

Міністерство освіти і науки України
Національний університет «Львівська політехніка»

Кафедра ЕОМ



Звіт

до лабораторної роботи № 4

З дисципліни: «Кросплатформенні засоби програмування»

На тему: **«ВИКЛЮЧЕННЯ»**

Варіант 16

Виконав: ст.гр. КІ-301

Онисько М.М.

Прийняв: доц.

Майдан М.В.

Львів 2023

Мета: оволодіти навиками використання механізму виключень при написанні програм мовою Java.

Завдання:

1. Створити клас, що реалізує метод обчислення виразу заданого варіантом. Написати на мові Java та налагодити програму-драйвер для розробленого класу. Результат обчислень записати у файл. При написанні програми застосувати механізм виключень для виправлення помилкових ситуацій, що можуть виникнути в процесі виконання програми. Програма має розміщуватися в пакеті Група.Прізвище.Lab5 та володіти коментарями, які дозволять автоматично згенерувати документацію до розробленого пакету.
2. Автоматично згенерувати документацію до розробленого пакету.
3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.
4. Дати відповідь на контрольні запитання.

Варіант:

16. $y=7x/\text{tg}(2x-4)$

Виконання роботи:

Код програми:

```
package KI301.Onysko.Lab4;

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

import static java.lang.System.out;

/**
 * Class <code>EquationsApp</code> Implements driver for Equations class
 * @author EOM Stuff
 * @version 1.0
 */
public class EquationsApp {
    /**
     * @param args
     */
    public static void main(String[] args)
    {
        try
        {
            out.print("Enter file name: ");
            Scanner in = new Scanner(System.in);
            String fName = in.nextLine();
            PrintWriter fout = new PrintWriter(new File(fName));
            try
            {
                try
                {
                    Equations eq = new Equations();
                    out.print("Enter X: ");
                    fout.print(eq.calculate(in.nextInt()));
                }
            }
        }
    }
}
```

```

        }
        finally
        {
            // Цей блок виконається за будь-яких обставин
            fout.flush();
            fout.close();
        }
    }
    catch (CalcException ex)
    {
        // Блок перехоплює помилки обчислень виразу
        out.print(ex.getMessage());
    }
}
catch (FileNotFoundException ex)
{
    // Блок перехоплює помилки роботи з файлом навіть якщо вони
    // виникли у блоці finally
    out.print("Exception reason: Perhaps wrong file path");
}
}
}

```

```

/**
 * Class <code>CalcException</code> more precises ArithmeticException
 * @author EOM Stuff
 * @version 1.0
 */

```

```

class CalcException extends ArithmeticException
{
    public CalcException(){}
    public CalcException(String cause)
    {
        super(cause);
    }
}

```

```

/**
 * Class <code>Equations</code> implements method for  $((2 / \tan(x)) / x)$  expression *
 * calculation
 * @author EOM Stuff
 * @version 1.0
 */

```

```

class Equations
{
    /**
     * Method calculates the  $((2 / \tan(x)) / x)$  expression
     * @param <code>x</code> Angle in degrees
     * @throws CalcException
     */
    public double calculate(int x) throws CalcException
    {
        double y, rad;
        rad = 2*x-4;

        try
        {
            //y = (2.0 / Math.tan(rad)) / x;
            y = 7*x/Math.tan(rad);
            // Якщо результат не є числом, то генеруємо виключення
            if (y==Double.NaN || y==Double.NEGATIVE_INFINITY ||
y==Double.POSITIVE_INFINITY || x==90 || x== -90)
                throw new ArithmeticException();

```

```

    }
    catch (ArithmeticException ex)
    {
        // створимо виключення вищого рівня з поясненням причини
        // виникнення помилки
        if (rad==Math.PI/2.0 || rad==Math.PI/2.0)
            throw new CalcException("Exception reason: Illegal value of
X for tangent calculation");
        else if (x==0)
            throw new CalcException("Exception reason: X = 0");
        else
            throw new CalcException("Unknown reason of the exception
during exception calculation");
        }
    return y;
}
}

```

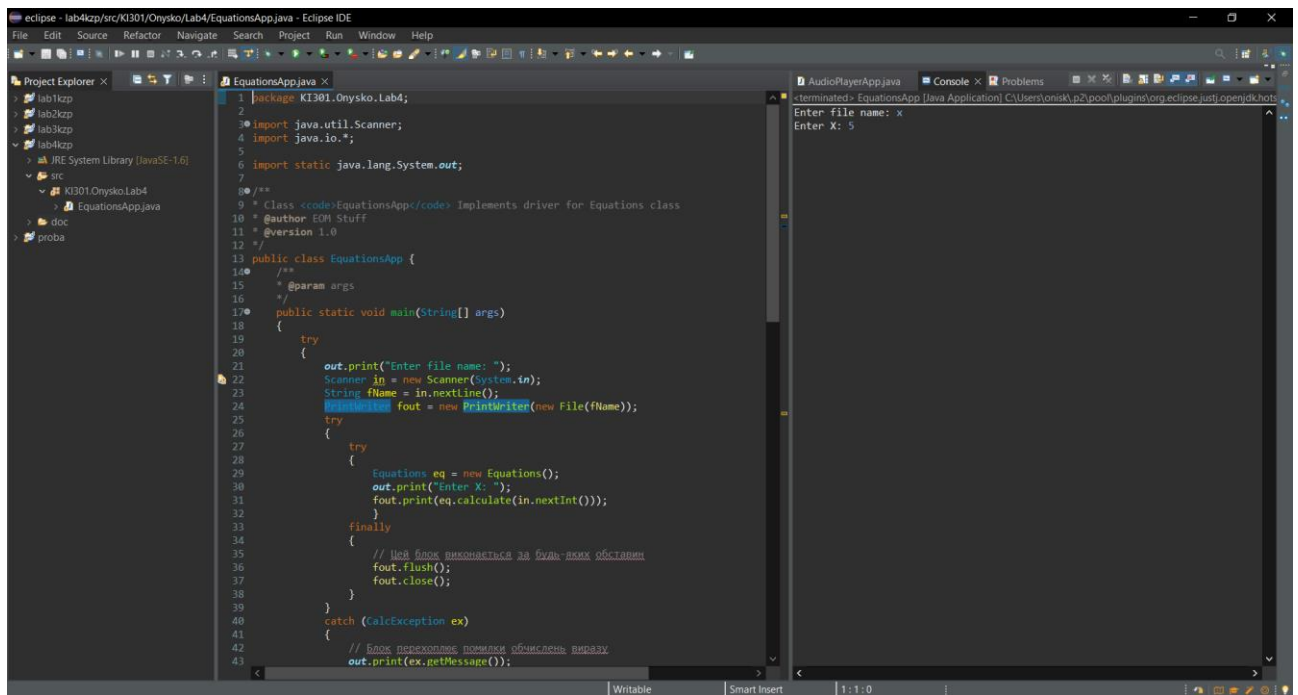


Рис.1 Код та вивід програми при $x=5$

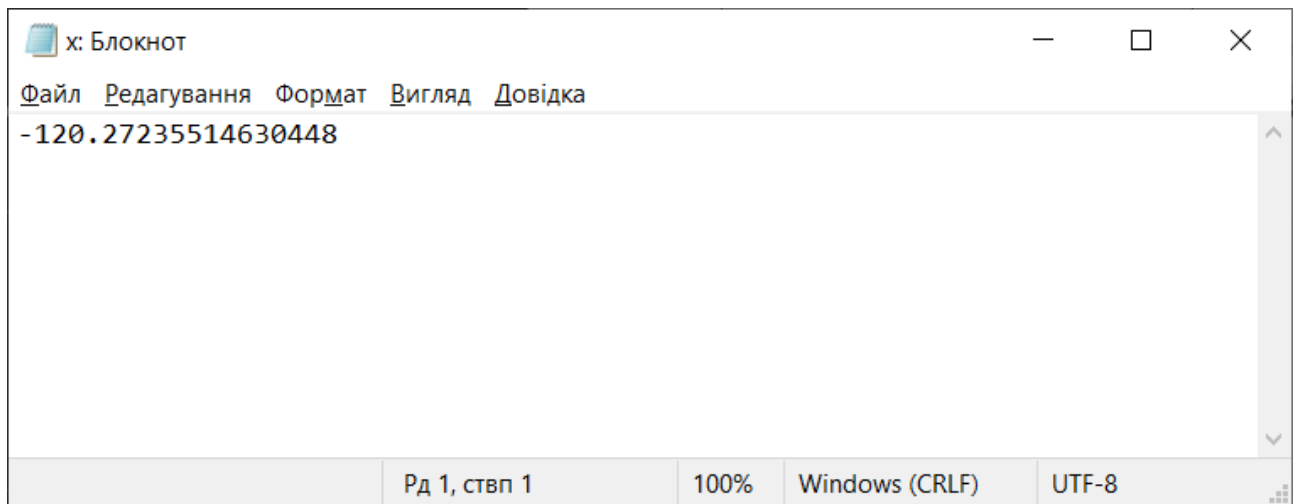


Рис.2 Результат програми при $x=5$

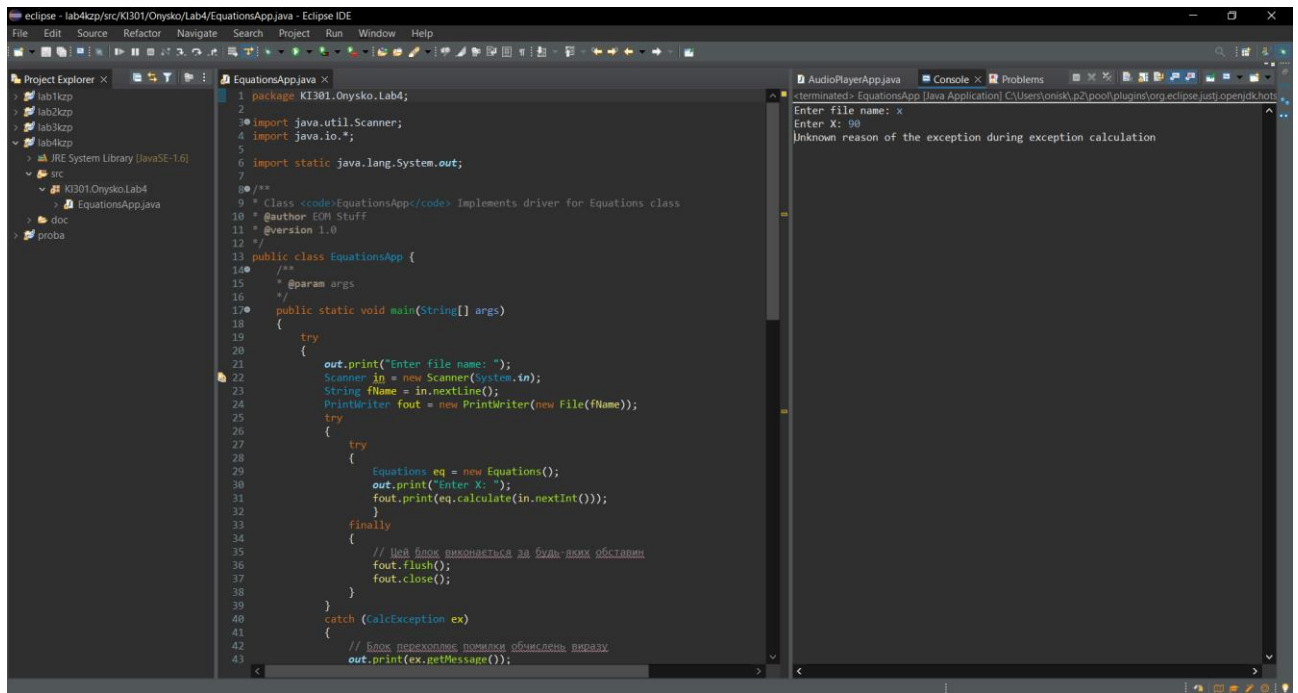


Рис.3 Код та вивід програми при $x=90$

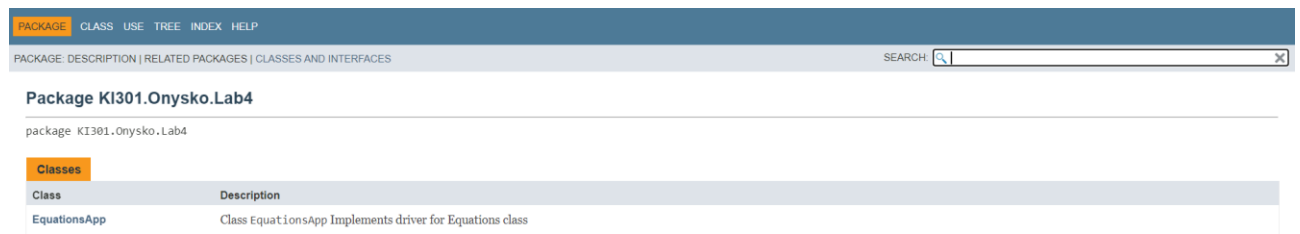


Рис.4 Документація Package KI301.Onysko.Lab4

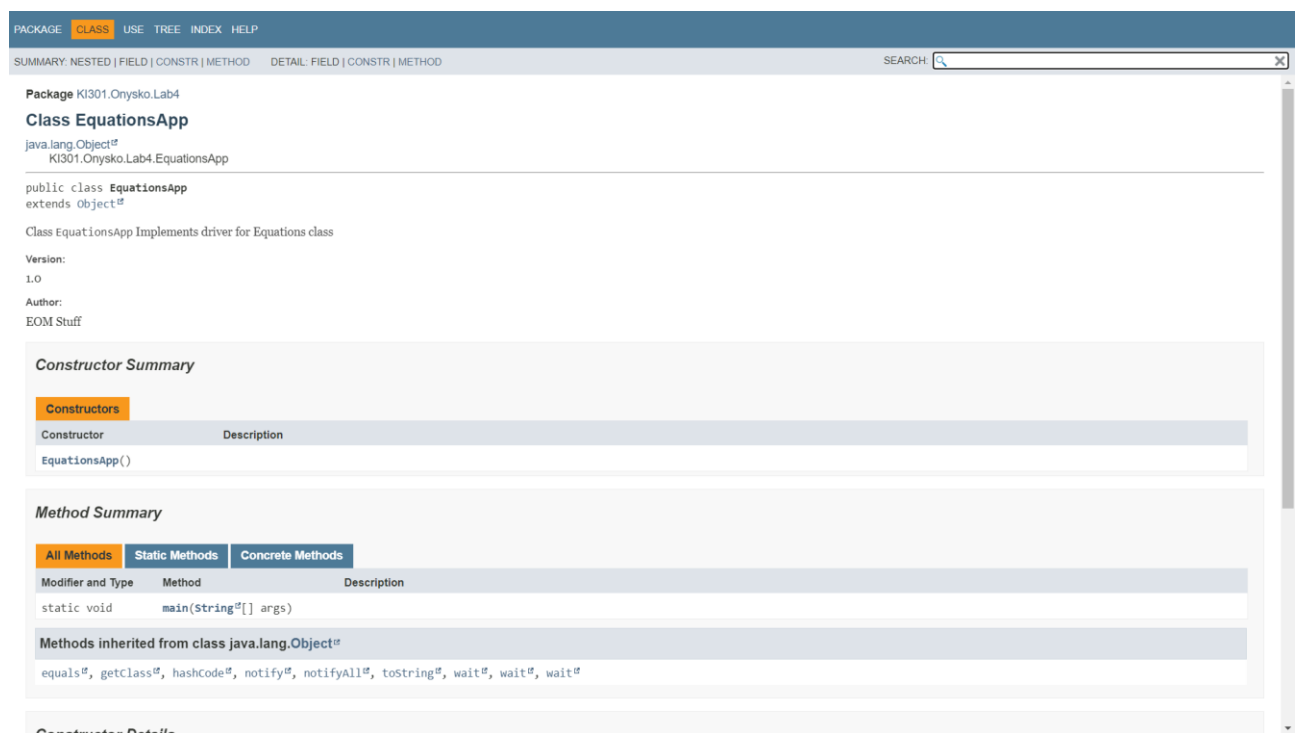


Рис.5 Документація EquationsApp

Висновок:

Освоєно навички використання механізму обробки виключень в Java шляхом створення класу для обчислення математичного виразу з реалізацією

перехоплення можливих помилок, розробки програми-драйвера для тестування роботи класу, збереження результатів у файл, організації коду у пакет з автоматичною генерацією документації за допомогою Javadoc.

Контрольні питання:

1. Розкрийте принципи роботи з файловою системою засобами мови Java.

В Java для роботи з файлами використовуються потокові класи. Існують класи для читання і запису текстових і двійкових даних. Також є класи для довільного доступу до файлів.

2. Охарактеризуйте клас Scanner.

Клас Scanner дозволяє зручно читати текстові дані із вхідного потоку, розбиваючи їх на токени за заданим роздільником.

3. Наведіть приклад використання класу Scanner.

Приклад: `Scanner in = new Scanner(new File("data.txt")); String line = in.nextLine();`

4. За допомогою якого класу можна здійснити запис у текстовий потік?

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

5. Охарактеризуйте клас `PrintWriter`.

`PrintWriter` дозволяє зручно записувати текстові дані у вихідний потік у вигляді рядків і примітивних типів.

6. Розкрийте методи читання/запису двійкових даних засобами мови Java.

Для читання і запису двійкових даних використовуються класи `DataInputStream` і `DataOutputStream` відповідно.

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

`DataInputStream/DataOutputStream` - це зручні обгортки для читання і запису двійкових даних у вигляді примітивних типів.

8. Який клас мови Java використовується для здійснення довільного доступу до файлів.

Довільний доступ до файлів забезпечує клас `RandomAccessFile`.

9. Охарактеризуйте клас `RandomAccessFile`.

RandomAccessFile дозволяє читати і записувати дані у довільну позицію у файлі.

10. Який зв'язок між інтерфейсом DataOutput і класом DataOutputStream?

DataOutputStream реалізує інтерфейс DataOutput, тобто надає реалізацію для запису двійкових даних.