Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №5

по курсу «Программирование на языке JAVA»

на тему «Многопоточность Java»

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**

Научиться создавать многопоточные приложения c использованием стандартных средств языка Java.

**Задание**

Вариант 1. Функция

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. Варианты с номерами до 5 включительно реализуют многопоточность путем наследования от класса Thread, остальные реализуют интерфейс Runnable.

**Ход работы**

В рамках доработки приложения из предыдущей лабораторной работы были добавлен новый класс: ThreadCalculateIntegral extends Thread.

В основе лабораторной работы лежит свойство определенного интеграла: если функция вида интегрируема на интервале получаем, что .

Экземпляр класса ThreadCalculateIntegral вычисляет часть интеграла. Затем нить вызывает метод synchronized void addResult(double res), который добавляет вычисленный интеграл к переменной. Метод является синхронным – в один момент времени там находиться только одна нить.

**Тестирование**

В данной лабораторной работе вычисление интеграла стало распределенным, но функция и метод вычисления остались те же. Следовательно, результаты, полученные в этой модификации лабораторной работы не должны разниться с результатами вычислений в предыдущих лабораторных работах. Однако некоторая погрешность допустима так как происходит деление интеграла на равные части по количеству нитей-вычислителей. Тестирование проводилось на 4 нитях. Окно приложения после запуска представлено на рисунке 1.

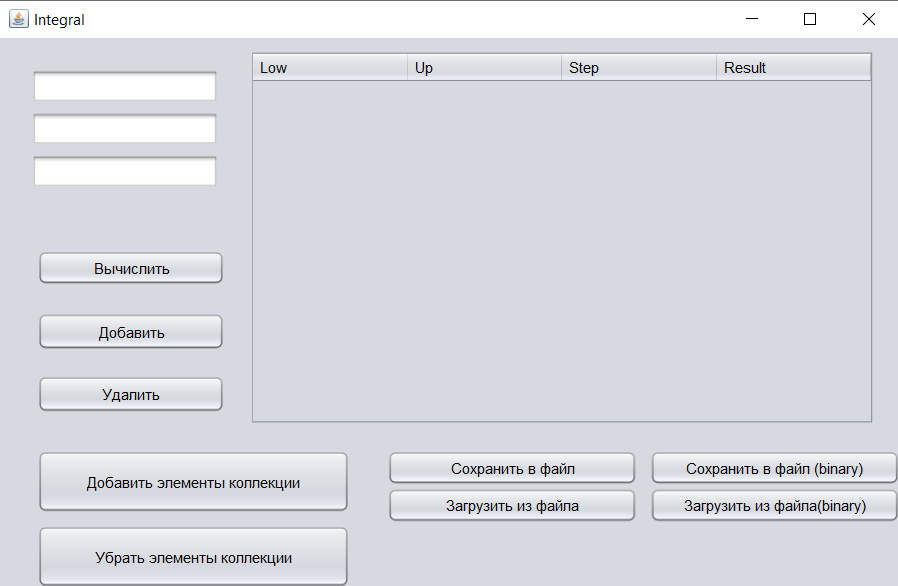


Рисунок 1 – Окно приложения после запуска

На рисунке 2 приведен фрагмент отчёта по 2 лабораторной работе с тестированием программы и проверкой результата на калькуляторе. На рисунке 3 представлены результаты вычисления тестовых примеров в текущей программе.

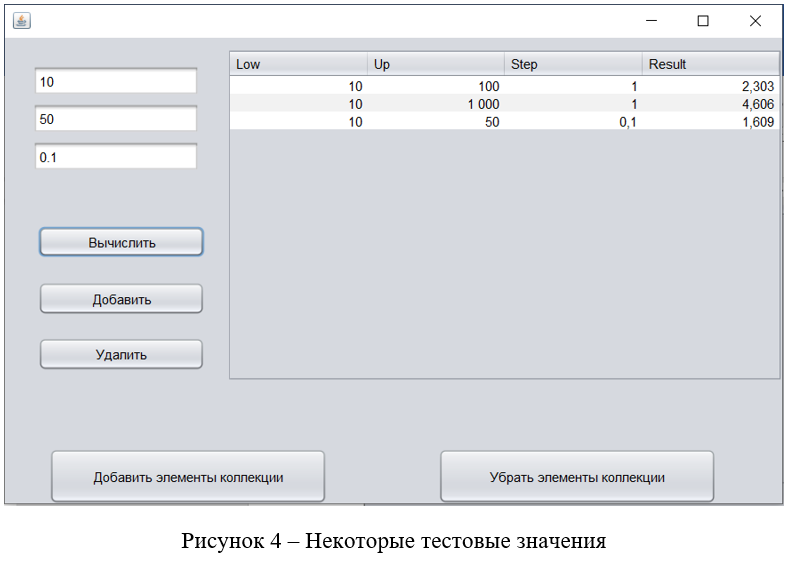


Рисунок 3 – Тестовые значения в 2 лабораторной работе

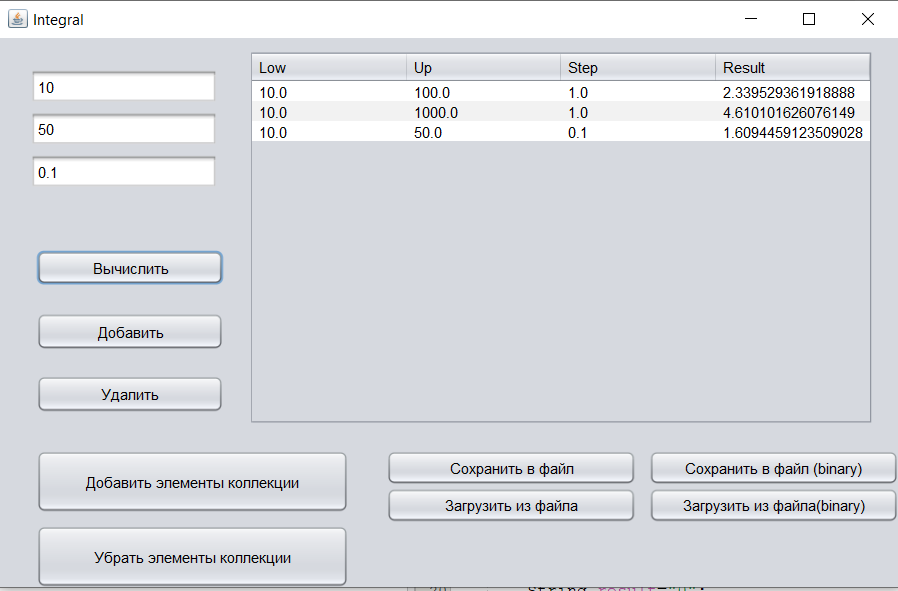


Рисунок 4 – Результаты вычисления тестовых примеров

**Листинг программы**

Файл Lab5.java

package com.mycompany.lab5;

import java.io.DataOutputStream;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.FileWriter;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.io.StringWriter;

import java.util.ArrayList;

import java.util.Random;

import java.util.Scanner;

import java.util.Vector;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFileChooser;

import javax.swing.JOptionPane;

import javax.swing.JTable;

import javax.swing.table.DefaultTableModel;

/\*\*

\*

\* @author kekos

\*/

public class Lab5 extends javax.swing.JFrame {

final ArrayList<CollectionIntegral> collection\_integral;

final Integral<Function> integral;

boolean AddIntegral=false;

/\*\*

\* Creates new form Lab1

\*/

public Lab5() {

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();

Add\_row\_Button = new javax.swing.JButton();

Delete\_row\_Button = new javax.swing.JButton();

Calculate\_Button = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

Add\_collection\_Button = new javax.swing.JButton();

Remove\_collection\_Button = new javax.swing.JButton();

Save\_as\_text\_Button = new javax.swing.JButton();

Load\_as\_text\_Button = new javax.swing.JButton();

Save\_as\_bit\_Button = new javax.swing.JButton();

Load\_as\_bit\_Button = new javax.swing.JButton();

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

setTitle("Integral");

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));

Add\_row\_Button.setText("Добавить");

Add\_row\_Button.addMouseListener(new java.awt.event.MouseAdapter() {

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

Add\_row\_ButtonMouseClicked(evt);

}

});

getContentPane().add(Add\_row\_Button, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 220, 150, 30));

Delete\_row\_Button.setText("Удалить");

Delete\_row\_Button.addMouseListener(new java.awt.event.MouseAdapter() {

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

Delete\_row\_ButtonMouseClicked(evt);

}

});

getContentPane().add(Delete\_row\_Button, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 270, 150, 30));

Calculate\_Button.setText("Вычислить");

Calculate\_Button.addMouseListener(new java.awt.event.MouseAdapter() {

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

Calculate\_ButtonMouseClicked(evt);

}

});

Calculate\_Button.addActionListener(new java.awt.event.ActionListener() {

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

Calculate\_ButtonActionPerformed(evt);

}

});

getContentPane().add(Calculate\_Button, 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));

Add\_collection\_Button.setText("Добавить элементы коллекции");

Add\_collection\_Button.setInheritsPopupMenu(true);

Add\_collection\_Button.addActionListener(new java.awt.event.ActionListener() {

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

Add\_collection\_ButtonActionPerformed(evt);

}

});

getContentPane().add(Add\_collection\_Button, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 330, 250, 50));

Remove\_collection\_Button.setText("Убрать элементы коллекции");

Remove\_collection\_Button.addActionListener(new java.awt.event.ActionListener() {

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

Remove\_collection\_ButtonActionPerformed(evt);

}

});

getContentPane().add(Remove\_collection\_Button, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 390, 250, 50));

Save\_as\_text\_Button.setText("Сохранить в файл");

Save\_as\_text\_Button.addActionListener(new java.awt.event.ActionListener() {

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

Save\_as\_text\_ButtonActionPerformed(evt);

}

});

getContentPane().add(Save\_as\_text\_Button, new org.netbeans.lib.awtextra.AbsoluteConstraints(310, 330, 200, -1));

Load\_as\_text\_Button.setText("Загрузить из файла");

Load\_as\_text\_Button.addActionListener(new java.awt.event.ActionListener() {

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

Load\_as\_text\_ButtonActionPerformed(evt);

}

});

getContentPane().add(Load\_as\_text\_Button, new org.netbeans.lib.awtextra.AbsoluteConstraints(310, 360, 200, -1));

Save\_as\_bit\_Button.setText("Сохранить в файл (binary)");

Save\_as\_bit\_Button.addActionListener(new java.awt.event.ActionListener() {

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

Save\_as\_bit\_ButtonActionPerformed(evt);

}

});

getContentPane().add(Save\_as\_bit\_Button, new org.netbeans.lib.awtextra.AbsoluteConstraints(520, 330, 200, -1));

Load\_as\_bit\_Button.setText("Загрузить из файла(binary)");

Load\_as\_bit\_Button.addActionListener(new java.awt.event.ActionListener() {

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

Load\_as\_bit\_ButtonActionPerformed(evt);

}

});

getContentPane().add(Load\_as\_bit\_Button, new org.netbeans.lib.awtextra.AbsoluteConstraints(520, 360, 200, -1));

pack();

}// </editor-fold>

private void Calculate\_ButtonActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void Calculate\_ButtonMouseClicked(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;

}

ArrayList<ThreadCalculateIntegral> arrThread = new ArrayList<ThreadCalculateIntegral>();

CollectionIntegral synchIntegral=new CollectionIntegral();

double piece=(rigth-left)/4;

for(int i=1;i<5;i++){

ThreadCalculateIntegral buf = new ThreadCalculateIntegral((left+(i-1)\*piece),(left+i\*piece),dx,synchIntegral);

buf.start();

arrThread.add(buf);

}

for(int i=0;i<4;i++){

try {

arrThread.get(i).join();

} catch (InterruptedException ex) {

Logger.getLogger(Lab5.class.getName()).log(Level.SEVERE, null, ex);

}

}

String result=synchIntegral.result;

if(row!=null){

row.insertElementAt(result, 3);

}

else {

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

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

model.addRow(arr);

}

// TODO add your handling code here:

}

private void Delete\_row\_ButtonMouseClicked(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 Add\_row\_ButtonMouseClicked(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 Add\_collection\_ButtonActionPerformed(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 Remove\_collection\_ButtonActionPerformed(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;

}

}

private void Save\_as\_text\_ButtonActionPerformed(java.awt.event.ActionEvent evt) {

JFileChooser fileChooser = new JFileChooser();

fileChooser.setDialogTitle("Сохранить в файл");

// Определение режима - только каталог

fileChooser.setFileSelectionMode(JFileChooser.FILES\_AND\_DIRECTORIES);

int resultChoose = fileChooser.showOpenDialog(Lab5.this);

File saveFile=null;

// Если директория выбрана, покажем ее в сообщении

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

//JOptionPane.showMessageDialog(Lab5.this,fileChooser.getSelectedFile());

saveFile=fileChooser.getSelectedFile();

}else{

JOptionPane.showMessageDialog(Lab5.this,"Ошибка выбора файла");

}

//Чтение таблицы

int CountRows = jTable1.getRowCount();

String data="";

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

String left=jTable1.getValueAt(i, 0).toString();

String right=jTable1.getValueAt(i, 1).toString();

String step=jTable1.getValueAt(i, 2).toString();

String result=jTable1.getValueAt(i, 3).toString();

data+=left+" "+right+" "+step+" "+result+"\n";

}

try{

FileWriter writer=new FileWriter(saveFile);

writer.write(data);

writer.close();

}catch(IOException e){

System.out.println(e.getMessage());

}

}

private void Load\_as\_text\_ButtonActionPerformed(java.awt.event.ActionEvent evt) {

//Загрузить из файла

JFileChooser fileChooser = new JFileChooser();

fileChooser.setDialogTitle("Загрузить из файла");

// Определение режима - только каталог

fileChooser.setFileSelectionMode(JFileChooser.FILES\_AND\_DIRECTORIES);

int resultChoose = fileChooser.showOpenDialog(Lab5.this);

File saveFile=null;

// Если директория выбрана, покажем ее в сообщении

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

//JOptionPane.showMessageDialog(Lab5.this,fileChooser.getSelectedFile());

saveFile=fileChooser.getSelectedFile();

}else{

JOptionPane.showMessageDialog(Lab5.this,"Ошибка выбора файла");

}

//Чтение таблицы

ArrayList<CollectionIntegral> data=new ArrayList<CollectionIntegral>();

try{

Scanner scanner = new Scanner(saveFile);

while(scanner.hasNextLine()){

String[] buf=scanner.nextLine().split(" ");

data.add(new CollectionIntegral(buf[0],buf[1],buf[2],buf[3]));

}

scanner.close();

}catch(IOException e){

System.out.println(e.getMessage());

}

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

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

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

}

}

private void Save\_as\_bit\_ButtonActionPerformed(java.awt.event.ActionEvent evt) {

JFileChooser fileChooser = new JFileChooser();

fileChooser.setDialogTitle("Сохранить в файл");

// Определение режима - только каталог

fileChooser.setFileSelectionMode(JFileChooser.FILES\_AND\_DIRECTORIES);

int resultChoose = fileChooser.showOpenDialog(Lab5.this);

File saveFile=null;

// Если директория выбрана, покажем ее в сообщении

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

//JOptionPane.showMessageDialog(Lab5.this,fileChooser.getSelectedFile());

saveFile=fileChooser.getSelectedFile();

}else{

JOptionPane.showMessageDialog(Lab5.this,"Ошибка выбора файла");

}

//Чтение таблицы

int CountRows = jTable1.getRowCount();

ArrayList<CollectionIntegral> data=new ArrayList<CollectionIntegral>();

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

String left=jTable1.getValueAt(i, 0).toString();

String right=jTable1.getValueAt(i, 1).toString();

String step=jTable1.getValueAt(i, 2).toString();

String result=jTable1.getValueAt(i, 3).toString();

data.add(new CollectionIntegral(left,right,step,result));

}

try{

FileOutputStream outputStream = new FileOutputStream(saveFile.getPath());

ObjectOutputStream objectOutputStream = new ObjectOutputStream(outputStream);

// сохраняем игру в файл

objectOutputStream.writeObject(data);

objectOutputStream.close();

}catch(IOException e){

System.out.println(e.getMessage());

}

}

private void Load\_as\_bit\_ButtonActionPerformed(java.awt.event.ActionEvent evt) {

//Загрузить из файла

JFileChooser fileChooser = new JFileChooser();

fileChooser.setDialogTitle("Загрузить из файла");

// Определение режима - только каталог

fileChooser.setFileSelectionMode(JFileChooser.FILES\_AND\_DIRECTORIES);

int resultChoose = fileChooser.showOpenDialog(Lab5.this);

File saveFile=null;

// Если директория выбрана, покажем ее в сообщении

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

//JOptionPane.showMessageDialog(Lab5.this,fileChooser.getSelectedFile());

saveFile=fileChooser.getSelectedFile();

}else{

JOptionPane.showMessageDialog(Lab5.this,"Ошибка выбора файла");

}

//Чтение таблицы

ArrayList<CollectionIntegral> data=new ArrayList<CollectionIntegral>();

try{

FileInputStream fileInputStream = new FileInputStream(saveFile.getPath());

ObjectInputStream objectInputStream = new ObjectInputStream(fileInputStream);

data= (ArrayList<CollectionIntegral>)objectInputStream.readObject();

System.out.println(data);

objectInputStream.close();

}catch(IOException e){

System.out.println(e.getMessage());

} catch (ClassNotFoundException ex) {

Logger.getLogger(Lab5.class.getName()).log(Level.SEVERE, null, ex);

}

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

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

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

}

}

/\*\*

\* @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(Lab5.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Lab5.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Lab5.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Lab5.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

/\* Create and display the form \*/

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

public void run() {

new Lab5().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton Add\_collection\_Button;

private javax.swing.JButton Add\_row\_Button;

private javax.swing.JButton Calculate\_Button;

private javax.swing.JButton Delete\_row\_Button;

private javax.swing.JButton Load\_as\_bit\_Button;

private javax.swing.JButton Load\_as\_text\_Button;

private javax.swing.JButton Remove\_collection\_Button;

private javax.swing.JButton Save\_as\_bit\_Button;

private javax.swing.JButton Save\_as\_text\_Button;

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.lab5;

import java.io.Serializable;

/\*\*

\*

\* @author kekos

\*/

public class CollectionIntegral implements Serializable{

private static final long serialVersionUID = 1L;

String left;

String right;

String step;

String result="0";

public CollectionIntegral(){

}

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

this.left=leftArg;

this.right=rightArg;

this.result=resultArg;

this.step=stepArg;

}

synchronized void addResult(double res){

this.result=String.valueOf(Double.parseDouble(this.result)+res);

}

}

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);

}

ThreadCalculateIntegral.java

package com.mycompany.lab5;

/\*\*

\*

\* @author kekos

\*/

public class ThreadCalculateIntegral extends Thread {

double left;

double right;

double step;

CollectionIntegral SynchIntegral;

Integral<Function> integral;

public ThreadCalculateIntegral(double leftArg, double rightArg, double stepArg, CollectionIntegral integral){

this.left=leftArg;

this.right=rightArg;

this.step=stepArg;

this.SynchIntegral=integral;

}

@Override

public void run() { // Этот метод будет вызван при старте потока

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

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

SynchIntegral.addResult(result);

}

}

Выводы

Научились создавать многопоточные приложения c использованием стандартных средств языка Java.