Министерство науки и образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №5

по курсу «Программирование на Java»

на тему «Обработка исключений»

Вариант 3

Выполнили студенты группы 19ВВ3:

Ланцов Андрей

Трубаненко Александр

Приняли:

Юрова О.В.

**2022**

**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.

**Листинг:**

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

final int threadsCount = Globals.CalculationThreadsCount;

int[] rowIndexes = jTable1.getSelectedRows();

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

IntegrationThread[] threads = new IntegrationThread[threadsCount];

int processedRowsCount = 0;

while (processedRowsCount != rowIndexes.length){

int iterationsCount = Integer.min(threadsCount,rowIndexes.length-processedRowsCount);

try{

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

double from = Double.parseDouble(

jTable1.getValueAt(rowIndexes[i+processedRowsCount],0).toString());

double to = Double.parseDouble(

jTable1.getValueAt(rowIndexes[i+processedRowsCount],1).toString());

double step = Double.parseDouble(

jTable1.getValueAt(rowIndexes[i+processedRowsCount],2).toString());

threads[i] = new IntegrationThread(from,to,step);

threads[i].start();

}

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

threads[i].join();

}

}catch (Exception e){

JOptionPane.showMessageDialog(null,"Потоки врменно вредничают =(");

return;

}

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

dt.setValueAt(threads[i].getResult(), rowIndexes[i+processedRowsCount],3);

}

processedRowsCount+=iterationsCount;

}

}

package Lab1;

import static java.lang.Math.abs;

/\*\*

\*

\* @author tront

\*/

class IntegrationThread extends Thread {

double from;

double to;

double step;

double result;

IntegrationThread(double from, double to, double step){

this.from = from;

this.to = to;

this.step = step;

}

IntegrationThread(){

from = to = step = result = 0;

}

public void run() {

result = TrapIntergration.integrate\_cos(from,to,step);

}

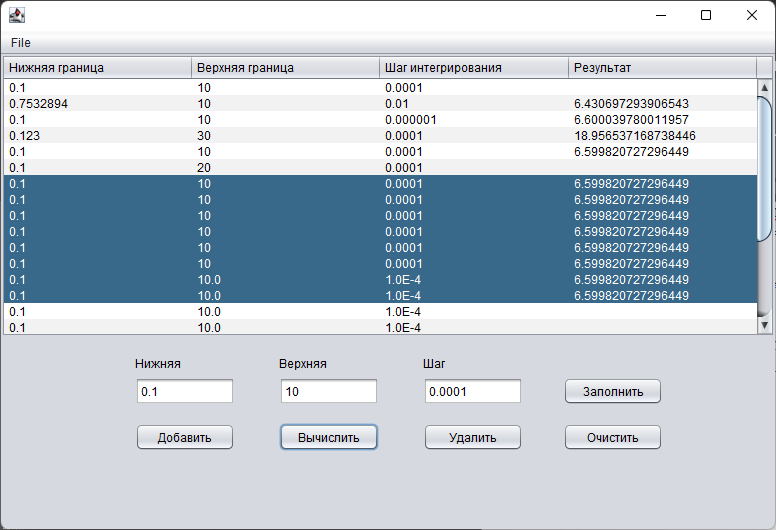
public synchronized double getResult(){

return result;

}

}

**Ход работы:**



**Вывод:**

Научились создавать многопоточные приложения c использованием стандартных средств языка Java.