Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №3

по курсу «Программирование на языке Java»

на тему «Обработка исключительных ситуаций»

Вариант №3

Выполнили студенты группы 20ВВП2:

Пантюшов Е.И.

Шмелёв Д.В.

Приняли:

Юрова О. В.

Карамышева Н. С.

Пенза 2023

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

**Лабораторное задание:**

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.

**Листинг:**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

import static java.lang.Math.cos;

import java.util.ArrayList;

import javax.swing.table.DefaultTableModel;

import java.util.Vector;

//package com.csharpcoderr.messageDialog;

import javax.swing.JOptionPane;

import static javax.swing.JOptionPane.showMessageDialog;

/\*\*

\*

\* @author PENZA

\*/

public class NewJFrame extends javax.swing.JFrame {

ArrayList<RecIntegral> Data = new ArrayList();

/\*\*

\* Creates new form NewJFrame

\*/

public NewJFrame() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

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

private void initComponents() {

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

jTextField1 = new javax.swing.JTextField();

jTextField2 = new javax.swing.JTextField();

jTextField3 = new javax.swing.JTextField();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jButton3 = new javax.swing.JButton();

jButton4 = new javax.swing.JButton();

jButton5 = new javax.swing.JButton();

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

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

new Object [][] {

},

new String [] {

"Top", "Bottom", "Step", "Result"

}

) {

boolean[] canEdit = new boolean [] {

true, true, true, false

};

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

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);

}

jTextField1.addActionListener(new java.awt.event.ActionListener() {

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

jTextField1ActionPerformed(evt);

}

});

jLabel1.setText("Up");

jLabel2.setText("Down");

jLabel3.setText("Step");

jButton1.setText("Add");

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

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

jButton1ActionPerformed(evt);

}

});

jButton2.setText("Delete");

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

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

jButton2ActionPerformed(evt);

}

});

jButton3.setText("Calculate");

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

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

jButton3ActionPerformed(evt);

}

});

jButton4.setText("Fill");

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

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

jButton4ActionPerformed(evt);

}

});

jButton5.setText("Clear");

jButton5.addActionListener(new java.awt.event.ActionListener() {

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

jButton5ActionPerformed(evt);

}

});

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

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

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

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

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

.addComponent(jButton5, javax.swing.GroupLayout.PREFERRED\_SIZE, 65, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(jButton4, javax.swing.GroupLayout.PREFERRED\_SIZE, 71, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(123, 123, 123))

.addGroup(layout.createSequentialGroup()

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

.addGroup(layout.createSequentialGroup()

.addContainerGap()

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

.addComponent(jLabel1)

.addComponent(jLabel2)

.addComponent(jLabel3))

.addGap(18, 18, 18)

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

.addComponent(jTextField3, javax.swing.GroupLayout.DEFAULT\_SIZE, 43, Short.MAX\_VALUE)

.addComponent(jTextField2)

.addComponent(jTextField1))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

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

.addComponent(jButton2, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, 77, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton1, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, 77, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton3, javax.swing.GroupLayout.Alignment.TRAILING)))

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup()

.addGap(25, 25, 25)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 345, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(30, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

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

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addGap(6, 6, 6)

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

.addComponent(jButton4)

.addComponent(jButton5))

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

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel1)

.addComponent(jButton2))

.addGap(26, 26, 26)

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

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel2)

.addComponent(jButton1))

.addGap(18, 18, 18)

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

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel3)

.addComponent(jButton3))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 17, Short.MAX\_VALUE)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 160, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap())

);

pack();

}// </editor-fold>

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

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

int SelectedRow =jTable1.getSelectedRow();

if (jTable1.getRowCount() != 0)

{

if (SelectedRow == -1)

{

model.removeRow(jTable1.getRowCount()-1);

}

else

{

model.removeRow(jTable1.getSelectedRow());

}

} // TODO add your handling code here:

}

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

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

String TextField1 = jTextField1.getText();

String TextField2 = jTextField2.getText();

String TextField3 = jTextField3.getText();

try

{

float step = Float.parseFloat(TextField3);

int up = Integer.parseInt(TextField1),

down = Integer.parseInt(TextField2);

if (((down < 0.000001 || down >= 100000) || (up < 0.000001 || up >= 100000)))

{

throw new MyException("Введён неверное значение предела!");

}

else if (step <= 0)

{

throw new MyException("Шаг задан неверно!");

}

else if (step >=up-down )

{

throw new MyException("Слишком большой шаг!");

}

RecIntegral Unit = new RecIntegral();

model.addRow(new Object[]{Integer.parseInt(TextField1), Integer.parseInt(TextField2), Float.parseFloat(TextField3), null});

// TODO add your handling code here:

Unit.addUnit(Integer.parseInt(TextField1), Integer.parseInt(TextField2), Float.parseFloat(TextField3));

Data.add(Unit);

}

catch(MyException code)

{

showMessageDialog(null, code.message);

return;

}

catch(Exception code)

{

showMessageDialog(null, "Неверно ввели данные!");

return;

}

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

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

FunctionIntegral funk = new FunctionIntegral();

Vector data = model.getDataVector();

for(int i = 0; i < data.size(); i++){

Vector CurrentData = (Vector)data.get(i);

double n,x1,x2 ;

int j,a,b;

double result=0;

if ((int)CurrentData.get(1)<(int)CurrentData.get(0)){

a=(int)CurrentData.get(1);

b=(int)CurrentData.get(0);

}

else{

a=(int)CurrentData.get(0);

b=(int)CurrentData.get(1);

}

n = (int)((b-a)/(float)CurrentData.get(2));

for(j = 0; j < n-1; j++)

{

x1=a+j\*(float)CurrentData.get(2);

x2=a+(float)CurrentData.get(2)\*(j+1);

result+= 0.5\*(x2-x1)\*(funk.f(x1) + funk.f(x2));

}

if((n-1)\*(float)CurrentData.get(2) < b){

float newstep = (float) (b - (n-1)\*(float)CurrentData.get(2));

x1=a+(n-1)\*(float)CurrentData.get(2);

result+= 0.5\*(b-x1)\*(funk.f(x1) + funk.f(b));

}

if ((int)CurrentData.get(1)<(int)CurrentData.get(0))

result=result\*(-1);

model.setValueAt(result, i, 3);

} // TODO add your handling code here:

}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

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

while(jTable1.getRowCount() != 0)

module.removeRow(jTable1.getRowCount()-1);

}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

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

for(int i = 0; i < Data.size(); i++){

RecIntegral Unit = Data.get(i);

module.addRow(new Object[]{Unit.Top, Unit.Button, Unit.Step, null});

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

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);

}

//</editor-fold>

/\* Create and display the form \*/

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

public void run() {

new NewJFrame().setVisible(true);

}

});

}

class FunctionIntegral {

// {Функция, площадь которой нужно вычислить}

public double f(double x) {

//double b = Math.toRadians(x)

double F= cos(x);

return F;

}

}

class RecIntegral{

public int Top, Button;

public float Step;

public void addUnit(int Top, int Button, float Step){

this.Top=Top;

this.Button=Button;

this.Step=Step;

}

}

static class MyException extends Exception

{

String message;

MyException(String par)

{

message = par;

}

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton3;

private javax.swing.JButton jButton4;

private javax.swing.JButton jButton5;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTable jTable1;

private javax.swing.JTextField jTextField1;

private javax.swing.JTextField jTextField2;

private javax.swing.JTextField jTextField3;

// End of variables declaration

}

**Результат выполнения программы:**

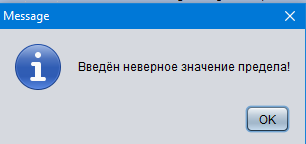


Рисунок 1 — Неверное значение предела

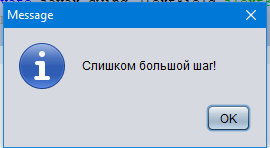


Рисунок 2 — Слишком большой шаг

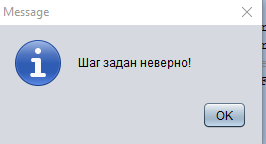


Рисунок 3 — Шаг задан неверно

**Вывод:** в ходе выполнения данной лабораторной работы были изучены механизм обработки исключительных ситуаций.