

# Введение в C++

## Лекция 4: Функции

Антон Гуликов

Санкт-Петербургский государственный университет

11 Декабря, 2015

# План

## 1 Функции

# Мотивация

Структурируем программы в сегменты кода, для выполнения отдельных задач.

# Функции

Множество операторов, имеющих имя, которые можно вызвать в некоторой точке программы.

$type_f \ name_f(type_1 \ par_1, type_2 \ par_2, \dots) \{ statements \}$

- $type_f$  - тип значения, возвращаемое функцией
- $name_f$  - имя функции
- $type_1, par_1$  - тип и значения параметров
- $statements$  - тело функции

# Функции без типов

Можно рассматривать как процедуру.

- *type<sub>f</sub>* - void
- *return* statement - return

# Main

- Вызывается в начале программы
- Если возвращает 0, то воспринимается, что все завершилось успешно.
- Другие значения могут быть восприняты, как сообщения об ошибках.

# Передача значения по ссылке и по значению

- Передача по значению - содержимое переменных копируется в переменные, представленные параметрами функции.
- Иногда полезно иметь доступ к внешним переменным, внутри функций. Для этого передадим их по ссылке.
- Если есть желание передавать по ссылке и гарантировать неизменяемость переменных, можно передать по константной ссылке.

# Inline function

- Вызов функции требует некоторых затрат
- Может быть удобно просто вставить кусок кода в тело программы, чем вызывать функцию
- Можно сказать компилятору, что мы хотим, чтобы наша функция по возможности была вставлена в код
- Оптимизацией занимается компилятор - ему виднее



# Значения по умолчанию

В C++ есть опциональные аргументы у функций. Они могут находиться в конце списка параметров функций, и используются, когда при вызове функции участвуют меньшее количество параметров.

# Рекурсия

- Рекурсия - свойство функции вызывать саму себя
- Используется почти везде
- Никогда не забывайте об условия выхода из рекурсии

# Задачи и материалы лекции

<https://www.hackerrank.com/contests/for-switch/challenges>

<https://github.com/antongulikov/vkunicpp/4>