

Міністерство освіти та науки України  
Національний університет «Львівська політехніка»  
Кафедра «Електронних обчислювальних машин»



## **ЗВІТ**

**з лабораторної роботи № 5**

**з дисципліни:** «Кросплатформенні засоби програмування»

**на тему:** «ВИКЛЮЧЕННЯ»

Виконав: ст. гр. КІ-35

Дністряк Я. В.

Прийняв:

доцент кафедри ЕОМ

Іванов Ю.С

Львів-2022

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

### ЗАВДАННЯ (Варіант - 4)

1. Створити клас, що реалізує метод обчислення виразу заданого варіантом. Написати на мові Java та налагодити програму-драйвер для розробленого класу.

$$4. y = \cos(x) / \sin(x)$$

Рис.1. Вираз згідно варіанту

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.

### Текст програми:

#### Лістинг Equation.java

```
package KI35.Dnistrian.lab5;
/**
 * Class <code>Equation</code> implements method for cos(x)/sin(x) expression
 calculation
 * @author EOM Stuff
 * @version 1.0
 */
class Equation
{
    /**
     * Method calculates the cos(x)/sin(x) expression
     * @param <code>x</code> Angle in degrees
     * @throws CalcException
     */
    public double calculate(int x) throws CalcException
    {
        double y, rad;
        rad = x * Math.PI / 180.0;
        try
        {
            y = Math.cos(rad)/Math.sin(rad);
        }
        // Якщо результат не є числом, то генеруємо виключення
    }
}
```

```

        if (y==Double.NaN || y==Double.NEGATIVE_INFINITY ||
y==Double.POSITIVE_INFINITY || rad % Math.PI==0)
            throw new ArithmeticException();
    }
    catch (ArithmeticException ex)
    {
        // створимо виключення вищого рівня з поясненням причини
        // виникнення помилки
        if (rad % Math.PI==0)
            throw new CalcException("Exception reason: cos(x) / 0");
        else
            throw new CalcException("Unknown reason of the exception during exception
calculation");
    }
    return y;
}
}

```

## Лістинг EquationsApp.java

```

package KI35.Dnistrian.Lab5;
import java.util.Scanner;
import java.io.*;
import static java.lang.System.out;
/**
 * Class <code>EquationsApp</code> Implements driver for cos(x)/sin(x) Equations class
 * @author Yaroslav Dnistrian
 * @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
                {
                    Equation eq = new Equation();
                    out.print("Enter X: ");
                    fout.print(eq.calculate(in.nextInt()));
                }
                finally
                {
                    // Цей блок виконається за будь-яких обставин
                    fout.flush();
                    fout.close();
                }
            }
            catch (ExceptionDiv0 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");
}
}
}

```

## Лістинг ExceptionDiv0.java

```

package KI35.Dnistrian.Lab5;

/**
 * Class <code>ExceptionDiv0</code> more precises ArithmeticException
 * @author Yaroslav Dnistrian
 * @version 1.0
 */
class ExceptionDiv0 extends ArithmeticException
{
    public ExceptionDiv0(){}
    public ExceptionDiv0(String cause)
    {
        super(cause);
    }
}

```

## Результат виконання програми

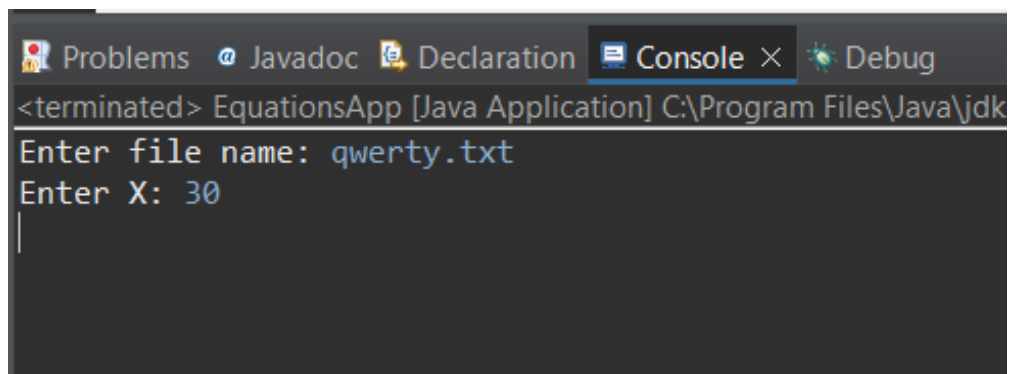


Рис.2.Результати роботи програми

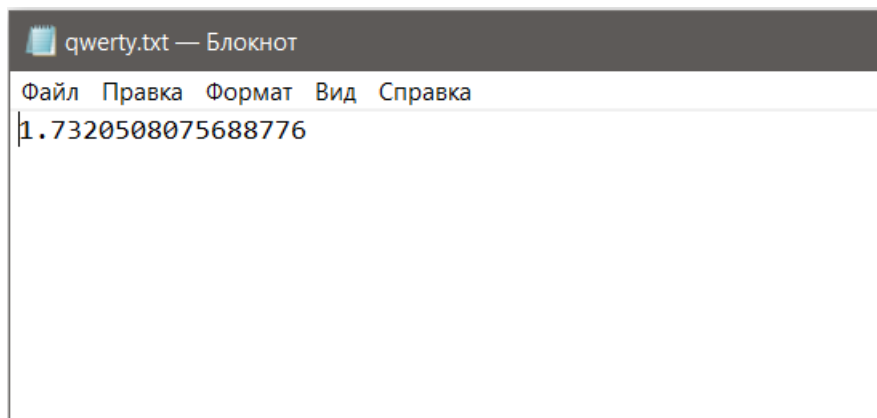


Рис. 3.Результат виводу у файл

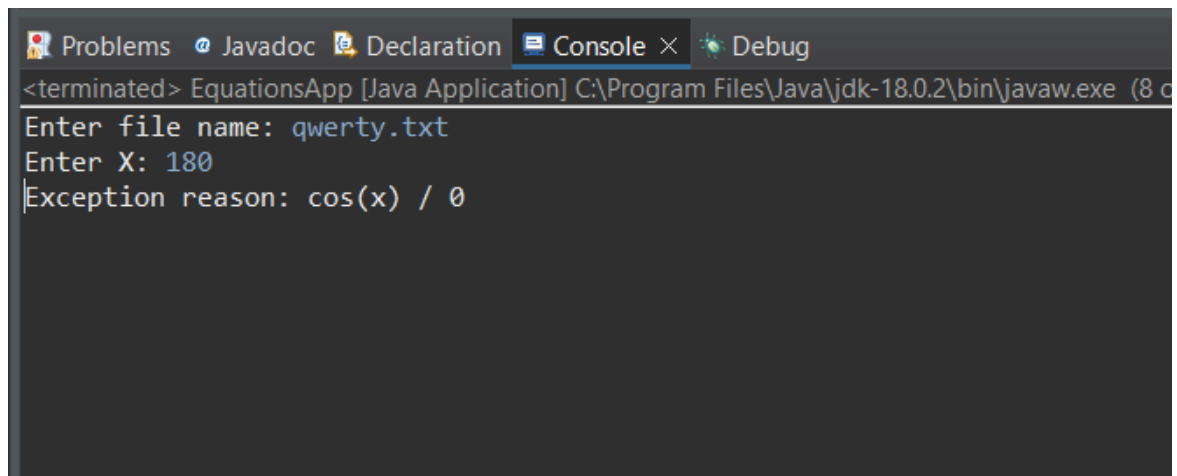
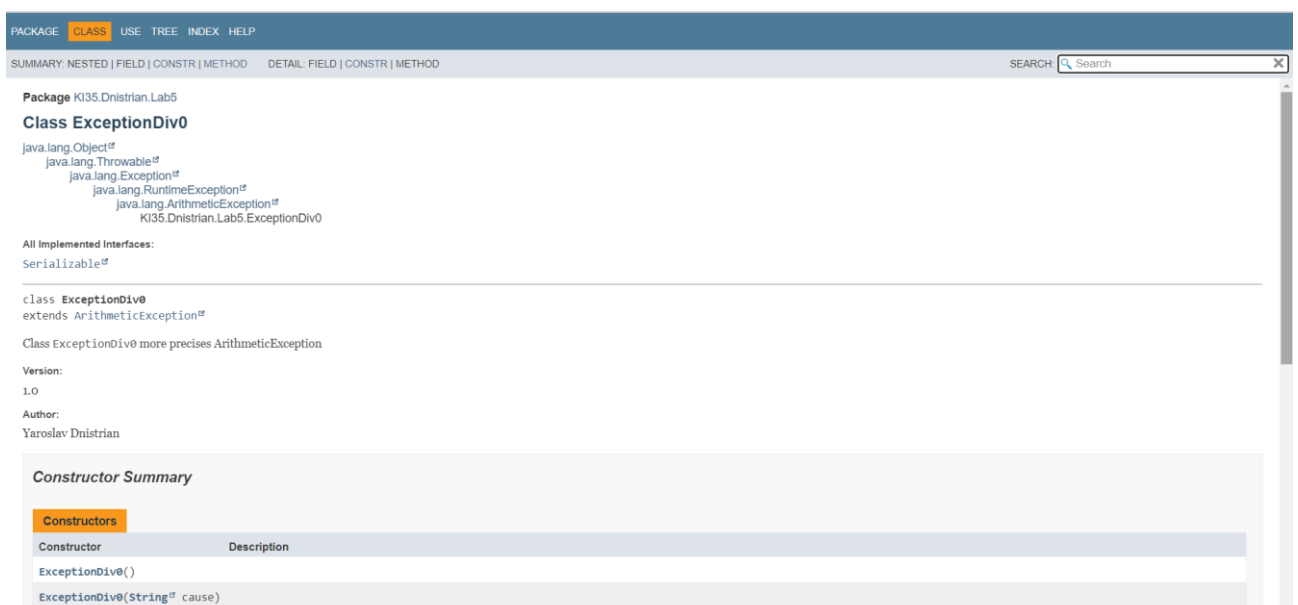


Рис.4. Виключена ситуація

## Фрагмент згенерованої документації



## **Відповіді на контрольні запитання:**

### **1. Дайте визначення терміну «виключення».**

Виключення – це механізм мови Java, що забезпечує негайну передачу керування блоку коду опрацювання критичних помилок при їх виникненні уникаючи процесу розкручування стеку

### **2. Розкрийте призначення та особливості роботи блоку try.**

Якщо код у блоці try згенерував виключення, то подальше виконання коду в цьому блоці припиняється і відбувається пошук блоку catch тип у заголовку якого співпадає з типом виключення.

Якщо код у блоці try не генерує ніяких виключень, то програма спочатку повністю виконає блок try, а потім блок finally.

**Висновок:** виконуючи лабораторну роботу №5, оволодів навиками використання механізму виключень при написанні програм мовою Java.