Пензенский государственный университет

Кафедра «Вычислительная техника»  
  
  
  
  
  
 **ОТЧЁТ**

по лабораторной работе №3

по курсу «Программирование на языке Java»

на тему «Обработка исключительных ситуаций»

Вариант 1

Выполнили:

cтуденты группы 21ВВП2

Давкин М.А.

Борисова А.В.

Михальцова П.В.

Приняли:

Юрова О.В.

Карамышева Н.С.

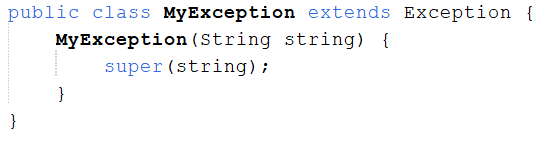
Пенза 2024

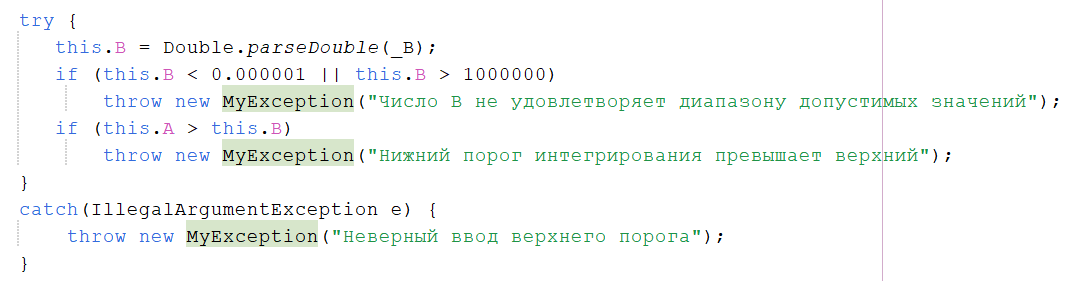
**Цель работы:** изучить механизм обработки исключительных ситуаций.

**Лабораторное задание:**

Модифицировать приложение из лабораторной работы 2, 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.

**Описание метода решения задачи:**

Создан класс MyException, который является подклассом стандартного класса исключений Exception. Конструктор принимает строку сообщения ошибки и передает ее в конструктор суперкласса с помощью ключевого слова super:

В классе RecIntegral происходит проверка значений A, B, H на соответствие диапазону допустимых значений. Если число не удовлетворяет этому диапазону, то выбрасывается исключение типа MyException с соответствующим сообщением. В блоке catch обрабатывается исключение, связанное с неверным вводом данных (например, введены буквы), также реализована проверка на превышение нижнего порога верхним:

**Листинг программы:**

Integral.java:

package llab1;

import javax.swing.table.\*;

import java.util.ArrayList;

import javax.swing.JOptionPane;

public class Integral extends javax.swing.JFrame {

ArrayList<RecIntegral> number = new ArrayList();

public Integral() {

initComponents();

}

@SuppressWarnings("unchecked")

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

private void initComponents() {

TextBottom = new javax.swing.JTextField();

TextTop = new javax.swing.JTextField();

TextStep = new javax.swing.JTextField();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

ButtonAdd = new javax.swing.JButton();

ButtonCalculate = new javax.swing.JButton();

ButtonClear = new javax.swing.JButton();

ButtonDelete = new javax.swing.JButton();

ButtonFill = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

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

setBackground(new java.awt.Color(153, 51, 0));

TextBottom.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED, null, null, null, new java.awt.Color(204, 204, 204)));

TextBottom.setMinimumSize(new java.awt.Dimension(1000, 1000));

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

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

TextBottomActionPerformed(evt);

}

});

TextTop.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED, null, null, null, new java.awt.Color(204, 204, 204)));

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

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

TextTopActionPerformed(evt);

}

});

TextStep.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED, null, null, null, new java.awt.Color(204, 204, 204)));

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

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

TextStepActionPerformed(evt);

}

});

jLabel1.setBackground(new java.awt.Color(0, 204, 204));

jLabel1.setFont(new java.awt.Font("Segoe UI Semibold", 0, 14)); // NOI18N

jLabel1.setText("Исходные данные:");

jLabel1.setBorder(javax.swing.BorderFactory.createEmptyBorder(1, 1, 1, 1));

jLabel2.setText("нижняя граница:");

jLabel3.setText("верхняя граница:");

jLabel4.setText("шаг интегрирования:");

jLabel5.setFont(new java.awt.Font("Segoe UI", 2, 12)); // NOI18N

jLabel5.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

jLabel5.setText("<html>вычисление определенного интеграла 1/x</html>");

ButtonAdd.setBackground(new java.awt.Color(255, 204, 204));

ButtonAdd.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N

ButtonAdd.setText("добавить");

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

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

ButtonAddActionPerformed(evt);

}

});

ButtonCalculate.setBackground(new java.awt.Color(255, 204, 204));

ButtonCalculate.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N

ButtonCalculate.setText("вычислить");

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

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

ButtonCalculateActionPerformed(evt);

}

});

ButtonClear.setBackground(new java.awt.Color(255, 204, 204));

ButtonClear.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N

ButtonClear.setText("очистить");

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

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

ButtonClearActionPerformed(evt);

}

});

ButtonDelete.setBackground(new java.awt.Color(255, 204, 204));

ButtonDelete.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N

ButtonDelete.setText("удалить");

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

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

ButtonDeleteActionPerformed(evt);

}

});

ButtonFill.setBackground(new java.awt.Color(255, 204, 204));

ButtonFill.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N

ButtonFill.setText("заполнить");

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

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

ButtonFillActionPerformed(evt);

}

});

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

new Object [][] {

},

new String [] {

"нижняя", "верхняя", "шаг", "результат"

}

) {

Class[] types = new Class [] {

java.lang.Object.class, java.lang.Object.class, java.lang.Object.class, java.lang.Double.class

};

boolean[] canEdit = new boolean [] {

false, true, false, false

};

public Class getColumnClass(int columnIndex) {

return types [columnIndex];

}

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

jTable1.addPropertyChangeListener(new java.beans.PropertyChangeListener() {

public void propertyChange(java.beans.PropertyChangeEvent evt) {

jTable1PropertyChange(evt);

}

});

jScrollPane1.setViewportView(jTable1);

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)

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel1)

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

.addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED\_SIZE, 88, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 317, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

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

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

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

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

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

.addGap(12, 12, 12))

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

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

.addComponent(jLabel3)

.addGroup(layout.createSequentialGroup()

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

.addComponent(jLabel4)

.addComponent(jLabel2))

.addGap(18, 18, 18)

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

.addComponent(TextTop, javax.swing.GroupLayout.PREFERRED\_SIZE, 68, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(TextStep, javax.swing.GroupLayout.PREFERRED\_SIZE, 68, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(TextBottom, javax.swing.GroupLayout.PREFERRED\_SIZE, 68, javax.swing.GroupLayout.PREFERRED\_SIZE))))

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

.addComponent(ButtonAdd, javax.swing.GroupLayout.PREFERRED\_SIZE, 94, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(108, 108, 108))))

);

layout.setVerticalGroup(

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

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

.addGroup(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(jLabel2)

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

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

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

.addComponent(jLabel3)

.addComponent(ButtonAdd)

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

.addGroup(layout.createSequentialGroup()

.addGap(19, 19, 19)

.addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED\_SIZE, 48, javax.swing.GroupLayout.PREFERRED\_SIZE)))

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

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

.addComponent(jLabel4)

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

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 25, Short.MAX\_VALUE)

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

.addGroup(layout.createSequentialGroup()

.addComponent(ButtonClear)

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

.addComponent(ButtonFill)

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

.addComponent(ButtonCalculate)

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

.addComponent(ButtonDelete)

.addGap(12, 12, 12))

.addGroup(layout.createSequentialGroup()

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 155, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap())))

);

pack();

}// </editor-fold>

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

// TODO add your handling code here:

}

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

try {

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

double H1, result = 0.0;

RecIntegral rec = number.get(jTable1.getSelectedRow());

rec.N = (rec.B - rec.A) / rec.H;

int N1 = (int)rec.N;

for (int i = 0; i < N1; i++) {

result += (1 / (rec.A + rec.H \* i) + 1 /(rec.A + rec.H \* (i + 1))) \* (rec.H / 2);

if (rec.N % 1 != 0 ) {

H1= rec.B - (rec.A + rec.H \* N1);

result += (1 / (rec.A + H1 \* i) + 1 /(rec.A + H1 \* (i + 1))) \* (H1 / 2);

}

}

rec.N = result;

jTable1.setValueAt(result,jTable1.getSelectedRow(), 3);

} catch(Throwable t) {}

}

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

// TODO add your handling code here:

}

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

// TODO add your handling code here:

}

private void jTable1PropertyChange(java.beans.PropertyChangeEvent evt) {

// TODO add your handling code here:

}

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

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

try {

number.remove(jTable1.getSelectedRow());

model.removeRow(jTable1.getSelectedRow());

} catch (Throwable t){}

}

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

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

try {

RecIntegral recIntegral = new RecIntegral(TextBottom.getText(), TextTop.getText(), TextStep.getText());

number.add(recIntegral);

TextBottom.setText("");

TextTop.setText("");

TextStep.setText("");

RecIntegral rec = number.get(number.size() - 1);

model.addRow(new Object[] {rec.A, rec.B, rec.H});

} catch(MyException ex) {

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

}

}

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

((DefaultTableModel)jTable1.getModel()).setRowCount(0);

}

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

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

((DefaultTableModel)jTable1.getModel()).setRowCount(0);

if (jTable1.getRowCount() != number.size())

for (RecIntegral rec:number)

model.addRow(new Object[] {rec.A, rec.B, rec.H, rec.N});

}

public static void main(String args[]) {

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

public void run() {

new Integral().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton ButtonAdd;

private javax.swing.JButton ButtonCalculate;

private javax.swing.JButton ButtonClear;

private javax.swing.JButton ButtonDelete;

private javax.swing.JButton ButtonFill;

private javax.swing.JTextField TextBottom;

private javax.swing.JTextField TextStep;

private javax.swing.JTextField TextTop;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTable jTable1;

// End of variables declaration

}

RecIntegral.java:

package llab1;

public class RecIntegral

{

double A, B, H, N;

public RecIntegral(String \_A, String \_B, String \_H) throws MyException {

try {

this.A = Double.parseDouble(\_A);

if (this.A < 0.000001 || this.A > 1000000)

throw new MyException("Число A не удовлетворяет диапазону допустимых значений");

}

catch(IllegalArgumentException e) {

throw new MyException("Неверный ввод нижнего порога");

}

try {

this.B = Double.parseDouble(\_B);

if (this.B < 0.000001 || this.B > 1000000)

throw new MyException("Число B не удовлетворяет диапазону допустимых значений");

if (this.A > this.B)

throw new MyException("Нижний порог интегрирования превышает верхний");

}

catch(IllegalArgumentException e) {

throw new MyException("Неверный ввод верхнего порога");

}

try {

this.H = Double.parseDouble(\_H);

if (this.H < 0.000001 || this.H > 1000000)

throw new MyException("Число H не удовлетворяет диапазону допустимых значений");

}

catch(IllegalArgumentException e) {

throw new MyException("Неверный ввод шага интегрирования");

}

}

public RecIntegral() {

A = 0;

B = 0;

H = 0;

}

}

MyException.java:

package llab1;

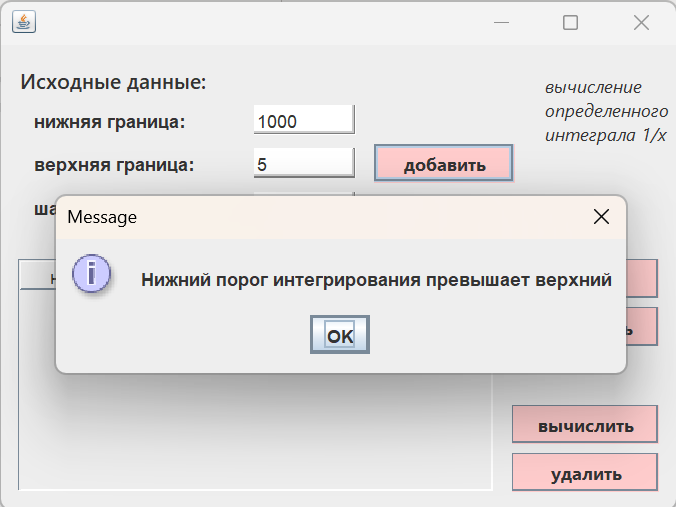
public class MyException extends Exception {

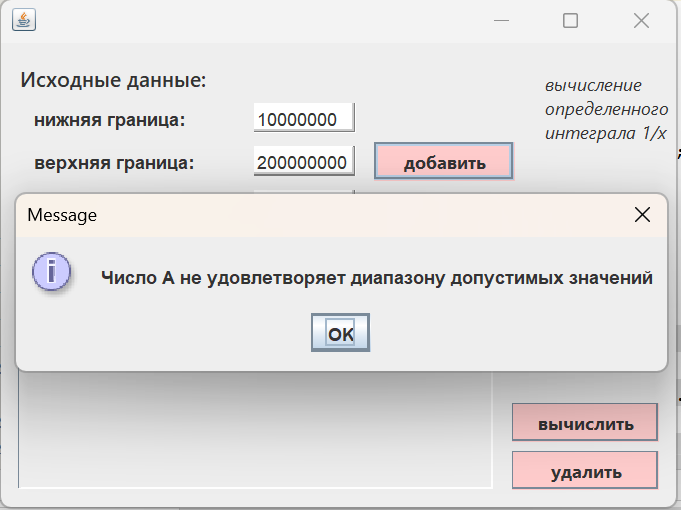
MyException(String string) {

super(string);

}

}

**Результаты работы программы:**

****

**Вывод:** изучен механизм обработки исключительных ситуаций: создан класс – наследник Exception, встроены проверки вводимых данных на формат ввода, диапазон допустимых значений и превышение нижнего порога верхним.