Разработчик на Spring Framework

Курс о разработке веб-приложений на Spring, о фреймворках и вспомогательных технологиях Spring.

Длительность курса: 164 академических часа

1 Введение

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 1 Программа по проведению тестирования студентов Цель: Цель задания: создать приложение с помощью Spring IoC, чтобы познакомиться с основной функциональностью IoC, на которой строится весь Spring.

Результат: простое приложение, сконфигурированное XML-контекстом.

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.

Требования:
1. Все сервисы в программе должны решать строго определённую задачу.
2. Контекст описывается XML-файлом.
3. Все зависимости должны быть настроены в IoC контейнере.
4. Имя ресурса с вопросами (CSV-файла) необходимо захардкодить в XML-файле с контекстом.
5. CSV с вопросами читается именно как ресурс, а не как файл.
6. Scanner и стандартные типы в контекст класть не нужно!
*Опционально: сервисы, по возможности, покрыть Unit-тестами.

Задание сдаётся в виде ссылки на pull-request в чат с преподавателем.
Вопросы можно задавать в чате, но для оперативности рекомендуем Slack.

Код, написанный в данном ДЗ будет использоваться дальше в домашних заданиях #2 (Занятие #2), #3 (Занятие #4), #4 (Занятие #5)

2 **Конфигурирование Spring-приложений**

конфигурировать Spring-приложения в современном Java-based стиле (как сейчас и все пишут),
ориентироваться в многослойной и луковой (Onion) архитектурах,
пользоваться Spring Expression Language (SpEL),
задавать параметры приложения с помощью .properties файлов,
локализовывать приложения

Домашние задания

1 Добавить файл настроек, Annotation- + Java-based конфигурация приложения

Цель: Цель: конфигурировать Spring-приложения современным способом, как это и делается в современном мире
Результат: готовое современное приложение на чистом Spring

Выполняется на основе предыдущего домашнего задания.

1. 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.
2. Если Вы пишите интеграционные тесты, то не забудьте добавить аналогичный файл и для тестов.
3. Локализовать выводимые сообщения и вопросы (в CSV-файле).
4. И переписать конфигурацию в виде Java + Annotation-based конфигурации.

Задание сдаётся в виде ссылки на pull-request в чат с преподавателем.
Вопросы можно задавать в чате, но для оперативности рекомендуем Slack.

Код, написанный в данном ДЗ будет использоваться дальше в домашних заданиях №3 (Занятие №4), №4 (Занятие №5)
Данное задание засчитывает ДЗ №1 (Занятие №1).
Если Вы хотите засчитать, то обязательно пришлите ссылку в чат соответствующего предыдущего занятия.

3	**AOP, Spring AOP**	использовать аспектно-ориентированное программирование (там где нужно), видеть в коде ключевую функциональность Spring - Spring AOP, реализовывать в приложениях crosscutting-функциональность с помощью Spring AOP

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 1 Перенести приложение для тестирования студентов на Spring Boot Цель: Цель: использовать возможности Spring Boot, чтобы разрабатывать современные приложения, так, как их сейчас и разрабатывают. Результат: Production-ready приложение на Spring Boot Это домашнее задание выполняется на

основе предыдущего.

1. Создать проект, используя Spring Boot Initializr (https://start.spring.io)
2. Перенести приложение проведения опросов из прошлого домашнего задания. MesageSource должен быть из автоконфигурации Spring Boot.
3. Перенести все свойства в application.yml
4. Сделать собственный баннер для приложения.
5. Перенести тесты и использовать spring-boot-test-starter для тестирования
*Опционально - использовать ANSI-цвета для баннера.

Коммитить wrapper или нет в репозиторий - решать Вам.

Задание сдаётся в виде ссылки на pull-request в чат с преподавателем.
Вопросы можно задавать в чате, но для оперативности рекомендуем Slack.

Написанное приложение будет использоваться в ДЗ №4 (к занятию №5).
Данное задание засчитывает ДЗ №1 (к занятию №1) и ДЗ №2 (к занятию №2).
Если Вы хотите засчитать, то обязательно пришлите ссылку в чат соответствующего предыдущего занятия.

5	**Продвинутая конфигурация Spring-приложений**	использовать Best Practices для конфигурирования Spring-приложений, максимально эффективно использовать аннотации конфигураций, писать приложения с использованием Spring Shell Домашние задания

1 Перевести приложение для проведения опросов на Spring Shell

Цель: Цель: Научиться использовать Spring Shell, чтобы писать интерфейс приложения без Web.
Результат: Приложение на Spring Shell

Домашнее задание выполняется на основе предыдущего.

Необходимо:
1. Подключить Spring Shell, используя spring-starter.
2. Написать набор команд, позволяющий проводить опрос.
3. Написать Unit-тесты с помощью spring-boot-starter-test, учесть, что Spring Shell в тестах нужно отключить.

Набор команд зависит только от Вашего желания. Вы можете сделать одну команду, запускающую Ваш Main, а можете построить полноценный интерфейс на Spring Shell.

Локализовывать команды Spring Shell НЕ НУЖНО (хотя можно, но это долго и непросто).

Задание сдаётся в виде ссылки на pull-request в чат с преподавателем.
Вопросы можно задавать в чате, но для оперативности рекомендуем Slack.

Это домашнее задание больше нигде не будет использоваться. Но интерфейс Spring Shell мы будет использовать в дальнейшем.

6	**Разбор домашних заданий, QnA**	писать код с учётом Best Practices, не допускать частых ошибок, получить ответы на вопросы

2 Работа с базами данных

1 **DAO на Spring JDBC**

эффективно использовать JDBC вместе со Spring JDBC для разработки приложений с мощью чистого SQL,
правильно применять паттерн DAO для подключения к БД,
пользоваться embedded БД для написания тестов и при разработке простых приложений

Домашние задания

1 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

---

2 **Основы ORM, JPA, Hibernate как провайдер JPA**

эффективно применять JPA для описания маппинга классов-entities на таблицы реляционной БД,
использовать Hibernate в качестве JPA Vendor

3	**JPQL, Spring ORM, DAO на основе Spring ORM + JPA**	разрабатывать ORM DAO с помощью Spring ORM + JPA + Hibernate (в качестве JPA Vendor-a) в Spring приложениях; использовать JPQL (аналог HQL) для построения SQL-подобных запросов. Домашние задания 1 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
4	**Транзакции, Spring Tx**	понимать и применять особенности транзакции в реляционных БД для правильной разработки слоя DAO, использовать декларативное и императивное управление транзакциями в Spring-приложениях с помощью Spring Tx

5	**"Белая магия" Spring Data: Spring Data JPA**	разбираться в проектах Spring Data для написания репозиториев в максимально простом виде; научатся использовать мощную "белую магию" Spring Data JPA для создания репозиториев для JPA сущностей Домашние задания 1 Билиотеку на Spring Data JPA Реализовать весь функционал работы с БД в приложении книг с использованием spring-data-jpa репозиториев.
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.
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 1 Использовать MogoDB и spring-data для хранения информации о книгах Тесты можно реализовать с помощью spring-boot-strter-embedded-mongodb
8	**Разбор домашних заданий, QnA**	писать код с учётом Best Practices, не допускать частых ошибок, получить ответы на вопросы

3 Разработка Web-приложений

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.

---

2 **Spring MVC View**

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

Домашние задания

1 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.

---

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 Web-приложений.

Домашние задания

1 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

---

4 **Реактивное программирование**

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

---

5 **Reactive Spring Frameworks**

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

---

6 **Spring WebFlux**

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

Домашние задания

1 Использовать WebFlux

Вместо классического потока и embedded Web-сервера использовать WebFlux.

4 "Около" и "Дзен"

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.

---

2 **Spring Security: Механизмы аутентификации**

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

Домашние задания

1 В CRUD Web-приложение добавить механизм аутентификации

В существующее CRUD-приложение добавить мехнизм Form-based аутентификации.

UsersServices реализовать самостоятельно.

Внимание! Задание выполняется на основе нереактивного приложения Sping MVC!

---

3 **Spring Security: Авторизация**

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

4	**Spring Security: ACL**	После прохождения данного модуля слушатели научатся внедрять в приложение безопасность на основе доменных сущностей: ACLs Домашние задания 1 Ввести авторизацию на основе URL и/или доменных сущностей Настроить в приложении авторизацию на уровне URL и/или доменных сущностей. Внимание! Задание выполняется на основе нереактивного приложения Sping MVC!
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 1 Разработать процедуру миграции данных из реляционного хранилища в NoSQL или наоборот Используя Spring Batch. Опционально: Сделать restart с помощью Spring Shell.
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.

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

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

1. Использовать метрики, healthchecks и logfile к приложению

   И любую другую функциональность на выбор.

   Опционально: переписать приложение на HATEOAS принципах.

---

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.

---

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

1 Обернуть приложение в docker-контейнер

Обернуть приложение в docker-контейнер, БД тоже. Настроить связь между ними.

Опционально: сделать это в локальном кубе.

---

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 best-practices.

---

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

1 Обернуть внешние вызовы в Hystrix

Обернуть все внешние вызовы в Hystrix, Hystrix Javanica.

Опционально: Поднять Turbine Dashboard для мониторинга.

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.

5 Проектная работа

1	**Вводное занятие по проектной работе**	Выбрать и обсудить предполагаемую тему проектной работы Домашние задания 1 Проектная работа Проект должен быть сделан на основе Spring Boot, включать работу с DB с ипользованием 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. Пектные обработки, утилиты поддержки должны быть реализованы с помощью Spring Batch + Spring Shell. Проект должен быть cloud-ready.
2	**Консультация по проекту + пробная защита проекта**	Консультирование слушателей по вопросам проектной работы.
3	**Защита проектных работ №1**	На данном занятии слушатели будут защищать собственные проекты.
4	**Защита проектных работ №2**	На этом занятии слушатели могут защитить свои проектные работы.