Частное учреждение образования

«Колледж бизнеса и права»

Тема: «Работа с файлами, JList, JScrollPane, HashMap. Создание запускного

jar-файла.»

Учащийся М.Н. Гончаров.

группа: Т-992

2022

Задание

Переделать пример со списком, чтобы в нем хранился список группы,

а выбрав студента, в другом поле появлялись его данные (возраст, адрес).

// Main

import java.awt.\*;  
import java.awt.event.\*;  
import java.io.\*;  
import java.util.HashMap;  
import javax.imageio.ImageIO;  
import javax.swing.\*;  
import javax.swing.event.ListSelectionEvent;  
import javax.swing.event.ListSelectionListener;  
  
class ListWork extends JFrame{  
 JPanel panel1=new JPanel();  
 private static HashMap<String, String> *myHash* = new HashMap<String, String>();  
 public static void main(String[] args) {  
 ListWork window= new ListWork("Work with List");  
 window.setDefaultCloseOperation(*EXIT\_ON\_CLOSE*);  
 window.setVisible(true);  
 window.pack();  
 window.setMinimumSize(window.getSize());  
 }  
 public ListWork(String s){  
 super(s); //задаем название окна  
 DefaultListModel myListModel = new DefaultListModel(); //создаем модель для  
  
 for (String key: *myHash*.keySet()) {  
 myListModel.addElement(key);  
 }  
  
 final JList myList = new JList(); //создаем объект, отвечающий за вид  
 JScrollPane myScroll = new JScrollPane(myList); //создаем  
 myList.setModel(myListModel); //задаем для списка  
 Box myBox1 = new Box(BoxLayout.*Y\_AXIS*);//создаем компоновку  
 final JTextField myText = new JTextField();//текстовое поле для ввода значений  
 myBox1.add(myText); //добавляем поле на компоновку  
 Box box1=new Box(BoxLayout.*X\_AXIS*);//создаем еще одну  
 JButton button1=new JButton("Add in List");  
 box1.add(button1); //добавляем кнопку на  
 button1.addActionListener(new ActionListener() {//создаем слушатель,  
 public void actionPerformed(ActionEvent e) {//отвечающий за добавление  
 String data = myText.getText();  
 String[] words = data.split(";");  
 *myHash*.put(words[0], words[0]);  
 System.*out*.println(*myHash*);  
 myListModel.addElement(words[0]);  
 }  
 });  
 JButton button2 = new JButton("Delete in list"); //создаем кнопку  
 button2.addActionListener(new ActionListener() {//создаем слушатель, отвечающий  
 public void actionPerformed(ActionEvent e) {//за удаление элементов списка  
 while (myListModel.contains(myText.getText())){ //равных значению  
 myListModel.removeElement(myText.getText()); //в текстовой строке  
 }  
 }  
 });  
 box1.add(button2); //добавляем кнопку на компоновку  
 JButton buttonClear=new JButton("Clear List");//еще кнопка для очистки  
 buttonClear.addActionListener(new ActionListener() {//к ней слушател  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 myListModel.clear(); //очищаем список  
 }  
 });  
 box1.add(buttonClear); //добавляем кнопку на компоновку  
 myBox1.add(box1);  
 //вставляем компоновку в другую компоновку  
 add(myScroll,BorderLayout.*CENTER*);//добавляем панель с прокруткой в центр  
 add(myBox1,BorderLayout.*NORTH*); //добавляем компоновку в верхнюю часть окна\  
  
 Box myBox2=new Box(BoxLayout.*X\_AXIS*); //новая компоновка  
 JButton button3=new JButton("Save...");//кнопка для сохранения списка  
 myBox2.add(button3);  
 final FileDialog fdlg=new FileDialog(this, "");//создаем диалоговое окно для чтения  
 button3.addActionListener(new ActionListener() { //слушатель для сохранениpublic  
 public void actionPerformed(ActionEvent e) {  
 fdlg.setMode(FileDialog.*SAVE*); //делаем созданный диалог диалогом сохранения  
 fdlg.setTitle("Save File"); //задаем ему заголовок  
 fdlg.setVisible(true);  
 FileWriter myWriter = null;//создаем объект типа FileWriter и приравниваем его к  
 try { //секция, в которой можно выполнять небезопасные действия созданному  
 myWriter=new FileWriter(fdlg.getDirectory()+fdlg.getFile());  
 BufferedWriter myBWriter=new BufferedWriter(myWriter);  
 for(int i=0;i<myListModel.getSize();i++){ //в цикле сохраняем каждый элемент в  
 myBWriter.write(""+myListModel.getElementAt(i)+"; "+*myHash*.get(myListModel.getElementAt(i)));//используя  
 myBWriter.newLine(); //и вставляем символ перехода на новую  
 }  
 myBWriter.close();//закрываем все соединения myWriter.close();  
 } catch (IOException e1) {  
 e1.printStackTrace(); //если произойдет ошибка, будет выведено сообщение }  
 }  
 }});  
  
 myBox2.add(Box.*createHorizontalGlue*()); //вставляем «пружину», чтобы кнопки  
 JButton button4 = new JButton("Upload...");//кнопка для загрузки списка из файла  
 button4.addActionListener(new ActionListener() { //слушатель для загрузки из файла  
 public void actionPerformed(ActionEvent e) {  
 fdlg.setMode(FileDialog.*LOAD*); //делаем созданный диалог диалогом загрузки  
 myListModel.clear();  
 fdlg.setTitle("Upload File");//задаем ему заголовок  
 fdlg.setVisible(true); //делаем видимым  
 FileReader myReader = null;  
 try{  
 File file=new File(fdlg.getDirectory()+fdlg.getFile());  
 FileReader fr=new FileReader(file);  
 BufferedReader reader=new BufferedReader(fr);  
 String line= reader.readLine();  
 while(line != null){  
 String[] temp=line.split(";");  
 *myHash*.put(temp[0],temp[1]);  
 line = reader.readLine();  
 }  
 }  
 catch (FileNotFoundException es) {  
 es.printStackTrace();  
 }  
 catch (IOException ez) {  
 ez.printStackTrace();  
 }  
 for (String key: *myHash*.keySet()) {  
 myListModel.addElement(key);  
 }  
 }  
 });  
 myBox2.add(button4);//добавляем кнопку на компоновку  
 add(myBox2,BorderLayout.*SOUTH*);//вставляе  
 JLabel label2 = new JLabel("Information:");  
 label2.setPreferredSize(new Dimension(250,10));  
 Box centerBox=new Box(BoxLayout.*X\_AXIS*); //создаем компоновку  
 centerBox.add(myScroll); //вставляем в компоновку панель с прокруткой  
 centerBox.add(label2); //и панель для изображений  
 add(centerBox,BorderLayout.*CENTER*);  
 myList.addListSelectionListener(new ListSelectionListener() {//слушатель выделения  
 public void valueChanged (ListSelectionEvent e){  
 JList tempList = (JList) e.getSource();  
 label2.setText(*myHash*.get(myList.getSelectedValue().toString()));  
 System.*out*.println();  
 }  
 });  
 }  
 public void ShowData(){ //функция для загрузки изображения на панель  
 }  
  
}

\\ output

