

Міністерство освіти і науки України  
Національний університет «Львівська Політехніка»

Кафедра ЕОМ



## **ЗВІТ**

з лабораторної роботи №5

з дисципліни: «Кросплатформенні засоби програмування»

на тему: «Файли у Java»

Варіант - 19

Виконав:

ст. гр. КІ-305

Лихограй А. В.

Прийняв:

доцент кафедри ЕОМ

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. Дати відповіді на контрольні запитання:
  - розкрийте принципи роботи з файловою системою засобами мови Java.
  - охарактеризуйте клас Scanner.
  - наведіть приклад використання класу Scanner.
  - за допомогою якого класу можна здійснити запис у текстовий потік?
  - охарактеризуйте клас PrintWriter.
  - розкрийте методи читання/запису двійкових даних засобами мови Java.
  - призначення класів DataInputStream і DataOutputStream.
  - який клас мови Java використовується для здійснення довільного доступу до файлів.
  - охарактеризуйте клас RandomAccessFile.
  - який зв'язок між інтерфейсом DataOutput і класом DataOutputStream?

**Вихідний код програми:**

До розробленого коду з лабораторної роботи №4 додав файл «*CalcWFio.java*» та змінено «*EguationsApp.java*» на «*FioApp.java*»

```
CalcWFio.java:
package KI305.Lykhohrai.Lab5;
import java.io.*;
import java.lang.*;
import java.util.Scanner;
/**
 * Class <code>CalcWFio</code> implements method for  $tg(x)*ctg(2x)$ 
 expression calculation and methods for writing and reading files
```

```

    * @version 1.0
    */

class CalcWFio
{
    /**
     * 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)
    {
        Equations eq = new Equations();
        result = eq.calculate(x);
    }

    public double getResult()
    {
        return result;
    }
    private double result;
}

```

### *FioApp.java*

```

package KI305.Lykhohrai.Lab5;
import java.util.Scanner;
import java.io.*;

/**
 * Class <code>EquationsApp</code> Implements driver for Equations class
 * @version 1.0
 */
public class FioApp{
    /**
     * @param args
     */
    public static void main(String[] args) throws FileNotFoundException,
    IOException
    {
        CalcWFio obj = new CalcWFio();
        try (Scanner s = new Scanner(System.in)) {
            System.out.print("Введіть X: ");
            double data = s.nextDouble();
            obj.calculate(data);
        }
        System.out.println("Результат обчислення: " + obj.getResult());
        obj.writeResTxt("textRes.txt");
        obj.writeResBin("BinRes.bin");
        obj.readResBin("BinRes.bin");
        System.out.println("Результат в бінарному файлі: " +
        obj.getResult());
        obj.readResTxt("textRes.txt");
        System.out.println("Результат в текстовому файлі: " +
        obj.getResult());
    }
}

```

### **Результат виконання програми:**

```

"C:\Program Files\Java\jdk-21\bin\java.exe" "-jav
Введіть X: 30
Результат обчислення: 0.39999999999999997
Результат в бінарному файлі: 0.39999999999999997
Результат в текстовому файлі: 0.4

```

*Рис. 1. Результат роботи програми.*

## Фрагмент згенерованої документації:

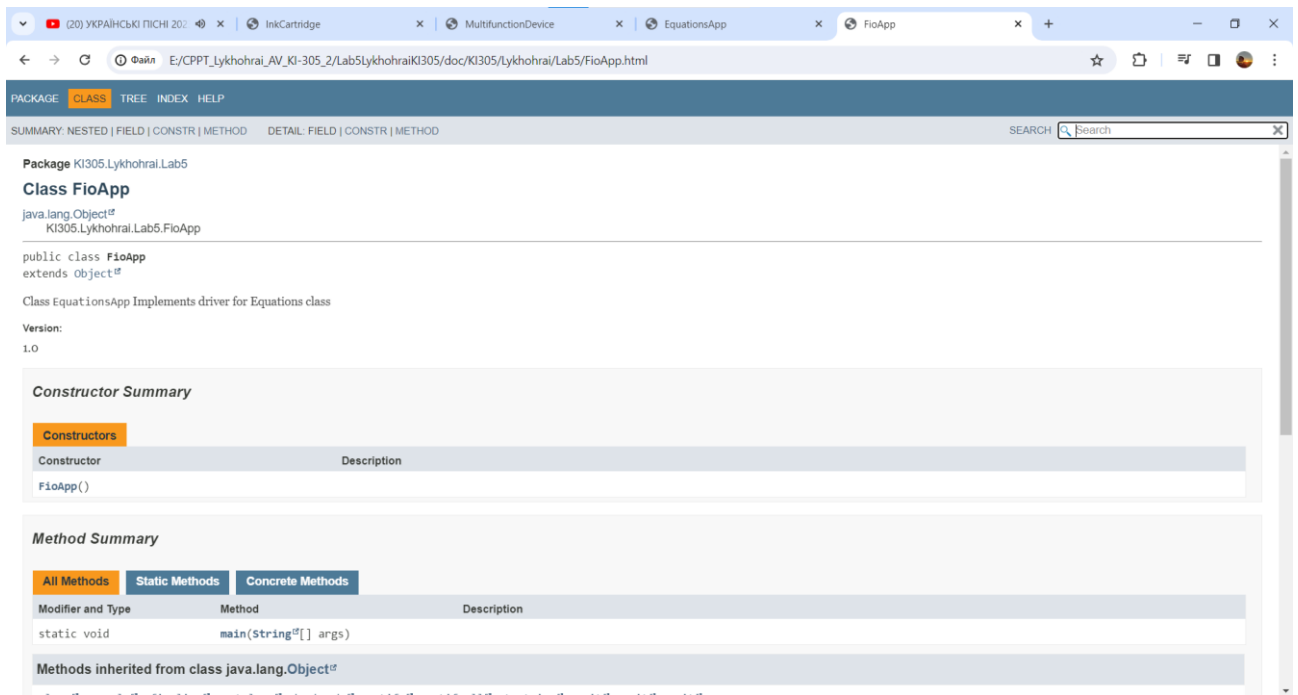


Рис. 2. Фрагмент документації.

## Відповіді на контрольні запитання:

### 1. Принципи роботи з файловою системою в Java:

- Використовуються класи 'File', 'FileReader', 'FileWriter', 'BufferedReader', 'BufferedWriter', 'FileInputStream', та 'FileOutputStream' для читання та запису файлів.

- Класи 'File' та 'Path' використовуються для роботи з іменами файлів та шляхами.

### 2. Характеристика класу Scanner:

- 'Scanner' - це клас у пакеті 'java.util', який надає зручний інтерфейс для читання введення з різних джерел, таких як клавіатура або файли.

### 3. Приклад використання класу Scanner:

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

public class ScannerExample {

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

```
Scanner scanner = new Scanner(System.in);

System.out.print("Введіть текст: ");

String input = scanner.nextLine();

System.out.println("Ви ввели: " + input);

}

}
```

#### 4. Клас для запису у текстовий потік:

- `PrintWriter` дозволяє легко записувати текст у текстовий файл чи інший вихідний потік.

#### 5. Характеристика класу `PrintWriter`:

- `PrintWriter` - це клас, який надає методи для виведення різних типів даних у текстовий потік.

#### 6. Методи читання/запису двійкових даних:

- Для цього використовуються класи `DataInputStream` та `DataOutputStream`.

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

- `DataInputStream` і `DataOutputStream` надають методи для читання і запису примітивних типів даних у бінарному форматі.

#### 8. Клас для довільного доступу до файлів:

- `RandomAccessFile` дозволяє читати та записувати дані в файлі в будь-якому порядку (довільний доступ).

#### 9. Характеристика класу `RandomAccessFile`:

- `RandomAccessFile` - це клас, який дозволяє читати та записувати дані в файлі в будь-якому порядку.

#### 10. Зв'язок між інтерфейсом DataOutput і класом DataOutputStream:

- Клас `DataOutputStream` реалізує інтерфейс `DataOutput`. Цей інтерфейс має методи для запису примітивних типів даних у бінарний потік.

**Висновок:** під час виконання лабораторної роботи я ознайомився з файлами в мові Java. Навчився працювати з ними. Розширив програму з лабораторної роботи №4, яка обчислює вираз:  $y = \frac{\operatorname{ctg}(x)}{\sin(2x) + 4\cos(x)}$ , додавши запис та читання в бінарний та текстовий файли.