### План обучения для студентов-программистов

### (на C++ с использованием Docker, Kubernetes)

**Уровни сложности проекта:**

BASE EDITION - необходим каждому Backend-разработчику, а некоторые модули важны в любом направлении разработки, например Docker.

EXTENDED EDITION - расширенные знания, необходимые DevOps инженерам и Backend-разработчикам. Крайне рекомендуется к изучению.

ULTIMATE EDITION - данный уровень предназначен для тех, кто уже хочет погрузиться в профессию и изучить основы дисциплин старших курсов. Полное выполнение данного курса заданий позволит овладеть всеми необходимыми основами Backend-разработки до уровня стажера на производстве. После данного уровня изучение данных тем, с использованием других технологий и языков, станет гораздо проще и приятнее.

#### Идея сервиса

### Создадим сервис, предоставляющий различные эндпоинты, которые возвращают полезную информацию, различные вычисления и интересные данные. Всё будет работать без Frontend.

#### Задание 1. Введение в **Backend-**разработку и Docker (BASE EDITION)

***Основы Backend-разработки***

* 1. Изучить основы клиент-серверной архитектуры:

<https://habr.com/ru/articles/495698/>

<https://habr.com/ru/articles/483202/>

<https://www.youtube.com/watch?v=08-RlaLgWx4&ab_channel=MerionAcademy>

* 1. Подготовить среду разработки, установить необходимое ПО (компилятор C++, Docker).

***Введение в Docker***

* 1. Изучить что такое Docker и зачем он нужен.

<https://habr.com/ru/articles/810777/>

<https://www.youtube.com/watch?v=aZTL2zRmOnA&ab_channel=MerionAcademy>

* 1. Установить Docker, изучить основные команды (docker run, docker build, docker images, docker ps).
  2. Ознакомиться с документацией Docker для начинающих.

#### Задание 2. Создание и контейнеризация простого C++ приложения (BASE EDITION)

***Создание Dockerfile для C++ приложения***

* 1. Изучить Dockerfile: основные инструкции.
  2. Написать Dockerfile для контейнеризации C++ приложения, собрать образ и запустить контейнер.

#### Задание 3. Разработка REST API на C++ (BASE EDITION)

***Основы REST API***

* 1. Изучить принципы REST API.
  2. Создать простое REST API приложение на C++ с использованием **Dragon.**

<https://habr.com/ru/articles/595901/>

<https://github.com/drogonframework/drogon/>

***Контейнеризация REST API***

* 1. Обновить Dockerfile для контейнеризации REST API приложения, собрать образ и запустить контейнер.

#### Задание 4. Создание эндпоинтов и контроллеров (BASE EDITION)

Выполнить разработку контроллера, который выполняет запрос и отправляет ответ по эндпоинту, заданному в индивидуальном варианте.

#### Задание 5.1. Основы Kubernetes (EXTENDED EDITION)

***Введение в Kubernetes***

* 1. Изучить что такое Kubernetes и как с ним работать.

<https://www.youtube.com/watch?v=klmpiHLSuXA&ab_channel=MerionAcademy>

* 1. Установить Minikube для локального развертывания кластера Kubernetes.

***Развертывание приложения в Kubernetes***

* 1. Изучить принципы создания и развертывания простого приложения в Kubernetes

<https://habr.com/ru/articles/589415/>

* 1. Создать Kubernetes манифесты для развертывания Docker контейнеров с приложением и настройка сервиса.

***Настройка и масштабирование приложения в Kubernetes***

* 1. Изучить основы масштабирования в Kubernetes.
  2. Настроить автоматическое масштабирование приложения.

#### Задание 5.2. Автоматическое тестирование (ULTIMATE EDITION)

***Введение в автоматическое тестирование***

* 1. Изучить основы unit-тестирования на C++.

<https://habr.com/ru/companies/tensor/articles/347358/>

* 1. Написать и запустить unit-тесты для C++ приложения.

***Настройка CI/CD***

* 1. Изучить основы CI/CD.

<https://habr.com/ru/companies/otus/articles/515078/>

<https://www.youtube.com/watch?v=Y-hYifHkjMs&ab_channel=MerionAcademy>

<https://habr.com/ru/articles/737148/>

<https://github.com/Eric-Jalal/cpp-pipeline>

<https://www.codeproject.com/Articles/5265628/Writing-CI-Pipeline-using-GitHub-Actions-to-Build>

* 1. Настроить простой CI/CD пайплайн с использованием GitHub Actions для автоматического тестирования и деплоя проекта.

### Результат

После завершения курса студенты смогут:

* разрабатывать и контейнеризировать C++ приложения;
* создавать и тестировать REST API;
* развёртывать приложения в Kubernetes;
* настраивать CI/CD пайплайны;
* разрабатывать полнофункциональные проекты, готовые к деплою в реальные среды.

Этот план поможет студентам овладеть современными инструментами и практиками Backend-разработки, получить ценные практические навыки и опыт.

**Варианты индивидуальных эндпоинтов:**

1. /api/hello - возвращает "Hello, World!".
2. /api/time - возвращает текущее время.
3. /api/date - возвращает текущую дату.
4. /api/random - возвращает случайное число.
5. /api/square/{number} - возвращает квадрат числа.
6. /api/factorial/{number} - возвращает факториал числа.
7. /api/fibonacci/{number} - возвращает число Фибоначчи.
8. /api/reverse/{string} - возвращает перевернутую строку.
9. /api/uppercase/{string} - возвращает строку в верхнем регистре.
10. /api/lowercase/{string} - возвращает строку в нижнем регистре.
11. /api/length/{string} - возвращает длину строки.
12. /api/isprime/{number} - проверяет, является ли число простым.
13. /api/gcd/{a}/{b} - возвращает наибольший общий делитель двух чисел.
14. /api/lcm/{a}/{b} - возвращает наименьшее общее кратное двух чисел.
15. /api/temperature/{celsius} - конвертирует температуру из Цельсия в Фаренгейт.
16. /api/weather - возвращает текущую погоду (фейковые данные).
17. /api/joke - возвращает случайную шутку.
18. /api/quote - возвращает случайную цитату.
19. /api/advice - возвращает случайный совет.
20. /api/name/{firstname}/{lastname} - возвращает полное имя.
21. /api/age/{year}/{month}/{day} - возвращает возраст на текущую дату.
22. /api/weekday/{year}/{month}/{day} - возвращает день недели для заданной даты.
23. /api/bmi/{weight}/{height} - рассчитывает индекс массы тела.
24. /api/area/{radius} - рассчитывает площадь круга.
25. /api/perimeter/{radius} - рассчитывает периметр круга.
26. /api/volume/{radius} - рассчитывает объем сферы.
27. /api/sqrt/{number} - возвращает квадратный корень числа.
28. /api/pow/{base}/{exponent} - возвращает число возведенное в степень.
29. /api/log/{number} - возвращает натуральный логарифм числа.
30. /api/exp/{number} - возвращает экспоненциальную функцию числа.
31. /api/palindrome/{string} - проверяет, является ли строка палиндромом.
32. /api/anagram/{string1}/{string2} - проверяет, являются ли строки анаграммами.
33. /api/sort/{numbers} - сортирует массив чисел.
34. /api/shuffle/{numbers} - перемешивает массив чисел.
35. /api/distance/{lat1}/{lon1}/{lat2}/{lon2} - рассчитывает расстояние между двумя географическими координатами.
36. /api/translate/{word}/{language} - возвращает перевод слова на указанный язык (использует фейковые данные).
37. /api/currency/{amount}/{from}/{to} - конвертирует сумму из одной валюты в другую (использует фейковые данные).
38. /api/morse/{string} - переводит строку в азбуку Морзе.
39. /api/binary/{number} - переводит число в двоичный формат.
40. /api/hex/{number} - переводит число в шестнадцатеричный формат.
41. /api/roman/{number} - переводит число в римские цифры.
42. /api/primefactors/{number} - возвращает простые множители числа.
43. /api/compoundinterest/{principal}/{rate}/{time} - рассчитывает сложный процент.
44. /api/simpleinterest/{principal}/{rate}/{time} - рассчитывает простой процент.
45. /api/leapyear/{year} - проверяет, является ли год високосным.
46. /api/season/{month} - возвращает сезон года для указанного месяца.
47. /api/capitalize/{string} - возвращает строку с заглавной буквы.
48. /api/titlecase/{string} - возвращает строку в формате заглавных слов.
49. /api/repeat/{string}/{times} - возвращает строку, повторенную указанное количество раз.
50. /api/uuid - генерирует уникальный идентификатор.
51. /api/password/{length} - генерирует случайный пароль указанной длины.
52. /api/wordcount/{string} - возвращает количество слов в строке.
53. /api/sentiment/{text} - анализирует текст и возвращает его эмоциональную окраску (позитивный, негативный, нейтральный) (фейковые данные).
54. /api/wordfrequency/{string} - возвращает частотный словарь для слов в строке.
55. /api/isbn/{number} - проверяет, является ли номер ISBN корректным.
56. /api/ipinfo/{ip} - возвращает информацию о IP-адресе (фейковые данные).
57. /api/hostname - возвращает имя хоста сервера.
58. /api/ispalindromenumber/{number} - проверяет, является ли число палиндромом.
59. /api/countryinfo/{country} - возвращает информацию о стране (фейковые данные).
60. /api/cityinfo/{city} - возвращает информацию о городе (фейковые данные).

Данные эндпоинты позволят студентам развить навыки работы с различными типами данных и алгоритмами, а также создать разнообразные и интересные API.