

C++ Advanced

Название курса: C++ Advanced

Количество уроков: 10

Продолжительность в часах: 5 часов

Продолжительность в днях: 5 дней

Описание курса

Курс Advanced C++ состоит из 10-ти уроков, на которых будут рассмотрены продвинутые возможности языка программирования C++ и его стандартной библиотеки. Особое внимание будет уделено тонким аспектам использования тех или иных функций языка C++, что позволит писать эффективный и понятный код на этом языке программирования.

После прохождения курса, Вы будете понимать, как эффективно использовать продвинутые возможности языка C++.

Предварительные требования

- Базовые знания языка программирования C++ (указатели, ссылки, классы, шаблоны и т.п.)

Целевая аудитория

Курс рассчитан на программистов среднего уровня, желающих получить лучшее понимание продвинутых возможностей языка программирования C++ и, возможно, получить некоторые новые знания.

По завершении курса Вы сможете

- Точно понимать принципы автоматического вывода типов C++.
- Эффективно использовать стандартные контейнеры и алгоритмы и создавать свои аллокаторы при необходимости.
- Использовать параллельное программирование для ускорения выполнения программ.
- Понять семантику перемещения и прямой передачи.
- Применять шаблоны программирования в языке C++.
- Понять принципы мета-программирования C++.

Список уроков курса:

Урок 1: Вывод типов при использовании шаблонов, auto и decltype.

- Вывод типов C++
- Вывод типов шаблонов
- Вывод типов auto
- decltype
- Совместное использование auto и decltype
- Средства определения типов переменных

Урок 2: Функции.

- Что такое функторы?
- Указатели на функции и их недостатки по сравнению с функторами
- `std::function`
- Лямбда выражения (C++11, C++14)

Урок 3: Стандартная библиотека шаблонов. Контейнеры.

- Типы контейнеров
 - Последовательные контейнеры
 - Ассоциативные контейнеры
 - Неупорядоченные ассоциативные контейнеры
 - Адаптеры контейнеров
- Структуры данных, использующиеся в стандартных контейнерах
- Выбор подходящих контейнеров

Урок 4: Стандартная библиотека шаблонов. Итераторы и алгоритмы.

- Типы итераторов
 - Вывода
 - Ввода
 - Однонаправленные
 - Двунаправленные
 - Произвольного доступа
- Типы алгоритмов
 - Немодифицирующие операции над последовательностями
 - Модифицирующие операции над последовательностями
 - Операции разделения
 - Операции сортировки
 - Операции двоичного поиска
 - Операции над множествами
 - Операции над кучей
 - Операции минимума/максимума
 - Операции сравнения
 - Операции перестановки
 - Числовые операции
- Использование алгоритмов

Урок 5: Параллельное программирование. Потоки и средства их синхронизации.

- Основные принципы параллельного программирования
- Использование потоков `std::thread`
- Средства синхронизации потоков
 - `mutex`, `recursive_mutex`, `timed_mutex`, `recursive_timed_mutex`
 - `std::lock_guard`, `unique_lock`, `condition_variable`

Урок 6: Параллельное программирование. Задачи и параллельные алгоритмы.

- Параллельное программирование на основе задач
 - `async`
 - `future`, `promise`
- Стратегии запуска
- Параллельные алгоритмы стандартной библиотеки

Урок 7: Rvalue-ссылки, семантика перемещения и прямая передача.

- Типы ссылок C++
- Семантика перемещения
- Использование прямой передачи

Урок 8: Шаблоны программирования и C

- Типы шаблонов программирования
 - Поведенческие
 - Порождающие
 - Структурные
- Реализация шаблонов в языке C++
- Рекомендации по использованию шаблонов в C++

Урок 9: Метапрограммирование.

- Что такое метапрограммирование
- Метапрограммирование в языке C++
- Советы по использованию метапрограммирования в C++

Урок 10: Полезные нововведения C++ 17

- «Синтаксический сахар» C++ 17
- Нововведения шаблонов
- Новые возможности лямбда выражений
- Новые атрибуты
- Новые полезные классы стандартной библиотеки
- Декомпозиция при объявлении
- Атрибуты `nodiscard`, `fallthrough`, `maybe_unused`
- `string_view`
- `optional` и `variant`
- `std::filesystem`

Урок 11: Работа с файлами

- base input/output stream classes
- `fstream`, `ofstream` `ifstream`
- `filebuf`
- базовые операции при работе с файлами