Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Ярославский государственный технический университет»

Кафедра «Информационные системы и технологии»

Отчет защищен

с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель

\_\_\_\_\_\_\_\_\_Д.В.Дидковская

«\_\_\_»\_\_\_\_\_\_\_\_\_\_ 2022

**ввод-вывод информации, многопоточное программирование, работа с файлами**

Отчёт о лабораторной работе №9

по курсу “Информационные технологии”

ЯГТУ 09.03.02-024 ЛР

|  |  |
| --- | --- |
|  | Отчет выполнил  студент группы ЭИС-26  \_\_\_\_\_\_\_\_\_\_\_ А.А.Хрящев  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2022 |

2022

Цель работы:

Изучить многопоточность в Java.

Задание:

1. Продемонстрировать создание нескольких потоков (thread) с синхронизацией. Например, в одном потоке запустить задачу «вычисление бесконечного ряд», а в другом задачу «табулирование функции».

2. Продемонстрировать ввод с клавиатуры в консольном режиме. Например, для выполнения варианта задания на одномерные массивы в консольном режиме исходный массив ввести с клавиатуры.

3. Продемонстрировать работу с файлами (чтение и запись) с использованием графического диалогового режима. Создать окно с меню. В меню должны быть пункт «Файл». В пункте «Файл» должны быть подпункты «Открыть», «Сохранить как…», «Выход».

Код программы:

1 задние.

public class thread {  
  
 public static void main(String[] args) {  
 var arg = new argument();  
 new tab(arg);  
 new inf(arg);  
 }  
}  
  
class argument  
{  
 double x,y;  
}  
  
class tab extends Thread{  
  
 argument arg;  
 tab(argument arg){  
 super("Новый поток");  
 this.arg = arg;  
 start();  
 }  
  
 public void run() {  
 synchronized (arg){ //заблокировать доступ к методу или части кода, если его уже использует другой поток  
 System.*out*.println("Табулирование функции: ");  
 final double pi4 = 0.7853981633974483;  
 double a = 2.857,  
 b = 3.25,  
 k = 2,  
 h = 0.5;  
 arg.x = -1;  
 arg.y =0;  
 System.*out*.println("\tX\t|\t\tZ");  
 while (arg.x <= k) {  
 if (arg.x > pi4)  
 arg.y = a \* Math.*sin*(arg.x) - b \* arg.x;  
 else if (0 <= arg.x || arg.x >= pi4)  
 arg.y = a \* Math.*exp*(-arg.x) - arg.x;  
 else if (arg.x < 0)  
 arg.y = Math.*pow*(arg.x, 3) + Math.*pow*(arg.x, 2) + Math.*exp*(arg.x) + 1.5;  
  
 System.*out*.printf("%.2f\t|\t%.3f\n", arg.x, arg.y);  
 arg.x += h;  
 }  
 }  
 }  
}  
  
class inf extends Thread{  
 argument arg;  
 inf(argument arg){  
 super("Новый поток1");  
 this.arg = arg;  
 start();  
 }  
  
 public void run() {  
 synchronized (arg){  
 System.*out*.println("\nБесконечные ряды: ");  
 final double e = 0.00001;  
 double xn = 0.1,  
 xk = 1,  
 h = 0.1,  
 T=1;  
 int n = 1;  
 arg.x = xn;  
 arg.y = 0;  
 System.*out*.println("\tX\t|\t\tZ");  
  
 while (arg.x <= xk && Math.*abs*(T) > e)  
 {  
 arg.y += T;  
 T \*= ((arg.x\*Math.*pow*((n+1),2)+arg.x)\*Math.*pow*(2,n))/  
 ((n+1)\*Math.*pow*(2,(n+1))\*((Math.*pow*(n,2)+1)\*Math.*pow*(arg.x,2)));  
 System.*out*.printf("%.2f\t|\t%.3f\n",arg.x,arg.y);  
 arg.x+=h;  
 n++;  
 }  
 }  
 }  
}

2 задание.

import javax.swing.\*;  
import java.util.Scanner;  
  
public class two {  
  
 public static void main(String[] args) {  
 var scanner = new Scanner(System.*in*);  
 System.*out*.println("Введите колличевство элементов в массиве: ");  
 int n = scanner.nextInt();  
 scanner.nextLine();  
 int i;  
 double x = 2;  
 System.*out*.println("Введите массив через запятую: ");  
 var str = scanner.nextLine().split(",");  
 var mas1 = new double[n+1];  
 for (i = 0; i<n;i++)  
 {  
 mas1[i] = Double.*parseDouble*(str[i].trim()); //trim удаляет пробелы  
 }  
  
 System.*out*.println("Введёный массив:");  
 for (i = 0; i < n; i++)  
 System.*out*.print(mas1[i]+", ");  
  
 System.*out*.println("\nВведите искомый элемент: ");  
 x = scanner.nextDouble();  
 scanner.close();  
 for (i = 0, mas1[n] = x; mas1[i] != x; i++) { }  
  
 if (i==n)  
 System.*out*.println("В массиве нет искомого значения");  
 else  
 System.*out*.println ("Найден элемент в массиве с номером: "+i);  
 }  
}

3 задание.

import javax.swing.\*;  
import java.awt.event.\*;  
import java.io.File;  
import java.io.IOException;  
import java.nio.charset.StandardCharsets;  
import java.nio.file.Files;  
import java.nio.file.Path;  
  
public class App{  
  
 static JMenuBar *mb* = new JMenuBar();;  
 static JMenu *x*= new JMenu("Menu");;  
 static JMenuItem *m1*= new JMenuItem("Открыть..."),  
 *m2*= new JMenuItem("Сохранить как..."),  
 *m3*= new JMenuItem("Выход");;  
 static JFrame *frame*;  
 JFileChooser fileChooser;  
  
  
 private JPanel panel;  
 private JTextArea textArea1;  
  
  
 public App() {  
  
  
 *m1*.addActionListener(new ActionListener()  
 {  
  
 @Override  
 public void actionPerformed(ActionEvent arg0) {  
 File file = null;  
 fileChooser = new JFileChooser();  
 int result = fileChooser.showDialog(null, "Открыть файл");  
 if (result == JFileChooser.*APPROVE\_OPTION* )  
 file = fileChooser.getSelectedFile();  
 if(file!=null) {  
 if (!file.exists()) {  
 textArea1.setText("file not found!");  
 return;  
 }  
 }  
 else return;  
 try {  
 textArea1.setText(Files.*readString*(file.toPath()));  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 });  
 *m2*.addActionListener(new ActionListener()  
 {  
 @Override  
 public void actionPerformed(ActionEvent arg0) {  
 File file = null;  
 fileChooser = new JFileChooser();  
 fileChooser.setFileSelectionMode(JFileChooser.*FILES\_ONLY*);  
 int result = fileChooser.showDialog(null,"Сохранить как");  
 if (result == JFileChooser.*APPROVE\_OPTION* )  
 file = fileChooser.getSelectedFile();  
 if(file!=null){  
 if(!file.exists()) {  
 try {  
 file.createNewFile();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }}  
 else return;  
 try {  
 Path path = Files.*writeString*(file.toPath(), textArea1.getText());  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 }  
 });  
 *m3*.addActionListener(new ActionListener()  
 {  
 @Override  
 public void actionPerformed(ActionEvent arg0) {  
 System.*exit*(0);  
 }  
 });  
 }  
  
 public static void main(String[] args) {  
 *frame* = new JFrame("App");  
 *frame*.setContentPane(new App().panel);  
 *frame*.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 *frame*.pack();  
 *frame*.setVisible(true);  
 *x*.add(*m1*);  
 *x*.add(*m2*);  
 *x*.add(*m3*);  
 *mb*.add(*x*);  
 *frame*.setJMenuBar(*mb*);  
 *frame*.setSize(500, 500);  
 *frame*.setVisible(true);  
  
 }  
}

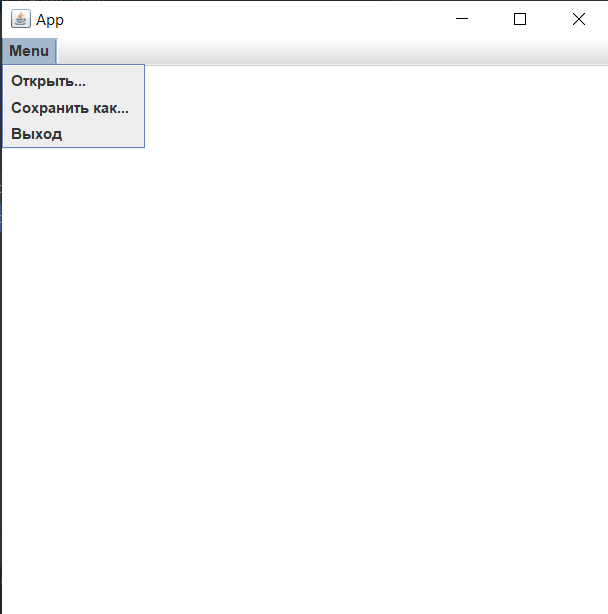


Рисунок 1 – Скриншот интерфейса

Скриншоты выполнения:

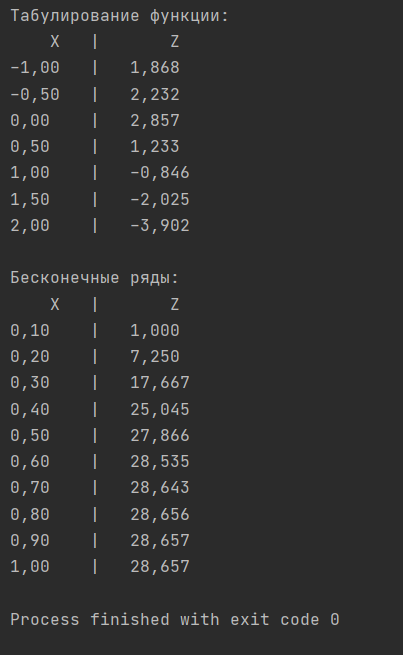


Рисунок 2 – Результат выполнения задания 1

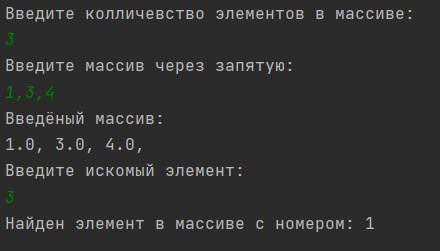


Рисунок 3 – Результат выполнения задания 2

Вывод:

Я ознакомился с потоками в Java, вводом и выводом информации, а также создал программу с меню, которая может открывать, сохранять файлы.