

Міністерство освіти і науки України  
Національний університет “Львівська політехніка”

Кафедра ЕОМ



### *Звіт*

до лабораторної роботи №5

З дисципліни: «Кросплатформні засоби програмування»

На тему: «ФАЙЛИ У JAVA»

Виконала: ст. гр. КІ-305

Романюк М. Р.

Прийняв: доцент кафедри ЕОМ

Іванов Ю. С.

Львів 2023

Мета: 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. Завантажити код на GitHub згідно методичних вказівок по роботі з GitHub.
4. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації та завантажити його у ВНС.
5. Дати відповідь на контрольні запитання.

Мій варіант №22:  $y = \text{tg}(4x)/x$

### Код програми:

#### Файл EquationsApp.java

```
package KI305RomaniukLab5;

import java.io.*;

/**
 * Клас EquationsApp є програмою-драйвером для обчислення виразу  $y = \text{tg}(4x) / x$  та
 * запису результату в файл.
 * Користувач вводить ім'я файлу та значення X, програма обчислює результат та
 * записує його в файл у текстовому і двійковому форматах.
 */
public class EquationsApp {

    /**
     * Головний метод програми, який виконує основну логіку.
     * @param args Аргументи командного рядка (не використовуються).
     */
    public static void main(String[] args) {
        try {
            BufferedReader reader = new BufferedReader(new
InputStreamReader(System.in));

            // Запит ім'я файлу виводу
            System.out.print("Enter output file name: ");
            String fileName = reader.readLine();

            // Запит значення X
            System.out.print("Enter X: ");
            int x = Integer.parseInt(reader.readLine());

            Equations eq = new Equations();
            double result = eq.calculate(x);

            // Запис результату у текстовий файл
            ResultIO.writeTextResult(fileName + ".txt", result);

            // Запис результату у двійковий файл
            ResultIO.writeBinaryResult(fileName + ".dat", result);

            System.out.println("Results have been written to files.");
        } catch (IOException ex) {
            System.out.println("Error: " + ex.getMessage());
        } catch (CalcException ex) {
            System.out.println(ex.getMessage());
        }
    }
}
```

```

    }
}

/**
 * Клас CalcException є підкласом ArithmeticException і використовується для
 обробки помилок обчислення виразу.
 */
class CalcException extends ArithmeticException {

    /**
     * Конструктор без параметрів.
     */
    public CalcException() {
    }

    /**
     * Конструктор з параметром, який приймає причину помилки.
     * @param cause Причина помилки.
     */
    public CalcException(String cause) {
        super(cause);
    }
}

/**
 * Клас Equations реалізує метод для обчислення виразу  $y = \text{tg}(4x) / x$ .
 */
class Equations {

    /**
     * Метод обчислює вираз  $y = \text{tg}(4x) / x$ .
     * @param x Кут в градусах.
     * @return Результат обчислення виразу.
     * @throws CalcException Виняток, який виникає при некоректних обчисленнях.
     */
    public double calculate(int x) throws CalcException {
        double y, rad;
        rad = 4 * x * Math.PI / 180.0; // Обчислення кута в радіанах
        try {
            y = Math.tan(rad) / x; // Обчислення  $\text{tg}(4x) / x$ 
            if (Double.isNaN(y) || Double.isInfinite(y) || x == 0) {
                throw new ArithmeticException();
            }
        } catch (ArithmeticException ex) {
            if (rad == Math.PI / 2.0 || rad == -Math.PI / 2.0) {
                throw new CalcException("Exception reason: Illegal value of X for tangent calculation");
            } else if (x == 0) {
                throw new CalcException("Exception reason: X = 0");
            } else {
                throw new CalcException("Unknown reason of the exception during exception calculation");
            }
        }
        return y;
    }
}

/**
 * Клас ResultIO реалізує методи для читання/запису результатів роботи класу
 Equations у текстовому і двійковому форматах.
 */
class ResultIO {

    // Метод для запису результату у текстовий файл
    public static void writeTextResult(String fileName, double result) {
        try (PrintWriter writer = new PrintWriter(new FileWriter(fileName))) {
            writer.println(result);
        } catch (IOException e) {

```

```

        e.printStackTrace();
    }
}

// Метод для читання результату з текстового файлу
public static double readTextResult(String fileName) {
    try (BufferedReader reader = new BufferedReader(new FileReader(fileName)))
    {
        String line = reader.readLine();
        if (line != null) {
            return Double.parseDouble(line);
        }
    } catch (IOException | NumberFormatException e) {
        e.printStackTrace();
    }
    return 0.0; // За замовчуванням, якщо читання не вдалося
}

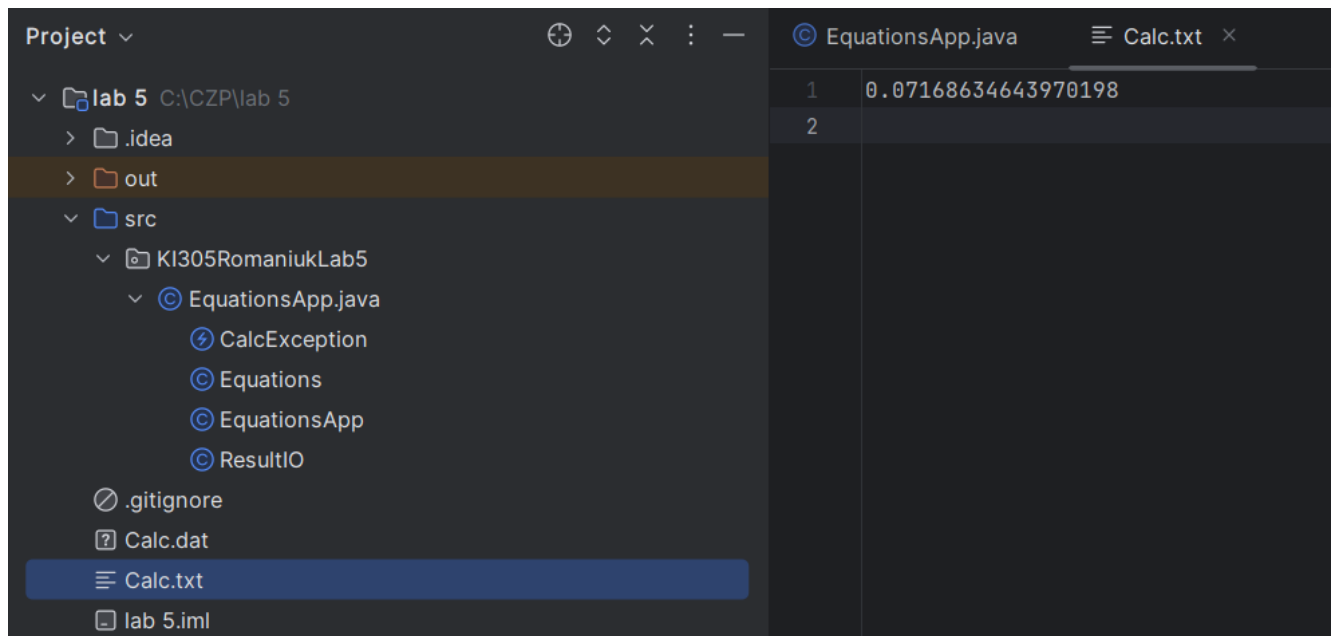
// Метод для запису результату у двійковий файл
public static void writeBinaryResult(String fileName, double result) {
    try (DataOutputStream dos = new DataOutputStream(new
FileOutputStream(fileName))) {
        dos.writeDouble(result);
    } catch (IOException e) {
        e.printStackTrace();
    }
}

// Метод для читання результату з двійкового файлу
public static double readBinaryResult(String fileName) {
    try (DataInputStream dis = new DataInputStream(new
FileInputStream(fileName))) {
        return dis.readDouble();
    } catch (IOException e) {
        e.printStackTrace();
    }
    return 0.0; // За замовчуванням, якщо читання не вдалося
}
}

```

## Результати роботи програми:

```
Enter output file name: Calc
Enter X: 4
Results have been written to files.
```



## Згенерована документація:

PACKAGE

CLASS

TREE

INDEX

HELP

SUMMARY: NESTED | FIELD | CONSTR | METHODDETAIL: FIELD | CONSTR | METHOD

SEARCH

Package KI305RomaniukLab5

**Class EquationsApp**

java.lang.Object<sup>Ⓜ</sup>  
KI305RomaniukLab5.EquationsApp

public class EquationsApp  
extends Object<sup>Ⓜ</sup>

Клас EquationsApp є програмою-драйвером для обчислення виразу  $y = \text{tg}(4x) / x$  та запису результату в файл. Користувач вводить ім'я файлу та значення X, програма обчислює результат та записує його в файл у текстовому і двійковому форматах.

Constructor Summary

Constructors

Constructor	Description
EquationsApp()	

Method Summary

All MethodsStatic MethodsConcrete Methods

Modifier and Type	Method	Description
static void	main(String <sup>Ⓜ</sup> [] args)	Головний метод програми, який виконує основну логіку.

Methods inherited from class java.lang.Object<sup>Ⓜ</sup>

clone<sup>Ⓜ</sup>, equals<sup>Ⓜ</sup>, finalize<sup>Ⓜ</sup>, getClass<sup>Ⓜ</sup>, hashCode<sup>Ⓜ</sup>, notify<sup>Ⓜ</sup>, notifyAll<sup>Ⓜ</sup>, toString<sup>Ⓜ</sup>, wait<sup>Ⓜ</sup>, wait<sup>Ⓜ</sup>, wait<sup>Ⓜ</sup>

### Methods inherited from class java.lang.Object<sup>ⓘ</sup>

clone<sup>ⓘ</sup>, equals<sup>ⓘ</sup>, finalize<sup>ⓘ</sup>, getClass<sup>ⓘ</sup>, hashCode<sup>ⓘ</sup>, notify<sup>ⓘ</sup>, notifyAll<sup>ⓘ</sup>, toString<sup>ⓘ</sup>, wait<sup>ⓘ</sup>, wait<sup>ⓘ</sup>, wait<sup>ⓘ</sup>

### Constructor Details

#### EquationsApp

```
public EquationsApp()
```

### Method Details

#### main

```
public static void main(Stringⓘ[] args)
```

Головний метод програми, який виконує основну логіку.

Parameters:

args - Аргументи командного рядка (не використовуються).

Windows-SSD (C:) > CZP > lab 5 > doc >



Search doc

Name	Date modified	Type	Size
index-files	13.10.2023 22:20	File folder	
KI305RomaniukLab5	13.10.2023 22:20	File folder	
legal	13.10.2023 22:20	File folder	
resources	13.10.2023 22:20	File folder	
script-dir	13.10.2023 22:20	File folder	
allclasses-index.html	13.10.2023 22:20	Chrome HTML Do...	4 KB
allpackages-index.html	13.10.2023 22:20	Chrome HTML Do...	3 KB
copy.svg	13.10.2023 22:20	Microsoft Edge HT...	1 KB
element-list	13.10.2023 22:20	File	1 KB
help-doc.html	13.10.2023 22:20	Chrome HTML Do...	9 KB
index.html	13.10.2023 22:20	Chrome HTML Do...	2 KB
member-search-index.js	13.10.2023 22:20	JavaScript File	1 KB
module-search-index.js	13.10.2023 22:20	JavaScript File	1 KB
overview-tree.html	13.10.2023 22:20	Chrome HTML Do...	4 KB
package-search-index.js	13.10.2023 22:20	JavaScript File	1 KB
script.js	13.10.2023 22:20	JavaScript File	10 KB
search.html	13.10.2023 22:20	Chrome HTML Do...	4 KB
search.js	13.10.2023 22:20	JavaScript File	16 KB
search-page.js	13.10.2023 22:20	JavaScript File	11 KB
stylesheet.css	13.10.2023 22:20	Cascading Style Sh...	32 KB
tag-search-index.js	13.10.2023 22:20	JavaScript File	1 KB
type-search-index.js	13.10.2023 22:20	JavaScript File	1 KB

## Відповіді на контрольні запитання

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

Принципи роботи з файловою системою засобами мови Java базуються на

використанні класів, таких як `FileInputStream` та `FileOutputStream` для роботи з байтовими потоками, та класів `FileReader` та `FileWriter` для роботи з символьними потоками. Засоби Java надають можливість відкривати, читати, записувати та закривати файли, використовуючи методи цих класів.

## 2. Охарактеризуйте клас `Scanner`.

Клас `Scanner` в Java використовується для зчитування введеного користувачем тексту. Він забезпечує зручний інтерфейс для читання даних з консолі або інших джерел введення, перетворюючи їх на різні типи даних, такі як цілі числа, рядки, числа з рухомою комою тощо.

## 3. Наведіть приклад використання класу `Scanner`.

Приклад використання класу `Scanner` для зчитування цілого числа з консолі:

```
import java.util.Scanner;
```

```
public class ScannerExample {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("Введіть ціле число: ");  
        int number = scanner.nextInt();  
        System.out.println("Ви ввели: " + number);  
        scanner.close();  
    }  
}
```

## 4. За допомогою якого класу можна здійснити запис у текстовий потік?

Запис у текстовий потік можна здійснити за допомогою класу `PrintWriter`.

## 5. Охарактеризуйте клас `PrintWriter`.

Клас `PrintWriter` в Java представляє зручний інтерфейс для запису текстових даних у вихідний потік. Він дозволяє виводити рядки тексту у файл або на консоль.

## 6. Розкрийте методи читання/запису двійкових даних засобами мови Java.

Для читання та запису двійкових даних в Java використовуються класи `DataInputStream` і `DataOutputStream`, які надають методи для читання та запису різних примітивних типів даних, таких як цілі числа, числа з рухомою комою, рядки та інші.

## 7. Призначення класів `DataInputStream` і `DataOutputStream`.

Класи `DataInputStream` і `DataOutputStream` використовуються для зчитування та запису примітивних типів даних у двійковому форматі. Вони надають методи для читання та запису цілих чисел, чисел з рухомою комою, рядків та інших типів даних.

## 8. Який клас мови Java використовується для здійснення довільного доступу до файлів.

Для здійснення довільного доступу до файлів використовується клас `RandomAccessFile`.

## 9. Охарактеризуйте клас `RandomAccessFile`.

Клас `RandomAccessFile` в Java дозволяє читати та записувати дані в файлі за допомогою вказаної позиції у файлі. Він надає можливість зчитувати та

записувати дані в будь-якому місці файлу, що робить його корисним для довільного доступу.

10. Який зв'язок між інтерфейсом `DataOutput` і класом `DataOutputStream`?

Інтерфейс `DataOutput` є інтерфейсом, який визначає методи для запису примітивних типів даних у двійковому форматі. Клас `DataOutputStream` реалізує цей інтерфейс та надає можливість запису даних у двійковому форматі у вихідний потік.

**Висновок:** оволоділа навичками використання засобів мови Java для роботи з потоками і файлами.