package tert;  
  
import java.awt.\*;  
import java.awt.event.\*;  
import java.io.\*;  
import javax.swing.\*;  
  
  
public class ListWork extends JFrame {  
 public static void main(String[] args) {  
 ListWork window = new ListWork("Работа со списком");  
 window.setVisible(true);  
 window.pack();  
 window.setMinimumSize(window.getSize());  
 }  
  
 public ListWork(String s) {  
 super(s); //задаем название окна  
 final DefaultListModel myListModel = new DefaultListModel(); //создаем модель для нашего  
 for (int i = 0; i < 10; i++) {//списка и заполняем ее элементами, модифкатор final делает  
 myListModel.addElement("" + i); //объект доступным для вложенных слушателей  
 }  
 final JList myList = new JList(); //создаем объект, отвечающий за вид нашего списка  
 JScrollPane myScroll = new JScrollPane(myList); //создаем панель с прокруткой  
 myList.setModel(myListModel); //задаем для списка созданную модель  
 Box myBox1 = new Box(BoxLayout.*Y\_AXIS*);//создаем компоновку  
 final JTextField myText = new JTextField();//текстовое поле для ввода значений элементов  
 myBox1.add(myText); //добавляем поле на компоновку  
 Box box1 = new Box(BoxLayout.*X\_AXIS*);//создаем еще одну компоновку  
 JButton button1 = new JButton("Добавить в список"); //создаем кнопку  
 box1.add(button1); //добавляем кнопку на компоновку  
 button1.addActionListener(new ActionListener() {//создаем слушатель,  
 public void actionPerformed(ActionEvent e) {//отвечающий за добавление  
 myListModel.addElement(myText.getText()); //элемента списка  
 }  
 });  
 JButton button2 = new JButton("Убрать из списка"); //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("Очистить список");//еще кнопка для очистки списка  
 buttonClear.addActionListener(new ActionListener() {//к ней слушатель  
 @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 окна  
 add(myBox1, BorderLayout.*NORTH*); //добавляем компоновку в верхнюю часть окна  
 Box myBox2 = new Box(BoxLayout.*X\_AXIS*); //новая компоновка  
 JButton button3 = new JButton("Сохранить...");//кнопка для сохранения списка  
 myBox2.add(button3);  
 final FileDialog fdlg = new FileDialog(this, "");//создаем диалоговое окно для чтения и записи //файла  
 button3.addActionListener(new ActionListener() { //слушатель для сохранения  
 public void actionPerformed(ActionEvent e) {  
 fdlg.setMode(FileDialog.*SAVE*); //делаем созданный диалог диалогом сохранения  
 fdlg.setTitle("Сохранить файл"); //задаем ему заголовок  
 fdlg.setVisible(true); //делаем видимым  
 FileWriter myWriter = null;//создаем объект типа FileWriter и приравниваем его к null  
 try { //секция, в которой можно выполнять небезопасные действия созданному объекту типа FileWriter задаем новый объект с параметрами каталога и файла, //выбранного пользователем в диалоге сохранения файла  
 myWriter = new FileWriter(fdlg.getDirectory() + fdlg.getFile());  
//создаем объект типа BufferedWriter, соединяя его с созданным объектом myWriter  
 BufferedWriter myBWriter = new BufferedWriter(myWriter);  
 for (int i = 0; i < myListModel.getSize(); i++) { //в цикле сохраняем каждый элемент в файл  
 myBWriter.write("" + myListModel.getElementAt(i));//используя  
 myBWriter.newLine(); //и вставляем символ перехода на новую строку  
 }  
 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 были по краям окна  
 JButton button4 = new JButton("Загрузить...");//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  
 fdlg.setTitle("Загрузить файл");//задаем ему заголовок  
 fdlg.setVisible(true); //делаем видимым  
 FileReader myReader = null;  
 try { //секция, в которой можно выполнять небезопасные действия  
// созданному объекту типа FileReader задаем новый объект с параметрами каталога и файла, //выбранного пользователем в диалоге загрузки файла  
 myReader = new FileReader(fdlg.getDirectory() + fdlg.getFile());  
 myListModel.clear(); //очищаем список, т.к. в него будут помещены новые данные  
//создаем объект типа BufferedReader, соединяя его с созданным объектом myReader  
 BufferedReader myBReader = new BufferedReader(myReader);  
 String s; //строка для временного хранения данных  
//в s записываем строку из файла, и если она не пустая (а пустой она будет, если файл закончился //или пустой), то добавляем в список новый элемент с параметром s  
 while ((s = myBReader.readLine()) != null) {  
 myListModel.addElement(s);  
 }  
 myBReader.close();//закрываем все соединения myReader.close();  
 } catch (IOException e1) {  
 e1.printStackTrace();  
 }  
 }  
 });  
 myBox2.add(button4);//добавляем кнопку на компоновку  
 add(myBox2, BorderLayout.*SOUTH*);//вставляем компоновку в нижнюю область окна  
  
 }  
}