



# C++ Developer. Professional

Особенности разработки программного обеспечения на C++

Длительность курса: 146 академических часов

## 1 Система сборки: build, test and deploy

### Цели занятия:

познакомиться;  
объяснить формат работы;  
обсудить формат взаимодействия;  
объяснить зачем нужна автоматизация сборки;  
расшифровывать аббревиатуру CI/CD;  
подготовить стенд для выполнения самостоятельных работ.

### Краткое содержание:

окружение программиста: команда программистов, менеджеры, администраторы;  
личный кабинет;  
настройка стенда для выполнения ДЗ.

### Домашние задания

#### 1 ДЗ-01. Сборка и публикация пакета.

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

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 с реализованным проектом и настроенным workflow для github actions с историей успешных сборок;
  - ссылку на релиз на github для вашего проекта.
-

## 2 Особенности C++11. auto, lambda, tuple

### Цели занятия:

освежить основы по C++11 (пригодится дальше по курсу);  
рассмотреть новые фишки C++11 (тоже пригодится);  
обсудить особенности, которые важно учитывать.

### Краткое содержание:

обсуждение ДЗ;  
online-инструменты;  
ключевое слово auto;  
спецификатор decltype;  
замыкания (они же лямбды);  
кортежи.

### Домашние задания

#### 1 ДЗ-02. Фильтрация IP-адресов

Цель: Решаем проблему фильтрации IP-адресов, проверяя свой входной уровень знаний;  
Продолжаем пользоваться сборочной фермой для публикации пакета.

Что требуется сделать:

1. Выполнить задание, описанное в 02.homework.pdf
  2. Отправить на проверку в "Чат с преподавателем":
    - ссылку на свой репозиторий на github.com с реализованным проектом и настроенным workflow для github actions с историей успешных сборок;
    - ссылку на релиз на github для вашего проекта.
-

3 **Особенности  
C++11. Move,  
variadic  
templates**

**Цели занятия:**

освежить основы по C++11 (пригодится дальше по курсу);  
рассмотреть новые детали;  
обсудить особенности, которые стоит учитывать.

**Краткое содержание:**

rvalue и lvalue ссылки;  
move semantic;  
variadic Templates;  
POD.

---

4 **Сравнение  
C++11/C++14 с  
C++ 03**

**Цели занятия:**

освежить знания по C++14;  
рассмотреть новые детали;  
обсудить особенности C++ 14, на которые стоит  
обратить внимание.

**Краткое содержание:**

семантика перемещения;  
двоичные константы;  
пользовательские литералы или странный оператор  
"operator";  
static\_assert;  
using;  
nullptr и nullptr\_t;  
enum class;  
constexpr;  
override, final;  
default, delete;  
ranged for. Delegate ctor;  
data member initializers;  
braced initialization. Initializer\_list.

---

## 5 Реализация умных указателей

### Цели занятия:

рассмотреть правило: "взял память - верни, когда больше не нужна";  
объяснить, как следовать этому правилу;  
разобраться с разными типами умных указателей.

### Краткое содержание:

область видимости;  
время жизни;  
проблемы управления ресурсами;  
глупый умный указатель;  
unique\_ptr. Shared\_ptr;  
weak\_ptr;  
enable\_shared\_from\_this.

---

### **Цели занятия:**

объяснить важность стратегий аллокации памяти;  
использовать разные аллокаторы;  
разобраться, как реализовать свой аллокатор;  
объяснить, как расширяется `std::vector`, и почему именно так.

### **Краткое содержание:**

стандартный аллокатор и его недостатки;  
различные методы управления памятью, их достоинства и недостатки;  
как можно повлиять на ситуацию;  
`Std::allocator`;  
`Std::allocator_traits`.

### **Домашние задания**

#### **1** ДЗ-03. 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 03.homework.pdf

2. На [acm.timus.ru](https://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](https://github.com) с реализованным проектом и настроенным workflow для github actions с историей успешных сборок;
  - ссылку на релиз на github для вашего проекта.
  - ссылку на аккаунт на [acm.timus.ru](https://acm.timus.ru) со списком принятых задач.
-

### **Цели занятия:**

перестать бояться шаблонов;  
пообещать себе перестать пользоваться макросами;  
прочитать STL;  
работать с документацией.

### **Краткое содержание:**

эволюция шаблонов;  
современное положение дел;  
метафункции;  
рекурсия и условия;  
документация;  
SFINAE.

### **Домашние задания**

#### **1** ДЗ-04. SFINAE + Doxygen

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

Что требуется сделать:

1. Выполнить задание описанное в 04.homework.pdf

2. Решить первые 15 задач на 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. Отправить на проверку в "Чат с преподавателем":

- ссылку на свой репозиторий на github.com с реализованным проектом и настроенным workflow для github actions с историей успешных сборок;
- ссылку на аккаунт на acm.timus.ru со списком принятых задач.

---

8 **Обзор C++17. Constexpr lambda. Fold expression. Attributes. Type deduction**

**Цели занятия:**

разобраться с основными нововведениями C++ 17; объяснить в каких случаях они используются; обойти несколько спорных моментов.

**Краткое содержание:**

constexpr lambda;  
fold expression. Attributes;  
type deduction;  
auto template parameter;  
trigraph (only for fun).

---

9 **Обзор C++17. If constexpr. Structured bindings. Statements with initializer. std::filesystem**

**Цели занятия:**

разобраться с основными нововведениями C++17; рассмотреть кейсы использования; обсудить несколько тонких моментов; пользоваться новым стандартом языка.

**Краткое содержание:**

if constexpr VS define;  
inline variables/ Structured bindings/ If/switch/ Statements with initializer;  
std::filesystem. Undefined behaviour.

---

10 **C++20 - что это и зачем?**

**Цели занятия:**

открыть для себя дивный мир последнего стандарта, о котором ходят столько легенд и слухов; объяснить, почему именно такие изменения были добавлены в стандарт; получить список нововведений для повседневного использования.

**Краткое содержание:**

стандарт языка C++.



## 1 Принципы проектирования ПО

### Цели занятия:

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

### Краткое содержание:

UML;  
принципы: SoC, DRY, YAGNI, KISS, TDA, LoD, SOLID.

### Домашние задания

#### 1 ДЗ-05. Editor.

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 05.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?](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. Выложить на ревью свои решения ДЗ-02, ДЗ-03 и ДЗ-04.

4. Сделать ревью решений ДЗ-02 двух коллег (которые его выполнили).

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

- ссылку на свой репозиторий на [github.com](https://github.com) с реализованным проектом и настроенным workflow

для github actions с историей успешных сборок;  
- ссылку на аккаунт на acm.timus.ru со списком  
принятых задач  
- ссылки на выполненные ревью

---

## 2 GRASP

### Цели занятия:

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

### Краткое содержание:

indirection;  
low Coupling;  
polymorphism;  
protected Variations;  
creator;  
controller;  
pure Fabrication;  
information Expert;  
high Cohesion.

---

## 3 Шаблоны GoF. Порождающие

### Цели занятия:

прочитать паттерны проектирования;  
использовать паттерны;  
объяснить ценность экспериментов.

### Краткое содержание:

одиночка (Singleton);  
пул одиночек (Multiton);  
пул объектов (Object pool);  
фабричный метод (Factory method);  
абстрактная фабрика (Abstract factory);  
строитель (Builder);  
прототип (Prototype);  
внедрение зависимости (Dependency Injection);  
ленивая инициализация (Lazy initialization).

---

## Шаблоны GoF.

### Цели занятия:

прочитать паттерны проектирования;  
использовать паттерны;  
объяснить ценность экспериментов.

### Краткое содержание:

адаптер (Adapter);  
мост (Bridge);  
компоновщик (Composite);  
декоратор (Decorator);  
фасад (Facade);  
приспособленец (Flyweight);  
заместитель (Proxy).

### Домашние задания

#### 1 ДЗ-06. Matrix.

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

Что требуется сделать:

1. Выполнить задание, описанное в 06.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?](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-03 двух коллег (которые его выполнили).

4. Опционально: сделать ревью на РЕВЬЮ ДЗ-03 (отзыв о выполненном ревью).

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

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

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

**Цели занятия:**

прочитать паттерны проектирования;  
использовать паттерны;  
объяснить ценность экспериментов.

**Краткое содержание:**

команда (Command);  
интерпретатор (Interpreter);  
итератор (Iterator);  
посредник (Mediator);  
хранитель (Memento);  
цепочка ответственности (Chain of responsibility).

---

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

**Цели занятия:**

прочитать паттерны проектирования;  
использовать паттерны;  
объяснить ценность экспериментов.

**Краткое содержание:**

наблюдатель (Observer);  
состояние (State);  
стратегия (Strategy);  
шаблонный метод (Template method);  
посетитель (Visitor).

---

7 **STL.  
Контейнеры**

**Цели занятия:**

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

**Краткое содержание:**

последовательные контейнеры;  
ассоциативные;

## Домашние задания

### 1 ДЗ-07. 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 07.homework.pdf

2. На [acm.timus.ru](http://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-05 двух коллег (которые его выполнили).

4. Опционально: сделать ревью на РЕВЬЮ ДЗ-05 (отзыв о выполненном ревью).

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

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

---

## 8 STL. Алгоритмы

### Цели занятия:

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

### Краткое содержание:

поиск;  
сортировка;  
модифицирующие алгоритмы;  
немодифицирующие;  
численные алгоритмы;  
перестановки.

---

## 9 Бонус. Красно-чёрные деревья

### Цели занятия:

перечислить правила размещения элементов в красно-чёрном дереве;  
выполнить балансировку красно-чёрного дерева;  
выполнить удаление элементов из красно-чёрного дерева.

### Краткое содержание:

принципы и правила красно-чёрных деревьев;  
случаи нарушения балансировки и способы их разрешения;  
примеры добавления и удаления элементов.

---

## 10 Boost. Разное

### Цели занятия:

объяснить философию проекта Boost;  
прочитать документацию Boost,  
выбирать и применять библиотеки.

### Краткое содержание:

философия boost;  
CMake+boost;  
boost::bimap;  
boost::multi\_index;  
boost::circular\_buffer;  
boost::fusion;  
boost::HANA;  
boost::MPL;  
boost::program\_options.

---

## 11 Boost. Паттерны проектирования

### Цели занятия:

рассмотреть Boost;  
посмотреть на реализацию паттернов проектирования;  
расширить знания про библиотеку.

### Краткое содержание:

boost::fusion;  
паттерн Фабрика;  
паттерн Слот-сигнал;  
паттерн Легковес.

### Домашние задания

#### 1 ДЗ-08. Boost

Цель: В этой самостоятельной работе применяем на практике навыки использования библиотеки Boost.

Что требуется сделать:

1. Выполнить задание, описанное в 08.homework.pdf

2. Опционально: на [acm.timus.ru](http://acm.timus.ru) решить 3 любые задачи с уровнем сложности от 401 до 450.  
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. Опционально. Сделать ревью решений ДЗ-06 двух коллег (которые его выполнили).
4. Опционально: сделать ревью на ревью ДЗ-06.
5. Отправить на проверку в "Чат с преподавателем":
- ссылку на свой репозиторий на [github.com](https://github.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.
- 

## 12 Бонус. Хэш-функции и хэш-таблицы

### Цели занятия:

создавать хэш-функции;  
вычислять хэш-значения для строк;  
добавлять данные в хэш-таблицы.

### Краткое содержание:

понятие хэш-функции и хэш-таблицы;  
способы составления хэш-функций для разных объектов;  
правила размещения данных в хэш-таблицах методом цепочек.



## 1 Многопоточность

### Цели занятия:

использовать потоки в своих приложениях;  
решать основные проблемы многопоточности;  
использовать инструменты стандартной библиотеки.

### Краткое содержание:

процессы и потоки;  
цель и проблемы многопоточности;  
std::thread, std::future / std::promise, std::async,  
std::condition\_variable.

---

## 2 Шаблоны 5

### Цели занятия:

объяснить многопоточные паттерны;  
использовать для решения задач.

### Краткое содержание:

Concurrency pattern;  
Active Object, Balking, Barrier, Double-checked locking,  
Guarded suspension, Leaders/followers, Monitor  
Object, Reactor, Read write lock, Scheduler, Thread  
pool, Thread-local storage.

---

### 3 Асинхронное программирование

#### Цели занятия:

объяснить разницу между многопоточностью, параллельностью и асинхронностью;  
выбрать подходящее решение для своей задачи;  
использовать возможности стандартной библиотеки.

#### Краткое содержание:

многопоточность;  
параллельность;  
асинхронности;  
стандартная библиотека;  
разбор домашней работы.

#### Домашние задания

##### 1 ДЗ-09. Многопоточная асинхронная обработка команд

1. Выполнить задание описанное в 09.homework.pdf
2. Опционально. На [acm.timus.ru](http://acm.timus.ru) решить 3 любые задачи с уровнем сложности от 501 до 550.
3. Опционально. Предоставить своё ревью на решения ДЗ-07 от двух коллег (которые его сделали).

---

### 4 Lockfree DS

#### Цели занятия:

разобраться в том, что такое lock-free подход;  
видеть сценарии его использования;  
прочитать lock-free код.

#### Краткое содержание:

понятие атомарности;  
CaS;  
неблокирующие структуры данных и алгоритмы;  
boost Lockfree&amp;

---

## 5 Атомарные типы данных

### Цели занятия:

использовать атомарные операции в своём коде;  
разобраться в принципах и стоимости их работы.

### Краткое содержание:

атомарность в стандартной библиотеке (`std::atomic`);  
принцип работы атомарных переменных;  
упорядочение памяти (`memory ordering`).

## 1 Выбор темы и организация проектной работы

### 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.

---

## 2 Сетевое взаимодействие

### Цели занятия:

объяснить, зачем нужна сеть в программах на C++,  
объяснить, как решать возникающие проблемы.

### Краткое содержание:

OSI/ISO;  
сокеты;  
протоколы: TCP и UDP;  
сериализация;  
HTTP;  
Json;  
Protobuf.

### Домашние задания

#### 1 ДЗ-10. Асинхронная сетевая обработка команд

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 551 до 600.
  3. Опционально. Предоставить своё ревью на решения ДЗ-07 от двух коллег (которые его сделали).
-

### **3 Асинхронный ввод/вывод**

#### **Цели занятия:**

использовать библиотеки для работы с сетью;  
выбирать подходящие инструменты.

#### **Краткое содержание:**

библиотеки и практики работы с неблокирующей  
передачей данных;  
Boost.Asio, POCO.

## 1 РСУБД

### Цели занятия:

использовать хранилища данных в своих проектах.

### Краткое содержание:

понятие РСУБД;  
реляционная алгебра;  
язык запросов SQL;  
детали реализации;  
пример кода SQLite.

### 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

### Цели занятия:

разобраться в том, что же такое NoSQL;  
выбирать решение под задачу.

### Краткое содержание:

идея NoSQL;  
Key-value;  
документоориентированные базы данных;  
графовые базы данных;  
колоночные хранилища.

---

### 3 Много данных

#### Цели занятия:

разобраться в идеологии map reduce;  
немного поэкспериментировать.

#### Краткое содержание:

стратегия "разделяй и властвуй";  
идея Map reduce;  
пример реализации.

#### Домашние задания

##### 1 ДЗ-11. mapreduce

1. Выполнить задание описанное в 11.homework.pdf
  2. Опционально. На 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

#### Цели занятия:

разобраться, зачем нужны ORM;  
применять их в своих приложениях.

#### Краткое содержание:

ООП при работе с базами данных;  
ORM, паттерны DAO и Repository.

1 **Dlib.**  
**Кластеризация**  
**и**  
**классификация**

**Цели занятия:**

разобраться, зачем нужно машинное обучение;  
решать базовые задачи ML;  
прочитать документацию Dlib.

**Краткое содержание:**

машинное обучение, и зачем оно нужно;  
задачи машинного обучения;  
кластеризация.

---

2 **Dlib. Регрессия**

**Цели занятия:**

разобраться с формулировкой и решение задачи  
регрессии.

**Краткое содержание:**

прогнозирование;  
оцифровка данных;  
тематическое моделирование;  
регрессия.

**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 приступить к проекту.



**1**    **Консультация  
по проектам и  
домашним  
заданиям**

**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.

**Домашние задания**

**1**    Проектная работа

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**    **Защита  
проектных  
работ**

**Цели занятия:**

защитить проект и получить рекомендации экспертов.

**Краткое содержание:**

презентация проектов перед комиссией;  
вопросы и комментарии по проектам.

---

### **3 Подведение итогов курса**

#### **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.