Java-разработчик

Специализация

👉 Научитесь профессионально создавать приложения на самом популярном в корпоративной среде языке программирования.

👉 Подготовите качественное портфолио в виде примера работ на GitHub.

👉 Создавать приложения для многопоточной распределённой обработки данных

Продолжительность

- 12 месяцев
- 4 академических часа в неделю
- 310 академическихчасов всего

Оглавление

Минимальные знания

Минимальные знания

Обязательно:

- понимание принципов ООП;
- опыт программирования на любом объектно-ориентированном языке программирования.

•Будет плюсом:

- знакомство с языком Java;
- знание шаблонов проектирования;
- умение писать чистый код;
- навыки использования git.

Курс 1 // Подготовительный курс по Java разработке

Язык Java

Модуль №1

Формат:

- 26 академ часов видеозаписей
- Тесты для проверки знаний
- 2 онлайн-консультации в формате вебинаров

Цель: учащийся должен уметь создавать новый проект, управлять потоком исполнения, работать с исключениями, разбираться в особенностях работы с примитивными типами.

Базовые синтаксические конструкции и операторы в Java

Учащийся изучит понятия:

- Типы данных, базовые типы данных
- Преобразование типов
- Битовые операторы, битовые маски
- Логические и математические операторы, приоритеты
- Операторы управления логикой работы приложения
- Циклы
- Структура консольного Java-приложения

Тест к занятию 1

Ссылочные типы данных и обработка ошибок

Учащийся изучит понятия:

- IDE и ее основные возможности
- Класс и объект, создание объекта
- Поля и методы класса
- Области видимости
- Передача по ссылке
- Особенности == и equals
- Исключительная ситуация, Stack trace ошибки

на уровне, достаточном для выполнения заданий данного раздела. **Тест к занятию 2**

Работа с массивами и строками
Учащийся изучит возможности базовых классов
• class Object
• == и equals()
• массивы
• String
• StringBuilder
• перегрузка методов
• java.util.Arrays

на уровне, достаточном для выполнения заданий данного раздела.

Тест к занятию 3

Проектирование и тестирование приложений **Модуль №2**

Цель: учащийся должен понимать UML диаграммы, разбираться в структуре классов приложений, понимать и уметь применять на языке Java основные концепции ООП, писать юнит-тесты.

Занятие 1. Объектно-ориентированное программирование в Java
Учащийся изучит понятия:
• Ссылки между объектами
• Инкапсуляция, модификаторы доступа
• Наследование, extends
• Абстракция, интерфейс, implements
• Ключевые слова this и super
• Generics
на уровне, достаточном для выполнения заданий данного раздела.

Тест к занятию 1

Занятие 2. Концепции объектно-ориентированного проектирования

Учащийся изучит понятия:

• Проблема проектирования

• Концепции ООП: наследование, полиморфизм, инкапсуляция, абстракция

• Нотация UML для обозначения иерархий классов и их взаимоотношений

• Базовые паттерны

на уровне, достаточном для выполнения заданий данного раздела.

Тест к занятию 2

Занятие 3. Unit-тестирование

Учащийся изучит понятия:

• Концепция Unit-тестирования

• Подключение библиотек

• Git

• Система сборки Maven

• Использования библиотеки Junit

на уровне, достаточном для выполнения заданий данного раздела.

Платформа Java

Модуль №3

Цель: учащийся должен понимать почему Java это не только язык но и платформа для разработки приложений, уметь собирать приложения без среды разработки, понимать жизненный цикл приложения и параметры, которые на него влияют.

Занятие 1. Устройство платформы Java

Учащийся изучит понятия:

- История Java
- Кроссплатформенность
- JVM
- JRE, JDK, Interpreter, JIT
- Vm start parameters

на уровне, достаточном для выполнения заданий данного раздела.

Тест к занятию 1

Занятие 2. Сборка и упаковка Java-приложения

Учащийся изучит понятия:

- Компилятор javac,
- .class файл, Bytecode
- Запуск приложения из консоли
- Подключение библиотек
- Утилита jar и .jar файл
- Манифест
- Ресурсы
- Maven как система сборки

Тест к занятию 2

• Класс Collections

на уровне, достаточном для выполнения заданий данного раздела.

Тест к занятию 1

Занятие 2. Исключения. Дата и время. Генерация случайных чисел.

Учащийся изучит понятия:

• Throwable

• Обработка исключений

• AutoCloseable

• Date and Time

• Random

на уровне, достаточном для выполнения заданий данного раздела.

Тест к занятию 2

Стандартная библиотека

Модуль №4

Цель: учащийся должен разбираться в основных контейнерах и классах, которые содержит стандартная библиотека, понимать когда и какие контейнеры нужно использовать, уметь работать со временем, случайными числами, читать и писать данные в файл.

Занятие 1. Классы-контейнеры

Учащийся изучит понятия:

- Интерфейсы List, Set, Map, Queue
- Реализации Collection
- Реализации Map
- Класс Collections

на уровне, достаточном для выполнения заданий данного раздела.

Тест к занятию 1

Занятие 2. Исключения. Дата и время. Генерация случайных чисел.

Учащийся изучит понятия:

- Throwable
- Обработка исключений
- AutoCloseable
- Date and Time
- Random

на уровне, достаточном для выполнения заданий данного раздела.

Тест к занятию 2

Занятие 3. Работа с файлами

Учащийся изучит понятия:

- IO and Streams
- Reader, Writer
- Запись в файл и чтение из файла

на уровне, достаточном для выполнения заданий данного раздела.

Тест к занятию 3

Консультация

Выпускной проект

На протяжении всего курса вы будете работать над приложением по поиску фильмов. Приложение будет включать в себя возможность поиска фильма или сериала, сортировки по контенту или ключевому слову. Будет содержать экраны с детальным описанием фильма, историей и любимыми фильмами. Также в приложении будет возможность поделиться с друзьями фильмом и мнением о нем, а также настройка уведомлений о выходе нового фильма или эпизода любимого сериала.

Последние 2 недели курса посвящены доработке проекта под руководством преподавателя, что позволит вам получить качественное портфолио.

Курс 2 // Разработчик Java

Язык и платформа Java

Модуль №1

Тема 1: Подготовка к курсу. ДЗ

Познакомиться с программой курса,
изучить основные инструменты

Домашние задания

Проект maven с модульной структурой

1. Создать аккаунт на github.com (если еще нет)
2. Создать репозиторий для домашних работ
3. Сделать checkout репозитория на свой компьютер
4. Создайте локальный бранч hw01-maven
5. Создать проект maven
6. В проект добавьте последнюю версию зависимости
   <groupId>com.google.guava</groupId>
   <artifactId>guava</artifactId>
7. Создайте модуль hw01-maven
8. В модуле сделайте класс HelloOtus
9. В этом классе сделайте вызов какого-нибудь метода из guava
10. Добавьте нужный плагин maven и соберите "толстый-jar"
11. Убедитесь, что "толстый-jar" запускается.
12. Сделайте pull-request в gitHub
13. Ссылку на PR отправьте на проверку.

Тема 2: Дополнение к maven, история изменения языка

Познакомиться со Shade Plugin Углубить знания о maven Познакомиться с текущей ситуацией в мире java

Тема 3: Контейнеры и алгоритмы. ДЗ

Познакомиться с Generic-ами в Java и со стандартными коллекциями

Домашние задания

DIY ArrayList Написать свою реализацию ArrayList на основе массива. class DIYarrayList implements List{...} Проверить, что на ней работают методы из java.util.Collections:
Collections.addAll(Collection c, T... elements)
Collections.static void copy(List dest, List src)
Collections.static void sort(List list, Comparator c)

1) Проверяйте на коллекциях с 20 и больше элементами.
2) DIYarrayList должен имплементировать ТОЛЬКО ОДИН интерфейс - List.
3) Если метод не имплементирован, то он должен выбрасывать исключение UnsupportedOperationException.

Тема 4: Инструменты для преобразования контейнеров, unsafe, jmh

На примере изучить принципы создания коллекций. Познакомиться с пакетом unsafe, утилитой JMH и популярными библиотеками коллекций.

Тема 5: QA и тестирование

Познакомиться с junit и mockito На примере понять, что такое «тестируемое приложение»

Тема 6: Аннотации. ДЗ

Познакомиться с механизмом Reflection. Узнать что такое Аннотации и как их можно сделать

Домашние задания

Свой тестовый фреймворк. Написать свой тестовый фреймворк. Поддержать свои аннотации @Test, @Before, @After. Запускать вызовом статического метода с именем класса с тестами.

Т.е. надо сделать:

1) создать три аннотации - @Test, @Before, @After.
2) Создать класс-тест, в котором будут методы, отмеченные аннотациями.
3) Создать "запускалку теста". На вход она должна получать имя класса с тестами, в котором следует найти и запустить методы отмеченные аннотациями и пункта 1.
4) Алгоритм запуска должен быть следующий:: метод(ы) Before текущий метод Test метод(ы) After для каждой такой "тройки" надо создать СВОЙ объект класса-теста.
5) Исключение в одном тесте не должно прерывать весь процесс тестирования.
6) На основании возникших во время тестирования исключений вывести статистику выполнения тестов (сколько прошло успешно, сколько упало, сколько было всего)

Тема 7: Углубленные основы (примитивные типы, Remote debug, Hot swap).

Узнать детали устройства типов данных в Java. Познакомиться с механизмами Remote Debug и Hot swap. Знакомство с утилитой Jol

Тема 8: Байт код, classloader, инструменатция, asm. ДЗ

Познакомиться с принципами работы виртуальной машины Java, ClassLoader-ами и байт-кодом

Домашние задания

Автомагическое логирование. Разработайте такой функционал:
метод класса можно пометить самодельной аннотацией @Log, например, так:

class TestLogging { @Log public void calculation(int param) {}; }

При вызове этого метода "автомагически" в консоль должны логироваться значения параметров. Например так.

class Demo { public void action() { new TestLogging().calculation(6); } }

В консоле дожно быть: executed method: calculation, param: 6

Обратите внимание: явного вызова логирования быть не должно.

Тема 9: Сборщик мусора. ДЗ

Знакомство со сборщиком мусора в Java

Домашние задания

Сравнение разных сборщиков мусора Написать приложение, которое следит за сборками мусора и пишет в лог количество сборок каждого типа (young, old) и время которое ушло на сборки в минуту.

Добиться OutOfMemory в этом приложении через медленное подтекание по памяти (например добавлять элементы в List и удалять только половину).

Настроить приложение (можно добавлять Thread.sleep(...)) так чтобы оно падало с OOM примерно через 5 минут после начала работы.

Собрать статистику (количество сборок, время на сборки) по разным GC.

!!! Сделать выводы !!! ЭТО САМАЯ ВАЖНАЯ ЧАСТЬ РАБОТЫ: Какой gc лучше и почему?

Выводы оформить в файле Conclusions.md в корне папки проекта. Результаты измерений сведите в таблицу.

Тема 10: Java 8

Введение в функциональное программироавние (ФП). Знакомство с возможностями ФП, которые появились в Java 8.

Проектирование

Модуль №2

Тема 1: Концепты проектирования ООП. ДЗ

Изучить принципы SOLID и общие критерии идеальной архитектуры

Домашние задания

Эмулятор банкомата

Написать эмулятор ATM (банкомата).

Объект класса ATM должен уметь:
- принимать банкноты разных номиналов (на каждый номинал должна быть своя ячейка)
- выдавать запрошенную сумму минимальным количеством банкнот или ошибку если сумму нельзя выдать

Это задание не на алгоритмы, а на проектирование. Поэтому оптимизировать выдачу не надо.
- выдавать сумму остатка денежных средств

Тема 2: Behavioral patterns

Изучить поведенческие паттерны проектирования

Тема 3: Structural patterns. ДЗ

Изучить структурные паттерны проектирования

Домашние задания

Департамент ATM

Написать приложение ATM Департамент:

1) Департамент может содержать несколько ATM.
2) Департамент может собирать сумму остатков со всех ATM.
3) Департамент может инициировать событие – восстановить состояние всех ATM до начального (начальные состояния у разных ATM могут быть разными).

Это тренировочное задание на применение паттернов. Попробуйте использовать как можно больше.

Тема 4: Creational patterns

Изучить "создающие" паттерны проектирования

Работа с окружением

Модуль №3

Тема 1: Сериализация. ДЗ

Познакомиться с функционалом сериализации объектов

Домашние задания

Свой json object writer
Напишите свой json object writer (object to JSON string) аналогичный gson на основе javax.json.

Поддержите:
- массивы объектов и примитивных типов
- коллекции из стандартный библиотеки.

Тема 2: NIO. Логирование

познакомиться с методами логирования в Java.
познакомиться с NIO

Тема 3: JDBC. ДЗ

Познакомиться с транзакцией в реляционной СУБД и jdbc

Домашние задания

Самодельный ORM
Работа должна использовать базу данных H2. Создайте в базе таблицу User с полями:
- id bigint(20) NOT NULL auto_increment
- name varchar(255)
- age int(3)

Создайте свою аннотацию @Id
Создайте класс User (с полями, которые соответствуют таблице, поле id отметьте аннотацией).
Напишите JdbcTemplate, который умеет работать с классами, в котрых есть поле с аннотацией @Id.
Executor должен сохранять объект в базу и читать объект из базы. Имя таблицы должно соответствовать имени класса, а поля класса - это колонки в таблице.

Методы JdbcTemplate'a:
- void create(T objectData);
- void update(T objectData);
- void createOrUpdate(T objectData); // опционально.
- T load(long id, Class clazz);

Проверьте его работу на классе User.

Метод createOrUpdate - необязательный. Он должен "проверять" наличие объекта в таблице и создавать новый или обновлять.

Создайте еще одну таблицу Account:
- no bigint(20) NOT NULL auto_increment
- type varchar(255)
- rest number

Создайте для этой таблицы класс Account и проверьте работу JdbcTemplate на этом классе.

Тема 4: Общие вопросы работы с СУБД, myBatis

Рассмотреть CAP-теорему Рассмотреть методы организации блокировок Познакомиться с MyBatis

Тема 5: Общие вопросы работы с СУБД, myBatis

Закомство с Hibernate

Домашние задания

Использование Hibernate Работа должна использовать базу данных H2.

Возьмите за основу предыдущее ДЗ (Самодельный ORM) и реализуйте функционал сохранения и чтения объекта User через Hibernate. (Рефлейсия больше не нужна) Конфигурация Hibernate должна быть вынесена в файл.

Добавьте в User поля:

адрес (OneToOne)
class AddressDataSet { private String street; }

и телефон (OneToMany) class PhoneDataSet { private String number; }

Разметьте классы таким образом, чтобы при сохранении/чтении объека User каскадно сохранялись/читались вложенные объекты. Не забывайте про сохранение абстракций в приложении (см. комментарий в вебинаре).

Тема 6: JPQL

Познакомиться с Connection Pool Узнать методы конструирования запросов в Hibernate

Тема 7: Типы ссылок. Кэширование. ДЗ

Узнать какие в java есть виды ссылок и для чего они нужны Понять как устроены кэши Познакомиться с "промышленным" кэшом Ehcache

Домашние задания

Свой cache engine Напишите свой cache engine с soft references. Добавьте кэширование в DBService из задания про Hibernate ORM

Тема 8: No SQL

Познакомиться с noSQL базами данных Понять отличия SQL от noSQL, когда и что следует использовать. Познакомится с MongoDB

Тема 9: Web сервер. ДЗ

На примере Jetty понять принципы работы Webсервера и servlet API

Домашние задания

Веб сервер Встроить веб сервер в приложение из ДЗ про Hibernate ORM. Сделать админскую страницу, на которой админ должен авторизоваться.

На странице должны быть доступны следующие функции:
- создать пользователя
- получить список пользователей

P.S. при желании вместо Hibernate и H2 можно использовать MongoDB

Тема 10: Dependency injection. ДЗ

Изучить принципы работы контейнера TomCat Изучить принципы работы framework Spring

Домашние задания

Приложение с IoC контейнером Собрать war для приложения из предыдущего ДЗ. Создавать кэш и DBService как Spring beans, передавать (inject) их в сервлеты. Запустить веб приложение во внешнем веб сервере.

Тема 11: Asynchronous Web applications

Узнать как можно сделать ассинхронный web-сервис на java. Познакомиться со Spring Boot

Многопоточность

Модуль №4

Тема 1: Thread

Познакомиться с основными принципами многопоточности Узнать как управлять потоками в Java

Тема 2: JMM. ДЗ

Познакомиться с основными проблемами многопоточности. Понять зачем придумали JMM Узнать основные положения JMM

Домашние задания

Последовательность чисел Два потока печатают числа от 1 до 10, потом от 10 до 1.

Надо сделать так, чтобы числа чередовались, т.е. получился такой вывод:
Поток 1:1 2 3 4 5 6 7 8 9 10 9 8 7 6 5 4 3 2 1 2 3 4....
Поток 2: 1 2 3 4 5 6 7 8 9 10 9 8 7 6 5 4 3 2 1 2 3....

Тема 3: Message System. ДЗ

Познакомиться с потокобезопасными контейнерами Познакомиться с паттерном - "система обмена сообщениями"

Домашние задания

MessageSystem Добавить систему обмена сообщениями в ДЗ про веб сервер с IoC контейнером. Пересылать сообщения из вебсокета в DBService и обратно. Организовать структуру пакетов без циклических зависимостей.

Тема 4: Executors

Познакомиться с пулами потоков в Java

Тема 5: Многопроцессные приложения. ДЗ

Изучение сетевого взаимодействия в java. Изучение принципов работы "клиент-серверного" приложения в Java

Домашние задания

MessageServer
Сервер из предыдущего ДЗ про MessageSystem разделить на три приложения:

- MessageServer
- Frontend
- DBServer

Запускать Frontend и DBServer из MessageServer. Сделать MessageServer сокет-сервером, Frontend и DBServer клиентами. Пересылать сообщения с Frontend на DBService через MessageServer.
Запустить приложение с двумя серверами фронтенд и двумя серверами баз данных на разных портах. Если у вас запуск веб приложения в контейнере, то MessageServer может копировать root.war в контейнеры при старте

Тема 6: NIO

Изучение основ сетевых возможностей NIO

Тема 7: Netty

Изучить основные принципы работы Netty.

Проектная работа

Тема 1: Консультация по ДЗ и проектам

получить ответы на вопросы по проекту

Домашние задания

Проектная работа Заключительный месяц курса посвящен проектной работе. Свой проект это то, что интересно писать студенту. То, что можно создать на основе знаний, полученных на курсе. При этом не обязательно закончить его за месяц. В процессе написания по проекту можно получить консультации преподавателей.

Проект должен стать примером кода, который можно показывать потенциальным работодателям.

Примеры тем проекта:
- web сервер (разберите протокол)
- socket сервер на NIO (как netty)
- свой ORM
- распределенный кэш
- кэш для hibernate

Тема 2: Консультация по ДЗ и проектам

Получить ответы на вопросы по проектной работе

Тема 3: Защита проектов

Защитить свой проект и получить рекомендации экспертов

Введение

Тема 1: Введение в Spring Framework

ориентироваться в проектах Spring для дальнейшего изучения, применять принцип IoC при написании классов и тестов, создавать контекст Spring, определять Spring Beans в контексте, организовывать правильный DI

Домашние задания

Программа по проведению тестирования студентов

Цель задания: научиться создавать приложение с помощью Spring IoC. Результат: простое приложение, сконфигурированное XML-контекстом. Специальная цена Разработчик на Spring Framework Курс о разработке веб-приложений на Spring, о фреймворках и вспомогательных технологиях Spring.

Описание задание: В ресурсах хранятся вопросы и различные ответы к ним в виде CSV файла (5 вопрсов). Программа должна спросить у пользователя фамилию и имя, спросить 5 вопросов из CSV-файла и вывести результат тестирования. Вопросы могут быть с выбором из нескольких вариантов или со свободным ответом - на Ваше желание и усмотрение.

Все сервисы в программе должны решать строго определённую задачу. Контекст описывается XML-файлом.

Все зависимости должны быть настроены в IoC контейнере. Имя ресурса с вопросами (CSV-файла) необходимо захардкодить в XML-файле с контекстом. CSV с вопросами читается именно как ресурс, а не как файл. Scanner и стандартные типы в контекст класть не нужно!

Опционально: сервисы, по возможности, покрыть Unit-тестами.

Код, написанный в данном ДЗ будет использоваться дальше в домашних заданиях №№ 2-5.

Тема 2: Конфигурирование Spring-приложений

Конфигурировать Spring-приложения в Javabased стиле, писать Spring Beans в Annotation-based стиле, познакомиться с многоуровневой и луковой (Onion) архитектурами, пользоваться Spring Expression Language (SpEL), задавать параметры приложения с помощью .properties файлов, локализовывать приложения

Домашние задания

Добавить файл настроек, Annotation- + Javabased конфигурация приложения

Цель: Цель: научится современным образом конфигурировать Spring-приложения Результат: готовое современное приложение на чистом Spring

Добавьте файл настроек для приложения тестирования студентов. В конфигурационный файл можно поместить путь до CSV-файла и/или текущую локаль, количество правильных ответов для зачёта - на Ваше усмотрение.

Если Вы пишите интеграционные тесты, то не забудьте добавить аналогичный файл и для тестов.

Локализовать выводимые сообщения и вопросы (в CSV-файле). И переписать конфигурацию в виде Java + Annotation-based конфигурации.

Тема 3: AOP, Spring AOP

узнать об аспектно-ориентированном программировании, разобраться в ключевой части Spring - Spring AOP, реализовывать в приложениях crosscuttingфункциональность с помощью Spring AOP

Тема 4: "Чёрная магия" Spring Boot

ориентироваться в возможностях Spring Boot для различных функциональностей и технологий, максимально быстро создавать production-grade standalone Spring-приложения с помощью Spring Boot Starters, писать автоконфигурации и использовать существующие, писать property в YAML-формате

Домашние задания

Перенести приложение для тестирования студентов на Spring Boot

Создать проект, используя Spring Boot Initializr.
Перенести приложение проведения опросов из прошлого домашнего задания с Annotationbased конфигурацией. Перенести все свойства в application.yml
Сделать собственный баннер для приложения.
Перенести тесты и использовать spring-boottest-starter для тестирования

Коммитить wrapper или нет в репозиторий - решать Вам.
*Опционально - использовать ANSI-цвета для баннера
Написанное приложение, будет использоваться в ДЗ №4 (к занятию №5).

Тема 5: Продвинутая конфигурация Spring-приложений

По окончанию данного занятия слушатели смогут использовать Best Practices для конфигурирования Spring-приложений, максимально эффективно использовать аннотации конфигураций.

Домашние задания

Перевести приложение для проведения опросов на Spring Shell Подключить Spring Shell, используя starter.
Написать юнит-тесты и использовать springboot-test-starter

Работа с базами данных Модуль №2

Тема 1: DAO на Spring JDBC

Слушатели смогут ориентироваться в архитекрутных паттернах, свзяанных с работой с БД. Слушатели смогут эффективно использовать Spring JDBC для разработки DAO в приложении.

Домашние задания

Создать приложение хранящее информацию о книгах в библиотеке
Использовать Spring JDBC и реляционную базу.

Опционально использовать настоящую реляционную БД, но можно использовать H2. Предусмотреть таблицы авторов, книг и жанров. Интерфейс на Spring Shell
Покрыть тестами, насколько это возможно. НЕ делать AbstractDao. НЕ делать наследования в тестах

Тема 2: Основы ORM, JPA, Hibernate как провайдер JPA

По окончанию этого занятия слушатели смогут эффективно применять JPA для описания маппинга классов-entities на таблицы реляционной БД. Также слушатели смогут использовать Hibernate, как провайдера JPA для подключения к БД.

Тема 3: JPQL, Spring ORM, DAO на основе Spring ORM + JPA

По окончанию данного модуля слушатели смогут разрабатывать ORM DAO в Spring-приложении с помощью Spring ORM + JPA + Hibernate (в качестве провайдера JPA). Также слушатели узнаю про JPQL (аналог HQL).

Домашние задания

Переписать приложение для хранения книг на ORM
Использовать JPA, Hibernate только в качестве JPA-провайдера.
Добавить комментарии к книгам, и высокоуровневые сервисы, оставляющие комментарии к книгам.
Покрыть DAO тестами используя H2 базу данных и соответствующий H2 Hibernate-диалект

Тема 4: Транзакции, Spring Tx

Слушатели погрузятся в теорию транзакций и поймут все особенности транзакций. Также слушатели смогут использовать декларативное и императивное управление транзакциями в Spring-приложениях с помощью Spring Tx.

Тема 5: "Белая магия" Spring Data: Spring Data JPA

После данного занятия слушатели узнаю про набор проектов Spring Data и понятие репозитория, которое вводит Spring Data. Также слушатели научатся использовать мощную "белую магию" Spring Data ORM для создания ORM DAO на основе JPA.

Домашние задания

Билиотеку на Spring Data JPA Реализовать весь функционал работы с БД в приложении книг с использованием spring-datajpa репозиториев.

Тема 6: SQL и NoSQL базы данных

По окончанию данного семинара слушатели начнут разбираться в особенностях реляционных и различных нереляционных (NoSQL) баз данных.

Также слушатели научатся правильно выбирать NoSQL БД для решения соответствующих задач.

Тема 7: Spring Data для подключения к нереляционным БД

После данного занятия слушатели смогу разрабатывать DAO, хранящие данные в нереляционных БД с помощью других Spring Data проектов.

Домашние задания

Использовать MogoDB и spring-data для хранения информации о книгах Тесты можно реализовать с помощью springboot-strter-embedded-mongodb

Разработка Web-приложений

Модуль №3

Тема 1: Введение в Spring MVC, Spring MVC на Spring Boot

Слушатели смогут ориентироваться в архитектуре MVC и Spring MVC, создавать простые классические веб-приложения на основе Spring MVC и Spring Boot.

Тема 2: Spring MVC View

По окончании данного занятия слушатели смогут разрабатывать View в классических Webприложениях, как с использованием JSP, так и с помощью современных технологий: Thymeleaf, Freemarker, и т.д.

Домашние задания

CRUD приложение с Web UI и хранением данных в БД Создайте приложение с хранением сущностей в БД (можно взять DAOs из прошлых занятий) Использовать классический View, предусмотреть страницу отображения всех сущностей и создания/ редактирования. View на Thymeleaf, classic Controllers.

Тема 3: Современные приложения на Spring MVC

Слушатели смогут создавать современные приложения (основанные на AJAX архитектуре и SPA-приложения). Ну и, конечно, после данного занятия слушатели смогут создавать контроллеры всех сортов и мастей для решения большого спектра задач в веб-приложениях. А также слушатели познакомятся с высокоуровневым WebFlow для описания Webприложений.

Домашние задания

Переписать приложение с использованием AJAX и REST-контроллеров Переписать приложение с классических View на AJAX архитектуру и REST-контроллеры. Опционально: Сделать SPA приложение на любом из Web-фреймоврков

Тема 4: Реактивное программирование

В данном модуле слушатели узнают, что такое Reactive программирование и познакомятся с библиотекой RxJava.

Тема 5: Reactive Spring Frameworks

По окончанию данного модуля слушатели узнают про реактивные фреймоврки в стеке Spring и научаться использовать Reactive-версию Spring Data репозиториев.

Тема 6: Spring WebFlux

После данного занятия слушатели смогу создавать современные Reactive Web приложения с помощью Spring WebFlux.

Домашние задания

Использовать WebFlux Вместо классического потока и embedded Web-сервера использовать WebFlux.

"Около" и "Дзен" Модуль №4

Тема 1: Spring Security: Архитектура

По окончанию занятия слушатели разберутся что такое аутентификация и авторизация, разберутся в архитектуре Spring Security, и смогут настроить HTTP Basic Auth аутнетификацию.

Тема 2: Spring Security: Механизмы аутентификации

По окончанию занятия слушатели смогут внедрять в приложение любой механизм аутентификации.

Домашние задания

В CRUD Web-приложение добавить механизм аутентификации В существующее CRUD-приложение добавить мехнизм Form-based аутентификации. UsersServices реализовать самостоятельно.

Тема 3: Spring Security: Авторизация

После занятия пользователи смогу внедрять в приложение различные механизмы авторизации - на основе URL, методов сервисов.

Тема 4: Spring Security: ACL

После прохождения данного модуля слушатели научатся внедрять в приложение безопасность на основе доменных сущностей: ACLs

Домашние задания

Ввести авторизацию на основе URL и/или доменных сущностей Настроить в приложении авторизацию на уровне URL и/или доменных сущностей.

Тема 5: Spring Batch

Слушатели смогут использовать всю мощь Spring Batch, узнают когда он необходим проекту и почему он нужен не только для больших проектов.

Домашние задания

Разработать процедуру миграции данных из реляционного хранилища в NoSQL или наоборот Используя Spring Batch. Опционально: Сделать restart с помощью Spring Shell.

Тема 6: Монолиты vs. Microservices Round 1, Messaging, Enterprise Integration Patterns (EIP)

По окончании данного модуля слушатели узнают два похода к разработке Enterprise-приложений - монолиты и микросервисы. Узнают, какие проблемы возникают при создании монолитов, что такое Messaging и Enterprise Integration Patterns (EIP) и где здесь Spring Integration.

Тема 7: Spring Integration: Messages и Channels

Слушатели узнают различные семантики каналов, все сорта различных каналов и где они используются. Также слушатели узнают о сообщениях, которые передаются в каналах и встроенный DSL для настройки связей в Spring Integration. Также слушатели узнают про базовые Endpoints и Flow Components.

Домашние задания

Реализовать обработку доменной сущности через каналы Spring Integration Выберите подходящий канал для каждого соответствующего запроса Опционально: протестируйте приложение под нагрузкой.

Тема 8: Spring Integration: Endpoints и Flow Components

Слушатели также узнают про другие Endpoints и Flow Components и смогут разрабатывать сложные Enterprise-приложения с почти любой интеграцией.

Тема 9: Монолиты vs. Microservices (Round 2), Spring Boot Actuator - must have в микросервисах

На данном занятии слушатели будут рассматривать возможности Spring Boot Actuator для создания production-grade приложений и микросервисов, а потом будут долго отходить от таких возможностей и изобилия. Также в данном разделе будет рассмотрен HATEOAS подход для разработки REST-сервисов.

Домашние задания

Использовать метрики, healthchecks и logfile к приложению И любую другую функциональность на выбор. Опционально: переписать приложение на HATEOAS принципах.

Тема 10: REST-клиенты, SOAP, Spring WebServices и клиенты к ним

Слушатели научаться писать REST-клиентов к микросервисам. Также, после занятия слушатели овладеют одним из самых простых способов создания SOAP-сервисов и клиентов к ним Spring WebServices, ну и, конечно будет рассмотрены SOA и SOAP.

Тема 11: Docker, оркестрация, облака, облачные хостинги

По окончании данного занятия слушатели смогут разбираться в вышеперечисленных словах, а также разбираться в современных принципах построения облачных систем.

Домашние задания

Обернуть приложение в docker-контейнер Обернуть приложение в docker-контейнер, БД тоже. Настроить связь между ними. Опционально: сделать это в локальном кубе.

Тема 12: Spring Cloud: Config Server, Service Registry, интеграция в облака

Слушатели научаться пользоваться возможностями Spring для интеграции с облаками: Config Server, Service Registry, Docker/Kubernates/AWS/Azure bestpractices.

Тема 13: Spring Cloud Data Flow, Hystrix Circuit Breaker

Слушатели смогут узнать как строятся огромные системы на Spring с использованием Spring Cloud Data Flow. Также будет рассмотрен популярный фреймворк для использования внешних систем и ресурсов - Hystrix (+Hystrix Javanica) и его интеграция со Spring.

Домашние задания

Обернуть внешние вызовы в Hystrix Обернуть все внешние вызовы в Hystrix, Hystrix Javanica. Опционально: Поднять Turbine Dashboard для мониторинга.

Тема 14: Обзор дополнительных технологий Spring, выбор архитектуры и технологий

По окончанию занятия слушатели познакомятся с другими проектами Spring для создания приложений. Смогут правильно выбирать архитектуру и стек технологий для проекта.

Проектная работа

Тема 1: Вводное занятие по проектной работе

Выбрать и обсудить предполагаемую тему проектной работы

Домашние задания

Проектная работа Проект должен быть сделан на основе Spring Boot, включать работу с DB с использованием Spring Data репозиториев и/или Spring JDBC. Проект должен иметь UI построенный на современных принципах разработки Web приложений (AJAX и/или SPA). Приложение должно содержать механизмы аутентификации и авторизации с использованием Spring Security Асинхронные части могут быть реализованы с помощью Spring Integration. Пакетные обработки, утилиты поддержки должны быть реализованы с помощью Spring Batch + Spring Shell. Проект должен быть cloud-ready.

Тема 2: Консультация по проекту + пробная защита проекта

Консультирование слушателей по вопросам проектной работы.

Тема 3: Защита проектных работ №1

На данном занятии слушатели будут защищать собственные проекты.

Тема 4: Защита проектных работ №2

На этом занятии слушатели могут защитить свои проектные работы.