

# C++ Developer. Professional

Особенности разработки программного обеспечения на C++

Длительность курса: 138 академических часов

## 1 Современный C++

### 1 Система сборки: **build, test and deploy.**

познакомиться;  
понять, как будем работать;  
узнать, с кем и как общаться;  
понять, зачем нужна автоматизация сборки;  
расшифровывать аббревиатуру CI/CD;  
подготовить стенд для выполнения самостоятельных работ.

Домашние задания

#### 1 ДЗ-01. Сборка и публикация пакета. Фильтрация IP-адресов.

Цель: В это самостоятельной работе создаём сборочную ферму для работы по схеме CI/CD на протяжении всего курса.

Решаем проблему фильтрации IP-адресов, проверяя свой входной уровень знаний.

Используем созданную сборочную ферму для сборки и публикации проекта.

Что требуется сделать:

1. Выполнить задание, описанное в 01.homework.pdf
2. Отправить на проверку в "Чат с преподавателем":
  - ссылку на свой репозиторий на [github.com](https://github.com) с реализованным проектом;
  - ссылку на проект на [travis-ci.org](https://travis-ci.org) с историей (и хотя бы одной успешной) сборок;
  - ссылку на репозиторий [bintray.com](https://bintray.com), откуда можно скачать получившийся deb-пакет.

### 2 Особенности

освежить основы по C++11 (пригодится дальше по курсу);

	<b>C++11. auto, lambda, tuple</b>	узнать новые фишки C++11 (тоже пригодится); обсудить особенности, которые важно учитывать.
3	<b>Особенности C++11. Move, variadic templates</b>	освежить основы по C++11 (пригодится дальше по курсу); узнать новые детали; обсудить особенности, которые стоит учитывать.
4	<b>Сравнение C++11/C++14 с C++ 03</b>	освежить знания по C++14; узнать новые детали; обсудить особенности C++ 14, на которые стоит обратить внимание.
5	<b>Реализация умных указателей</b>	запомнить правило: "взял память - верни, когда больше не нужна"; понять, как следовать этому правилу; разобраться с разными типами умных указателей.
6	<b>Идея аллокаторов</b>	объяснить важность стратегий аллокации памяти; использовать разные аллокаторы, разобраться, как реализовать свой аллокатор; объяснить, как расширяется std::vector, и почему именно так.
		Домашние задания
		1 ДЗ-02. allocator
		Цель: В это самостоятельной работе реализуем собственный аллокатор, используем его в стандартных контейнерах. Пишем свой контейнер и пробуем использовать аллокатор.
		Что требуется сделать:
		1. Выполнить задание описанное в 02.homework.pdf
		2. На acm.timus.ru выполнить задачи: 1409, 1877, 2001, 1264, 1787
		3. Отправить на проверку в "Чат с преподавателем":
		- ссылку на свой репозиторий на github.com;
		- ссылку на проект на travis-ci.org;
		- ссылку на репозиторий bintray.com;
		- ссылку на аккаунт на acm.timus.ru со списком принятых задач.
7	<b>Обзор C++17. Constexpr lambda. Fold expression. Attributes. Type deduction</b>	разобраться с основными нововведениями C++ 17; понять, в каких случаях они используются; обойти несколько спорных моментов.
8	<b>Обзор C++17. If constexpr.</b>	разобраться с основными нововведениями C++17; понять кейсы использования;

**Structured bindings. Statements with initializer. std::filesystem**

обсудить несколько тонких моментов;  
пользоваться новым стандартом языка.

---

**9 Шаблонная магия**

перестать бояться шаблонов;  
пообещать себе перестать пользоваться макросами,  
читать STL;  
искать документацию.

### 1 Принципы проектирования ПО

разобраться, зачем нужны принципы; пользоваться принципами, читать UML диаграммы.

Домашние задания

#### 1 ДЗ-03. Doxygen

Цель: В это самостоятельной работе применяем полученные знания про шаблоны, реализуем функционал вывода IP-адреса в общем виде, подключаем к своему проекту doxygen и генерируем документацию.

Что требуется сделать:

1. Выполнить задание описанное в 03.homework.pdf

2. Решить первые 15 задач на [acm.timus.ru](http://acm.timus.ru) (от простых к сложным). Отсортированный по сложности список: <http://acm.timus.ru/problemset.aspx?space=1&page=all&skipac=False&sort=difficulty>

3. Сделать ревью решений ДЗ-01 двух коллег (которые его выполнили)

4. Опционально: предоставить примеры, из своей практики, использования Принципов Проектирования

5. Отправить на проверку в "Чат с преподавателем":

- ссылку на свой репозиторий на [github.com](https://github.com);

- ссылку на проект на [travis-ci.org](https://travis-ci.org);

- ссылку на репозиторий [bintray.com](https://bintray.com);

- ссылку на аккаунт на [acm.timus.ru](http://acm.timus.ru) со списком принятых задач;

- ссылки на выполненные ревью.

### 2 GRASP

обращать внимание на архитектурные детали кода; улучшать метрики кода; находить ошибки в коде коллег.

### 3 Шаблоны GoF. Порождающие

читать паттерны проектирования; использовать паттерны; объяснить ценность экспериментов.

Домашние задания

#### 1 ДЗ-04. Editor.

Цель: В это самостоятельной работе разрабатываем архитектуру проекта, применяем полученные знания принципов проектирования, используем паттерны, генерируем документацию.

Что требуется сделать:

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

2. На [acm.timus.ru](http://acm.timus.ru) решить любые 3 задачи с уровнем сложности от 100 до 150.  
Отсортированный по сложности список:  
<http://acm.timus.ru/problemset.aspx?space=1&page=all&skipac=False&sort=difficulty>  
Если выбрать сложно, есть ещё списки задач по темам (ближе к концу страницы):  
<http://acm.timus.ru/problemset.aspx>

3. Сделать ревью решений ДЗ-03 двух коллег (которые его выполнили).

4. Опционально: сделать ревью на ревью ДЗ-02.

5. Отправить на проверку в "Чат с преподавателем":  
- ссылку на свой репозиторий на [github.com](https://github.com);  
- ссылку на проект на [travis-ci.org](https://travis-ci.org);  
- ссылку на репозиторий [bintray.com](https://bintray.com);  
- ссылку на аккаунт на [acm.timus.ru](http://acm.timus.ru) со списком принятых задач;  
- ссылки на выполненные ревью;  
- опционально: ссылку на ревью на ревью.

---

4 **Шаблоны GoF.  
Структурные**

читать паттерны проектирования;  
использовать паттерны;  
объяснить ценность экспериментов.

---

5 **Шаблоны GoF.  
Поведенческие.  
Command,  
Interpreter,  
Iterator, Mediator,  
Memento, Chain of  
responsibility**

читать паттерны проектирования;  
использовать паттерны;  
объяснить ценность экспериментов.

Домашние задания

1 ДЗ-05. Matrix.

Цель: В это самостоятельной работе создаём проект бесконечной разряженной матрицы с использованием паттернов проектирования, не забываем про принципы и документацию.

Что требуется сделать:

1. Выполнить задание, описанное в 05.homework.pdf

2. На [acm.timus.ru](http://acm.timus.ru) решить любые 3 задачи с уровнем сложности от 151 до 200.

Отсортированный по сложности список:  
<http://acm.timus.ru/problemset.aspx?space=1&page=all&skipac=False&sort=difficulty>  
Если выбрать сложно, есть ещё списки задач по темам (ближе к концу страницы):  
<http://acm.timus.ru/problemset.aspx>

3. Сделать ревью решений ДЗ-03 двух коллег (которые его выполнили).

4. Опционально: сделать ревью на РЕВЬЮ ДЗ-03.

- 5. Отправить на проверку в "Чат с преподавателем":
    - ссылку на свой репозиторий на [github.com](https://github.com);
    - ссылку на проект на [travis-ci.org](https://travis-ci.org);
    - ссылку на репозиторий [bintray.com](https://bintray.com);
    - ссылку на аккаунт на [acm.timus.ru](https://acm.timus.ru) со списком принятых задач;
    - ссылки на выполненные ревью;
    - опционально: ссылку на ревью на ревью.
- 

6 **Шаблоны GoF. Поведенческие. Observer, State, Strategy, Template method, Visitor**

читать паттерны проектирования;  
использовать паттерны;  
объяснить ценность экспериментов.

---

7 **STL. Контейнеры**

изучить контейнеры STL;  
объяснить, какие типы контейнеров есть в STL;  
настраивать поведение контейнеров.

Домашние задания

1 ДЗ-06. cmd

Цель: В это самостоятельной работе реализуем проект обработки последовательностей команд фиксированного и произвольного размера. Стремимся к снижению зависимостей между модулями.

Что требуется сделать:

1. Выполнить задание, описанное в 06.homework.pdf

2. На [acm.timus.ru](https://acm.timus.ru) решить 3 любые задачи с уровнем сложности от 201 до 300.

Отсортированный по сложности список:

[http://acm.timus.ru/problemset.aspx?](http://acm.timus.ru/problemset.aspx?space=1&page=all&skipac=False&sort=difficulty)

[space=1&page=all&skipac=False&sort=difficulty](http://acm.timus.ru/problemset.aspx?space=1&page=all&skipac=False&sort=difficulty)

Если выбрать сложно, есть ещё списки задач по темам (ближе к концу страницы):

<http://acm.timus.ru/problemset.aspx>

3. Сделать ревью решений ДЗ-04 двух коллег (которые его выполнили).

4. Опционально: сделать ревью на ревью ДЗ-04.

5. Отправить на проверку в "Чат с преподавателем":

- ссылку на свой репозиторий на [github.com](https://github.com);
  - ссылку на проект на [travis-ci.org](https://travis-ci.org);
  - ссылку на репозиторий [bintray.com](https://bintray.com);
  - ссылку на аккаунт на [acm.timus.ru](https://acm.timus.ru) со списком принятых задач;
  - ссылки на выполненные ревью;
  - опционально: ссылку на ревью на ревью.
- 

8 **STL. Алгоритмы**

разобраться, почему алгоритмов в STL так много;  
объяснить, зачем нужны алгоритмы в STL;  
использовать алгоритмы.

---

9	<b>Boost. Разное</b>	объяснить философию проекта Boost; читать документацию Boost, выбирать и применять библиотеки7
10	<b>Boost. Паттерны проектирования</b>	изучить Boost; посмотреть на реализацию паттернов проектирования; расширить знания про библиотеку.  Домашние задания  1 ДЗ-07. Boost  Цель: В этой самостоятельной работе применяем на практике навыки использования библиотеки Boost.  Что требуется сделать: 1. Выполнить задание, описанное в 07.homework.pdf  2. Опционально: на acm.timus.ru решить 3 любые задачи с уровнем сложности от 401 до 450. Отсортированный по сложности список: <a href="http://acm.timus.ru/problemset.aspx?space=1&amp;page=all&amp;skipac=False&amp;sort=difficulty">http://acm.timus.ru/problemset.aspx?space=1&amp;page=all&amp;skipac=False&amp;sort=difficulty</a> Если выбрать сложно, есть ещё списки задач по темам (ближе к концу страницы): <a href="http://acm.timus.ru/problemset.aspx">http://acm.timus.ru/problemset.aspx</a>  3. Опционально. Сделать ревью решений ДЗ-05 двух коллег (которые его выполнили).  4. Опционально: сделать ревью на ревью ДЗ-01.  5. Отправить на проверку в "Чат с преподавателем": - ссылку на свой репозиторий на github.com; - ссылку на аккаунт на acm.timus.ru со списком принятых задач; - ссылки на выполненные ревью; - опционально: ссылку на ревью на ревью.

- |   |                                     |   |
|---|-------------------------------------|---|
| 1 | <b>Многопоточность</b>              | использовать потоки в своих приложениях;<br>решать основные проблемы многопоточности;<br>использовать инструменты стандартной библиотеки.   |
| 2 | <b>Шаблоны 5</b>                    | объяснить многопоточные паттерны;<br>использовать для решения задач.  |
| 3 | <b>Асинхронное программирование</b> | объяснить разницу между многопоточностью, параллельностью и асинхронностью;<br>выбирать подходящее решение для своей задачи;<br>использовать возможности стандартной библиотеки.<br><br>Домашние задания<br><br>1 ДЗ-08. Многопоточная асинхронная обработка команд<br><br>1. Выполнить задание описанное в 08.homework.pdf<br>2. Опционально. На <a href="http://acm.timus.ru">acm.timus.ru</a> решить 3 любые задачи с уровнем сложности от 501 до 550.<br>3. Опционально. Предоставить своё ревью на решения ДЗ-06 от двух коллег (которые его сделали).<br>4. Опционально: Предоставить ревью на РЕВЬЮ ДЗ-04. |
| 4 | <b>Lockfree DS</b>                  | разобраться в том, что такое lock-free подход;<br>видеть сценарии его использования;<br>читать lock-free код.   |
| 5 | <b>Атомарные типы данных</b>        | использовать атомарные операции в своём коде;<br>разобраться в принципах и стоимости их работы.   |



- 1 **Сетевое взаимодействие** объяснить, зачем нужна сеть в программах на C++, объяснить, как решать возникающие проблемы.
- Домашние задания
- 1 ДЗ-09. Асинхронная сетевая обработка команд
1. Выполнить задание описанное в 09.homework.pdf
  2. Опционально. На [acm.timus.ru](http://acm.timus.ru) решить 2 любые задачи с уровнем сложности от 551 до 600.
  3. Опционально. Предоставить своё ревью на решения ДЗ-07 от двух коллег (которые его сделали).
  4. Опционально: Предоставить ревью на РЕВЬЮ ДЗ-05.
- 
- 2 **Асинхронный ввод/вывод** использовать библиотеки для работы с сетью; выбирать подходящие инструменты.

- 1 **РСУБД** использовать хранилища данных в своих проектах.
- Домашние задания
- 1 ДЗ-10. outer join
1. Выполнить задание описанное в 10.homework.pdf
  2. Опционально. На [acm.timus.ru](http://acm.timus.ru) решить 2 любые задачи с уровнем сложности от 601 до 650.
  3. Опционально. Предоставить своё ревью на решения ДЗ-8 от двух коллег (которые его сделали).
- 
- 2 **Key Value** разобраться в том, что же такое NoSQL; выбирать решение под задачу.
- 
- 3 **Много данных** разобраться в идеологии map reduce; немного поэкспериментировать.
- Домашние задания
- 1 ДЗ-11. mapreduce
1. Выполнить задание описанное в 11.homework.pdf
  2. Опционально. На [acm.timus.ru](http://acm.timus.ru) решить 2 любые задачи с уровнем сложности от 651 до 700.
  3. Выложить свою реализацию на ревью.
  4. Опционально. Предоставить два своих ревью на ДЗ-09.
- 
- 4 **ORM** разобраться, зачем нужны ORM, применять их в своих приложениях.

- |   |                                |   |
|---|--------------------------------|---|
| 1 | <b>Dlib.<br/>Кластеризация</b> | разобраться, зачем нужно машинное обучение;<br>решать базовые задачи ML;<br>читать документацию Dlib. |
| 2 | <b>Dlib.<br/>Классификация</b> | разобраться с принципами классификации;<br>решать задачу классификации.                               |
| 3 | <b>Dlib. Регрессия</b>         | разобраться с формулировкой и решение задачи регрессии.   |

Домашние задания

1 ДЗ-12. Недвижимость

1. Выполнить задание описанное в 12.homework.pdf
2. Выложить свою реализацию на ревью.
3. Да и хватит на этом. Пора приступать к проекту.

<b>1</b>	<b>Выбор темы и организация проектной работы</b>  выбрать и обсудить тему проектной работы; спланировать работу над проектом; ознакомиться с регламентом работы над проектом.  Домашние задания  <b>1</b> 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.
<b>2</b>	<b>Консультация по проектам и домашним заданиям</b>  получить ответы на вопросы по проекту, ДЗ и по курсу.
<b>3</b>	<b>Защита проектных работ</b>  защитить проект и получить рекомендации экспертов.
<b>4</b>	<b>Подведение итогов курса</b>  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.