

Циклы. Цикл for

- Циклом в языке C++ является условная обработка выражения с действием в конце каждой операции.
- Ниже представлена синтаксис:

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

Цикл for

- Итерацией цикла называется один проход этого цикла
- Существует частный случай этой записи, который мы сегодня и разберем:

```
for (счетчик = значение; счетчик < значение; шаг цикла) {  
    тело цикла;  
}
```

Счетчик цикла — это переменная, в которой хранится количество проходов данного цикла.

описание синтаксиса цикла for

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

Пример:

- Напишем программу, которая будет считать сумму всех чисел от 1 до 1000

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

int main()
{
    int i; // счетчик цикла
    int sum = 0; // сумма чисел от 1 до 1000.
    setlocale(0, "");
    for (i = 1; i <= 1000; i++) // задаем начальное значение 1, конечное 1000 и задаем шаг
        цикла - 1.
    {
        sum = sum + i;
    }
    cout << "Сумма чисел от 1 до 1000 = " << sum << endl;
    return 0;
}
```

- Если мы скомпилируем этот код и запустим программу, то она покажет нам ответ: 500500. Это и есть сумма всех целых чисел от 1 до 1000. Если считать это вручную, понадобится очень много времени и сил. Цикл выполнил всю рутинную работу за нас.
- Конечное значение счетчика задано не строгим неравенством. В случае если бы был указан знак «меньше» – программа выполнила бы 999 операций, как следствие была бы ошибка.
- Значение шага цикла я задал равное единице. $i++$ — это тоже самое, что и $i = i + 1$
- В теле цикла при каждом проходе переменная `sum` увеличивает свое значение на 1 единицу.
- Обратите внимание на то, что при инициализации этой переменной ей задали начальное значение = 0
- Если бы я этого не сделал, в переменной мог оказаться «мусор» в виде хаотичных данных.
- Запомните - при объявлении переменной без ее инициализации что эта переменная будет хранить «мусор».

Цикл while

- Когда мы не знаем, сколько итераций должен произвести цикл, нам понадобится цикл **while** или **do...while**. Синтаксис цикла **while** в C++ выглядит следующим образом

```
while (Условие) {  
    Тело цикла;  
}
```

- Данный цикл будет выполняться, пока условие, указанное в круглых скобках является истиной. Решим ту же задачу с помощью цикла **while**. Хотя здесь мы точно знаем, сколько итераций должен выполнить цикл, очень часто бывают ситуации, когда это значение неизвестно
- Ниже приведен исходный код программы, считающей сумму всех целых чисел от 1 до 1000.

Пример программы с циклом while:

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

int main()
{
    setlocale(0, "");
    int i = 0; // инициализируем счетчик цикла.
    int sum = 0; // инициализируем счетчик суммы.
    while (i < 1000)
    {
        i++;
        sum += i;
    }
    cout << "Сумма чисел от 1 до 1000 = " << sum << endl;
    return 0;
}
```

Разбор выполнения программы

- После компиляции программа выдаст результат, аналогичный результату работы предыдущей программы. Но поясним несколько важных моментов. Я задал строгое неравенство в условии цикла и инициализировал счетчик i нулем, так как в цикле **while** происходит на одну итерацию больше, потому он будет выполняться, до тех пор, пока значение счетчика перестает удовлетворять условию, но данная итерация все равно выполнится. Если бы мы поставили нестрогое неравенство, то цикл бы закончился, когда переменная i стала бы равна 1001 и выполнилось бы на одну итерацию больше
- Теперь давайте рассмотрим по порядку исходный код нашей программы. Сначала мы инициализируем счетчик цикла и переменную, хранящую сумму чисел
- В данном случае мы обязательно должны присвоить счетчику цикла какое-либо значение, т.к. в предыдущей программе мы это значение присваивали внутри цикла **for**, здесь же, если мы не инициализируем счетчик цикла, то в него попадет «мусор» и компилятор в лучшем случае выдаст нам ошибку, а в худшем, если программа соберется — сегфолт практически неизбежен

Разбор выполнения программы

- Затем мы описываем условие цикла – **«пока переменная i меньше 1000 — выполняй цикл»**
- При каждой итерации цикла значение переменной счетчика i увеличивается на 1
- Когда выполнится 1000 итераций цикла, счетчик станет равным 999 и следующая операция уже не выполнится т.к. 1000 не меньше 1000.
- Выражение $sum += i$ это укороченная запись $sum = sum + i$.

Цикл do while

Do while очень похож на while, различие в том, что при выполнении do while один проход цикла будет выполнен независимо от условия. Рассмотрим тот же пример:

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

int main ()
{
    setlocale(0, "");
    int i = 0; // инициализируем счетчик цикла.
    int sum = 0; // инициализируем счетчик суммы.
    do { // выполняем цикл.
        i++;
        sum += i;
    } while (i < 1000); // пока выполняется условие.
    cout << "Сумма чисел от 1 до 1000 = " << sum << endl;
    return 0;
}
```

задание

- Создать свой первый проект, назвав его своей Ф.И.группа.
- Добавить в проект исходный файл, назвав его Ф.
- Переписать в него код с примеров и откомпилировать поочередно.
- К каждому примеру сделать скриншот работы консольного окна.

- Написать программу которая будет делить на 2 переменные а и б, если их разность $>$ либо $= 1$. Операцию деление а и б применить не более 5-ти раз подряд. Если разность оказалась $<$ либо $= 1$ в таком случае вывести в консоль сообщение и указать количество выполненных циклом итераций.
Исходные данные:
Вариант 1 (а = 36, б = 18), Вариант 2 (а = 23, б = 21). Каждый выполняет оба варианта.
Программа должна выводить в консоль результат каждой итерации и конечный ответ
- Создать отчет по данной работе.

Отчет (Фамилия_№лабораторной.doc)

- Титульный лист

Тема работы: «Изучение работы циклов».

- Отчет

Цель работы: Научиться работать с циклами в C++.

Задачи: 1. Создать проект. 2. Написать программу на языке программирования C++ и продемонстрировать ее работоспособность, используя примеры из лекции.

3. Выполнить задание.

Ход работы.

<Ваша работа, шаги, рисунки>

Вывод.

*Отчет присылаете на почту.

**Требования к оформлению отчета ГОСТ.