**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.

**Контрольные вопросы:**

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

JDBC — это платформенно независимый промышленный стандарт взаимодействия Java-приложений с реляционными базами данных.

JPA (Java Persistence API) — это спецификация Java EE и Java SE, описывающая систему управления сохранением java объектов в таблицы реляционных баз данных в удобном виде.

Hibernate — это фреймворк для языка Java, предназначенный для работы с базами данных.

JPA это спецификация, подход к работе с БД, когда Hibernate - это один из методов ее реализации, фреймворк для работы с данной спецификацией.

1. **Как развернуть приложение Spring Boot на сервере (tomcat)?**

1. Установите Apache Tomcat на свой сервер, если он еще не установлен. Вы можете загрузить его с официального сайта Apache Tomcat.

2. Создайте JAR-файл вашего Spring Boot приложения. Это можно сделать с помощью инструментов сборки, таких как Maven или Gradle. Убедитесь, что вы включили зависимости, необходимые для работы вашего приложения в JAR-файле.

3. Скопируйте JAR-файл в директорию "webapps" вашего установленного Tomcat сервера. Если Tomcat установлен в каталоге /opt/tomcat/, то путь для копирования JAR-файла будет /opt/tomcat/webapps/.

4. Запустите сервер Tomcat, выполнив скрипт start.sh (для Unix) или startup.bat (для Windows) в директории bin вашего установленного Tomcat-сервера.

5. Проверьте, работает ли ваше приложение, открыв веб-браузер и перейдя по адресу localhost:8080/имя\_вашего\_приложения. Если все настроено правильно, вы должны увидеть страницу вашего Spring Boot приложения.

1. **В чем разница между Spring Boot и Spring MVC?**

Spring MVC - это модуль Spring Framework для разработки веб-приложений на основе паттерна MVC, а Spring Boot - это инструмент, который упрощает процесс создания, запуска и развертывания приложений на основе Spring.

1. **Что такое War-файл и отличие от Jar-файла?**

.jar файлы используются для распространения Java приложений или библиотек, в то время как .war файлы используются для распространения веб-приложений на платформе Java EE.

1. **Зачем использовать Spring Boot?**

Модуль упрощает настройку фреймворка Spring и ускоряет запуск проектов. Он может автоматически сконфигурировать приложение и создать веб-сервер для его запуска.

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

Есть несколько способов запустить Spring Boot приложение:

1. Использование командной строки:

- Установите Maven или Gradle, если еще не установлены.

- В терминале перейдите в корневую директорию проекта.

- Запустите следующую команду:

- С помощью Maven: mvn spring-boot:run

- С помощью Gradle: gradle bootRun

2. Через IDE:

- Откройте проект в вашей среде разработки (например, IntelliJ IDEA, Eclipse).

- Найдите и запустите класс с аннотацией @SpringBootApplication (обычно это класс с именем Application или Main).

- Нажмите кнопку "Запуск" или используйте сочетание клавиш, чтобы запустить приложение.

3. С помощью jar файла:

- Сначала вам необходимо создать jar-файл проекта.

- В командной строке перейдите в директорию с jar-файлом.

- Запустите приложение с помощью команды: java -jar <название\_jar\_файла>.jar

Перед запуском приложения необходимо настроить соответствующую конфигурацию, такую ​​как база данных, порт сервера и другие параметры, в файле application.properties или application.yml вашего проекта.

1. **Требуется ли для запуска Spring Boot отдельный контейнер?**

Для запуска Spring Boot приложений не требуется отдельный контейнер. Spring Boot включает в себя встроенный веб-сервер (Tomcat, Jetty или Undertow), который может быть запущен непосредственно из самого приложения.

1. **Как настроить приложение Spring Boot с помощью maven?**

1. Создайте новый проект Maven в IDE вашего выбора (например, IntelliJ IDEA или Eclipse).

2. Укажите Maven GroupId, ArtifactId и версию проекта.

3. В файле pom.xml добавьте зависимость для Spring Boot:

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter</artifactId>

<version>2.5.4</version>

</dependency>

</dependencies>

4. Создайте класс с методом main и аннотацией @SpringBootApplication:

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class MyApp {

public static void main(String[] args) {

SpringApplication.run(MyApp.class, args);

}

}

5. Добавьте в pom.xml плагин Spring Boot Maven для упаковки приложения в исполняемый JAR-файл:

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

<version>2.5.4</version>

</plugin>

</plugins>

</build>

6. Выполните команду mvn clean install для сборки проекта и загрузки всех зависимостей.

7. Запустите приложение, выполнив команду mvn spring-boot:run или запустив класс MyApp с помощью IDE.

Spring Boot будет автоматически настраивать и запускать ваше приложение со стандартными настройками.

1. **Какие встроенные серверы поддерживает Spring Boot?**

Для сервлет-приложений Spring Boot предусматривает поддержку встроенных серверов Tomcat, Jetty, и Undertow.

1. **Для чего используются аннотации @RestController и @Controller?**

Аннотация @RestController в Spring MVC – это сочетание аннотации @Controller и @ResponseBody. Задача @Controller – создать карту объекта модели и найти представление, но @RestController просто возвращает объект, и данные объекта напрямую записываются в HTTP-ответ в виде JSON или XML.

1. **Что такое Thymeleaf?**

Thymeleaf — современный серверный механизм Java-шаблонов для веб- и автономных сред, способный обрабатывать HTML, XML, JavaScript, CSS и даже простой текст.