

Міністерство освіти і науки України  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра ЕОМ



Звіт

**Лабораторна робота № 5**

З дисципліни «Кросплатформні засоби програмування»

На тему: «ФАЙЛИ У JAVA»

*Виконала: ст. гр. КI-303*

*Кілик І.Р.*

*Перевірив:*

*доцент кафедри ЕОМ*

*Іванов Ю.С.*

**Львів 2025**

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

## **ТЕОРЕТИЧНІ ВІДОМОСТІ**

Бібліотека класів мови Java має більше 60 класів для роботи з потоками. Потоками у мові Java називаються об'єкти з якими можна здійснювати обмін даними. Цими об'єктами найчастіше є файли, проте ними можуть бути стандартні пристрой вводу/виводу, блоки пам'яті і мережеві підключення тощо. Класи по роботі з потоками об'єднані у кілька ієархій, що призначенні для роботи з різними видами даних, або забезпечувати додаткову корисну функціональність, наприклад, підтримку ZIP архівів. Класи, що спадкоємцються від абстрактних класів InputStream і OutputStream призначенні для здійснення байтового обміну інформацією. Підтримка мовою Java одиниць Unicode, де кожна одиниця має кілька байт, зумовлює необхідність у іншій ієархії класів, що спадкоємцються від абстрактних класів Reader і Writer. Ці класи дозволяють виконувати операції читання/запису не байтних даних, а двобайтних одиниць Unicode. Принцип здійснення читання/запису даних нічим не відрізняється від такого принципу у інших мовах програмування. Все починається з створення потоку на запис або читання після чого викликаються методи, що здійснюють обмін інформацією. Після завершення обміну даними потоки необхідно закрити щоб звільнити ресурси.

## **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.

$$\text{Варіант} - y = \operatorname{tg}(x)/\operatorname{ctg}(x)$$

Class	Description
ExpressionCalculator	Клас ExpressionCalculator реалізує методи обчислення виразу $y = \operatorname{tg}(x)/\operatorname{ctg}(x)$ та збереження/читання результатів у текстовому і двійковому форматах.
Main	Тестова програма для ExpressionCalculator

**Висновок:** Завдяки даній лабораторній роботі, я оволоділа навиками використання засобів мови Java для роботи з потоками і файлами.

## Додатки

```
package ki303kilyk.lab5;

import java.io.IOException;
import java.util.Scanner;

/** Тестова програма для ExpressionCalculator */
public class Main {
    public static void main(String[] args) {
        ExpressionCalculator calc = new ExpressionCalculator();
        Scanner scanner = new Scanner(System.in);

        System.out.print("Введіть x у радіанах: ");
        double x = scanner.nextDouble();

        try {
            double y = calc.calculate(x);
            System.out.println("Результат y = " + y);

            // Текстовий файл
            String textFile = "result.txt";
            calc.writeText(textFile);
            System.out.println("Результат записано у текстовий файл: " + textFile);

            // Двійковий файл
            String binFile = "result.bin";
            calc.writeBinary(binFile);
            System.out.println("Результат записано у двійковий файл: " + binFile);

            // Зчитування для перевірки
            ExpressionCalculator calc2 = new ExpressionCalculator();
            calc2.readText(textFile);
            System.out.println("Прочитано з текстового файлу: x=" +
calc2.getLastX() + ", y=" + calc2.getLastY());

            calc2.readBinary(binFile);
            System.out.println("Прочитано з двійкового файлу: x=" +
calc2.getLastX() + ", y=" + calc2.getLastY());

        } catch (ArithmetException e) {
            System.out.println("Помилка обчислення: " + e.getMessage());
        } catch (IOException e) {
            System.out.println("Помилка файлу: " + e.getMessage());
        } finally {
            scanner.close();
        }
    }
}
```

```

package ki303kilyk.lab5;

import java.io.*;

/**
 * Клас {@code ExpressionCalculator} реалізує методи обчислення виразу у =
tg(x)/ctg(x)
 * та збереження/читання результатів у текстовому і двійковому форматах.
 */
public class ExpressionCalculator {

    private double lastX;
    private double lastY;

    /** Обчислює y = tg(x)/ctg(x) */
    public double calculate(double x) throws ArithmeticException {
        double tan = Math.tan(x);
        double cot = 1 / tan;

        if (Double.isNaN(tan) || Double.isNaN(cot) || Double.isInfinite(tan) ||
Double.isInfinite(cot)) {
            throw new ArithmeticException("Некоректне значення для x = " + x);
        }

        lastX = x;
        lastY = tan / cot;
        return lastY;
    }

    /** Запис результату у текстовий файл */
    public void writeText(String filename) throws IOException {
        try (BufferedWriter writer = new BufferedWriter(new FileWriter(filename))) {
            writer.write(lastX + " " + lastY);
        }
    }

    /** Читання результату з текстового файлу */
    public void readText(String filename) throws IOException {
        try (BufferedReader reader = new BufferedReader(new FileReader(filename))) {
            String line = reader.readLine();
            if (line != null) {
                String[] parts = line.split(" ");
                lastX = Double.parseDouble(parts[0]);
                lastY = Double.parseDouble(parts[1]);
            }
        }
    }
}

```

```
/** Запис результату у двійковий файл */
public void writeBinary(String filename) throws IOException {
    try (DataOutputStream dos = new DataOutputStream(new
FileOutputStream(filename))) {
        dos.writeDouble(lastX);
        dos.writeDouble(lastY);
    }
}

/** Читання результату з двійкового файлу */
public void readBinary(String filename) throws IOException {
    try (DataInputStream dis = new DataInputStream(new
FileInputStream(filename))) {
        lastX = dis.readDouble();
        lastY = dis.readDouble();
    }
}

/** Повертає останнє обчислене значення x */
public double getLastX() {
    return lastX;
}

/** Повертає останнє обчислене значення y */
public double getLastY() {
    return lastY;
}
}
```