Утверждено
Приказом Генерального директора
№ 16-О от 10.08.2026
Генеральный директор
ООО «НТЦ ПРОТЕЙ»

_____ Апостолова Н.А.

ПРОТЕЙ

Программа курса
«Разработчик высоконагруженных отказоустойчивых программ на C++»

Санкт-Петербург
2026 г.

Содержание

Длительность курса

Программа обучения рассчитана на 46 часов. Количество часов, отводимых на изучение теоретической программы – 24 часа, отработка практических навыков – 22 часа, из них 2 часа выделено на итоговую аттестацию.

По результатам итоговой аттестации, показавшие лучший результат обучающиеся допускаются к выполнению квалификационной работы.

Цель

Подготовка начинающих специалистов и студентов технических направлений к профессиональной деятельности в командах разработки программного обеспечения Группы Компаний «ПРОТЕЙ».

Категория обучающихся

Студенты IT- и телеком-направлений, начинающие специалисты в области:

- мобильных сетей;
- прикладной информатики;
- программная инженерия;
- сетевых технологий;
- телекоммуникаций.

Уровень обучающихся

- знают предпосылки разработки, принципы и структуру базовой эталонной модели взаимодействия открытых систем OSI;
- имеют понятие о протоколах семейства TCP/IP;
- знают правила работы с различными информационными системами и базами данных;
- умеют работать с различными информационными системами и базами данных, обрабатывать информацию с использованием современных технических средств;
- умеют декомпозировать задание на отдельные задачи, определять перечень требуемых технологий для реализации;
- умеют разрабатывать алгоритмы, выбирать оптимальные структуры данных для реализации задачи;
- умеют использовать современные инструменты индивидуальной и групповой разработки;
- имеют представления о современных тенденциях в разработке высоконагруженного отказоустойчивого программного обеспечения
- знают основы процедурного и объектно-ориентированного программирования;
- владеют синтаксисом и приоритетом операций в языке программирования C++ на уровне стандарта не ниже C++11, компонентами библиотеки STL;
- имеют представление об аппаратных комплектующих современной вычислительной техники, о цикле выполнения программы в рамках пользовательских операционных систем общего назначения;
- понимают основы реализации наследования и полиморфизма в C++;
- знают различие между перегрузкой и переопределением функций;

Планируемые результаты обучения

После прохождения курса обучающиеся

будут знать:

- основы архитектуры мобильных сетей;
- основы архитектуры мобильного ядра пакетной сети;
- назначение основных элементов ядра сети;
- принципы организации проектов программного обеспечения на языке C++;
- стадии и процессы построения исполнительного файла из программного кода;
- уровни представления памяти в современных компьютерных системах;
- особенности работы с памятью в программах, написанных на языке программирования C++;
- представление данных в памяти в зависимости от платформы компьютерной системы;
- стек протоколов семейства TCP/IP, сферы их применения;
- принципы построения клиент-серверных приложений и организации информационного обмена между ними;
- различия между видами приложений реального времени;

будут уметь:

- организовывать настраиваемые проекты при помощи системы сборки CMake;
- обеспечивать модульное тестирование, применяя Google Tests;
- выполнять сравнение производительности алгоритмов при помощи Google Benchmark;
- осуществлять отладку приложений с применением статических анализаторов кода, санитайзеров и(или) при помощи отладчика gdb;
- вести разработку в парадигме ветвления GitHub-Flow, применять подход Pull-Request для предоставления кодовой базы на оценку;
- обоснованно выбирать тип STL-контейнера, формировать структуры данных для решения задачи;
- применять подходы шаблонного программирования для обобщения программного кода, ограничивать вариации инстанцирования по допустимым типам данных;
- применять паттерны проектирования для структурирования архитектурных решений;
- применять примитивы синхронизации и сущности многопоточной обработки данных;
- создавать клиент-серверные приложения на основе сокетов, организовывать обмен на основе TCP для передачи текстовых (в формате JSON) и бинарных данных (с учётом порядка байт);
- определять наиболее подходящие архитектурные решения для организации приложений;
- документировать проекты, комментировать код и формировать логи выполнения программы.

Материально-техническое обеспечение

Проектор с воспроизведением на интерактивную панель, маркерная доска, учебная мебель (стулья, столы для слушателей в достаточном количестве), стол и стул для преподавателя, стеллаж для хранения документов, необходимое количество ноутбуков для работы обучающихся, а также ноутбук для преподавателя.

Учебный план

№ п/п	Наименование разделов и тем	Всего, часов	Теория	Практика	Форма проведения занятий
1.	Тема 1. Эволюция и архитектура сетей связи	2	2	0	Лекция
2.	Тема 2. Основы построения проектов программ	4	2	2	Лекция, практическое задание
3.	Тема 3. Представление данных и работа с памятью	4	2	2	Лекция, практическое задание
4.	Тема 4. ООП и построение объектов	4	2	2	Лекция, практическое задание
5.	Тема 5. Стратегии обработки ошибок и модульное тестирование	4	2	2	Лекция, практическое задание
6.	Тема 6. Обобщённое программирование	4	2	2	Лекция, практическое задание
7.	Тема 7. Структуры данных, алгоритмы и производительность кода	4	2	2	Лекция, практическое задание
8.	Тема 8. Внешние источники данных и сетевое программирование	4	2	2	Лекция, практическое задание
9.	Тема 9. Многопоточное программирование	4	2	2	Лекция, практическое задание
10.	Тема 10. Оптимизация и шаблоны проектирования	4	2	2	Лекция, практическое задание
11.	Тема 11. Цикл разработки программного продукта	4	2	2	Лекция, практическое задание
12.	Тема 12. Архитектуры программного обеспечения	4	2	2	Лекция, итоговая аттестация
Итого:		46	24	22	

Рабочая программа

Тема 1. Эволюция и архитектура сетей связи

- Определение стационарной и мобильной сетей связи, их фундаментальные отличия;
- Эволюция мобильных сетей связи;
- Компоненты (узлы) мобильной сети связи;
- Процедуры мобильной сети связи (attach, roaming, TAU, handover);
- Параметры качества (QoS) и туннели передачи данных (bearer);
- Продукты ГК ПРОТЕЙ в контексте структуры сетей связи;
- Проецирование требований к качеству ПО для телеком-продуктов на цикл разработки программных продуктов.

Тема 2. Основы построения проектов программ

- Структура проекта программы;
- Этапы компиляции и компоновки программы;
- Стратегии организации иерархии файлов исходного кода;
- Система сборки CMake, подходы к подключению внешних зависимостей, сборки под различные задачи разработки и целевые платформы;
- Стратегии ветвления Git Flow, GitHub Flow, GitLab Flow;
- Документарное оформление проекта: комментирование, readme;
- Обзор базового минимум языка C++. Частые ошибки в применении директив препроцессора.

Тема 3. Представление данных и работа с памятью

- Память процессора, оперативная память, пространство памяти процесса в операционной системе;
- Организация памяти программы, особенности работы с каждым из блоков;
- Фундаментальные типы данных, представление в памяти и обработка;
- «Сырые» указатели и ссылки, выделение и освобождение памяти;
- Константные переменные; объекты enum и enum class;
- Размещение тривиальных данных в памяти, выравнивание и упаковка байт в объектах struct;
- Понятие маппирования данных, объекты union;
- Преобразование типов; идентичность типов, RTTI;
- Понятие управления ресурсами, концепция RAII; «умные» указатели.

Тема 4. ООП и построение объектов

- Философия ООП: зачем создавать классы; инварианты объектов;
- Построение объектов в памяти; виртуальные таблицы, данные родительских классов;
- Ограничения неявных приведений типов к объектам классов;
- Наследование интерфейса, наследование поведения, сокрытие изменений от переборки объектов, паттерн PIMPL;
- Особенности выбрасывания исключений в конструкторе и деструкторе объекта;
- Особенности применения виртуальных функций;
- Конструкторы копирования и перемещения;
- Переопределение операторов.

Тема 5. Стратегии обработки ошибок и модульное тестирование

- Виды ошибок: синтаксические, семантические, ресурсные;
- Обработка кодов ошибок, исключений; сравнение подходов к информированию об ошибках через коды ошибок и путём выброса исключений;
- Логирование программы, отладка программы при помощи журналов логов;
- Отладка программы посредтвам отладчика на примере gdb;
- Статический анализ кода. Знакомство и применение инструмента cppcheck;
- Динамический анализ кода. Сборка с применением санитайзеров, определение критических мест кода по выводу санитайзеров и исправление ошибок;
- Модульное тестирование программы. Применение библиотеки Google Test.

Тема 6. Обобщённое программирование

- Понятие обобщённого программирования, выделение шаблонов функций, классов;
- Специализация, частичная специализация;
- Инстанцирование шаблонов, SFINAE;
- Ограничения применения шаблонов: enable_if, concept;
- Изменение функциональности применением if constexpr;
- Лямбда-выражения: аспекты передачи данных внутрь анонимного объекта;
- Особенности применения лямбда-выражений.

Тема 7. Структуры данных, алгоритмы и производительность кода

- Стандартные структуры данных и алгоритмы;
- Алгоритмическая сложность;
- Библиотека STL, реализации контейнеров и алгоритмов;
- Требования к интерфейсу контейнируемых классов;
- Инструменты оценки быстродействия. Сравнение алгоритмов при помощи Google Benchmark;
- Сторонние открытые библиотеки;
- Влияние способа лицензирования на применимость сторонних библиотек в проекте.

Тема 8. Внешние источники данных и сетевое программирование

- Структурированные и неструктурированные форматы файлов;
- Виды баз данных и аспекты их применения;
- Обмен данными по сети. Модель OSI и TCP/IP. MAC-адрес, IP-адрес, порт;
- Сетевые протоколы стека TCP/IP; сравнение TCP, UDP и SCTP;
- Порядок размещения байт на различных платформах и при передаче по сети;
- Сетевое программирование через сокеты: цикл клиента и цикл сервера;
- Методики построения прикладных протоколов информационного обмена на основе JSON, Proto-Buffer, бинарном формате;
- Машина состояний на примерах: анализа файла формата Bitmap, процесса авторизации и аутентификации;
- Перехват и анализ сетевого трафика при помощи WireShark;
- Ускорение работы сетевого обмена: библиотека DPDK.

Тема 9. Многопоточное программирование

- Понятие и характеристики мягкого и жёсткого реального времени;
- Многопроцессное, многопоточное и параллельное выполнение программы;
- Разделяемые ресурсы, примитивы синхронизации: семафор, мьютекс, фьючерсы, атомарные операции;
- Неблокирующая многопоточность;
- Барьеры памяти, как механизм синхронизации обработки данных;
- Подходы многопоточного программирования через async и coroutins;
- Модель акторов.

Тема 10. Оптимизация и паттерны проектирования

- Виды и стратегии оптимизации программы;
- Режимы оптимизации компилятором;
- Оптимизация через особенности реализации контейнеров STL, ручное раскручивание циклов;
- Паттерны проектирования;
- Структурные паттерны: Одиночка, Фасад;
- Порождающие паттерны: Фабрика, Строитель;
- Поведенческие паттерны: Шаблонный метод, Стратегия, Декоратор, Нулевой объект;
- Ключевое слово volatile;
- Инструменты оценки производительности программ: построение flamechart.

Тема 11. Цикл разработки программного продукта

- Сравнение подходов к разработке программного продукта: водопад, итеративный, Agile;
- Соотнесение Agile и CI/CD;
- Стадии разработки в Agile: Релиз, Итерация (спринт), Задача;
- Оценивание объёма работы;
- Scrum-подход, структура Scrum-команды;
- Процессы Scrum-команды;
- Инструменты команды: трекеры задач, канбан-доски – на примере YouTrack; учёт рабочего времени;
- Организация работы над задачей: формирование pull-request и проведение code-review;
- Виды тестирования программного продукта. Процессы тестирования со стороны разработчика;
- Подходы к разработке программ: TDD, XP;

Тема 12. Архитектуры программного обеспечения

- Моделирование архитектур при помощи нотаций C4 и UML.
- Виды архитектур программных систем: монолит, клинет-серверная, микросервисная;
- Виды резервирования;
- Виды масштабирования: горизонтальное и вертикальное.

Тематика практических заданий

Задание к лекции 1

Практическое задание к лекции – не предусмотрено.

Домашнее задание к лекции 1 – создать закрытый репозиторий на GitHub.com и выслать приглашение преподавателям курса стать участниками разработки.

Задание к лекции 2

Обучающиеся создают ветки в репозитории в соответствии с парадигмой ветвления GitHub Flow, формируют файл проекта и ресурсный файл с функцией main, после чего проходят процедуру Pull-Request.

Домашнее задание к лекции 2 состоит в написании каркаса приложения, которое принимает на вход конфигурационные данные. Предоставляет консольный интерфейс с пользователем и позволяет обрабатывать команды.

Задание к лекции 3

Обучающиеся получают примеры кода с сформированной структурой данных; они анализируют структуру, её назначение, и реорганизуют таким образом, чтобы структура данных была оптимальной для применения в программе.

Задание к лекции 4

Обучающиеся берут одну из сущностей: термометр, рулетка, человек (или придумывают что-то своё) и описывают граничные значения инварианта для класса.

Обучающиеся описывают классы и их иерархию для сущности из некоторой предметной области. Реализовать полный набор функций по “Правилу пяти”.

Домашнее задание к лекции 4 – определить, какие сущности присутствуют в коде из предыдущего домашнего задания и выделить их в отдельные классы.

Задание к лекции 5

На основе кода из предыдущей лекции обучающиеся пишут модульные тесты, обеспечивающие проверку соответствия инвариантам и(или) использования потенциально небезопасных исходных данных.

Домашнее задание к лекции 5 – добавить проверки в код, запись журнала логирования и написать модульные тесты.

Задание к лекции 6

Для выданного преподавателем алгоритма написать шаблонную функцию, применимую только типам данных, которыми оперирует заданный алгоритм.

Задание к лекции 7

Каждый обучающийся получает задачу, реализует её при помощи контейнеров и алгоритмов STL двумя способами с целью сравнительного анализа наилучшей производительности при помощи инструмента Google Benchmark. По рекомендации преподавателя обучающиеся дополняют код предложенными преподавателем вариантами и также сравнивают производительность.

Задание к лекции 8

Обучающиеся получают pcap-файл, который анализируют при помощи WireShark с целью определения применяемого транспортного протокола и содержимого пакета.

Домашнее задание к лекции 8 – организовать информационный обмен между двумя существующими приложениями посредствам сокетов при помощи бинарного протокола.

Задание к лекции 9

Каждому обучающемуся предоставляется набор исходных данных и задание по их обработке. Задача обучающегося – реализовать многопоточный алгоритм обработки данных.

Домашнее задание к лекции 9 – организовать работу приложения-клиента таким образом, чтобы потоки получения пользовательского ввода и сетевого обмена работали параллельно.

Задание к лекции 10

Каждому обучающемуся выдаётся алгоритм, который необходимо проанализировать, построить “flamechart”. По результатам анализа данную программу необходимо оптимизировать, используя знания, полученные на лекции, и показать прирост производительности при помощи инструмента Google Benchmark.

Задание к лекции 11

Слушатели разбиваются на подгруппы, всем выдаётся набор карточек с описаниями пользовательских историй. Подгруппам необходимо оценить объём пользовательских историй в “пунктах”. Происходит обсуждение результатов каждой подгруппы.

После этого подгруппы занимаются анализом пользовательских историй: выделяют задачи, роли, инструменты, оценивают свои возможности по реализации и какие навыки и(или) дополнительный персонал им необходим для реализации.

Квалификационная работа

Задание представляет собой дальнейшее развитие программы, созданной в рамках выполнения домашних заданий, и нацелено на создание упрощённой имитации группы функциональных узлов мобильной сети для оказания услуг мобильной связи.

Список рекомандованной литературы

1. Сети связи. Б.С. Гольдштейн, Н.А. Соколов, Г.Г. Яновский. СПб.: БХВ, 2011.
2. Программирование. Принципы и практика с использованием C++. 2-е изд. Бьёрн Страуструп. Москва: Вильямс, 2018.
3. C++ для профи. Джош Лоспинозо. СПб.: Питер, 2021.
4. Чистый код. Роберт Мартин. СПб.: Питер, 2023.
5. Эффективное использование C++. Скотт Мэйерс. Москва: ДМК-Пресс, 2017.
6. Наиболее эффективное использование C++. Скотт Мэйерс. Москва: Диалектика, 2017.
7. Скользкие места C++. Стефан К. Дьюхэрст. Москва: ДМК-Пресс, 2017.
8. Приёмы объектно-ориентированного проектирования. Паттерны проектирования. Эрик Гамма, Рихард Хелм, Ральф Джонсон, Джон Влиссидес. Москва: ДМК-Пресс, 2018.
9. Командная строка Linux. Дениел Джей Барретт. СПб.: Питер, 2023.
10. Разработка через тестирование. Кент Бэк. СПб.: Питер, 2022.
11. Тестирование программного обеспечения. Контекстно-ориентированный подход. Кем Кейнер, Джеймс Бах, Брет Петтикорд. СПб.: Питер, 2025.
12. BDD в действии. Джон Фергюсон Смарт, Ян Молак. Москва: ДМК-Пресс, 2024.
13. Введение в UML. Гради Буч, Джеймс Рамбо, Ивар Якобсон. Москва: ДМК-Пресс, 2012.
14. Рефакторинг. Улучшение проекта существующего кода. Робертс Дон, Мартин Фаулер, Брант Джон, Кент Бэк. Киев: Диалектика, 2019.
15. Игровой движок, 3-е изд. Джейсон Грегори. СПб.: Питер, 2025.
16. Теоретический минимум по Computer Science. Владстон Фило. СПб.: Питер, 2026.

17. Теоретический минимум по Computer Science. Сети, криптография, data science. Владстон Фило, Мото Пиктет. СПб.: Питер, 2022.
18. Алгоритмы для начинающих. Панос Луридас. Москва: Торговый дом Эксмо, 2022.
19. Построение компиляторов. Никлаус Вирт. Москва: ДМК-Пресс, 2013.
20. Дата-ориентированное программирование. Йонатан Шарвит. СПб.: БХВ, 2024.
21. Beej`s guide to network programming. Brian Hall. (электронная книга): https://beej.us/guide/bgnet/pdf/bgnet_a4_c_1.pdf

Список используемых источников

1. https://cpp-kt.github.io/cpp-notes/course.html
2. https://cppreference.com/
3. https://isocpp.github.io/CppCoreGuidelines/
4. https://habr.com/ru/articles/54762/
5. https://habr.com/ru/articles/101430/
6. https://habr.com/ru/articles/210288/
7. https://habr.com/ru/articles/212101/
8. https://habr.com/ru/articles/228181/
9. https://habr.com/ru/articles/339406/
10. https://habr.com/ru/articles/339492/
11. https://habr.com/ru/articles/348198/
12. https://habr.com/ru/articles/352570/
13. https://habr.com/ru/articles/441742/
14. https://habr.com/ru/articles/478124/
15. https://habr.com/ru/articles/480422/
16. https://habr.com/ru/articles/482040/
17. https://habr.com/ru/articles/509004/
18. https://habr.com/ru/articles/540984/
19. https://habr.com/ru/articles/579490/
20. https://habr.com/ru/articles/599801/
21. https://habr.com/ru/articles/664044/
22. https://habr.com/ru/articles/673428/
23. https://habr.com/ru/articles/683204/
24. https://habr.com/ru/articles/801531/
25. https://habr.com/ru/articles/828802/
26. https://habr.com/ru/articles/831754/
27. https://habr.com/ru/articles/896954/
28. https://habr.com/ru/articles/927574/
29. https://habr.com/ru/articles/957024/
30. https://habr.com/ru/articles/1005972/
31. https://habr.com/ru/articles/1007746/
32. https://habr.com/ru/companies/amvera/articles/1010466/
33. https://habr.com/ru/companies/avito/articles/710660/
34. https://habr.com/ru/companies/itq_group/articles/705598/
35. https://habr.com/ru/companies/kaiten/articles/893866/
36. https://habr.com/ru/companies/kaiten/articles/911566/
37. https://habr.com/ru/companies/otus/articles/444524/
38. https://habr.com/ru/companies/otus/articles/455978/
39. https://habr.com/ru/companies/otus/articles/515078/
40. https://habr.com/ru/companies/otus/articles/762548/
41. https://habr.com/ru/companies/otus/articles/793278/

42. https://habr.com/ru/companies/otus/articles/794821/
43. https://habr.com/ru/companies/otus/articles/801045/
44. https://habr.com/ru/companies/ruvds/articles/777764/
45. https://habr.com/ru/companies/sberbank/articles/672022/
46. https://habr.com/ru/companies/voximplant/articles/354502/
47. https://habr.com/ru/companies/wunderfund/articles/325634/
48. https://pvs-studio.ru/ru/blog/lessons/0019/
49. https://pvs-studio.ru/ru/blog/posts/cpp/1053/
50. https://pvs-studio.ru/ru/blog/posts/cpp/1339/
51. https://pvs-studio.ru/ru/docs/warnings/v824/
52. https://pvs-studio.ru/ru/docs/warnings/v1016/
53. https://agilemanifesto.org
54. https://c4model.com/
55. https://cmake.org/cmake/help/latest/index.html
56. https://compress.ru/article.aspx?id=11931&part=index41ext1
57. https://gcc.gnu.org/onlinedocs/gcc/Optimize-Options.html
58. https://git-scm.com/about
59. https://github.com/github/gitignore
60. https://google.github.io/benchmark/user_guide.html
61. https://learn.microsoft.com/ru-ru/cpp/cpp/
62. https://learngitbranching.js.org/?locale=ru_RU
63. https://scrumguides.org
64. https://systems.education/ software_styles_and_patterns_with_cheatsheet
65. https://tproger.ru/articles/luchwie-praktiki-v-kod-revyu--glavnye-owibki-i-kak-ih-iz-bezhat
66. https://webdevkin.ru/courses/git/start
67. https://7universum.com/ru/tech/archive/item/20480

Технические стандарты

1. C++23
https://www.open-std.org/jtc1/sc22/wg21/docs/papers/2023/n4950.pdf

2. IP
https://datatracker.ietf.org/doc/html/rfc791

3. TCP
https://datatracker.ietf.org/doc/html/rfc793

4. SCTP
https://datatracker.ietf.org/doc/html/rfc9260

5. UDP
https://datatracker.ietf.org/doc/html/rfc768

6. JSON
https://datatracker.ietf.org/doc/html/rfc8259