

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

**Інститут комп'ютерних технологій, автоматики та метрології**  
**кафедра “Електронних обчислювальних машин”**



**Звіт**  
**з лабораторної роботи №5**  
**дисципліни «Кросплатформні засоби програмування»**  
**Варіант 26**

**Виконала:**  
студент групи КІ-306  
Тимків С. В.  
**Прийняв:**  
Олексів М. В.

Львів – 2024

## ЛАБОРАТОРНА РОБОТА №5

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

### Завдання (Варіант №26)

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.

GitHub Repository:

[https://github.com/SofiaTymkiv/CPPT\\_Tymkiv\\_SV\\_KI-36\\_2.git](https://github.com/SofiaTymkiv/CPPT_Tymkiv_SV_KI-36_2.git)

### Виконання завдання

```
/**
```

```
 * Драйвер для запуску програми обчислення виразу та тестування
 методів читання і запису.
```

```
 */ public class EquationsApp {      public
```

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

```
    ExpressionCalculator calculator = new
    ExpressionCalculator();
```

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

```
{            // Отримання значення x від користувача
```

```
System.out.print("Введіть значення x: ");
```

```
double x = scanner.nextDouble();
```

```
        // Обчислення результату

double result = calculator.calculate(x);

System.out.println("Результат: " + result);

        // Запис результату у текстовий файл

String textFilePath = "result.txt";

calculator.writeResultToFile(result, textFilePath);

System.out.println("Результат записано у текстовий
файл: " + textFilePath);

        // Запис результату у двійковий файл

String binaryFilePath = "result.bin";

calculator.writeResultToBinaryFile(result, binaryFilePath);

System.out.println("Результат записано у двійковий
файл: " + binaryFilePath);

        // Читання результату з текстового файлу

double textResult =
calculator.readResultFromFile(textFilePath);

System.out.println("Результат, зчитаний з текстового
файлу: " + textResult);

        // Читання результату з двійкового файлу

double binaryResult =
calculator.readResultFromBinaryFile(binaryFilePath);
```

```

        System.out.println("Результат, зчитаний з двійкового
файлу: " + binaryResult);

    } catch (IllegalArgumentException e) {

        System.out.println("Помилка: " + e.getMessage());

    } catch (IOException e) {

        System.out.println("Помилка запису або читання файлу:
" + e.getMessage());

    }

} }

```

```

public class ExpressionCalculator {

    public double calculate(double x) throws ArithmeticException
{
    double cosValue = Math.cos(4 * x);          if
(cosValue == 0) {                                throw new
ArithmeticException("Ділення на нуль: cos(4 * " + x + ") =
0");

    }          return 1.0 /
cosValue;

}

```

/\*\*

*\* Записує результат обчислення у текстовий файл.*

*\**

*\* @param result результат обчислення*

```

* @param filePath шлях до файлу
* @throws IOException якщо виникає помилка при записі у файл
    */    public void writeResultToFile(double result,
String filePath) throws IOException {        try
(FileWriter writer = new FileWriter(filePath))
{
            writer.write("Результат обчислення: " +
result);
        }
    }

    /**
* Записує результат обчислення у двійковий файл.
*
* @param result результат обчислення
* @param filePath шлях до двійкового файлу
* @throws IOException якщо виникає помилка при записі у файл
    */    public void writeResultToBinaryFile(double result,
String filePath) throws IOException {        try
(DataOutputStream dos = new DataOutputStream(new
FileOutputStream(filePath)))
{
            dos.writeDouble(result);
        }
    }

    /**
* Читає результат з текстового файлу.

```

```

*

* @param filePath шлях до файлу
* @return результат обчислення, зчитаний з файлу
* @throws IOException якщо виникає помилка при читанні файлу

    */    public double readResultFromFile(String filePath)
throws IOException {        try (BufferedReader reader =
new BufferedReader(new FileReader(filePath)))
{            String line = reader.readLine();
return Double.parseDouble(line.replaceAll("
\d.-]", ""));
        }
    }

/**

* Читає результат з двійкового файлу.

*

* @param filePath шлях до двійкового файлу
* @return результат обчислення, зчитаний з файлу
* @throws IOException якщо виникає помилка при читанні файлу

    */    public double readResultFromBinaryFile(String
filePath) throws IOException {
        try (DataInputStream dis = new DataInputStream(new
FileInputStream(filePath))) {            return
dis.readDouble();
        }
    }
}

```

}

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