



# ПРОГРАММИРОВАНИЕ ЦИКЛИЧЕСКИХ АЛГОРИТМОВ

## НАЧАЛА ПРОГРАММИРОВАНИЯ

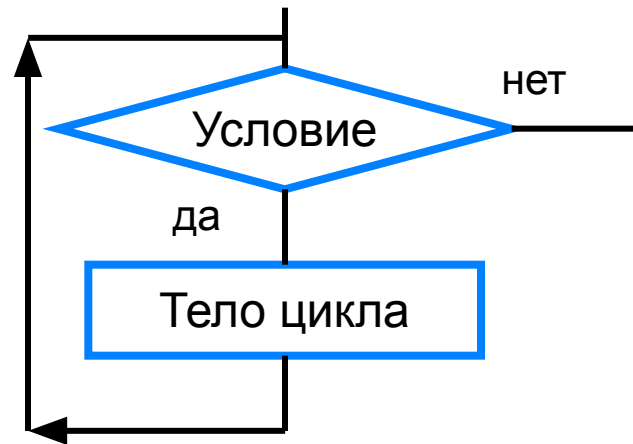
**8 класс**

# Ключевые слова

- **while** ( цикл-ПОКА)
- **repeat** (цикл-ДО)
- **for** (цикл с параметрами)



# Программирование циклов с заданным условием продолжения работы



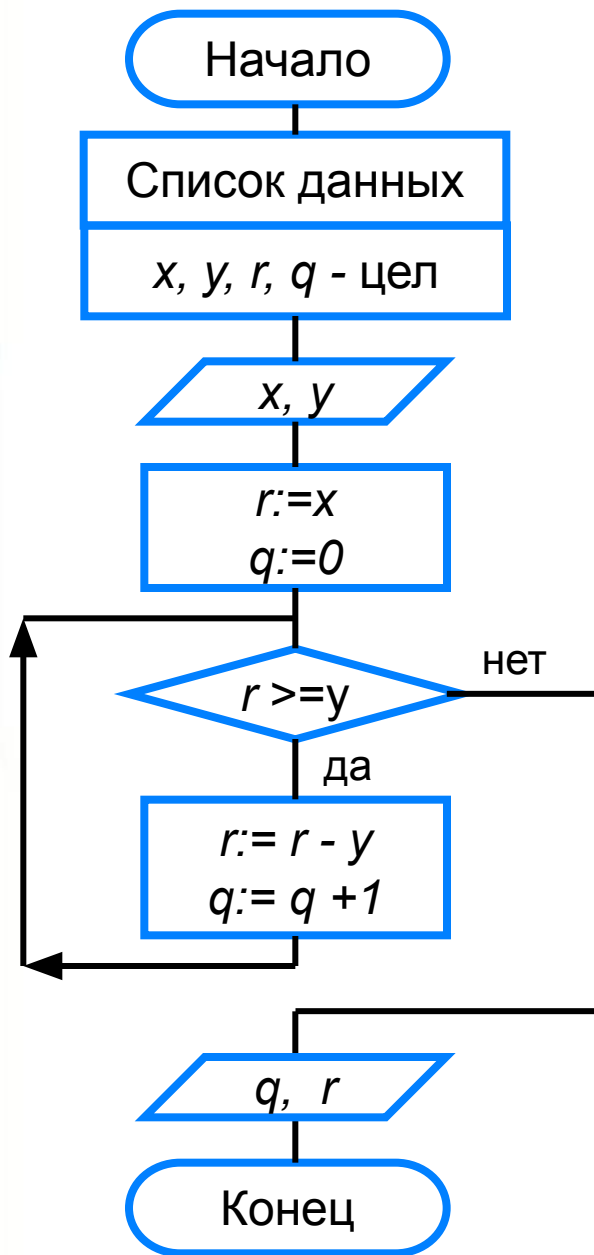
Общий вид оператора:

**while** <условие> **do** <оператор>

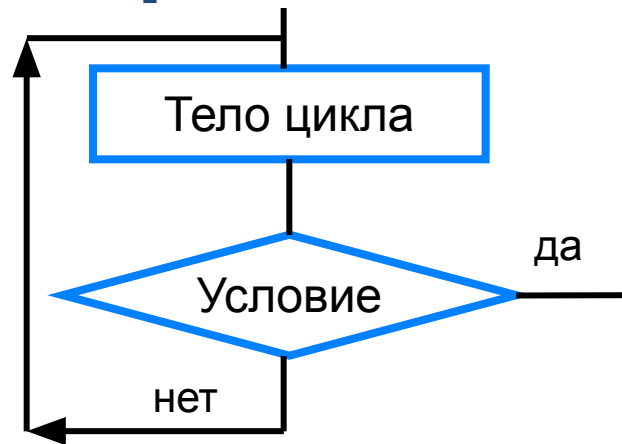
Здесь:

<условие> - логическое выражение;  
пока оно истинно, выполняется тело цикла;

<оператор> - простой или составной оператор,  
с помощью которого записано тело цикла.



# Программирование циклов с заданным условием окончания работы



Общий вид оператора:

**repeat** <оператор1; оператор2; ...; > **until** <условие>

Здесь:

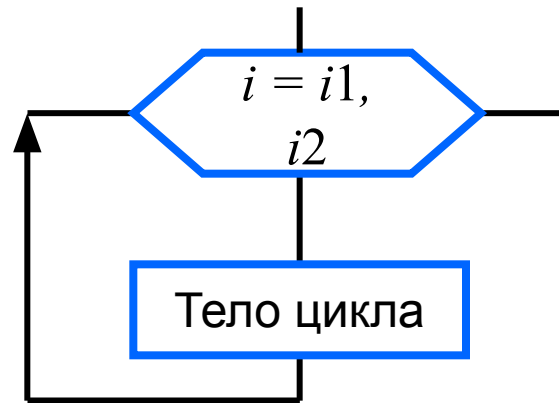
<оператор1>; <оператор2>; ... - операторы, образующие тело цикла;

<условие> - логическое выражение; если оно ложно, то выполняется тело цикла.

```
program n_15;  
  var i: integer; x: real;  
begin  
  writeln ('График тренировок');  
  i:=1;  
  x:=10;  
  repeat  
    i:=i+1;  
    x:=x+0.1*x;  
  until x>=25;  
  writeln ('Начиная с ', i, '-го дня  
спортсмен будет пробегать 25 км')  
end.
```



# Программирование циклов с заданным числом повторений



Общий вид оператора:

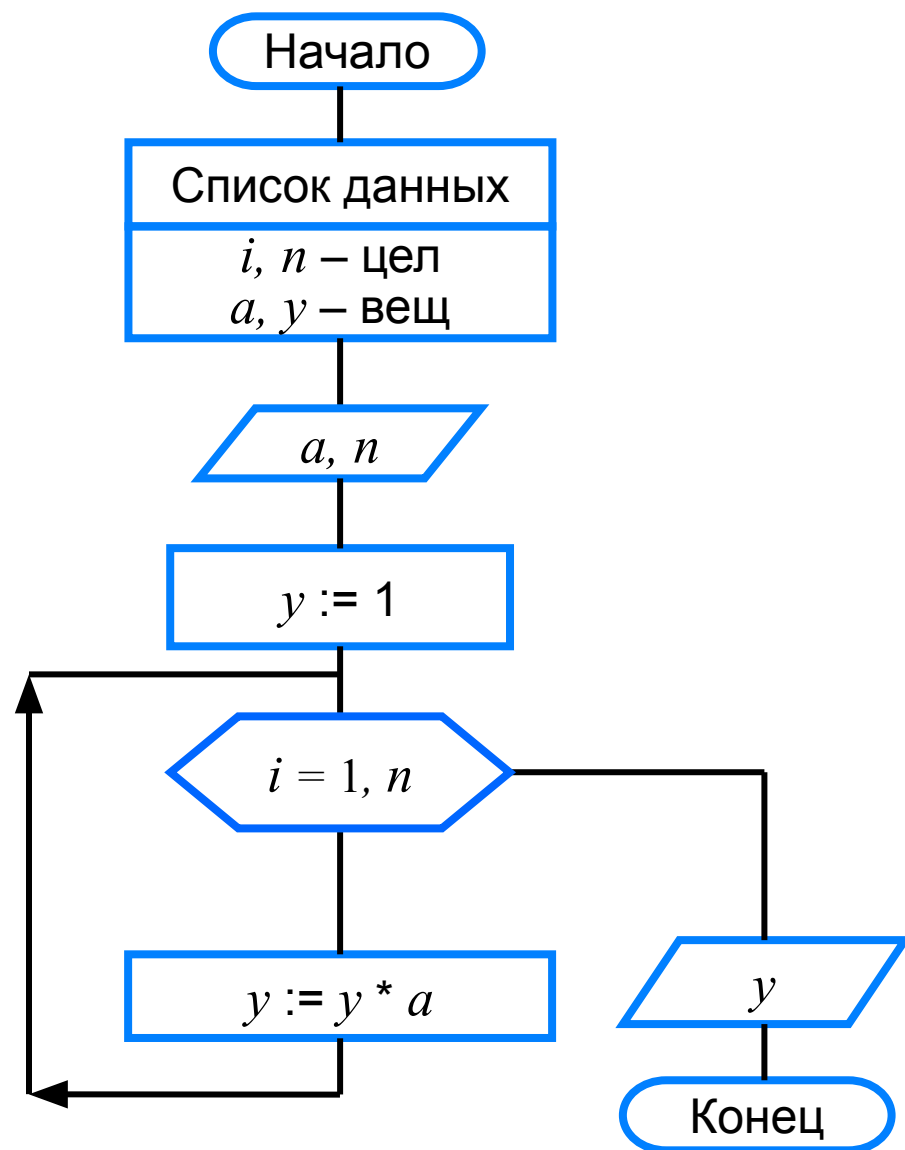
```
for <параметр>:=<начальное_значение>  
to <конечное_значение> do <оператор>
```

Здесь:

<параметр> - переменная целого типа;

После каждого выполнения тела цикла происходит увеличение на единицу параметра цикла; условие выхода из цикла - превышение параметром конечного значения.

<оператор> - простой или составной оператор - тело цикла.





# Различные варианты программирования циклического алгоритма

Для решения одной и той же задачи могут быть созданы разные программы.

Организуем ввод целых чисел и подсчёт количества введённых положительных и отрицательных чисел. Ввод должен осуществляться до тех пор, пока не будет введён ноль.

В задаче в явном виде задано условие окончания работы.

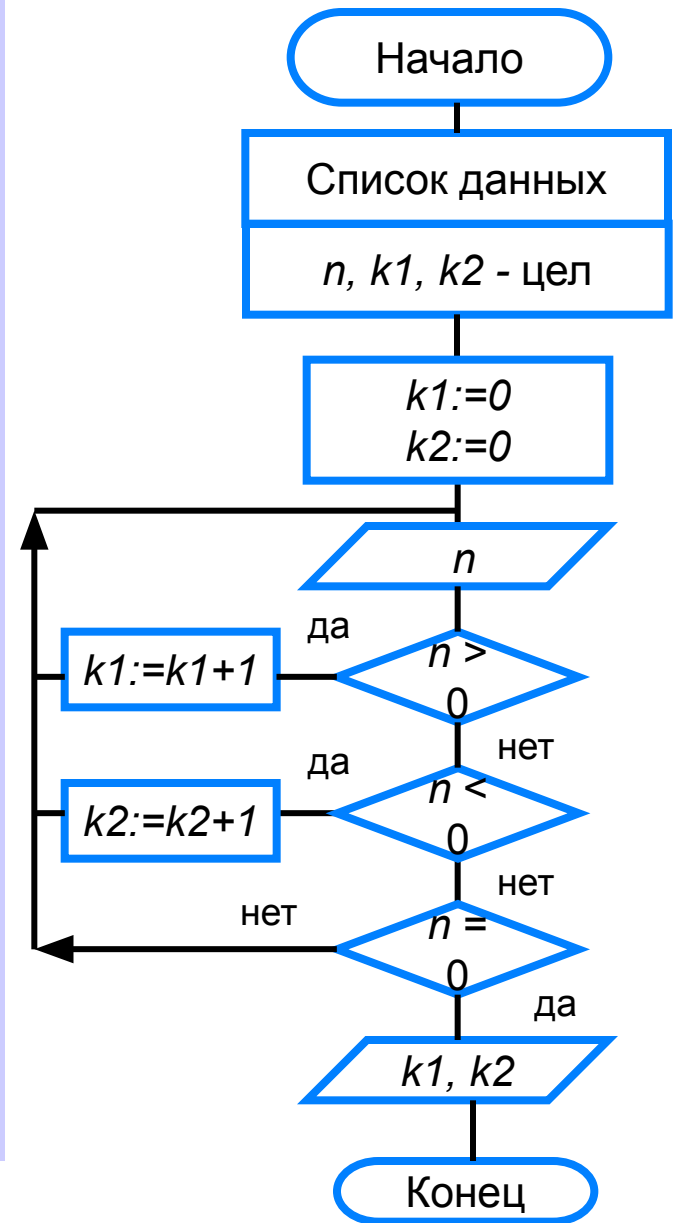


Воспользуемся оператором **repeat**.

```

program n_17;
  var n, k1, k2: integer;
begin
  k1:=0;
  k2:=0;
  repeat
    write ('Введите целое число>>');
    readln (n);
    if n>0 then k1:=k1+1;
    if n<0 then k2:=k2+1;
  until n=0;
  writeln ('Введено:');
  writeln ('положительных чисел – ', k1);
  writeln ('отрицательных чисел – ', k2)
end.

```



Ввод осуществляется до тех пор, пока не будет введен ноль.

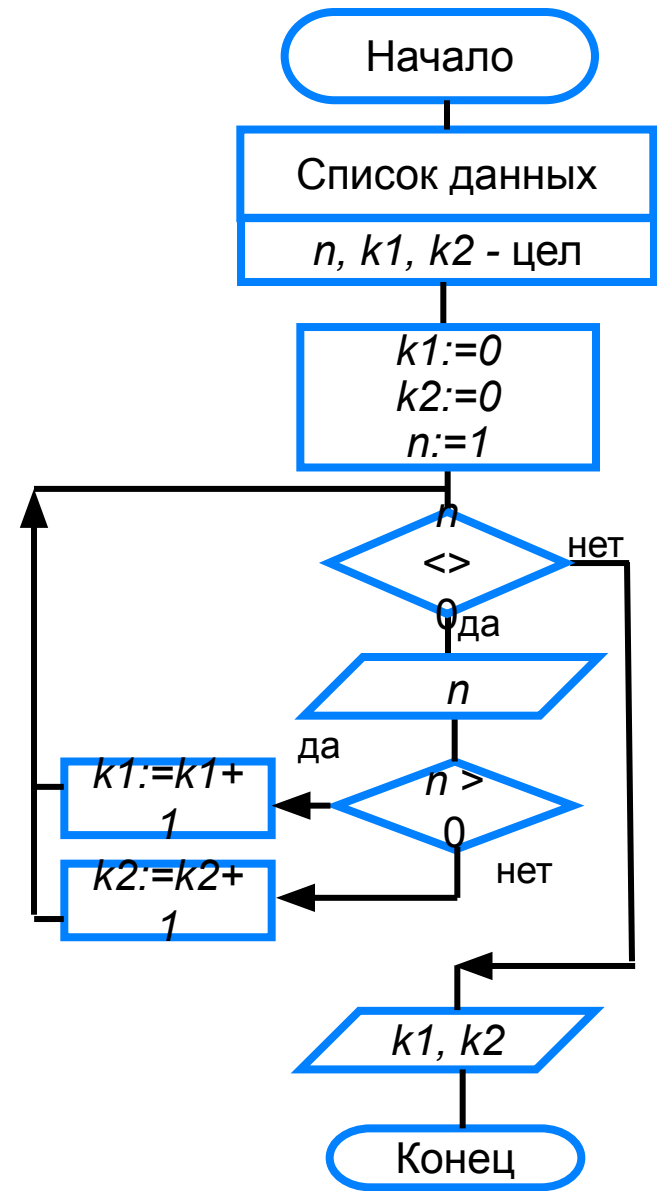


Работа продолжается, пока  $n \neq 0$ .



Воспользуемся оператором **while**:

```
program n_18;  
  var n, k1, k2: integer;  
begin  
  k1:=0;  
  k2:=0;  
  n:=1;  
  while n<>0 do  
  begin  
    writeln ('Введите целое число>>');  
    read (n);  
    if n>0 then k1:=k1+1;  
    if n<0 then k2:=k2+1;  
  end;  
  writeln ('Введено:');  
  writeln ('положительных – ', k1);  
  writeln ('отрицательных – ', k2)  
end.
```



# Самое главное

В языке Паскаль имеются три вида операторов цикла:

*while* (цикл-ПОКА)

*repeat* (цикл-ДО)

*for* (цикл с параметром).

Если число повторений тела цикла известно, то лучше воспользоваться оператором *for*; в остальных случаях используются операторы *while* и *repeat*.



# Вопросы и задания

Напишите программу, которая выводит на экран таблицу умножения для заданного натурального числа  $n$ . Таблица должна содержать  $n$  строк и  $n$  столбцов. В каждой строке и столбце должны быть натуральные числа от 1 до  $n$ . Выводите таблицу в формате, который вам нравится. Пример вывода для  $n=10$ :

Какой из трёх рассмотренных операторов цикла является, по вашему мнению, основным, таким, что

именно он должен использоваться для вычисления произведения чисел в диапазоне от 1 до  $n$ ?

Обозначьте оператор `repeat` как `1`, `while` как `2`, `do...until` как `3`. Сколько раз будет выполнен цикл `do...until` с какими значениями переменных `a`, `b`, `c`? Допишите необходимые операторы. этой последовательности операторов?

Пример входных данных	Пример выходных данных
Введите $n > 5$	$1 * 1 = 1$ $1 * 2 = 2$ $1 * 3 = 3$ $1 * 4 = 4$ $1 * 5 = 5$ $2 * 1 = 2$ $2 * 2 = 4$ $2 * 3 = 6$ $2 * 4 = 8$ $2 * 5 = 10$ $3 * 1 = 3$ $3 * 2 = 6$ $3 * 3 = 9$ $3 * 4 = 12$ $3 * 5 = 15$ $4 * 1 = 4$ $4 * 2 = 8$ $4 * 3 = 12$ $4 * 4 = 16$ $4 * 5 = 20$ $5 * 1 = 5$ $5 * 2 = 10$ $5 * 3 = 15$ $5 * 4 = 20$ $5 * 5 = 25$
Введите $n > 6$	$1 * 1 = 1$ $1 * 2 = 2$ $1 * 3 = 3$ $1 * 4 = 4$ $1 * 5 = 5$ $1 * 6 = 6$ $2 * 1 = 2$ $2 * 2 = 4$ $2 * 3 = 6$ $2 * 4 = 8$ $2 * 5 = 10$ $2 * 6 = 12$ $3 * 1 = 3$ $3 * 2 = 6$ $3 * 3 = 9$ $3 * 4 = 12$ $3 * 5 = 15$ $3 * 6 = 18$ $4 * 1 = 4$ $4 * 2 = 8$ $4 * 3 = 12$ $4 * 4 = 16$ $4 * 5 = 20$ $4 * 6 = 24$ $5 * 1 = 5$ $5 * 2 = 10$ $5 * 3 = 15$ $5 * 4 = 20$ $5 * 5 = 25$ $5 * 6 = 30$ $6 * 1 = 6$ $6 * 2 = 12$ $6 * 3 = 18$ $6 * 4 = 24$ $6 * 5 = 30$ $6 * 6 = 36$

# Опорный конспект

В языке Паскаль имеются три вида операторов цикла:

