Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №5

по курсу «Разработка кроссплатформенных приложений»

на тему: «Многопоточность в JAVA»

**Выполнили студенты группы 21ВВВ1:**

Кривенков И.В.

Хорошильцев Е.А.

**Приняли:**

Юрова О.В.

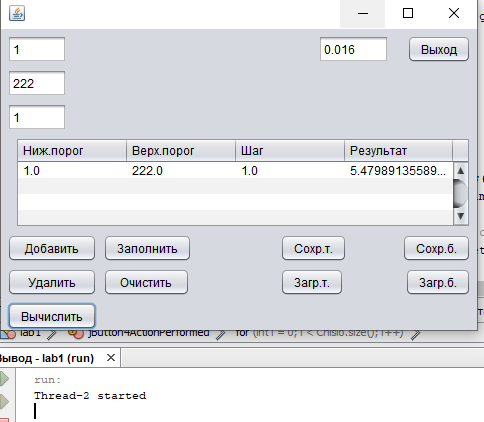
Карамышева Н.С.

Пенза 2024

**Цель работы:**

Научиться создавать многопоточные приложения с использованием стандартных средств языка JAVA.

**Результат работы программы:**



**Вывод:**

Выполняя данную лабораторную работу, мы научились создавать многопоточные приложения с использованием стандартных средств языка JAVA.

Листинг программы:

/\*

\* 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 javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.\*;

import java.util.LinkedList;

import java.util.Scanner;

import java.io.Serializable;

import java.util.ArrayList;

import java.util.Objects;

import java.io.EOFException;

import java.lang.Thread;

import java.util.Collections;

import java.util.List;

/\*\*

\*

\* @author maxwe

\*/

public class lab1 extends javax.swing.JFrame {

private double upperBound;

private double lowerBound;

private double step;

/\*\* Creates new form lab1 \*/

public lab1() {

initComponents();

}

public class Invalid extends Exception{

public Invalid(String message) {

//super(message);

JOptionPane.showMessageDialog(null, message, "Ошибка", JOptionPane.ERROR\_MESSAGE);

}

}

/\*\* 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();

jButton4 = new javax.swing.JButton();

jButton5 = new javax.swing.JButton();

jButton6 = new javax.swing.JButton();

jButton7 = new javax.swing.JButton();

jButton8 = new javax.swing.JButton();

jButton9 = new javax.swing.JButton();

jButton10 = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

jTextField4 = new javax.swing.JTextField();

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

jTextField1.setText("Ниж");

jTextField2.setText("Верх");

jTextField3.setText("Шаг");

jButton1.setText("Выход");

jButton2.setText("Добавить");

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

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

jButton2ActionPerformed(evt);

}

});

jButton3.setText("Удалить");

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

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

jButton3ActionPerformed(evt);

}

});

jButton4.setText("Вычислить");

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

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

jButton4ActionPerformed(evt);

}

});

jButton5.setText("Заполнить");

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

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

jButton5ActionPerformed(evt);

}

});

jButton6.setText("Очистить");

jButton6.addActionListener(new java.awt.event.ActionListener() {

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

jButton6ActionPerformed(evt);

}

});

jButton7.setText("Сохр.т.");

jButton7.addActionListener(new java.awt.event.ActionListener() {

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

jButton7ActionPerformed(evt);

}

});

jButton8.setText("Загр.т.");

jButton8.addActionListener(new java.awt.event.ActionListener() {

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

jButton8ActionPerformed(evt);

}

});

jButton9.setText("Сохр.б.");

jButton9.addActionListener(new java.awt.event.ActionListener() {

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

jButton9ActionPerformed(evt);

}

});

jButton10.setText("Загр.б.");

jButton10.addActionListener(new java.awt.event.ActionListener() {

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

jButton10ActionPerformed(evt);

}

});

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

new Object [][] {

{null, null, null, null},

{null, null, null, null},

{null, null, null, null},

{null, null, null, null}

},

new String [] {

"Ниж.порог", "Верх.порог", "Шаг", "Результат"

}

));

jScrollPane1.setViewportView(jTable1);

jTextField4.setText("jTextField4");

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()

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

.addComponent(jTextField1)

.addComponent(jTextField2, javax.swing.GroupLayout.DEFAULT\_SIZE, 60, Short.MAX\_VALUE)

.addComponent(jTextField3))

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

.addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(jButton1))

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

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

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

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

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

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

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

.addComponent(jButton5)

.addComponent(jButton6, javax.swing.GroupLayout.PREFERRED\_SIZE, 87, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(88, 88, 88)

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

.addGroup(layout.createSequentialGroup()

.addComponent(jButton8)

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

.addComponent(jButton10))

.addGroup(layout.createSequentialGroup()

.addComponent(jButton7)

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

.addComponent(jButton9))))

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

.addGap(0, 8, Short.MAX\_VALUE)

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

.addContainerGap())

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

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

.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(jButton1)

.addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

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

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

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

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

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

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 91, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

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

.addComponent(jButton2)

.addComponent(jButton7)

.addComponent(jButton5)

.addComponent(jButton9))

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

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

.addComponent(jButton3)

.addComponent(jButton6)

.addComponent(jButton8)

.addComponent(jButton10))

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

.addComponent(jButton4))

);

pack();

}// </editor-fold>

class ReSUS {

private double result = 0.0;

public synchronized void sum(double input\_res) {

result += input\_res;

}

public synchronized double get() {

return result;

}

}

public class RecIntegral implements Serializable {

private static final long serVerUID = 1L;

public String step ;

public String b;

public String a;

public String res;

public RecIntegral (String step, String b, String a, String res){

this.step = step;

this.b = b;

this.a = a;

this.res = res;

}

public void getComp(String step, String b, String a, String res){

if(Objects.equals(step, this.step) && Objects.equals(b, this.b) && Objects.equals(a, this.a)){

this.res = res;

}

}

}

public class MyThread extends Thread {

private double upperBound;

private double lowerBound;

private double step;

private ReSUS result;

public MyThread(double upperBound, double lowerBound, double step, ReSUS result) {

this.upperBound = upperBound;

this.lowerBound = lowerBound;

this.step = step;

this.result = result;

}

public void run() {

double res = 0.0;

for (double i = lowerBound; i < upperBound; i += step) {

if ((i + step) < upperBound) {

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

} else {

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

}

}

result.sum(res);

}

}

private ArrayList<RecIntegral> Chislo = new ArrayList<>();

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

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

try {

double text1 = Double.parseDouble(jTextField1.getText());

double text2 = Double.parseDouble(jTextField2.getText());

double text3 = Double.parseDouble(jTextField3.getText());

if ((text1 < 0.000001) || (text1 > 1000000) || (text2 < 0.000001) || (text2 > 1000000) || (text3 < 0.000001) || (text3 > 1000000)) {

throw new Invalid("Число вне диапазона [0.000001-1000000]");

}

else if (text2 < text3) {

throw new Invalid("Верхний предел меньше нижнего");

}

else if (text1 > text2 - text3){

throw new Invalid("Шаг превышает дозволенность");

}

model.insertRow(0, new Object[]{text1, text2, text3});

RecIntegral obj = new RecIntegral(String.valueOf(text1), String.valueOf(text2), String.valueOf(text3), "");

Chislo.add(obj);

} catch (Invalid ex) { //?

} catch (NumberFormatException ex) {

JOptionPane.showMessageDialog(null, "Ошибка парсинга текста в Double", "Ошибка", JOptionPane.ERROR\_MESSAGE);

}

}

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

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

final int row = jTable1.getSelectedRow();

try {

model.removeRow(row);

Chislo.remove(row);

} catch (Exception ex){}

}

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

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

int row = 0;

// Создаем и запускаем потоки для вычислений

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

RecIntegral rec = Chislo.get(i);

double upperBound = Double.parseDouble(rec.b);

double lowerBound = Double.parseDouble(rec.a);

double step = Double.parseDouble(rec.step);

ReSUS result = new ReSUS();

MyThread thread = new MyThread(upperBound, lowerBound, step, result);

double startTime = System.currentTimeMillis(); // Засекаем время начала выполнения потока

thread.start();

System.out.printf("%s started\n", thread.getName());

// Ждем завершения потока

try {

thread.join();

} catch (InterruptedException e) {

e.printStackTrace();

}

double endTime = System.currentTimeMillis(); // Засекаем время окончания выполнения потока

double executionTime = (endTime - startTime) / 1000; // Вычисляем время выполнения в секундах

// Обновляем результат в соответствующей строке таблицы

rec.res = String.valueOf(thread.result.get());

model.setValueAt(rec.res, row, 3);

jTextField4.setText(String.valueOf(executionTime)); // Выводим время выполнения в секундах в jTextField4

}

}

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

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

for (RecIntegral chislo : Chislo) {

Object[] rowData = {chislo.b, chislo.a, chislo.step, chislo.res};

model.addRow(rowData); // Добавление данных в таблицу

}

}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

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

model.setRowCount(0); // Очистка таблицы

}

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {

String path = "";

JFileChooser fileChooser = new JFileChooser();

int returnVal = fileChooser.showOpenDialog(fileChooser);

if(returnVal == JFileChooser.APPROVE\_OPTION) {

path = fileChooser.getSelectedFile().getPath();

}

try {

FileWriter fw = new FileWriter(path);

for(RecIntegral step: Chislo) {

fw.write(step.step +" "+ step.b +" "+ step.a +" "+ step.res + "\n");

}

fw.close();

} catch (IOException ex) {

throw new RuntimeException(ex);

}

}

private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {

String path = "";

JFileChooser fileChooser = new JFileChooser();

int returnVal = fileChooser.showOpenDialog(fileChooser);

if(returnVal == JFileChooser.APPROVE\_OPTION) {

path = fileChooser.getSelectedFile().getPath();

}

try {

FileReader fr = new FileReader(path);

BufferedReader br = new BufferedReader(fr);

Chislo.clear();

String line = br.readLine();

while (line != null)

{

String[] strs = line.trim().split("\\s+");

if(strs.length != 3)

Chislo.add(new RecIntegral(strs[0], strs[1], strs[2], strs[3]));

else

Chislo.add(new RecIntegral(strs[0], strs[1], strs[2], ""));

line = br.readLine();

}

} catch (IOException ex) {

throw new RuntimeException(ex);

}

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

model.setRowCount(0); // Очищаем таблицу перед заполнением новыми данными

for (RecIntegral chislo : Chislo) {

Object[] rowData = {chislo.a, chislo.b, chislo.step, ""};

model.addRow(rowData); // Добавляем данные в таблицу

}

}

private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {

String path = "";

JFileChooser fileChooser = new JFileChooser();

int returnVal = fileChooser.showOpenDialog(fileChooser);

if(returnVal == JFileChooser.APPROVE\_OPTION) {

path = fileChooser.getSelectedFile().getPath();

}

try {

FileOutputStream outputStream = new FileOutputStream(path);

ObjectOutputStream objectOutputStream = new ObjectOutputStream(outputStream);

for(RecIntegral step: Chislo) {

objectOutputStream.writeObject(step.step +" "+ step.b +" "+ step.a +" " + "\n");

}

objectOutputStream.close();

outputStream.close();

JOptionPane.showMessageDialog(null, "Успешно сохранено", "Ура", JOptionPane.INFORMATION\_MESSAGE);

} catch (IOException ex) {

throw new RuntimeException(ex);

}

}

private void jButton10ActionPerformed(java.awt.event.ActionEvent evt) {

String path = "";

JFileChooser fileChooser = new JFileChooser();

int returnVal = fileChooser.showOpenDialog(fileChooser);

if(returnVal == JFileChooser.APPROVE\_OPTION) {

path = fileChooser.getSelectedFile().getPath();

}

try {

FileInputStream fileInputStream = new FileInputStream(path);

BufferedInputStream bufferedInputStream = new BufferedInputStream(fileInputStream);

ObjectInputStream objectInputStream = new ObjectInputStream(bufferedInputStream);

while (true) {

try {

Object obj = objectInputStream.readObject();

if (obj instanceof RecIntegral) {

RecIntegral rec = (RecIntegral) obj;

Chislo.add(rec);

}

} catch (EOFException ex) {

break; // Выход из цикла в конце файла

}

}

} catch (EOFException ex){

}

catch (IOException | ClassNotFoundException ex) {

throw new RuntimeException(ex);

}

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

model.setRowCount(0); // Очищаем таблицу перед заполнением новыми данными

for (RecIntegral chislo : Chislo) {

Object[] rowData = {chislo.a, chislo.b, chislo.step, ""};

model.addRow(rowData); // Добавляем данные в таблицу

}

}

/\*\*

\* @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(lab1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(lab1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(lab1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(lab1.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 lab1().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton10;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton3;

private javax.swing.JButton jButton4;

private javax.swing.JButton jButton5;

private javax.swing.JButton jButton6;

private javax.swing.JButton jButton7;

private javax.swing.JButton jButton8;

private javax.swing.JButton jButton9;

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;

private javax.swing.JTextField jTextField4;

// End of variables declaration

}