

Мидл Java-разработчик

6 месяцев

продолжительность курса

3 проекта

в портфолио



Бесплатный вводный модуль

1 час

Модуль 1. Разработка приложения при помощи Spring Framework и Spring Boot

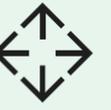
Погружение в современную Java	Спринт 1	29 часов
Погружение в современные средства разработки и паттерны	Спринт 2	30 часов
Spring Framework	Спринт 3	30 часов
Spring Boot	Спринт 4	30 часов
	Каникулы	1 неделя

Модуль 2. Разработка классического и реактивного приложения при помощи Spring Data и Spring Web

Разработка классического приложения при помощи Spring Data и Spring Web: Servlet Stack	Спринт 5	30 часов
Разработка реактивного приложения при помощи Spring Data и Spring Web: Reactive Stack	Спринт 6	30 часов
Создание RESTful сервисов с использованием Spring Rest, OpenAPI и Redis	Спринт 7	30 часов
Spring Security	Спринт 8	30 часов
	Каникулы	1 неделя

Модуль 3. Разработка микросервисного приложения

Разработка приложений с использованием микросервисной архитектуры	Спринт 9	30 часов
Развертывание приложения с использованием CI/CD, Kubernetes, Helm	Спринт 10	30 часов
Apache Kafka	Спринт 11	30 часов
Мониторинги и логи с помощью ELK-стека	Спринт 12	30 часов
Завершающий модуль		1 час



[1 час]

Узнаете детали о строении программы: почему важно изучать не только паттерны, но и инструменты, зачем нужно учиться на реальных кейсах, как устроена практика. Также познакомитесь с форматом обучения, как будут проходить спринты, и кто будет сопровождать вас на пути. Пройдёте входной тест. Он будет полезен и вам, и нам. Вы — сможете убедиться в том, что курс будет оптимален по сложности. Мы — будем уверены, что наши студенты обладают достаточными навыками для прохождения курса. Всё это поможет понять, подходит ли вам курс.

Темы

1. Подробнее о курсе

2. Входной тест

Разработка приложения при помощи Spring Framework и Spring Boot

01

[8 недель],
[1 проект]

- Освоите главные нововведения в Java за последние несколько лет, которые появились в LTS версии Java 21
- Изучите функциональный подход к программированию, поработаете со Stream API, узнаете про легковесные потоки, современные возможности создания многопоточных приложений, а также уменьшение boilerplate-кода за счёт Records и Pattern Matching
- Научитесь тестировать свои приложения с помощью фреймворков JUnit 5 и Mockito
- Изучите стандарт при написании корпоративных приложений — фреймворка Spring Framework, его логичного развития — Spring Boot, а также возможностей их тестирования через TestContext Framework, SpringBoot Test и кэширование контекстов

Спринт 1. Погружение в современную Java

- | | | |
|--|---------------------------------------|--------------------------------------|
| 1. Java 8 как отправная точка | 4. Современная многопоточность в Java | 7. Assertions JUnit 5 и Mock-объекты |
| 2. На пути к Java 21 - изменения в языке | 5. Тестирование на Java | 8. Типы тестов JUnit 5 |
| 3. На пути к Java 21 - изменения в API | 6. Архитектура JUnit 5 | 9. Миграция с JUnit 4 |

Практика

Перепишете тесты с JUnit 4 на JUnit 5



Спринт 2. Погружение в современные средства разработки и паттерны

- | | | |
|---|---|---|
| 1. Принципы разработки YAGNI, DRY, KISS, DRY и бритва Оккама | 4. Базовые паттерны разработки корпоративных приложений | 7. Продвинутая работа с коммитами |
| 2. Принципы проектирования и организации кода SOLID | 5. Система контроля версий Git и парадигмы разработки | 8. Системы сборки Maven и Gradle |
| 3. Архитектурные паттерны разработки корпоративных приложений | 6. Продвинутая работа с удаленными репозиториями, тэгами и псевдонимы | 9. Управление зависимостями |
| | | 10. Использование плагинов |
| | | 11. Жизненный цикл сборки проекта и его профили |

Практика

Напишите свой собственный плагин для Gradle



Спринт 3. Spring Framework

- | | | |
|---|---|--|
| 1. Spring как IoC-контейнер | 5. Основные возможности Spring | 8. Тестирование Spring-приложений |
| 2. Инфраструктурные бины Spring | 6. Аспектно-ориентированное программирование в Spring | 9. TestContext Framework |
| 3. Создание бинов через Java-аннотации | 7. Практика по разработке на Spring Framework | 10. Практика по тестированию Spring-приложений |
| 4. Создание бинов через Java-конфигурации | | |

Практика

Создадите собственное MVC-приложение блог на Spring Framework и протестируете Web и DAO слои приложения



Спринт 4. Spring Boot

1. SpringBoot как развитие Spring Framework

2. Сборка SpringBoot-приложений в Maven

3. Сборка SpringBoot-приложений в Gradle

4. Запуск SpringBoot-приложения

5. Настройка приложения

6. Автоконфигурации и стартеры

7. Мониторинг приложения с использованием SpringBoot Actuator

8. Практика по разработке SpringBoot-приложения

9. SpringBootTest для тестирования SpringBoot-приложений

10. Практика по тестированию SpringBoot-приложения

Практика

Перепишите ранее созданное приложение блог с использованием SpringBoot и протестируете Web и DAO слои приложения



Разработка классического и реактивного приложения при помощи Spring Data и Spring Web

02

[8 недель],
[1 проект]

- Узнаете о возможностях написания классических web-приложений в парадигме Model-View-Controller с походами за данными в реляционную базу данных с использованием связки для объектно-реляционного маппинга Spring Data + Hibernate
- Изучите достоинства и недостатки реактивного подхода в написании web-ориентированных приложений, попробуйте Reactive Stack и R2DBS для доступа в базу данных, а также научитесь писать тесты с использованием TestContainers и Spring MVC Test
- Изучите возможности написания RESTful сервисов и использования OpenAPI-спецификации для генерации RestController, а также обеспечение аутентификации и авторизации пользователей с использованием фреймворка Spring Security

Спринт 5. Разработка классического приложения при помощи Spring Data и Spring Web: Servlet Stack

1. Доступ к данным в Spring Data JDBC

2. Доступ к данным в Spring Data JPA

3. Миграция данных с помощью Liquibase

4. Тестирование Spring Data

5. Фреймворк WebMVC в Spring

6. Контроллеры в WebMVC

Практика

Создадите витрину магазина на Spring Boot и протестируете его при помощи Spring Data JPA



Спринт 6. Разработка реактивного приложения при помощи Spring Data и Spring Web: Reactive Stack

1. Реактивное программирование на Java

2. Проекты для создания реактивных приложений

3. Реактивный доступ к данным в Spring Data R2DBC

4. Концепция сущностей для работы с данными

5. Репозитории в Spring Data R2DBC

6. Тестирование Spring Data R2DBC

7. Фреймворк WebFlux в Spring

8. Обработка запросов в функциональном стиле в WebFlux

9. Конфигурирование и тестирование WebFlux

Практика

Создадите витрину магазина на реактивном движке Spring Boot и протестируете его при помощи Spring Data R2DBC и Spring WebFlux



Спринт 7. Создание RESTful сервисов с использованием Spring Rest, OpenAPI и Redis

1. Построение RESTful сервисов

2. Поддержка RESTful сервисов в Spring

3. Клиенты для выполнения REST-запросов

4. Спецификация OpenAPI для описания RESTful сервисов

5. Использование Spring Data Redis в качестве кэша

6. Создание RESTful сервиса с использованием OpenAPI и Redis

Практика

Доработаете витрину магазина на реактивном движке Spring Boot - добавьте RESTful сервис с использованием OpenAPI и Redis



Спринт 8. Spring Security

1. Spring Security на стеке сервлетов

2. Аутентификация на стеке сервлетов

3. Хранение данных аутентификации, X509 и разлогинивание

4. Авторизация на стеке сервлетов

5. Авторизация OAuth 2.0

6. Интеграция Spring Security с Spring MVC на стеке сервлетов

7. Аутентификация и авторизации в Spring Security на реактивном стеке

8. Авторизация OAuth 2.0 на реактивном стеке

9. Интеграция Spring Security с Spring WebFlux на реактивном стеке

Практика

Доработаете витрину магазина на реактивном движке Spring Boot - добавьте Spring Security



Разработка микросервисного приложения

03

[8 недель],
[1 проект]

- Узнаете об основных плюсах использования микросервисного подхода к написанию приложений: Saga, Circuit Breaker, API Gateway и Service Discovery
- Изучите основные паттерны и поддержку со стороны фреймворка Spring Cloud в лице Modulith, Consul и Zookeeper
- Научитесь обеспечивать безопасность при обмене данными между микросервисами
- Сможете развёртывать приложения с использованием CI/CD, популярного в настоящее время Kubernetes и Helm
- Узнаете о возможностях стриминговой платформы Apache Kafka и её использовании, чтобы доставлять данные в ELK-стек для обеспечения мониторингов и поставки логов

Спринт 9. Разработка приложений с использованием микросервисной архитектуры

1. Паттерны микросервисной архитектуры

2. Паттерны межсервисного взаимодействия

3. Паттерны обеспечения безопасности, состояния, тестирования и UI сервисов

4. Стек технологий Spring Cloud

5. Безопасность и тестирование в Spring Cloud

6. Интеграции в Spring Cloud

Практика

Создайте микросервисное приложение - инфраструктуру банка с использованием микросервисной архитектуры



Спринт 10. Развертывание приложения с использованием CI/CD, Kubernetes, Helm

1. Оркестрация контейнеризованных приложений

2. Внутреннее устройство Kubernetes

3. Возможности Kubernetes

4. Пакетный менеджер Helm

5. Тестирование, обеспечение безопасности и лучшие практики в Helm

6. Непрерывная интеграция и доставка микросервисов

7. Возможности Jenkins

8. Интеграции Jenkins

Практика

Продолжите работу над микросервисным приложением банка - настройте его развертывание при помощи CI/CD, Kubernetes, Helm



Спринт 11. Apache Kafka

1. Стриминговая платформа Apache Kafka

2. Возможности Apache Kafka

4. Интеграция Apache Kafka со Spring

5. Возможности Spring Kafka

7. Неблокирующие ретраи обработки сообщений в Spring Kafka

8. Поддержка потоковой обработки данных и тестирование в Spring Kafka

Практика

Продолжите работу над микросервисным приложением банка - настройте взаимодействие микросервисов при помощи Apache Kafka



Спринт 12. Мониторинги и логи с помощью ELK-стека

1. Трейсинг с помощью Spring Cloud Sleuth и Zipkin

2. Мониторинги и алерты с помощью Prometheus и Grafana

3. Сбор логов с использованием Apache Kafka и Logstash

4. Использование Elasticsearch для индексации, хранения и поиска логов

5. Использование системы визуализации данных Kibana

Практика

Завершите работу над микросервисным приложением банка - добавьте логирование и мониторинг при помощи ELK-стека

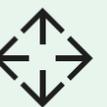


Воркшопы



Опытные наставники проведут воркшопы по разбору сложных тем и сессии Q&A

Завершающий модуль



[1 час]

Подведёте итоги обучения