Пензенский государственный университет

Факультет вычислительной техники

Кафедра «Вычислительная техника»

Обработка исключительных ситуаций

Отчет о лабораторной работе № 3

по дисциплине «Разработка кроссплатформенных приложений»

Вариант № 2

Выполнили: ст-ты гр. 21ВОЭ1

Кокорин А. А.

Тишаев М. Р.

Приняли: к.т.н., доцент каф. ВТ

Юрова О. В.

к.т.н., доцент каф. ВТ

Карамышева Н. С.

2024

1 Цель работы

Изучить механизм обработки исключительных ситуаций.

2 Формулировка задания

Модифицировать приложение из предыдущей лабораторной работы, реализовав проверку вводимых данных с использованием механизма исключений. Необходимо создать свой класс, унаследованный от класса Exception, и генерировать исключение, если возникает попытка создать экземпляр класса RecIntegral со значениями, не являющимися числами в диапазоне от 0,000001 до 1000000. В качестве обработки исключения необходимо выводить диалог, содержащий предупреждение о некорректности введенных данных. Оформление лабораторной работы должно быть выполнено в соответствии с требованиями.

3 Графическое описание алгоритма

Алгоритм метода *jButtonAdd* для добавления данных интеграла в таблицу представлен на рисунке 1. Алгоритм начинается с проверки условия, *isNumeric(Low)* и *isNumeric(Top)* и *isNumeric(Step)* = ложь (блок 2). Если условие истинно, то выводится сообщение об ошибке (блок 3) и метод завершает свою работу. В противном случае происходит проверка условия, *jTextFieldLow* ≥ *jTextFieldTop* (блок 4). Если условие истинно, то выводится сообщение об ошибке (5) и метод завершает свою работу. В противном случае происходит добавление данных из текстовых полей *jTextFieldStep*, *jTextFieldLow*, *jTextFieldTop* в таблицу (блок 6). Затем данные из текстовых полей *jTextFieldStep*, *jTextFieldLow*, *jTextFieldTop* добавляются в конец списка *List* (блок 7). Далее метод завершает свою работу.

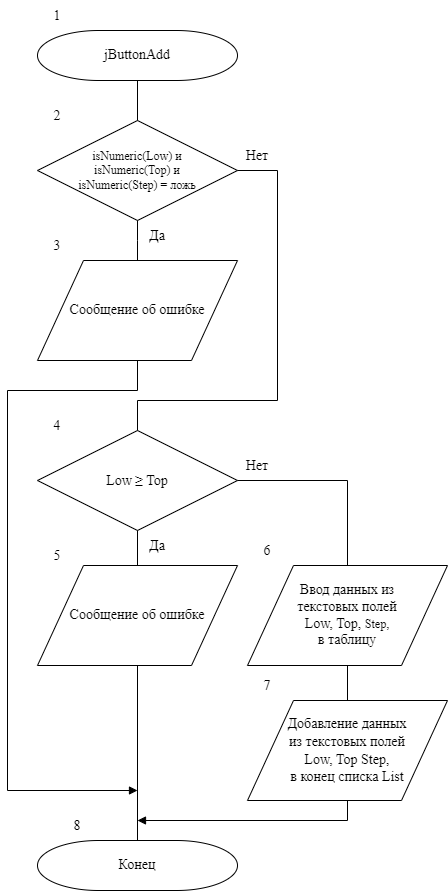


Рисунок 1 – Схема метода *jButtonAdd*

Алгоритм метода *isNumeric(num)* класса *RecIntegral* для проверки, является ли введенное значение числом в диапазоне от 0,000001 до 1000000 представлен на рисунке 2. Алгоритм начинается с проверки условия, является ли значение *num* числом (блок 2). Если условие ложно, то *isNumeric(num)* возвращает значение ложь (блок 3) и метод завершает свою работу. В противном случае осуществляется проверка условия, *num* > 0,000001 и *num* < 1000000 (блок 4). Если условие истинно, то *isNumeric(num)* возвращает значение истина (блок 5). В противном случае *isNumeric(num)* возвращает значение ложь (блок 6). Затем метод завершает свою работу.

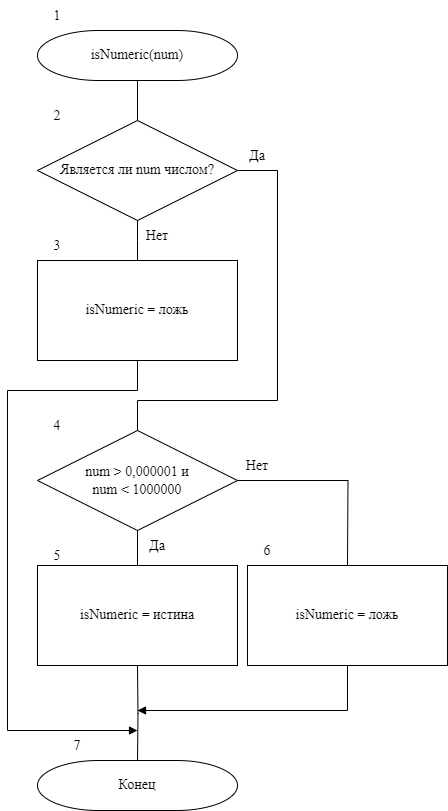


Рисунок 2 – Схема метода *isNumeric(num)*

4 Описание программы

Программа построена на основе лабораторной работы № 2.

Был добавлен класс *ExceptionRecIntegral* унаследованный от класса *Exception* для генерации исключения.

В классе *ExceptionRecIntegral* были описан конструктор *ExceptionRecIntegral(String message)*, который вызывает конструктор родительского класса и передает ему параметр *message.*

В класс *RecIntegral* был добавлен метод *isNumeric(num)* для проверки, является ли *num* числом в диапазоне от 0,000001 до 1000000.

5 Программа и методика испытаний

Для проверки правильности работы программы подготовлен вручную тестовый набор данных в таблице 1.

Таблица 1 – Тестовый набор данных

|  |  |  |  |
| --- | --- | --- | --- |
| Нижняя граница | Верхняя граница | Шаг | Результат работы программы |
| 13qwr | 14 | 1 | Сообщение об ошибке |
| 13 | 12 | 1 | Сообщение об ошибке |
| 0 | 10 | 1 | Сообщение об ошибке |
| 1 | – | 1 | Сообщение об ошибке |

При работе программы с тестовыми данными получены результаты, приведенные в Приложении Б. Результат, приведенный в таблице 1, совпадают с результатом работы программы на рисунке Б.1, Б.2, Б.3, Б.4. Таким образом, можно сделать вывод, что программа работает правильно.

Вывод

Результатом выполнения данной лабораторной работы является изучение механизма обработки исключительных ситуаций и добавление в программу своего класса, унаследованного от класса Exception, для создания исключения, если возникает попытка создать экземпляр класса RecIntegral со значениями, не являющимися числами в диапазоне от 0,000001 до 1000000 .

ТЕКСТ ПРОГРАММЫ

Приложение А

(обязательное)

Класс NewJFrame

package my.javaapplication1;

import javax.swing.table.DefaultTableModel;

import java.util.LinkedList;

public class NewJFrame extends javax.swing.JFrame {

LinkedList<RecIntegral> List = new LinkedList<RecIntegral>();

public NewJFrame() {

initComponents();

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jDialog1 = new javax.swing.JDialog();

jLabelError = new javax.swing.JLabel();

jButtonAdd = new javax.swing.JButton();

jButtonDel = new javax.swing.JButton();

jButtonFind = new javax.swing.JButton();

jTextFieldLow = new javax.swing.JTextField();

jTextFieldTop = new javax.swing.JTextField();

jTextFieldStep = new javax.swing.JTextField();

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jButtonClear = new javax.swing.JButton();

jButtonFill = new javax.swing.JButton();

jDialog1.setTitle("Ошибка ввода");

jLabelError.setText("Ошибка");

javax.swing.GroupLayout jDialog1Layout = new javax.swing.GroupLayout(jDialog1.getContentPane());

jDialog1.getContentPane().setLayout(jDialog1Layout);

jDialog1Layout.setHorizontalGroup(

jDialog1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 400, Short.MAX\_VALUE)

.addGroup(jDialog1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jDialog1Layout.createSequentialGroup()

.addGap(0, 0, Short.MAX\_VALUE)

.addComponent(jLabelError)

.addGap(0, 0, Short.MAX\_VALUE)))

);

jDialog1Layout.setVerticalGroup(

jDialog1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 300, Short.MAX\_VALUE)

.addGroup(jDialog1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jDialog1Layout.createSequentialGroup()

.addGap(0, 0, Short.MAX\_VALUE)

.addComponent(jLabelError)

.addGap(0, 0, Short.MAX\_VALUE)))

);

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Lab3");

jButtonAdd.setText("Добавить");

jButtonAdd.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButtonAddActionPerformed(evt);

}

});

jButtonDel.setText("Удалить");

jButtonDel.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButtonDelActionPerformed(evt);

}

});

jButtonFind.setText("Вычислить");

jButtonFind.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButtonFindActionPerformed(evt);

}

});

jTable1.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

new String [] {

"Нижняя граница", "Верхняя граница", "Шаг", "Результат"

}

) {

boolean[] canEdit = new boolean [] {

true, true, true, false

};

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

jTable1.getTableHeader().setReorderingAllowed(false);

jScrollPane1.setViewportView(jTable1);

if (jTable1.getColumnModel().getColumnCount() > 0) {

jTable1.getColumnModel().getColumn(0).setResizable(false);

jTable1.getColumnModel().getColumn(1).setResizable(false);

jTable1.getColumnModel().getColumn(2).setResizable(false);

jTable1.getColumnModel().getColumn(3).setResizable(false);

}

jLabel1.setText("Нижняя граница");

jLabel2.setText("Верхняя граница");

jLabel3.setText("Шаг");

jButtonClear.setText("Очистить");

jButtonClear.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButtonClearActionPerformed(evt);

}

});

jButtonFill.setText("Заполнить");

jButtonFill.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButtonFillActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 551, Short.MAX\_VALUE)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jButtonFind, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButtonAdd, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButtonDel, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGap(30, 30, 30)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jTextFieldLow)

.addComponent(jTextFieldTop))

.addGap(79, 79, 79)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jButtonFill, javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButtonClear, javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(109, 109, 109))

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT\_SIZE, 94, Short.MAX\_VALUE)

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel3)

.addGap(0, 0, Short.MAX\_VALUE)))

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 102, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextFieldStep))

.addGap(288, 288, 288))))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButtonAdd)

.addComponent(jTextFieldLow, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButtonFill))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jLabel2)

.addGap(6, 6, 6)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButtonDel)

.addComponent(jTextFieldTop, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButtonClear))

.addGap(17, 17, 17)

.addComponent(jLabel3)

.addGap(3, 3, 3)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButtonFind)

.addComponent(jTextFieldStep, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 34, Short.MAX\_VALUE)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 176, javax.swing.GroupLayout.PREFERRED\_SIZE))

);

pack();

setLocationRelativeTo(null);

}// </editor-fold>

private void jButtonAddActionPerformed(java.awt.event.ActionEvent evt) {

try {

DefaultTableModel TModel = (DefaultTableModel) jTable1.getModel();

if ((RecIntegral.isNumeric(jTextFieldLow.getText()) && RecIntegral.isNumeric(jTextFieldTop.getText()) && RecIntegral.isNumeric(jTextFieldStep.getText())) == false) {

throw new ExceptionRecIntegral("Ошибка при вводе исходных данных");

}

else if (Double.parseDouble(jTextFieldLow.getText()) >= Double.parseDouble(jTextFieldTop.getText())) {

throw new ExceptionRecIntegral("Ошибка при вводе исходных данных");

}

else{

TModel.addRow(new Object[] {jTextFieldLow.getText(), jTextFieldTop.getText(), jTextFieldStep.getText()});

List.addLast(new RecIntegral(Double.valueOf(jTextFieldLow.getText()), Double.valueOf(jTextFieldTop.getText()), Double.valueOf(jTextFieldStep.getText())));

}

}

catch (ExceptionRecIntegral ExRec) {

jDialog1.setSize(250, 100);

jDialog1.setLocationRelativeTo(null);

jDialog1.setModal(true);

jLabelError.setText(ExRec.getMessage());

jDialog1.setVisible(true);

}

}

private void jButtonDelActionPerformed(java.awt.event.ActionEvent evt) {

int row = jTable1.getSelectedRow();

if (row != -1) {

DefaultTableModel TModel = (DefaultTableModel) jTable1.getModel();

TModel.removeRow(row);

List.remove(row);

}

}

private void jButtonFindActionPerformed(java.awt.event.ActionEvent evt) {

int row = jTable1.getSelectedRow();

if (row != -1) {

double h, a, b, res = 0;

a = Double.parseDouble(jTable1.getValueAt(row, 0).toString());

b = Double.parseDouble(jTable1.getValueAt(row, 1).toString());

h = Double.parseDouble(jTable1.getValueAt(row, 2).toString());

if ((a < b) && ((b - a) % h < 0.0001 || (b - a) % h > h - 0.0001)) {

for (double i = a; i < b; i += h)

res += ((Math.sin(i \* Math.PI / 180) + Math.sin((i + h) \* Math.PI / 180)) \* (h / 2));

jTable1.setValueAt(String.format("%.4f",res), row, 3);

}

}

}

private void jButtonFillActionPerformed(java.awt.event.ActionEvent evt) {

if (List.size() != -1) {

DefaultTableModel TModel = (DefaultTableModel) jTable1.getModel();

for (RecIntegral Rec: List)

TModel.addRow(new Object[] {Rec.geta(), Rec.getb(), Rec.geth()});

}

}

private void jButtonClearActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel TModel = (DefaultTableModel) jTable1.getModel();

TModel.setRowCount(0);

}

public static void main(String args[]) {

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new NewJFrame().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton jButtonAdd;

private javax.swing.JButton jButtonClear;

private javax.swing.JButton jButtonDel;

private javax.swing.JButton jButtonFill;

private javax.swing.JButton jButtonFind;

private javax.swing.JDialog jDialog1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabelError;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTable jTable1;

private javax.swing.JTextField jTextFieldLow;

private javax.swing.JTextField jTextFieldStep;

private javax.swing.JTextField jTextFieldTop;

// End of variables declaration

}

Класс RecIntegral

package my.javaapplication1;

import java.io.\*;

public class RecIntegral implements Serializable {

private double a, b, h;

public RecIntegral(double a, double b, double h) {

this.a = a;

this.b = b;

this.h = h;

}

public double geta(){

return this.a;

}

public double getb(){

return this.b;

}

public double geth(){

return this.h;

}

public static boolean isNumeric(String num) {

try {

if((Double.parseDouble(num) > 0.000001 && Double.parseDouble(num) < 1000000))

return true;

else

return false;

}

catch(NumberFormatException e) {

return false;

}

}

}

Класс ExceptionRecIntegral

package my.javaapplication1;

public class ExceptionRecIntegral extends Exception {

public ExceptionRecIntegral(String message){

super(message);

}

}

РЕЗУЛЬТАТЫ ИСПЫТАНИЙ

Приложение Б

(обязательное)

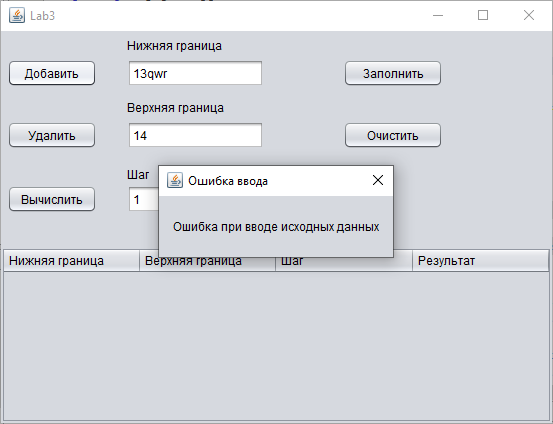


Рисунок Б.1 – Сообщение об ошибке

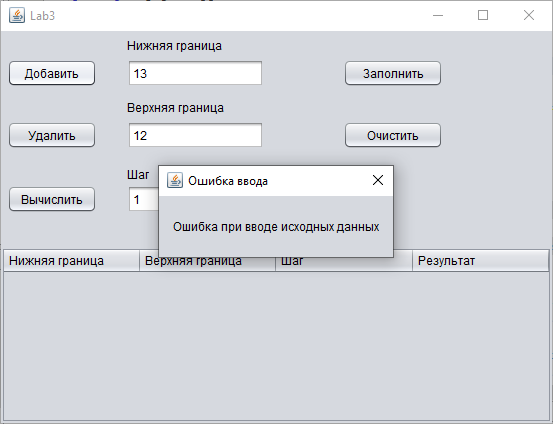


Рисунок Б.2 – Сообщение об ошибке

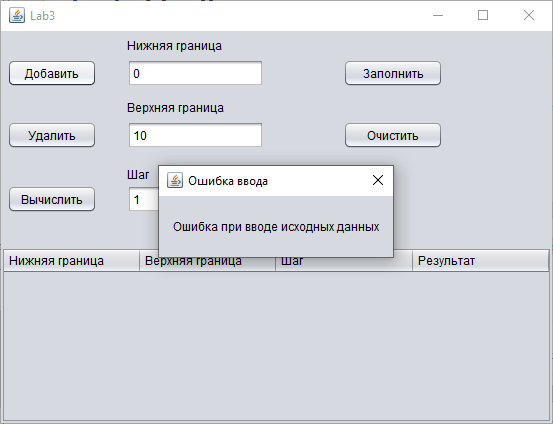


Рисунок Б.3 – Сообщение об ошибке

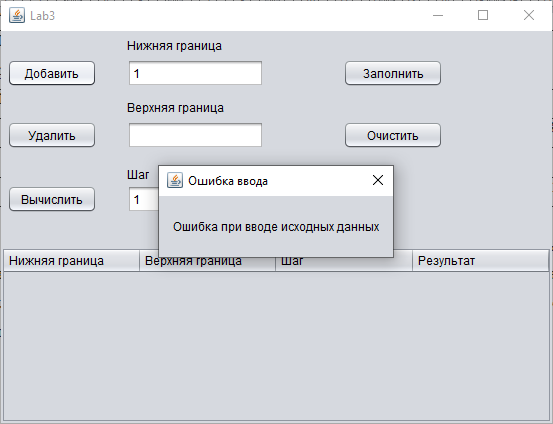


Рисунок Б.4 – Сообщение об ошибке