

# ЦИКЛИЧЕСКИЕ АЛГОРИТМЫ

ВИДЫ ЦИКЛОВ В ЯЗЫКЕ C++

# ДВЕ ГРУППЫ ЦИКЛОВ

**ЦИКЛЫ СО  
счетом  
(арифметические)**

ЦИКЛЫ, ВЫПОЛНЯЮЩИЕ  
свое тело,  
определенное  
количество раз

**ЦИКЛЫ С  
условием  
(логические)**

ЦИКЛЫ, ВЫПОЛНЯЮЩИЕ  
свое тело, пока  
заданное условие  
либо истинно,  
либо ложно

Циклы позволяют многократно выполнять одну или группу команд, причем в тексте программы нет необходимости записывать эти команды несколько раз.

В языке программирования C существует три вида циклов:

1. Арифметический цикл **ДЛЯ**  
**for**

2. Логический цикл **ПОКА** (с предусловием) **while**

3. Логический цикл **ДО** (с послеусловием)  
**do..while**

# Цикл ДЛЯ(с параметром)

$i := a, b, 1$

Операторы  
(тело цикла)

```
for(i = a; i <= b; i++)  
{  
    ... // операторы  
}
```

$i$  – параметр цикла (счетчик),  
переменная целого типа  
 $a, b$  – начальное и конечное  
значения счетчика

**ПАРАМЕТР (счетчик)** - это переменная как правило целого типа, которая определяет количество выполнений (повторений) цикла. Чаще всего параметр изменяется с шагом в 1 ( то есть  $i++$  или  $i--$ )

# Особенности цикла `for`

1. Число повторений операторов цикла должно быть известно заранее
2. Условие продолжения цикла проверяется до начала его прохождения
3. Тело цикла может быть не выполнено ни разу
4. Цикл `for` можно использовать и без инструкций (пустой цикл)

```
for (num = 1; num <= 1000; num++);
```



Пустой оператор

# Порядок выполнения цикла for

for (инициализация; условие; приращение) оператор;

**ИЛИ**

for (инициализация; условие; приращение) { операторы; }

1. Блок *инициализация* (счетчик принимает свое начальное значение)
2. Блок *условие*: если истинно, то п.3, иначе п.6
3. Блок *тело цикла* (один или несколько операторов)
4. Блок *приращение* (изменение счетчика)
5. К п.2
6. Выход

# Сколько раз будет выполнено тело цикла с данным заголовком?

1. `for(i = -10; i < -4; i++)`
2. `for(i = 6; i < 2; i++)`
3. `for(i = 3; i <= 3; i++)`
4. `for(i = 10; i > 3; i -= 2)`

# ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ ЦИКЛА FOR

Решение задач



Распечатать числа от 1 до n :

- а) в строчку, б) в столбик; с) от n до 0
- д) четные числа, е) нечетные числа

```
#include <iostream>
using namespace std;
main()
{
    int n,i;
cin>>n;
For (i=0;i<=n;++i)
cout>>i;
}
```

## Найти сумму чисел от 0 до 5

```
#include <iostream>
using namespace std;
main()
{
    int i; s=0;
    For (i=0;i<=5;++i)
    s=s+i;
    cout>>s;
}
```

## Задача:

**Вычислить произведение чисел от 1 до 5  
используя различные варианты цикла.**

**Математическая модель:**

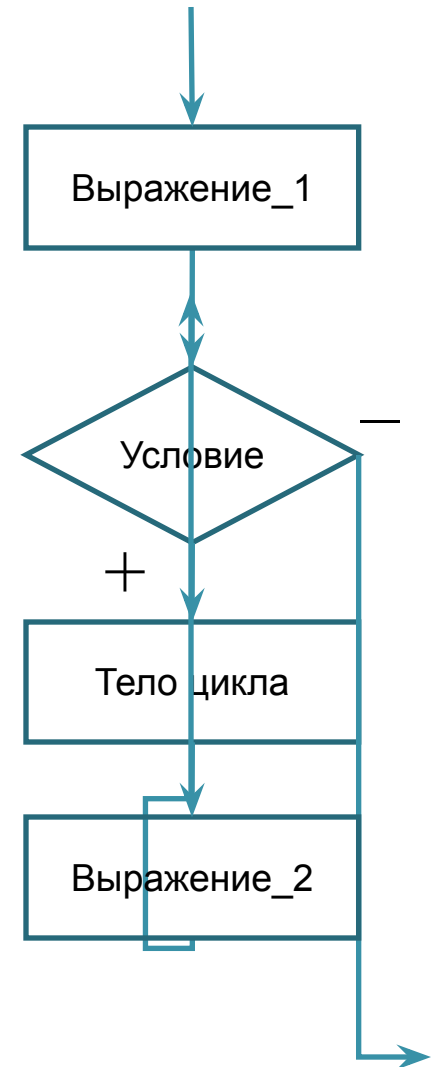
$$P = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$$

# Цикл `for` (параметрический)

```
for (Выражение_1; Условие;  
Выражение_2)  
    Тело цикла
```

**Пример:** вычисление 5!

```
int i, n=5, s=1;  
  
for (i = 1; i <= n; i++)  
    s = s*i;  
  
// в s - результат
```



1. Вывести на экран числа от 1 до  $n$  в строчку ?
2. Вывести на экран числа от 1 до  $n$  в столбик ?
3. Вывести на экран числа от 1 до  $n$  кратные 5 в строчку ?
4. Вывести на экран числа от 1 до  $n$  кратные 3 в строчку ?
5. Найти сумму чисел от 1 до  $n$  ?
6. Найти произведение чисел от 1 до  $n$  ?
7. Найти сумму четных чисел от 1 до  $n$  ?
8. Найти произведение нечетных чисел от 1 до  $n$  ?
9. Найти произведение чисел кратных 5 и сумму чисел кратных 3 в диапазоне от 1 до  $n$  ?

Вывести на экран столько элементов ряда Фибоначчи, сколько указал пользователь. Например, если на ввод поступило число 6, то вывод должен содержать шесть первых чисел ряда Фибоначчи: 1 1 2 3 5 8 13.

Написать программу, которая выводит таблицу перевода расстояний в дюймах в сантиметры для значений от  $n$  до  $m$  дюймов (1 дюйм = 2,54 см)

Написать программу, которая выводит таблицу умножения числа  $N$  в интервале от  $L$  до  $K$ . Ввод  $N, L, K$  выполнять с клавиатуры.

Бизнесмен взял ссуду  $m$  тысяч рублей в банке под  $k\%$  годовых. Рассчитать его долг через  $N$  лет, если за это время он не будет отдавать долг?

## При составлении программ часто возникают ситуации, когда:

1. Заранее не известно количество повторений цикла;
2. Переменная-счетчик цикла должна изменяться с шагом  $\neq 1$ .

*В таких случаях используют*

# Сравнение арифметического и логического циклов

1. Количество повторений в арифметическом цикле известно до начала выполнения цикла.
2. Количество повторений в логическом цикле зависит от проверки условия.
  - Логический цикл **с предусловием** может не выполняться ни разу.
  - Логический цикл **с послеусловием** выполняется хотя бы один раз.

Присесть 10 раз

Приседать пока не устанешь.



# Логические циклы – циклы с условием

это циклическое повторение блока команд,  
пока выполняется (или не выполняется)  
некоторое условие

Условие —

*простое выражение*

*отношения*

ИЛИ

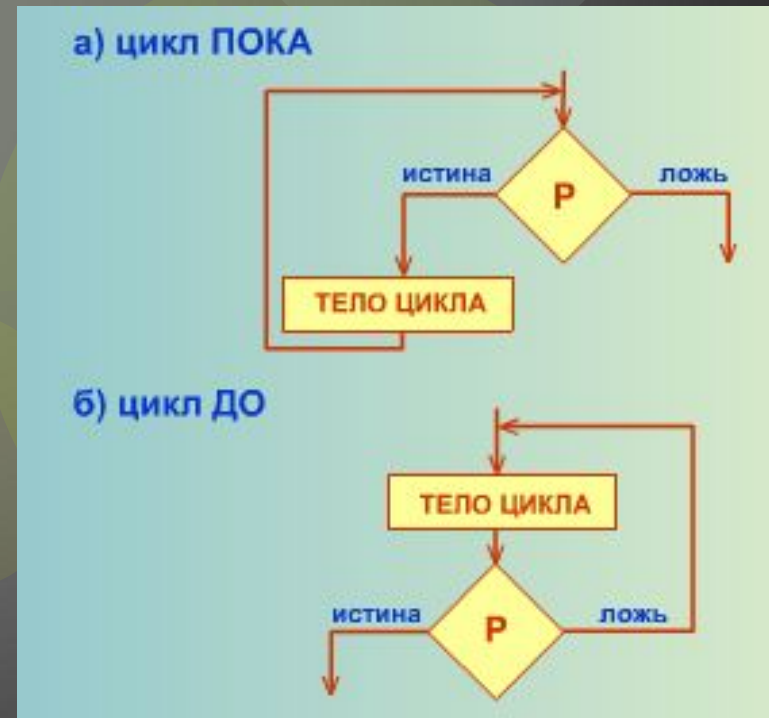
*сложное выражение*

*отношения*

со значением

*true* (истина)

или *false* (ложь)



# Цикл ПОКА (с предусловием)

*while*



```
i = a;  
while (i <= b)  
{  
    ... // операторы;  
    i += h;  
}
```

*i* – параметр цикла (счетчик)  
*a*, *b* – начальное и конечное значения счетчика  
*h* – шаг изменения параметра

# Цикл ДО (с послеусловием)


do...while



```
i = a;  
do  
{  
    ... // операторы;  
    i += h;  
} while (i > b);
```

*i* – параметр цикла (счетчик)  
*a*, *b* – начальное и конечное значения счетчика  
*h* – шаг изменения параметра

В операторе цикла **do..while** точка с запятой *никогда не ставится* после слова **do**



Если точку с запятой поставить в операторе **while** после задания условия, то цикл будет пустой

```
while (a<5);
```

---

Инструкции тела цикла **while** *могут быть не выполнены ни разу*

Цикл ПОКА	Цикл ДО
<pre>while(усл) { &lt;операторы&gt; }</pre>	<pre>do { &lt;операторы&gt; }while(усл);</pre>

1. В цикле **while** проверка условия выполнения цикла находится в начале цикла, а в **do..while** – в конце.

2. Цикл **do..while** всегда выполняется хотя бы один раз, а цикл **while** может не выполняться ни разу.

3. В цикле **while** выход из цикла осуществляется, если условие ложно, а в **do..while** – если условие истинно.



# **ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ ЛОГИЧЕСКИХ ЦИКЛОВ**

**Решение задач**

## Задача:

**Вычислить произведение чисел от 1 до 5  
используя различные варианты цикла.**

**Математическая модель:**

$$P = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$$

# Цикл **while** (с предусловием)

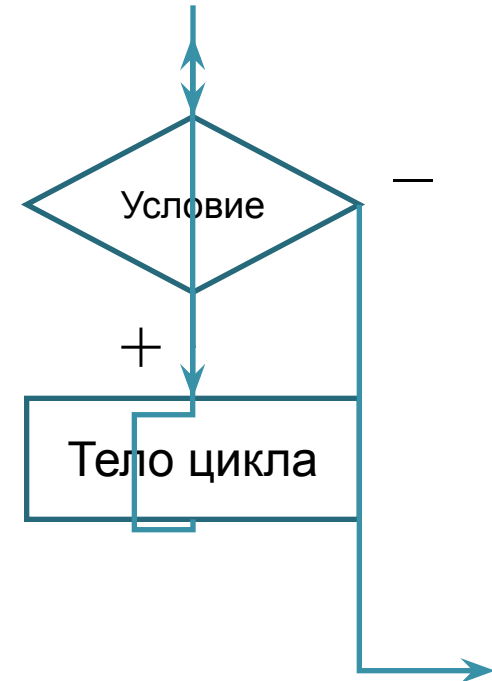
**while**

(Условие)

*Тело цикла*

Пример: вычисление 5!

```
int i, n=5, s=1;
i = 1;
while (i <= n)
{
    s *= i;
    i++;
}
// в s - результат
```





# Цикл **while** (с предусловием)

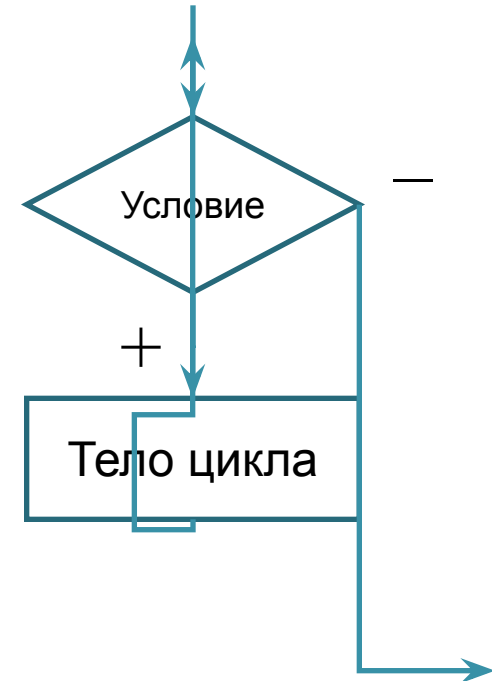
**while**

(Условие)

*Тело цикла*

Пример: вычисление 5!

```
int i, n=5, s=1;
i = 1;
while (i <= n)
{
    s *= i;
    i++;
}
// в s - результат
```



# Цикл `do while` (с постусловием)

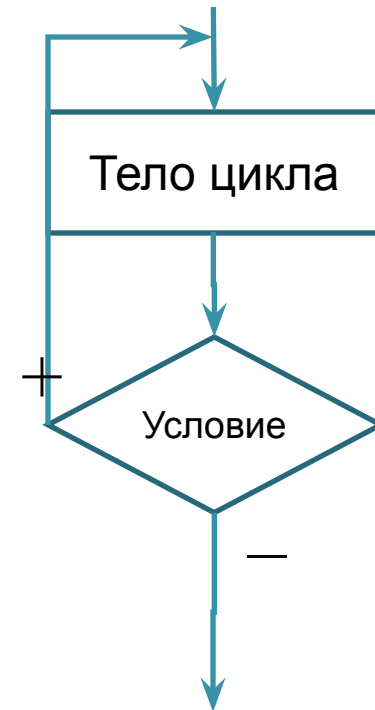
**do**

*Тело цикла*

**while** (*Условие*)

**Пример:** вычисление 5!

```
int i, n=5, s=1;
i = 1;
do
{
    s *= i;
    i++;
}
while (i <= n)
// в s - результат
```



# Операторы прерывания исполнения цикла

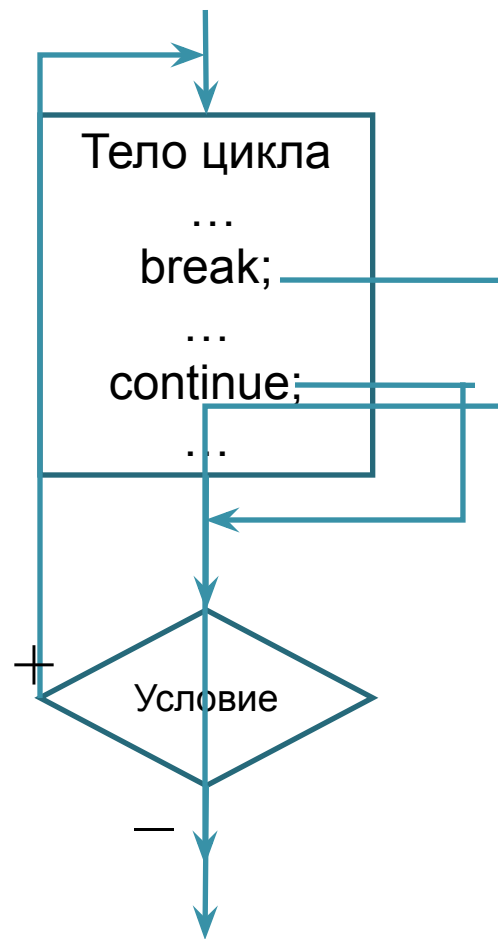
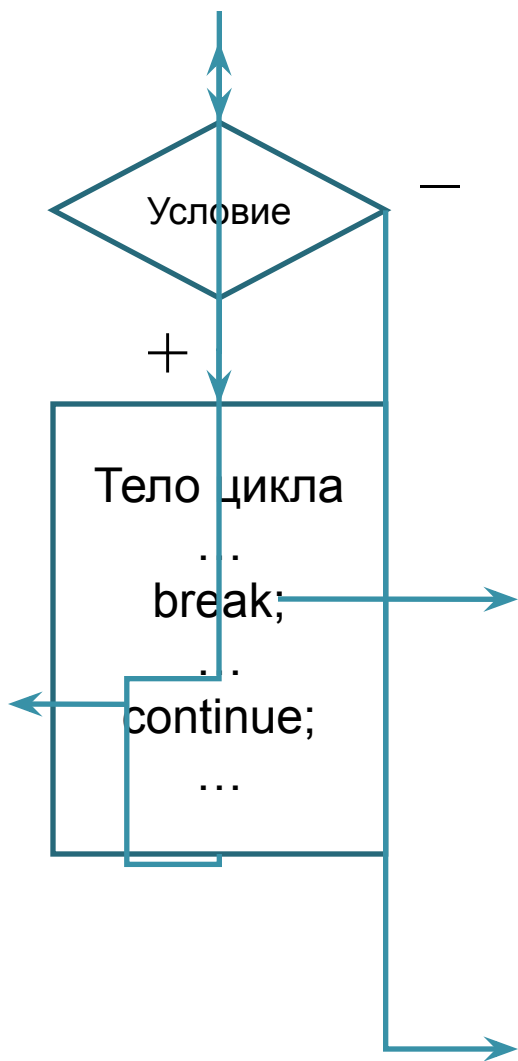
Оператор **break** прекращает выполнение оператора цикла и передает управление следующему за этим циклом оператору.

Наиболее естественна следующая форма тела цикла:

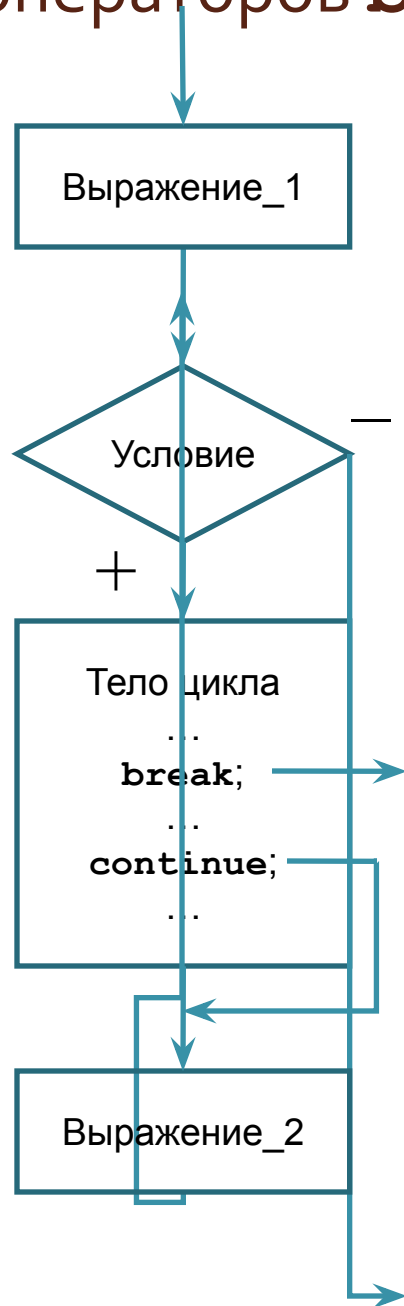
```
{  
    операторы  
    break;  
    операторы  
}
```

Оператор **continue** позволяет в любой точке тела цикла прервать текущую итерацию и перейти к проверке условия выполнения следующей итерации

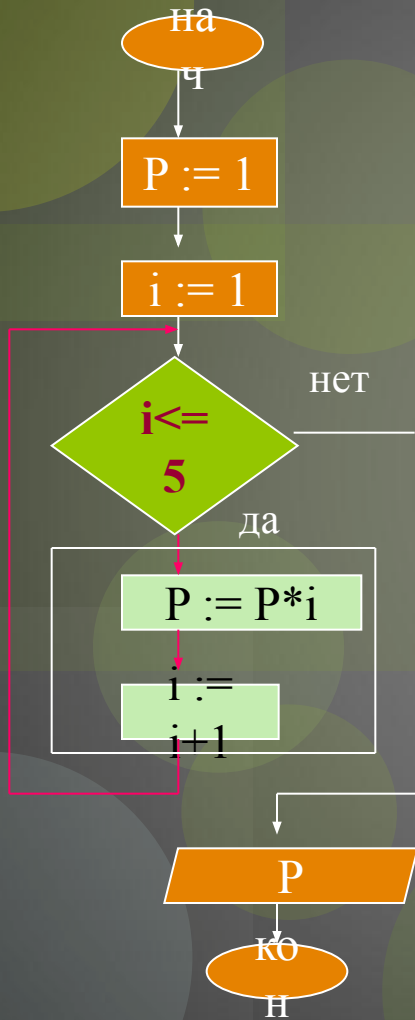
# Схемы работы операторов **break** и **continue**



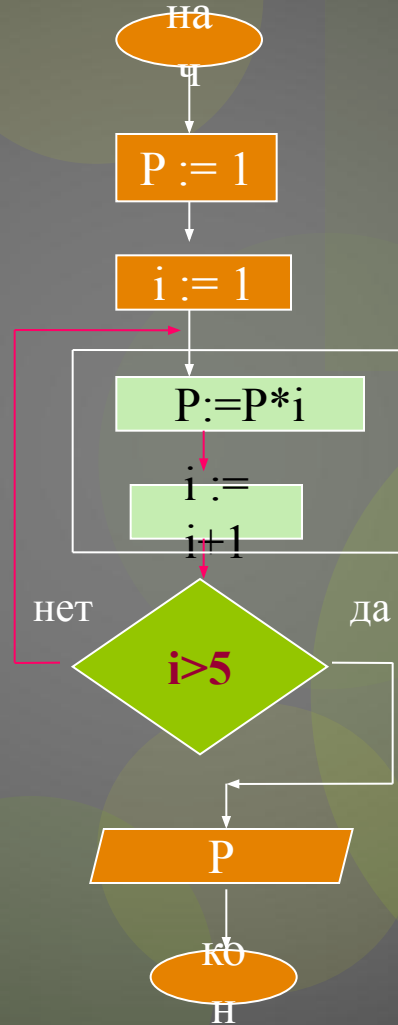
# Схемы работы операторов **break** и **continue**



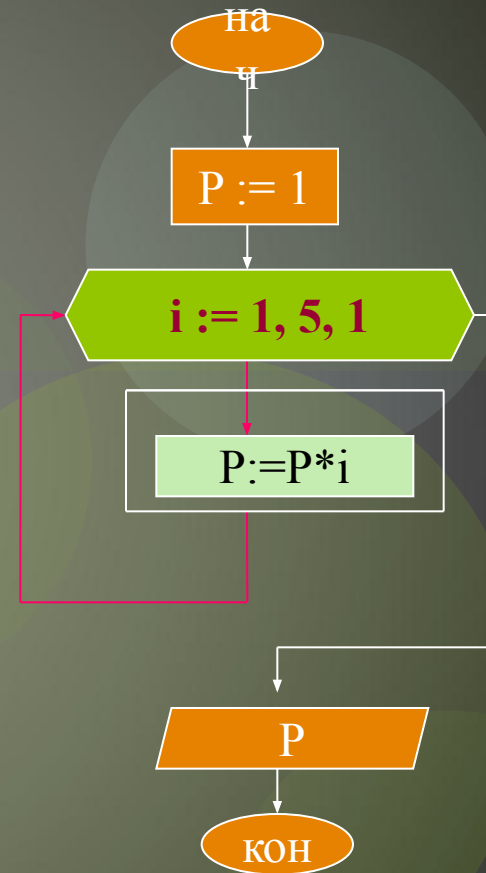
# «Пока»



# «ДО»



# «ДЛЯ»



# Сопоставь

Линейный алгоритм – это ...

... алгоритм,  
выполняющий некоторую  
последовательность,  
в зависимости от  
наложенного условия.

Разветвляющийся  
алгоритм – это...

... алгоритм, многократно  
выполняющий некоторую  
последовательность  
действий.

Чтобы в теле цикла со счетчиком и  
предусловием выполнилось несколько  
операторов ...

...алгоритм,  
выполняющий  
последовательность  
действий  
в том порядке, в котором  
они записаны.

Циклический алгоритм – это ...

... необходимо поставить  
операторные скобки { }

**Цикл с постусловием выполняется ...**

**... пока условие работы цикла ложно.**

**Цикл с постусловием выполняет свою работу до тех пор...**

**... условие работы цикла истинно.**

**Цикл с предусловием выполняется пока...**

**... всегда хотя бы один раз.**