Содержание

[ОБЩАЯ ИНФОРМАЦИЯ 2](#_heading=h.gjdgxs)

[Тема занятия](#_heading=h.30j0zll) 2

[Цели и задачи 2](#_heading=h.1fob9te)

[Ожидаемый результат 2](#_heading=h.3znysh7)

[Структура занятия 3](#_heading=h.tyjcwt)

[ОПИСАНИЕ ЗАНЯТИЯ 4](#_heading=h.3dy6vkm)

# ОБЩАЯ ИНФОРМАЦИЯ

### Тема занятия

Функции. Процедурное программирование.

### Цели и задачи

* Познакомиться с понятием процедурного программирования
* Научиться работать с функциями

### Ожидаемый результат

По результатам занятия Слушатель должен знать:

* Что такое функция
* Какие существуют подходы к решению задач в программировании

По результатам занятия Слушатель должен уметь:

* Понимать и разбираться в видах функций
* Использовать функций для решения поставленных задач

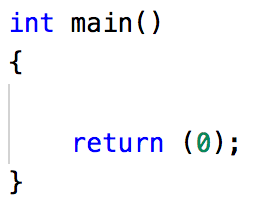
### Структура занятия

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Тайминг занятия** | |  |  |  | Таблица 1 |
| № | Этапы | Что делает преподаватель | Что делает Слушатель | Время | Общее время |
| 1 | Приветственное слово преподавателя |  |  | 5 мин. | 5 мин. |
| 2 | Повторение пройденного |  |  | 5 мин. | 40 мин. |
| 3 | Теоретическая часть |  |  | 30 мин. |
| 4 | Вопросы по теоретической части |  |  | 5 мин. |
|  | *Перерыв* |  |  | 15 мин. | 15 мин. |
| 5 | Практическая часть |  |  | 40 мин. | 45 мин. |
| 6 | Рефлексия и вопросы |  |  | 5 мин. |

# ОПИСАНИЕ ЗАНЯТИЯ

**Теоретическая часть**

****

****

**Очень важно объяснить разницы функциональной зависимости от функции в программировании, так как часто ученики путают понятия**

**Функции**

Функция это группа выражений, которая выполняет конкретное задание.

Функции в программировании - это изолированный блок кода, обращение к которому в процессе выполнения программы может быть многократным.

Функции используют, чтобы сократить объём исходного кода: рационально вынести часто повторяющиеся выражения в отдельный блок и, затем, по мере надобности, обращаться к нему.

Вы можете определять ваши собственные функции в C++.

Использование функций может дать множество преимуществ, включая следующие:

- Вы можете использовать код многократно, используя функции.

- Вы можете легко тестировать отдельные функции.

- Если есть необходимость сделать некоторые изменения в программе, вы можете модифицировать отдельную функцию, не изменяя структуру программы.

- Вы можете использовать одну и ту же функцию для ввода разной информации.

В каждой исправной программе на языке C++ должна быть хотя бы одна функция - функция main().

“Точка входа в программу”

**Возвращаемый тип**

Функция main имеет следующий общий вид.

Возвращаемый тип функции объявляется до ее имени. В вышеуказанном примере возвращаемый тип int, который указывает, что функция возвращает целочисленное значение.

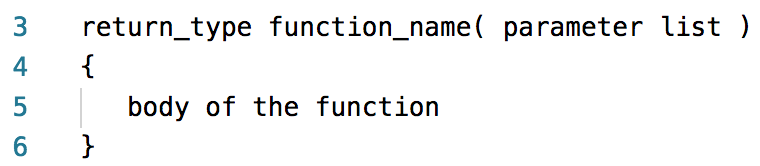
Иногда, функция выполняет необходимые операции без возвращения значения. Такие функции определяются с помощью ключевого слова void.

**Возвращаемый тип**

void это базовый тип данных, имеющий пустое множество значений.

**Определяем функцию**

Для определения функции в языке C++ используется следующий синтаксис:



**return-type**(возвращаемый тип): Тип данных возвращаемый функцией.

**function name**: Имя функции.

**parameters**(список параметров): При вызове функции, вы

передаете значения в параметр. Это значение ссылается на актуальный параметр или аргумент. Для каждого параметра указывается его тип и имя (в объявлении имена можно опускать). Параметры опциональны; это значит, что у вас может быть функция без параметров.

**body of the function**(тело функции): Набор выражений, который определяет, что делает функция.

**Определяем функцию**

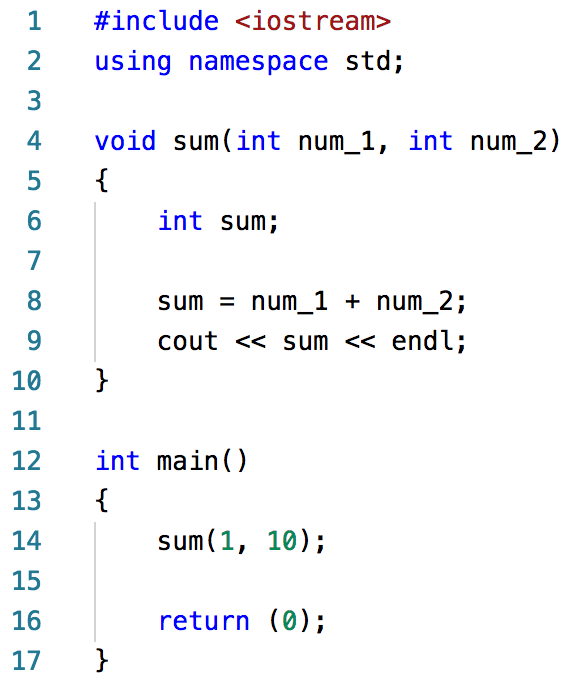
Для примера, давайте определим функцию, которая не возвращает значение, а только выводит строку текста на экран.

Наша функция, названная print\_hello, возвращает void и не имеет параметров.

Теперь мы используем нашу функцию в функции main().

Вы должны объявить функцию перед ее вызовом.

**Размещение объявления функции после main() приведет к ошибке.**

****

Давайте создадим простую функцию, которая возвращает сумму двух параметров.

Функция sum принимает два параметра типа int, и возвращает int.

**Множество параметров**

Вы можете определить в функции столько параметров, сколько вы захотите, разделяя их запятыми.

(Суммирование не двух чисел, а трёх, четырёх и т д)

Размещение объявления функции после main() приведет к ошибке.

НО!

Объявление функции, или прототип функции сообщает компилятору имя функции и как ее вызвать. Тело функции может быть объявлено отдельно.

Объявление(declaration) функции необходимо при ее определении(definition) в одном исходном файле и вызове в другом файле. В таком случае, вам необходимо объявить функцию вверху файла, вызывающего функцию.

**Практическая часть**

**Задание 1**

Написать программу, в которой будет функция (помимо функции main), которая принимает 2 аргумента (2 целочисленных числа) и вывод в стандартный выходной поток (печатает на экран) частное двух чисел.

\*

Обработать случай нулей

**Задание 2**

Программа должна переводить число, введенное с клавиатуры в метрах, в километры (создать отдельную функцию!)

**Задание 3**

Представьте, что вы создаёте собственную игру. Вам нужно реализовать печать карты на экран (под картой в вашей игре понимается прямоугольник с заданными размерами, ограниченный символом ‘#’) Напишите функцию, которая будет реализовывать данное поведение.

void print\_map(int width, int height);

\*

Добавьте возможность выбора символа границы карты (ещё один аргумент в функции)

**Задание 4**

Реализовать шифр Цезаря. Написать функцию, которая будет применять шифр Цезаря к строке, переданной в качестве параметра функции. Реализовать для букв нижнего регистра. Смещение по алфавиту - 10 позиций

\*Для верхнего регистра

**Задание 5**

Улучшить функцию из задания 2. На вход функции подают целое число - значение, на которое и происходит сдвиг по алфавиту.

\*Реализовать функцию дешифровки (расшифровать строку обратно)