



# ОСНОВЫ ЯЗЫКОВ C/C++

Кафедра ЭО  
Петрухин О.М.

## Задача:

вывести пять раз на консоль фразу “Hello, World!”

```
std::cout << "Hello, world!" << std::endl;  
std::cout << "Hello, world!" << std::endl;  
std::cout << "Hello, world!" << std::endl;  
std::cout << "Hello, world!" << std::endl;  
std::cout << "Hello, world!" << std::endl;
```

Циклы — это разновидность условных конструкций. Они позволяют выполнять команды несколько раз, если определённое утверждение верно. Например, с помощью циклов можно пять раз вывести на экран какую-нибудь надпись или посчитать сумму всех чисел.

**Существует три основных вида циклов:**

**1. Цикл с предусловием:**

- условие проверяется перед выполнением тела цикла
- тело цикла может не выполниться ни разу

**2. Цикл с постусловием:**

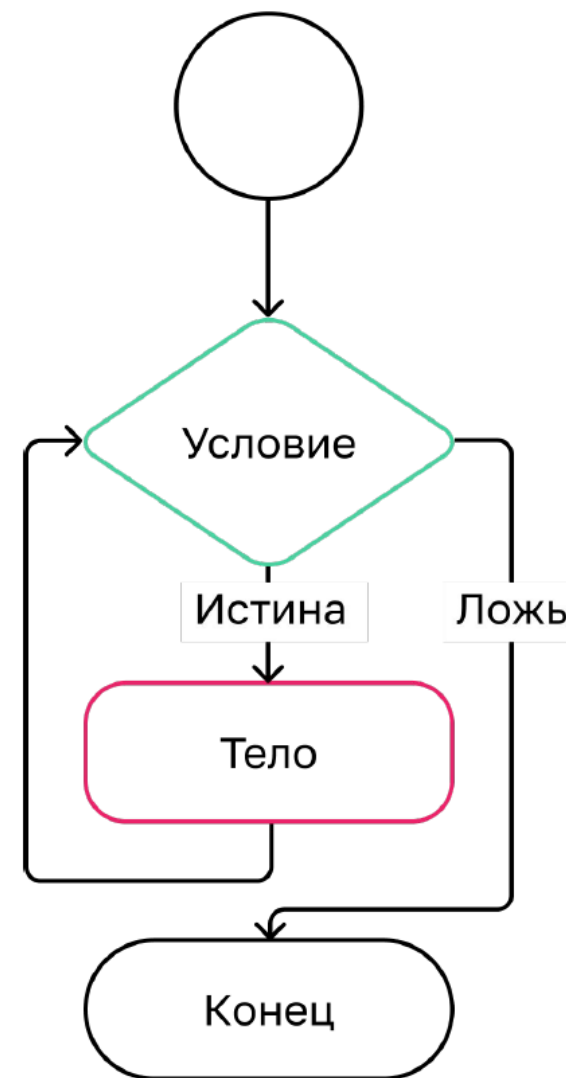
- условие проверяется после выполнения тела цикла
- тело цикла выполнится хотя бы один раз

**3. Цикл с подсчётом:**

- выполняется определённое количество раз
- параметр, условие и изменения которого задаются в начале цикла%

Оператор **while** выполняет оператор или блок операторов, пока логическое выражение равно значению **true**. Так как это выражение вычисляется и проверяется перед каждым выполнением цикла, цикл **while** выполняется ноль или несколько раз

```
while (условие)  
{  
    тело цикла;  
}
```



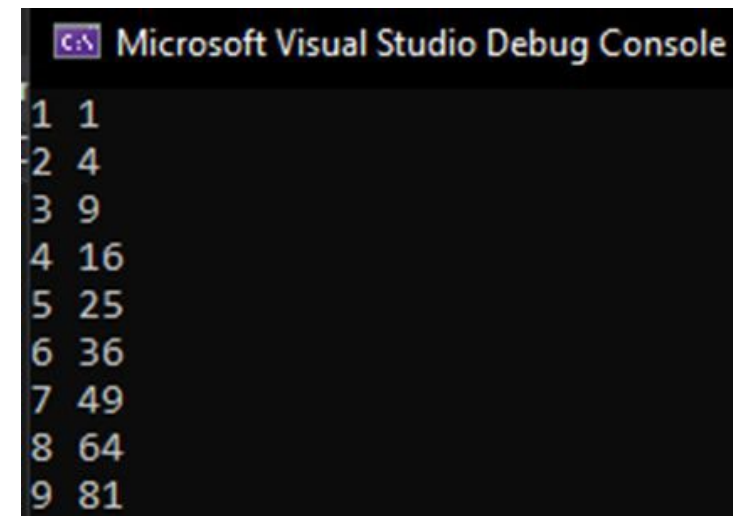
- Условие (проверяемое утверждение) определяет, продолжать выполнение цикла или нет
- Тело цикла определяет действия, которые необходимо выполнить на каждом шаге
- Шаг цикла — однократное выполнение тела цикла



```
int count = 5;
while (count-- > 0)
{
    std::cout << "Hello, world!" << std::endl;
}
```

Для чисел, квадрат которых меньше 100, вывести на экран число и его квадрат.

```
int a = 1;
while (a * a < 100)
{
    std::cout << a << " " << a * a << std::endl;
    a++;
}
```

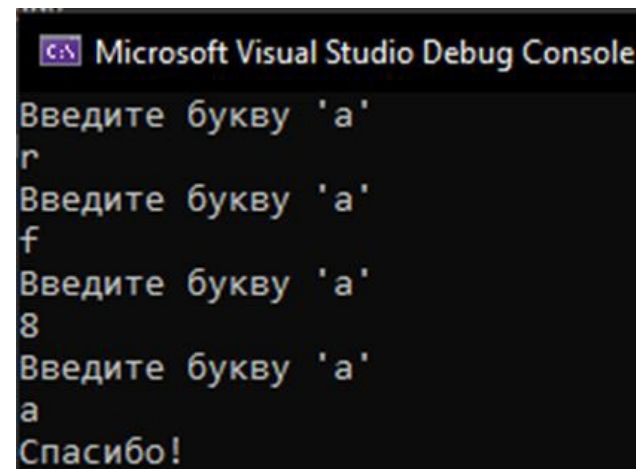


Microsoft Visual Studio Debug Console

```
1 1
2 4
3 9
4 16
5 25
6 36
7 49
8 64
9 81
```

Запросить у пользователя ввести целое число. При некорректном вводе запросить повторный ввод

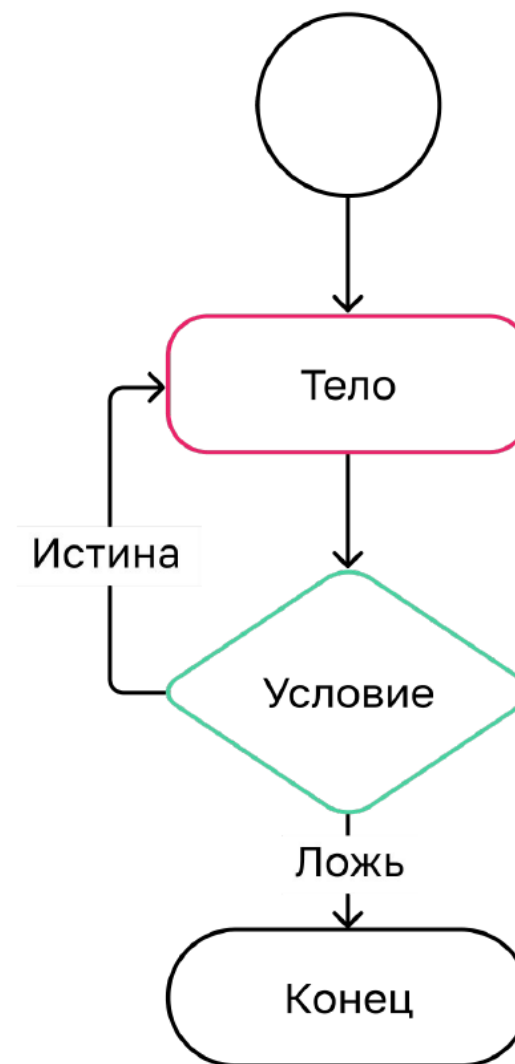
```
std::string s;  
std::cout << "Введите букву 'а'" << std::endl;  
std::cin >> s;  
while (s != "а")  
{  
    std::cout << "Введите букву 'а'" << std::endl;  
    std::cin >> s;  
}  
std::cout << "Спасибо!" << std::endl;
```



```
Microsoft Visual Studio Debug Console  
Введите букву 'а'  
г  
Введите букву 'а'  
f  
Введите букву 'а'  
8  
Введите букву 'а'  
а  
Спасибо!
```

Оператор **do** выполняет оператор или блок операторов, пока определённое логическое выражение равно значению **true**. Так как это выражение вычисляется и проверяется после каждого выполнения цикла, цикл **do... while** выполняется один или несколько раз

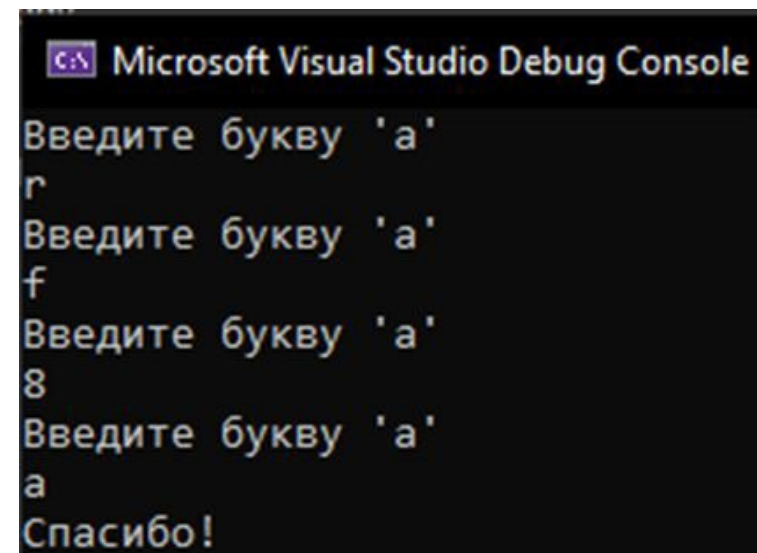
```
do  
{  
    тело цикла  
} while (условие);
```





Попросить пользователя ввести целое число. При некорректном вводе запросить повторный ввод

```
std::string s;  
do  
{  
    std::cout << "Введите букву 'а'" << std::endl;  
    std::cin >> s;  
} while (s != "а");  
std::cout << "Спасибо!" << std::endl;
```



```
Microsoft Visual Studio Debug Console  
Введите букву 'а'  
r  
Введите букву 'а'  
f  
Введите букву 'а'  
8  
Введите букву 'а'  
а  
Спасибо!
```

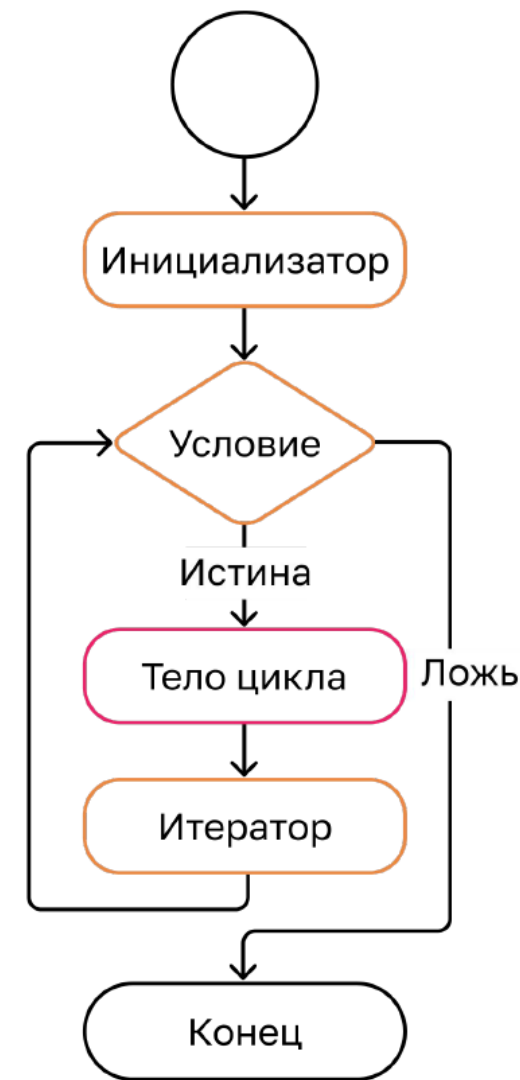
Оператор **for** выполняет оператор или блок операторов, пока определённое логическое выражение равно значению **true**.

```
for (инициализатор; условие; итератор)
{
    тело цикла
}
```

**Инициализатор** выполняется один раз перед входом в цикл.

**Условие** — условное выражение, вычисляется и проверяется перед каждым шагом.

**Итератор** выполняется после каждого шага цикла

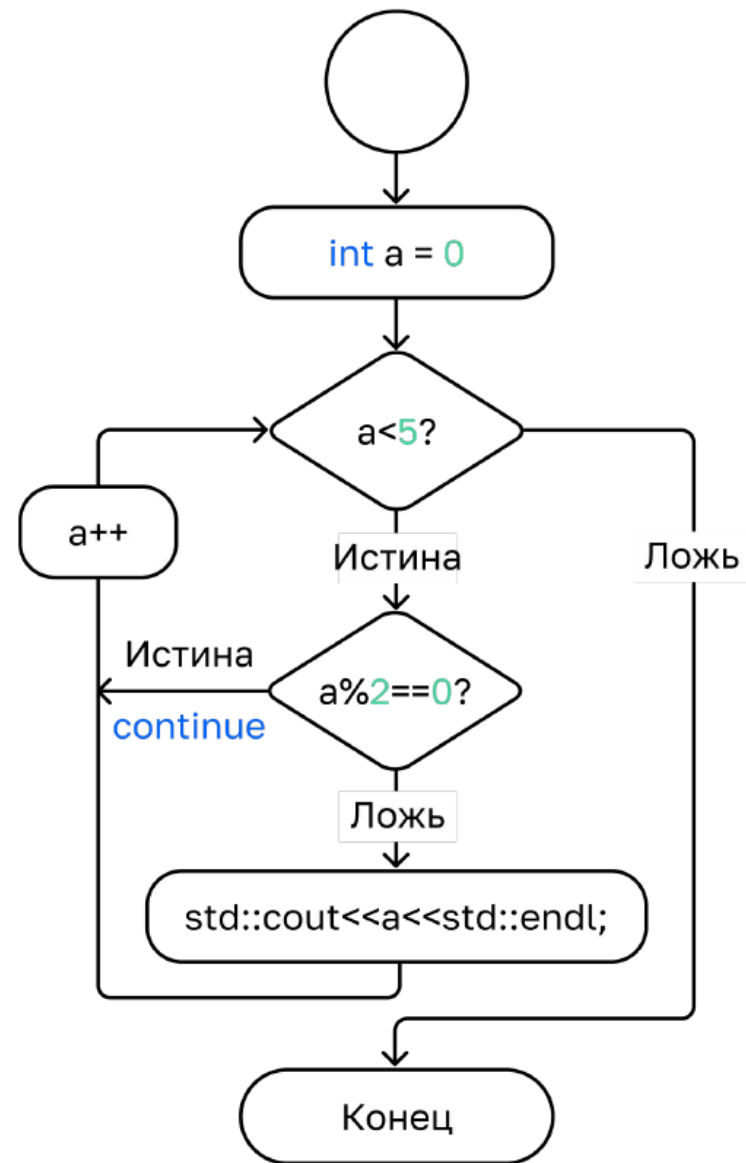


Напишем программу с помощью **for**, которая выводит степени введенного пользователем числа

```
#include <iostream>
#include <cmath>
int main() {
    int a;
    std::cout << "Введите число: ";
    std::cin >> a;
    for(int i = 1; i <= 4; i++)
    {
        std::cout << a << " в степени " << i << " = ";
        std::cout << std::pow(a, i) << std::endl;
    }
}
```

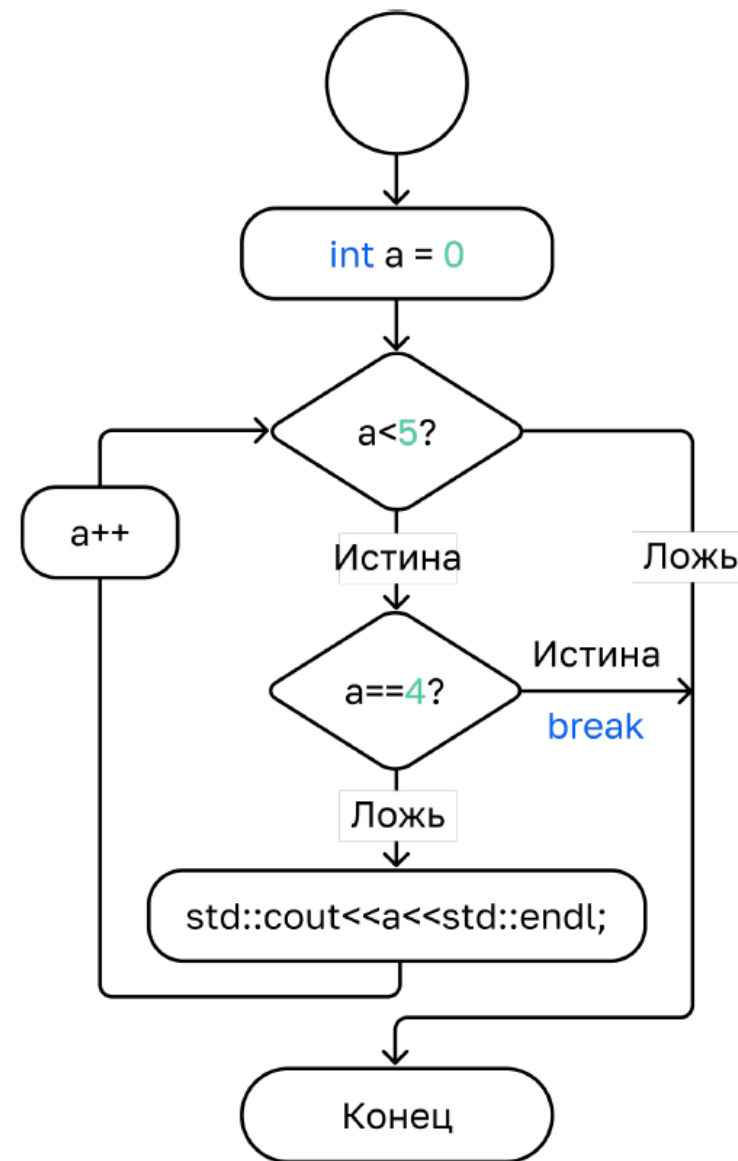
Оператор `continue` передаёт управление следующей итерации цикла `while`, `do-while` или `for`, в котором она встречается

```
for (int a = 0; a < 5; a++)  
{  
    if (a % 2 == 0)  
    {  
        continue;  
    }  
    std::cout << a << std::endl;  
}
```



Оператор **break** завершает выполнение цикла **while**, **do-while** или **for**, в котором он встречается. Управление передаётся оператору, который расположен после завершённого цикла

```
for (int a = 0; a < 5; a++)  
{  
    if (a == 4)  
    {  
        break;  
    }  
    std::cout << a << std::endl;  
}
```



Цикл, размещённый в теле другого цикла, называется вложенным

Программа, которая выводит на консоль таблицу умножения заданного размера

```
int size = 5;
for (int i = 1; i <= size; i++)
{
    for (int j = 1; j <= size; j++)
    {
        std::cout << i * j << "\t";
    }
    std::cout << std::endl;
}
```

```
int n = 6;
for (int i = 1; i <= n; i++)
{
    int j = 0;
    while (j < n)
    {
        std::cout << i * j << "\t";
        j++;
    }
    std::cout << std::endl;
}
```

Спасибо за внимание!