

№ урока: 7 **Курс:** Функции**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, что такое функция.
- Понимать различия между процедурами и функциями.
- Понимать, что такое прототип функции.
- Понимать работу функций.

Содержание урока

1. Обзор функций.
2. Различие между функциями и процедурами.
3. Правила создания функций.
4. Рассмотрение примера: Создание функции
5. Рассмотрение понятия прототипа, его назначение и применение.
6. Создание функции и ее описание с помощью прототипа.
7. Рассмотрение управляющей структуры **return**.
8. Какие параметры функций бывают.

Резюме

- Функция — это именованная часть программы, которая может вызываться из других частей программы столько раз, сколько необходимо.
- О функциях и процедурах. В некоторых языках программирования (например, в Паскале) функции и процедуры (подпрограммы, не возвращающие значения) чётко разграничены синтаксисом языка. В языке C++, — процедуры являются частным случаем (подмножеством) функций, возвращающими значение типа **void** — пустое значение.
- В C++ прежде чем определить функцию необходимо создать прототип функции.
- Прототип — это программная конструкция в языке C++, которая описывает интерфейс функции для компилятора (возвращаемое значение, имя, тип и количество параметров).
- По стандарту, каждая C++ программа должна иметь прототипы к функциям, которые в ней используются.
- Для определения функции необходимо:
 - Указать тип возвращаемого значения, если функция ничего не возвращает указать **void**.
 - Выбрать имя функции в соответствии с действием, которое она выполняет.
 - Если функция принимает аргументы — обязательно указать их тип и имя, если нет — оставить аргументные скобки пустыми или указать в них **void**.
 - Если функция возвращает значение, обязательно в теле функции должно присутствовать ключевое слово **return**. Тип возвращаемого значения функции должен соответствовать типу значения, указанному после ключевого слова **return**.
- Для вызова функции необходимо:
 - Написать имя функции.
 - Обязательно указать после имени аргументные скобки (), если функция принимает какие-то аргументы, передать соответствующее количество аргументов соответствующего типа.

- Оператор **return**— это управляющая структура, которая позволяет программе в нужный момент завершить работу функции. В результате функция завершается через нормальный канал выхода, возвращая управление вызывающей функции.
- Используйте **return**, если это повышает читабельность метода.
- Используйте **return**, как сторожевой оператор досрочного выхода.
- Функции могут возвращать значения вызывающим их объектам. Если тип возвращаемого значения, указываемый перед именем функции, не равен **void**, для возвращения значения используется ключевое слово **return**.
- В результате выполнения инструкции с ключевым словом **return**, после которого указано значение нужного типа, вызвавшей функции будет возвращено это значение.
- Ключевое слово **return** останавливает выполнение функции.
- Если тип возвращаемого значения **void**, инструкцию **return** без значения все равно можно использовать для завершения выполнения функции.
- Если ключевое слово **return** отсутствует, выполнение функции завершится, когда будет достигнут конец ее блока кода.
- Для возврата значений функциями с типом возвращаемого значения отличным от **void** необходимо обязательно использовать ключевое слово **return**.
- Чтобы использовать возвращаемое функцией значение в вызывающей функции, вызов функции можно поместить в любое место кода, где требуется значение соответствующего типа.
- Возвращаемое значение функции можно присвоить переменной.
- Функции, которые возвращают логическое значение, называют функциями предикатами.
- Параметры принято разделять на формальные параметры и фактические параметры.
- В языке C++ разделяют глобальную и локальную области видимости.
- Локальная область ограничивается блоком функции, или вложенным блоком (переменные, объявленные в локальной области доступны только в ее пределах).
- Глобальная область видимости ограничивается файлом, в котором объявлена переменная (переменная доступна всем составляющим программы, которые находятся после объявления переменной).

Закрепление материала

- Что такое функция?
- Для чего используются функции?
- Что такое прототип функции и для чего он нужен?
- Какие виды параметров у функций могут быть?
- Что такое глобальная область видимости?
- Что такое локальная область видимости?

Дополнительное задание

Задание

Используя Visual Studio, создайте проект по шаблону Win32 Console Application.

Создайте функцию, которая принимает два целочисленных параметра и выводит на экран их сумму.

Самостоятельная деятельность учащегося

Задание 1

Выучите основные конструкции и понятия, рассмотренные на уроке.

Задание 2

Используя Visual Studio, создайте проект по шаблону Win32 Console Application.

Создайте функцию с именем Hello(), которая выводит на экран приветствие к пользователю. Реализуйте подсчет количества вызовов функции Hello().

Задание 3

Используя Visual Studio, создайте проект по шаблону Win32 Console Application.

Создайте функцию, которая принимает 3 параметра и возвращает среднее арифметическое. В главной функции выведите возвращаемое значение функции с тремя аргументами на экран.

Задание 4

Используя Visual Studio, создайте проект по шаблону Win32 Console Application.

Создайте функции, которые выполняют логические операции над двумя операндами.

Рекомендуемые ресурсы

Функции в C++

<http://cppstudio.com/post/396/>