

Міністерство освіти і науки України  
Національний університет «Львівська Політехніка»

Кафедра ЕОМ



## **ЗВІТ**

з лабораторної роботи №4

з дисципліни: “Кросплатформенні засоби програмування”

на тему: «**Виключення**»

Варіант - **19**

Виконав:

ст. гр. КІ-305

Лихограй А. В.

Прийняв:

доцент кафедри ЕОМ

Іванов Ю. С.

Львів 2023

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

**Завдання:**

1. Створити клас, що реалізує метод обчислення виразу  $y = \frac{ctg(x)}{\sin(2x)+4\cos(x)}$ .

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.Lab4 та володіти коментарями, які дозволять автоматично згенерувати документацію до розробленого пакету.

2. Дати відповіді на контрольні запитання:

- дайте визначення терміну «виключення».
- у яких ситуаціях використання виключень є виправданим?
- яка ієрархія виключень використовується у мові Java?
- як створити власний клас виключень?
- який синтаксис оголошення методів, що можуть генерувати виключення?
- які виключення слід вказувати у заголовках методів і коли?
- як згенерувати контрольоване виключення?
- розкрийте призначення та особливості роботи блоку try.
- розкрийте призначення та особливості роботи блоку catch.
- розкрийте призначення та особливості роботи блоку finally.

**Вихідний код програми:**

```
CalcException.java:
package KI305.Lykhohrai.Lab4;
/**
 * Class <code>CalcException</code> more precises ArithmeticException
 * @version 1.0
 */

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

### *Eguations.java*

```
package KI305.Lykhohrai.Lab4;
/**
 * @version 1.0
 */
class Equations
{
    public double calculate(int x) throws CalcException
    {
        double y, rad;
        rad = x * Math.PI / 180.0;
        try
        {
            y = (1 / Math.tan(rad)) / ( Math.sin(2 * rad) + 4 *
Math.cos(rad));
            if (x==90 || x== -90 || x==0 || x== -180 || x==180 )
                throw new ArithmeticException();
        }
        catch (ArithmeticException ex)
        {
            if (x==90 || x== -90 || x== -180 || x==180)
                throw new CalcException("Недопустиме значення при
обчислені значення знаменника!");
            else if (x==0)
                throw new CalcException("Недопустиме значення при
обчислені значення котангенса!");
            else
                throw new CalcException("Інша причина винятку!");
        }
        return y;
    }
}
```

### *EguationsApp.java*

```
package KI305.Lykhohrai.Lab4;
import java.util.Scanner;
import java.io.*;
import static java.lang.System.out;
/**
 * Class <code>EquationsApp</code> Implements driver for Equations class
 * @version 1.0
 */
public class EquationsApp {
    public static void main(String[] args) {
        try
        {
            out.print("Введіть назву файлу, куди зберегти результат: ");
            Scanner in = new Scanner(System.in);
            String fName = in.nextLine();
            PrintWriter fout = new PrintWriter(new File(fName));
            try
            {
                Equations eq = new Equations();
                out.print("Введіть X: ");
                int x = in.nextInt();
                fout.print(eq.calculate(x));
                out.print(eq.calculate(x));
            }
            finally
            {
                fout.flush();
                fout.close();
            }
        }
    }
}
```

```

    }
}
catch (CalcException | FileNotFoundException ex)
{
    if (ex instanceof CalcException) {
        out.print(((CalcException)ex).getMessage());
    } else {
        out.print("Можливо, неправильний шлях до файлу!");
    }
}
}
}
}

```

## Результат виконання програми:

```

"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C
Введіть назву файлу, куди зберегти результат: MyFile.txt
Введіть X: 90
Недопустиме значення при обчисленні значення знаменника!
Process finished with exit code 0

```

Рис. 1. Результат роботи програми.

```

"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C
Введіть назву файлу, куди зберегти результат: MyFile.txt
Введіть X: 30
0.39999999999999997
Process finished with exit code 0

```

Рис. 2. Результат роботи програми

## Фрагмент згенерованої документації:

PACKAGE **CLASS** TREE INDEX HELP

SUMMARY NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD SEARCH

Package KI305.Lykhohrai.Lab4

**Class EquationsApp**

java.lang.Object<sup>Ⓢ</sup>  
KI305.Lykhohrai.Lab4.EquationsApp

---

public class EquationsApp  
extends Object<sup>Ⓢ</sup>

Class EquationsApp Implements driver for Equations class

Version:  
1.0

**Constructor Summary**

Constructor	Description
EquationsApp()	

**Method Summary**

Modifier and Type	Method	Description
static void	main(String <sup>Ⓢ</sup> [] args)	

Methods inherited from class java.lang.Object<sup>Ⓢ</sup>

Рис. 3. Фрагмент документації.

## Відповіді на контрольні запитання:

### 1. Визначення терміну «виключення»:

- Виключення (або помилка) - це ситуація, що виникає під час виконання програми і вказує на помилку або непередбачуваний стан.

### 2. Ситуації використання виключень:

- Обробка помилок і винятків.
- Забезпечення нормального виконання програми при виникненні непередбачуваних обставин.

### 3. Ієрархія виключень в мові Java:

- Всі виключення є підкласами класу `Throwable`. Він має два основних підкласи: `Error` (непоправні помилки, які зазвичай не обробляються програмою) та `Exception` (переважно виняткові ситуації, які можуть бути оброблені програмою).

### 4. Створення власного класу виключень:

- Створіть клас, що наслідується від `Exception` або його підкласів.

### 5. Синтаксис оголошення методів, що можуть генерувати виключення:

```
```java  
  
returnType methodName(parameters) throws ExceptionType {  
  
    // тіло методу  
  
}  
  
```
```

### 6. Вказання виключень у заголовках методів:

- Вказуйте виключення, якщо метод може генерувати певний тип помилки. Виключення слід вказувати для всіх контрольованих виключень.

### 7. Генерація контрольованого виключення:

- Створіть об'єкт відповідного класу виключення і використайте оператор ``throw``.

#### 8. Блок ``try``:

- Блок, в якому розміщується код, який може призвести до виникнення виключення.

#### 9. Блок ``catch``:

- Блок, який обробляє виключення, виниклі в блоку ``try``. Може бути декілька блоків ``catch`` для обробки різних типів виключень.

#### 10. Блок ``finally``:

- Блок, який завжди виконується, незалежно від того, чи виникло виключення, чи ні. Використовується для звільнення ресурсів чи завершення коду, який повинен виконатися завжди.

**Висновок:** під час виконання лабораторної роботи я ознайомився з виключеннями в мові Java. Навчився працювати з ними. Розробив програму, яка обчислює вираз:  $y = \frac{\operatorname{ctg}(x)}{\sin(2x) + 4\cos(x)}$ .