Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра «Електронних обчислювальних машин»



Звіт

з лабораторної роботи № 5

з дисципліни: «Кросплатформенні засоби програмування»

на тему: «ФАЙЛИ У JAVA»

**Виконав:**

студент групи КІ-306

Чаус Б.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 (варіант № 24)**

**24. y=sin(x-9)/(x-cos(2x))**

1. Створити клас, що реалізує методи читання/запису у текстовому і двійковому форматах результатів роботи класу, що розроблений у лабораторній роботі No4.

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.

**Вихідний код програми**

Файл EquationsApp.java

package KI306.Chaus.Lab5;

import java.io.\*;

import static java.lang.System.out;

import java.util.Scanner; // Added import statement

import java.io.BufferedReader;

import java.io.DataInputStream;

/\*\*

\* This class demonstrates the usage of the Equations class by taking user input for 'x',

\* calculating the result, and saving it to a file named Result.txt.

\*/

public class EquationsApp {

public static void main(String[] args) throws IOException {

String fName = "Result.txt";

String out\_file\_bin = "Out\_binary.dat";

PrintWriter fout = null;

try {

fout = new PrintWriter(new File(fName));

Equations eq = new Equations();

Scanner in = new Scanner(System.in);

out.print("Enter X: ");

int x = in.nextInt();

try {

double result = eq.calculate(x);

out.println("Result: " + result);

fout.print(result);

fout.flush();

DataOutputStream fout\_bin = new DataOutputStream(new FileOutputStream(out\_file\_bin));

fout\_bin.writeChars("Result " + result + "\n");

fout\_bin.close();

} catch (CalcException ex) {

out.print(ex.getMessage());

}

in.close();

} catch (FileNotFoundException ex) {

out.print("Exception reason: File not found");

} finally {

if (fout != null) {

fout.close();

}

}

// Reading from files

BufferedReader fin = null;

DataInputStream fin\_bin = null;

try {

fin = new BufferedReader(new FileReader(fName));

fin\_bin = new DataInputStream(new FileInputStream(out\_file\_bin));

String resultText = fin.readLine();

String binaryResult = fin\_bin.readLine();

System.out.println("Result from Result.txt: " + resultText);

System.out.println("Result from Out\_binary.dat: " + binaryResult);

} catch (FileNotFoundException ex) {

System.out.println("Exception reason: File not found");

} catch (IOException ex) {

System.out.println("Exception while reading the file");

} finally {

try {

if (fin != null) fin.close();

if (fin\_bin != null) fin\_bin.close();

} catch (IOException e) {

System.out.println("Exception while closing the file");

}

}

}

}

Файл Main.java

package KI306.Chaus.Lab5;

class CalcException extends ArithmeticException {

public CalcException() {} // Constructor without a cause message

public CalcException(String cause) {

super(cause); // Constructor with a cause message

}

}

class Equations {

public double calculate(int x) throws CalcException {

double y, rad;

rad = x \* Math.***PI*** / 180.0;

try {

double denominator = x - Math.*cos*(2 \* rad);

if (denominator == 0) {

throw new ArithmeticException();

}

y = Math.*sin*(rad - 9) / denominator;

if (Double.*isNaN*(y) || Double.*isInfinite*(y)) {

throw new ArithmeticException();

}

} catch (ArithmeticException ex) {

if (x == 0) {

throw new CalcException("Exception reason: X = 0");

} else {

throw new CalcException("Exception reason: Division by zero");

}

}

return y;

}

}

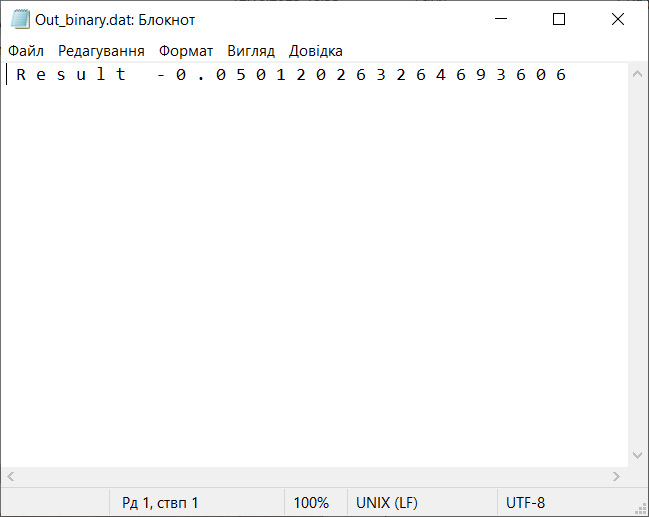
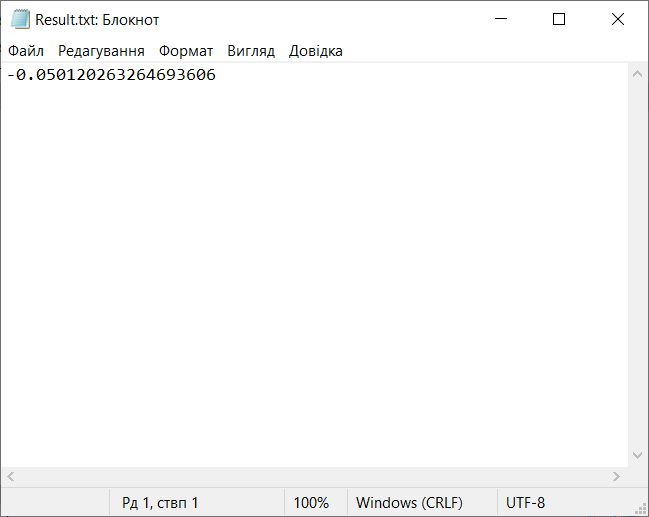
**Результат виконання програми**

Enter X: 13

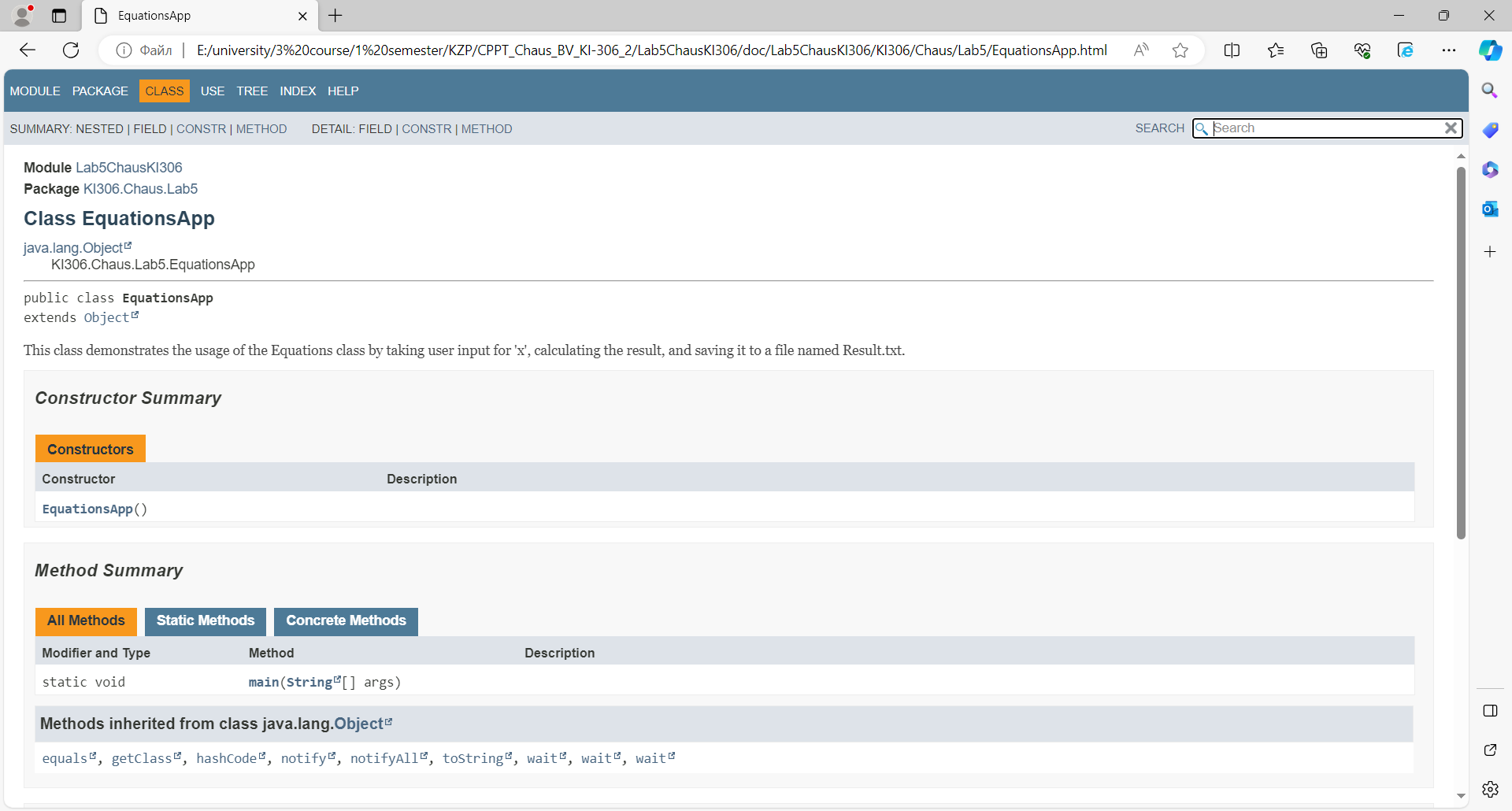
Result: -0.050120263264693606

Result from Result.txt: -0.050120263264693606

**Текстовий файл з результатом виконання програми**



**Фрагмент згенерованої документації**



**Відповіді на контрольні запитання**

1. В Java для роботи з файловою системою використовуються класи з пакету `java.io`, які надають методи для читання та запису даних в файли.

2. Клас `Scanner` в Java використовується для зручного зчитування введених користувачем даних з консолі або з інших вхідних потоків.

3. Приклад використання класу `Scanner`:

Scanner scanner = new Scanner(System.in);

System.out.print("Введіть число: ");

int number = scanner.nextInt();

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

4. Запис у текстовий потік можна здійснити за допомогою класу `PrintWriter`.

5. Клас `PrintWriter` в Java використовується для запису текстових даних в потік.

6. Для читання/запису двійкових даних використовуються класи `DataInputStream` та `DataOutputStream`.

7. Класи `DataInputStream` і `DataOutputStream` призначені для читання та запису примітивних даних та рядків у бінарному форматі.

8. Для здійснення довільного доступу до файлів використовується клас `RandomAccessFile`.

9. Клас `RandomAccessFile` в Java дозволяє читати та записувати дані у файлі за допомогою вказання точної позиції у файлі.

10. Інтерфейс `DataOutput` визначає методи для запису примітивних типів даних, а клас `DataOutputStream` реалізує цей інтерфейс, надаючи можливість запису даних у потік у бінарному форматі.

**Висновок**

Під час лабораторної роботи, я оволодів навиками використання засобів мови Java для роботи з потоками і файлами.