
ОПЕРАТОРЫ ЦИКЛА C++

***ВСЕ В НАШЕЙ ЖИЗНИ ЦИКЛИЧНО... ДНИ НЕДЕЛИ,
МЕСЯЦЫ, ГОДА... И В ПРОГРАММИРОВАНИИ ТОЖЕ
НЕ ОБОЙТИСЬ БЕЗ ЭТОГО...***

ЦИКЛ WHILE И

ЦИКЛ FOR

Рассмотрим, так называемые циклы , а именно цикл **while** и цикл **for** . Для чего они нужны? Для того, чтобы ответить на этот вопрос давайте представим себе такую ситуацию: ну к примеру вы хотите напечатать на экране свое имя ровно 20 раз. Как вы уже знаете для этого мы должны воспользоваться оператором консольного вывода (**cout** - поместить в поток).

Вот такая строка кода должна выводить ваше имя:

```
1. cout << "Your name" << endl;
```

ДЛЯ ТОГО, ЧТОБЫ ВЫВЕСТИ ИМЯ 20 РАЗ НАМ НУЖНО НАПИСАТЬ В СВОЕЙ ПРОГРАММЕ ЭТУ СТРОКУ РОВНО 20 РАЗ! ХОРОШО, ЕСЛИ ВЫ РЕШИЛИ ВЫВЕСТИ СВОЕ ИМЯ 20 РАЗ, А НЕ, К ПРИМЕРУ, 100 ИЛИ 1000! КОНЕЧНО ЖЕ, ВЫ ПОНИМАЕТЕ, ЧТО ЭТО ОЧЕНЬ НАКЛАДНО И НЕРЕАЛЬНО. В ЭТОМ СЛУЧАЕ НА ПОМОЩЬ ПРОГРАММИСТУ ПРИХОДЯТ ЦИКЛЫ **WHILE** И **FOR**, КОТОРЫЕ ПОЗВОЛЯЮТ ВЫПОЛНЯТЬ ОПРЕДЕЛЕННЫЙ УЧАСТОК КОДА (В НАШЕМ СЛУЧАЕ ЭТО ПЕЧАТЬ ИМЕНИ) СТОЛЬКО РАЗ, СКОЛЬКО МЫ УКАЖЕМ. ИНЫМИ СЛОВАМИ, ПОКА НЕ ВЫПОЛНИТСЯ КАКОЕ-ЛИБО УСЛОВИЕ, ТРЕБУЕМОЕ ДЛЯ ЗАВЕРШЕНИЯ ЦИКЛА. В НАШЕМ СЛУЧАЕ ЭТИМ УСЛОВИЕМ, КАК ВЫ УЖЕ ДОГАДЫВАЕТЕСЬ, ЯВЛЯЕТСЯ ЧИСЛО 20, Т.К. ПЕЧАТАТЬ НАМ НУЖНО ИМЕННО 20 РАЗ.

РАССМОТРИМ

ЦИКЛ

WHILE

ЦИКЛ *WHILE*

С ПРЕДУСЛОВИЕМ

СИНТАКСИС ДАННОГО ОПЕРАТОРА ЦИКЛА ТАКОВ:

1. **while** (условие выполнения цикла)
2. {
3. оператор 1;
4. оператор 2;
5. ...
6. оператор n;
7. }

ИТАК, ПРОГРАММА, ПЕЧАТАЮЩАЯ ИМЯ, БУДЕТ ВЫГЛЯДЕТЬ ТАК:

```
01. //операторы циклов - цикл while
02.
03. #include <iostream>
04.
05. using namespace std;
06.
07. int main()
08. {
09.     int i;
10.
11.     i = 1;
12.     while (i <= 20)
13.     {
14.         cout << «Your name» << endl;
15.         i = i + 1;
16.     }
17.
18.     return 0;
19. }
```

МЫ ОПРЕДЕЛЯЕМ ПЕРЕМЕННУЮ **i**, КОТОРАЯ У НАС БУДЕТ СЛУЖИТЬ СЧЕТЧИКОМ, ЗАДАЕМ НАШЕМУ СЧЕТЧИКУ НАЧАЛЬНОЕ ЗНАЧЕНИЕ РАВНОЕ ЕДИНИЦЕ ПЕРЕД ВХОДОМ В ЦИКЛ. В ЦИКЛЕ ОПРЕДЕЛЯЕМ УСЛОВИЕ, ПРИ КОТОРОМ У НАС БУДЕТ ОН РАБОТАТЬ, Т.Е. ВЫПОЛНЯТЬСЯ ЗАКЛЮЧЕННЫЕ В НЕГО ОПЕРАТОРЫ. ПОСЛЕ ТОГО, КАК УСЛОВИЕ ПЕРЕСТАНЕТ ВЫПОЛНЯТЬСЯ, ЦИКЛ ЗАВЕРШИТСЯ И ПРОГРАММА ВЫЙДЕТ ИЗ НЕГО И ПЕРЕЙДЕТ К ВЫПОЛНЕНИЮ СЛЕДУЮЩИХ ПОСЛЕ ЦИКЛА ОПЕРАТОРОВ (У НАС ЭТО ОПЕРАТОР **RETURN 0;**)

ЧТОБЫ ЦИКЛ РАБОТАЛ ДОЛЖНО ВЫПОЛНЯТЬСЯ
УКАЗАННОЕ В НЕМ УСЛОВИЕ **(У НАС ЭТО $i \leq 20$)**,
КОГДА **i** СТАНОВИТСЯ РАВНЫМ **21**, ТО ОПЕРАТОРЫ
ЗАКЛЮЧЕННЫЕ В ТЕЛО ЦИКЛА УЖЕ ВЫПОЛНЯТЬСЯ НЕ
БУДУТ. ЧТОБЫ ЭТО УСЛОВИЕ ВЫХОДА ИЗ ЦИКЛА
КОГДА-НИБУДЬ СРАБОТАЛО, ИНАЧЕ ПРОИЗОЙДЕТ
ЗАЦИКЛИВАНИЕ И ПРОГРАММА НИКОГДА НЕ
ЗАВЕРШИТЬСЯ (ПРИДЕТСЯ АВАРИЙНО ЗАВЕРШАТЬ ЕЕ).
ДЛЯ ЭТОГО У НАС ЕСТЬ СЧЕТЧИК, КОТОРЫЙ В КАЖДОЙ
ИТЕРАЦИИ (ОДНО ВЫПОЛНЕНИЯ ЦИКЛА) УВЕЛИЧИВАЕТ
СВОЕ ЗНАЧЕНИЕ НА ЕДИНИЦУ.

•Важно!
Запомнить:

программы, заключенной в его тело (между `{}`), столько раз - сколько мы укажем, либо пока не наступит определенное условие его завершения.

Обязательно нужно предусмотреть условие выхода из цикла, иначе произойдет ошибка зацикливания программы.

В программировании C++ существует возможность увеличения значения переменной на единицу с помощью оператора инкремента (`++`).

ЦИКЛ ***WHILE***

С

ПОСТУСЛОВИЕМ

ЦИКЛ **WHILE** С ПРЕДУСЛОВИЕМ МЫ РАССМОТРЕЛИ,
НО ЕСТЬ ЕЩЕ И ВТОРОЙ ВАРИАНТ ИСПОЛЬЗОВАНИЯ
ЦИКЛА **WHILE** – ЭТО ЦИКЛ С ПОСТУСЛОВИЕМ.
СИНТАКСИС ЕГО ТАКОВ:

```
1.  do  
2.  {  
3.      оператор 1;  
4.      оператор 2;  
5.      ...  
6.      оператор n;  
7.  }  
8.  while (условие выполнения цикла); //не забывайте про  
точку с запятой
```

-
- **Перепишем нашу программу с использованием данной разновидности цикла **while**:**

01. //операторы циклов - цикл while

02.

03. #include <iostream>

04.

05. using namespace std;

06.

07. int main()

08. {

09. //инициализируем переменную значением
1 при объявлении

10. int i = 1;

11.

12. do

13. {

14. cout << «Your name» << endl;

15. i++;

16. }

17. while (i <= 20);

18.

19. return 0;

20. }

-
- В чем же разница!? Разница одна лишь в том, что тело (содержимое) такого цикла **while** выполнится как минимум один раз, т.к. условие теперь уже проверяется не в начале цикла, а в конце.

ВАЖНО!

ЗАПОМНИТЬ:

1. ЦИКЛ **WHILE** БЫВАЕТ ДВУХ ТИПОВ: С ПРЕДУСЛОВИЕМ И ПОСТУСЛОВИЕМ. В ПЕРВОМ ВАРИАНТЕ ТЕЛО ЦИКЛА МОЖЕТ НИ РАЗУ НЕ ВЫПОЛНИТЬСЯ, ЕСЛИ УСЛОВИЕ НАЧАЛА ЦИКЛА НЕ ВЫПОЛНЯЕТСЯ. ВО ВТОРОМ ВАРИАНТЕ ТЕЛО ЦИКЛА ВЫПОЛНИТЬСЯ КАК МИНИМУМ ОДИН РАЗ, Т.К УСЛОВИЕ НАЧАЛА УЖЕ ВТОРОЙ ИТЕРАЦИИ ЦИКЛА ПРОВЕРЯЕТСЯ НЕ В НАЧАЛЕ, А В КОНЦЕ.
2. В КОНЦЕ ЦИКЛА **WHILE** С ПОСТУСЛОВИЕМ НЕ ЗАБЫВАЙТЕ СТАВИТЬ ТОЧКУ С ЗАПЯТОЙ.

Операторы

ЦИКЛОВ:

цикл for

РАССМОТРИМ ВТОРУЮ РАЗНОВИДНОСТЬ ЦИКЛОВ, ИСПОЛЬЗУЕМЫХ В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ **C++** - ЭТО ЦИКЛ **FOR**. ЧЕМ ЖЕ ОН ОТЛИЧАЕТСЯ? ГЛАВНОЕ ЕГО ОТЛИЧИЕ В ТОМ, ЧТО В ПАРАМЕТРАХ САМОГО ЦИКЛА МОЖНО ИНИЦИАЛИЗИРОВАТЬ (ОБЪЯВИТЬ) ПЕРЕМЕННУЮ-СЧЕТЧИК, КОТОРАЯ У НАС БУДЕТ СЧИТАТЬ ИТЕРАЦИИ (ПОВТОРЕНИЯ) ЦИКЛОВ, И ЗАДАТЬ ЕЙ НАЧАЛЬНОЕ ЗНАЧЕНИЕ, ЗАДАТЬ УСЛОВИЕ ВЫПОЛНЕНИЯ ЦИКЛА И ИЗМЕНЯТЬ ЗНАЧЕНИЕ СЧЕТЧИКА.

СИНТАКСИС ЦИКЛА FOR ТАКОВ:

1. **for** (начальное значение ; условие выполнения ; приращение)
2. {
3. оператор 1;
4. оператор 2;
5.
6. оператор n;
7. }

ТЕПЕРЬ ПЕРЕИНАЧИМ ПРОГРАММУ, ПЕЧАТАЮЩУЮ ВАШЕ ИМЯ, ПОД ЭТОТ ЦИКЛ:

```
01. //операторы циклов - цикл for
02.
03. #include <iostream>
04.
05. using namespace std;
06.
07. int main()
08. {
09.     for (int i = 1; i <= 20; i++)
10.     {
11.         cout << «Your name» << endl;
12.     }
13.
14.     return 0;
15. }
```

**РАЗМЕР КОДА ПРОГРАММЫ У НАС УМЕНЬШИЛСЯ,
СТАЛ БОЛЕЕ КОМПАКТНЫМ. ЛЕГКО
ПРОСМАТРИВАЮТСЯ ПО ПАРАМЕТРАМ ЦИКЛА **FOR**
НАШИ НАЧАЛЬНЫЕ ЗНАЧЕНИЯ, УСЛОВИЯ И
ПРИРАЩЕНИЯ, ЧТО ОЧЕНЬ УДОБНО.**

ЭЛЕКТРОННЫЙ РЕСУРС

[HTTP://IGUANIA.RU/UCHEBNIK-PO-OSNOVAM-PROGRAMMIROVANIYA-DLYA-NACHINAIOUSCHICH/GLAVA-9-VSE-V-NASHEY-ZHIZNI-TSIKLICHNO.HTML](http://IGUANIA.RU/UCHEBNIK-PO-OSNOVAM-PROGRAMMIROVANIYA-DLYA-NACHINAIOUSCHICH/GLAVA-9-VSE-V-NASHEY-ZHIZNI-TSIKLICHNO.HTML)