



# ОПЕРАТОРЫ ЦИКЛА

*Автор: Юркова Татьяна  
Яковлевна*

*Учитель информатики*

*Г.Ростов-на-Дону МБОУ СОШ№97*



# СОДЕРЖАНИЕ

1. [Понятие цикла](#)
2. [Оператор цикла с параметром](#)
3. [Оператор цикла с предусловием](#)
4. [Оператор цикла с постусловием](#)
5. [Обобщение работы операторов цикла](#)



# Определение цикла

**Циклом** - называется последовательность действий, выполняемых многократно, каждый раз при новых значениях параметров.

Последовательность действий, выполняемых многократно называется – **телом цикла**.

Если заранее известно количество необходимых повторений, то цикл называется **арифметическим**.

Если количество повторений заранее неизвестно, то говорят об **итерационном** цикле.



# Циклы.

---

В **итерационных** циклах производится проверка некоторого условия, и в зависимости от результата этой проверки происходит либо выход из цикла, либо повторение выполнения тела цикла.

Если проверка условия производится перед выполнением блока операторов, то такой итерационный цикл называется циклом **с предусловием** (цикл "пока"), а если проверка производится после выполнения тела цикла, то это цикл **с постусловием** (цикл "до").



# Цикл For

Синтаксис:

**for** переменная := значение-1 **to** значение-2 **do** оператор

**for** переменная := значение-1 **downto** значение-2 **do** оператор

Пример 1. Квадраты чисел от 2-х до 10-и.

**for** x := 2 **to** 10 **do** WriteLn(x\*x);

Пример 2. Латинский алфавит.

**for** ch := 'A' **to** 'Z' **do** WriteLn(ch);

Пример 3. Использование цикла с **downto**.

**for** i := 10 **downto** 1 **do** WriteLn(i);

Пример 4. Использование составного оператора.

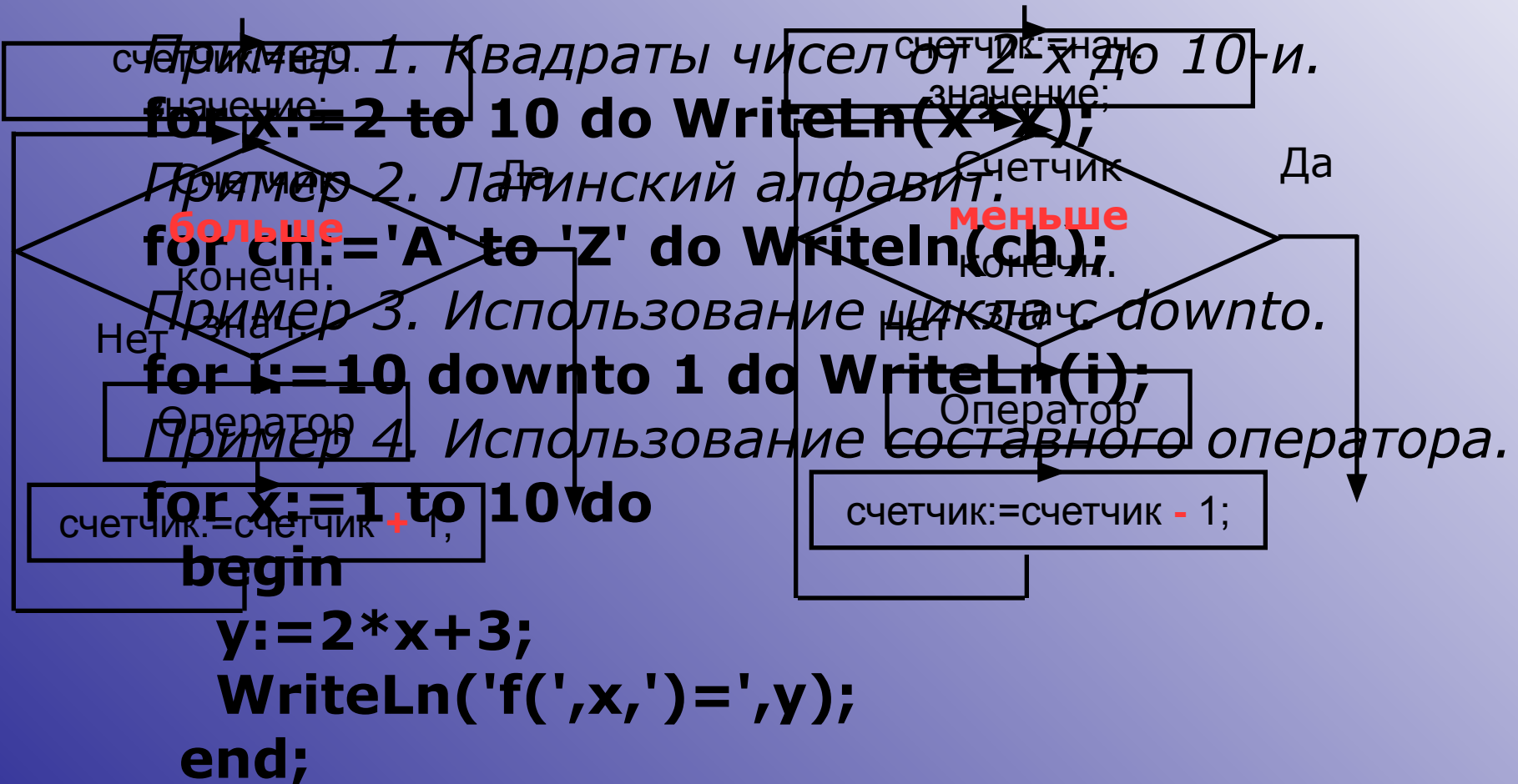
**for** x := 1 **to** 10 **do**

**begin**

y := 2\*x+3;

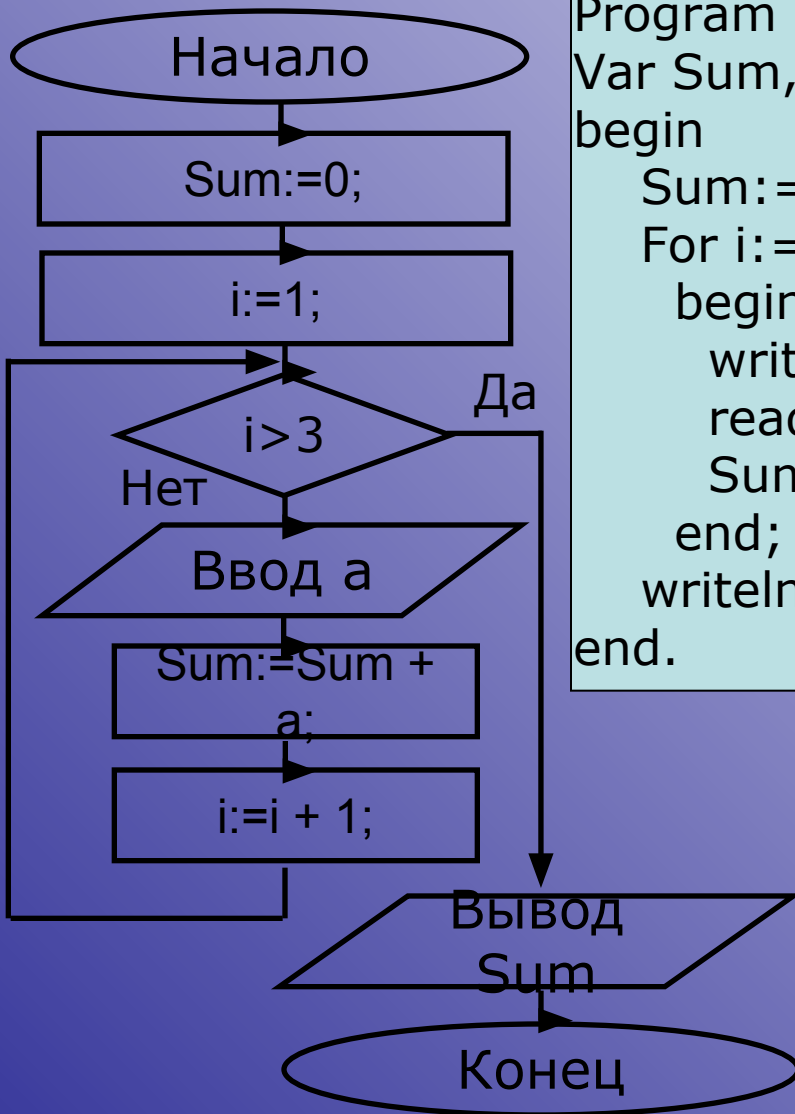
WriteLn('f(',x,')=',y);

**end;**



# Цикл For

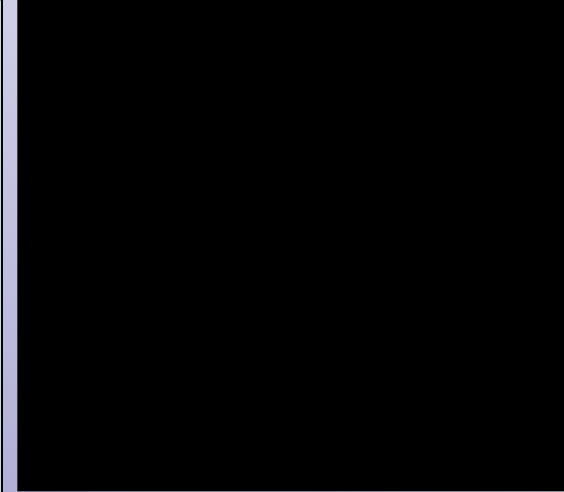
## Блок-схема



## Программный код

```
Program SumPoz;  
Var Sum, a, i: integer;  
begin  
  Sum:=0;  
  For i:=1 to 3 do  
    begin  
      write('Ввод числа');  
      readln(a);  
      Sum:=Sum + a;  
    end;  
  writeln('Sum = ', Sum);  
end.
```

## Экран программы



## Содержимое памяти

a = ~~0~~ неизвестно

Sum = ~~0~~ неизвестно

i = ~~1~~ неизвестно

# Цикл For

Сколько раз будут выполнены инструкции между **Begin** и **End**?

```
for i:=j to j+1 do  
  Begin  
    ...  
  End;
```

**Ответ: 2**

```
k:=0;  
for i:=2 downto k do  
  Begin  
    ...  
  End;
```

**Ответ: 3**

## Задания для индивидуальной работы

1. Ввести с клавиатуры 10 произвольных чисел. Подсчитать и напечатать количество положительных, отрицательных и нулевых чисел
2. Дана последовательность  $Y=n*x+(x-n)$   
 $X$  – изменяется от 2 до 10 с шагом 1,  $n$  – ввести с клавиатуры.  
Напечатать  $X$  и  $Y$
3. Вывести таблицу умножения на 5 от 9 до 4  
**(1вар)**
4. С клавиатуры ввести 10 чисел. Вычислить и напечатать среднее арифметическое этих чисел.**(2 вар)**
5. Найти площади прямоугольников со сторонами  $a$  и  $b=4,5,6,7,8$ **(3вар)**



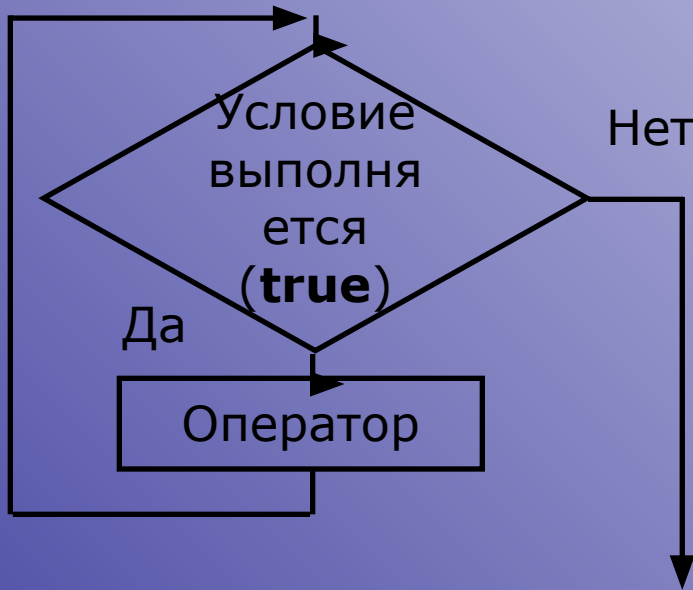




# Цикл While

Синтаксис:

**while** *выражение* **do** оператор



Пример.

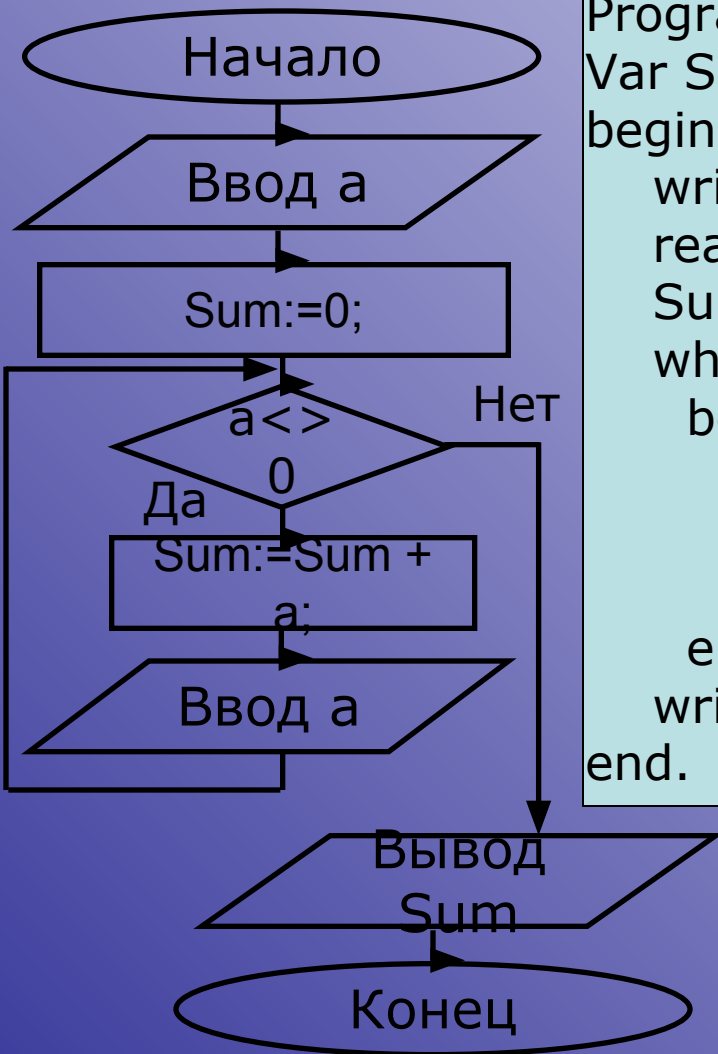
```
eps:=0.001;
```

```
while x > eps do
```

```
  x:=x/2;
```

# Цикл While

## Блок-схема



## Программный код

```
Program SumPoz;  
Var Sum, a: integer;  
begin  
  write('Ввод числа ');  
  readln(a);  
  Sum:=0;  
  while a<>0 do  
    begin  
      Sum:=Sum + a;  
      write('Ввод числа');  
      readln(a);  
    end;  
  writeln('Sum = ', Sum);  
end.
```

## Экран программы



## Содержимое памяти

**a = ~~0~~ неизвестно**

**Sum = ~~0~~ неизвестно**



# Цикл While

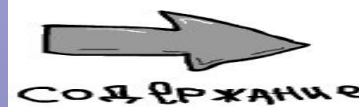
Чему будет равно значение переменной **n** после выполнения инструкций?

```
n:=0;  
While n<5 do  
    n:=n+1;
```

**Ответ: 5**

## Задания для индивидуальной работы

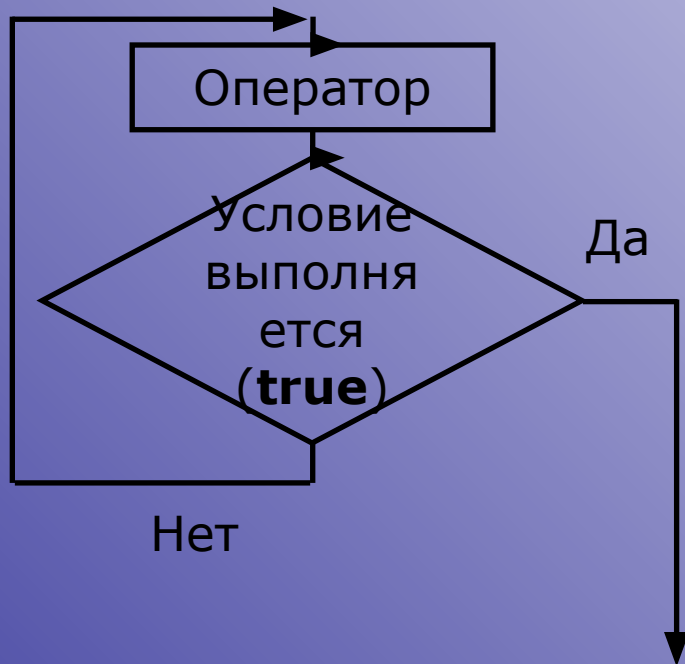
- 1 Вычислить и напечатать таблицу значений функции  
 $Y = \cos X(X+1)(X-1)$   
X изменяется от -2 до +2 с шагом 0,1
- 2 Вычислить произведение элементов последовательности четных чисел от 2 до 10
- 3 Вывести элементы последовательности нечетных чисел от 1 до 9 в обратном порядке.
- 4 Возвести число A в степень n
- 5 Написать программу подсчета N!





# Цикл Repeat

Синтаксис:



**repeat**

*оператор;*

*оператор;*

*...*

*оператор;*

**until** *выражение;*

Пример.

**repeat**

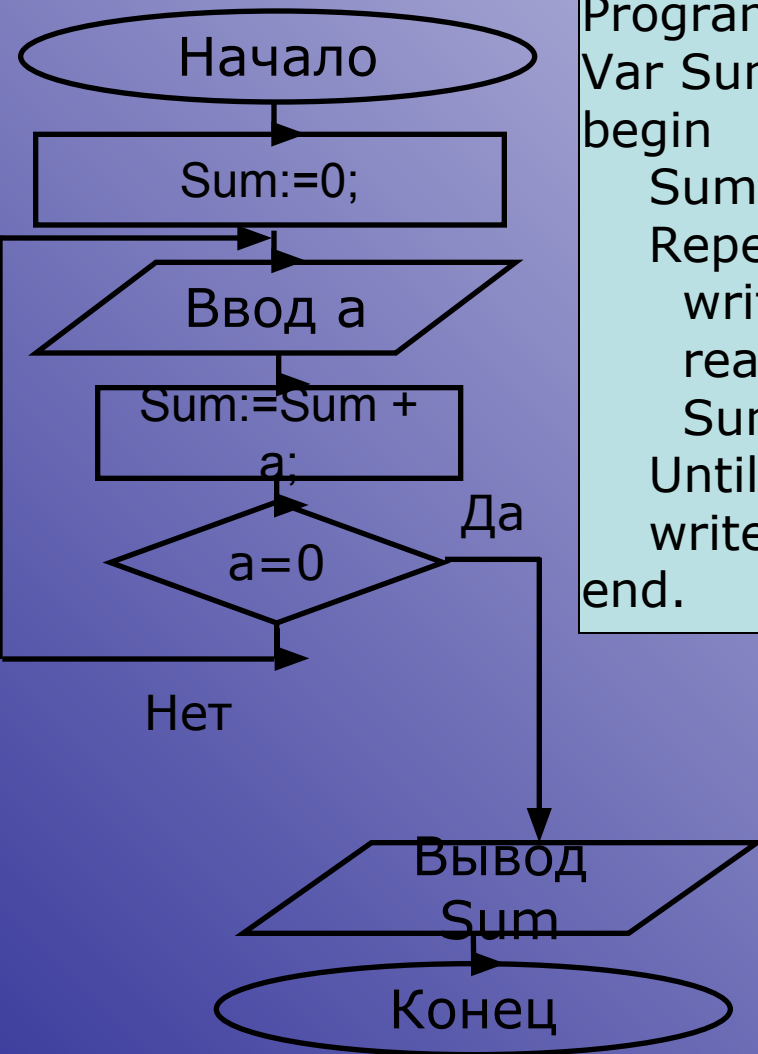
**WriteLn('Введите положительное число');**

**ReadLn(x);**

**until x > 0;**

# Цикл Repeat

## Блок-схема



## Программный код

```
Program SumPoz;  
Var Sum, a: integer;  
begin  
  Sum:=0;  
  Repeat  
    write('Ввод числа');  
    readln(a);  
    Sum:=Sum + a;  
  Until a=0;  
  writeln('Sum = ', Sum);  
end.
```

## Экран программы



## Содержимое памяти

**a = неизвестно**

**Sum = неизвестно**

# Цикл Repeat

Что делают следующие инструкции?

```
n:=0;
```

```
Repeat
```

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

```
  n:=n+1;
```

```
Until n<5;
```

**Ответ: Выводят  
одну звездочку**

Что нужно изменить в условии окончания цикла для предыдущего примера, чтобы было выведено 5 звездочек?

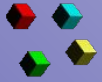
**Ответ: n=5**

# Задания для индивидуальной работы

1. Произведение первых **K** четных чисел равно **Pr**. Сколько сомножителей взято.
2. Числа Фибоначчи **f(n)** определяются формулами:  $f_0=f_1=1$ ;  $f_n=f_{n-1}+f_{n-2}$  при  $n=2,3,\dots$   
Составить программу:
  - Определения **f**– **20**- число Фибоначчи;
  - Поиска - первого числа Фибоначчи, большего **m(m>1)**
  - Вычисления **S-суммы** всех чисел Фибоначчи, которые не превосходят 10000
3. Ввести с клавиатуры **n** чисел. Вычислить сумму положительных чисел, порядковые номера которых четные.



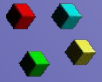




## Цикл For

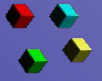
**for** переменная := значение-1 **to** значение-2 **do** оператор

**for** переменная := значение-1 **downto** значение-2 **do** оператор



## Цикл While

**while** выражение **do** оператор



## Цикл Repeat

**repeat**

оператор;

оператор;

...

оператор;

**until** выражение;

# Отличия и особенности хорошего стиля работы с циклическими операторами



Цикл с предусловием **WHILE**



Цикл с постусловием **REPEAT**

1. До начала цикла должны быть сделаны начальные установки переменных, управляющих условием цикла, для корректного входа в цикл.
2. В теле цикла должны присутствовать операторы, изменяющие переменные условия так, чтобы цикл через некоторое число итераций завершился.

3. Цикл работает пока условие истинно ( <i>пока true</i> )	3. Цикл работает пока условие ложно ( <i>пока false</i> )
4. Цикл завершается, когда условие становится ложным ( <i>do false</i> )	4. Цикл завершается, когда условие становится истинным ( <i>do true</i> )
5. Цикл может <b>не выполниться ни разу</b> , если исходное значение условия при входе в цикл равно <b>false</b>	5. Цикл обязательно <b>выполнится как минимум один раз</b> .
6. Если в теле цикла требуется более одного оператора, то необходимо использовать <b>составной оператор</b>	6. Независимо от количества операторов в теле цикла использование <b>составного оператора не требуется</b>



## Цикл со счетчиком for

1. Начальная установка переменной счетчика циклов до заголовка не требуется.
2. Изменение в теле цикла значений переменных, стоящих в заголовке цикла, не допускается.
3. Количество итераций цикла неизменно и точно определяется значениями нижней и верхней границ и шага цикла.
4. Нормальный ход работы может быть нарушен оператором goto.
5. Цикл может не выполниться ни разу, если шаг цикла будет изменять значение счетчика от нижней границы в направлении, противоположной верхней границе.

ПРИМЕР:

Составить программу нахождения произведения чисел от 1 до 10.



### 1. WHILE

```
S:=1;  
i:=1;  
While i<=10 do  
s:=s*i;  
i:=i+1;
```



### 2. REPEAT

```
S:=1; n:=10;  
i:=1;  
Repeat  
s:=s*i;  
i:=i+1;  
Until i>n;
```



### 3. For

```
S:=1;  
n:=10;  
For i:=1 to n do  
s:=s*i;
```

