Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт

Кафедра «Информатика»
кафедра

ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ № 4

Spring MVC
тема

Преподаватель _____ А.С. Кузнецов
подпись, дата инициалы, фамилия

Студент КИ20-16/1б, 031945012 _____ Ф.В. Пироженко
номер группы, зачетной книжки подпись, дата инициалы, фамилия

Красноярск 2022

СОДЕРЖАНИЕ

1 Цель

Познакомиться с шаблоном MVC в Spring и тем как он используется при создании web-приложений.

2 Задание

Изменить практическую работу №3 таким образом, чтобы она представляла собой web-приложение. Web-приложение должно иметь следующие страницы:

1) Главная страница, содержит приветствие и ссылки на другие (которые дублируют по функционалу пункты меню из работы №4).

2) Страница просмотра таблицы записей.

3) Страница добавления новой записи в таблицу.

4) Страница редактирования записи.

5) Страница удаления записи из таблицы БД.

6) Страница просмотра записей согласно некоторому критерию (аналогично пункту в работе №4).

Помимо всего должны быть осуществлены проверки (не менее двух) входных данных, сопровождающиеся соответствующими сообщениями об ошибках.

Вариант 20. Настольная игра

3 Ход выполнения

3.1 Реализация программы

Решение задания, представлено в листингах 1-15.

Листинг 1 – конфигурационный файл pom (pom.xml)

```
<?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 http://ma-
ven.apache.org/xsd/maven-4.0.0.xsd">
```

Продолжение листинга 1

```
  <modelVersion>4.0.0</modelVersion>

  <groupId>ru.sfu</groupId>
  <artifactId>Lab4_try2</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>war</packaging>

  <name>Lab4_try2 Maven Webapp</name>
  <!-- FIXME change it to the project's website -->
  <url>http://www.example.com</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.springframework/spring-core -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-core</artifactId>
      <version>5.3.10</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.springframework/spring-context -
->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
      <version>5.3.10</version>
    </dependency>
```

Продолжение листинга 1

```
    <!-- https://mvnrepository.com/artifact/org.springframework/spring-beans -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-beans</artifactId>
      <version>5.3.10</version>
    </dependency>

    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-jdbc</artifactId>
      <version>5.3.10</version>
    </dependency>

    <dependency>
      <groupId>org.springframework.data</groupId>
      <artifactId>spring-data-jpa</artifactId>
      <version>2.5.5</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.postgresql/postgresql -->
    <dependency>
      <groupId>org.postgresql</groupId>
      <artifactId>postgresql</artifactId>
      <version>42.2.18</version>
    </dependency>

    <dependency>

      <groupId>javax.annotation</groupId>
      <artifactId>javax.annotation-api</artifactId>
      <version>1.3.2</version>

    </dependency>

    <!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc --
>
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-webmvc</artifactId>
      <version>5.3.12</version>
```

Продолжение листинга 1

```
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.springframework/spring-web -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-web</artifactId>
      <version>5.3.12</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.thymeleaf/thymeleaf-spring5 -->
    <dependency>
      <groupId>org.thymeleaf</groupId>
      <artifactId>thymeleaf-spring5</artifactId>
      <version>3.0.12.RELEASE</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>javax.servlet-api</artifactId>
      <version>4.0.1</version>
      <scope>provided</scope>
    </dependency>



    <dependency>
      <groupId>org.hibernate.validator</groupId>
      <artifactId>hibernate-validator</artifactId>
      <version>6.0.10.Final</version>
    </dependency>

  </dependencies>


  <build>
    <finalName>Lab4_try2</finalName>
    <pluginManagement><!-- lock down plugins versions to avoid using Maven de-
faults (may be moved to parent pom) -->
      <plugins>
        <plugin>
```

Продолжение листинга 1

```
          <artifactId>maven-clean-plugin</artifactId>
          <version>3.1.0</version>
        </plugin>
        <!-- see http://maven.apache.org/ref/current/maven-core/default-bind-
ings.html#Plugin_bindings_for_war_packaging -->
        <plugin>
          <artifactId>maven-resources-plugin</artifactId>
          <version>3.0.2</version>
        </plugin>
        <plugin>
          <artifactId>maven-compiler-plugin</artifactId>
          <version>3.8.0</version>
        </plugin>
        <plugin>
          <artifactId>maven-surefire-plugin</artifactId>
          <version>2.22.1</version>
        </plugin>
        <plugin>
          <artifactId>maven-war-plugin</artifactId>
          <version>3.2.2</version>
        </plugin>
        <plugin>
          <artifactId>maven-install-plugin</artifactId>
          <version>2.5.2</version>
        </plugin>
        <plugin>
          <artifactId>maven-deploy-plugin</artifactId>
          <version>2.8.2</version>
        </plugin>
      </plugins>
    </pluginManagement>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <configuration>
          <source>8</source>
          <target>8</target>
        </configuration>
      </plugin>
```

Окончание листинга 1

```
    </plugins>
  </build>
</project>
```

Листинг 2 – Файл index (index.jsp)

```
<html>
<body>
<h2>Hello!</h2>
<a               href="/table_games">Begin               to               work</a>
</body>
</html>
```

Листинг 3 – Файл web (web.xml)

```
<!DOCTYPE web-app PUBLIC
 "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
 "http://java.sun.com/dtd/web-app_2_3.dtd" >

<web-app>
  <display-name>Archetype Created Web Application</display-name>
</web-app>
```

Листинг 4 – Файл add (add.html)

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>Add New Game</title>
</head>
<body>
<a href="/table_games">Back</a>
<form th:method="POST" th:action="@{/table_games}" th:object="${game}">
    <label for="gameName">Enter name: </label>
    <input type="text" th:field="*{gameName}" id="gameName"/>
    <div th:if="${#fields.hasErrors('gameName')}" th:errors="*{gameName}">ER-
ROR</div>
    <br/>

    <label for="price">Price </label>
    <input name="price" type="number" id="price"/>
    <div th:if="${#fields.hasErrors('price')}" th:errors="*{price}">ERROR</div>
```

Окончание листинга 4

```
    <br/>

    <label for="genre">Genre </label>
    <input type="text" th:field="*{genre}" id="genre"/>
    <div th:if="${#fields.hasErrors('genre')}" th:errors="*{genre}">ERROR</div>
    <br/>

    <label for="playerAmount">Player amount </label>
    <input name="playerAmount" type="number" id="playerAmount"/>
    <div th:if="${#fields.hasErrors('playerAmount')}" th:errors="*{playerA-
mount}">ERROR</div>
    <br/>

    <input type="submit" value="Add game"/>
</form>
</body>
</html>
```

Листинг 5 – Файл edit (edit.html)

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>Edit Game</title>
</head>
<body>
<a href="/table_games">Back</a>
<form th:method="POST" th:action="@{/table_games/edit/{id}(id=${game.getId()})}"
th:object="${game}">
    <label for="idx">ID</label>
    <input th:field="*{id}" type="number" id="idx" disabled/>
    <br/>

    <label for="gameName">Enter name: </label>
    <input type="text" th:field="*{gameName}" id="gameName"/>
    <div th:if="${#fields.hasErrors('gameName')}" th:errors="*{gameName}">ER-
ROR</div>
    <br/>

    <label for="price">Price </label>
    <input th:field="*{price}" type="number" id="price"/>
```

Окончание листинга 5

```
    <div th:if="${#fields.hasErrors('price')}" th:errors="*{price}">ERROR</div>
    <br/>

    <label for="genre">Genre </label>
    <input type="text" th:field="*{genre}" id="genre"/>
    <div th:if="${#fields.hasErrors('genre')}" th:errors="*{genre}">ERROR</div>
    <br/>

    <label for="playerAmount">Player amount </label>
    <input th:field="*{playerAmount}" type="number" id="playerAmount"/>
    <div th:if="${#fields.hasErrors('playerAmount')}" th:errors="*{playerA-
mount}">ERROR</div>
    <br/>

    <input type="submit" value="Edit game"/>
</form>

<form th:method="POST" th:action="@{/table_games/de-
lete/{id}(id=${game.getId()})}" th:object="${game}">
    <input type="submit" value="Delete Game"/>
</form>
</body>
</html>
```

Листинг 6 – Файл find (find.html)

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>Find</title>
</head>
<body>
<a href="/table_games">Back</a>
<form th:method="GET" th:action="@{/table_games/findGame}">
    <label for="price">Max price</label>
    <input type="number" id="price" name="price"/>
    <br/>
    <input type="submit" value="Find"/>
</form>
</body>
</html>
```

Листинг 7 – Файл menu (menu.html)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Menu</title>
</head>
<body>
<a href="/table_games/add">Add game</a><br/>
<a href="/table_games/show">Show all games</a><br/>
<a href="/table_games/find">Find games</a><br/>
</body>
</html>
```

Листинг 8 – Файл show (show.html)

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>Show</title>
</head>
<body>
    <a href="/table_games">Back</a>
    <div th:each="game : ${games}">
        <a th:href="@{{id}/edit(id=${game.getId()})}"
th:text="${game.toString()}">game</a>
    </div>
</body>
</html>
```

Листинг 9 – Файл showGame (showGame.html)

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>th:text="${game.getGameName()}"</title>
</head>
<body>
<a href="/table_games">Back</a>
<p th:text="${game.toString()} ">game</p>
</body>
</html>
```

Листинг 10 – Файл application (application.properties)

```
#--- Postgres ---
dataSource.driverClassName =org.postgresql.Driver
jpa.database=POSTGRESQL
dataSource.url=jdbc:postgresql://localhost:5432/lab3_db
dataSource.username=postgres
dataSource.password=qwerty
```

Листинг 11 – Класс DispServletInit(DispServInit.java)

```
package ru.sfu.config;

import org.springframework.web.servlet.support.AbstractAnnotationConfigDispatch-
erServletInitializer;

public class DispServletInit extends AbstractAnnotationConfigDispatcherServlet-
Initializer {

    @Override
    protected Class<?>[] getRootConfigClasses() {
        return null;
    }

    @Override
    protected Class<?>[] getServletConfigClasses() {
        return new Class[] {SpringConfig.class};
    }

    @Override
    protected String[] getServletMappings() {
        return new String[] {"/"};
    }
}
```

Листинг 12 – Класс SpringConfig (SpringConfig.java)

```
package ru.sfu.config;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.ApplicationContext;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.PropertySource;
import org.springframework.core.env.Environment;
```

Продолжение листинга 12

```
import org.springframework.jdbc.datasource.DriverManagerDataSource;

import javax.sql.DataSource;

import org.springframework.web.servlet.config.annotation.EnableWebMvc;
import org.springframework.web.servlet.config.annotation.ViewResolverRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;
import org.thymeleaf.spring5.SpringTemplateEngine;
import org.thymeleaf.spring5.templateresolver.SpringResourceTemplateResolver;
import org.thymeleaf.spring5.view.ThymeleafViewResolver;
import ru.sfu.model.TableGame;
import ru.sfu.dao.TableGameDAO;

@Configuration
@ComponentScan("ru.sfu.*")
@PropertySource("classpath:application.properties")
@EnableWebMvc
public class SpringConfig implements WebMvcConfigurer
{
    @Autowired
    private Environment env;

    @Bean
    DataSource dataSource() {
        DriverManagerDataSource dataSource = new DriverManagerDataSource();

        dataSource.setDriverClassName(env.getProperty("dataSource.driverClass-
Name"));
        dataSource.setUrl(env.getProperty("dataSource.url"));
        dataSource.setUsername(env.getProperty("dataSource.username"));
        dataSource.setPassword(env.getProperty("dataSource.password"));
        return dataSource;
    }

    @Bean
    TableGameDAO tableGameDAO() {
        TableGameDAO tableGameDAO =  new TableGameDAO();
        tableGameDAO.setDateSource(dataSource());
        return  tableGameDAO;
    }
```

Окончание листинга 12

```
    private final ApplicationContext applicationContext;

    @Autowired
    public SpringConfig(ApplicationContext applicationContext) {
        this.applicationContext = applicationContext;
    }

    @Bean
    public SpringResourceTemplateResolver templateResolver() {
        SpringResourceTemplateResolver templateResolver = new SpringResourceTem-
plateResolver();
        templateResolver.setApplicationContext(applicationContext);
        templateResolver.setPrefix("/WEB-INF/views/");
        templateResolver.setSuffix(".html");
        return templateResolver;
    }

    @Bean
    public SpringTemplateEngine templateEngine() {
        SpringTemplateEngine templateEngine = new SpringTemplateEngine();
        templateEngine.setTemplateResolver(templateResolver());
        templateEngine.setEnableSpringELCompiler(true);
        return templateEngine;
    }

    @Override
    public void configureViewResolvers(ViewResolverRegistry registry) {
        ThymeleafViewResolver resolver = new ThymeleafViewResolver();
        resolver.setTemplateEngine(templateEngine());
        registry.viewResolver(resolver);
    }

}
```

Листинг 13 – Класс TableGamesController (TableGamesController.java)

```
package ru.sfu.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
```

Продолжение листинга 13

```
import org.springframework.ui.Model;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.*;
import ru.sfu.dao.TableGameDAO;
import ru.sfu.model.TableGame;

import javax.validation.Valid;

@Controller
@RequestMapping("/table_games")
public class TableGamesController {
    private final TableGameDAO dao;

    @Autowired
    public TableGamesController(TableGameDAO dao) {
        this.dao = dao;
    }


    @GetMapping()
    public String menu(){
        return "table_games/menu";
    }

    @GetMapping("/show")
    public String showGames(Model model){
        model.addAttribute("games", dao.findAll());
        return "table_games/show";
    }

    @GetMapping("/add")
    public String newGame(Model model){
        model.addAttribute("game", new TableGame());
        return "table_games/add";
    }

    @PostMapping
    public String addGame(@ModelAttribute("game") @Valid TableGame game,
                          BindingResult bindingResult){
        if (bindingResult.hasErrors()){
```

Продолжение листинга 13

```
            return "table_games/add";
        }

        dao.insert(game);
        return "redirect:/table_games";
    }

    @GetMapping("/{id}")
    public String showGame(@PathVariable("id") int id, Model model){
        model.addAttribute("game", dao.get(id));
        return "table_games/showGame";
    }

    @GetMapping("/{id}/edit")
    public String editGame(@PathVariable("id") int id, Model model){
        model.addAttribute("game", dao.get(id));
        return "table_games/edit";
    }

    @PostMapping("edit/{id}")
    public String updateGame(@PathVariable("id") int id,
                             @ModelAttribute("game") @Valid TableGame game,
                             BindingResult bindingResult){
        if (bindingResult.hasErrors()){
            return "table_games/edit";
        }
        game.setId(id);
        dao.edit(game);
        return "redirect:/table_games";
    }

    @GetMapping("findGame")
    public String findGame(@RequestParam("price") int maxPrice, Model model)
    {
        model.addAttribute("games", dao.findGameBelowPrice(maxPrice));
        return "/table_games/show";
    }

    @GetMapping("find")
    public String showFoundedGames(Model model){
```

Окончание листинга 13

```
        return "table_games/find";
    }

    @GetMapping("delete/{id}")
    public String deleteGame(@PathVariable("id") int id){
        dao.delete(id);
        return "redirect:/table_games";
    }


}
```

Листинг 14 – Класс TableGameDAO (TableGameDAO.java)

```
package ru.sfu.dao;

import org.springframework.stereotype.Component;
import ru.sfu.model.TableGame;
import org.springframework.jdbc.core.BeanPropertyRowMapper;
import org.springframework.jdbc.core.JdbcTemplate;

import javax.sql.DataSource;
import java.util.List;


@Component
public class TableGameDAO  {
    JdbcTemplate jdbcTemplate;

    public void setDateSource(DataSource dataSource){
        this.jdbcTemplate = new JdbcTemplate(dataSource);
    }

    public List<TableGame> findAll() {
        return jdbcTemplate.query("select * from table_games ORDER BY ta-
ble_games.id",
                new BeanPropertyRowMapper<>(TableGame.class));
    }

    public void insert(TableGame tg){
        jdbcTemplate.update(
```

Окончание листинга 14

```
                "insert into table_games (gamename, price, playeramount, genre)
" +
                        "values(?, ?, ?, ?)",
                tg.getGameName(),
                tg.getPrice(),
                tg.getPlayerAmount(),
                tg.getGenre()
        );
    }

    public void edit(TableGame tg){
        jdbcTemplate.update(
                "update table_games " +
                        "set gamename=?, price=?, playeramount=?, genre=?" +
                        "where id=?",
                tg.getGameName(),
                tg.getPrice(),
                tg.getPlayerAmount(),
                tg.getGenre(),
                tg.getId());
    }

    public TableGame get(int id){
        return jdbcTemplate.query("SELECT * FROM table_games WHERE id=?", new
Object[]{id},
                new BeanPropertyRowMapper<>(TableGame.class))
                .stream().findAny().orElse(null);
    }

    public void delete(int id) {
        jdbcTemplate.update("delete from table_games where id=?", id);
    }

    public List<TableGame> findGameBelowPrice(int upperPrice){
        return jdbcTemplate.query(
                "select * from table_games where price<?",
                new Object[]{upperPrice},
                new BeanPropertyRowMapper<>(TableGame.class));
    }
}
```

Листинг 15 – Класс TableGame (TableGame.java)

```
package ru.sfu.model;
import javax.validation.constraints.*;
public class TableGame {
    int id;
    @NotNull(message = "should not be empty")
    @NotEmpty(message = "Name should not be empty")
    @Size(min = 3, max = 30, message="Name should  be between 3 and 30")
    String gameName;
    @Min(value = 1, message = "price should be greater than 0")
    @Max(value = 9000, message = "price should be less than 9000")
    int price;
    @NotNull
    @Min(value = 1, message = "player amount should be greater than 0")
    @Max(value = 20, message = "player amount should be less than 20")
    int playerAmount;
    @NotEmpty(message = "Genre should not be empty")
    @Size(min = 3, max = 30, message="Genre should  be between 3 and 30")
    String genre;
    public TableGame(){}
    public TableGame(int id, String gamename, int price, int playerAmount,
String genre) {
        this.id = id;
        this.gameName = gamename;
        this.price = price;
        this.playerAmount = playerAmount;
        this.genre = genre;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getGameName() {
        return gameName;
    }
```

Окончание листинга 15

```
    public void setGameName(String gameName) {
        this.gameName = gameName;
    }

    public int getPrice() {
        return price;
    }

    public void setPrice(int price) {
        this.price = price;
    }

    public int getPlayerAmount() {
        return playerAmount;
    }

    public void setPlayerAmount(int playerAmount) {
        this.playerAmount = playerAmount;
    }

    public String getGenre() {
        return genre;
    }

    public void setGenre(String genre) {
        this.genre = genre;
    }

    @Override
    public String toString(){
        return "id: " + id + ", name: " + gameName + ", price: " + price + ",
playersAmount: " + playerAmount + ", genre: " + genre;
    }
}
```

3.2 Демонстрация работы программы

Демонстрация проделанной работы представлена на рисунках 1-7.

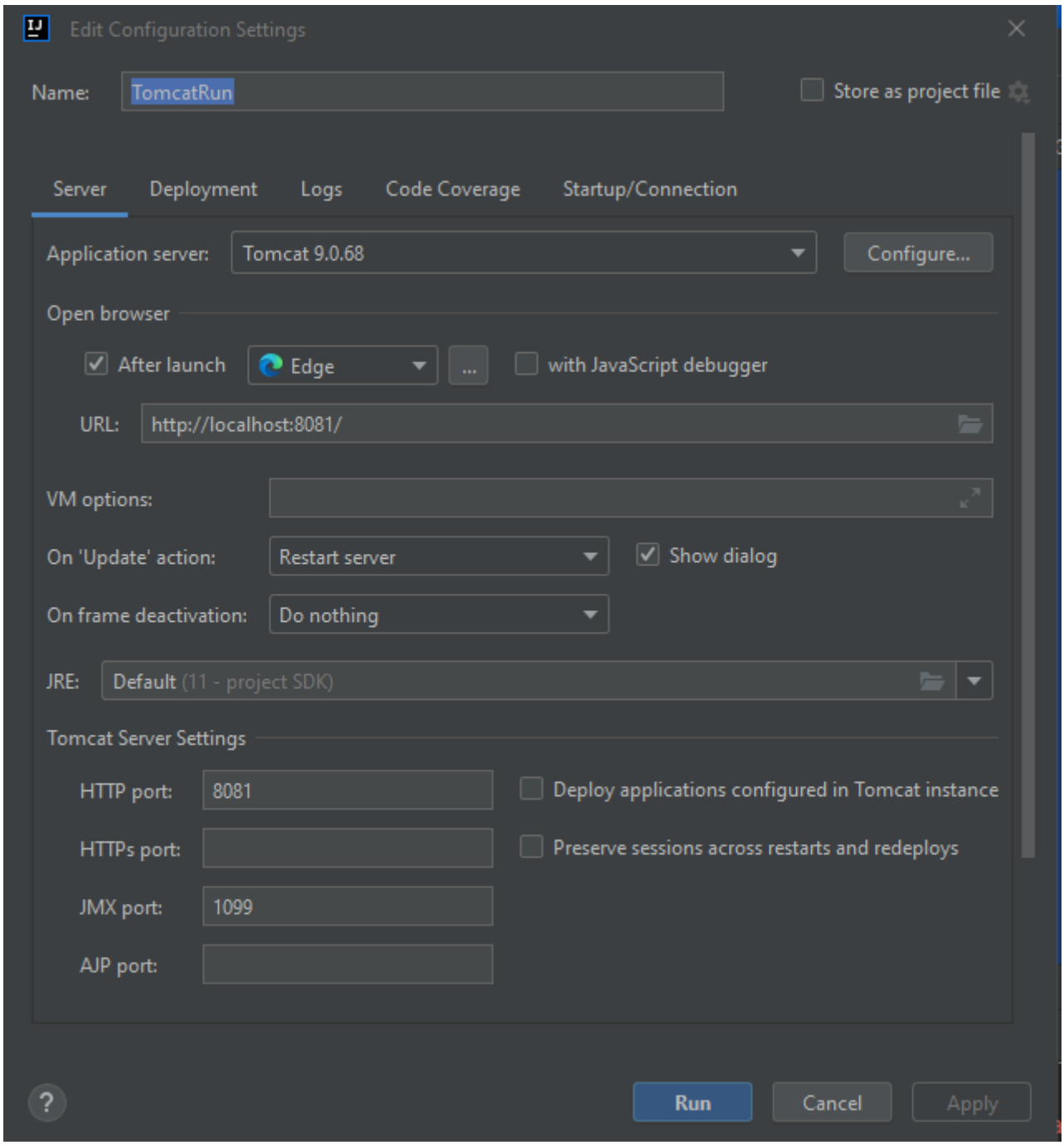

Рисунок 1 – Настройка Tomcat

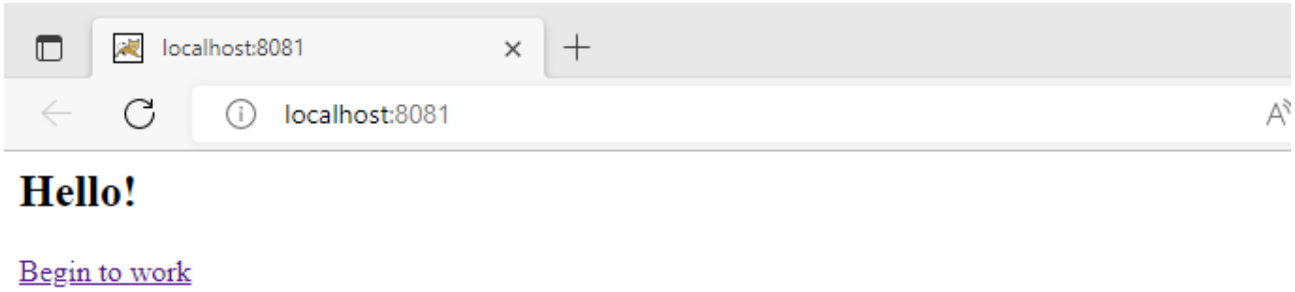

Рисунок 2 – Главная страница

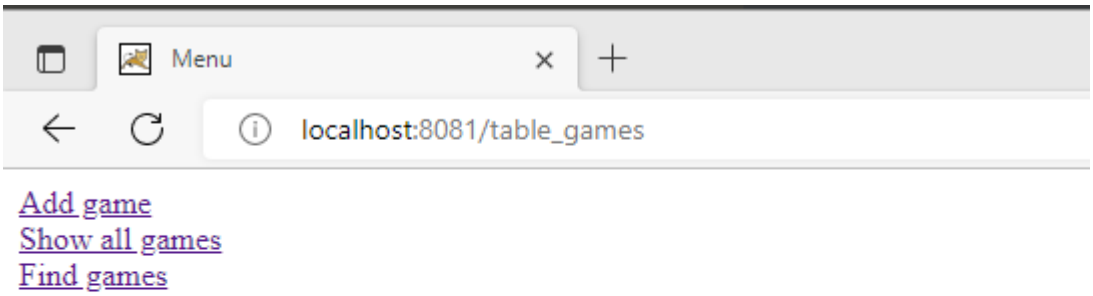

Рисунок 3 – Страница меню

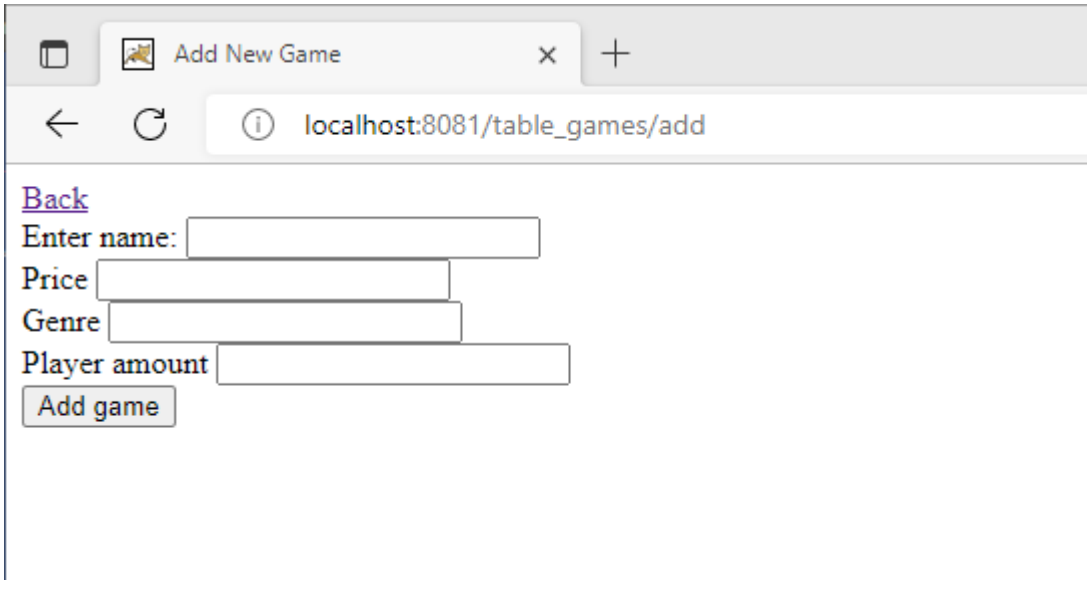

Рисунок 4 – Страница добавления новой записи

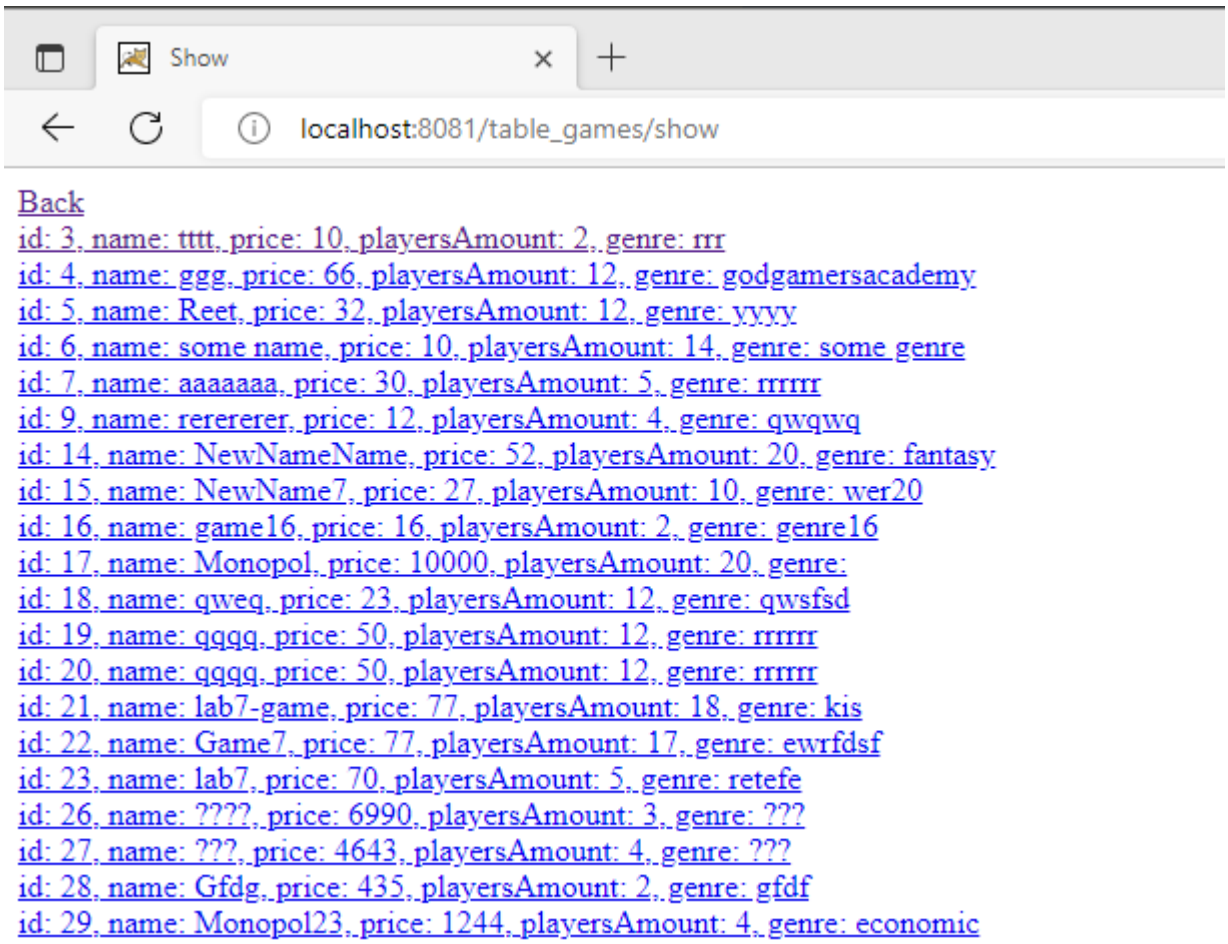

Рисунок 5 – Страница просмотра записей

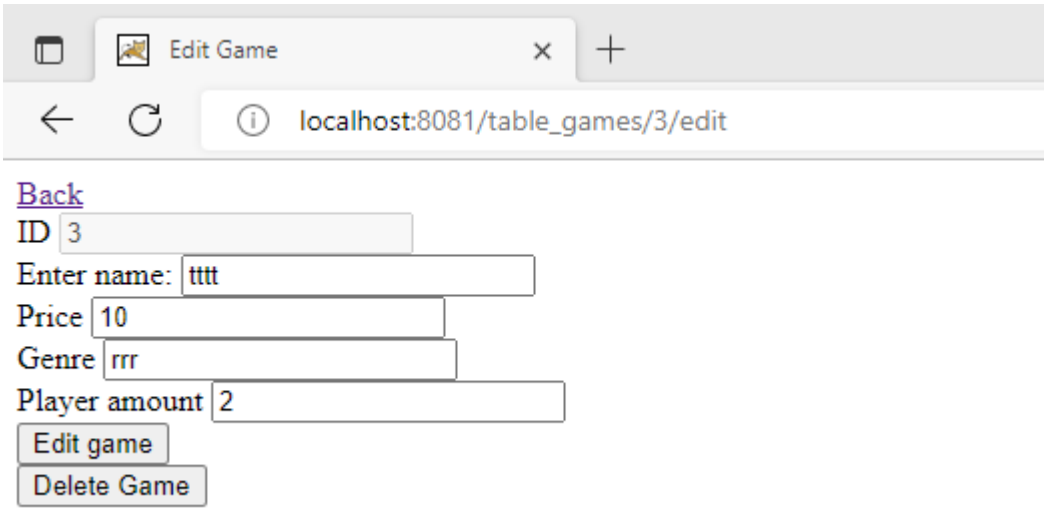

Рисунок 6 – Страница редактирования и удаления записи

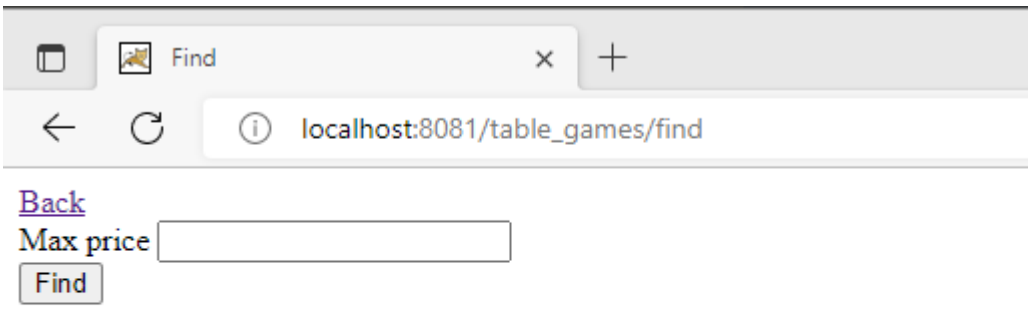

Рисунок 7 – Страница поиска записи по критерию максимальной цены

Вывод

В результате проделанной практической работы была выполнена следующая задача:

- были ознакомлены с шаблоном MVC в Spring и тем, как он используется при создании web-приложений.