

C++ Starter

№ урока: 6 **Курс:** Циклические конструкции

Средства обучения: Персональный компьютер с установленной Visual Studio

Обзор, цель и назначение урока

Целью урока является рассмотрение циклических конструкций и их видов.

Изучив материал данного занятия, учащийся сможет:

- Понимать работу циклических операторов.
- Понимать работу операторов безусловного перехода.
- Применять циклические конструкции **while**, **do-while**, **for**.
- Работать с вложенными циклами.

Содержание урока

1. Обзор циклических конструкций.
2. Рассмотрение цикла со счетчиком **for**.
3. Рассмотрение примеров: Использование цикла со счетчиком **for**.
4. Рассмотрение вложенных циклов.
5. Рассмотрение примера: Использование вложенных циклов.
6. Рассмотрение цикла с предусловием **while**.
7. Рассмотрение примеров: Использование цикла с предусловием **while**.
8. Рассмотрение цикла с постусловием **do-while**.
9. Рассмотрение примеров: Использование цикла с постусловием **do-while**.
10. Рассмотрение оператора безусловного перехода **goto**.
11. Рассмотрение примеров: Оператор безусловного перехода **goto**.
12. Аргументы за и против использования оператора **goto**.
13. Бесконечные циклы.

Резюме

- Цикл — это управляющая конструкция, предназначенная для организации многократного исполнения набора инструкций.
- Также циклом может называться любая многократно исполняемая последовательность инструкций, организованная любым способом (например, с помощью условного перехода).
- Итерация — это один проход цикла.
- Цикл со счетчиком **for** — это цикл, в котором переменная — счетчик итераций цикла, с определенным шагом, изменяет свое значение до заданного конечного значения.
- Блок выражений цикла **for**, содержит три выражения: **for** (начальное-выражение; условное-выражение; выражение-цикла) { тело цикла } или в другой нотации это звучит так: **for** (инициализация; условие; модификация) { тело цикла }
- В теле цикла **for**, разрешено изменение значения начального выражения (т.е. счетчика итераций).
- Цикл, с предусловием **while** — это цикл, который выполняется до тех пор, пока условие удовлетворяет истинности. Условие проверяется до выполнения тела цикла. Если изначально условие не удовлетворяет истинности, то тело цикла **while** ни разу не выполнится.
- Цикл, с постусловием **do-while** — это цикл, в котором условие проверяется после выполнения тела цикла. Отсюда следует, что тело **do-while** выполняется хотя бы один раз.

- **Вложенные циклы** – это циклы, организованные в теле другого цикла. Вложенный цикл в тело, другого цикла, называется внутренним циклом. Цикл, в теле которого существует вложенный цикл, называется **внешним**.
- Полное число исполнений внутреннего цикла, всегда равно произведению числа итераций внутреннего цикла на произведение чисел итераций всех внешних циклов.
- **goto** (англ. go to — «перейти к») — оператор безусловного перехода к определённой точке программы, обозначенной номером строки либо меткой.
- В 1968 году, в своей статье «Обоснование пагубности оператора **goto**», Эдсгер Дейкстра отметил, что качество кода обратно пропорционально количеству **goto**, используемых в коде. Дейкстра утверждал (это просто его мнение), что:
 - корректность кода без **goto** доказать легче
 - код с операторами **goto** трудно форматировать
 - **goto** влияет на логическую структуру программы
 - применение **goto** препятствует оптимизации, выполняемой компилятором
 - **goto** усложняет анализ кода
- На практике применение оператора **goto**, приводит к нарушению принципа хода алгоритма строго сверху вниз.
- Сторонники **goto** выступают за осторожное применение оператора, при определенных условиях.
- Большинство аргументов против **goto**, не говорит о полном отказе от этого оператора, а предостерегает от неразборчивого его использования.
- Хорошее программирование не означает исключение всех **goto**.
- Стремление к коду без **goto** не должно быть целью.
- Десятилетия исследований оператора **goto** не смогли подтвердить его вредоносность, а теоретические и экспериментальные доводы, выдвигаемые против этого оператора, оказались не убедительными.
- И наконец, операторы **goto** входят во множество современных языков.

Закрепление материала

- Что такое цикл?
- Перечислите известные Вам циклические конструкции.
- Где и для чего используются циклические конструкции?
- Что делает оператор **goto**?
- Кто и почему рекомендовал не использовать оператор **goto**?
- Значения, какого типа можно передавать в качестве параметра **while()**?
- Что такое итерация?
- В чем разница между циклом **while** и **do-while**?
- Для чего используются служебные слова **continue** и **break**?

Дополнительное задание

Задание

Используя Visual Studio, создайте проект по шаблону Console Application.

Создайте две целочисленные переменные и задайте им некоторые значения. Применяя технику вложенных циклов, нарисуйте прямоугольник из звездочек. Используйте значения ранее созданных переменных для указания высоты и ширины прямоугольника.

Самостоятельная деятельность учащегося

Задание 1

Выучите основные конструкции и понятия, рассмотренные на уроке.

Задание 2

Используя Visual Studio, создайте проект по шаблону Console Application.

Дано два числа А и В ($A < B$) выведите сумму всех чисел расположенных между данными числами на экран.
Дано два числа А и В ($A < B$) выведите все нечетные значения, расположенные между данными числами.

Задание 3

Используя Visual Studio 2010, создайте проект по шаблону Console Application.

Используя циклы и оператор: `cout << "*" , cout << " " , cout << endl` (для перехода на новую строку).

Выведите на экран:

- прямоугольник
- прямоугольный треугольник
- равносторонний треугольник
- ромб

Задание 4

Имеется N клиентов, которым компания производитель должна доставить товар. Сколько существует возможных маршрутов доставки товара, с учетом того, что товар будет доставлять одна машина?

Используя Visual Studio, создайте проект по шаблону Console Application.

Напишите программу, которая будет рассчитывать, и выводить на экран количество возможных вариантов доставки товара. Для решения задачи, используйте факториал $N!$, рассчитываемый с помощью цикла `do-while`.

Рекомендуемые ресурсы

Цикл for

<http://cppstudio.com/post/348/>

Цикл while

<http://cppstudio.com/post/352/>

Цикл do-while

<http://cppstudio.com/post/361/>

Оператор break

<http://cppstudio.com/post/364/>

Оператор continue

<http://cppstudio.com/post/4271/>