Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра экономической информатики

Лабораторная работа №2

**Работа с сессионными бинами Java EE**

Выполнил: Проверил:

Студент: гр.714301 Кунцевич А.А.

Александрова Е.И.

Минск 2020

**Цель –** разработать веб-ориентированное приложение на базе Java EE, позволяющее при помощи сессионных бинов управлять состоянием приложения (ЛР2).

**Задание:** разработать приложение на базе Java EE для управления информационной системой библиотек (статистика количества читателей). Использовать для организации логики работу с сессионными бинами.

**Текст программы:**

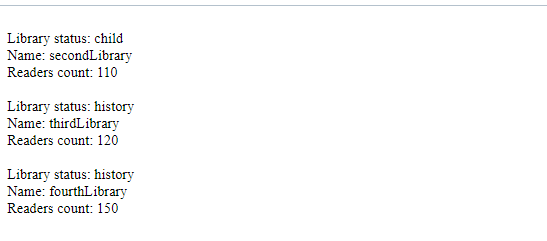
package com.aleksandrova.entities;  
  
import java.math.BigDecimal;  
import java.util.Objects;  
  
public class Library {  
  
 private Long id;  
  
 private String name;  
  
 private String status;  
  
 private BigDecimal readersCount;  
  
 public Library() {  
 }  
  
 public Library(Long id, String name, String status, BigDecimal readerCount) {  
 this.id = id;  
 this.name = name;  
 this.status = status;  
 this.readersCount = readerCount;  
 }  
  
 public Long getId() {  
 return id;  
 }  
  
 public void setId(Long id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getStatus() {  
 return status;  
 }  
  
 public void setStatus(String status) {  
 this.status = status;  
 }  
  
 public BigDecimal getReadersCount() {  
 return readersCount;  
 }  
  
 public void setReadersCount(BigDecimal readersCount) {  
 this.readersCount = readersCount;  
 }  
  
 @Override  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 if (o == null || getClass() != o.getClass()) return false;  
 Library that = (Library) o;  
 return id.equals(that.id) &&  
 name.equals(that.name) &&  
 status.equals(that.status) &&  
 readersCount.equals(that.readersCount);  
 }  
  
 @Override  
 public int hashCode() {  
 return Objects.*hash*(id, name, status, readersCount);  
 }  
  
 @Override  
 public String toString() {  
 return "Id: " + id +  
 "\nType: " + name +  
 "\nTrainNumber: " + status +  
 "\nCost: " + readersCount +"\n";  
 }  
}

package com.aleksandrova.sessions;  
  
import com.aleksandrova.entities.Library;  
  
import javax.ejb.Singleton;  
import java.math.BigDecimal;  
import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.List;  
import java.util.stream.Collectors;  
  
import static java.util.Comparator.*comparing*;  
  
@Singleton  
public class Libraries {  
  
 static ArrayList<Library> *libraries* = new ArrayList<>(  
 Arrays.*asList*(  
 new Library(1L, "firstLibrary", "scince", new BigDecimal("100")),  
 new Library(2L, "secondLibrary", "child", new BigDecimal("110")),  
 new Library(3L, "thirdLibrary", "history", new BigDecimal("120")),  
 new Library(4L, "fourthLibrary", "history", new BigDecimal("150"))  
 )  
 );  
  
 public List<Library> getLibrariesByCount() {  
 *libraries*.sort(*comparing*(Library::getReadersCount));  
 List<Library> foundLibraries = *libraries*.stream()  
 .skip(1)  
 .collect(Collectors.*toList*());  
  
 return foundLibraries;  
 }  
  
 public List<Library> getLibrariesByStatus(String status) {  
 List<Library> foundLibraries = new ArrayList<>();  
 for (Library libraries : *libraries*) {  
 if (libraries.getStatus().compareTo(status) == 0) {  
 foundLibraries.add(libraries);  
 }  
 }  
  
 return foundLibraries;  
 }  
  
}

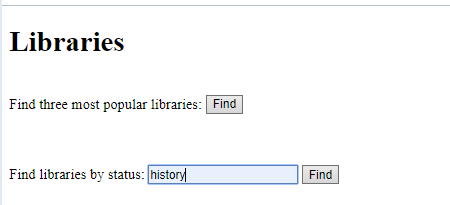
package com.aleksandrova.sessions;  
  
import com.aleksandrova.entities.Library;  
  
import javax.ejb.EJB;  
import javax.servlet.ServletException;  
import javax.servlet.annotation.WebServlet;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import java.io.IOException;  
import java.io.PrintWriter;  
import java.util.List;  
  
@WebServlet(urlPatterns = {"/library-servlet"})  
public class LibraryServlet extends HttpServlet {  
  
 @EJB  
 private Libraries libraries;  
  
 @Override  
 protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
  
 String button = req.getParameter("button");  
 if (button.equals("count\_btn")) {  
 List<Library> librariesList = libraries.getLibrariesByCount();  
 doResp(resp, librariesList);  
 }  
 if (button.equals("status\_btn")) {  
 List<Library> librariesList = libraries.getLibrariesByStatus(req.getParameter("status"));  
 doResp(resp, librariesList);  
 }  
 }  
  
 private void doResp(HttpServletResponse resp, List<Library> libraries) throws ServletException, IOException {  
 PrintWriter writer = resp.getWriter();  
 writer.println("<!DOCTYPE html>\n" +  
 "<html lang=\"en\">\n" +  
 "<head>\n" +  
 " <meta charset=\"UTF-8\"/>\n" +  
 " <link rel=\"stylesheet\" href=\"style.css\"/>\n" +  
 " <title>Libraries</title>\n" +  
 "</head>\n" +  
 "<body>\n");  
 if (!libraries.isEmpty()) {  
 for (Library library : libraries) {  
 writer.println("<br>" +  
 "Library status: " + library.getStatus()  
 + "<br> Name: " + library.getName()  
 + "<br> Readers count: " + library.getReadersCount()  
 + "<br/>"+  
 "</body>\n" +  
 "</html>");  
 }  
 } else {  
 writer.println("Libraries not found!" +  
 "<br/>\n" +  
 " <br/>\n" +  
 " Find libraries by status:\n" +  
 " <input name=\"status\"/>\n" +  
 " <button type=\"submit\" name=\"button\" value=\"status\">Find</button>\n" +  
 " <br/>\n" +  
 " <br/>\n" +  
 " <br/>\n" +  
 "</body>\n" +  
 "</html>");  
 }  
  
  
 }  
}

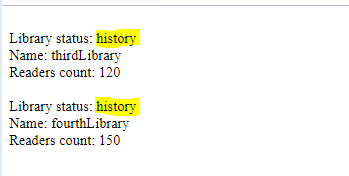
**Скриншоты работы:**



Работа кнопки поиска трёх наиболее популярных библиотек: 

Работа кнопки поиска по статусу библиотеки:





Реакция на пустой ввод:  
