

Java Advanced I: функциональное, асинхронное и реактивное программирование

Код: JVA-074

Длительность: 36 ч.

Описание:

Курс посвящен современным функциональным и реактивным подходам к разработке на Java, повышающим производительность бекэнда, и включает изучение неблокирующего ввода-вывода NIO2, асинхронных промисов `CompletableFuture`, библиотек реактивного программирования RxJava и Reactor, интерфейса асинхронного доступа к БД R2DBC, потоковый протокол SSE, реактивные репозитории Spring Data reactive, WebClient, протоколы WebSocket и RSocket, а также практические примеры применения.

Цели:

- Познакомить слушателей с современными асинхронным и реактивным программированием.
- Дать примеры применения.

Разбираемые темы:

Модуль 1. Функциональная Java (3 часа)

- Функциональные интерфейсы
- Ссылки на метод
- Потоки данных
- Map / filter / reduce
- Агрегационные функции
- Reduce при параллельной и последовательной обработке
- Цепочки потребителей
- Потоковые коллекторы

Домашняя работа. Практика (1 час).

Модуль 2. Executor framework. Fork-Join pool (3 часа)

- Использование Executors
- Future интерфейс
- Использование интерфейса Callable
- Отмена задач
- ForkJoin Framework

- Создание пула задач ForkJoin
- Параллельные потоки в Java 8+
- ForkJoin vs. параллельные потоки vs. последовательные потоки

Модуль 3. Неблокирующий ввод-вывод NIO (5 часов)

- Основные различия между Java NIO и IO
- Поточно-ориентированный или буферно-ориентированный
- Блокирующий против неблокирующего ввода-вывода
- Java NIO Buffer
- Java NIO Channel
- Direct буфер
- Mapped file буфер
- Java NIO Selector
- Асинхронный ввод / вывод с NIO
- Блокировка файла
- Сервер NIO
- Архитектура сервера Netty

Домашняя работа. Практика (2 часа).

Модуль 4. Асинхронная Java (Completable Future) (5 часов)

- Проблемы синхронного кода
- NodeJS архитектура
- Параллельная обработка потока данных
- CompletableFuture: sync и async методы
- Объединение (combine), составление (composing) и быстрееший побеждает (quicker wins)
- Обработка исключений в CompletableFuture
- Отмена CompletableFuture
- Использование CompletableFuture для реального потока данных

Домашняя работа. Практика (2 часа).

Модуль 5. Реактивное программирование (8 часов)

- Что такое реактивность?
- Основы RxJava: наблюдатель и наблюдаемый (Observable и Observer)
- Реактивные потоки в Java 9 (Java Reactive Streams)
- Методы для побочных эффектов (side effects)
- Обработка ошибок
- Горячие и холодные потоки (hot/cold)
- Разделяемые и подключаемые потоки (shared/connectable)
- Освобождение потоков (disposing)

- Реактивные операторы (полный каталог)
- Subjects
- Распараллеливание потоков (Schedulers)
- Противодействие (backpressure) и интерфейс Flowable
- Тестирование реактивных потоков

Модуль 6. Проект Reactor (4 часа)

- Операторы Flux / Mono в Reactor
- Работа с backpressure в Reactor
- Reactor и многопоточность (Reactor Schedulers)
- Parallel Flux потоки
- Оборачивание синхронных вызовов
- Тестирование Reactor
- Процессоры (Reactor processors)

Модуль 7. Spring WebFlux и практическое реактивное программирование (8 часов)

- Реактивный доступ к БД
- Реактивные драйверы R2DBC
- Spring Data - реактивные репозитории
- Поддержка страниц (paging) в Spring Data reactive
- WebFlux: функциональные контроллеры
- Spring REST контроллеры, возвращающие реактивные данные (Mono / Flux)
- Spring REST контроллеры, возвращающие SSE (server-sent event)
- WebClient: получение реактивных данных с сервера
- SSE и протоколы WebSocket
- Использование WebSocket для передачи / получения данных JSON
- Использование WebSocket для передачи / извлечения двоичных данных
- Протокол RSocket
- Контроллер RSocket на стороне сервера
- RSocket клиент: обмен JSON и двоичными данными
- RSocket с балансировкой нагрузки
- RSocket с взвешенной балансировкой нагрузки
- Использование реактивных потоков с брокером сообщений (RabbitMQ)
- Spring Data MongoDB реактивные репозитории
- Бенчмарки: R2DBC против JDBC и WebFlux против Web MVC
- Архитектура и шаблоны реактивного программирования

Домашняя работа. Практика (2 часа).

Целевая аудитория:

Разработчики Java, архитекторы Java

Предварительная подготовка - общее:

Уверенное владение Java, желательно знание Spring.

Рекомендуемые дополнительные материалы, источники:

1. Core Java SE 9 for the Impatient, Second Edition by Cay Horstmann - sections Streams, Concurrent programming
2. Java I/O, NIO and NIO.2 by Jeff Friesen
3. Learning RxJava - Second Edition by Nick Samoylov and Thomas Nield
4. Project Reactor reference <https://projectreactor.io/docs/core/release/reference/>
5. Spring WebFlux reference <https://docs.spring.io/spring/docs/current/spring-framework-reference/web-reactive.html>
6. Hands-On Reactive Programming in Spring 5: Build Cloud-ready, Reactive Systems with Spring 5 and Project Reactor, by Oleh Dokuka(Author), Igor Lozynskyi (Author)