Lesson 2

Break, continue операторы. Switch. Циклы in C++. Класс String. Полезные методы работы с классом String. (+harvard task)

Break + continue операторы.

Оператор **break** предназначен для искусственного прерывания выполнения:

1) *последовательность итераций в операторах цикла for, while или do-while;*
2) *последовательность итераций в операторе switch.*

Чаще всего **break** используется в сочетании с оператором **if**. В этом случае проверяется некоторое условие и, если оно выполняется, вызывается break.

Оператор **continue** предназначен для перехода к выполнению следующей итерации цикла. Таким образом, если есть continue, то:

1) *Выполнение текущей итерации прекращается;*
2) *Цикл переходит к следующей итерации цикла.*

Оператор **continue** может использоваться во всех видах циклов: for, while и do-while.

Введение в циклы.

Цикл в программировании - это как **повторяющаяся** лента в плеере, которую компьютер выполняет снова и снова, пока не выполнит определенное условие. Самое главное, что цикл позволяет нам писать **меньше кода**, делая программу более **компактной** и **эффективной**.

Основная задача цикла - повторять определенные действия до тех пор, пока выполняется определенное условие. *Например*, мы можем использовать цикл, чтобы **прочитать все письма** в почтовом ящике, пока не прочтем все, или чтобы **вывести на экран числа** от 1 до 1000.

**В C++ есть несколько видов циклов.*

Циклы C++.

Существует 3 вида циклических операторов:

1) цикл **for**
2) цикл **while** с предусловием
3) цикл **do-while** с постусловием

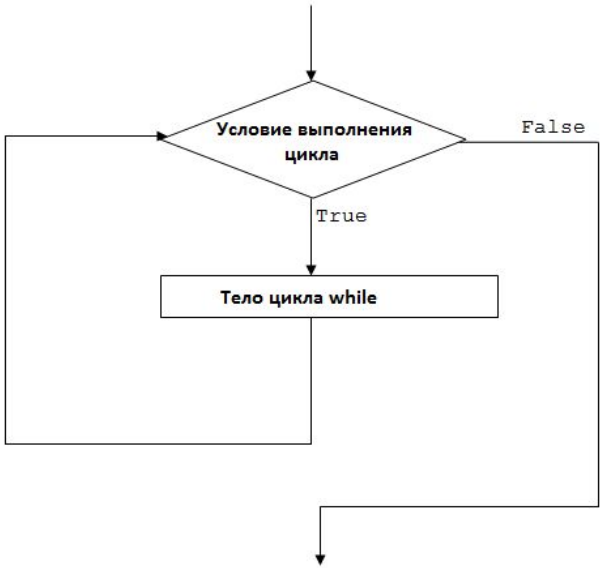
Каждый из этих циклов имеет свои особенности использования.

Цикл **for** может иметь очень широкое разнообразие реализаций и применений. Цикл **for** также называют циклом с параметрами.

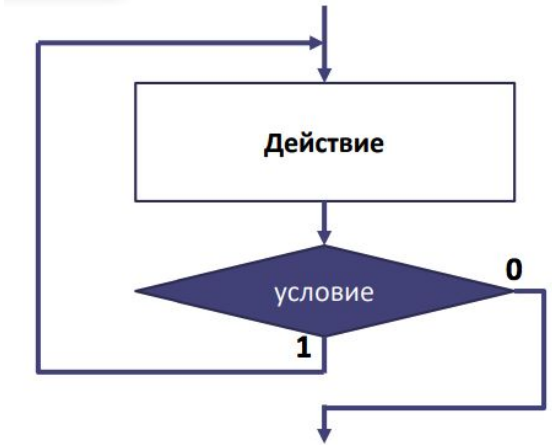
Цикл **while** называется циклом с предусловием.

Цикл **do-while** полезен, если необходимо выполнить итерацию хотя бы один раз.

Блок-схема. while.



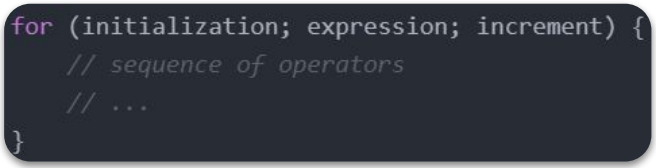
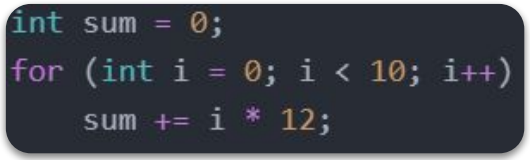
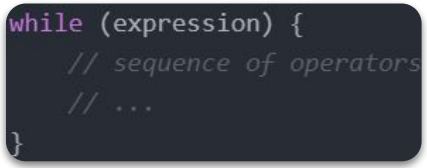
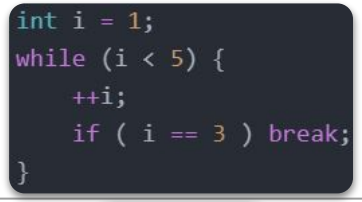
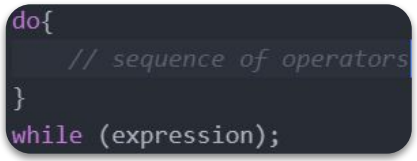
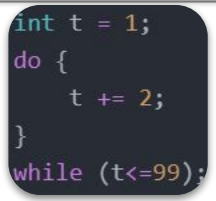
Блок-схема. do-while.



**действие* - это блок do.
**условие* - while(..)

1 - true
0 - false

Примеры исп-я циклов.

FOR		
WHILE		
DO-WHILE		

Задачи while

1) Просто выведите все числа от 1 до 100, используя цикл **while**;

2) Выведите все степени числа 2, которые меньше 1000, используя цикл **while**.

3) Посчитайте сумму цифр введенного пользователем числа, используя цикл **while**.

4) Выведите цифры введенного числа в обратном порядке, используя цикл **while**.

5) Сгенерируйте и выведите 10 случайных чисел в диапазоне от 1 до 100, используя цикл **while**.

```
#include <cstdlib> // Для функции rand()
srand(static_cast<unsigned int>(time(0))); // инициализация генератора случ. чисел
rand() % 100 + 1;
```

Задачи for

1) Выведите таблицу умножения для определенного числа (например, 7) от 1 до 10, используя цикл **for**.

7 x 1 = 7

7 x 2 = 14

...

7 x 10 = 70

2) Подсчитайте сумму всех чисел от 1 до 500, используя цикл **for**.

3) Выведите все четные числа от 2 до 80, используя цикл **for**.

4) Рассчитайте факториал числа 5 (5!), используя цикл **for**. (например факториалы **3!** = 3 * 2 * 1; **4!** = 4 * 3 * 2 * 1; **5!** = 5 * 4 * 3 * 2 * 1)

5) Выведите все нечетные положительные числа от -100 до 100, используя цикл **for**.

Задачи do-while

1) Выведите все четные числа от 20 до 2 в обратном порядке, используя цикл **do-while**.

2) Запрашивайте у пользователя ввод чисел до тех пор, пока он не введет 0, и выводите сумму введенных чисел. Используйте цикл **do-while**.

3) Запросите у пользователя ввод пароля. Если он введет неверный пароль, повторяйте запрос до тех пор, пока он не введет правильный пароль. Используйте цикл **do-while**. (*пароль целое число - пин код*)

4) Реализуйте игру "Угадай число", в которой пользователь должен угадать число от 1 до 10. Программа должна продолжать запрашивать ввод, пока пользователь не угадает число. Используйте цикл **do-while**.

String class.

Класс string предназначен для работы со строками типа char*, которые представляют собой строки с завершающим нулем. Для использования возможностей класса string необходимо подключить библиотеку <string> и пространство имен std.

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

Объявление переменной типа string производится так же, как и обычной переменной.

```
string str = "Hello, world";
string s;
```

Методы работы с String.

1) **assign**() - присвоение одной строки другой
2) **append**() - объединение строк
3) **insert**() - вставка одной строки в указанную позицию другой строки
4) **erase**() - удаление символов
5) **length**() - длина строки
6) **empty**() - проверка на наличие пустой строки
7) **find**() - поиск строки в подстроке
8) **replace**() - замена символов в вызывающей строке
9) **reverse**(string.begin(), string.end()); - для переворота строки

Задачи String

1) Напишите программу, которая принимает строку от пользователя и выводит ее в обратном порядке.
2) Реализуйте программу, которая считает количество гласных букв в введенной строке.
3) Проверьте, является ли введенная строка палиндромом (читается одинаково слева направо и справа налево).
4) Подсчитайте количество символов в введенной строке.
5) Подсчитайте количество слов в введенной строке. Слова разделяются пробелами.
6) Замените все пробелы в введенной строке на символ подчеркивания ('_').
7) Преобразуйте все буквы в верхний регистр в введенной строке.
8) Попросите пользователя ввести две строки и проверьте, равны ли они. Выведите сообщение о результате.
9) Позвольте пользователю ввести две строки, затем объедините их в одну и выведите результат.

HARVARD TASK.

1) Вывести на экран треугольник из символов (*).

a)

The switch selection operator.

Оператор **switch** позволяет в зависимости от значения выражения выбрать один вариант решения задачи из нескольких вариантов. Таким образом, обеспечивается разнонаправленное ветвление в программе. Синтаксис оператора switch выглядит следующим образом:

```
switch (expression)
{
    case const1:
        sequence_operators1;
    break;
    case const2:
        sequence_operators2;
    break;

    ...

    ...

    case constN:
        sequence_operatorsN;
    break;
    default
        sequence_operators;
}
```

Задачи Switch

1) Напишите программу, которая запрашивает у пользователя номер дня недели (1 - Понедельник, 2 - Вторник, и так далее) и выводит соответствующее название дня.

2) Реализуйте простой калькулятор с использованием **switch**. Пользователь вводит два числа и оператор (+, -, *, /). Программа выполняет соответствующую операцию и выводит результат.

3) Создайте программу, которая принимает оценку от пользователя (от 1 до 5) и выводит оценку в текстовом виде ("Отлично", "Хорошо", "Удовлетворительно", "Неудовлетворительно"). Используйте **switch** для обработки различных случаев.

Thank you for your attention!