ЛАБОРАТОРНАЯ РАБОТА 8

РАЗРАБОТКА ЭФФЕКТИВНЫХ ВЕБ-ПРИЛОЖЕНИЙ

(SPRING MVC)

**Цель:** разработка веб-приложений с реализацией шаблона MVC.  
**Необходимая теоретическая подготовка:**

* объектно-ориентированное программирование;
* • Maven;
* • шаблоны проектирования.

**Задания к лабораторной работе**

* создать проект Spring MVC и подключить к нему все необходимые зависимости (2 часа).
* создать html формы добавления параметров объектов в проект. Поля, которые
* впоследствии будут получены из базы данных реализовать в виде заглушек.
* все поля для ввода текста должны иметь валидацию.
* Ознакомиться как передать в обработчик параметры запроса и реализовать CRUD (редактирование, удаление и просмотр полной информации об объекте)\*

### Листинг

### Подключение зависимостей(IDEA Ultimate)

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

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

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

<version>3.2.0</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>com.example</groupId>

<artifactId>blog</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>blog</name>

<description>blog</description>

<properties>

<java.version>17</java.version>

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

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

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

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

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

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.33</version>

</dependency>

<dependency>

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

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

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>5.8.0</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.12</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

</build>

</project>

**Результат**

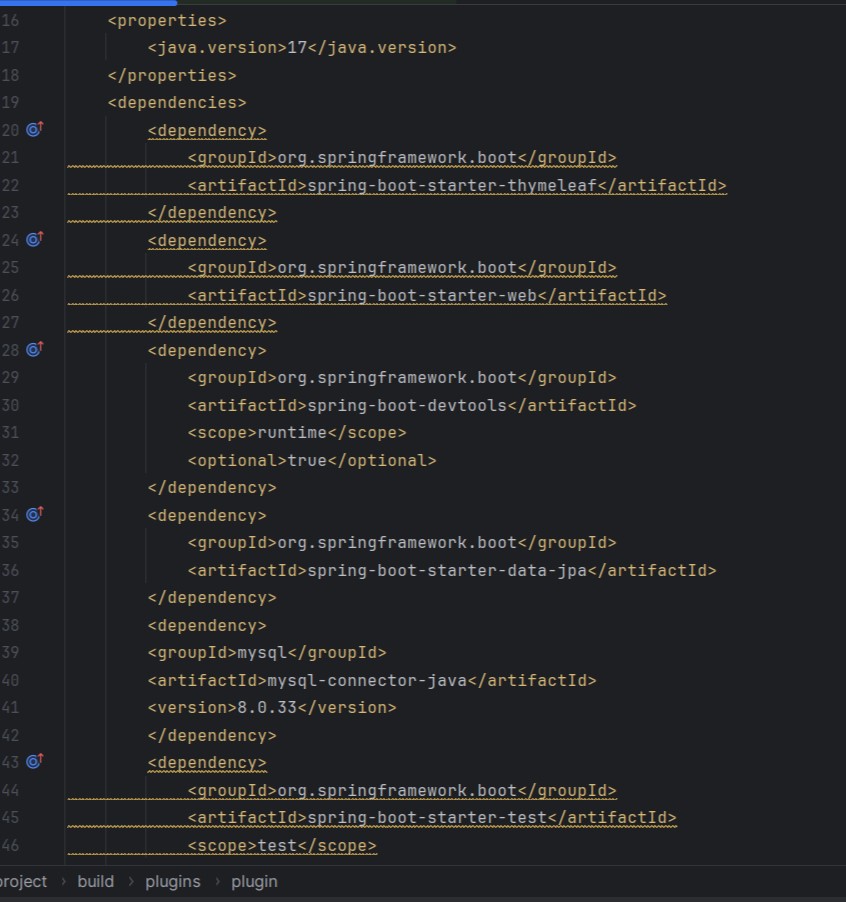


Рисунок 1- подключение зависимостей

### Листинг

### Подключение страниц html(IDEA Ultimate Thymeleaf шаблоны)

### Добавление записи

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org" lang="ru">

<head>

<title>Главная страница</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css">

</head>

<body>

<div class="cover-container d-flex w-100 h-100 p-3 mx-auto flex-column">

<header th:insert="~{blocks/header ::header}"></header>

<div class="container mt-5 mb-5">

<h1>Course adding</h1>

<form action="/blog/add" method="post">

<input type="text" name="title" placeholder="Введите название записи" class="form-control"><br>

<input type="text" name="anons" placeholder="Введите анонс записи" class="form-control"><br>

<textarea name="full\_text" placeholder="Введите полный текст записи" class="form-control"></textarea><br>

<button type="submit" class="btn btn-success">Add</button>

</form>

</div>

<footer th:insert="~{blocks/footer::footer}"></footer>

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KFdBIe4zVF0s0G1M5b4hcpxyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4" crossorigin="anonymous"></script>

</body>

</html>

### Редактирование записи

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org" lang="ru">

<head>

<title>Главная страница</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css">

</head>

<body>

<div class="cover-container d-flex w-100 h-100 p-3 mx-auto flex-column">

<header th:insert="~{blocks/header ::header}"></header>

<div class="container mt-5 mb-5">

<h1>Редактировать записи</h1>

<div th:each="el : ${post}">

<form method="post">

<input type="text" th:value="${el.title}" name="title" placeholder="Введите название записи" class="form-control"><br>

<input type="text" th:value="${el.anons}" name="anons" placeholder="Введите анонс записи" class="form-control"><br>

<textarea name="full\_text" placeholder="Введите полный текст записи" class="form-control" th:text="${el.full\_text}"/><br>

<button type="submit" class="btn btn-success">Update</button>

</form>

</div>

</div>

<footer th:insert="~{blocks/footer::footer}"></footer>

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KFdBIe4zVF0s0G1M5b4hcpxyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4" crossorigin="anonymous"></script>

</body>

</html>

### Удаление записи

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org" lang="ru">

<head>

<title>details</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css">

</head>

<body>

<div class="cover-container d-flex w-100 h-100 p-3 mx-auto flex-column">

<header th:insert="~{blocks/header ::header}"></header>

<div class="container mt-50">

<div th:each="el : ${post}" class="alert alert-info mt-2">

<h3 th:text="${el.title}"/>

<p th:text="${el.full\_text}"/>

<p><b>Просмотры </b><span th:text=" ${el.views}"/></p>

<a th:href="'/blog/'+${el.id}+'/edit'" class="btn-btn-warning"><button class="btn btn-primary">Edit</button> </a><br><br>

<form th:action="'/blog/'+${el.id}+'/remove'" method="post">

<button class="btn btn-primary" type="submit">Remove</button>

</form>

</div>

</div>

<footer th:insert="~{blocks/footer::footer}"></footer>

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KFdBIe4zVF0s0G1M5b4hcpxyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4" crossorigin="anonymous"></script>

</body>

</html>

**Результат**

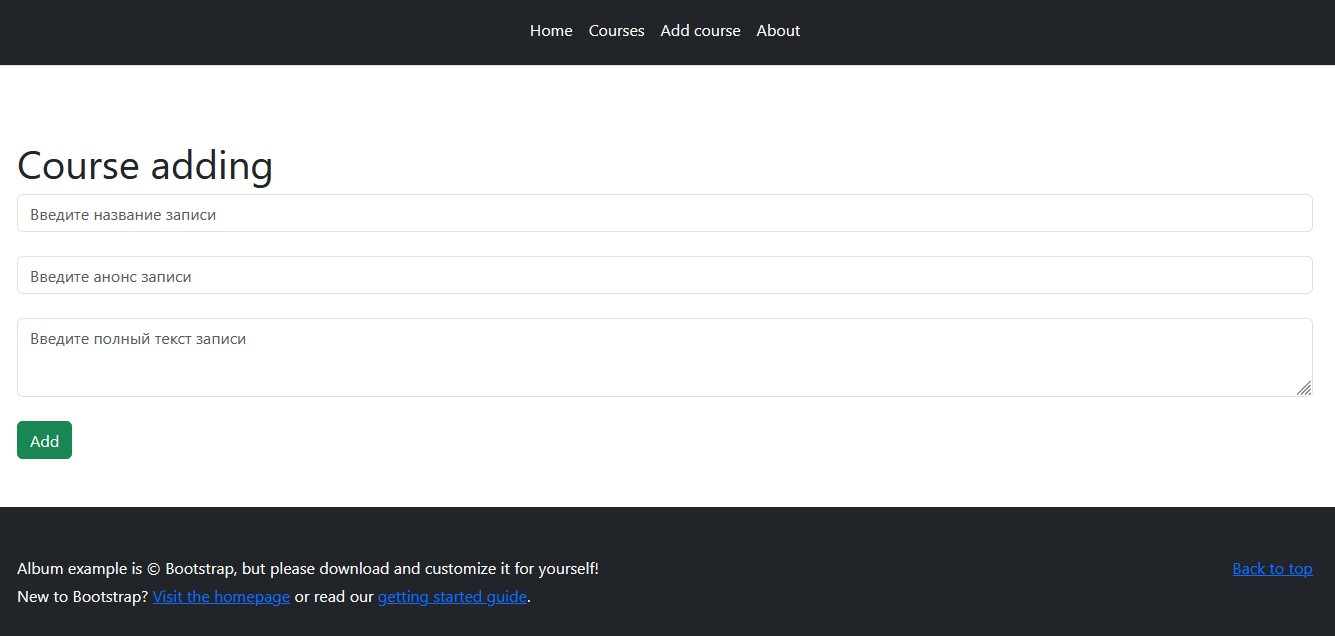


Рисунок 2 – Страница добавления записей

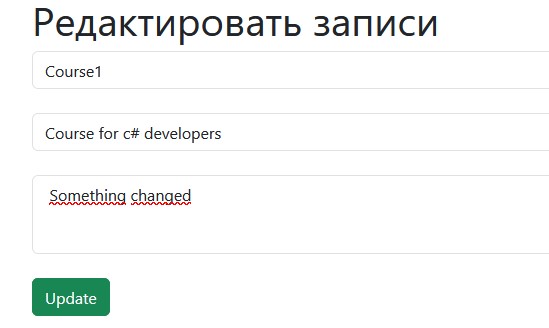


Рисунок 3 – Изменение записи

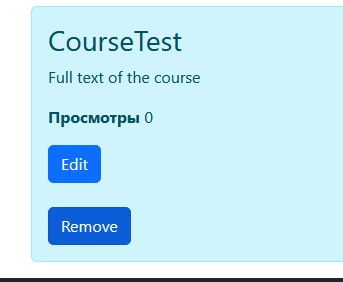


Рисунок 4 – Удаление записи

### Листинг

### Подключение к БД и модель(IDEA Ultimate)

### Application properties

spring.jpa.hibernate.ddl-auto=update

spring.datasource.url=jdbc:mysql://${MYSQL\_HOST:localhost}:3306/spring\_web\_blog

spring.datasource.username=root

spring.datasource.password=root

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

#spring.jpa.show-sql: true

### Model Post class

package com.example.blog.models;

import jakarta.persistence.\*;

@Entity

public class Post {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private Long id;

private String title,anons,full\_text;

private int views;

public Post(String title, String anons, String full\_text) {

this.title = title;

this.anons = anons;

this.full\_text = full\_text;

}

public Post() {

}

public Post(long id, String someName, String someShortText, String someLongText) {

}

public long getId() {

return id;

}

public void setId(long id) {

this.id = id;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getAnons() {

return anons;

}

public void setAnons(String anons) {

this.anons = anons;

}

public String getFull\_text() {

return full\_text;

}

public void setFull\_text(String full\_text) {

this.full\_text = full\_text;

}

public int getViews() {

return views;

}

public void setViews(int views) {

this.views = views;

}

}

**Результат**



Рисунок 5 – Новая запись в БД

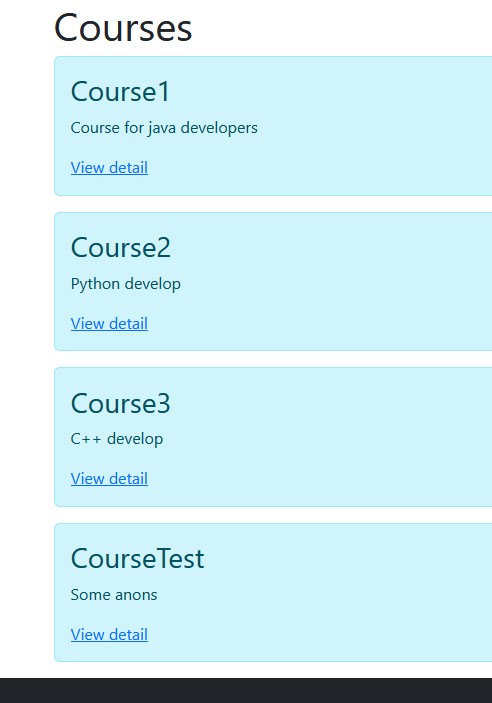


Рисунок 6 – Новая запись на сайте

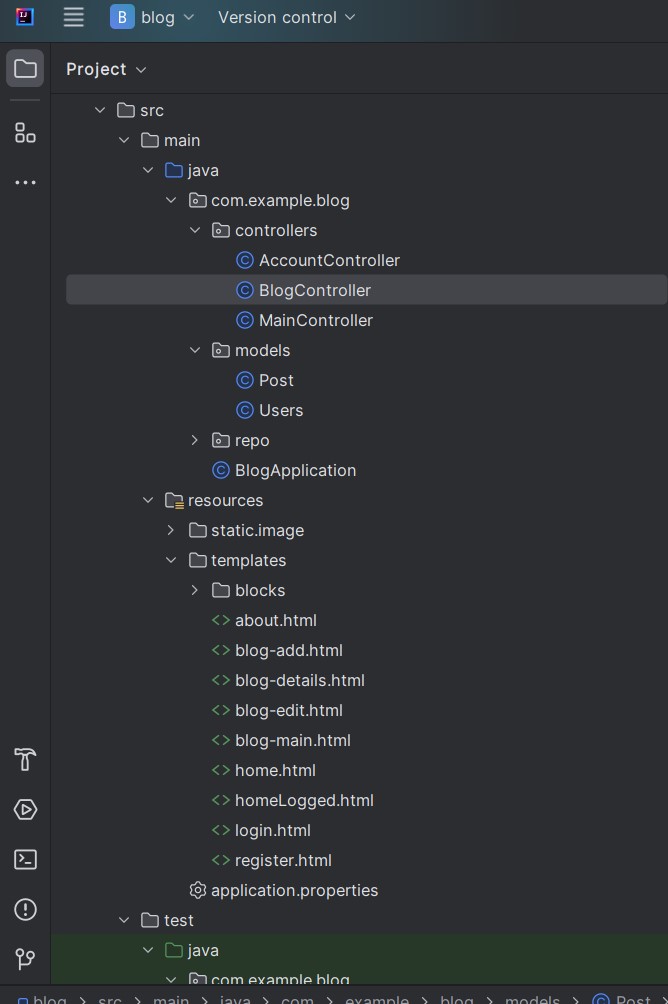


Рисунок 7 – Структура проекта

### Листинг

### Контроллер(IDEA Ultimate)

package com.example.blog.controllers;

import com.example.blog.models.Post;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestParam;

import com.example.blog.repo.PostRepository;

import java.util.ArrayList;

import java.util.Optional;

@Controller

public class BlogController {

@Autowired

private PostRepository postRepository;

@GetMapping("/blog")

public String blogMain(Model model) {

Iterable<Post> posts = postRepository.findAll();

model.addAttribute("posts",posts);

return "blog-main";

}

@GetMapping("/blog/add")

public String blogAdd(Model model) {

return "blog-add";

}

@PostMapping("/blog/add")

public String blogPostAdd(@RequestParam String title,@RequestParam String anons,@RequestParam String full\_text, Model model) {

Post post=new Post(title,anons,full\_text);

postRepository.save(post);

return "redirect:/blog";

}

@GetMapping("/blog/{id}")

public String blogDetails(@PathVariable(value = "id") long id, Model model) {

if(!postRepository.existsById(id)){

return "redirect:/blog";

}

Optional<Post> post = postRepository.findById(id);

ArrayList<Post> res = new ArrayList<>();

post.ifPresent(res::add);

model.addAttribute("post",res);

return "blog-details";

}

@GetMapping("/blog/{id}/edit")

public String blogEdit(@PathVariable(value = "id") long id, Model model) {

if(!postRepository.existsById(id)){

return "redirect:/blog";

}

Optional<Post> post = postRepository.findById(id);

ArrayList<Post> res = new ArrayList<>();

post.ifPresent(res::add);

model.addAttribute("post",res);

return "blog-edit";

}

@PostMapping("/blog/{id}/edit")

public String blogPostUpdate(@RequestParam String title,@PathVariable(value = "id") long id,@RequestParam String anons,@RequestParam String full\_text, Model model) {

Post post=postRepository.findById(id).orElseThrow();

post.setTitle(title);

post.setAnons(anons);

post.setFull\_text(full\_text);

postRepository.save(post);

return "redirect:/blog";

}

@PostMapping("/blog/{id}/remove")

public String blogPostDelate(@PathVariable(value = "id") long id, Model model) {

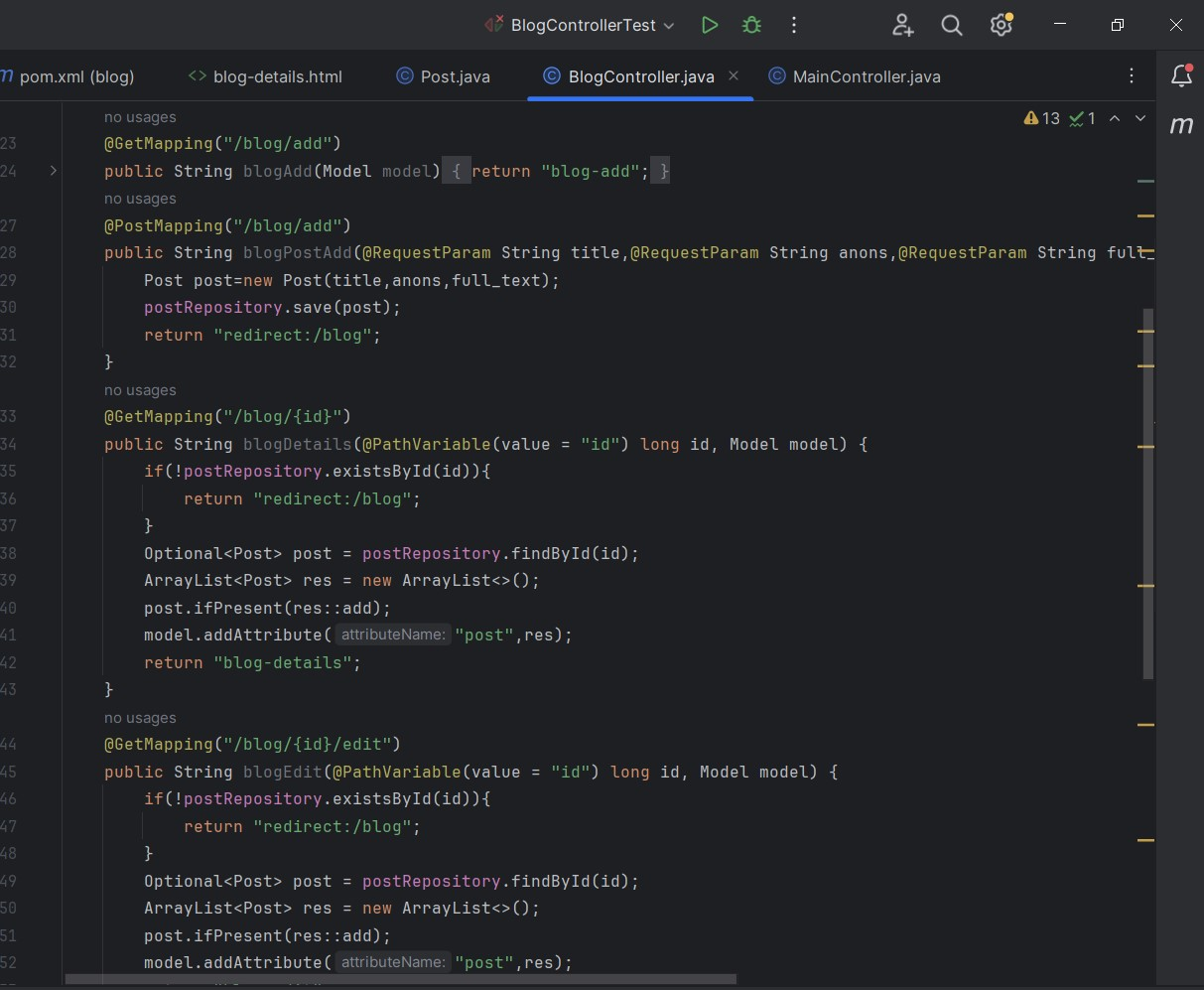
Post post=postRepository.findById(id).orElseThrow();

postRepository.delete(post);

return "redirect:/blog";

}

}

Рисунок 8 – Контроллер

**Вывод:** Научились создавать веб-приложения с реализацией MVC шаблона,

с помощью Java Spring.