Министерство образования и науки Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение науки и высшего образования**

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

**Институт информационных технологий и радиотехники**

**(ИИТР)**

Кафедра информационных систем и программной инженерии

**Лабораторная работа № 06\_2**

**по дисциплине**

**«Отображение таблиц и использование библиотек компонентов в JSF-приложениях»**

Выполнил:

ст. гр. ПРИ-117

Хлызова В.Г.

Принял:

Трифонов Д.А.

Владимир, 2019

ЦЕЛЬ РАБОТЫ

Изучить расширенные возможности каркаса JavaServer Faces для разработки веб-приложений: отображение данных в виде таблиц, использование сторонних библиотек компонентов; получить навыки разработки JSF-приложения, ориентированного на табличную обработку данных.

ВЫПОЛНЕНИЕ РАБОТЫ

В данной лабораторной работе было разработано приложение, состоящее из двух jsf-страниц: страница с выводом таблицы и различными операциями с данными (удаление строки, обновление статусов у выбранных пользователей, сортировка по дате, числу постов и id и возможностью изменить запись выбранной строки) и страница с изменением данных. Была использована библиотека компонентов PrimeFaces.

Класс пользователя:

**import** java.util.Date;  
  
**public class** User {  
   
 **final int** id;  
 **private** String login;  
 **private** Date dateRegistration;  
 **private** String state;  
 **private int** numberPost;  
 **private boolean** check;  
   
 **public** User(**int** id, String login,   
 Date dateRegistration, String state, **int** numberPost){  
 **this**.id=id;  
 **this**.login=login;   
 **this**.dateRegistration=dateRegistration;  
 **this**.state=state;  
 **this**.numberPost=numberPost;  
 }  
  
 **public** String getLogin() {  
 **return** login;  
 }  
  
 **public void** setLogin(String login) {  
 **this**.login = login;  
 }  
  
  
 **public** Date getDateRegistration() {  
 **return** dateRegistration;  
 }  
  
 **public void** setDateRegistration(Date dateRegistration) {  
 **this**.dateRegistration = dateRegistration;  
 }  
  
 **public int** getNumberOrder() {  
 **return** numberPost;  
 }  
  
 **public void** setNumberOrder(**int** numberPost) {  
 **this**.numberPost = numberPost;  
 }  
  
 **public** String getState() {  
 **return** state;  
 }  
  
 **public void** setState(String state) {  
 **this**.state = state;  
 }  
  
 **public int** getId() {  
 **return** id;  
 }  
  
 **public boolean** isCheck() {  
 **return** check;  
 }  
  
 **public void** setCheck(**boolean** check) {  
 **this**.check = check;  
 }  
}

Управляемый бин:

**import** java.text.SimpleDateFormat;  
**import** java.util.ArrayList;  
**import** java.util.Arrays;  
**import** java.util.Date;  
**import** java.util.List;  
  
**public class** UserBean{

//создаём список, который будет отражаться в таблице.  
 **private static** List<User> users = **new** ArrayList<User>(Arrays.asList(  
 **new** User(1, **"Иванов"**, **new** Date(2019, 5, 15), **"online"**, 5),  
 **new** User(2, **"Семёнов"**, **new** Date(2008, 5, 2), **"ofline"**, 1),  
 **new** User(3, **"Булгаков"**, **new** Date(2017, 1, 1), **"ofline"**, 5),  
 **new** User(4, **"Митина"**, **new** Date(2009, 8, 5), **"online"**, 2)  
 ));

//получение всех пользователей из списка  
 **public** List<User> getUsers(){  
 **return** users;  
   
 }

//удаление пользователя  
 **public** String deleteUser(**int** idUser){  
 **try** {  
 **int** n=-1;  
 **for** (User u: users){  
 n++;  
 **if** (u.id==idUser) users.remove(n);  
 }  
 }  
 **catch** (Exception e){  
 **return "index"**;  
 }  
 **return "index"**;  
 }

//Пользователь который изменяется  
 **private** User editUser;  
   
 **public** User getEditUser() {  
 **return** editUser;  
 }  
   
 **public void** setEditUser(User editUser) {  
 **this**.editUser = editUser;  
 }

//переход на страницу изменений с передачей данных о пользователе  
 **public** String edit(User user){  
 setEditUser(user);  
 **return "edit"**;  
 }

//изменение данных о пользователе и вывод новых данных в таблицу. Возращаемся на страницу index.  
 **public** String editGo(){  
 **int** n=-1, id=0;  
 **for** (User u: users){  
 n++;  
 **if** (u.id==editUser.id){  
 id=n;  
 **break**;  
 }  
 }  
 users.set(id,editUser);  
 **return "index"**;  
 }

//изменение статусов поьзователей  
 **public** String upgrateState(){  
 **for** (User u: users){  
 **if** (u.isCheck()) u.setState(**"online"**);  
 **else** u.setState(**"ofline"**);  
 }  
 **return "index"**;  
 }

//далее идут сортировки по количеству постов пользователя, id и дате  
 **public** String sortPost(){  
 List<User> preSort = users;  
 List<User> postSort = **new** ArrayList<User>(); **int** min=100, id=0, n;  
   
 **try** {  
 **while** (!preSort.isEmpty()){  
 n = -1; min=100;  
 **for** (User u: preSort){  
 n++;  
 **if** (u.getNumberOrder()<min){  
 min=u.getNumberOrder();  
 id=n;  
 }  
 }  
 postSort.add(preSort.get(id));  
 preSort.remove(id);  
 }users=postSort;  
 }**catch** (Exception e){  
 **return "index"**;  
 }  
 **return "index"**;  
 }  
   
 **public** String sortId(){  
 List<User> preSort = users;  
 List<User> postSort = **new** ArrayList<User>(); **int** min=10, id=0, n;  
   
 **try** {  
 **while** (!preSort.isEmpty()){  
 n = -1; min=10;  
 **for** (User u: preSort){  
 n++;  
 **if** (u.getId()<min){  
 min=u.getId();  
 id=n;  
 }  
 }  
 postSort.add(preSort.get(id));  
 preSort.remove(id);  
 }users=postSort;  
 }**catch** (Exception e){  
 **return "index"**;  
 }  
 **return "index"**;  
 }  
   
 **public** String sortDate(){  
 List<User> preSort = users;  
 List<User> postSort = **new** ArrayList<User>(); **long** max=0; **int** id=0, n;  
 SimpleDateFormat simpleDateFormat = **new** SimpleDateFormat(**"yyyy-MM-dd"**);  
   
 **try** {  
 **while** (!preSort.isEmpty()){  
 n = -1; max=0;  
 **for** (User u: preSort){  
 n++;  
 **if** (u.getDateRegistration().getTime()/(24\*60\*60\*1000)>max){  
 max=u.getDateRegistration().getTime()/(24\*60\*60\*1000);  
 id=n;  
 }  
 }  
 postSort.add(preSort.get(id));  
 preSort.remove(id);  
 }users=postSort;  
 }**catch** (Exception e){  
 **return "index"**;  
 }  
 **return "index"**;  
 }  
}

Страница с таблицей:

*<?***xml version='1.0' encoding='UTF-8'** *?>***<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"*>***<**html xmlns="http://www.w3.org/1999/xhtml"  
 xmlns:h="http://xmlns.jcp.org/jsf/html"  
 xmlns:p="http://primefaces.org/ui"  
 xmlns:f="http://xmlns.jcp.org/jsf/core"  
 xmlns:ui="http://xmlns.jcp.org/jsf/facelets"**>  
 <**h:head**>  
 <**title**>Таблица пользователей</**title**>  
 </**h:head**>  
 <**h:body**>  
 <**h:form id="form"**>

//выводим таблицу пользователей. Всех пользователей получаем методом **getUsers()**   
 <**p:dataTable id="userTable" var="\_user"   
 value="#{userBean.getUsers()}"   
 rowKey="#{\_user.id}"  
 style="width**: 850**px**;**"**>  
 //заголовок таблицы  
 <**f:facet name="header"**>  
 <**h:outputText value="ТАБЛИЦА ПОЛЬЗОВАТЕЛЕЙ"**/>  
 </**f:facet**>  
  
 <**p:column**>

//cheсkBox для выбора пользователей статус которых надо обновить   
 <**h:selectBooleanCheckbox id="chkRememberMe" value="#{\_user.check}"** />  
 </**p:column**>  
 //вывод данных в колонку  
 <**p:column headerText="Id"**>  
 <**h:outputText value="#{\_user.id}"** />  
 </**p:column**>  
   
 <**p:column headerText="Логин"**>  
 <**h:outputText value="#{\_user.login}"** />  
 </**p:column**>  
   
 <**p:column headerText="Дата регистрации"**>  
 <**h:outputText value="#{\_user.dateRegistration}"**>  
 <**f:convertDateTime type="date" dateStyle="medium"** />  
 </**h:outputText**>  
 </**p:column**>  
   
 <**p:column headerText="Кол-во постов"**>  
 <**h:outputText value="#{\_user.numberOrder}"** />  
 </**p:column**>  
   
 <**p:column headerText="Статус"**>  
 <**h:outputText value="#{\_user.state}"** />  
 </**p:column**>  
   
 <**p:column**>

//удаление и изменение  
 <**h:commandButton value="Удалить" action="#{userBean.deleteUser(\_user.id)}"**/>  
 <**h:commandButton value="Изменить" action="#{userBean.edit(\_user)}"**/>  
 </**p:column**>  
 //подвал страницы  
 <**f:facet name="footer"**>

//вызов метдов на обновление статусов и сортировку  
 <**h:commandButton value="Обновить статусы" action="#{userBean.upgrateState()}"**/>  
 <**h:commandButton value="Сортировка по id" action="#{userBean.sortId()}"**/>  
 <**h:commandButton value="Сортировка по дате" action="#{userBean.sortDate()}"**/>  
 <**h:commandButton value="Сортировка по кол-ву постов" action="#{userBean.sortPost()}"**/>  
 </**f:facet**>  
 </**p:dataTable**>   
 </**h:form**>  
  
 </**h:body**>  
</**html**>

Страница с редактированием строки:

*<?***xml version='1.0' encoding='UTF-8'** *?>***<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"*>***<**html xmlns="http://www.w3.org/1999/xhtml"  
 xmlns:h="http://xmlns.jcp.org/jsf/html"  
 xmlns:p="http://primefaces.org/ui"  
 xmlns:f="http://xmlns.jcp.org/jsf/core"**>  
 <**h:head**>  
 <**title**>Изменить данные</**title**>  
 </**h:head**>  
 <**h:body**>  
 <**h:form**>  
 <**h:outputText value="ВЫ МОЖЕТЕ ИЗМЕНИТЬ СЛЕДУЮЩИЕ ДАННЫЕ:" style="font-size**: 20**px**;**"**/>  
 <**br**/>  
 <**h:panelGrid columns="2" style="font-size**: 20**px**;**"**>  
 <**h:outputText value="Логин:"**/>  
 <**p:inputText value="${userBean.editUser.login}"** />

//дата регистрации выводится с помощью компонента календаря  
 <**h:outputText value="Дату регистрации:"**/>  
 <**p:calendar value="${userBean.editUser.dateRegistration}" showOn="button" pattern="dd.MM.yyyy"**>  
 <**f:convertDateTime type="date" dateStyle="medium"** />  
 </**p:calendar**>  
 <**h:outputText value="Кол-во постов:"**/>

//количество постов выводится компонентов спинер(можно не вводить циферки а накручивать в большую или меньшую сторону)  
 <**p:spinner value="${userBean.editUser.numberOrder}"**/>  
 <**h:commandButton value="Назад" action="index"**/>  
 <**h:commandButton value="Принять изменения" action="${userBean.editGo()}"**/>  
 </**h:panelGrid**>  
 </**h:form**>  
 </**h:body**>  
</**html**>

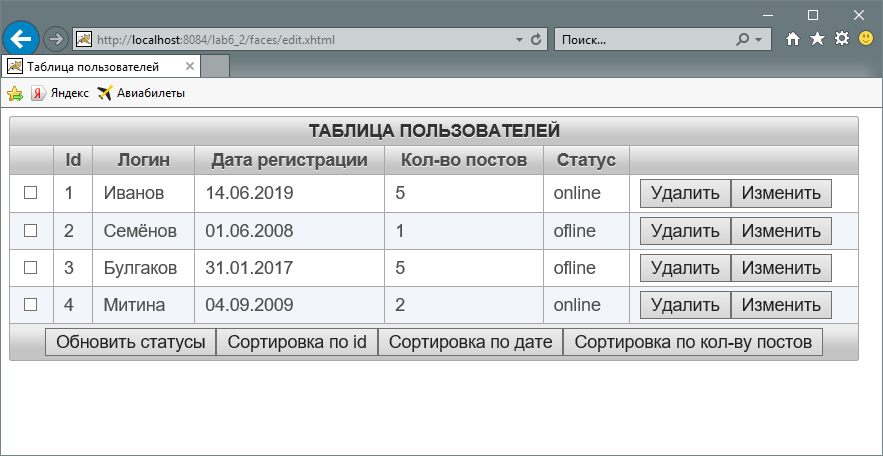


Рисунок 1. Отображение таблицы.

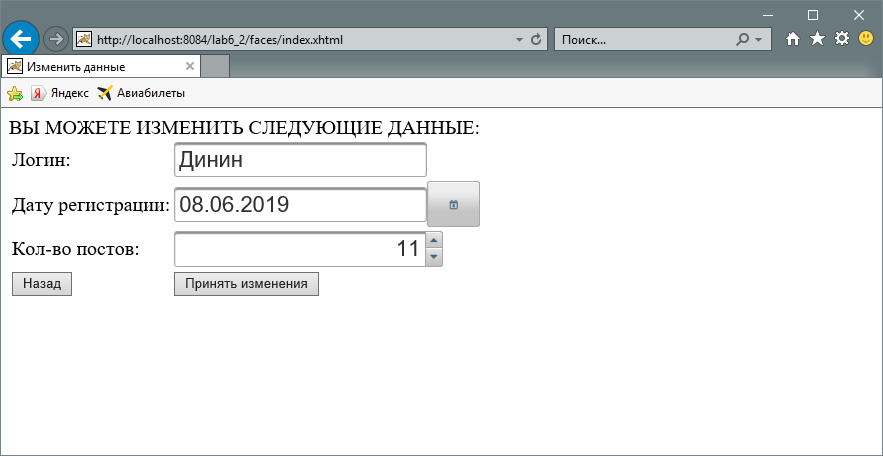


Рисунок 2. Страница редактирования.

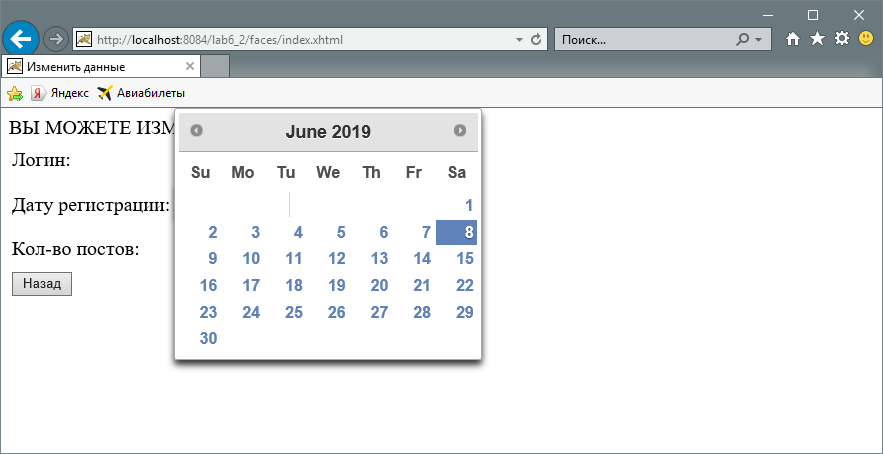


Рисунок 3. Использование компонента календарь.

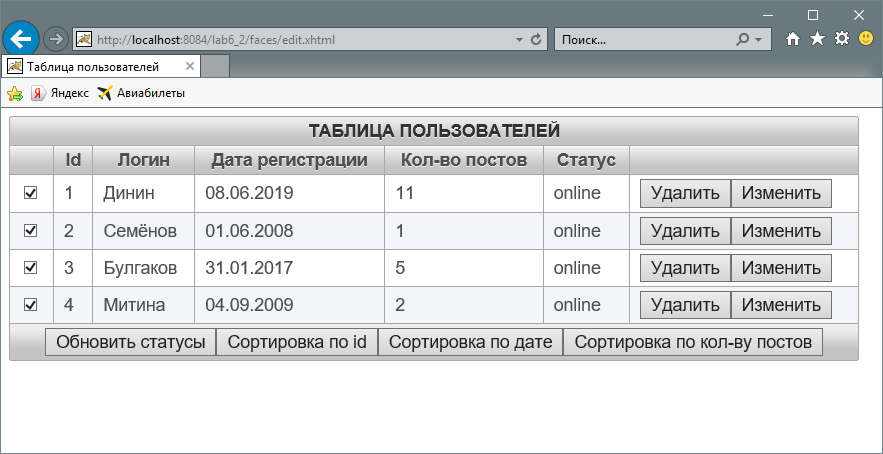


Рисунок 4. Выполнение редактирования.

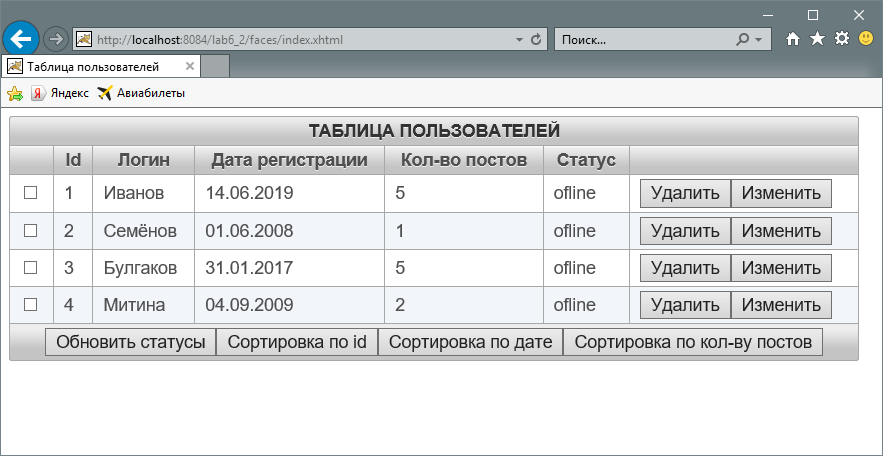


Рисунок 5. Обновление статуса у списка пользователей.

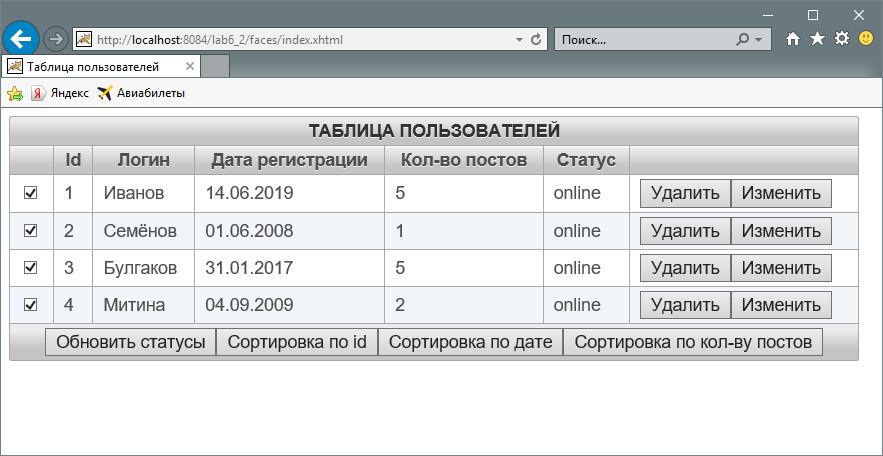


Рисунок 6. Обновление статусов у списка пользователей.

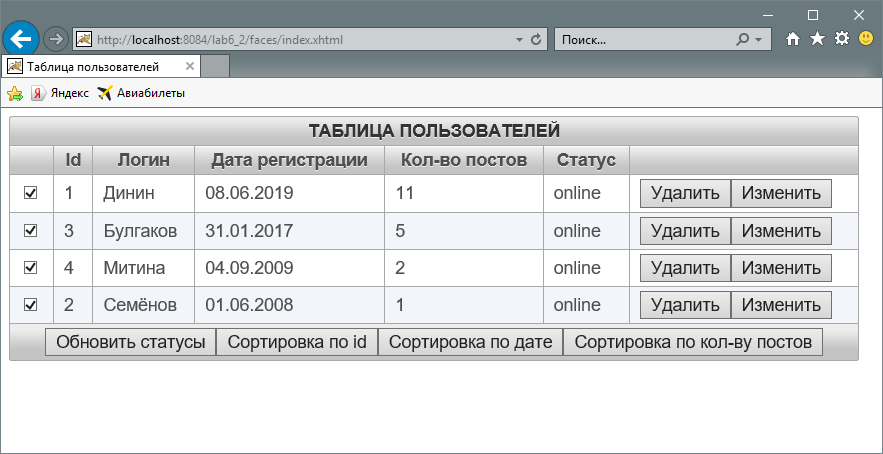


Рисунок 7. Сортировка по дате.

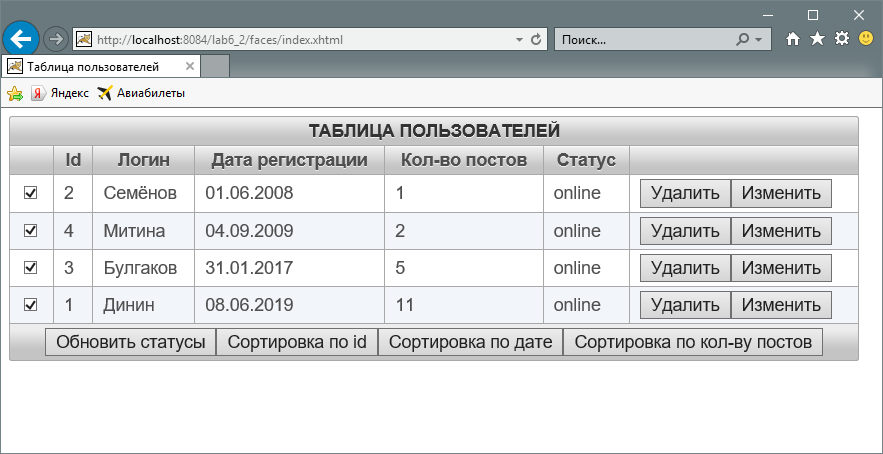


Рисунок 8. Сортировка по кл-ву постов.

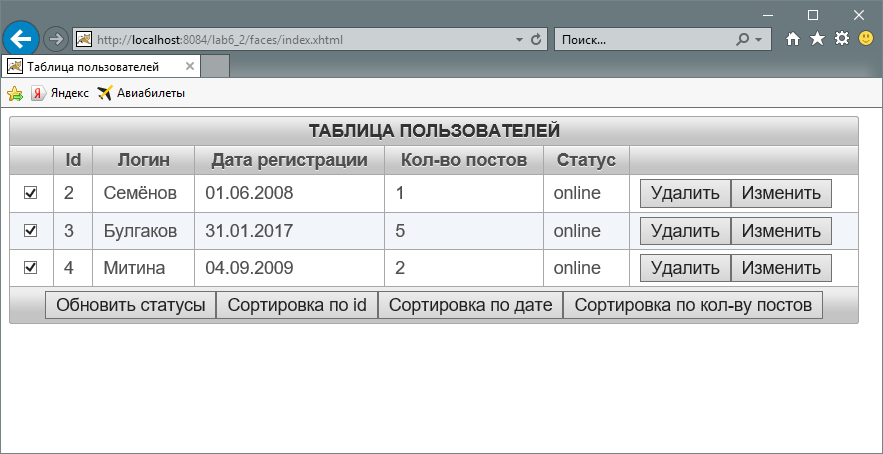


Рисунок 9. Удаление.

ВЫВОД

В ходе лабораторной работы были изучены расширенные возможности каркаса JavaServer Faces для разработки веб-приложений: отображение данных в виде таблиц, использование сторонних библиотек компонентов; получены навыки разработки JSF-приложения, ориентированного на табличную обработку данных.