Практическая программа

C++ Developer. Professional

Подготовка к профессиональной карьере в IT
Разработчик на C++ с нуля до профессионала

Для выпускников вузов, 18 месяцев

Модуль 5. Современный C++

Тема 1
Основы. Формат, переменные, типы, ввод-вывод, циклы, условия

Теория
- Компилятор, компоновщик, стандартная библиотека
- Формат программы, переменные, арифметика, ввод-вывод
- Условия и циклы, арифметика, обработка ошибок
- Отладка и профилирование: gdb, valgrind, strace, ltrace, perf

Домашнее задание
- Задание на арифметику и ввод-вывод
- Задание на условия и циклы: калькулятор, проверка чисел, таблица умножения
- Задание на отладку и профилирование: найти утечку и ошибку в предоставленном коде

Тема 2
Функции и стек, рекурсия, массивы, строки

Теория
- Функции, стек, рекурсия
- Массивы, строки, C-style vs std::string
- Ввод-вывод, потоки, буферизация

Домашнее задание
- Задание на функции и рекурсию: факториал, числа Фибоначчи, ханойские башни
- Задание на массивы и строки: сортировка, поиск, обработка строк
- Задание на ввод-вывод: парсинг CSV, обработка потока данных

Тема 3
Указатели и ссылки, динамическая память

Теория
- Указатели и ссылки, арифметика указателей
- Динамическая память: new/delete, утечки, valgrind
- Передача по ссылке, const-корректность

Домашнее задание
- Задание на указатели: обмен значениями, обход массива, поиск в массиве
- Задание на динамическую память: динамический массив, матрица, строка
- Задание на valgrind: найти и исправить утечки в предоставленном коде

Тема 4
Классы и объекты, конструкторы, деструкторы, RAII

Теория
- Классы и объекты, конструкторы, деструкторы
- Инкапсуляция, геттеры и сеттеры, friend
- Статические члены, RAII, Rule of Three

Домашнее задание
- Задание на классы: прямоугольник, точка, дата, строка, вектор
- Задание на RAII: обёртка над файлом, обёртка над сокетом, обёртка над мьютексом

Тема 5
Наследование и полиморфизм

Теория
- Наследование, виды наследования, приведение типов
- Виртуальные функции, vtable, vptr, виртуальный деструктор
- Абстрактные классы, чистые виртуальные функции

Домашнее задание
- Задание на наследование: геометрические фигуры, иерархия сотрудников, иерархия транспорта
- Задание на полиморфизм: калькулятор фигур, сериализация, обработчик событий

Тема 6
Шаблоны и STL

Теория
- Шаблоны функций и классов, специализация, инстанцирование
- Итераторы, контейнеры, алгоритмы, лямбды
- Шаблоны с несколькими параметрами, variadic templates, std::pair, std::tuple

Домашнее задание
- Задание на шаблоны: универсальный контейнер, универсальная сортировка, универсальный фильтр
- Задание на STL: частотный анализатор слов, обработка CSV, индекс по ключевым словам

Тема 7
Исключения и RAII

Теория
- Исключения, try/catch/throw, иерархия исключений
- Гарантии безопасности: basic, strong, nothrow
- RAII, scope guard, транзакционная модель

Домашнее задание
- Задание на исключения: парсер с обработкой ошибок, валидатор, калькулятор с проверкой переполнения
- Задание на RAII: транзакционный файл, scope guard, транзакционный менеджер

Тема 8
Файлы и потоки, сериализация, парсинг

Теория
- Потоки, буферизация, бинарный ввод-вывод
- Сериализация: бинарная, текстовая, JSON, CSV
- Парсинг: recursive descent, state machine, regex

Домашнее задание
- Задание на потоки: копирование файла, подсчёт строк, сортировка больших файлов
- Задание на сериализацию: сохранение/загрузка графа, сериализация дерева, сериализация таблицы
- Задание на парсинг: парсер CSV, парсер JSON, парсер конфигурационного файла

Тема 9
Многопоточность и синхронизация

Теория
- Потоки, mutex, condition_variable, atomic, memory ordering
- Deadlock, livelock, race condition, false sharing
- Thread pool, producer-consumer, lock-free structures

Домашнее задание
- Задание на потоки: параллельная сортировка, параллельный поиск, параллельный обход файлов
- Задание на синхронизацию: thread pool, producer-consumer, lock-free counter
- Задание на lock-free: lock-free stack, lock-free queue, lock-free hash table

Тема 10
Сокеты и сетевое программирование

Теория
- Сокеты, TCP, UDP, адресация, порты, DNS
- Клиент-сервер, протоколы, сериализация сообщений
- Асинхронный ввод-вывод, epoll, event loop

Домашнее задание
- Задание на сокеты: эхо-сервер, эхо-клиент, чат, передача файлов
- Задание на протоколы: сериализация сообщений, парсинг HTTP, простой веб-сервер
- Задание на асинхронность: асинхронный эхо-сервер, event loop, асинхронный клиент

Тема 11
Итоговый проект

Проект
- Выбор проекта из предложенного списка
- Проектирование: архитектура, интерфейсы, разбиение на модули
- Реализация, тестирование, отладка, профилирование
- Документация, README, сборка, CI

Защита
- Презентация проекта
- Ответы на вопросы преподавателя и ревьюеров
- Обсуждение: что сделано хорошо, что можно улучшить

Модуль 2. Архитектура

Тема 1
Git: ветвление, merge, rebase, cherry-pick, bisect, hooks, submodules, worktree

Теория
- Ветвление, merge, rebase, cherry-pick
- Git bisect, hooks, submodules
- Git worktree, git rerere, git filter-branch

Домашнее задание
- Задание на ветвление и разрешение конфликтов: сценарии merge и rebase, разрешение конфликтов, git rerere
- Задание на bisect: найти коммит, в котором появился баг, автоматизация с bisect run
- Задание на hooks: pre-commit hook, commit-msg hook, pre-push hook

Тема 2
Сборка: Make, CMake, Ninja, Conan, vcpkg

Теория
- Make: цели, переменные, шаблонные правила, phony-цели, clean
- CMake: target-based, find_package, FetchContent, presets, CTest
- Ninja, Conan, vcpkg: быстрая сборка, управление зависимостями, кросс-компиляция

Домашнее задание
- Задание на Make: Makefile для проекта из нескольких файлов, с тестами и clean
- Задание на CMake: CMakeLists.txt для библиотеки и приложения, подключение Google Test через FetchContent
- Задание на Conan/vcpkg: подключение сторонней библиотеки, сборка с зависимостями

Тема 3
Отладка: gdb, lldb, strace, ltrace, perf, valgrind, sanitizers

Теория
- gdb: breakpoints, watchpoints, backtrace, core dump, attach, TUI
- strace, ltrace, perf: системные вызовы, библиотечные вызовы, профилирование
- Valgrind, ASan, UBSan, TSan, MSan: memcheck, leak check, sanitizers

Домашнее задание
- Задание на gdb: найти и исправить баг в предоставленном коде, с использованием breakpoints, watchpoints, backtrace
- Задание на strace/ltrace: анализ системных вызовов и вызовов библиотек, определение узкого места
- Задание на sanitizers: найти утечку памяти, переполнение буфера, гонку данных

Тема 4
Тестирование: Google Test, Google Mock, property-based testing, fuzzing, code coverage, CI/CD

Теория
- Google Test: assertions, fixtures, parameterized tests, death tests
- Google Mock: mock-объекты, expectations, matchers, actions
- Property-based testing, fuzzing, code coverage, CI/CD

Домашнее задание
- Задание на unit-тесты: тесты для библиотеки из предыдущего модуля, покрытие не менее 80%
- Задание на mock-объекты: тестирование клиента сетевого протокола, тестирование обработчика событий
- Задание на property-based testing: свойства для функции сортировки, свойства для парсера

Тема 5
Профилирование и оптимизация: perf, flamegraph, cachegrind, callgrind, massif, hotspot, heaptrack

Теория
- perf: sampling, counting, flamegraph, hotspot
- cachegrind, callgrind: кэш-промахи, граф вызовов, kcachegrind
- massif, heaptrack: профилирование памяти, flame graph для памяти

Домашнее задание
- Задание на perf: найти горячие точки в предоставленном коде, построить flamegraph
- Задание на cachegrind: найти кэш-промахи, сравнить версии до и после оптимизации
- Задание на heaptrack: найти утечки памяти, построить flame graph для памяти

Тема 6
Docker: Dockerfile, multi-stage, compose, volumes, networks, debug, CI/CD

Теория
- Dockerfile: FROM, RUN, COPY, ENTRYPOINT, multi-stage builds
- Docker compose, volumes, networks, healthcheck
- Отладка контейнеров, CI/CD с Docker

Домашнее задание
- Задание на Dockerfile: Dockerfile для своего проекта, multi-stage build, минимальный размер образа
- Задание на compose: compose-файл для приложения с базой данных и кэшем
- Задание на CI/CD: сборка и тесты в контейнере, docker compose в CI

Тема 7
Git: Git internals, reflog, submodules, worktree, git rerere, bisect run, hooks, git filter-branch

Теория
- Git internals: blob, tree, commit, tag, refs, HEAD, packfile
- Reflog, submodules, worktree, git rerere
- Git bisect run, hooks, git filter-branch

Домашнее задание
- Задание на git internals: вручную создать коммит, ветку, тег, разобрать packfile
- Задание на reflog и worktree: восстановить потерянный коммит, параллельная разработка в нескольких worktree
- Задание на bisect run: автоматическая бисекция с тестом, git filter-branch для удаления файла из истории

Тема 8
Сборка: Make, CMake, Ninja, Conan, vcpkg

Теория
- Make: цели, переменные, шаблонные правила, phony-цели, clean
- CMake: target-based, find_package, FetchContent, presets, CTest
- Ninja, Conan, vcpkg: быстрая сборка, управление зависимостями, кросс-компиляция

Домашнее задание
- Задание на Make: Makefile для проекта из нескольких файлов, с тестами и clean
- Задание на CMake: CMakeLists.txt для библиотеки и приложения, подключение Google Test через FetchContent
- Задание на Conan/vcpkg: подключение сторонней библиотеки, сборка с зависимостями

Тема 9
Отладка: gdb, lldb, strace, ltrace, perf, valgrind, sanitizers

Теория
- gdb: breakpoints, watchpoints, backtrace, core dump, attach, TUI
- strace, ltrace, perf: системные вызовы, библиотечные вызовы, профилирование
- Valgrind, ASan, UBSan, TSan, MSan: memcheck, leak check, sanitizers

Домашнее задание
- Задание на gdb: найти и исправить баг в предоставленном коде, с использованием breakpoints, watchpoints, backtrace
- Задание на strace/ltrace: анализ системных вызовов и вызовов библиотек, определение узкого места
- Задание на sanitizers: найти утечку памяти, переполнение буфера, гонку данных

Тема 10
Тестирование: Google Test, Google Mock, property-based testing, fuzzing, code coverage, CI/CD

Теория
- Google Test: assertions, fixtures, parameterized tests, death tests
- Google Mock: mock-объекты, expectations, matchers, actions
- Property-based testing, fuzzing, code coverage, CI/CD

Домашнее задание
- Задание на unit-тесты: тесты для библиотеки из предыдущего модуля, покрытие не менее 80%
- Задание на mock-объекты: тестирование клиента сетевого протокола, тестирование обработчика событий
- Задание на property-based testing: свойства для функции сортировки, свойства для парсера

Тема 11
Профилирование и оптимизация: perf, flamegraph, cachegrind, callgrind, massif, hotspot, heaptrack

Теория
- perf: sampling, counting, flamegraph, hotspot
- cachegrind, callgrind: кэш-промахи, граф вызовов, kcachegrind
- massif, heaptrack: профилирование памяти, flame graph для памяти

Домашнее задание
- Задание на perf: найти горячие точки в предоставленном коде, построить flamegraph
- Задание на cachegrind: найти кэш-промахи, сравнить версии до и после оптимизации
- Задание на heaptrack: найти утечки памяти, построить flame graph для памяти

Тема 12
Docker: Dockerfile, multi-stage, compose, volumes, networks, debug, CI/CD

Теория
- Dockerfile: FROM, RUN, COPY, ENTRYPOINT, multi-stage builds
- Docker compose, volumes, networks, healthcheck
- Отладка контейнеров, CI/CD с Docker

Домашнее задание
- Задание на Dockerfile: Dockerfile для своего проекта, multi-stage build, минимальный размер образа
- Задание на compose: compose-файл для приложения с базой данных и кэшем
- Задание на CI/CD: сборка и тесты в контейнере, docker compose в CI

Тема 13
Git: Git internals, reflog, submodules, worktree, git rerere, bisect run, hooks, git filter-branch

Теория
- Git internals: blob, tree, commit, tag, refs, HEAD, packfile
- Reflog, submodules, worktree, git rerere
- Git bisect run, hooks, git filter-branch

Домашнее задание
- Задание на git internals: вручную создать коммит, ветку, тег, разобрать packfile
- Задание на reflog и worktree: восстановить потерянный коммит, параллельная разработка в нескольких worktree
- Задание на bisect run: автоматическая бисекция с тестом, git filter-branch для удаления файла из истории

Тема 14
Итоговый проект

Проект
- Выбор проекта из предложенного списка
- Проектирование: архитектура, интерфейсы, разбиение на модули
- Реализация, тестирование, отладка, профилирование
- Документация, README, сборка, CI

Защита
- Презентация проекта
- Ответы на вопросы преподавателя и ревьюеров
- Обсуждение: что сделано хорошо, что можно улучшить

Модуль 3. Параллельные вычисления

Тема 1
Многопоточность

Теория
- Потоки и процессы, жизненный цикл, pthread_create, pthread_join, pthread_detach, pthread_exit, pthread_cancel
- Передача аргументов, возвращаемые значения, stack size, affinity
- Race condition, deadlock, livelock, starvation, priority inversion

Домашнее задание
- Задание на потоки: параллельная сортировка, параллельный поиск, параллельный обход файлов

Тема 2
Синхронизация

Теория
- Mutex, condition_variable, atomic, memory ordering
- Deadlock, livelock, race condition, false sharing
- Thread pool, producer-consumer, lock-free structures

Домашнее задание
- Задание на синхронизацию: thread pool, producer-consumer, lock-free counter
- Задание на lock-free: lock-free stack, lock-free queue, lock-free hash table

Тема 3
Сокеты и сетевое программирование

Теория
- Сокеты, TCP, UDP, адресация, порты, DNS
- Клиент-сервер, протоколы, сериализация сообщений
- Асинхронный ввод-вывод, epoll, event loop

Домашнее задание
- Задание на сокеты: эхо-сервер, эхо-клиент, чат, передача файлов
- Задание на протоколы: сериализация сообщений, парсинг HTTP, простой веб-сервер
- Задание на асинхронность: асинхронный эхо-сервер, event loop, асинхронный клиент

Тема 4
Итоговый проект

Проект
- Выбор проекта из предложенного списка
- Проектирование: архитектура, интерфейсы, разбиение на модули
- Реализация, тестирование, отладка, профилирование
- Документация, README, сборка, CI

Защита
- Презентация проекта
- Ответы на вопросы преподавателя и ревьюеров
- Обсуждение: что сделано хорошо, что можно улучшить

Модуль 4. Сеть

Тема 1
Linux: сетевой стек, netfilter/iptables, nftables, tc, ip, ss, netstat, tcpdump, wireshark

Теория
- Сетевой стек Linux, пакеты, сокеты, маршрутизация
- netfilter/iptables, nftables, conntrack, NAT, MASQUERADE
- tc: qdisc, class, filter, traffic control, ограничение скорости, задержка, потери, дублирование

Домашнее задание
- Задание на netfilter: настроить firewall для сервера, разрешить только SSH и HTTP, логирование отброшенных пакетов
- Задание на tc: ограничить скорость исходящего трафика до 1 Мбит/с, добавить задержку 100 мс, потери 5%
- Задание на анализ: захватить трафик curl в pcap, разобрать в wireshark, найти TCP-handshake и HTTP-запросы

Тема 2
Асинхронный ввод-вывод

Теория
- Мультиплексирование, select, poll, epoll, event loop
- Неблокирующий ввод-вывод, callbacks, state machine
- Сравнение подходов: thread-per-connection, event loop, io_uring

Домашнее задание
- Задание на epoll: эхо-сервер на epoll, обработка нескольких клиентов в одном потоке
- Задание на state machine: парсер протокола, обработка частичных данных, буферизация
- Задание на сравнение: реализовать эхо-сервер тремя способами, замерить latency и throughput

Тема 3
Linux: сетевой стек, netfilter/iptables, nftables, tc, ip, ss, netstat, tcpdump, wireshark

Теория
- Сетевой стек Linux, пакеты, сокеты, маршрутизация
- netfilter/iptables, nftables, conntrack, NAT, MASQUERADE
- tc: qdisc, class, filter, traffic control, ограничение скорости, задержка, потери, дублирование

Домашнее задание
- Задание на netfilter: настроить firewall для сервера, разрешить только SSH и HTTP, логирование отброшенных пакетов
- Задание на tc: ограничить скорость исходящего трафика до 1 Мбит/с, добавить задержку 100 мс, потери 5%
- Задание на анализ: захватить трафик curl в pcap, разобрать в wireshark, найти TCP-handshake и HTTP-запросы

Тема 4
Асинхронный ввод-вывод

Теория
- Мультиплексирование, select, poll, epoll, event loop
- Неблокирующий ввод-вывод, callbacks, state machine
- Сравнение подходов: thread-per-connection, event loop, io_uring

Домашнее задание
- Задание на epoll: эхо-сервер на epoll, обработка нескольких клиентов в одном потоке
- Задание на state machine: парсер протокола, обработка частичных данных, буферизация
- Задание на сравнение: реализовать эхо-сервер тремя способами, замерить latency и throughput

Тема 5
Итоговый проект

Проект
- Выбор проекта из предложенного списка
- Проектирование: архитектура, интерфейсы, разбиение на модули
- Реализация, тестирование, отладка, профилирование
- Документация, README, сборка, CI

Защита
- Презентация проекта
- Ответы на вопросы преподавателя и ревьюеров
- Обсуждение: что сделано хорошо, что можно улучшить

Модуль 5. Хранилища

Тема 1
SQLite3 API

Теория
- SQLite3 API: prepared statements, bind, step, column, finalize
- Транзакции, WAL, busy_timeout, PRAGMA
- Схема БД, миграции, user_version

Домашнее задание
- Задание на SQLite3 API: обёртка над SQLite3 API на языке C++, RAII для prepared statements и транзакций
- Задание на миграции: миграции по user_version, откат миграции при ошибке
- Задание на производительность: замерить latency и throughput, сравнить с fsync=off

Тема 2
Key-Value

Теория
- Key-Value store: get, set, del, exists, TTL, persistence
- Append-only log, snapshot, compaction
- Redis protocol, RESP, pipelining

Домашнее задание
- Задание на Key-Value store: in-memory key-value store с поддержкой TTL
- Задание на persistence: append-only log, snapshot, compaction
- Задание на протокол: реализовать парсер RESP, клиент redis-cli, pipelining

Тема 3
Многопоточность и БД

Теория
- Многопоточность и БД: thread-safety, connection pool, transaction isolation
- Deadlock, retry, exponential backoff, idempotency
- Lock-free structures, MVCC, snapshot isolation

Домашнее задание
- Задание на thread-safety: thread-safe обёртка над SQLite3, connection pool, retry с exponential backoff
- Задание на lock-free: lock-free stack, lock-free queue, lock-free hash table
- Задание на MVCC: реализовать snapshot isolation, version chain, garbage collection

Тема 4
Итоговый проект

Проект
- Выбор проекта из предложенного списка
- Проектирование: архитектура, интерфейсы, разбиение на модули
- Реализация, тестирование, отладка, профилирование
- Документация, README, сборка, CI

Защита
- Презентация проекта
- Ответы на вопросы преподавателя и ревьюеров
- Обсуждение: что сделано хорошо, что можно улучшить

Модуль 6. ML

Тема 1
Numpy: N-мерные массивы

Теория
- Numpy: ndarray, dtype, shape, strides, broadcasting, views vs copy
- Векторизация, ufunc, axis, reshape, slicing
- Numpy vs loops: почему numpy быстрее, memory layout, cache locality

Домашнее задание
- Задание на numpy: нормализация матрицы, softmax, one-hot encoding, cosine similarity, batch processing
- Задание на сравнение: сравнить numpy vs pure Python loops, замерить speedup, объяснить почему

Тема 2
Pandas: DataFrame, Series, groupby, merge, pivot, time series, pipelines

Теория
- Pandas: DataFrame, Series, dtypes, loc, iloc, boolean indexing
- groupby, merge, pivot, melt, time series, resampling
- Pipelines, apply, vectorized operations, memory optimization

Домашнее задание
- Задание на pandas: загрузка и очистка данных, groupby, merge, pivot, time series
- Задание на pipelines: цепочка трансформаций, custom transformer, сравнение с numpy

Тема 3
Итоговый проект

Проект
- Выбор проекта из предложенного списка
- Проектирование: архитектура, интерфейсы, разбиение на модули
- Реализация, тестирование, отладка, профилирование
- Документация, README, сборка, CI

Защита
- Презентация проекта
- Ответы на вопросы преподавателя и ревьюеров
- Обсуждение: что сделано хорошо, что можно улучшить

Модуль 7. Проектная работа

Тема 1
Выбор и проектирование проекта

Теория
- Выбор и проектирование проекта, постановка задачи, MVP, roadmap
- Архитектура, интерфейсы, разбиение на модули, выбор технологий
- Оценка сложности, риски, критерии приёмки

Домашнее задание
- Задание на проектирование: выбрать проект из предложенного списка, составить ТЗ, нарисовать диаграмму модулей, оценить сроки

Тема 2
Реализация

Теория
- Реализация, тестирование, отладка, профилирование, CI/CD
- Code review, рефакторинг, документация, README, сборка

Домашнее задание
- Задание на реализацию: реализовать MVP по ТЗ, покрыть тестами, настроить CI, написать README

Тема 3
Защита проекта

Теория
- Подготовка к защите: презентация, демонстрация, ответы на вопросы
- Ревью: что сделано хорошо, что можно улучшить, что изучить дальше
- Рекомендации: следующие проекты, ресурсы, сообщество, карьера

Домашнее задание
- Задание на подготовку к защите: подготовить презентацию, отрепетировать демонстрацию, подготовить ответы на вопросы
- Задание на ревью: оценить проекты других студентов, дать обратную связь, выбрать лучшие проекты