Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**по курсу «Программирование на языке JAVA»

по лабораторной работе №4

на тему: «Работа с файлами»  
Вариант 10.

Выполнил студент группы 20ВВП1:

Кодиров И. Н.

Приняла:

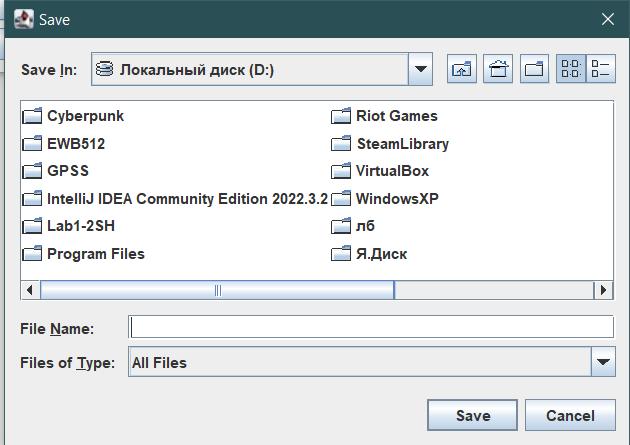
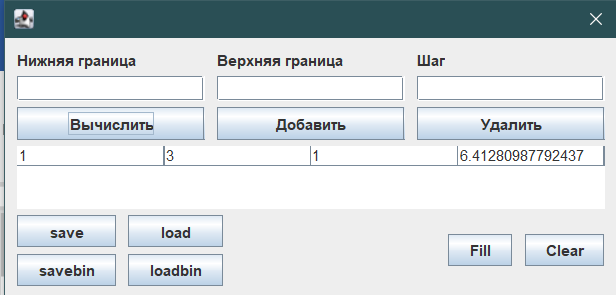
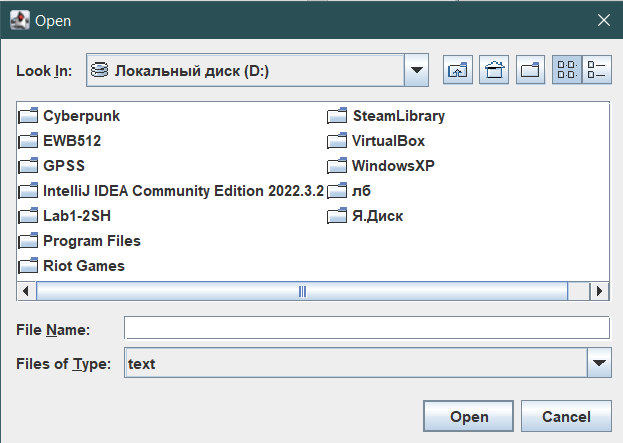
Юрова О. В.

Пенза 2023

**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.

**Результаты выполнения программы:**

**Листинг:**  
import javax.swing.\*;  
import javax.swing.filechooser.FileNameExtensionFilter;  
import javax.swing.table.DefaultTableModel;  
import javax.swing.JMenuBar;  
import java.awt.event.\*;  
import java.io.\*;  
import java.util.ArrayList;  
  
public class gui extends JDialog {  
 private JPanel contentPane;  
 private JTextField textFieldlow;  
 private JTextField textFieldup;  
 private JTextField textFieldstep;  
 private JButton buttonadd;  
 private JButton buttondelete;  
 private JButton buttoncalc;  
 private JTable table;  
 private JButton fill;  
 private JButton clear;  
 private JButton savebinButton;  
 private JButton saveButton;  
 private JButton loadButton;  
 private JButton loadbinButton;  
 private JFileChooser fileChooser = new JFileChooser(("d:"));  
 private File file;  
 private Recintegral recintegral = new Recintegral();  
 private ArrayList<Recintegral> digits = new ArrayList();  
 DefaultTableModel model = (DefaultTableModel) table.getModel();  
 boolean check;  
 public gui()  
 {  
 setContentPane(contentPane);  
 setModal(true);  
 clear.setEnabled(false);  
 fill.setEnabled(false);  
 buttoncalc.setEnabled(false);  
 buttondelete.setEnabled(false);  
  
 createtable();  
 savebinButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent actionEvent) {  
 FileNameExtensionFilter filter;  
 filter = new FileNameExtensionFilter("Binary file", "bin");  
 fileChooser.addChoosableFileFilter(filter);  
 fileChooser.setFileFilter(filter);  
 int choice = fileChooser.showSaveDialog(fileChooser);  
 //if (choice != JFileChooser.APPROVE\_OPTION) return;  
 file = fileChooser.getSelectedFile();  
 saveBinFile();  
 }  
 });  
 saveButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent actionEvent) {  
 FileNameExtensionFilter filter;  
 filter = new FileNameExtensionFilter("Text file", "txt");  
 fileChooser.addChoosableFileFilter(filter);  
 fileChooser.setFileFilter(filter);  
 int choice = fileChooser.showSaveDialog(fileChooser);  
 //if (choice != JFileChooser.APPROVE\_OPTION) return;  
 file = fileChooser.getSelectedFile();  
 try {  
 saveFile();  
 } catch (IOException ex) {  
 throw new RuntimeException(ex);  
 }  
 }  
 });  
 loadbinButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent actionEvent) {  
 FileNameExtensionFilter filter = new FileNameExtensionFilter("Binary file", "bin");  
 fileChooser.setFileFilter(filter);  
 int choice = fileChooser.showOpenDialog(fileChooser);  
 if (choice != JFileChooser.APPROVE\_OPTION) return;  
 file = fileChooser.getSelectedFile();  
 loadbinFile();  
 }  
 });  
 loadButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent actionEvent) {  
 FileNameExtensionFilter filter = new FileNameExtensionFilter("Text file", "txt");  
 fileChooser.setFileFilter(filter);  
 int choice = fileChooser.showOpenDialog(fileChooser);  
 if (choice != JFileChooser.APPROVE\_OPTION) return;  
 file = fileChooser.getSelectedFile();  
 try {  
 loadFile();  
 } catch (IOException ex)  
 {  
 throw new RuntimeException(ex);  
 } catch (NumberFormatException ex)  
 {  
 throw new RuntimeException(ex);  
 }  
 }  
 });  
 buttonadd.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 add();  
 }  
 });  
  
 buttondelete.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 delete();  
 }  
 });  
  
 buttoncalc.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 try {  
 calc();  
 } catch (excep ex) {  
 throw new RuntimeException(ex);  
 }  
 }  
 });  
 fill.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e)  
 {  
 tabfill();  
 }  
 });  
 clear.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e)  
 {  
 tabclear();  
 }  
 });  
 // call onCancel() when cross is clicked  
 setDefaultCloseOperation(DO\_NOTHING\_ON\_CLOSE);  
 addWindowListener(new WindowAdapter() {  
 public void windowClosing(WindowEvent e) {  
 onCancel();  
 }  
 });  
  
 // call onCancel() on ESCAPE  
 contentPane.registerKeyboardAction(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 onCancel();  
 }  
 }, KeyStroke.getKeyStroke(KeyEvent.VK\_ESCAPE, 0), JComponent.WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT);  
 }  
  
 public void createtable()  
 {  
 model.addColumn("Верхняя граница интегрирования");  
 model.addColumn("Нижняя граница интегрирования");  
 model.addColumn("Шаг интегрирования");  
 model.addColumn("Результат");  
 }  
 public void saveBinFile(){  
  
 try(ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(file)))  
 {  
 oos.writeObject(digits);  
 System.out.println("File has been written");  
 }  
 catch(Exception ex)  
 {  
 System.out.println(ex.getMessage());  
 }  
  
  
 }  
 public void loadbinFile() {  
 try(ObjectInputStream ois = new ObjectInputStream(new FileInputStream(file)))  
 {  
 digits=((ArrayList<Recintegral>)ois.readObject());  
 }  
 catch(Exception ex){  
 System.out.println(ex.getMessage());  
 }  
 AddInCollection();  
 }  
 public void saveFile() throws IOException {  
 int countData = digits.size();  
  
 FileWriter myfile = new FileWriter(file);  
  
 for(int i=0;i<countData;i++) {  
 myfile.write(digits.get(i).toString() + "\n");  
 }  
 myfile.flush();  
 myfile.close();  
 }  
 public void loadFile() throws IOException, NumberFormatException {  
 FileReader myfile = new FileReader(file);  
 BufferedReader reader = new BufferedReader(myfile);  
 int i=0;  
 digits.add(i, new Recintegral());  
 String line = reader.readLine();  
  
 String[] dblArray = line.split(",");  
  
 digits.get(0).setMin(Double.valueOf(dblArray[0]));  
 digits.get(0).setMax(Double.valueOf(dblArray[1]));  
 digits.get(0).setStep(Double.valueOf(dblArray[2]));  
 digits.get(0).setResult(Double.valueOf(dblArray[3]));  
  
 while (line != null) {  
 // считываем остальные строки в цикле  
 i++;  
 line = reader.readLine();  
 if(line==null)break;  
 dblArray = line.split(",");  
 digits.add(i, new Recintegral());  
 digits.get(i).setMin(Double.valueOf(dblArray[0]));  
 digits.get(i).setMax(Double.valueOf(dblArray[1]));  
 digits.get(i).setStep(Double.valueOf(dblArray[2]));  
 digits.get(i).setResult(Double.valueOf(dblArray[3]));  
  
 }  
 AddInCollection();  
  
 }  
 private void AddInCollection()  
 {  
 for(int k=0;k< digits.size();k++) {  
 model.addRow(digits.get(k).addMod());  
 }  
 }  
 private void tabfill(){  
 model.addRow(digits.get(0).addMod());  
 str++;  
 }  
 private void tabclear(){  
 model.setRowCount(0);  
 str=-1;  
 clear.setEnabled(false);  
 fill.setEnabled(false);  
 buttoncalc.setEnabled(false);  
 buttondelete.setEnabled(false);  
 }  
  
 private void onCancel() {  
 // add your code here if necessary  
 dispose();  
 }  
 int str=-1;  
 private void add(){  
 try  
 {  
 if(textFieldup.getText().equals("")) {throw new excep();}  
 if(textFieldlow.getText().equals("")){throw new excep();}  
 if(textFieldstep.getText().equals("")){throw new excep();}  
 double a = Double.parseDouble(textFieldlow.getText());  
 double b = Double.parseDouble(textFieldup.getText());  
 double h = Double.parseDouble(textFieldstep.getText());  
 if(a <= 0.000001 || a>=1000000)  
 throw new excep();  
 if(b <= 0.000001 || b>=1000000)  
 throw new excep();  
 if(h <= 0.000001 || h>=1000000)  
 throw new excep();  
 check = true;  
 model.addRow( new Object[]{textFieldlow.getText(), textFieldup.getText(), textFieldstep.getText()});  
 recintegral.setAllField(Double.valueOf(textFieldlow.getText()),Double.valueOf(textFieldup.getText()),Double.valueOf(textFieldstep.getText()));  
 digits.add(recintegral);  
 str++;  
 }  
 catch (excep ex)  
 {  
 throw new RuntimeException(ex);  
 }  
 if (check)  
 {  
 clear.setEnabled(true);  
 fill.setEnabled(true);  
 buttoncalc.setEnabled(true);  
 buttondelete.setEnabled(true);  
 }  
 }  
  
 private void delete(){  
 if(str == 0 && check)  
 {  
 clear.setEnabled(false);  
 fill.setEnabled(false);  
 buttoncalc.setEnabled(false);  
 buttondelete.setEnabled(false);  
 }  
 DefaultTableModel model = (DefaultTableModel)table.getModel();  
 model.removeRow(table.getSelectedRow());  
 str--;  
 }  
  
 private void calc() throws excep{  
 table.setValueAt(recintegral.getResult(), str, 3);  
 textFieldlow.setText("");  
 textFieldup.setText("");  
 textFieldstep.setText("");  
 }  
  
 public double counted() throws excep  
 {  
 double a = Double.parseDouble(textFieldlow.getText());  
 double b = Double.parseDouble(textFieldup.getText());  
 double h = Double.parseDouble(textFieldstep.getText());  
 return integral(a, b, h);  
 }  
 public double integral(double a, double b, double h)  
 {  
 double area = 0;  
 if (h == 0) return area;  
  
 for(int i = 0; i < (b-a)/h; i++){  
 area +=f(a + i\*h);  
 }  
 area += (f(a)+f(b))/2;  
 area \*= h;  
 return area;  
 }  
  
 public double f(double x){  
 return ((Math.pow(Math.E, x))/x);  
 }  
  
 public static void main(String[] args) {  
 gui dialog = new gui();  
 dialog.pack();  
 dialog.setVisible(true);  
 System.exit(0);  
 }  
  
}

**Вывод:** научился разрабатывать приложения, обладающие графическим интерфейсом пользователя, с использованием библиотеки Swing.