|  |
| --- |
| Пензенский государственный университет  Факультет вычислительной техники  Кафедра «Вычислительная техника» |
| Отчет  по лабораторной работе №3  по дисциплине «Программирование на языке Java»  на тему «Работа с коллекциями объектов»  Вариант № 12 |
|  |
|  |
| Сделал студент группы 22ВВП1:  Перфилов А.В.  Проверил:  Карамышева Н.С.  Юрова О.В. |
| Пенза 2025 |

**Цель работы.**

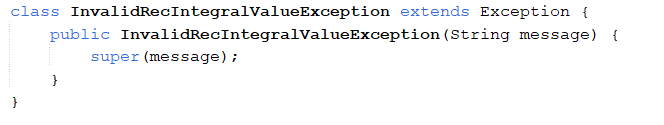
Изучить механизм обработки исключительных ситуаций.

**Задание.**

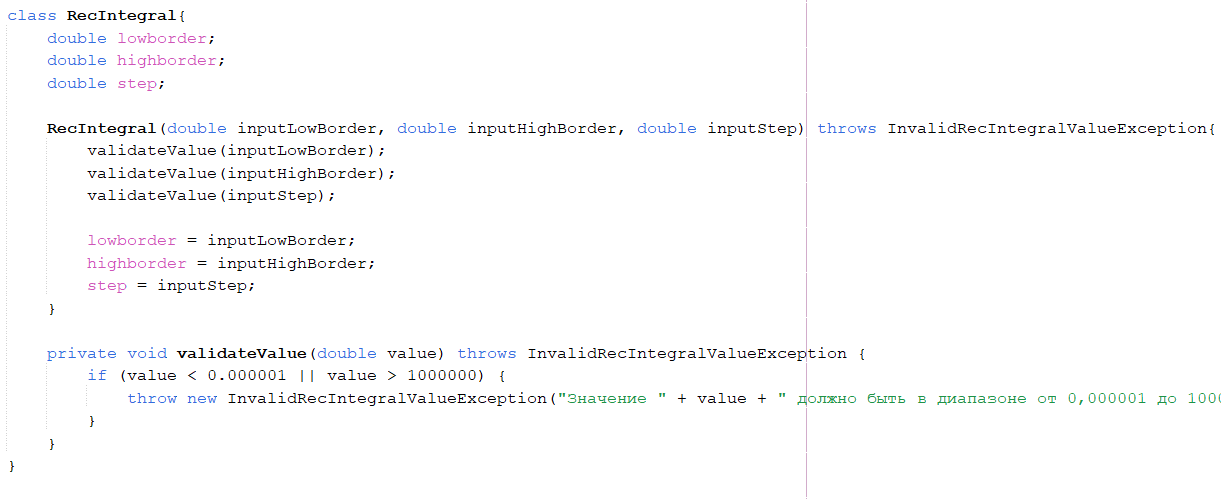
Модифицировать приложение из предыдущей лабораторной работы, реализовав проверку вводимых данных с использованием механизма исключений. Необходимо создать свой класс, унаследованный от класса Exception, и генерировать исключение, если возникает попытка создать экземпляр класса RecIntegral со значениями, 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.

**Ход работы.**

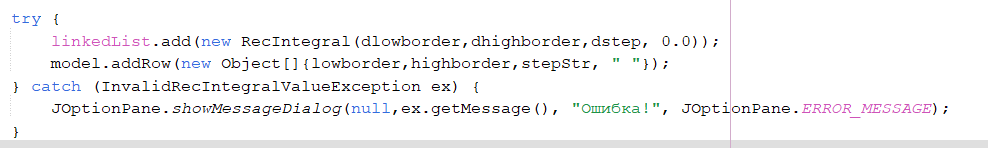
Написал класс исключения:



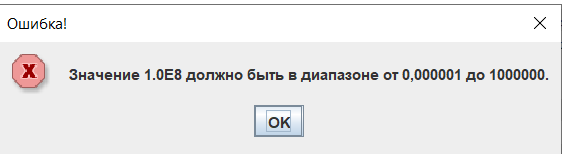
**Рис.1 – Класс исключения.**

****

**Рис.2 – Генерация исключения.**

****

**Рис.3 – Генерация исключения.**

****

**Рис.4 – Исключение.**

**Листинг программы.**

package com.mycompany.laba1;

import java.util.ArrayList;

import java.util.LinkedList;

import javax.swing.JOptionPane;

class InvalidRecIntegralValueException extends Exception {

public InvalidRecIntegralValueException(String message) {

super(message);

}

}

class RecIntegral{

double lowborder;

double highborder;

double step;

RecIntegral(double inputLowBorder, double inputHighBorder, double inputStep) throws InvalidRecIntegralValueException{

validateValue(inputLowBorder);

validateValue(inputHighBorder);

validateValue(inputStep);

lowborder = inputLowBorder;

highborder = inputHighBorder;

step = inputStep;

}

private void validateValue(double value) throws InvalidRecIntegralValueException {

if (value < 0.000001 || value > 1000000) {

throw new InvalidRecIntegralValueException("Значение " + value + " должно быть в диапазоне от 0,000001 до 1000000.");

}

}

}

public class laba1 extends javax.swing.JFrame {

int rowIndex = 0;

int selectedRow = 0;

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

public laba1() {

initComponents();

}

@SuppressWarnings("unchecked")

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

private void initComponents() {

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();

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

hbordtextfield = new javax.swing.JTextField();

steptextfield = new javax.swing.JTextField();

lbordtextfield = new javax.swing.JTextField();

jButton4 = new javax.swing.JButton();

jButton5 = new javax.swing.JButton();

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

jLabel1.setText("нижняя граница");

jLabel2.setText(" шаг");

jLabel3.setText("верхняя граница");

jButton1.setText("Вычислить");

jButton1.setActionCommand("jButton3");

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

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

jButton1ActionPerformed(evt);

}

});

jButton2.setText("Удалить");

jButton2.setActionCommand("jButton2");

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

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

jButton2ActionPerformed(evt);

}

});

jButton3.setText("Добавить");

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

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

jButton3ActionPerformed(evt);

}

});

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

new Object [][] {

},

new String [] {

"Нижняя граница", "Верхняя граница", "Шаг", "Значение"

}

));

jTable1.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT\_CURSOR));

jTable1.setShowHorizontalLines(true);

jTable1.setShowVerticalLines(true);

jScrollPane1.setViewportView(jTable1);

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

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

hbordtextfieldActionPerformed(evt);

}

});

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

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

steptextfieldActionPerformed(evt);

}

});

lbordtextfield.setName("lbordtextfield"); // NOI18N

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

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

lbordtextfieldActionPerformed(evt);

}

});

jButton4.setLabel("Заполнить");

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

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

jButton4ActionPerformed(evt);

}

});

jButton5.setLabel("Очистить");

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(layout.createSequentialGroup()

.addGap(24, 24, 24)

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

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

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

.addComponent(jLabel3, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addComponent(jLabel1))

.addGap(34, 34, 34)

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

.addComponent(lbordtextfield, javax.swing.GroupLayout.DEFAULT\_SIZE, 97, Short.MAX\_VALUE)

.addComponent(hbordtextfield)

.addComponent(steptextfield))

.addGap(36, 36, 36)

.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.PREFERRED\_SIZE, 90, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(35, 35, 35)

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

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

.addComponent(jButton5, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

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

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

.addContainerGap()

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 509, Short.MAX\_VALUE)

.addContainerGap())

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

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

.addGroup(layout.createSequentialGroup()

.addGap(32, 32, 32)

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

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton3)

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

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

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

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton2)

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

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

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

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton1)

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

.addGroup(layout.createSequentialGroup()

.addGap(50, 50, 50)

.addComponent(jButton4)

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

.addComponent(jButton5)))

.addGap(18, 18, 18)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 112, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

);

pack();

}// </editor-fold>

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

javax.swing.table.DefaultTableModel model = (javax.swing.table.DefaultTableModel) jTable1.getModel();

model.removeRow(rowIndex - 1);

rowIndex--;

}

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

javax.swing.table.DefaultTableModel model = (javax.swing.table.DefaultTableModel) jTable1.getModel();

String lowborder = lbordtextfield.getText();

//model.setValueAt(low, rowIndex, 0);

String highborder = hbordtextfield.getText();

String stepStr = steptextfield.getText();

model.addRow(new Object[]{lowborder,highborder,stepStr, " "});

rowIndex++;

}

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

javax.swing.table.DefaultTableModel model = (javax.swing.table.DefaultTableModel) jTable1.getModel();

selectedRow = jTable1.getSelectedRow();

if (selectedRow != -1) { // Проверка, выбрана ли строка

Object lowBorderObj = model.getValueAt(selectedRow, 0);

Object highBorderObj = model.getValueAt(selectedRow, 1);

Object stepObj = model.getValueAt(selectedRow, 2);

// Преобразование значений из строки в double

double lowBorder = Double.parseDouble(lowBorderObj.toString());

double highBorder = Double.parseDouble(highBorderObj.toString());

double step = Double.parseDouble(stepObj.toString());

double answer = CalculationOfValue(lowBorder, highBorder, step);

ShowTheAnswer(answer);

} else {

JOptionPane.showMessageDialog(this, "Пожалуйста, выберите строку для вычисления.", "Ошибка", JOptionPane.ERROR\_MESSAGE);

}

}

private double CalculationOfValue(Object low,Object high, Object step){

double dbllow = (Double) low;

double dblhigh = (Double) high;

double dblstep = (Double) step;

double answer = 0;

ArrayList<Double> numbers = new ArrayList<>();

//если шаг равен 0

if (dblstep == 0){

return answer;

}

//подсчет шагов

for (double currentNumber = dbllow; currentNumber <= dblhigh; currentNumber += dblstep) {

numbers.add(currentNumber);

}

//если шаг равен 1

if (numbers.isEmpty()){

answer = (Math.sin(dbllow) + Math.sin(dblhigh)) \* (dblstep / 2);

return answer;

}

for (int counter = 0; counter < numbers.size() - 1; counter++) {

double currentElement = numbers.get(counter);

double nextElement = numbers.get(counter + 1);

double stepAnswer = (dblstep / 2) \* (Math.sin(currentElement) + Math.sin(nextElement));

answer += stepAnswer;

}

//если есть промежуток меньше шага между последним элементом и верхней границей

if (dblhigh != numbers.get(numbers.size()-1)){

answer += (dblstep / 2) \* (Math.sin(numbers.get(numbers.size()-1)) + Math.sin(dblhigh));

}

return answer;

}

private void ShowTheAnswer(double answer){

javax.swing.table.DefaultTableModel model = (javax.swing.table.DefaultTableModel) jTable1.getModel();

model.setValueAt(answer, selectedRow, 3);

}

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

// TODO add your handling code here:

}

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

// TODO add your handling code here:

}

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

// TODO add your handling code here:

}

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

javax.swing.table.DefaultTableModel model = (javax.swing.table.DefaultTableModel) jTable1.getModel();

try{

linkedList.add(new RecIntegral(100000000.0,7.0,1.0));

}

catch(InvalidRecIntegralValueException exception){

JOptionPane.showMessageDialog(null,exception.getMessage(), "Ошибка!", JOptionPane.ERROR\_MESSAGE);

}

for (RecIntegral ri : linkedList) {

model.addRow(new Object[]{ri.lowborder, ri.highborder, ri.step});

}

}

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

javax.swing.table.DefaultTableModel model = (javax.swing.table.DefaultTableModel) jTable1.getModel();

model.setRowCount(0);

rowIndex = 0;

}

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new RunnableImpl());

}

// Variables declaration - do not modify

private javax.swing.JTextField hbordtextfield;

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 lbordtextfield;

private javax.swing.JTextField steptextfield;

// End of variables declaration

private static class RunnableImpl implements Runnable {

public RunnableImpl() {

}

@Override

public void run() {

new laba1().setVisible(true);

}

}

}