|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Инструментального и прикладного программного обеспечения

**ПРАКТИЧЕСКАЯ РАБОТА №5**

по дисциплине «Разработка серверных частей интернет-ресурсов»

**Тема практической работы:**

**Студент группы** ИКБО-33-21 Май Ван Туан

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

**Руководитель практической работы** преподаватель Волков М.Ю.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Работа представлена «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Допущен к работе «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Москва 2023

СОДЕРЖАНИЕ

[**1. Цель работы** 3](#_Toc145190172)

[**2. Ход работы** 3](#_Toc145190173)

[**3. Ответы на вопросы** 5](#_Toc145190174)

## Постановка задачи

Приложение должно стать основанием для создания маркетплейса. В работе должны быть реализованы такие модели:

1. Book – модель, которая описывает сущность книги, с полями: автор, номер продавца, тип продукта (подразумевается электроника, книги, сантехника и т. п.), стоимость, название.
2. Client – модель, которая описывает сущность клиента сайта. У модели должны быть поля: имя, электронная почта, логин, пароль.
3. Telephone – модель, описывающая сущность телефона. Обязательные поля: производитель, объем аккумулятора, номер продавца, тип продукта (подразумевается электроника, книги, сантехника и т. п.), стоимость, название.
4. WashingMachine – модель, которая описывает сущность стиральной машины, с такими полями: производитель, объем бака, номер продавца, тип продукта (подразумевается электроника, книги, сантехника и т. п.), стоимость, название.

Так же должны быть контроллеры, которые позволят манипулировать модели (CRUD). Вся настройка приложения должна быть реализована в директории configuration. Приложение должно запускаться с помощью docker-compose.

## Ход работы

Для начала создадим все нужные файлы для создания конфигурации серверного ПО (рис. 1).

A screenshot of a computer

Description automatically generated

1. Структура образа

Содержимое всех необходимых файлов показано на рисунках 2-7.

A computer screen shot of a black screen

Description automatically generated

1. Файл application.properties

A black rectangular object with a black border

Description automatically generated

1. Файл Dockerfile

A screenshot of a computer program

Description automatically generated

1. Файл docker-composer.yml

A screen shot of a computer program

Description automatically generated

1. Файл Book

A screen shot of a computer

Description automatically generated

1. Файл BoolRepository

A screenshot of a computer program

Description automatically generated

1. Сборка BookController

Далее была протестирована функциональность серверного Postmen (Рисунок 8 -10).

A screenshot of a computer

Description automatically generated

1. POST

A screenshot of a computer

Description automatically generated

1. GET

A screenshot of a computer

Description automatically generated

1. DELETE

## Ответы на вопросы:

* **Что такое JDBC и JPA, в чем отличия?**

JDBC (Java Database Connectivity) - это набор Java API для взаимодействия с реляционными базами данных с использованием SQL запросов. JPA (Java Persistence API) - это стандарт Java EE для управления объектами и их хранения в базе данных. Основное отличие заключается в том, что JDBC предоставляет более низкоуровневый доступ к базе данных, требует написания SQL-запросов, в то время как JPA позволяет работать с объектами и скрывает детали взаимодействия с базой данных.

* **Основные интерфейсы для реализации Spring Data?**

Основные интерфейсы для реализации Spring Data включают:

* JpaRepository: Предоставляет CRUD-операции для работы с сущностями базы данных.
* CrudRepository: Предоставляет базовые CRUD-операции для сущностей.

PagingAndSortingRepository: Позволяет выполнить пагинацию и сортировку результатов запросов.

* **Аннотации @Entity, @Table. В чем разница?**

@Entity - это JPA-аннотация, которая помечает класс как сущность, которая может быть сохранена в базе данных. @Table - это JPA-аннотация, которая позволяет настраивать свойства таблицы, связанные с сущностью. Разница заключается в том, что @Entity применяется к классу, чтобы указать, что он является сущностью, а @Table применяется к классу для определения настраиваемых параметров таблицы.

* **Что такое Lombok. Приведите несколько примеров использования.**

Lombok - это библиотека для упрощения разработки на Java. Она генерирует код для методов доступа, конструкторов и других рутинных операций. Примеры использования Lombok:

* @Data: Генерация геттеров, сеттеров, toString, equals и hashCode.
* @Getter и @Setter: Генерация геттеров и сеттеров.
* @NoArgsConstructor и @AllArgsConstructor: Генерация конструкторов без аргументов и с аргументами.
* **Назовите способы инициализации бинов в Java.**

Способы инициализации бинов в Java включают:

* + Аннотация @Component: Можно использовать для автоматического обнаружения и инициализации бинов.
  + XML-конфигурация: Можно определить бины в файле XML и затем загрузить их с помощью контейнера Spring.
  + Java-конфигурация: Можно использовать Java-классы для определения конфигурации и инициализации бинов.
* **Что такое миграции и как они работают в библиотеке Flyway?**

Миграции - это способ управления изменениями в структуре базы данных. В библиотеке Flyway миграции представляют собой SQL-скрипты, которые описывают изменения схемы базы данных. Flyway автоматически применяет миграции в порядке их версий при запуске приложения, обновляя базу данных.

* **Что такое сущность и модуль. В чем разница?**

Сущность (Entity) - это объект, который представляет данные, которые могут быть сохранены в базе данных. Модуль - это набор классов и ресурсов, организованных в логическую единицу для выполнения определенных функций. Разница заключается в том, что сущность связана с данными и их хранением, в то время как модуль представляет собой логический компонент системы, который может включать в себя множество классов и функциональность.