Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №4

по дисциплине: "Разработка кроссплатформенных приложений"

на тему: "Работа с файлами"

Выполнили:

Студенты группы 21ВВB1

Кирьянов В.Е.

Аляев А.О.

Приняли

К.т.н. Юрова О.В.

К.т.н. Карамышева Н.С.

Пенза 2024

**Название работы:** «Работа с файлами».

**Цель работы:** изучить работу с файлами и механизмы сериализации данных.

**Задание на лабораторную работу**

Модифицировать приложение из предыдущей лабораторной работы, реализовав сохранение в файл и загрузку данных из файла. Предусмотреть сохранение данных, как в текстовом виде, так и в двоичном (с использованием механизма сериализации). Для этого нужно добавить 4 кнопки для сохранения и загрузки в текстовом и двоичном виде соответственно. Кроме того, в программе нужно предусмотреть использование стандартного диалога открытия файла (JFileChooser). Оформление лабораторной работы должно быть выполнено в соответствии с требованиями, приведенными в Приложении 2.

Ход работы:

Реализовали сохранение в файл и загрузку данных из файла, сохранение данных реализовано, как в текстовом, так и в двоичном форматах.

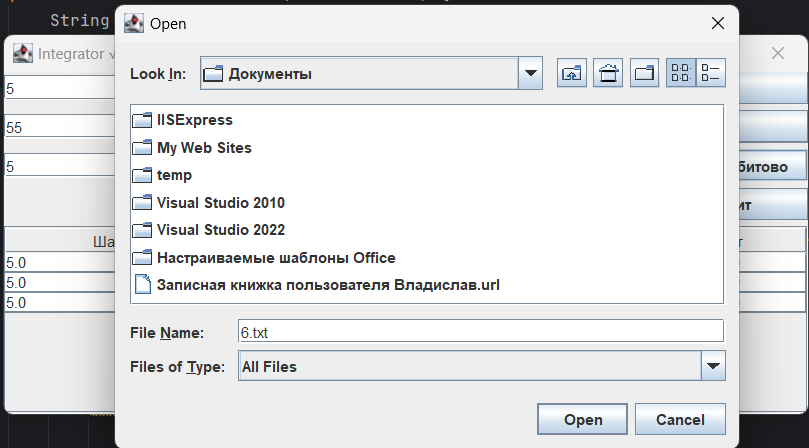
Листинг программы.

Class NewForm.

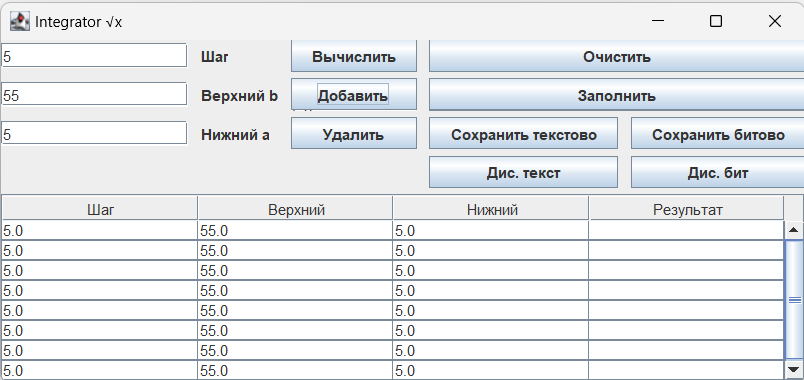
import javax.swing.\*;  
import javax.swing.table.DefaultTableModel;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.io.\*;  
import java.util.LinkedList;  
import java.util.Scanner;  
  
public class NewForm {  
 public LinkedList<RecIntegral> TableList = new LinkedList<>();  
 private JPanel panel1;  
 private JTextField textField1;  
 private JTextField textField2;  
 private JTextField textField3;  
 private JButton CalcButton;  
 private JButton AddButton;  
 private JButton DelButton;  
 private JTable table1;  
 private JScrollPane scrollPane1;  
 private JButton clearTableButton;  
 private JButton addTableButton;  
 private JButton DisBitButton;  
 private JButton DisTextButton;  
 private JButton SaveBitButton;  
 private JButton SaveTextButton;  
  
 DefaultTableModel model = (DefaultTableModel) table1.getModel();  
 //String[] columnNames = {"Шаг", "Верхний", "Нижний", "Результат"};  
 //String[][] data = { };  
 public NewForm() {  
 CalcButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 try {  
 double a = 0; //down  
 double b = 0; //upp  
 double step = 0; //step  
 double res = 0;  
 for (int row = 0; row < table1.getRowCount(); row++) {  
 if (table1.getValueAt(row, 0) != "" || table1.getValueAt(row, 1) != "" || table1.getValueAt(row, 2) != "") {  
 step = Double.*parseDouble*(table1.getValueAt(row, 0).toString());  
 b = Double.*parseDouble*(table1.getValueAt(row, 1).toString());  
 a = Double.*parseDouble*(table1.getValueAt(row, 2).toString());  
 res = Main.*Integral*(a, b, step);  
 table1.setValueAt(res, row, 3);  
 //TableList.equals(new Object[]{String.valueOf(step), String.valueOf(b), String.valueOf(a), String.valueOf(res)});  
 RecIntegral temp = TableList.get(row);  
 temp.res = String.*valueOf*(res);  
 TableList.set(row, temp);  
  
  
 //for(RecIntegral iter: TableList) {  
 //iter.getComp(iter.step, iter.b, iter.a, String.valueOf(res));  
 //}  
 }  
 }  
 } catch (Exception ex){ System.*out*.printf("Ошибка! %s", ex);}  
 }  
 });  
 AddButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 try {  
 double text1 = Double.*parseDouble*(textField1.getText());  
 double text2 = Double.*parseDouble*(textField2.getText());  
 double text3 = Double.*parseDouble*(textField3.getText());  
  
 if ((text1 < 0.000001) || (text1 > 1000000) || (text2 < 0.000001) || (text2 > 1000000) || (text3 < 0.000001) || (text3 > 1000000)) {  
 throw new Invalid("Число вне диапазона [0.000001-1000000]");  
 }  
 else if (text2 < text3) {  
 throw new Invalid("Верхний предел меньше нижнего");  
 }  
 else if (text1 > text2 - text3){  
 throw new Invalid("Шаг превышает дозволенность");  
 }  
 model.addRow(new Object[]{text1, text2, text3});  
 RecIntegral obj = new RecIntegral(String.*valueOf*(text1), String.*valueOf*(text2), String.*valueOf*(text3), "");  
 TableList.add(obj);  
 } catch (Invalid ex) { //?  
 } catch (NumberFormatException ex) {  
 JOptionPane.*showMessageDialog*(null, "Ошибка парсинга текста в Double", "Ошибка", JOptionPane.*ERROR\_MESSAGE*);  
 }  
 }  
 });  
 DelButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 final int row = table1.getSelectedRow();  
 //final int column = table1.getSelectedColumn();  
 //for(int i = 0; i < 4; i++) {  
 //table1.setValueAt("", row, i);  
 try {  
 model.removeRow(row);  
 TableList.remove(row);  
 //}  
 } catch (Exception ex){}  
 }  
 });  
 clearTableButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 //TableList.clear();  
 model.getDataVector().removeAllElements();  
 model.fireTableDataChanged();  
 }  
 });  
 addTableButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 model.setRowCount(0);  
 //model.removeRow(model.getRowCount()-1);  
 //model.rowsRemoved(new DefaultTableModel(data,  
 //columnNames));  
 //System.out.println(TableList);  
 for(RecIntegral step: TableList){  
 //RecIntegral dt = new RecIntegral(step);  
 //RecIntegral Obj = new RecIntegral(step);  
 model.addRow(new Object[]{step.step,step.b,step.a,step.res});  
 //RecIntegral A = step.getStep();  
 //String value = new Object[]{step}.toString();  
 //model.addRow(new Object[]{value}  
 //model.addRow(step.getStep(), step.getB(), step.getA(), step.get());  
 //step.getStep();  
 }  
 }  
 });  
 SaveTextButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 String path = "";  
 JFileChooser fileChooser = new JFileChooser();  
 int returnVal = fileChooser.showOpenDialog(fileChooser);  
  
 if(returnVal == JFileChooser.*APPROVE\_OPTION*) {  
 path = fileChooser.getSelectedFile().getPath();  
 }  
 try {  
 FileWriter fw = new FileWriter(path);  
 for(RecIntegral step: TableList) {  
  
 fw.write(step.step +" "+ step.b +" "+ step.a +" "+ step.res + "\n");  
 }  
 fw.close();  
 } catch (IOException ex) {  
 throw new RuntimeException(ex);  
 }  
 }  
 });  
 DisTextButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 String path = "";  
 JFileChooser fileChooser = new JFileChooser();  
 int returnVal = fileChooser.showOpenDialog(fileChooser);  
 if(returnVal == JFileChooser.*APPROVE\_OPTION*) {  
 path = fileChooser.getSelectedFile().getPath();  
 }  
 try {  
  
 FileReader fr = new FileReader(path);  
 BufferedReader br = new BufferedReader(fr);  
 TableList.clear();  
  
 String line = br.readLine();  
 while (line != null)  
 {  
 String[] strs = line.trim().split("\\s+");  
 if(strs.length != 3)  
 TableList.add(new RecIntegral(strs[0], strs[1], strs[2], strs[3]));  
 else  
 TableList.add(new RecIntegral(strs[0], strs[1], strs[2], ""));  
 line = br.readLine();  
 }  
  
 } catch (IOException ex) {  
 throw new RuntimeException(ex);  
 }  
 }  
 });  
 SaveBitButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 String path = "";  
 JFileChooser fileChooser = new JFileChooser();  
 int returnVal = fileChooser.showOpenDialog(fileChooser);  
 if(returnVal == JFileChooser.*APPROVE\_OPTION*) {  
 path = fileChooser.getSelectedFile().getPath();  
 }  
 try {  
 FileOutputStream outputStream = new FileOutputStream(path);  
 ObjectOutputStream objectOutputStream = new ObjectOutputStream(outputStream);  
 for(RecIntegral step: TableList) {  
 //Ser ser = new Ser(step.step, step.b, step.a, step.res);  
 //RecIntegral rec = new RecIntegral(step.step, step.b, step.a, step.res);  
 objectOutputStream.writeObject(step);  
 }  
 objectOutputStream.close();  
 outputStream.close();  
 JOptionPane.*showMessageDialog*(null, "Успешно сохранено", "Ура", JOptionPane.*INFORMATION\_MESSAGE*);  
 } catch (IOException ex) {  
 throw new RuntimeException(ex);  
 }  
 }  
 });  
 DisBitButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 String path = "";  
 JFileChooser fileChooser = new JFileChooser();  
 int returnVal = fileChooser.showOpenDialog(fileChooser);  
 if(returnVal == JFileChooser.*APPROVE\_OPTION*) {  
 path = fileChooser.getSelectedFile().getPath();  
 }  
 try {  
 FileInputStream fileInputStream = new FileInputStream(path);  
 ObjectInputStream objectInputStream = new ObjectInputStream(fileInputStream);  
 TableList.clear();  
  
 //while (objectInputStream.readObject() != null) {  
  
 //RecIntegral obj = (RecIntegral) objectInputStream.readObject();  
 while (true) {  
  
 TableList.add((RecIntegral) objectInputStream.readObject());  
 }  
 //objectInputStream.close();  
 } catch (EOFException ex){  
  
 }  
  
 catch (IOException | ClassNotFoundException ex) {  
 throw new RuntimeException(ex);  
 }  
 }  
 });  
 }  
 public static void main(String[] args) {  
 JFrame frame = new JFrame("Integrator √x");  
 frame.setLocationRelativeTo(null);  
 frame.setContentPane(new NewForm().panel1);  
 //frame.setContentPane(new NewForm().scrollPane1);  
 frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 frame.pack();  
 frame.setVisible(true);  
 }  
 private void createUIComponents() {  
  
 scrollPane1 = new JScrollPane(table1);  
 //scrollPane1.max  
 scrollPane1.setPreferredSize(new Dimension(0, 150));  
  
 String[] columnNames = {"Шаг", "Верхний", "Нижний", "Результат"};  
 String[][] data = { };  
  
 table1 = new JTable(data, columnNames);  
 //table1.setBounds(30, 40, 200, 300);  
 table1.setModel(new DefaultTableModel(data,  
 columnNames  
 ) {public boolean isCellEditable(int row, int column){return (column == 0 || column == 1 || column == 2);}}  
 );  
 }  
}

Результаты программы.

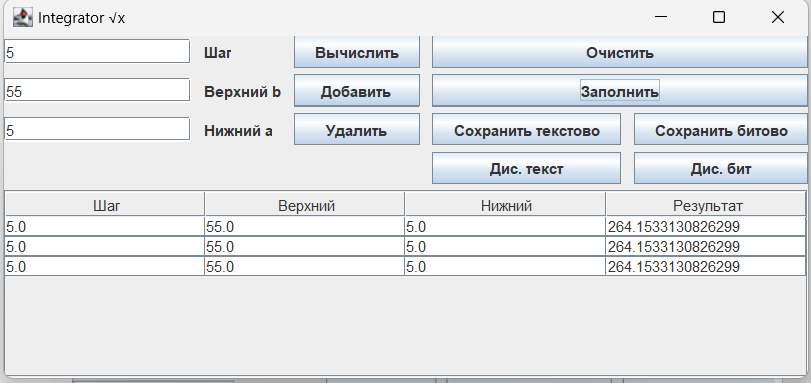
Сохранили данные в выбранном формате.



Заполнили текущий лист другими данными.



Выбрали файл через диалоговое окно и восстановили данные.



Вывод: изучили работу с файлами и механизмы сериализации данных.