Пензенский государственный университет

Факультет вычислительной техники

Кафедра «Вычислительная техника»

## Обработка исключительных ситуаций

Вариант №1

Отчёт по лабораторной работе №3

По дисциплине «Разработка кроссплатформенных приложений»

Выполнили: ст-ты гр.21ВОЭ1

Куимов Н.А

Комраков А.М

Проверили:д.т.н.,

к.т.н., доцент каф.ВТ

Карамышева Н.С.

Юрова О.Н

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, приведенными в Приложении 2.

3 Графическое описание алгоритма

Алгоритм вычисления интеграла в виде схемы программы представлен на рисунке 1. Затем происходит проверка условия *stroka !=-1* (блок 3). Если условие ложно, то программа завершает свою работу. В противном случае осуществляется ввод: *nzg* –нижняя граница интегрирования, *vzg* – верхняя граница интегрирования, *shg* – шаг интегрирования (блок 4). Далее происходит цикл res += (((1 / i) + (1 / (i + shg)))) \* (shg / 2) (блок 5).

Если условие истинно, то происходит вычисление определенного интеграла (блоки 6 - 8) и вывод результата вычисления (блок 9). В противном случае программа завершает свою работу.

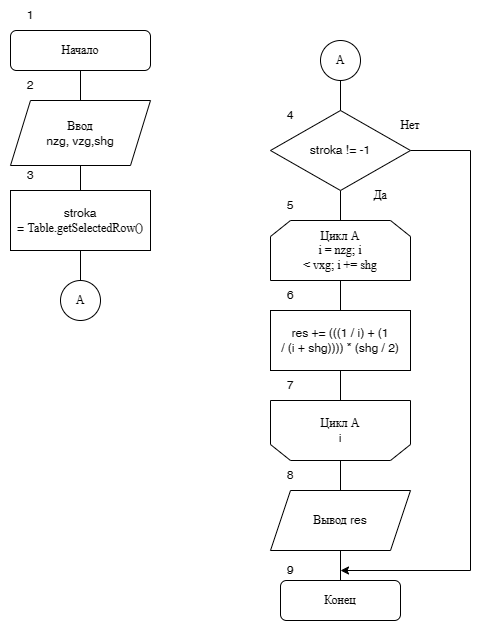


Рисунок 1 – Схема программы для вычисления интеграла

4 Описание программы

Текст программы приведен в приложении А.

На Форме расположены следующие объекты:

три однострочных редактора: *jTextFieldLow* (для ввода нижней границы интегрирования), *jTextFieldTop* (для ввода верхней границы интегрирования), *jTextFieldStep* (для ввода шага интегрирования);

три объекта метки: *jLabel1*, *jLabel2*, *jLabel3*. В свойстве *text* этих объектов помещены пояснения для пользователя программы:

*jLabel1.text* — «Нижняя граница»;

*jLabel2.text* — «Верхняя граница»;

*jLabel3.text* — «Шаг»;

одна таблица — *jTable1* (для ввода исходных данных);

кнопки *jButtonAdd* для создания обработчика события с целью добавления строки с введенными данными в таблицу, *jButtonDel* для создания обработчика события с целью удаления строки из таблицы, *jButtonFind* для создания обработчика события с целью вычисления интеграла и его вывода в таблицу, *Button2.setText* для заполнения таблицы, *ButtonClearData.setText* для полной очистки таблицы. В свойстве *text* этих объектов помещены пояснения для пользователя программы:

*ButtonAdd.text* — «Добавить»;

*ButtonDel.text* — «Удалить»;

*ButtonFind.text* — «Вычислить»;

*Button2.setText* —«Заполнить»;

*ButtonClearData.setText* —«Очистить»;

Программа состоит из пяти обработчиков событий: обработчик события клика на кнопку формы *jButtonAdd* — *jButtonAddActionPerformed*, обработчик события клика на кнопку формы *jButtonDel* — *jButtonDelActionPerformed*, обработчик события клика на кнопку формы *jButtonFind* — *jButtonFindActionPerformed, Button2.setText* —*jButton2ActionPerformed, ButtonClearData.setText* —*jButtonClearDataActionPerformed.* В методе *jButtonFindActionPerformed* описаны:

переменная целого типа *integer:*

*stroka*– количество строк в таблице.

переменные вещественного типа *double:*

*nzg* – нижняя граница интегрирования;

*vzg* – верхняя граница интегрирования;

*shg* – шаг интегрирования;

*res* – результат вычисления интеграла.

5 Программа и методика испытаний

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

Таблица 1 – Тестовый набор данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a | b | h | Результат работы программы | Результат ручного просчета |
| 5 | 15 | 1 | 1,101562… | 1,101562… |
| 6 | 32 | 1 | 1,676235… | 1,676235… |
| 6 | 12 | 4 | 0 | 0 |
| 10 | 20 | 4 | 0 | 0 |

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.2, Б.3. Таким образом, можно сделать вывод, что программа работает правильно.

Вывод

Результатом выполнения данной лабораторной работы является получение навыков разработки приложений, обладающих графическим интерфейсом пользователя, с использованием библиотеки Swing и написание программы, обладающей графическим интерфейсом, для вычисления определенного интеграла.

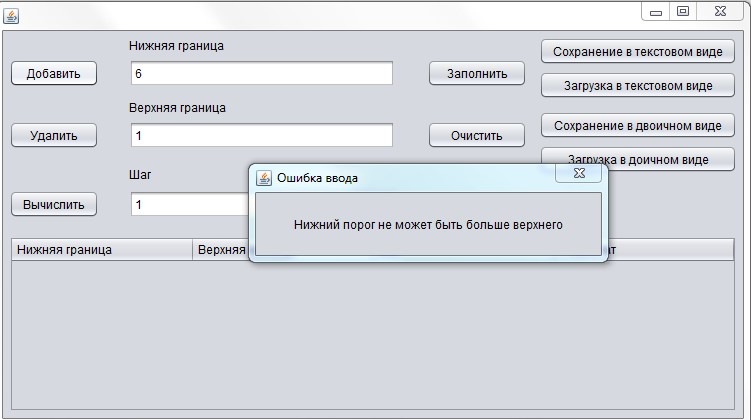


Рисунок 1 – Ошибка ввода (Нижний порог не может быть больше верхнего)

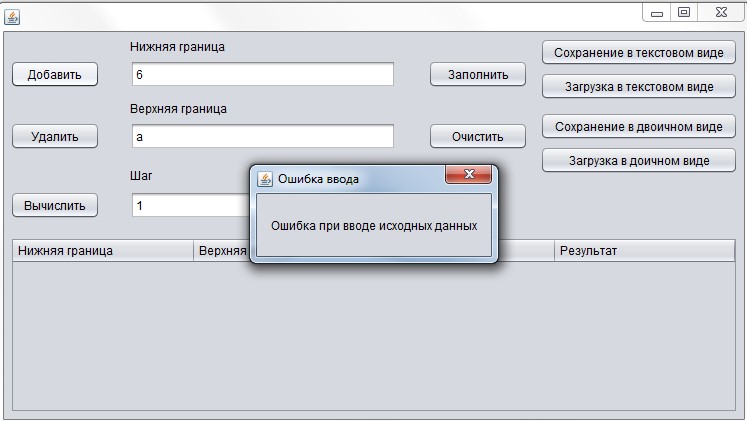


Рисунок 2 – Ошибка ввода (Ошибка при вводе исходных данных)

Листинг программы

package my.javaapplication1;

import javax.swing.table.DefaultTableModel;

import java.util.ArrayList;

import java.io.\*;

import java.util.Scanner;

public class NewJFrame extends javax.swing.JFrame {

ArrayList<RecIntegral> AList = new ArrayList<RecIntegral>();

public NewJFrame() {

initComponents();

}

@SuppressWarnings("unchecked")

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

private void initComponents() {

jFileChooser1 = new javax.swing.JFileChooser();

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();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jButton3 = new javax.swing.JButton();

jButton4 = new javax.swing.JButton();

jDialog1.setTitle("Ошибка ввода");

jLabelError.setText("jLabel4");

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, 0, 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, 0, 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);

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);

}

});

jButton1.setText("Сохранение в текстовом виде");

jButton1.addActionListener(new java.awt.event.ActionListener() {

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

jButton1ActionPerformed(evt);

}

});

jButton2.setText("Загрузка в текстовом виде");

jButton2.addActionListener(new java.awt.event.ActionListener() {

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

jButton2ActionPerformed(evt);

}

});

jButton3.setText("Сохранение в двоичном виде");

jButton3.addActionListener(new java.awt.event.ActionListener() {

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

jButton3ActionPerformed(evt);

}

});

jButton4.setText("Загрузка в доичном виде");

jButton4.addActionListener(new java.awt.event.ActionListener() {

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

jButton4ActionPerformed(evt);

}

});

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

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

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

.addGroup(layout.createSequentialGroup()

.addContainerGap()

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

.addComponent(jScrollPane1)

.addGroup(layout.createSequentialGroup()

.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.LEADING)

.addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT\_SIZE, 266, Short.MAX\_VALUE)

.addComponent(jTextFieldTop)

.addComponent(jTextFieldStep)

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel3)

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

.addComponent(jTextFieldLow))

.addGap(32, 32, 32)

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

.addComponent(jButtonFill, javax.swing.GroupLayout.DEFAULT\_SIZE, 100, Short.MAX\_VALUE)

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

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 108, javax.swing.GroupLayout.PREFERRED\_SIZE))

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

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

.addComponent(jButton1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton3, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton4, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))))

.addContainerGap())

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addContainerGap()

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

.addGroup(layout.createSequentialGroup()

.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(8, 8, 8)

.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)))

.addGroup(layout.createSequentialGroup()

.addComponent(jButton1)

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

.addComponent(jButton2)

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

.addComponent(jButton3)

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

.addComponent(jButton4)))

.addGap(18, 18, 18)

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

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

);

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 {

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

AList.add(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);

AList.remove(row);

}

}

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

int stroka;

double shg = 0, nzg = 0, vxg = 0, res = 0;

stroka = jTable1.getSelectedRow();

if (stroka != -1) {

nzg = Double.parseDouble(jTable1.getValueAt(stroka, 0).toString());

vxg = Double.parseDouble(jTable1.getValueAt(stroka, 1).toString());

shg = Double.parseDouble(jTable1.getValueAt(stroka, 2).toString());

if ((nzg < vxg) && ((vxg - nzg) % shg < 0.0001 || (vxg - nzg) % shg > shg -0.0001)) {

for (double i = nzg; i < vxg; i += shg)

{

res += (((1 / i) + (1 / (i + shg)))) \* (shg / 2);

}

}

}

jTable1.setValueAt(res, stroka, 3);

AList.set(stroka, new RecIntegral (nzg, vxg, shg, res));

}

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

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

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

for (RecIntegral Rec: AList)

TModel.addRow(new Object[] {Rec.getnzg(), Rec.getvxg(), Rec.getshg(), Rec.getres()});

}

}

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

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

TModel.setRowCount(0);

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

try {

jFileChooser1.showDialog(null,"");

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

int stroka = TModel.getRowCount();

if (stroka != 0) {

File ft = jFileChooser1.getSelectedFile();

FileWriter fwt = new FileWriter(ft);

ft.delete();

for (int i = 0; i < stroka; i++) {

fwt.write(jTable1.getValueAt(i, 0).toString() + "\n");

fwt.write(jTable1.getValueAt(i, 1).toString() + "\n");

fwt.write(jTable1.getValueAt(i, 2).toString() + "\n");

fwt.write(jTable1.getValueAt (i, 3).toString() + "\n");

}

fwt.close();

}

}

catch(Exception ex) {

}

}