Разработчик C++ расширенный

12 месяцев
продолжительность курса

12 проектов
в портфолио

01	**Основы C++ с Qt**	Спринт 1 Спринт 2 Спринт 3 Спринт 4 Спринт 5	10 недель, 160 часов
02	**Производительность и оптимизация**	Спринт 6 Спринт 7 Спринт 8	6 недель, 105 часов
03	**Качество кода**	Спринт 9 Спринт 10 Спринт 11 Спринт 12 Спринт 13	11 недель, 180 часов
04	**Проектирование сложных программ**	Спринт 14 Спринт 15	4 недели, 80 часов
05	**Дипломный проект "Электронная таблица"**		3 недели, 45 часов
06	**C++ для бэкенда**	Спринт 16 Спринт 17 Спринт 18 Спринт 19	10 недель, 150 часов

Основы C++ с Qt 01

Изучите как базовые основы C++, так и некоторые продвинутые возможности. Особое внимание в модуле уделено созданию графических приложений. Используя фреймфорк Qt, вы создадите более 10 полноценных полезных программ, таких как будильник, фотоальбом и некоторые другие.

[10 недель, 160 часов], [2 проекта]

1. Графический калькулятор.
Создадите калькулятор с адаптивным интерфейсом, производящий вычисления над числами различных типов. В числе особенностей калькулятора — работа с дробями.

2. Интерфейс для дека.
Применяя паттерн MVC, создадите графическую оболочку контейнера дек, поддерживающую большинство операций над контейнером. Также в ней будет реализован бинарный поиск и сортировка.

Спринт 1

1. Онбординг

2. Hello, C++
Изучите базовые понятия программирования на примере C++ — переменные, типы, операторы.

3. Условные конструкции
Поработаете с логическими выражениями и операторами ветвления.

4. Циклы и алгоритмы
Изучите операторы циклов, а также базовые и широкоиспользуемые алгоритмы.

5. Функции
Научитесь писать свои функции и вызывать их, передавать и принимать аргументы. Также поделите программу на файлы.

6. Система Git
Узнаете основные команды Git и поработаете с репозиторием.

Спринт 2

7. Классы
Узнаете, как работать с агрегированными типами. Изучите такие понятие, как класс, метод, объект.

8. Классы. Продолжение
Более подробно изучите классы. Научитесь писать конструкторы и константные методы.

9. Знакомство с Qt
Установите Qt и создадите в нём графическое приложение.

10. Создаем графическое приложение
Изучите основные элементы Qt, компоновки, создадите несколько полноценных графических программ.

Спринт 3

11. Перегрузка операций
Узнаете, как добавить поддержку операций для собственных типов. Научитесь делать кастомизированный ввод-вывод.

12. Шаблоны
Изучите основы метапрограммирования, создадите шаблонные классы и функции.

13. Лямбды
Поработаете с функциональными объектами, компараторами, предикатами и функциями обратного вызова.

Спринт 4

14. И снова вектор
Углубитесь в устройство вектора, изучите итераторы и создадите графический интерфейс для вектора.

15. Ассоциативные контейнеры
Изучите контейнеры map и set, концепцию словаря, а также контейнеры Qt.

16. Другие контейнеры и поиск
Изучите контейнеры стек, очередь, дек, алгоритм бинарного поиска и рекурсию.

Спринт 5

17. Написание тестов и обработка ошибок
Попробуете один из фрейморков для написания тестов. Примените логирование в своих программах. Изучите обработку ошибок через исключения.

Производительность и оптимизация

02

Научитесь оценивать эффективность высокопроизводительного приложения до и после его разработки. Изучите работу с файлами и регулярными выражениями. Познакомитесь со внутренним устройством контейнеров. Это позволит понять, как настраивать элементы языка на эффективную работу. Чтобы избегать лишних копирований в коде, освоите move-семантику.

[6 недель, 105 часов], [3 проекта]

1. Свой препроцессор.

Самостоятельно реализуете часть C++, а именно препроцессор, который проходит исходный код до компилятора и обрабатывает директивы.

2. Односвязный список.

Создадите собственную версию односвязного списка - контейнера, поддерживающего эффективное добавление и удаление элементов.

3. Простой вектор.

Напишете собственную версию вектора - контейнера, который обеспечивает эффективный доступ к элементам.

Спринт 6

1. Профилируем и ускоряем

Узнаете, зачем нужна профилировка, упростите логирование, чтобы ускорить программу.

2. Потоковые хитрости

Познакомитесь с буфером и его возможностями, узнаете, какой может быть связь потоков.

3. Просто о сложности. Теория быстродействия

Поймёте, что такое асимптотическая сложность. Научитесь оценивать алгоритмы по сложности и разберётесь, как оценивается сложность нескольких алгоритмов.

4. Работа с файлами

Продолжите знакомство с буфером и потоками и научитесь работать с ними через файлы.

Спринт 7

5. Модель памяти в C++

Изучите атомарность, эксклюзивность и изменение порядка. Рассмотрите примеры переупорядочивания памяти. Узнаете, что такое «модель памяти», Volatile и атомарные переменные. Рассмотрите примеры сломанного кода и решения, как его починить.

6. Односвязный список

Познакомитесь с тем, как устроен односвязный список, вставка и очистка элементов. RAII обертка для массивов.

Спринт 8

7. Эффективные линейные контейнеры
Узнаете о контейнерах и итераторах, какие бывают категории итераторов. Элементы стандартной библиотеки. Линейные контейнеры. Односвязный и двусвязный список.

8. Семантика перемещения
Введение в move, rvalue и lvalue. Универсальные ссылки. Свёртывание ссылок. Copy/move elision.

Качество кода 03

Изучите инструменты, которые делают код опрятным и защищают его от человеческих ошибок: пространства имён, константность, RAII. Узнаете, как избавить программу от проблем роста. Освоите динамический полиморфизм.

[11 недель, 180 часов],
[1 проект]

1. Транспортный справочник.
Напишете программу, позволяющую хранить информацию об автобусных остановках и автобусах. Затем добавите возможность строить автобусные маршруты в городе.

Спринт 9

1. Ассоциативные контейнеры
Изучите принципы объектно ориентированного програмиирования. Разберёте полиморфизм. Также узнаете, что такое разнородный поиск и множества.

2. Имена и пространства имён
Познакомитесь с инструментированием кода для выявления ошибок в среде выполнения. Начнете использование санитайзеров. Научитесь создавать отчеты об ошибках.

Спринт 10

3. Умные указатели
Изучите указатели unique_ptr, shared_ptr и weak_ptr, их понятия и применение.

4. Undefined Behavior
Узнаете, что такое неопределенное поведение, точки следования и побочные эффекты.

Спринт 11

5. Наследование и полиморфизм
Познакомитесь с понятием и синтаксисом наследования и полиморфизма. Узнаете, что такое статическая привязка и виртуальная функция.

Спринт 12

6. Константность как элемент проектирования программ
Продолжите осваивать основы объектно ориентированного программирования. Подробнее изучите константные объекты классов, функции-члены и получение константных объектов через передачу по константной ссылке.

7. Идиома RAII
Научитесь управлять временем жизни и ресурсами объекта.

Спринт 13

8. Философия хороших функций
Поймёте, что лучше использовать: функцию или метод класса.

9. Передача данных
Как лучше передавать данные: по ссылке или по значению. Посмотрите ассемблер. Запустите бенчмарки. Познакомитесь с Forwarding reference. Узнаете 3 способа передачи функций.

10. Получение данных из функции
Узнаете, как лучше получать данные из функции.

11. Рефакторинг
Изучите методы и средства рефакторинга, а также познакомитесь с 3 типами конструкторов.

Проектирование сложных программ

04

Напишете полноценный вектор, познакомитесь с variadic templates и гарантиями безопасности исключений.

[4 недели, 80 часов], [2 проекта]

1. Контейнер vector.

Разработаете вектор, который автоматически увеличивает свой размер, эффективно работает с памятью и не копирует объекты зря.

2. Конвертер изображений.

Напишете программу, которая будет переводить изображения из одного формата в другой. Она будет поддерживать BMP, PPM, JPEG.

Спринт 14

1. Vector своими руками

Что нужно знать для реализации: указатели, move-семантика, rValue и lValue ссылки, шаблоны, итераторы, переопределение операторов. Разработаете вектор, который автоматически увеличивает свой размер и эффективно работает с памятью.

2. Хранение объектов в памяти

Узнаете, что такое оперативная память, представление объектов, указатели. Подробнее изучите динамическую память и стек. Также сможете отслеживать утечки памяти.

Спринт 15

3. Таблица виртуальных методов

Изучите конструкторы и деструкторы, таблицу виртуальных методов.

4. Сборка по-новому. CMake

Узнаете изнутри, как работает полиморфизм, изучите популярную систему сборки CMake, научитесь пользоваться сторонними библиотеками и напишете свои.

Дипломный проект «Электронная таблица» 05

В заключение вы напишете настоящий дипломный проект — крупную программу, которую сможете развивать. Именно так вы закрепите свои навыки написания понятного и масштабируемого кода.

[3 недели, 45 часов], [1 проект]

Электронная таблица.
Сначала вы защитите дизайн-проект таблицы с ячейками и формулами, затем напишете код.

C++ для бэкенда 06

Расширенный формат курса включает в себя больше тем и проектов. На расширенном курсе вы освоите C++ для бэкенда.

- Разработаете программное обеспечение под Linux.
- Получите навык нагрузочного тестирования.
- Научитесь работать с Docker и СУБД PostgreSQL.
- Расширите свои знания о библиотеках разработки на C++ и используете в проекте boost:asio и boost:beast.

[10 недель, 150 часов], [3 проекта]

1. Мониторинг и логирование.
Разработаете собственный дашборд на основе логирования Grafana и Prometheus, а также проведёте нагрузочное тестирование, используя Яндекс Танк.

2. Букипедия.
Создадите приложение, использующее БД PostgreSQL для хранения данных. При этом ваш код сможет продемонстрировать добавление, удаление и вывод всего списка именований из базы данных.

3. Четвероногий курьер.
Напишете бэкенд для небольшой игры, в которой игроки-курьеры соревнуются в том, кто оптимальнее пройдет по лабиринту улиц и доставит максимальное количество заказов.

Спринт 16

1. Hello, Linux!
Настроите рабочее окружение: поднимите виртуальную машину с операционной системой Linux и установите библиотеку Boost.

2. Hello, web-server!
Рассмотрите сетевые протоколы: TCP, UDP, HTTP. Разработаете собственный веб-сервер.

3. Hello, docker!
Научитесь устанавливать Docker и создавать образ проекта.

Спринт 17

4. Передаем данные через сеть
Подключите к своему проекту базу данных и СУБД PostgreSQL.

5. Эффективное логирование
Нырнёте в мир библиотеки Boost.Log и научитесь оптимально хранить большие объёмы логов.

6. Авторизация и аутентификация
Освоите базовые техники авторизации и аутентификации пользователей.

7. Механизмы синхронизации
Узнаете о тредах и процессах в операционной системе, состоянии гонки и методах синхронизации.

8. Мониторинг
Наладите мониторинг системных ресурсов в операционной системе Linux и настроите отправку уведомлений в случаях, когда память переполняется. Это позволит держать ресурсы системы под контролем.

Спринт 18

9. Распараллеливание
Изучите распараллеливание вычислений между тредами, фреймворк для тестов и варианты отладки программ.

10. Тестирование и отладка
Фреймворк для тестов catch. Обзор GBD. Выбор файлов. Отладка программы через присоединение к выполняемому процессу с помощью gdb. Вход и выход из GDB. Вызов GDB.

11. Профилирование
Стек-трейс. Тред-дамп. Понятия, назначение, использование. Аналитика программного года через статистику путей выполнения.

12. Нагрузочное тестирование
Понятие нагрузочного тестирования и его места в верификации программного решения.

Спринт 19

13. Сохранение состояния системы
Описание состояния системы. Сериализация. Сохранение и восстановление состояния системы.

14. База данных PostgresSQL
Познакомитесь с архитектурой баз данных и научитесь отправлять в них запросы. Узнаете, как описывать, сохранять и восстанавливать состояние системы. Освоите SQL.

15. Продвинутая работа с базой данных
Подключите к своему проекту базу данных и СУБД PostgreSQL.

Медведев Кирилл
Директор АНО ДПО «Образовательные технологии Яндекса»