



C++ Developer. Basic

Базовый курс C++

Длительность курса: 112 академических часов

1 Вводное занятие

Цели занятия:

знакомство с курсом.

Краткое содержание:

рассмотрим на занятии:

- состав курса по модулям и по занятиям
- где брать материалы
- самостоятельные работы – где брать условия и как происходит процедура сдачи
- к кому обращаться при возникновении проблем или вопросов
- где происходит общение слушателей курса друг с другом и с преподавателями
- проектная работа

Домашние задания

1 Создание github-репозитория

Цель: Создаём github-репозиторий, который нам пригодится на протяжении всего обучения на курсе.

Что требуется сделать:

1. Выполнить задание, описанное в 01.homework.pdf
2. Отправить на проверку в "Чат с преподавателем" все, что указано в файле 01.homework.pdf

2 Пишем свой "Hello, World". Учимся собирать простые проекты

Цели занятия:

Учимся собирать простые проекты

Краткое содержание:

Начнём с простой классической программы вывода строки "Hello, World!" в консоль. На примере этого простого проекта разберём такие вопросы как:

- структура файлов с исходным кодом в языке C++
- подключение заголовочных файлов

- базовое понимание функции main
- несколько базовых деталей о синтаксисе языка C++ (именование переменных; перечень зарезервированных ключевых слов; необходимость ; в конце строки)

Одной из главных проблем начинающих в языке C++ является компиляция. На примере простого проекта научимся сборке C++ исходников:

- сборка из консоли с использованием g++ (linux)
- сборка из консоли с использованием MSBuild (Windows)

Далее познакомимся с более высокоуровневыми инструментами. Тоже пока на базовом уровне:

- MS Visual Studio (Windows)
- make и Makefile (linux)
- CLion (linux)
- VSCode (linux и Windows)

Поймём, насколько это все неудобно с точки зрения кросс-платформенной разработки и под конец познакомимся с инструментом CMake.

Домашние задания

1 Hello, World!

Цель: Настраиваем окружение для дальнейшей работы

Что требуется сделать:

1. Выполнить задание, описанное в 02.homework.pdf
 2. Отправить на проверку в "Чат с преподавателем" все, что указано в файле 02.homework.pdf
-

3 Базовый синтаксис языка C++

Цели занятия:

подробное знакомство с синтаксисом C++.
Рассматриваем основные языковые конструкции, доставшиеся нам из языка Си, на базовом уровне используем стандартную библиотеку.

Краткое содержание:

Обзор основных типов данных, переменных, циклов, ветвлений, структур, функций, простейшего ввода-вывода. Поговорим о конструкциях языка if-else, while, for, do-while, упомянем недобрым словом goto, поймём ценность функций и научимся передавать функциям аргументы. Восхитимся возможностью объединения разнотипных данных в структуры. Научимся выводить наши данные на экран или в файл.

4 Такие разные функции

Цели занятия:

подробнее поговорим о функциях.

Краткое содержание:

Рассмотрим следующие темы:

- функции
 - аргументы и возвращаемые значения
 - рекурсия
 - перегрузка
 - ссылки и указатели
 - стек вызовов
 - выделение памяти
-

5 Модульность. Линковка

Цели занятия:

обсудим вопросы модульности в C++.

Краткое содержание:

Вынесем объявления функций в отдельный заголовочный файл, подключим его в разные единицы трансляции.

Проследим за ошибками, разберёмся в их причинах. Далее создадим статическую библиотеку и подключим её. Разберёмся со сборкой и линковкой.

И под конец обсудим разницу между статической библиотекой и динамической.

Домашние задания

1 Игра "угадай число"

Цель: Закрепим полученные знания и навыки и немного развлечёмся.

Что требуется сделать:

1. Выполнить задание, описанное в 03.homework.pdf
 2. Отправить на проверку в "Чат с преподавателем" все, что указано в файле 03.homework.pdf
-

6 Подробнее о CMake

Цели занятия:

изучив вопрос модульности в C++, а также получив базовые знания об использовании сторонних библиотек, можем более подробно изучить CMake.

Краткое содержание:

В этом занятии рассмотрим:

- как собирать с помощью CMake-а проекты из нескольких модулей
 - как создавать собственные библиотеки – статические и динамические
 - как подключать сторонние библиотеки к своему проекту
-

7 Ответы на вопросы

Цели занятия:

ответить на все вопросы, возникшие по ходу изучения первого модуля

Краткое содержание:

В модуле 1 мы рассмотрели все основные и важные вопросы работы на языке C++.

На текущий момент слушатели курса имеют возможность реализовать практически любую логику на языке C++ в своих приложениях.

Однако, так как времени на детальное рассмотрение каждой темы у нас было маловато, в этом вебинаре мы займемся обсуждением конкретных моментов и вопросов слушателей.

1 Структуры и классы

Цели занятия:

посмотрим, чем же всё-таки С++ отличается от языка С.

Краткое содержание:

Рассмотрим, что такое структуры и классы, методы, конструкторы и деструкторы, наследование, перегрузка методов и наследование реализации.

Постигнем дзен перегрузки операторов.

Коснёмся страшного (но востребованного) понятия ООП.

Домашние задания

1 Инженерный калькулятор

Цель: Погружаемся в ООП, разрабатываем приложение для расчёта математических функций

Что требуется сделать:

1. Выполнить задание, описанное в 04.homework.pdf

2. Отправить на проверку в "Чат с преподавателем" все, что указано в файле 04.homework.pdf

2 Полиморфизм и все-все-все

Цели занятия:

продолжим изучать C++ и ООП.

Краткое содержание:

Затронем вопросы - проектирование и ещё раз наследование, композиция и дружба, агрегация и владение.

Поймём силу наследования и полиморфизма, рассмотрим понятие интерфейса.

Вернёмся с небес наследования на землю композиции. Обсудим вопросы владения.

Домашние задания

- 1 Реализация расчёта статистики с использованием полиморфизма

Цель: Знакомимся с полезными статистическими характеристиками и тренируемся использовать полиморфизм для решения похожих задач.

Что требуется сделать:

1. Выполнить задание, описанное в 05.homework.pdf
 2. Отправить на проверку в "Чат с преподавателем" все, что указано в файле 05.homework.pdf
-

3 Шаблоны классов и функций

Цели занятия:

сложная, но полезная тема.

Поймём силу шаблонов, как одного из основных преимуществ языка C++.

Краткое содержание:

Рассмотрим темы:

- шаблоны классов,
 - шаблоны функций,
 - специализация шаблонов,
 - шаблонные параметры.
-

4 **Тонкости
(перегрузка
операторов,
конструкторы и
деструкторы,
move-
семантика)**

Цели занятия:

обсудим тонкости и нюансы для решения специфических проблем, а также для улучшения повседневного кода.

Краткое содержание:

перегрузка операторов, конструкторы и деструкторы, move-семантика

Домашние задания

1 Использование шаблонов - контейнеры

Что требуется сделать:

1. Выполнить задание, описанное в 06.homework.pdf
2. Отправить на проверку в "Чат с преподавателем" все, что указано в файле 06.homework.pdf

1 Понятие unit-тестирования, цель, что тестируем

Цели занятия:

изучим понятие unit-тестирование, поймём – зачем это нужно и какие преимущества даёт. Реализуем несколько тестов на код простыми способами.

Краткое содержание:

что есть суть unit-тесты, зачем тратить на них время, как это делать правильно, как их использовать.

2 Инструменты для unit-тестирования. Библиотека Googletest

Цели занятия:

познакомимся с библиотекой (framework) для написания unit-тестов.

Поймем, какие рутинные задачи она позволяет решать с легкостью.

Рассмотрим основные возможности.

Домашние задания

1 Unit-тестирование контейнеров

Что требуется сделать:

1. Выполнить задание, описанное в 07.homework.pdf

2. Отправить на проверку в "Чат с преподавателем" все, что указано в файле 07.homework.pdf

1 Общий обзор стандартной библиотеки

Цели занятия:

быстро пройдемся по всему часто используемому, чтобы знать, что там вообще есть.

Краткое содержание:

Посмотрим, какие имеются контейнеры в `std`.
Что такое ввод-вывод с точки зрения `std`.
Какие имеются инструменты для разработки многопоточных приложений.
Коснёмся некоторых модулей из раздела "Miscellaneous".

2 Подробнее о контейнерах и вводе-выводе

Цели занятия:

обсудим, почему в `std` так много контейнеров и чем они отличаются.

Краткое содержание:

Рассмотрим сложность выполнения операций различных контейнеров `std`.
Поговорим о вводе-выводе в языке C++.

3 Алгоритмы, которые лучше знать лучше

Цели занятия:

посмотрим, какие алгоритмы есть в стандартной библиотеке.
Поймём, как и когда их использовать.
Узнаем, где смотреть сложность этих алгоритмов.

1 Работа с библиотеками вручную

2 Современные средства автоматизации. Пакетные менеджеры, снова CMake

3 Сборка библиотек из исходников

1 Семантика перемещения, поехсерт, и как они нам помогают

2 Работа с динамической памятью

3 Исключения в С++.
Использование и нюансы.
Гарантии безопасности

Цели занятия:

поговорим про обработку ошибок и исключительных ситуаций.

Краткое содержание:

Рассмотрим, что такое исключения, чем они лучше (и чем хуже) возврата кода ошибки. Реализуем своё исключение и научимся его обрабатывать.

1	Понятие потока выполнения программы	Цели занятия: <ul style="list-style-type: none">- поток выполнения- планировщик задач операционной системы- класс <code>std::thread</code> <hr/>
2	Две главные проблемы - гонка и взаимная блокировка	Цели занятия: <ul style="list-style-type: none">- проблемы работы с разделяемыми ресурсами- понятие состояния гонки- способы решения - классы <code>std::mutex</code> и <code>std::lock_guard</code>- новая проблема - взаимная блокировка- пути решения deadlock-ов <hr/>
3	Структуры данных в многопоточной среде	Цели занятия: <ul style="list-style-type: none">- использование структур данных для связи потоков- особенности такого использования- <code>std::condition_variable</code> <hr/>
4	Более высокие абстракции и асинхронное исполнение	Цели занятия: <ul style="list-style-type: none">- <code>std::async</code>- <code>std::future/promise</code>- <code>std::packaged_task</code> <hr/>
5	Дополнительные тонкости и детали	Цели занятия: <ul style="list-style-type: none">- <code>std::once_flag</code> + <code>std::call_once</code>- <code>thread_local</code>- <code>execution policy</code> в алгоритмах- отладка многопоточных приложений

1 Идиомы и полезные практики C++

Цели занятия:

выбрать и утвердить темы проектов

Краткое содержание:

правила работы над проектом и специфика проведения итоговой защиты;
требования к результату проекта и итоговой документации.

2 Идиомы и полезные практики C++. Часть 2

3 Полезные в работе инструменты

1 Вводное занятие по проекту

Цели занятия:

выбрать и обсудить тему проектной работы;
спланировать работу над проектом;
ознакомиться с регламентом работы над проектом

Краткое содержание:

правила работы над проектом и специфика проведения итоговой защиты;
требования к результату проекта и итоговой документации.

Домашние задания

1 Проектная работа

Цель: выбрать тему проекта;
закрепить тему в чат с преподавателем;
защиты проект.

Предполагаемые этапы выполнения:

1. Проработанные требования к реализации
2. Планируемая архитектура
3. Полная итоговая реализация.

Сдавать можно как по частям, так и сразу итоговый вариант.

Не обязательно придерживаться указанному плану.

2 Консультация

Цели занятия:

получить ответы на вопросы по проекту, ДЗ и по курсу

Краткое содержание:

вопросы по улучшению и оптимизации работы над проектом;
затруднения при выполнении ДЗ; вопросы по программе.

**3 Защита
проектных
работ**

**4 Подведение
итогов группы**

Цели занятия:

узнать, как получить сертификат об окончании курса, как взаимодействовать после окончания курса с OTUS и преподавателями, какие вакансии и позиции есть для выпускников (опционально - в России и за рубежом) и на какие компании стоит обратить внимание.

Краткое содержание:

организационные вопросы;
рынок вакансий по направлению;
статистика курса и вопросы по курсу.