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»

Выполнил: ст. гр. 18ВВ2

Холодков Д.В.

Приняли:

к.т.н., доцент Карамышева Н.С.

к.т.н., доцент Юрова О.В.

Пенза 2022

**Цель работы**

Изучить механизм обработки исключительных ситуаций.

**Задание**

Вариант 1. Функция

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.

**Ход работы**

В рамках модификации приложения из предыдущей лабораторной работы были добавлены новые классы:

* ValidatorException extends Exception;
* ExceptionValidFormat extends ValidatorException;
* ExceptionValidRangeDouble extends ValidatorException;
* ValueValidator.

ValidatorException – абстрактный класс, потомок Exception и предок ExceptionValidRangeDouble и ExceptionValidFormat.

ExceptionValidRangeDouble срабатывает в случае, когда введенное пользователем значение оказывается за пределами интервала [0,000001; 1000000].

ExceptionValidFormat срабатывает в случае, когда введенное пользователем значение не может быть распознано из-за необрабатываемого формата выражения: выражение содержит несколько точек; выражение содержит буквы; в выражении встречается точка, но цифры за ней отсутствуют, и другие ситуации.

В процессе модификации лабораторной работы изменился алгоритм метода private void computeButtonMouseClicked(java.awt.event.MouseEvent evt), вызываемого при клике на кнопку “Вычислить”. Новый алгоритм представлен на рисунке 1.



Рисунок 1 – Схема алгоритма метода computeButtonMouseClicked

**Тестирование**

Окно приложения после запуска представлено на рисунке 2. В данной лабораторной работе была добавлена валидация данных, теперь приложение осуществляет контроль ввода данных и выводит окно с пояснением ошибки и выражением, которое невозможно обработать.

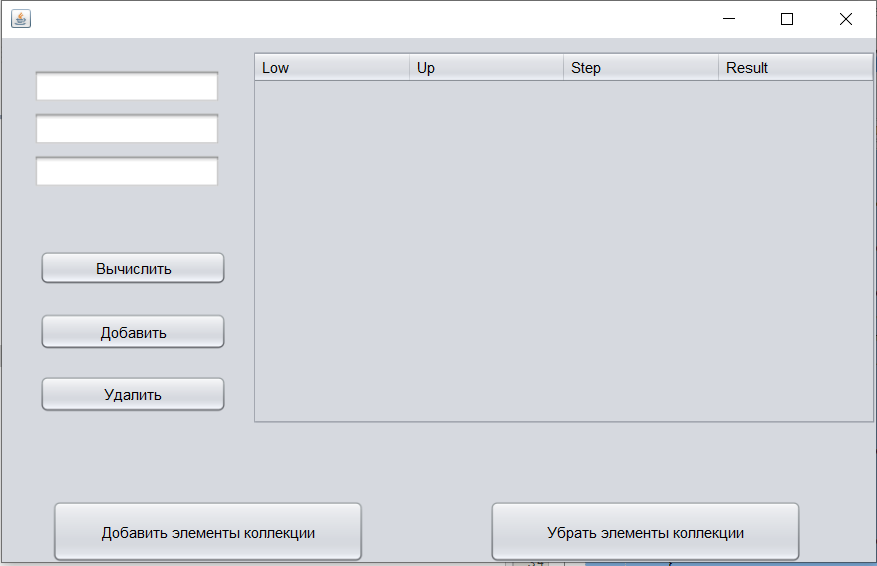


Рисунок 2 – Окно приложения после запуска

Пользовательское значение на предмет попадания в интервал [0.000001;1000000] проверяется на рисунке 3 приведен пример выхода значения за пределы интервала.

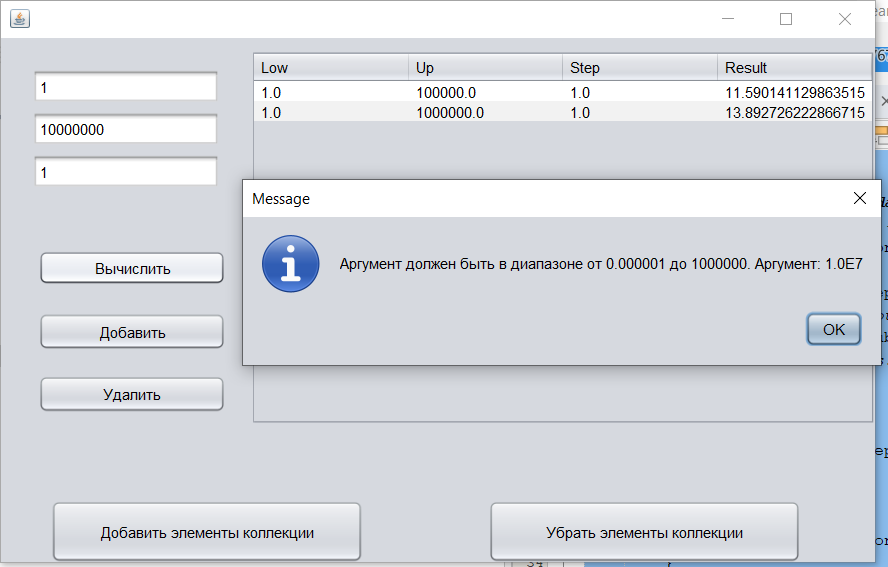


Рисунок 3 – Аргумент превысил максимальное значение

Программа не вычисляет интеграл, если пользовательское выражение не удовлетворяет регулярному выражению "[\\d]\*[.]?[\\d]+". Пример проверки изображен на рисунке 4.

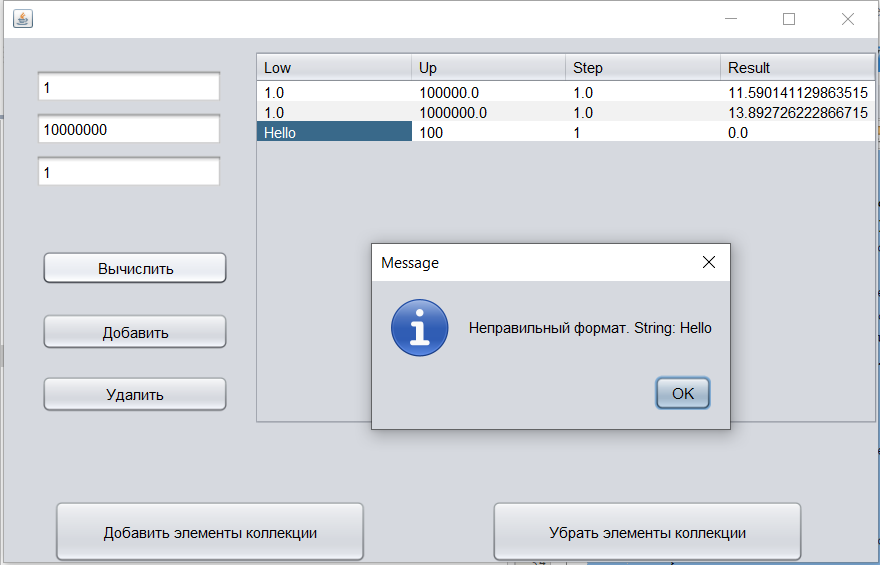


Рисунок 4 – Аргумент не соответствует регулярному выражению

**Листинг программы**

Файл Lab3.java

package com.mycompany.lab3;

import java.util.ArrayList;

import java.util.Random;

import java.util.Vector;

import javax.swing.JOptionPane;

import javax.swing.JTable;

import javax.swing.table.DefaultTableModel;

/\*\*

\*

\* @author kekos

\*/

public class Lab3 extends javax.swing.JFrame {

final ArrayList<CollectionIntegral> collection\_integral;

final Integral<Function> integral;

boolean AddIntegral=false;

/\*\*

\* Creates new form Lab1

\*/

public Lab3() {

integral = new Integral<>((x)->1/x);

collection\_integral=new ArrayList<CollectionIntegral>();

Random rand = new Random();

double left;

double right;

double step;

double result;

for(int i=0;i<10;i++){

left=rand.nextDouble()\*10;

right=rand.nextDouble()\*100;

step=rand.nextDouble();

result=integral.integrateInBounds(left, right, step);

collection\_integral.add(new CollectionIntegral(String.valueOf(left),String.valueOf(right),String.valueOf(step),String.valueOf(result)));

}

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() {

jTextField1 = new javax.swing.JTextField();

jTextField2 = new javax.swing.JTextField();

jTextField3 = new javax.swing.JTextField();

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();

jButton4 = new javax.swing.JButton();

jButton5 = new javax.swing.JButton();

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

getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

getContentPane().add(jTextField1, new org.netbeans.lib.awtextra.AbsoluteConstraints(25, 25, 150, -1));

getContentPane().add(jTextField2, new org.netbeans.lib.awtextra.AbsoluteConstraints(25, 59, 150, -1));

getContentPane().add(jTextField3, new org.netbeans.lib.awtextra.AbsoluteConstraints(25, 93, 150, -1));

jButton1.setText("Добавить");

jButton1.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

jButton1MouseClicked(evt);

}

});

getContentPane().add(jButton1, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 220, 150, 30));

jButton2.setText("Удалить");

jButton2.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

jButton2MouseClicked(evt);

}

});

getContentPane().add(jButton2, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 270, 150, 30));

jButton3.setText("Вычислить");

jButton3.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

jButton3MouseClicked(evt);

}

});

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

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

jButton3ActionPerformed(evt);

}

});

getContentPane().add(jButton3, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 170, 150, 28));

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

new Object [][] {

},

new String [] {

"Low", "Up", "Step", "Result"

}

) {

Class[] types = new Class [] {

java.lang.String.class, java.lang.String.class, java.lang.String.class, java.lang.String.class

};

public Class getColumnClass(int columnIndex) {

return types [columnIndex];

}

});

jTable1.setColumnSelectionAllowed(true);

jScrollPane1.setViewportView(jTable1);

jTable1.getColumnModel().getSelectionModel().setSelectionMode(javax.swing.ListSelectionModel.SINGLE\_SELECTION);

getContentPane().add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(200, 10, 500, 300));

jButton4.setText("Добавить элементы коллекции");

jButton4.setInheritsPopupMenu(true);

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

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

jButton4ActionPerformed(evt);

}

});

getContentPane().add(jButton4, new org.netbeans.lib.awtextra.AbsoluteConstraints(40, 370, 250, 50));

jButton5.setText("Убрать элементы коллекции");

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

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

jButton5ActionPerformed(evt);

}

});

getContentPane().add(jButton5, new org.netbeans.lib.awtextra.AbsoluteConstraints(390, 370, 250, 50));

pack();

}// </editor-fold>

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

// TODO add your handling code here:

}

private void jButton3MouseClicked(java.awt.event.MouseEvent evt) {

int selectedRow = jTable1.getSelectedRow();

double left, rigth, dx;

Vector<String> row = null;

try{

if(selectedRow != -1){

row = ((DefaultTableModel) jTable1.getModel()).getDataVector().elementAt(selectedRow);

left = ValueValidator.validateAndParse(row.get(0));

rigth = ValueValidator.validateAndParse(row.get(1));

dx = ValueValidator.validateAndParse(row.get(2));

} else {

left = ValueValidator.validateAndParse(jTextField1.getText());

rigth = ValueValidator.validateAndParse(jTextField2.getText());

dx = ValueValidator.validateAndParse(jTextField3.getText());

}

}catch(ValidatorException e){

JOptionPane.showMessageDialog(null, e.getMessage());

return;

}

double result = integral.integrateInBounds(left, rigth, dx);

if(row!=null){

row.insertElementAt(String.valueOf(result), 3);

}

else {

String[] arr = {String.valueOf(left), String.valueOf(rigth), String.valueOf(dx), String.valueOf(result)};

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

model.addRow(arr);

}

// TODO add your handling code here:

}

private void jButton2MouseClicked(java.awt.event.MouseEvent evt) {

int selectedRow = jTable1.getSelectedRow();

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

if(selectedRow >= 0){

model.removeRow(selectedRow);

}

// TODO add your handling code here:

}

private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {

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

model.addRow(new Object[]{0.0,0.0,0.0,0.0});

// TODO add your handling code here:

}

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

//ДОБАВИТЬ

if(!AddIntegral){

AddIntegral=true;

for(int i=0;i<10;i++){

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

model.addRow(new String[]{collection\_integral.get(i).left,collection\_integral.get(i).right,collection\_integral.get(i).step,collection\_integral.get(i).result});

}

}

}

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

//УБРАТЬ

if(AddIntegral){

int CountRows = jTable1.getColumnCount();

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

for(int i=0;i<10;i++){

for(int j=0;j<CountRows;j++){

boolean check=jTable1.getValueAt(j, 0)==collection\_integral.get(i).left;

check=check&(jTable1.getValueAt(j, 1)==collection\_integral.get(i).right);

check=check&(jTable1.getValueAt(j, 2)==collection\_integral.get(i).step);

check=check&(jTable1.getValueAt(j, 3)==collection\_integral.get(i).result);

if(check){

model.removeRow(j);

break;

}

}

}

AddIntegral=false;

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<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(Lab3.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Lab3.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Lab3.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Lab3.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

/\* Create and display the form \*/

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

public void run() {

new Lab3().setVisible(true);

}

});

}

// 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.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

}

Файл Integral.java

package com.mycompany.lab3;

/\*\*

\*

\* @author kekos

\* @param <T>

\*/

public class Integral<T extends Function> {

private final T function;

Integral(T function){

this.function = function;

}

public double integrateInBounds(double left, double right, double dx){

double sum = 0.0;

double currentRigth = left + dx;

while(true){

sum += (function.compute(left) + function.compute(currentRigth)) \* 0.5 \* dx;

if(currentRigth >= right){

break;

}

left += dx;

currentRigth = Math.min(currentRigth + dx, right);

}

return sum;

}

public T getFunction(){

return function;

}

}

Файл CollectionIntegral.java

package com.mycompany.lab3;

/\*\*

\*

\* @author kekos

\*/

public class CollectionIntegral {

String left;

String right;

String step;

String result;

public CollectionIntegral(String leftArg,String rightArg,String stepArg,String resultArg){

this.left=leftArg;

this.right=rightArg;

this.result=resultArg;

this.step=stepArg;

}

}

Файл ExceptionValidFormat.java

package com.mycompany.lab3;

/\*\*

\*

\* @author kekos

\*/

public class ExceptionValidFormat extends ValidatorException{

private final String doubleStr;

public ExceptionValidFormat(String str){

doubleStr = str;

}

@Override

public String getMessage(){

return "Неправильный формат. String: " + doubleStr;

}

}

Файл ValidatorException.java

package com.mycompany.lab3;

/\*\*

\*

\* @author kekos

\*/

public abstract class ValidatorException extends Exception {

@Override

public abstract String getMessage();

}

Файл ExceptionValidRangeDouble.java

package com.mycompany.lab3;

/\*\*

\*

\* @author kekos

\*/

public class ExceptionValidRangeDouble extends ValidatorException{

private final double value;

public ExceptionValidRangeDouble(double value){

this.value = value;

}

@Override

public String getMessage(){

return "Аргумент должен быть в диапазоне от 0.000001 до 1000000. Аргумент: " + Double.toString(value);

}

}

Файл ValueValidator.java

package com.mycompany.lab3;

/\*\*

\*

\* @author kekos

\*/

public class ValueValidator {

private static class Bounds{

public final static double MIN = 0.000001;

public final static double MAX = 1000000;

public final static String PATTERN = "[\\d]\*[.]?[\\d]+";

}

public static double validateAndParse(String rawValue) throws ValidatorException{

if(rawValue.isEmpty()){

throw new ExceptionValidFormat("Пустая строка");

}

rawValue = rawValue.replace(',', '.');

if(rawValue.matches(Bounds.PATTERN)){

double value = Double.parseDouble(rawValue);

if(value >= Bounds.MIN && value <= Bounds.MAX){

return value;

}

else {

throw new ExceptionValidRangeDouble(value);

}

}

else {

throw new ExceptionValidFormat(rawValue);

}

}

}

Файл Function.java

package com.mycompany.lab3;

/\*\*

\*

\* @author kekos

\*/

public interface Function {

double compute(double x);

}

**Выводы**

Изучили механизм обработки исключительных ситуаций.