

Міністерство освіти і науки України  
Національний університет «Львівська політехніка»

Кафедра ЕОМ



## **Лабораторна робота №4**

з дисципліни: «Кросплатформенні засоби програмування»

на тему: «Виключення»

Варіант № 25

Виконав:

ст. гр. КІ-305

Федусь Н.В.

Прийняв:

Іванов Ю. С.

Львів – 2023

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

**Завдання:**

25.  $y=1/\sin(x)$

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.

**Код EquationsApp.java:**

```
package KI305.Fedus.Lab4;

import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;

/**
 * The EquationsApp class contains the main method to demonstrate and write
 * the results of mathematical equations to a file.
 */
public class EquationsApp {

    /**
     * The main method where the program execution begins.
     *
     * @param args The command-line arguments (not used in this program).
     */
    public static void main(String[] args) {
        // Create an instance of CalculateTheEquationInterface
        CalculateTheEquationInterface calc = new CalculateTheEquation();
```

```
// File path where you want to save the results
String filePath = "equation_results.txt";

try (PrintWriter writer = new PrintWriter(new FileWriter(filePath))) {
    // Perform calculations and write results to the file
    writer.println(" y=1/sin(x) = " + calc.doCalculation(100));
    writer.println(" y=1/sin(x) = " + calc.doCalculation(90));
    writer.println(" y=1/sin(x) = " + calc.doCalculation(-1111));

    // Input from the user and write the result to the file
    double userInputResult = calc.doCalculationWithInputInside();
    writer.println("Result with our entered number:");
    writer.println(" y=1/sin(x) = " + userInputResult);
    System.out.println("Result written to file: " + filePath);
} catch (IOException e) {
    // Handle IOException (e.g., if there's an issue writing to the file)
    System.out.println("Error writing to file: " + e.getMessage());
}
}
}
```

### **Код CalculateTheEquationInterface.java:**

```
package KI305.Fedus.Lab4;
```

```
/**
 * The CalculateTheEquationInterface interface defines methods for calculating mathematical
 * equations.
 */
public interface CalculateTheEquationInterface {

    /**
     * Calculates the result of the equation  $1/\sin(x)$ .
     *
     * @param variable The input value for the equation.
     * @return The result of the equation  $1/\sin(x)$ .
     */
    double doCalculation(double variable);

    /**
     * Takes user input and calculates the result of the equation  $1/\sin(x)$ .
     *
     */
}
```

```

* @return The result of the equation  $1/\sin(x)$  based on user input.
*/
double doCalculationWithInputInside();
}

```

## Код CalculateTheEquation.java:

```

package KI305.Fedus.Lab4;

```

```

import java.util.InputMismatchException;
import java.util.Scanner;

```

```

/**
 * The CalculateTheEquation class implements the CalculateTheEquationInterface
 * and provides methods for calculating mathematical equations.
 */
public class CalculateTheEquation implements CalculateTheEquationInterface {

    /**
     * Calculates the result of the equation  $1/\sin(x)$ .
     *
     * @param variable The input value for the equation.
     * @return The result of the equation  $1/\sin(x)$ .
     */
    @Override
    public double doCalculation(double variable) {
        try {
            // Calculate and return the result
            return 1 / Math.sin(variable);
        } catch (ArithmeticException e) {
            // Handle arithmetic exception (e.g., division by zero)
            System.out.println("Arithmetic exception: illegal value.");
        }
        // Return 0 in case of an exception
        return 0;
    }

    /**
     * Takes user input and calculates the result of the equation  $1/\sin(x)$ .
     *
     * @return The result of the equation  $1/\sin(x)$  based on user input.
     */
}

```

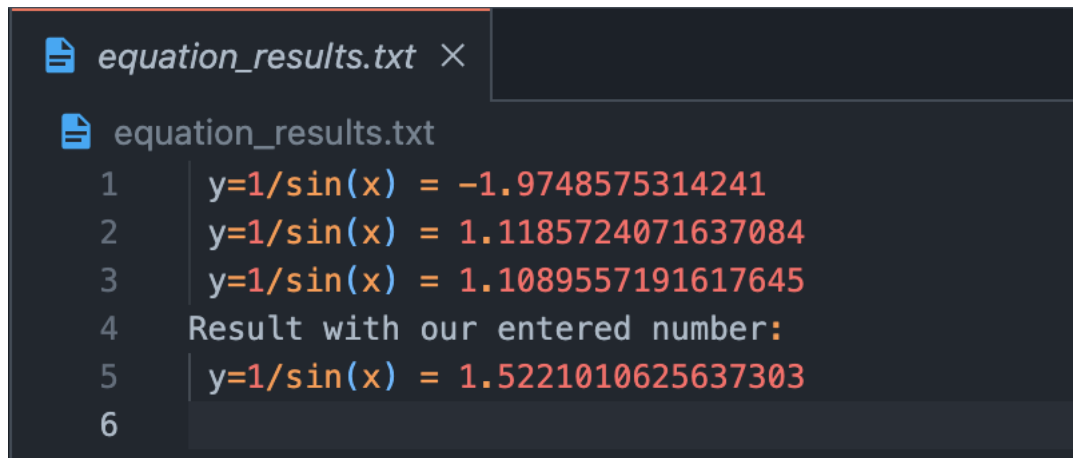
```

@Override
public double doCalculationWithInputInside() {
try {
// Get user input
Scanner inputScanner = new Scanner(System.in);
System.out.print("Enter value: ");
double value = inputScanner.nextDouble();
inputScanner.nextLine();

// Calculate and return the result
return 1 / Math.sin(value);
} catch (ArithmeticException e) {
// Handle arithmetic exception (e.g., division by zero)
System.out.println("Arithmetic exception: illegal value.");
} catch (InputMismatchException e) {
// Handle input mismatch exception (e.g., non-numeric input)
System.out.println("Input exception: illegal value.");
}
// Return 0 in case of an exception
return 0;
}
}

```

### Виконання програми:



```

equation_results.txt
1  y=1/sin(x) = -1.9748575314241
2  y=1/sin(x) = 1.1185724071637084
3  y=1/sin(x) = 1.1089557191617645
4  Result with our entered number:
5  y=1/sin(x) = 1.5221010625637303
6

```

Рис. 1. Результат роботи програми.

## Контрольні питання:

1. Дайте визначення терміну «виключення».

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

2. У яких ситуаціях використання виключень є виправданим?

Відповідь: помилках введення, збоях обладнання, помилках, що пов'язані з фізичними обмеженнями комп'ютерної системи та помилках програмування

3. Яка ієрархія виключень використовується у мові Java?

Відповідь: Всі виключення в мові Java поділяються на контрольовані і неконтрольовані та спадкуються від суперкласу Throwable. Безпосередньо від цього суперкласу спадкуються 2 класи Error і Exception.

4. Як створити власний клас виключень?

Відповідь: заекстендити Error або Exception(або дочірні).

5. Який синтаксис оголошення методів, що можуть генерувати виключення?

Відповідь: [access modifier] [return type] [name]([parameters]) throws Exception  
{  
...  
}

6. Які виключення слід вказувати у заголовках методів і коли?

Відповідь: Оголошувати слід лише всі контрольовані виключення. Якщо цього не зробити, то компілятор видасть повідомлення про помилку. Якщо метод оголошує, що він може генерувати виключення певного класу, то він може також генерувати виключення і його підкласів.

7. Як згенерувати контрольоване виключення?

Відповідь:

1. throw new Exception();

2. Exception ex = new Exception(); throw ex;

8. Розкрийте призначення та особливості роботи блоку try.

Відповідь: Блок виконання коду.

9. Розкрийте призначення та особливості роботи блоку `catch`.

Відповідь: Блок перехоплення виключення.

10. Розкрийте призначення та особливості роботи блоку `finally`.

Відповідь: Блок гарантованого виконання коду після `try-catch` блоку.

### **Висновок:**

Виконавши лабораторну роботу, я оволодів навиками використання механізму виключень при написанні програм мовою Java.