Министерство образования и науки Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №5

по курсу «Программирование на языке Java»

на тему «Многопоточность в Java»

Вариант 4

Выполнили:

ст. гр. 19ВВ3

Яцков В.В

Татаршев А.А

Проверил:

Юрова О.В

Пенза 2022

Цель работы

Научиться создавать многопоточные приложения c использованием стандартных средств языка Java.

Лабораторное задание

Модифицировать приложение из предыдущей лабораторной работы, реализовав вычисление определенного интеграла в нескольких дополнительных потоках (число потоков определяется номером варианта), снимая нагрузку с основного потока и предотвращая "подвисание" графического интерфейса. Реализовать многопоточность путем наследования от класса Thread.

Описание метода решения задачи

Был добавлен класс «IntCalc», унаследованный от Thread, в котором метод «run» был переопределен для расчета интеграла. Также в интерфейс была добавлена кнопка, при нажатии на которую в программе создавались 4 нити, каждая из которых выполняла часть вычислений. Исходные данные для вычисления присылаются в конструктор нити. После подсчета все результаты складываются и выводятся в окно ответа на интерфейсе. При больших объемах вычислений кнопка расчета блокируется, до получения результата.

Листинг

**Класс LogFrame:**

package Lab1;

import java.io.BufferedInputStream;

import java.io.BufferedOutputStream;

import java.io.BufferedReader;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.util.LinkedList;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

import java.math.MathContext;

import java.util.ArrayList;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFileChooser;

/\*\*

\*

\* @author Owl

\*/

public class LogFrame extends javax.swing.JFrame {

/\*\*

\* Creates new form LogFrame

\*/

public LogFrame() {

initComponents();

}

public LinkedList<RecIntegral> collection = new LinkedList<RecIntegral>();

/\*\*

\* 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() {

jPanel1 = new javax.swing.JPanel();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

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();

jButton4 = new javax.swing.JButton();

jButton5 = new javax.swing.JButton();

jLabel1 = new javax.swing.JLabel();

jButton6 = new javax.swing.JButton();

jButton7 = new javax.swing.JButton();

jButton3 = new javax.swing.JButton();

jButton8 = new javax.swing.JButton();

jButton9 = new javax.swing.JButton();

jPanel1.setBorder(new javax.swing.border.MatteBorder(null));

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 100, Short.MAX\_VALUE)

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 100, Short.MAX\_VALUE)

);

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

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

setPreferredSize(new java.awt.Dimension(700, 380));

setSize(new java.awt.Dimension(700, 500));

jButton1.setText("Добавить");

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

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

jButton1ActionPerformed(evt);

}

});

jButton2.setText("Удалить");

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

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

jButton2ActionPerformed(evt);

}

});

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

new Object [][] {

},

new String [] {

"Нижняя граница интеграла", "Верхняя граница интеграла", "Шаг интегрирования", "Результат"

}

) {

boolean[] canEdit = new boolean [] {

false, false, false, false

};

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

jTable1.getTableHeader().setReorderingAllowed(false);

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);

}

});

jTextField2.setToolTipText("");

jButton4.setText("Заполнить");

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

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

jButton4ActionPerformed(evt);

}

});

jButton5.setText("Отчистить");

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

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

jButton5ActionPerformed(evt);

}

});

jLabel1.setText(" С ДНЕМ РОЖДЕНИЯ!");

jButton6.setText("Вычислить");

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

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

jButton6ActionPerformed(evt);

}

});

jButton7.setText("Сохранить в txt");

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

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

jButton7ActionPerformed(evt);

}

});

jButton3.setText("Заполнить из txt");

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

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

jButton3ActionPerformed(evt);

}

});

jButton8.setText("Сохранить в bin");

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

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

jButton8ActionPerformed(evt);

}

});

jButton9.setText("Заполнить в bin");

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

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

jButton9ActionPerformed(evt);

}

});

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

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

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

.addComponent(jScrollPane1)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, 155, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(12, 12, 12)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, 155, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, 155, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT\_SIZE, 204, Short.MAX\_VALUE)

.addContainerGap())

.addGroup(layout.createSequentialGroup()

.addGap(25, 25, 25)

.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(jButton1, javax.swing.GroupLayout.DEFAULT\_SIZE, 100, Short.MAX\_VALUE))

.addGap(26, 26, 26)

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

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 99, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton5, javax.swing.GroupLayout.PREFERRED\_SIZE, 99, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(31, 31, 31)

.addComponent(jButton6)

.addGap(41, 41, 41)

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

.addComponent(jButton7)

.addComponent(jButton3))

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

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

.addComponent(jButton9)

.addComponent(jButton8))

.addGap(51, 51, 51))

);

layout.setVerticalGroup(

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

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

.addContainerGap()

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

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

.addComponent(jLabel1))

.addGap(14, 14, 14)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 252, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

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

.addComponent(jButton1)

.addComponent(jButton2)

.addComponent(jButton6)

.addComponent(jButton7)

.addComponent(jButton8))

.addGap(26, 26, 26)

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

.addComponent(jButton4)

.addComponent(jButton5)

.addComponent(jButton3)

.addComponent(jButton9))

.addContainerGap(18, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

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

// TODO add your handling code here:

}

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

// TODO add your handling code here:

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

if(!(jTextField1.getText().isEmpty())&&!(jTextField2.getText().isEmpty())&&!(jTextField3.getText().isEmpty()))

{

try{

collection.add(new RecIntegral(jTextField1.getText(),jTextField2.getText(),jTextField3.getText()));

model.addRow(new Object[]{jTextField1.getText(),jTextField2.getText(),jTextField3.getText()});

}catch(Errors err){

JOptionPane.showMessageDialog(null, err.showErrorMessage());

}catch(Err err){

JOptionPane.showMessageDialog(null, err.showErrorMessage());

}

jTextField1.setText("");

jTextField2.setText("");

jTextField3.setText("");

}

else{

JOptionPane.showMessageDialog(this, "Введите значения");

}

}

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

// TODO add your handling code here:

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

if(jTable1.getSelectedRowCount() == 1){

collection.remove(jTable1.getSelectedRow());

model.removeRow(jTable1.getSelectedRow());

}else{

if(jTable1.getRowCount()==0){

JOptionPane.showMessageDialog(this, "Таблица пуста");

}else{

JOptionPane.showMessageDialog(this, "Выберите строку, которую хотите удалить");

}

}

}

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

// TODO add your handling code here:

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

collection.forEach((g) -> {RecIntegral l = (RecIntegral)g; model.addRow(new Object[]{ l.step, l.lowerBound, l.upperBound});});

}

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

// TODO add your handling code here:

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

model.setRowCount(0);

}

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

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

// float answer = 0;

// if(jTable1.getSelectedRowCount() == 1){

// for(float i = Float.parseFloat(jTable1.getValueAt(jTable1.getSelectedRow(), 0).toString()); i < Float.parseFloat(jTable1.getValueAt(jTable1.getSelectedRow(), 1).toString()); i += Float.parseFloat(jTable1.getValueAt(jTable1.getSelectedRow(), 2).toString()))

// {

// answer += ((Math.tan(i)+Math.tan(i + Float.parseFloat(jTable1.getValueAt(jTable1.getSelectedRow(), 2).toString())))/2)\*Float.parseFloat(jTable1.getValueAt(jTable1.getSelectedRow(), 2).toString());

// }

// jTable1.setValueAt(answer, jTable1.getSelectedRow(), 3);

// }else{

// if(jTable1.getRowCount()==0){

// JOptionPane.showMessageDialog(this, "Таблица пуста");

// }

// else{

// int counRow = model.getRowCount();

// for(int j=0; j<counRow; j++){

// for(float i = Float.parseFloat(jTable1.getValueAt(j, 0).toString()); i < Float.parseFloat(jTable1.getValueAt(j, 1).toString()); i += Float.parseFloat(jTable1.getValueAt(j, 2).toString()))

// {

// answer += ((Math.tan(i)+Math.tan(i + Float.parseFloat(jTable1.getValueAt(j, 2).toString())))/2)\*Float.parseFloat(jTable1.getValueAt(j, 2).toString());

// }

// jTable1.setValueAt(answer, j, 3);

// }

// }

// }

IntCalc cal = new IntCalc(Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 0).toString()),Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 1).toString())\*(float)0.25,Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 2).toString()));

IntCalc cu = new IntCalc(Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 1).toString())\*(float)0.25,Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 1).toString())\*(float)0.5,Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 2).toString()));

IntCalc la = new IntCalc(Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 1).toString())\*(float)0.5,Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 1).toString())\*(float)0.75,Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 2).toString()));

IntCalc te = new IntCalc(Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 1).toString())\*(float)0.75,Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 1).toString()),Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 2).toString()));

cal.start();

cu.start();

la.start();

te.start();

Thread t = new Thread()

{

public void run()

{

jButton4.setEnabled(false);

try

{

cal.join();

cu.join();

la.join();

te.join();

}

catch (InterruptedException ex)

{

Logger.getLogger(LogFrame.class.getName()).log(Level.SEVERE, null, ex);

}

jTable1.setValueAt(String.valueOf(cal.subanswer + cu.subanswer + la.subanswer + te.subanswer), jTable1.getSelectedRow(), 3);

// jTextField1.setText(String.valueOf(cal.subanswer + cu.subanswer + la.subanswer + te.subanswer));

jButton4.setEnabled(true);

}

};

t.start();

}

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

JFileChooser openInputFile = new JFileChooser();

int dlgWnd = openInputFile.showDialog(null, "Сохранить в тектсовый файл");

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

File fileOpen = openInputFile.getSelectedFile();

try(FileWriter writeFile = new FileWriter(fileOpen, false)){

collection.forEach((s)->{

RecIntegral rc = (RecIntegral)s;

try{

writeFile.write(String.valueOf(rc.lowerBound) + " " + String.valueOf(rc.upperBound) + " " + String.valueOf(rc.step) + "\n");

}catch(IOException e){

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

}

});

}catch(IOException e){

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

}

}

}

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

collection.clear();

double downValue;

double upValue;

double step;

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

table.setRowCount(0);

JFileChooser openInputFile = new JFileChooser();

int dlgWnd = openInputFile.showDialog(null, "Открыть тектсовый файл");

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

File fileOpen = openInputFile.getSelectedFile();

try(FileReader readFile = new FileReader(fileOpen)){

String buf; BufferedReader rBuf = new BufferedReader(readFile);

while ((buf = rBuf.readLine()) != null){

String[] v = buf.split(" ");

downValue = Double.parseDouble(v[0]);

upValue = Double.parseDouble(v[1]);

step = Double.parseDouble(v[2]);

try{

RecIntegral obj = new RecIntegral(String.valueOf(downValue), String.valueOf(upValue), String.valueOf(step));

collection.add(obj);

}catch(Errors err){

JOptionPane.showMessageDialog(null, err.showErrorMessage());

} catch (Err ex) {

Logger.getLogger(LogFrame.class.getName()).log(Level.SEVERE, null, ex);

}

}

}catch(IOException e){

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

}

}

collection.forEach((s)->{

RecIntegral rc = (RecIntegral)s;

table.addRow(new Object[]{rc.lowerBound, rc.upperBound, rc.step});

});

}

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

collection.clear();

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

table.setRowCount(0);

JFileChooser openInputFile = new JFileChooser();

int dlgWnd = openInputFile.showDialog(null, "Открыть бинарный файл");

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

File fileOpen = openInputFile.getSelectedFile();

ObjectInputStream arrayRead = null;

try{

arrayRead = new ObjectInputStream(new BufferedInputStream(new FileInputStream(fileOpen)));

collection = (LinkedList)arrayRead.readObject();

}catch(IOException e){

e.printStackTrace();

}catch(ClassNotFoundException classErr){

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

}finally{

try{

arrayRead.close();

}catch(IOException e){

e.printStackTrace();

}

}

}

collection.forEach((s)->{

RecIntegral rc = (RecIntegral)s;

table.addRow(new Object[]{rc.lowerBound, rc.upperBound, rc.step});

});

}

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

JFileChooser openInputFile = new JFileChooser();

int dlgWnd = openInputFile.showDialog(null, "Сохранить в бинарный файл");

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

File fileOpen = openInputFile.getSelectedFile();

ObjectOutputStream arrayWrite = null;

try{

arrayWrite = new ObjectOutputStream(new BufferedOutputStream(new FileOutputStream(fileOpen)));

arrayWrite.writeObject(collection);

}catch(IOException e){

e.printStackTrace();

}finally{

try{

arrayWrite.close();

}catch(IOException e){

e.printStackTrace();

}

}

}

}

/\*\*

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

} catch (InstantiationException ex) {

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

} catch (IllegalAccessException ex) {

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

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(LogFrame.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 LogFrame().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.JButton jButton6;

private javax.swing.JButton jButton7;

private javax.swing.JButton jButton8;

private javax.swing.JButton jButton9;

private javax.swing.JLabel jLabel1;

private javax.swing.JPanel jPanel1;

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

}

**Класс IntCalc.java**

import javax.swing.JTable;

import javax.swing.JTextField;

public class IntCalc extends Thread{

float low, high, step;

float subanswer = 0;

IntCalc(float low, float high, float step)

{

this.low = low;

this.high = high;

this.step = step;

}

public void run()

{

for(float i = low; i < high; i += step)

{

subanswer += ((Math.tan(i)+Math.tan(i + step))/2)\*step;

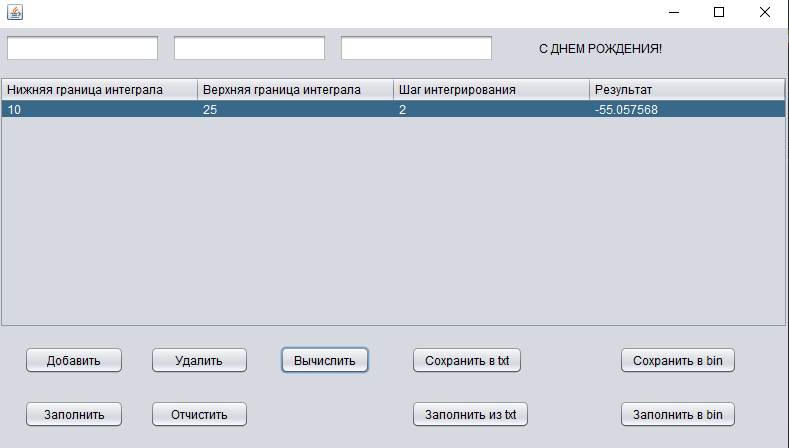
}

}

}

Результаты работы программы

Результаты работы программы показаны на рисунке 1.

  
Рисунок 1 — Результаты работы программы при загрузке

Выводы

В ходе выполнения лабораторной работы была доработана программа, позволяющая научиться создавать многопоточные приложения c использованием стандартных средств языка Java.