется цикл repeat – цикл с постусловием.

В цикле **repeat** логическое выражение стоит после тела цикла. Причем, в отличие от цикла **while**, здесь всё наоборот: в случае **true** происходит выход из цикла, в случае **false** – его повторение.

```
var
    i, n: integer;

begin
```

```
    write ('Количество знаков: ');
    readln (n);

    i := 1;
    repeat
        write ('(*) ');
        i := i + 1
    until i > n;

readln
end.
```

В примере, даже если n будет равно 0, одна звездочка все равно будет напечатана.

```
uses graphABC;

procedure Shar(xx,yy: integer);

begin

setpencolor(clblack);

setbrushcolor(clred);

circle(xx,yy,10);

setbrushcolor(clmaroon);

setpencolor(clmaroon);

circle(xx+2,yy+2,5);

end;


procedure Triangle(x,y,r: integer);

begin

setpencolor(clgreen);

line(x-r,y,x,y-(r*2));

line(x,y-(r*2),x+r,y);

line(x-r,y,x+r,y);

floodfill(x,y-2,clgreen);

Shar(x,y-(r div 2)-(r div 3));

Shar(x-(r div 4),y-(r div 4));

Shar(x+(r div 4),y-(r div 4));

Shar(x-r,y+10);
```

```
Shar(x+r,y+10);
end;

procedure Gir(x,y,r: integer);
var i: integer;
begin
i:=-r;
while i<=r do
    begin
    if (getpixel(x+i,y+i)=clgreen) then
     begin
     setbrushcolor(clrandom);
     setpencolor(clrandom);
     circle(x+i,y+i,5);
     end;
    inc(i,10);
    end;
end;

begin
SetWindowTitle('С Новым Годом!');
setwindowsize(400,500);
lockdrawing;
repeat
clearwindow;
Triangle(200,200,50);
Triangle(200,300,80);
Triangle(200,400,100);
setpencolor(clbrown);
```

```
setbrushcolor(clbrown);

rectangle(180,400,220,440);

Gir(200,200,70);

Gir(200,300,100);

Gir(200,400,150);

sleep(100);

redraw;

until false;


end.
```