Министерство образования 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 №1

Выполнили:

студенты группы 20ВВП1

Тумасов В.В

Ермолаев А.Д

Портнов Н.С

Приняли:

к.т.н Юрова О.В

к.т.н Карамышева Н.С

Пенза 2023

**Цель работы:** научиться создавать многопоточные приложения c использованием стандартных средств языка Java.

Задание на лабораторную работу

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. Оформление лабораторной работы должно быть выполнено в соответствии с требованиями, приведенными в Приложении 2.

Этапы выполнения:

1)Для реализации интерфейса Runnable создали новый класс “RunOverride”.Данный класс мы наследовали от “Thread”.

2)Далее был создан метод Input(),в котором мы присваиваем нашим переменным,которые будет использовать в вычислении интеграла в переопределенном методе run(), значения,введенные с клавиатуры.

public void Input(float arg1,float arg2,float arg3)

{

a1 = arg1;

a2 = arg2;

a3 = arg3;

}

3)В метод run() переместил наш процесс вычисления интеграла.

public void run() {

float sch,b;

sch=0;

a4=0;

b=a1;

float t1=(float)((a2-a1)%a3);

// t1=Math.abs(t1);

if(t1 > 0)

{

a4=t1\*((float)0.5\*((1/a1)+(1/(a1+t1))));

}

b=a1+t1;

while((b + a3) <= a2)

{

a4+=a3\*(0.5\*((1/b)+(1/(b+a3))));

b+=a3;

}

System.out.println("Выполнен поток " + getName());

}

4)Создали метод result,в котором возвращается результат вычисления интеграла.

public Float result()

{

return a4;

}

Листинг

Класс NewJframe

public class NewJFrame extends javax.swing.JFrame {

Scanner sc= new Scanner(System.in);

public NewJFrame(){

initComponents();

jTextField6.setEditable(false);

jTextField5.setEditable(false);

jTextField8.setEditable(false);

jTextField7.setEditable(false);

// file = new File("result.txt");

}

public void actionPerformed(ActionEvent e) {

JFileChooser fc=new JFileChooser();

fc.setCurrentDirectory(new File("."));

String filename="";

fc.setSelectedFile(new File(filename));

fc.setMultiSelectionEnabled(true);

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

int i=fc.showOpenDialog(fc);

if(i==JFileChooser.APPROVE\_OPTION)

{

filename=fc.getSelectedFile().getPath();

}

}

void input1(){

JFileChooser fc=new JFileChooser();

fc.setCurrentDirectory(new File("."));

String filename="";

fc.setSelectedFile(new File(filename));

fc.setMultiSelectionEnabled(true);

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

int i=fc.showOpenDialog(fc);

if(i==JFileChooser.APPROVE\_OPTION)

{

filename=fc.getSelectedFile().getPath();

file=new File(filename);

}

try{

fw = new FileWriter(file);

fw.write("Нижняя граница Верхняя граница Шаг интегрирования Результат\n");

fw.close();

}catch(IOException e){

System.out.println("Completed!");

}

}

void input(){

try {

fw = new FileWriter(file,true);

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

{

fw.write(record.get(i).inp1(0));

fw.write(" ");

fw.write(record.get(i).inp1(1));

fw.write(" ");

fw.write(record.get(i).inp1(2));

fw.write(" ");

fw.write(record.get(i).inp1(3));

fw.write("\n");

}

fw.close();

} catch(IOException e) {

System.out.println("Completed!");

}

}

void input\_binary(){

JFileChooser fc=new JFileChooser();

fc.setCurrentDirectory(new File("."));

String filename="";

fc.setSelectedFile(new File(filename));

fc.setMultiSelectionEnabled(true);

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

int i=fc.showOpenDialog(fc);

if(i==JFileChooser.APPROVE\_OPTION)

{

filename=fc.getSelectedFile().getPath();

file=new File(filename);

}

ObjectOutputStream out = null;

try {

out = new ObjectOutputStream(new BufferedOutputStream(

new FileOutputStream("record.ser")));

out.writeObject(record);

out.close();

} catch ( IOException ex ) {

ex.printStackTrace();

}

}

void output\_binary(){

JFileChooser fc=new JFileChooser();

fc.setCurrentDirectory(new File("."));

String filename="";

fc.setSelectedFile(new File(filename));

fc.setMultiSelectionEnabled(true);

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

int i=fc.showOpenDialog(fc);

if(i==JFileChooser.APPROVE\_OPTION)

{

filename=fc.getSelectedFile().getPath();

file=new File(filename);

}

ObjectInputStream in = null;

try {

in = new ObjectInputStream(new BufferedInputStream(

new FileInputStream("record.ser")));

record = (ArrayList<RecIntegral>)in.readObject();

in.close();

} catch (IOException | ClassNotFoundException ex) {

ex.printStackTrace();

}

}

void output(){

JFileChooser fc=new JFileChooser();

fc.setCurrentDirectory(new File("."));

String filename="";

fc.setSelectedFile(new File(filename));

fc.setMultiSelectionEnabled(true);

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

int i=fc.showOpenDialog(fc);

if(i==JFileChooser.APPROVE\_OPTION)

{

filename=fc.getSelectedFile().getPath();

file=new File(filename);

}

String[] subStr;

String delimeter = " ";

record.clear();

try {

fr = new FileReader(file);

scan = new Scanner(fr);

scan.nextLine();

while (scan.hasNextLine())

{

subStr = scan.nextLine().split(delimeter);

record.add(new RecIntegral(Float.parseFloat(subStr[0]),Float.parseFloat(subStr[1]),Float.parseFloat(subStr[2]),Float.parseFloat(subStr[3])));

}

fr.close();

} catch(IOException e) {

System.out.println("Completed!");

}

}

FileWriter fw;

FileReader fr;

Scanner scan;

File file;

BufferedWriter bw;

int func;

float num1,num2,num3;

ArrayList<RecIntegral> record = new ArrayList<RecIntegral>();

ExceptionNew test=new ExceptionNew("");

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

DefaultTableModel model = (DefaultTableModel)table1.getModel();

jTextField6.setText("");

jTextField5.setText("");

jTextField8.setText("");

jTextField7.setText("");

model.removeRow(table1.getSelectedRow());

}

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

jTextField6.setEditable(true);

jTextField5.setEditable(true);

jTextField8.setEditable(true);

}

private void jTextField6ActionPerformed(java.awt.event.ActionEvent evt) {

String a4;

a4="";

Float d;

int o=3;

num1=Float.parseFloat(jTextField6.getText());

num2=Float.parseFloat(jTextField5.getText());

num3=Float.parseFloat(jTextField8.getText());

try

{

RunOverride thread=new RunOverride();

thread.Input(num1, num2, num3);

thread.start();

thread.join();

a4 = Float.toString(thread.result());

record.add(new RecIntegral(num1,num2,num3));

}

catch (InterruptedException ex1)

{

}

String s=Float.toString(record.get(record.size()-1).result());

jTextField7.setText(a4);

d=Float.parseFloat(jTextField7.getText());

Float data[] = {num1,num2,num3,d};

DefaultTableModel model = (DefaultTableModel)table1.getModel();

model.addRow(data);

try

{

test.validate(data,o);

}

catch (ExceptionNew ex)

{

JOptionPane.showMessageDialog(null, "Нижняя граница");

// printing the message from InvalidAgeException object

System.out.println("Exception occured: " + ex);

}

try

{

test.validate1(data,o);

}

catch (ExceptionNew ex)

{

JOptionPane.showMessageDialog(null, "Верхняя граница");

// printing the message from InvalidAgeException object

System.out.println("Exception occured: " + ex);

}

try

{

test.validate2(data,o);

}

catch (ExceptionNew ex)

{

JOptionPane.showMessageDialog(null, "Шаг интегрирования");

// printing the message from InvalidAgeException object

System.out.println("Exception occured: " + ex);

}

System.out.println("rest of the code...");

}

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

DefaultTableModel model = (DefaultTableModel)table1.getModel();

model.setRowCount(0);

}

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

DefaultTableModel model = (DefaultTableModel)table1.getModel();

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

{

model.addRow(record.get(i).n1());

}

}

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

input1();

input();

try {

fw.close();

} catch(IOException e) {

System.out.println("Completed!");

}

}

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

output();

DefaultTableModel model = (DefaultTableModel)table1.getModel();

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

{

model.addRow(record.get(i).n1());

}

}

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

input\_binary();

}

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

output\_binary();

DefaultTableModel model = (DefaultTableModel)table1.getModel();

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

{

model.addRow(record.get(i).n1());

} // TODO add your handling code here:

}

public static void main(String args[]){

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

//public void run() {

new NewJFrame().setVisible(true);

// NewJFrame om=new NewJFrame(8);

/\* om.setSize(500,500);

om.setLayout(null);

om.setVisible(true);

om.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

\*/

// }

//});

}

Класс ExceptionNew

package lab1;

public class ExceptionNew extends Exception {

public ExceptionNew (String str)

{

super(str);

}

void validate (Float massiv[],int size) throws ExceptionNew{

int i=0;

if((massiv[0]>=0.000001)&&(massiv[0]<=1000000)&&(massiv[0]<massiv[1])){

System.out.println("Все хорошо");

}

else {

throw new ExceptionNew("Измените нижнюю границу");

}

}

void validate1 (Float massiv[],int size) throws ExceptionNew{

int i=0;

if((massiv[1]>=0.000001)&&(massiv[1]<=1000000)){

System.out.println("Все хорошо");

}

else {

throw new ExceptionNew("Измените верхнюю границу");

}

}

void validate2 (Float massiv[],int size) throws ExceptionNew{

int i=0;

if(massiv[2]<(massiv[1] - massiv[0])){

System.out.println("Все хорошо");

}

else {

throw new ExceptionNew("Измените шаг интегрирования");

}

}

}

Класс RecIntegral

package lab1;

import java.util.Arrays;

import java.io.Serializable;

public class RecIntegral implements Serializable {

Float inf[];

float a4;

RunOverride thread = new RunOverride();

public RecIntegral(float a1,float a2,float a3)

throws InterruptedException {

thread.Input(a1, a2, a3);

thread.start();

thread.join();

a4 = thread.result();

inf=new Float[4];

inf[0]=a1;

inf[1]=a2;

inf[2]=a3;

inf[3]=a4;

}

public RecIntegral(float a1,float a2,float a3,float a4)

{

inf=new Float[4];

inf[0]=a1;

inf[1]=a2;

inf[2]=a3;

inf[3]=a4;

}

public String inp1(int i)

{

String l1= Float.toString(inf[i]);

return l1;

}

public Float[] n1()

{

return inf;

}

public Float result()

{

return a4;

}

}

Класс RunOverride

package lab1;

class RunOverride extends Thread {

float a1;

float a2;

float a3;

float a4;

public void Input(float arg1,float arg2,float arg3)

{

a1 = arg1;

a2 = arg2;

a3 = arg3;

}

public void run() {

float sch,b;

sch=0;

a4=0;

b=a1;

float t1=(float)((a2-a1)%a3);

// t1=Math.abs(t1);

if(t1 > 0)

{

a4=t1\*((float)0.5\*((1/a1)+(1/(a1+t1))));

}

b=a1+t1;

while((b + a3) <= a2)

{

a4+=a3\*(0.5\*((1/b)+(1/(b+a3))));

b+=a3;

}

System.out.println("Выполнен поток " + getName());

}

public Float result()

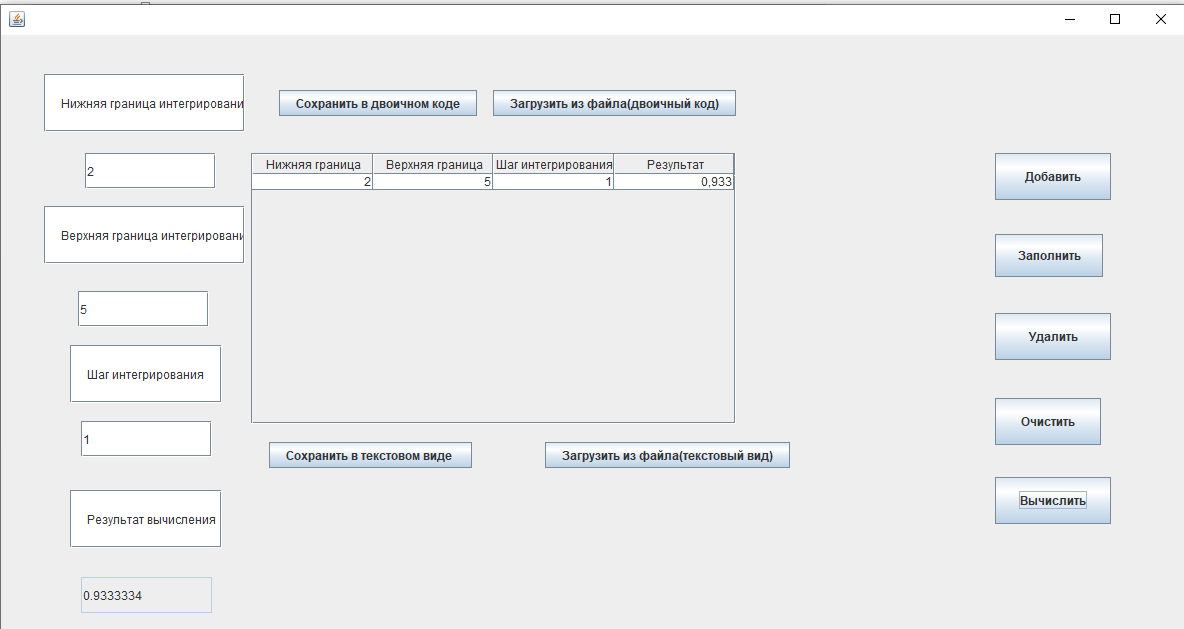
{

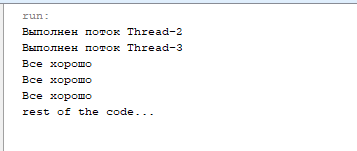
return a4;

}

}

Результат работы программы:





Вывод:в процессе выполнения данной лабораторной работы мы научились создавать многопоточные приложения и,в частности,реализировать интерфейс Runnable.