МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра «ВТ»

ОТЧЁТ

по лабораторной работе № 3

Оформление документов по управлению проектом

по дисциплине **«Программирование на языке Java»**

Выполнили: ст. гр. 21ВВП1

Назаров Е.А.

Макаров И.С  
Козлова К.С.

Принял:

Юрова О.В.  
Карамышева Н.С.

2024

**Цель работы:** изучить механизм обработки исключительных ситуаций.

**Вариант:** №1

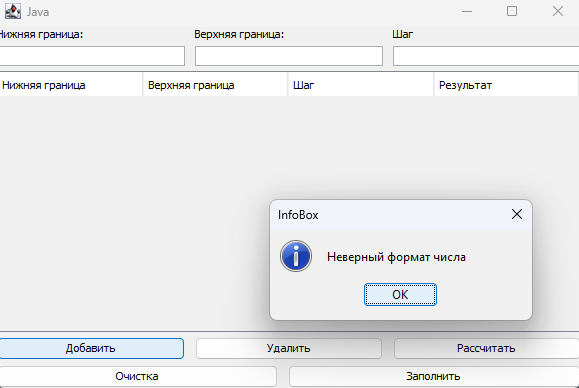
**Ход работы:**

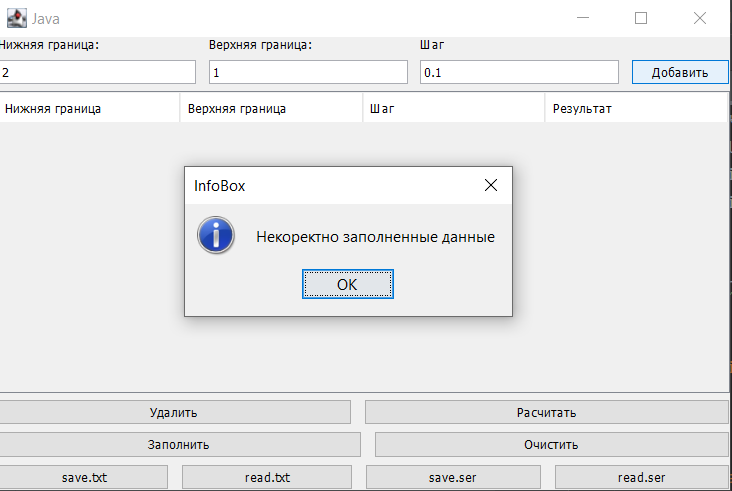
Добавили класс MyException, который будет реализовывать наши собственные исключения

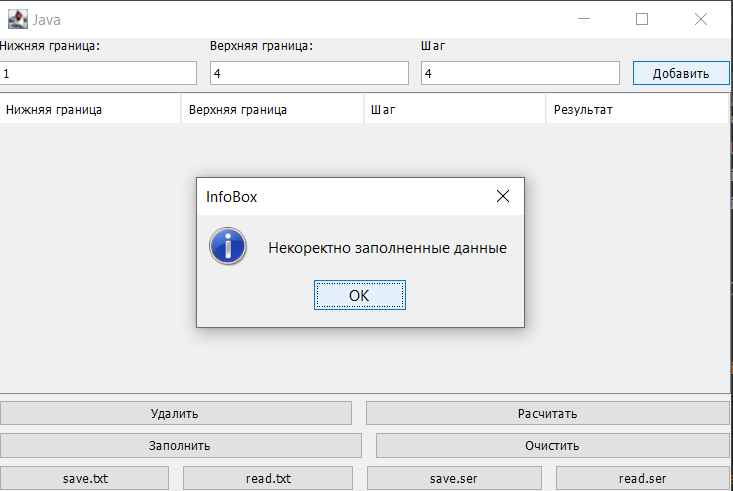
public class MyException extends Exception{  
 private String message;  
 public MyException(String msg){  
 super(msg);  
 this.message = msg;  
 }

}  
 Добавили в класс MyException класс, который будет выводить на экран диалоговое окно с ошибкой.

public void GetFormMessage(){  
 JOptionPane.*showMessageDialog*(null, message, "InfoBox", JOptionPane.*INFORMATION\_MESSAGE*);  
 }

**Исключения   
**

****

****

**Листинг:  
 main**import javax.swing.\*;  
  
public class Main {  
 public static void main(String[] args) throws UnsupportedLookAndFeelException, ClassNotFoundException, InstantiationException, IllegalAccessException {  
 UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());  
 lab app = new lab();  
 app.setVisible(true);  
 }  
}  
  
 **lab**import javax.swing.\*;  
import java.io.\*;  
import java.awt.\*;  
import javax.swing.table.DefaultTableModel;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.util.ArrayList;  
  
public class lab extends JFrame {  
 private JTextField input1;  
 private JTextField input2;  
 private JTextField input3;  
 private JTable table;  
 private JButton button1;  
 private JButton button2;  
 private JButton button3;  
 private JButton button4;  
 private JButton button5;  
 private JLabel label1;  
 private JLabel label2;  
 private JLabel label3;  
 private JPanel rootPanel;  
  
 private DefaultTableModel model;  
 ArrayList<Recintegral> contenttable=new ArrayList<>();  
  
 public lab() {  
 add(rootPanel);  
 setTitle("Java");  
 setSize(600, 400);  
  
 // Добавляем обработчики событий к кнопкам  
 button1.addActionListener(new Button1EventListener());  
 button2.addActionListener(new Button2EventListener());  
 button3.addActionListener(new Button3EventListener());  
 button4.addActionListener(new Button4EventListener());  
 button5.addActionListener(new Button5EventListener());  
 }  
  
 public double parseDouble(String input) throws MyException {  
 input = input.replace(',', '.');  
 if (input.matches("-?\\d+(\\.\\d+)?")) {  
 double tmp = Double.*parseDouble*(input);  
 if(tmp > 1000000 || tmp< 0.000001)  
 throw new MyException("Выход за диапазон значений");  
 return tmp;  
 } else {  
 throw new MyException("Неверный формат числа");  
 }  
 }  
  
 private void createUIComponents() {  
 String[] columns = {"Нижняя граница", "Верхняя граница", "Шаг", "Результат"};  
 String[][] data = {};  
  
 model = new DefaultTableModel(data, columns);  
 table = new JTable(model);  
 }  
  
 class Button1EventListener implements ActionListener {  
 public void actionPerformed(ActionEvent e) {  
 try {  
 Recintegral temp = new Recintegral(parseDouble(input1.getText()),  
 parseDouble(input2.getText()), parseDouble(input3.getText()));  
 model.addRow(new Object[]{input1.getText(), input2.getText(), input3.getText()});  
 contenttable.add(temp);  
 }  
 catch (MyException ex) {  
 ex.GetFormMessage();  
 }  
 }  
 }  
  
 class Button2EventListener implements ActionListener {  
 public void actionPerformed(ActionEvent e) {  
 int temp=table.getSelectedRow();  
 model.removeRow(temp);  
 contenttable.remove(temp);  
 }  
 }  
 class Button3EventListener implements ActionListener {  
 public void actionPerformed(ActionEvent e) {  
 int n = table.getSelectedRow();  
 double a = Double.*parseDouble*(model.getValueAt(n, 0).toString());  
 double b = Double.*parseDouble*(model.getValueAt(n, 1).toString());  
 double h = Double.*parseDouble*(model.getValueAt(n, 2).toString());  
 Level1 l1 = new Level1();  
 String Res= String.*valueOf*(l1.Trap(a, b, h));  
 model.setValueAt(Res, table.getSelectedRow(), 3);  
 Recintegral temp = new Recintegral(a,b,h,Res);  
 contenttable.set(n, temp);  
 }  
 }  
 class Button4EventListener implements ActionListener {  
 public void actionPerformed(ActionEvent e) {  
 model.setRowCount(0);  
 for (Recintegral con:contenttable){  
 if (con.Result==null)  
 model.addRow(new Object[]{con.LowBorder,con.HighBorder,con.Step});  
 else  
 model.addRow(new Object[]{con.LowBorder,con.HighBorder,con.Step,con.Result});  
 }  
 }  
 }  
 class Button5EventListener implements ActionListener {  
 public void actionPerformed(ActionEvent e) {  
 model.setRowCount(0);  
 }  
 }  
  
 class Level1 {  
 public double f(double x) {  
 return 1/x;  
 }  
 public double Trap(double a, double b, double h){  
 double result = 0;  
 for(double i = a; i < b; i+=h) {  
 if((i+h) < b )  
 result += (f(i) + f(h+i)) \* h/2;  
 else  
 result += (f(i)+f(b)) \* (b-i)/2;  
 }  
  
 return result;  
 }  
 };  
 public class Recintegral {  
 double LowBorder;  
 double HighBorder;  
 double Step;  
 String Result;  
  
 public Recintegral(double LowBorder,double HighBorder, double Step, String Result ) {  
 this.LowBorder=LowBorder;  
 this.HighBorder=HighBorder;  
 this.Step=Step;  
 this.Result=Result;  
 }  
  
 public Recintegral(double LowBorder,double HighBorder, double Step) {  
 this.LowBorder=LowBorder;  
 this.HighBorder=HighBorder;  
 this.Step=Step;  
 this.Result=null;  
 }  
  
 }  
  
}

**MyException**import javax.swing.\*;  
public class MyException extends Exception{  
 private String message;  
 public MyException(String msg){  
 super(msg);  
 this.message = msg;  
 }  
  
 public void GetFormMessage(){  
 JOptionPane.*showMessageDialog*(null, message, "InfoBox", JOptionPane.*INFORMATION\_MESSAGE*);  
 }  
  
}

**Вывод.** изучили механизм обработки исключительных ситуаций.