Міністерство освіти і науки України

Національний університет «Львівська політехніка»



**ЗВІТ**

з лабораторної роботи №5

з дисципліни: “Кросплатформні засоби програмування”

на тему: “Файли у Java”

Виконав:

ст. гр. КІ-305

Шевчук М.О.

Прийняв:

Іванов Ю.С.

Львів – 2023

**Мета роботи:** оволодіти навиками використання засобів мови Java для роботи з потоками і файлами.

**Завдання:**

1. Створити клас, що реалізує методи читання/запису у текстовому і двійковому форматах результатів роботи класу, що розроблений у лабораторній роботі №4. Написати програму для тестування коректності роботи розробленого класу.
2. Для розробленої програми згенерувати документацію.
3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагмент згенерованої документації.
4. Дати відповідь на контрольні запитання.

**Варіант 28**

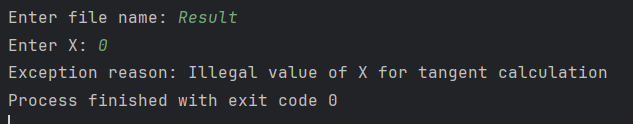


**Лістинг програми:**

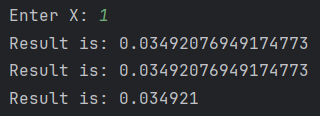
**Lab5ShevchoukKI305.java**

import java.util.Scanner;  
import java.io.\*;  
import static java.lang.System.*out*;  
  
*/\*\**  
 *\* Class <code>EquationsApp</code> Implements driver for Equations class*  
 *\**   
 *\* @author Shevchouk Maxim*  
 *\* @version 1.0*  
 *\*/*  
public class Lab5ShevchoukKI305 {  
 */\*\**  
 *\* @param args*  
 *\* @throws IOException*  
 *\*/*  
public static void main(String[] args) throws IOException {  
 try {  
 CalcWFio obj = new CalcWFio();  
 Scanner in = new Scanner(System.*in*);  
 try {  
 try {  
 Equations eq = new Equations();  
 *out*.print("Enter X: ");  
 *// fout.print(eq.calculate(in.nextInt()));*  
double data = eq.calculate(in.nextDouble());  
 obj.calculate(data);  
 System.*out*.println("Result is: " + obj.getResult());  
 obj.writeResTxt("textRes.txt");  
 obj.writeResBin("BinRes.bin");  
 obj.readResBin("BinRes.bin");  
 System.*out*.println("Result is: " + obj.getResult());  
 obj.readResTxt("textRes.txt");  
 System.*out*.println("Result is: " + obj.getResult());  
 } finally {  
 }  
 } catch (CalcException ex) {  
 *// Блок перехоплює помилки обчислень виразу*  
 *out*.print(ex.getMessage());  
 }  
 in.close();  
 } catch (FileNotFoundException ex) {  
 *// Блок перехоплює помилки роботи з файлом навіть якщо вони*  
 *// виникли у блоці finally*  
 *out*.print("Exception reason: Perhaps wrong file path");  
 }  
  
 }  
}  
  
*/\*\**  
 *\* Class <code>CalcException</code> more precises ArithmeticException*  
 *\**   
 *\* @author Shevchouk Maxim*  
 *\* @version 1.0*  
 *\*/*  
class CalcException extends ArithmeticException {  
 public CalcException() {  
 }  
  
 public CalcException(String cause) {  
 super(cause);  
 }  
}  
  
*/\*\**  
 *\* Class <code>Equations</code> implements method for 7x/tg(2x-4)*  
 *\* expression*  
 *\* calculation*  
 *\**   
 *\* @author Shevchouk Maxim*  
 *\* @version 1.0*  
 *\*/*  
class Equations {  
 */\*\**  
 *\* Method calculates the ctg(8x)/x expression*  
 *\**   
 *\* @param <code>x</code> Angle in degrees*  
 *\* @throws CalcException*  
 *\*/*  
public double calculate(double x) throws CalcException {  
 double y, rad, numPI = Math.*PI*;  
 rad = (2 \* x) \* numPI / 180.0;  
 try {  
 *// 1/ctg(2x) = tg(2x)*  
y = Math.*tan*((2 \* x) \* numPI / 180.0);   
 if (y == Double.*NaN* || y == Double.*NEGATIVE\_INFINITY* ||  
 y == Double.*POSITIVE\_INFINITY* || rad % numPI == 0)  
 throw new ArithmeticException();  
   
 } catch (ArithmeticException ex) {  
 *// створимо виключення вищого рівня з поясненням причини*  
 *// виникнення помилки*  
if (rad==Math.*PI*/2.0 || rad==-Math.*PI*/2.0)  
 throw new CalcException("Exception reason: Illegal value of X for tangent calculation");  
 else  
 throw new CalcException("Unknown reason of the exception during exception calculation");   
 }  
 return y;  
 }  
}  
  
*/\*\**  
 *\* Class <code>CalcWFio</code> work with binary and text files*  
 *\* expression*  
 *\* calculation*  
 *\**   
 *\* @author Shevchouk Maxim*  
 *\* @version 1.0*  
 *\*/*  
class CalcWFio {  
 */\*\**  
 *\* Method write result to text file*  
 *\**   
 *\* @param <code>fName</code> Name of the file*  
 *\* @throws*  
 *\*/*  
public void writeResTxt(String fName) throws FileNotFoundException {  
 PrintWriter f = new PrintWriter(fName);  
 f.printf("%f ", result);  
 f.close();  
 }  
  
 */\*\**  
 *\* Method read result from text file*  
 *\**   
 *\* @param <code>fName</code> Name of the file*  
 *\* @throws*  
 *\*/*  
public void readResTxt(String fName) {  
 try {  
 File f = new File(fName);  
 if (f.exists()) {  
 Scanner s = new Scanner(f);  
 result = s.nextDouble();  
 s.close();  
 } else  
 throw new FileNotFoundException("File " + fName + "not found");  
 } catch (FileNotFoundException ex) {  
 System.*out*.print(ex.getMessage());  
 }  
 }  
  
 */\*\**  
 *\* Method write result to binary file*  
 *\**   
 *\* @param <code>fName</code> Name of the file*  
 *\* @throws*  
 *\*/*  
public void writeResBin(String fName) throws FileNotFoundException, IOException {  
 DataOutputStream f = new DataOutputStream(new FileOutputStream(fName));  
 f.writeDouble(result);  
 f.close();  
 }  
  
 */\*\**  
 *\* Method read result from binary file*  
 *\**   
 *\* @param <code>fName</code> Name of the file*  
 *\* @throws*  
 *\*/*  
public void readResBin(String fName) throws FileNotFoundException, IOException {  
 DataInputStream f = new DataInputStream(new FileInputStream(fName));  
 result = f.readDouble();  
 f.close();  
 }  
  
 */\*\**  
 *\* Method set x for result*  
 *\*/*  
public void calculate(double x) {  
 result = x;  
 }  
  
 */\*\**  
 *\* Method return result*  
 *\*/*  
public double getResult() {  
 return result;  
 }  
  
 private double result;  
}

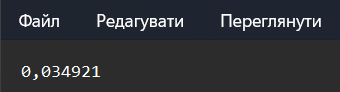
**Результат виконання програми:**



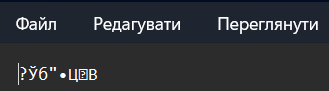
*Рис.1 Виключення при значенні нуль*

**

*Рис.2.Результат виконання програми*

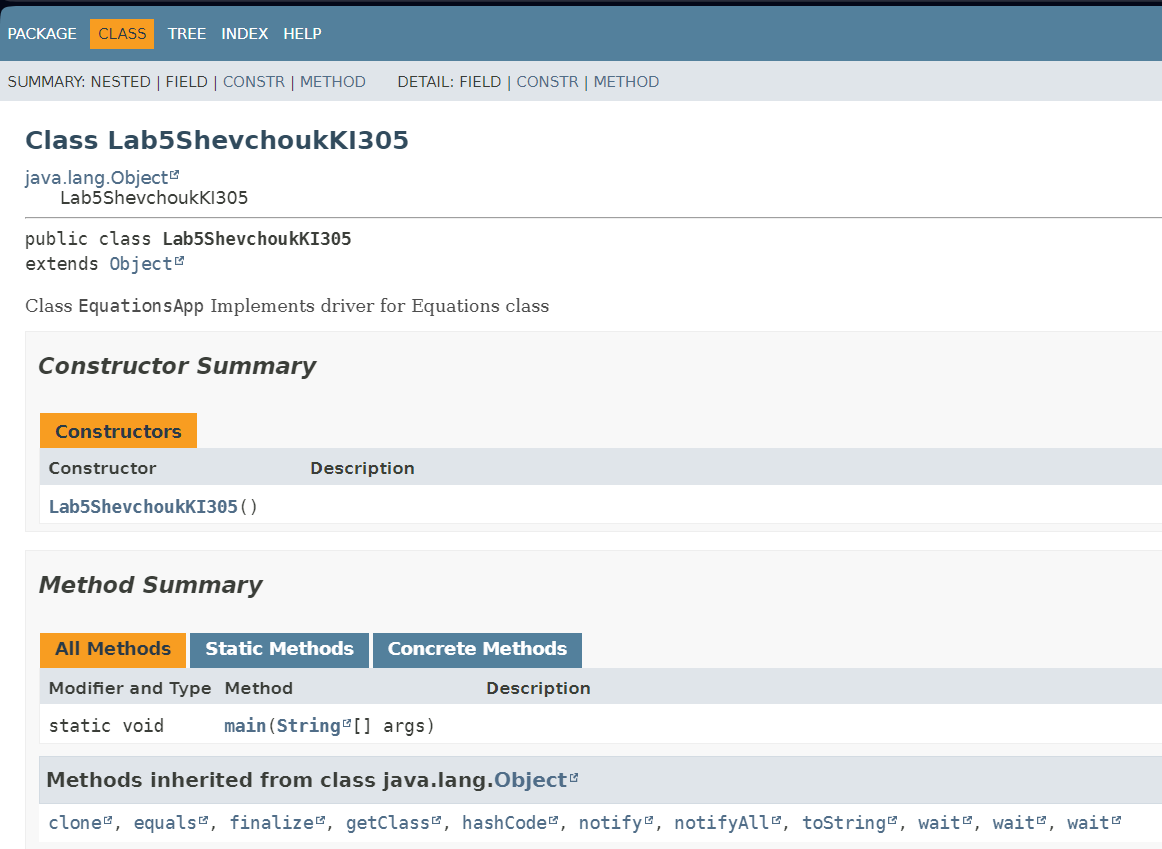


*Рис.3.Успішний запис результату у текстовий файл*

**

*Рис.4.Успішний запис результату у бінарний файл*

**Згенерована документація**



**Відповіді на контрольні запитання:**

1. **Розкрийте принципи роботи з файловою системою засобами мови Java.**

Java надає низку класів для роботи з файловою системою. Основні операції включають в себе читання, запис і видалення файлів. Це можна робити засобами класів File, FileReader, FileWriter, BufferedReader, і BufferedWriter. File використовується для представлення шляху до файлу або каталогу. FileReader і FileWriter використовуються для читання та запису текстових файлів відповідно. BufferedReader і BufferedWriter забезпечують буферизоване читання і запис, що може покращити продуктивність при роботі з файлами.

1. **Охарактеризуйте клас Scanner.**

Scanner  це клас в мові java, який дозволяє зчитувати вхідні дані з різних джерел, таких як консоль, файли тощо. Він надає методи для розбору примітивних типів даних і рядків.

1. **Наведіть приклад використання класу Scanner.**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Введіть ваше ім'я: ");

String name = scanner.nextLine();

System.out.println("Привіт, " + name + "!");

scanner.close();

}

}

1. **За допомогою якого класу можна здійснити запис у текстовий потік?**

Для запису у текстовий потік можна використовувати класи FileWriter або PrintWriter.

1. **Охарактеризуйте клас PrintWriter.**

PrintWriter  це клас, який дозволяє записувати рядки та інші типи даних у текстовий файл. Він забезпечує методи, схожі на System.out.println() для запису даних у файл.

1. **Розкрийте методи читання/запису двійкових даних засобами мови Java.**

Для читання і запису двійкових даних можна використовувати класи FileInputStream і FileOutputStream, які спадають від класів InputStream і OutputStream, відповідно.

1. **Призначення класів DataInputStream і DataOutputStream.**

DataInputStream і DataOutputStream  це класи в Java, які дозволяють зчитувати і записувати примітивні типи даних у бінарний формат.

1. **Який клас мови Java використовується для здійснення довільного доступу до файлів.**

Для довільного доступу до файлів використовується клас RandomAccessFile.

1. **Охарактеризуйте клас RandomAccessFile.**

RandomAccessFile дозволяє читати і записувати дані в файлі як текстові, так і бінарні. Він надає можливість переміщення вказівника у файлі і зчитування/запис даних з конкретної позиції.

1. **Який зв’язок між інтерфейсом DataOutput і класом DataOutputStream?ʼ**

DataOutputStream реалізує інтерфейс DataOutput. Це означає, що він надає всі методи, визначені в інтерфейсі DataOutput, які дозволяють записувати дані примітивних типів у бінарний потік.

**Висновок:**

На цій лабораторній роботі я оволодів навиками використання засобів мови Java для роботи з потоками і файлами.