Циклы в C++

Виды циклов

Циклы — это разновидность условных конструкций. Они позволяют выполнять команды несколько раз, если определённое утверждение верно.

- Цикл с параметрами (**for()**)
- Цикл с предусловием (**while()**)
- Цикл с постусловием(**do … while()**)

Особенности циклов

1. Код, который находится далее в фигурных скобках (**{}**) после служебного слова (наименования используемого самим языком, в данном случае **for**, **while**, **do**), называется **кодом находящимся в теле цикла**.

2. В любом цикле используется **Счетчик цикла** — это целочисленная переменная, которая объявляется с единственной целью: считать, сколько раз выполнился цикл. Один шаг цикла называется **итерацией,** а счётчик — итератором. Поэтому чаще всего для счётчика создаётся переменная i. (jjj или kkk или «реальное имя»*)*

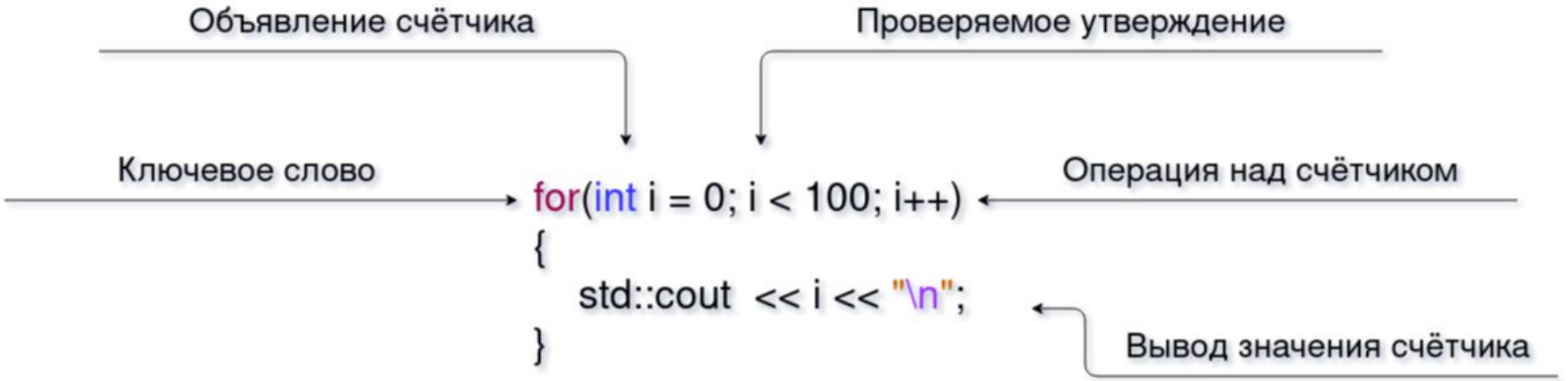
Правило: Всегда используйте тип signed int для счетчиков цикла.
3. Важно! Следите за тем, чтобы выход из цикла был возможен, иначе он будет выполняться бесконечно.

Цикл с параметрами (for())(синтаксис)

for (действие до начала цикла; условие продолжения цикла; действия в конце каждой итерации цикла)
 {
   инструкция цикла;
   инструкция цикла 2;
   инструкция цикла N;
 }

Или

for (объявление переменных; условие; инкремент/декремент счетчика)
тело цикла;



Цикл for применяется, если тело цикла необходимо выполнить определенное число раз

Переменные, определенные внутри цикла for, имеют специальный тип области видимости: область видимости цикла. Такие переменные существуют только внутри цикла и недоступны за его пределами

Описание синтаксиса

- Сначала присваивается **первоначальное значение** счетчику, после чего ставится точка с запятой.
- Затем задается **конечное значение счетчика цикла**. После того, как значение счетчика достигнет указанного предела, цикл завершится. Снова ставим точку с запятой.
- Задаем шаг цикла. **Шаг цикла** — это значение, на которое будет увеличиваться или уменьшаться счетчик цикла при каждом проходе.

Работа for

Цикл выполняется в следующем порядке:

1. инициализация счетчика;
2. проверка условия — если результат равен **false** — цикл завершается;
3. тело цикла;
4. изменение счетчика;
5. проверка условия — если результат равен **false** — цикл завершается;
6. тело цикла;
7. изменение счетчика;
8. … (и так далее пока при проверке условия не будет получен **false**).

Пример программы с for (инкремент счетчика)

Задача: вывести цифры от 0 до 9

```
#include <iostream>


int main()
{
    for (int count = 0; count < 10; ++count)
        std::cout << count << " ";


    return 0;
}
```

Результат: 0 1 2 3 4 5 6 7 8 9

Пример программы с for (декремент счетчика)

```
#include <iostream>

int main()
{
	for (int count = 8; count >= 0; --count)
		std::cout << count << " ";

	return 0;
}
```

Результат: 8 7 6 5 4 3 2 1 0

Пример программы с for (счетчик уменьшается/увеличивается больше чем на 1)

```
#include <iostream>

int main()
{
    for (int count = 9; count >= 0; count -= 2)
        std::cout << count << " ";

    return 0;
}
```

Результат: 9 7 5 3 1

Особенности for: Пропущенные выражения в цикле

```
#include <iostream>


int main()
{
    int count = 0;
    for (; count < 10; )
    {
        std::cout << count << " ";
        ++count;
    }
```

Особенности for: объявления нескольких переменных в цикле for

```
#include <iostream>

int main()
{
    int aaa, bbb;
    for (aaa = 0, bbb = 9; aaa < 10; ++aaa, --bbb)
        std::cout << aaa << " " << bbb << std::endl;

    return 0;
}
```

Результат выполнения программы:

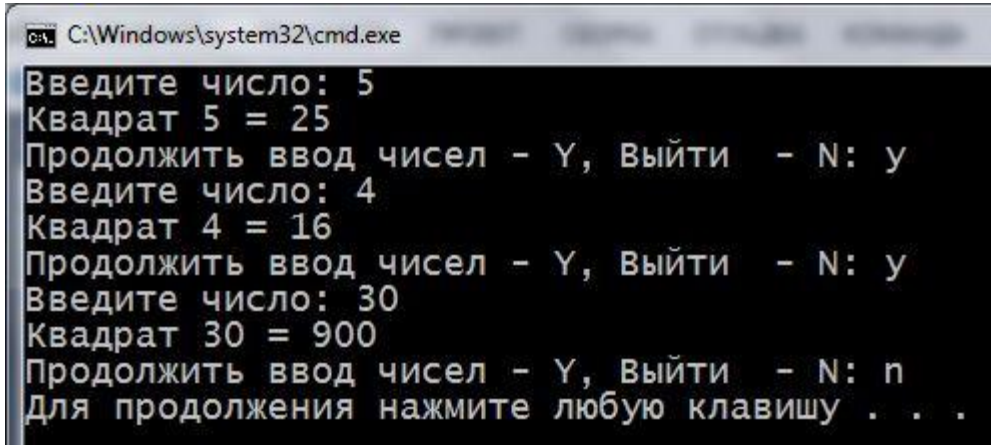
```
0 9
1 8
2 7
3 6
4 5
5 4
6 3
7 2
8 1
9 0
```

Задача на выход из цикла

Написать программу, которая будет показывать на экран квадрат числа, введенного пользователем. Пользователь должен сам решать – выйти из программы или продолжить ввод. (Подсказка – необходимо запустить бесконечный цикл, в котором предусмотреть его прерывание, при наступлении определённого условия).

```cpp
#include <iostream>
using namespace std;

int main()
{
setlocale(LC_ALL, "rus");
 int digit = 0; // число для расчета
char exit = 'y'; // для выхода или продолжения
 for (;;)
{
cout << "Введите число: ";
cin >> digit;
cout << "Квадрат " << digit << " = " << digit * digit;
cout << "\nПродолжить ввод чисел - Y, Выйти  - N: ";
cin >> exit; // выбор пользователя
if (exit != 'y' && exit != 'Y')
break; // прервать цикл
}
return 0;
}
```



Задача 1

Даны натуральные числа от 35 до 87. Вывести на консоль те из них, которые при делении на 7 дают остаток 1, 2 или 5.

Задача 2

Найдите количество четных цифр данного натурального числа

Задача 3

Найдите наибольшую цифру данного натурального числа

Задача 4

Найдите все четырехзначные числа, сумма цифр каждого из которых равна 15.

Задача 5

Напишите программу, в которой пользователь вводит с консоли число, а программа вычисляет факториал этого числа и выводит на консоль.

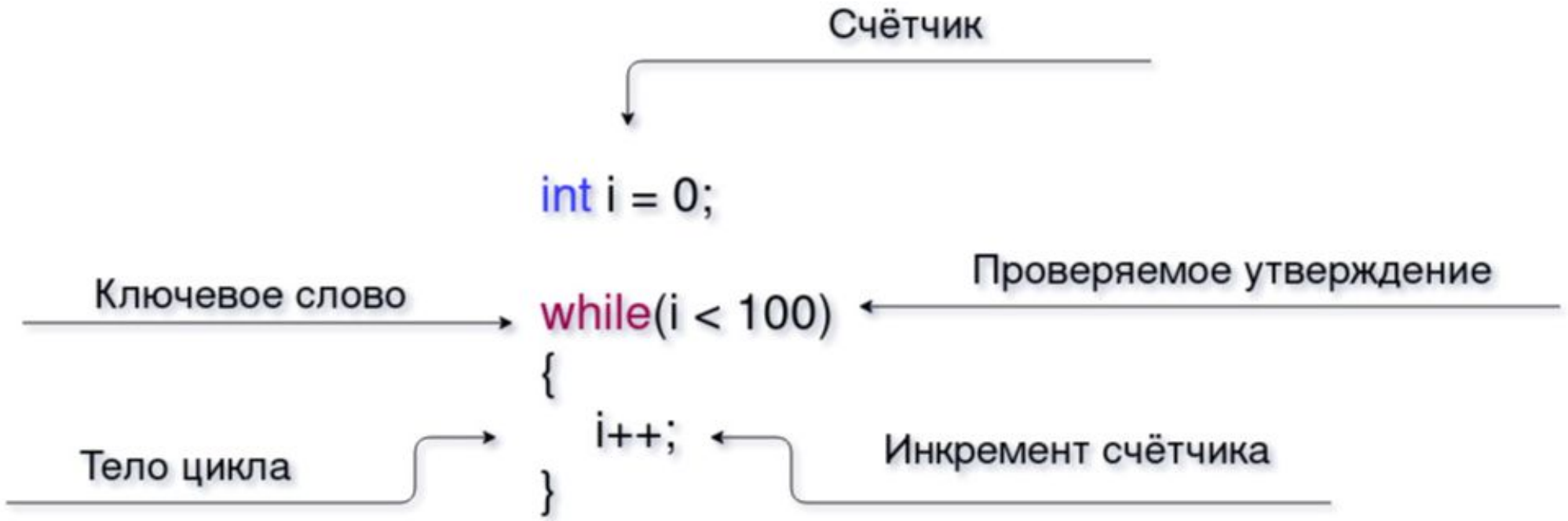
Условные циклы - Цикл с предусловием while (синтаксис)

while (Условие) { Тело цикла;}

или

while (условие)
    тело цикла;

Цикл будет выполняться, пока условие, указанное в круглых скобках является истиной



Условные циклы - Цикл с предусловием while

1. Цикл while объявляется с использованием **ключевого слова while**
2. В начале цикла обрабатывается **условие**
3. Если его значением является **true** (любое ненулевое значение), то тогда выполняется **тело цикла**
4. После завершения выполнения **тела цикла,** управление возвращается обратно к **while** и процесс проверки условия повторяется.
5. Если условие опять является **true**, то тогда **тело цикла** выполняется еще раз.

Пример программы с while

Задача: вывести цифры от 0 до 9

```
#include <iostream>

int main()
{
    int count = 0;
    while (count < 10)
        {
        std::cout << count << " ";
        ++count;
        }
    std::cout << "done!";

    return 0;
}
```

Результат выполнения программы:

```
0 1 2 3 4 5 6 7 8 9 done!
```

Особенности циклов: когда объявлять переменные

Любые переменные, объявленные внутри тела цикла, создаются, а затем и уничтожаются по новой!

```
#include <iostream>
 int main()
{
   int count = 1;
   int result = 0; // переменная result определена здесь, поскольку она нам понадобится позже (вне тела цикла)
   while (count <= 6) // итераций будет 6
   {
      int z; // z создается здесь по новой с каждой итерацией

      std::cout << "Enter integer #" << count << ':';
      std::cin >> z;

      result += z;

      // Увеличиваем значение счетчика цикла на единицу
      ++count;
   } // z уничтожается здесь по новой с каждой итерацией
   std::cout << "The sum of all numbers entered is: " << result;

   return 0;
}
```

Пример программы с while

Задача: вывести числа от 0 до 50. Числа первого десятка с 0 впереди, каждый новый десяток начать с новой строчки

```
#include <iostream>
 int main()
{
   int count = 1;
   while (count <= 50)
   {
      // Выводим числа до 10 (перед каждым числом добавляем 0)
      if (count < 10)
         std::cout << "0" << count << " ";
      else
         std::cout << count << " "; // выводим остальные числа
      // Если счетчик цикла делится на 10 без остатка, то тогда вставляем символ новой строки
      if (count % 10 == 0)
         std::cout << "\n";
      // Увеличиваем значение счетчика цикла на единицу
      ++count;
   }
   return 0;
}
```

Результат выполнения программы:

```
01 02 03 04 05 06 07 08 09 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49 50
```

Задача 1

Дано целое число, не меньшее 2. Выведите его наименьший натуральный делитель, отличный от 1.

Задача 2

Дано натуральное число N. Выведите слово YES, если число N является точной степенью двойки, или слово NO в противном случае.

Задача 3

Определите сумму всех элементов последовательности, завершающейся числом 0.

Задача 4

Последовательность состоит из натуральных чисел и завершается числом 0. Определите, какое количество элементов этой последовательности, равны ее наибольшему элементу.

Задача 5

Вклад в банке составляет x рублей. Ежегодно он увеличивается на p процентов, после чего дробная часть копеек отбрасывается. Каждый год сумма вклада становится больше. Определите, через сколько лет вклад составит не менее y рублей.

Программа получает на вход три натуральных числа: x, p, y и должна вывести одно целое число.