****

**Программа курса «Фронтенд-разработка»**

**Продолжительность курса – 168 часов.**

**Продолжительность одного занятия – 2 академических часа**

**Общее количество занятий – 42.**

**О курсе:**

Для создания программы курса по фронтенд-разработке, учитывая продолжительность в 168 часов с общим количеством занятий в 42 (каждое по   
2 академических часа), можно предложить следующий план. Курс будет охватывать основы веб-разработки, современные инструменты и фреймворки, включая HTML, CSS, JavaScript, React и дополнительные технологии для разработки и развертывания веб-приложений.

**Цели курса:**

После изучения курса слушатели смогут:

* Создавать современные веб-приложения, используя HTML, CSS, и JavaScript;
* Применять фреймворки и библиотеки, такие как React, для разработки интерфейсов;
* Отлаживать и тестировать фронтенд-приложения;
* Разбираться в принципах адаптивного и отзывчивого дизайна;
* Разрабатывать и интегрировать RESTful API;
* Понимать основы работы с Git и системами контроля версий;
* Знать методы развертывания веб-приложений на серверах и в облаке.

|  |  |  |
| --- | --- | --- |
| **Тематика аудиторных занятый:** | | |
| **№** | **Тема** | **Кол-во часов** |
| **Основы веб-разработки и JavaScript** | | **36** |
| 1 | Введение в HTML и CSS | 2 |
| 2 | Изучение структуры HTML-документа | 2 |
| 3 | Основы CSS, включая селекторы, свойства и макеты | 2 |
| 4 | Основы JavaScript | 2 |
| 5 | Переменные, типы данных, операторы | 4 |
| 6 | Управляющие конструкции: условия и циклы | 4 |
| 7 | Функции и объекты в JavaScript | 2 |
| 8 | Объявление и вызов функций | 4 |
| 9 | Работа с объектами и массивами | 4 |
| 10 | Введение в DOM и события | 2 |
| 11 | Манипуляции с DOM | 4 |
| 12 | Обработка событий | 4 |
| **Продвинутый JavaScript и введение в React** | | **44** |
| 1 | Продвинутые темы JavaScript | 3 |
| 2 | Замыкания, контекст выполнения и this | 3 |
| 3 | Промисы и асинхронное программирование | 3 |
| 4 | Введение в React | 3 |
| 5 | Основы React: JSX, компоненты, props и state | 4 |
| 6 | Создание первого приложения на React | 4 |
| 7 | Углубление в React | 4 |
| 8 | Управление состоянием компонентов | 4 |
| 9 | Жизненный цикл компонентов | 4 |
| 10 | Работа с формами и списками в React | 4 |
| 11 | Обработка форм | 4 |
| 12 | Рендеринг списков и ключи | 4 |
| **Маршрутизация, стейт-менеджмент и API** | | **36** |
| 1 | Маршрутизация в React-приложениях | 4 |
| 2 | React Router: основы маршрутизации | 4 |
| 3 | Управление глобальным состоянием | 4 |
| 4 | Введение в Redux или Context API | 4 |
| 5 | Работа с API | 4 |
| 6 | Fetch API/Axios для работы с внешними API | 4 |
| 7 | Подключение к простому REST API и вывод данных | 4 |
| 8 | Практика и проект | 4 |
| 9 | Разработка небольшого приложения, использующего API, маршрутизацию и управление состоянием | 4 |
| **Дополнительные инструменты и завершение проекта** | | **52** |
| 1 | Инструменты сборки и развертывания | 4 |
| 2 | Webpack, Babel | 8 |
| 3 | Развертывание приложения на платформе типа Netlify или Vercel | 8 |
| 4 | Планирование и разработка собственного проекта на React | 8 |
| 5 | Работа над пользовательским интерфейсом, взаимодействием с API и маршрутизацией | 8 |
| 6 | Тестирование, отладка и оптимизация проекта | 8 |
| 7 | Подготовка к собеседованию: типовые вопросы и задачи по React, портфолио | 8 |
| Итого: | | **168** |

****

**Программа курса «Язык программирования Python»**

**Продолжительность курса – 168 часов.**

**Продолжительность одного занятия – 2 академических часа**

**Общее количество занятий – 47.**

Язык программирования Python — это мощный инструмент для создания программ самого разнообразного назначения. Это язык программирования общего назначения, который применяется в таких проектах как Google, NASA, Instagram.

Для успешного изучения материалов курса, слушатель должен обладать следующим навыками и знаниями:

* Умение работать с персональным компьютером;
* Знание основ операционных систем;
* Знание основ программирования;
* Желательно знание технического английского языка.

После изучения пользователь будет:

* Уметь создавать программы с использованием языка Python;
* Уметь отлаживать созданные программы;
* Понимать основные принципы объектно-ориентированного и функционального программирования;
* Уметь создавать графический интерфейс для реализуемых программ;
* Иметь представление о событиях, происходящих с графическим интерфейсом;
* Знать, как организовать сетевое взаимодействие при помощи языка Python;
* Уметь взаимодействовать с базами данных;

|  |  |  |
| --- | --- | --- |
| **Тематика аудиторных занятый:** | | |
| **№** | **Тема** | **Кол-во часов** |
| **Основы Python** | | **36** |
| 1 | Введение в Python | 2 |
| 2 | История Python. Основы синтаксиса | 2 |
| 3 | Переменные, типы данных, операторы | 4 |
| 4 | Основы программирования на Python | 2 |
| 5 | Управляющие конструкции: условные операторы и циклы | 2 |
| 6 | Функции и модули | 4 |
| 7 | Продвинутые темы Python | 2 |
| 8 | Коллекции: списки, словари, множества, кортежи | 2 |
| 9 | Исключения и обработка ошибок | 4 |
| 10 | Объектно-ориентированное программирование | 4 |
| 11 | Классы и объекты. Наследование и композиция | 4 |
| 12 | Магические методы и итераторы | 4 |
| **Веб-разработка на Python** | | **44** |
| 1 | Введение в веб-разработку | 3 |
| 2 | Основы HTTP. Введение в WSGI | 3 |
| 3 | Введение во Flask | 3 |
| 4 | Flask | 3 |
| 5 | Маршрутизация и шаблоны в Flask | 4 |
| 6 | Работа с формами и базами данных в Flask | 4 |
| 7 | Django - часть 1 | 4 |
| 8 | Введение в Django. Основы моделей | 4 |
| 9 | Админка, миграции и работа с данными | 4 |
| 10 | Django - часть 2 | 4 |
| 11 | Шаблоны и формы в Django | 4 |
| 12 | Аутентификация и авторизация. Django REST Framework | 4 |
| **Разработка REST API и тестирование** | | **36** |
| 1 | Создание REST API | 3 |
| 2 | Введение в REST. Разработка API с использованием Flask | 3 |
| 3 | Разработка API с использованием Django REST Framework | 3 |
| 4 | Тестирование в Python | 3 |
| 5 | Основы тестирования. Unit tests | 3 |
| 6 | Тестирование Flask и Django приложений | 3 |
| 7 | Безопасность веб-приложений | 3 |
| 8 | Основы безопасности веб-приложений | 3 |
| 9 | Защита от распространенных уязвимостей | 3 |
| 10 | Работа с базами данных | 3 |
| 11 | SQL и SQLAlchemy | 3 |
| 12 | Работа с NoSQL базами данных (например, MongoDB) | 3 |
| **Микросервисы, контейнеризация и развертывание** | | **52** |
| 1 | Микросервисы | 4 |
| 2 | Введение в микросервисную архитектуру | 4 |
| 3 | Разработка и коммуникация микросервисов с использованием Flask и RabbitMQ/Kafka | 4 |
| 4 | Docker и контейнеризация | 4 |
| 5 | Введение в Docker. Основы контейнеризации | 4 |
| 6 | CI/CD и развертывание приложений | 4 |
| 7 | Введение в CI/CD. GitHub Actions для Python проектов | 4 |
| 8 | Развертывание приложений на примере Heroku и AWS | 6 |
| 9 | Завершение курса и финальный проект | 6 |
| 10 | Планирование финального проекта | 6 |
| 11 | Разработка финального проекта | 6 |
| Итого: | | **168** |

****

**Программа курса «Язык программирования JAVA»**

**Продолжительность курса – 168 часов.**

**Продолжительность одного занятия – 2 академических часа**

**Общее количество занятий – 42.**

**О курсе:**

Для создания программы курса по разработке на Java с использованием Spring Boot на 4 месяца, предполагая общую продолжительность курса в 168 часов и количество занятий – 42 (каждое занятие по 2 академических часа), можно разработать следующий план. Курс будет включать изучение Java, основ работы с Spring Framework, разработку веб-приложений на Spring Boot, а также работу с базами данных и безопасностью приложений.

**Цели курса**

По завершению курса учащиеся смогут:

* Понимать основы языка программирования Java и его применение;
* Разрабатывать веб-приложения с использованием Spring Boot;
* Работать с базами данных через Spring Data JPA;
* Интегрировать системы безопасности в приложения с помощью Spring Security;
* Создавать и управлять RESTful веб-сервисами;
* Развертывать веб-приложения в облачной среде.
* Предварительные требования
* Знание основ программирования;
* Основные знания в области работы с базами данных и SQL;
* Понимание принципов объектно-ориентированного программирования.

|  |  |  |
| --- | --- | --- |
| **Тематика аудиторных занятый:** | | |
| **№** | **Тема** | **Кол-во часов** |
| **Основы Java** | | **36** |
| 1 | Введение в Java | 2 |
| 2 | История Java. Основные принципы и синтаксис | 2 |
| 3 | Переменные, типы данных, операторы. | 4 |
| 4 | ООП в Java | 2 |
| 5 | Классы и объекты | 2 |
| 6 | Наследование, интерфейсы, абстрактные классы | 4 |
| 7 | Коллекции и исключения | 2 |
| 8 | Работа с коллекциями | 2 |
| 9 | Исключения и обработка ошибок | 4 |
| 10 | Многопоточность и I/O | 4 |
| 11 | Основы многопоточности | 4 |
| 12 | Работа с файлами, потоками ввода/вывода | 4 |
| **Введение в Spring и Spring Boot** | | **44** |
| 1 | Основы Spring Framework | 3 |
| 2 | Inversion of Control (IoC) и Dependency Injection (DI) | 3 |
| 3 | Конфигурация Spring приложений | 3 |
| 4 | Spring Data | 3 |
| 5 | Работа с данными в Spring | 4 |
| 6 | Введение в Spring Boot | 4 |
| 7 | Spring MVC | 4 |
| 8 | Создание первого Spring Boot приложения | 4 |
| 9 | Основы Spring MVC | 4 |
| 10 | Глубже в Spring Boot | 4 |
| 11 | Продвинутая конфигурация Spring Boot | 4 |
| 12 | Создание REST API с Spring Boot | 4 |
| **Работа с данными и безопасность** | | **36** |
| 1 | Spring Data JPA | 3 |
| 2 | Введение в Spring Data JPA | 3 |
| 3 | Продвинутая работа с базами данных | 3 |
| 4 | Spring Security | 3 |
| 5 | Основы Spring Security | 3 |
| 6 | Аутентификация и авторизация | 3 |
| 7 | Создание RESTful веб-сервисов | 3 |
| 8 | Документирование API с Swagger | 3 |
| 9 | Работа над проектом | 4 |
| 10 | Планирование проекта | 4 |
| 11 | Работа над проектом | 4 |
| **Тестирование, микросервисы и развертывание** | | **52** |
| 1 | Тестирование в Spring | 4 |
| 2 | Unit-тестирование | 6 |
| 3 | Интеграционное тестирование | 6 |
| 4 | Микросервисы с Spring Cloud | 6 |
| 5 | Введение в микросервисную архитектуру | 8 |
| 6 | Основы Spring Cloud | 6 |
| 7 | Развертывание приложений | 8 |
| 8 | Контей | 8 |
| Итого: | | **168** |