**ЛАБОРАТОРНАЯ РАБОТА №8**

**РАЗРАБОТКА ЭФФЕКТИВНЫХ ВЕБ-ПРИЛОЖЕНИЙ**

**(SPRING MVC)**

**Цель работы:** разработка веб-приложений с реализацией шаблона MVC.

**Задание:**

1. создать проект Spring MVC и подключить к нему все необходимые зависимости (2 часа).
2. вывести на экран надпись.
3. создать дополнительный контроллер и представление (JSP) с отображением полей объекта класса из лабораторной работы №7 (Вариант 3).
4. прокомментировать код программы.

**Результат:**

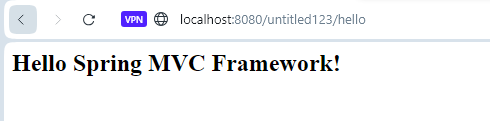


Рисунок 1 **– Результат заданий 1-2**

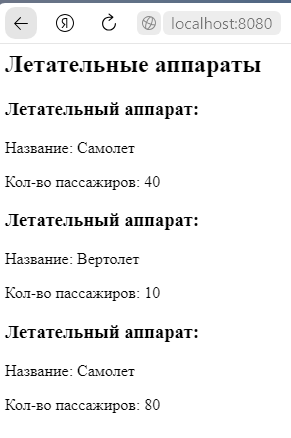


Рисунок 2 **– Результат задания 3**

**Листинг:**

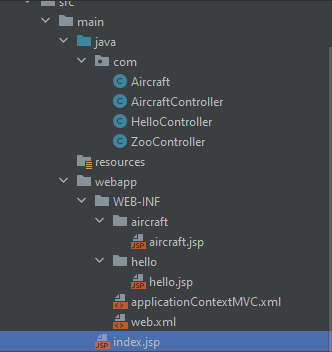


Рисунок 3 **– Структура приложения**

**Класс Aircraft:**

package com;  
  
class Aircraft { //класс летательный аппарат  
 private String name;  
 private double number;  
  
 public Aircraft (String name, double number) {  
 this.name = name;  
 this.number = number;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getNumber () {  
 return number;  
 }  
  
 public void setNumber (double number) {  
 this.number = number;  
 }  
}

Это модель данных для летательных аппаратов, включая его название и кол-во пассажиров.

**Класс AircraftController**

package com;  
  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.ModelMap;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
  
import java.util.ArrayList;  
import java.util.List;  
  
@Controller  
@RequestMapping("/aircraft")  
public class AircraftController {  
 @RequestMapping(method = RequestMethod.GET)  
 public String displayAircraft(ModelMap model) {  
 List<Aircraft> aircrafts = new ArrayList<Aircraft>();  
 aircrafts.add(new Aircraft ("Самолет", 40));  
 aircrafts.add(new Aircraft ("Вертолет", 10));  
 aircrafts.add(new Aircraft ("Самолет", 80));  
  
 model.addAttribute("aircrafts", aircrafts);  
  
 return "aircraft";  
 }  
}

Обрабатывает GET-запросы и создает список объектов **Aircraft.**

**Класс HelloController**

package com;  
  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.ModelMap;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
  
@Controller  
@RequestMapping("/hello")  
public class HelloController {   
 @RequestMapping(method = RequestMethod.*GET*)  
 public String printHello(ModelMap model) {  
 model.addAttribute("message", "Hello Spring MVC Framework!");  
 return "hello";  
 }  
}

**aircraft.jsp**

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>  
<%@ page contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>Летательные аппараты</title>  
</head>  
<body>  
 <h2>Летательные аппараты</h2>  
 <c:forEach items="${aircrafts}" var="aircraft">  
 <h3>Летательный аппарат:</h3>  
 <p>Название: ${aircraft.name}</p>  
 <p>Кол-во пассажиров: ${aircraft.number}</p>  
 </c:forEach>  
</body>  
</html>

JSP-страница для отображения информации о летательных аппаратах. Используется <c:forEach> для итерации по списку.

**hello.jsp**

<%@ page contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>Hello Spring MVC</title>  
</head>  
  
<body>  
<h2>${message}</h2>  
</body>  
</html>

**applicationContextMVC.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xmlns:context="http://www.springframework.org/schema/context"  
 xmlns:mvc="http://www.springframework.org/schema/mvc"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd http://www.springframework.org/schema/context https://www.springframework.org/schema/context/spring-context.xsd http://www.springframework.org/schema/mvc https://www.springframework.org/schema/mvc/spring-mvc.xsd">  
 <context:component-scan base-package="com"/>  
 <mvc:annotation-driven/>  
  
 <bean class = "org.springframework.web.servlet.view.InternalResourceViewResolver">  
 <property name = "prefix" value = "/WEB-INF/hello/" />  
 <property name = "prefix" value = "/WEB-INF/aircraft/" />  
 <property name = "suffix" value = ".jsp" />  
 </bean>  
</beans>

**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 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xmlns="http://xmlns.jcp.org/xml/ns/javaee"  
 xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee  
 http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"  
 id="WebApp\_ID" version="4.0">  
  
 <display-name>Archetype Created Web Application</display-name>  
  
 <servlet>  
 <servlet-name>dispatcher</servlet-name>  
 <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>  
 <init-param>  
 <param-name>contextConfigLocation</param-name>  
 <param-value>/WEB-INF/applicationContextMVC.xml</param-value>  
 </init-param>  
 <load-on-startup>1</load-on-startup>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>dispatcher</servlet-name>  
 <url-pattern>/</url-pattern>  
 </servlet-mapping>  
</web-app>

**Вывод:** разработали веб-приложений с реализацией шаблона MVC.