

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»

Варіант 16

Виконав:

ст. групи КІ-306

Мілян М.О.

Прийняв:

доцент кафедри ЕОМ

Олексів М.В.

Львів – 2024

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.

Функція згідно варіанту №16 – « $y=7x/\operatorname{tg}(2x-4)$ »

Хід роботи

Код програми:

Driver.java

```
package KI.Milian.Lab5;

import java.util.Scanner;
import java.io.*;

/**
 * Class <code>EquationsApp</code> Implements driver for Equations class
 * @version 1.0
 */
public class Driver {
    /**
     * @param args
     */
    public static void main(String[] args) throws FileNotFoundException,
    IOException
    {
        FunctionIO func = new FunctionIO();

        try (Scanner s = new Scanner(System.in)) {
            System.out.print("Введіть X: ");
            double data = s.nextDouble();
            func.calculate(data);
        }

        System.out.println("Результат обчислення: " + func.getResult());
        func.writeResTxt("textRes.txt");
        func.writeResBin("BinRes.bin");

        func.readResBin("BinRes.bin");
        System.out.println("Результат в бінарному файлі: " + func.getResult());

        func.readResTxt("textRes.txt");
        System.out.println("Результат в текстовому файлі: " + func.getResult());
    }
}
```

Function.java

```
package KI.Milian.Lab5;

/**
 * Class <code>Functions</code> implements method for  $7x / \operatorname{tg}(2x - 4)$  expression
 * calculation
 * @version 1.0
 */

public class Function {
    /**
     * Method calculates the  $7x / \operatorname{tg}(2x - 4)$  expression*
     * @param x Angle in degrees
     * @throws CalcException
     * @return result
     */
    public static double calculate(double x) throws CalcException {
        double y;
        double rad = x * Math.PI / 180.0;

        try {
            y = 7 * rad / Math.tan(2*rad - 4);

            // Якщо результат не є числом, то генеруємо виключення
            if (Double.isNaN(y) || y == Double.NEGATIVE_INFINITY || y ==
                Double.POSITIVE_INFINITY)
                throw new ArithmeticException();
        }
        // виникнення помилки
        catch (ArithmeticException ex) {
            System.out.println(ex.getMessage());
            throw new CalcException();
        }

        return y;
    }
}
```

FunctionIO.java

```
package KI.Milian.Lab5;

import java.io.*;
import java.lang.*;
import java.util.Scanner;

/**
 * Class <code>FunctionIO</code> implements method for  $7x / \operatorname{tg}(2x - 4)$ 
 * expression calculation and methods for writing and reading files
 * @version 1.0
 */

public class FunctionIO
{
    private double result;

    /**
     * Method writes txt file
     * @param fName File's name
     * @throws FileNotFoundException error
     */
    public void writeResTxt(String fName) throws FileNotFoundException
    {
        PrintWriter f = new PrintWriter(fName);
        f.printf("%f ", result);
    }
}
```

```

        f.close();
    }

    /**
     * Method read txt file
     * @param fName File's name
     */
    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 writes bin file
     * @param fName File's name
     * @throws IOException error
     */
    public void writeResBin(String fName) throws FileNotFoundException,
IOException
    {
        DataOutputStream f = new DataOutputStream(new FileOutputStream(fName));
        f.writeDouble(result);
        f.close();
    }

    /**
     * Method reads bin file
     * @param fName File's name
     * @throws IOException error
     */
    public void readResBin(String fName) throws FileNotFoundException,
IOException
    {
        DataInputStream f = new DataInputStream(new FileInputStream(fName));
        result = f.readDouble();
        f.close();
    }

    public void calculate(double x)
    {
        result = Function.calculate(x);
    }

    public double getResult() {
        return result;
    }
}

```

CalcException.java

```
package KI.Milian.Lab5;

/**
 * Class <code>CalcException</code> more precises ArithmeticException
 * @version 1.0
 */
public class CalcException extends ArithmeticException
{
    public CalcException() {}
    public CalcException(String cause) {
        super(cause);
    }
}
```

```
Введіть X: 30
Результат обчислення: 19.182896761290873
Результат в бінарному файлі: 19.182896761290873
Результат в текстовому файлі: 19.182897

Process finished with exit code 0
```

Рис.1 Вивід результату у консоль

textRes.txt: Блокнот

Файл Редагування Формат Вигляд Довідка

19,182897

Рис.2.1 Вивід результату у текстовий файл

```
00 01 02 03 04 05 06 07
40 33 2E D2 52 78 49 E3
```

Рис.2.2 Вивід результату у бінарний файл

Package KI.Milian.Lab5

package KI.Milian.Lab5

All Classes and Interfaces	Classes	Exception Classes
Class	Description	
CalcException	Class CalcException more precises ArithmeticException	
Driver	Class EquationsApp Implements driver for Equations class	
Function	Class Functions implements method for $7x / \lg(2x - 4)$ expression calculation	
FunctionIO	Class FunctionIO implements method for $7x / \lg(2x - 4)$ expression calculation and methods for writing and reading files	

Рис.3.1 Фрагмент згенерованої документації

Package `Kl.Milian.Lab5`

Class `FunctionIO`

`java.lang.Object`
`Kl.Milian.Lab5.FunctionIO`

```
public class FunctionIO
extends Object
```

Class `FunctionIO` implements method for $7x / \lg(2x - 4)$ expression calculation and methods for writing and reading files

Constructor Summary

Constructors

Constructor	Description
<code>FunctionIO()</code>	

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
<code>void</code>	<code>calculate(double x)</code>	
<code>double</code>	<code>getResult()</code>	
<code>void</code>	<code>readResBin(String fName)</code>	Method reads bin file
<code>void</code>	<code>readResTxt(String fName)</code>	Method read txt file
<code>void</code>	<code>writeResBin(String fName)</code>	Method writes bin file
<code>void</code>	<code>writeResTxt(String fName)</code>	Method writes txt file

Рис.3.2 Фрагмент згенерованої документації

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