Національний університет «Львівська Політехніка»

Інститут комп’ютерних технологій, автоматики та метрології

Кафедра електронних обчислювальних машин



Звіт

Про виконання лабораторної роботи №5

### З дисципліни «Кросплатформлені засоби програмування»

**Виконав:**

студент групи КІ-305

Рудий В.В.

**Перевірив:**

Доцент кафедри ЕОМ

Іванов Ю. С.

Львів – 2023

**Тема:** Файли у java.

**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

**Завдання згідно варіанту:**

***Варіант: 23 Завдання: ***

**Виконання:**

**Код програми:**

import java.util.Scanner;

import java.io.\*;

import static java.lang.System.out;

/\*\*

 \* Class <code>EquationsApp</code> Implements driver for Equations class

 \*

 \* @author Rudyi Volodymyr

 \* @version 1.0

 \*/

public class Lab5RudyiKI305 {

    /\*\*

     \* @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 Rudyi Volodymyr

 \* @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 Rudyi Volodymyr

 \* @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 = (8 \* x) \* numPI / 180.0;

        try {

            y = 1 / Math.tan((8 \* x) \* numPI / 180.0) / x;

            // Якщо результат не є числом, то генеруємо виключення

            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 % numPI == 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;

    }

}

/\*\*

 \* Class <code>CalcWFio</code> work with binary and text files

 \* expression

 \* calculation

 \*

 \* @author Rudyi Volodymyr

 \* @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() {

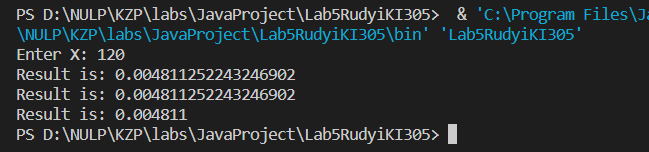
        return result;

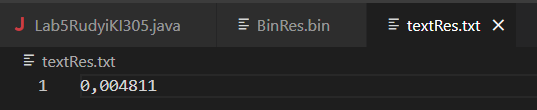
    }

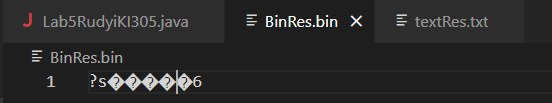
    private double result;

}

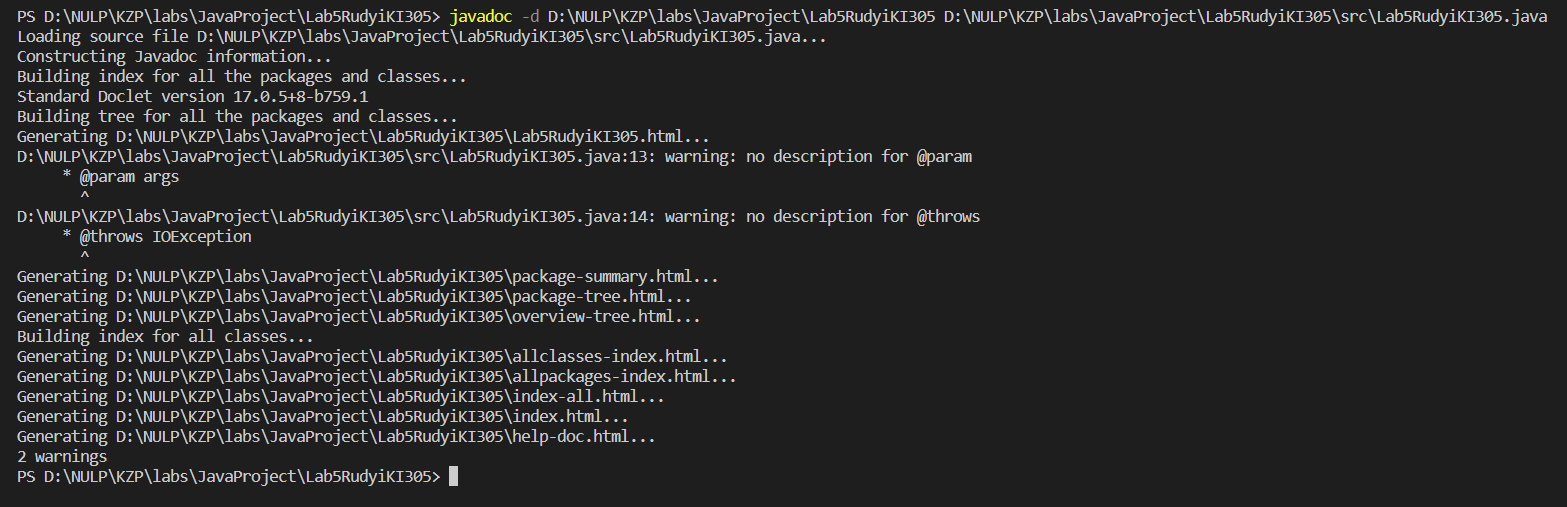
**Результат роботи програми у консолі та файлі:**

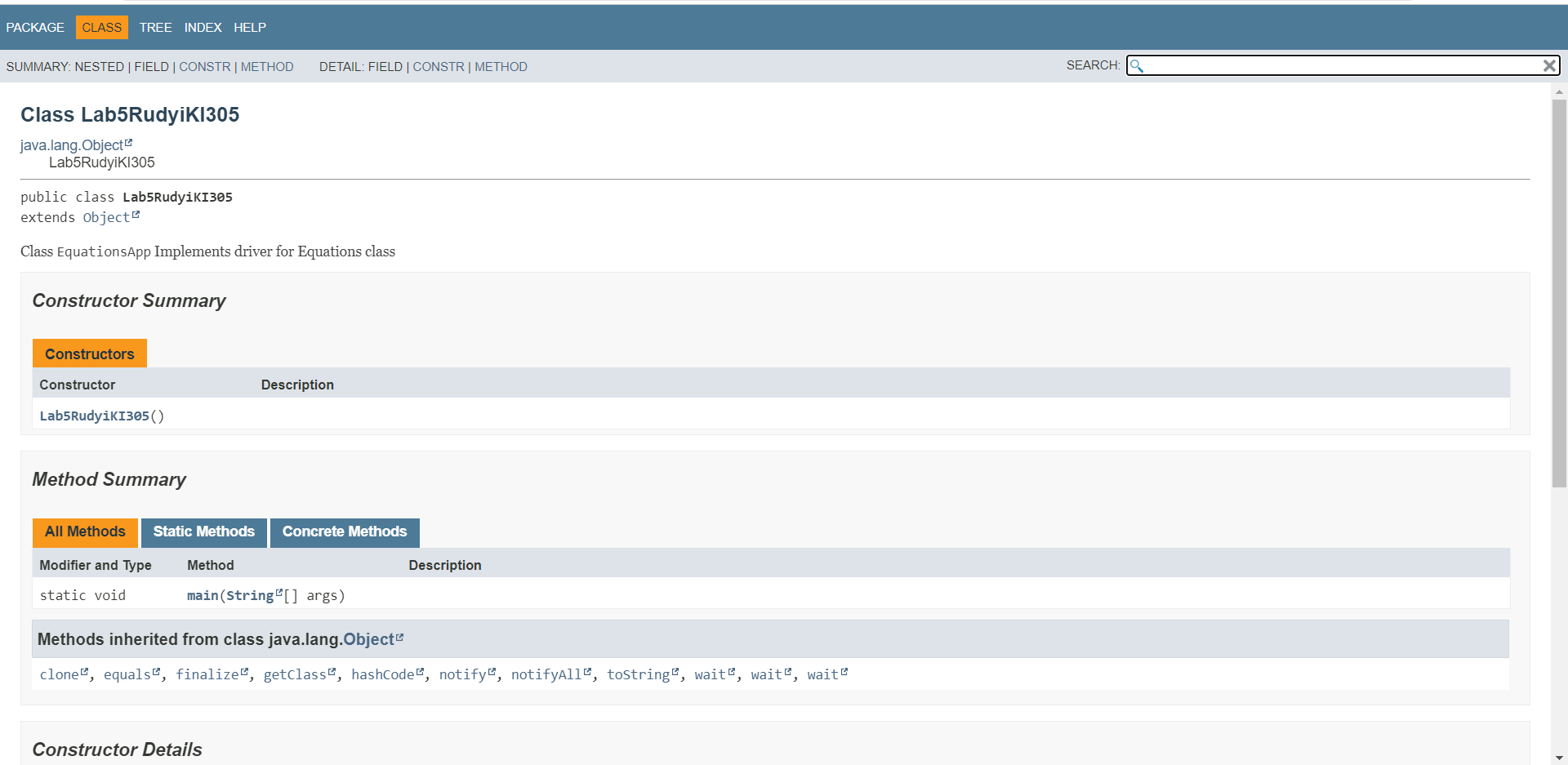






**Фрагмент генерації документації:**

****

****

**Висновок:** на цій лабораторній роботі я оволодів навиками використання засобів мови Java для роботи з потоками і файлами. Також модернізував програму з попередньої лабораторної роботи таким чином, щоб вона записувала і зчитувала результат з бінарного і текстового файлів.