**МИНИСТЕРСТВО ТРАНСПОРТА И КОММУНИКАЦИЙ РЕСПУБЛИКИ БЕЛАРУСЬ**

**Учреждение образования**

**«Белорусский государственный университет транспорта»**

**Кафедра «Управление эксплуатационной работой и охрана труда**

**Отчёт  
по лабораторным работам**

по дисциплине «Средства и технологии анализа и разработки информационных систем»

Выполнил Проверил

Студент группы ГИ-31 м.т.н., ст.пр.

Козлова А.С. Козлов В. Г.

Гомель 2024

**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.
4. прокомментировать код программы.

**Результат:**

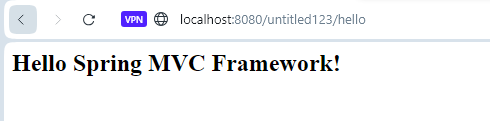


Рисунок 1 **– Результат заданий 1-2**

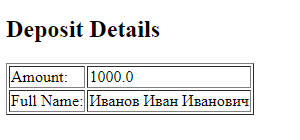


Рисунок 2 **– Результат задания 3**

**Листинг:**

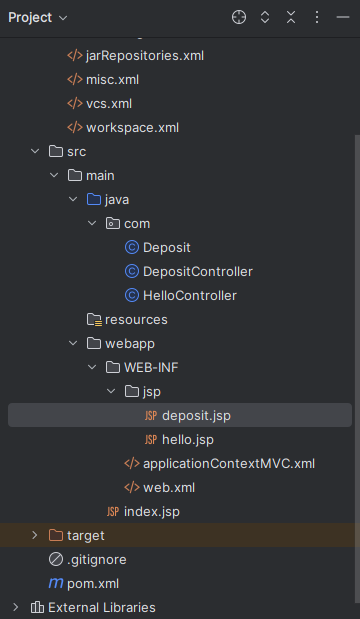


Рисунок 3 **– Структура приложения**

**Класс Deposit:**

package com;  
public class Deposit {  
 private double amount;  
 private String fullName;  
  
 public Deposit(double amount, String fullName) {  
 this.amount = amount;  
 this.fullName = fullName;  
 }  
  
 public double getAmount() {  
 return amount;  
 }  
  
 public String getFullName() {  
 return fullName;  
 }  
}

Это модель данных для вклада в банк, включая ФИО вкладчика и сумму вложения.

**Класс DepositController**

package com;  
  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.Model;  
import org.springframework.web.bind.annotation.GetMapping;  
  
@Controller  
public class DepositController {  
  
 @GetMapping("/deposit")  
 public String showDeposit(Model model) {  
 // Создаем новый объект Deposit для отображения  
 Deposit deposit = new Deposit(1000.0, "Иванов Иван Иванович");  
  
 // Передаем объект Deposit в представление  
 model.addAttribute("deposit", deposit);  
  
 // Возвращаем имя представления  
 return "deposit";  
 }  
}

Обрабатывает GET-запросы и создает список объектов Deposit.

**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";  
 }  
}

**deposit.jsp**

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  
<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="UTF-8">  
 <title>Deposit Details</title>  
</head>  
<body>  
 <h2>Deposit Details</h2>  
 <table border="1">  
 <tr>  
 <td>Amount:</td>  
 <td>${deposit.amount}</td>  
 </tr>  
 <tr>  
 <td>Full Name:</td>  
 <td>${deposit.fullName}</td>  
 </tr>  
  
 </table>  
</body>  
</html>

JSP-страница для отображения информации о сотрудниках. 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/jsp/" />  
 <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.