**ЛАБОРАТОРНАЯ РАБОТА №3**

**JAVA SPRING BOOT**

**Цель работы:** познакомиться с фреймворком Spring Boot и уровнем абстракции базы данных Java Persistence Api.

Программное обеспечение: IntelliJ IDEA, JDK 1.8+, Spring Boot

Необходимая теоретическая подготовка: Maven, Java Spring, JPA.

1. **Что такое JDBC, JPA и Hibernate? В чем их разница?**

JDBC (Java Database Connectivity), JPA (Java Persistence API) и Hibernate - это технологии, используемые в Java для работы с базами данных. Вот их краткое описание и разница между ними:

1) JDBC (Java Database Connectivity):

- Это низкоуровневый API, предоставляющий Java-приложениям доступ к базам данных.

- JDBC позволяет установить соединение с базой данных, выполнить SQL-запросы и обрабатывать результаты.

- Разработчикам приходится самим писать SQL-запросы и управлять соединениями с базой данных.

2) JPA (Java Persistence API):

- Это высокоуровневый стандартный интерфейс Java для работы с объектно-реляционными отображениями (ORM).

- JPA абстрагирует работу с базой данных, позволяя разработчикам оперировать объектами Java, а не SQL-запросами.

- JPA предоставляет аннотации для маппинга Java-объектов на таблицы в базе данных.

3) Hibernate:

- Hibernate - это конкретная реализация JPA и один из самых популярных фреймворков ORM для Java.

- Hibernate обеспечивает управление жизненным циклом объектов, маппинг объектов на таблицы и генерацию SQL-запросов автоматически.

- Он также предоставляет богатый набор функциональных возможностей для работы с базами данных.

Разница между JDBC и JPA/Hibernate заключается в уровне абстракции. JDBC - это более низкоуровневый подход, требующий написания много SQL-кода и управления соединениями. JPA и Hibernate предоставляют более высокоуровневые средства для работы с базой данных, позволяя разработчикам более легко оперировать объектами и скрывая детали работы с SQL. Hibernate, в частности, является одной из реализаций JPA и предоставляет расширенные функции ORM.

1. **Как можно запустить Spring Boot?**

Для запуска приложения на основе Spring Boot, необходимо выполнить следующие шаги:

1. Создать проект Spring Boot:

Можно создать новый проект Spring Boot с использованием инструмента Spring Initializr. Это можно сделать вручную через веб-интерфейс (https://start.spring.io/) или с использованием среды разработки, такой как IntelliJ IDEA или Eclipse.

1. Определить зависимости:

Выбрать необходимые зависимости (например, Spring Web, Spring Data JPA, Spring Security) при создании проекта или позже, добавив их в файл pom.xml (для Maven) или build.gradle (для Gradle).

1. Написать код:

Разработать свое приложение, создав Java-классы, контроллеры, сервисы и другие компоненты. Конфигурировать приложение, используя аннотации Spring и файлы конфигурации, если это необходимо.

1. Запустить приложение:

Можно запустить свое приложение Spring Boot, запустив метод main в классе, помеченном аннотацией @SpringBootApplication. Этот метод автоматически настроит и запустит встроенный веб-сервер (например, Tomcat или Jetty).

1. Проверить приложение:

После запуска приложения открыть веб-браузер и перейдите по адресу http://localhost:8080 (по умолчанию) или другому порту.

1. Развернуть приложение:

Если нужно развернуть приложение на боевом сервере, необходимо убедиться, что у вас есть настроенное окружение (например, база данных, настройки безопасности) и использовать инструменты развертывания, такие как Apache Tomcat, чтобы разместить приложение.