

Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра ЕОМ



Звіт

З лабораторної роботи №4

Варіант – 3

З дисципліни: «Кросплатформні засоби програмування»

На тему: «Виключення»

Виконав: ст. гр. КІ-306

Братівник Д. А.

Прийняв:

доцент кафедри ЕОМ

Іванов Ю. С.

Львів 2023

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

Завдання(Варіант 3)

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

Варіант завдання: $y = \sin(x) / \cos(x)$

Код програми:

Файл EquationsApp.java:

```
package org.example; // Замініть на відповідний пакет

import java.util.Scanner;
import java.io.*;
import static java.lang.System.out;

/**
 * Клас <code>EquationsApp</code> реалізує головний драйвер для
 класу Equations.
 * Версія 1.0
 */
public class EquationsApp {
    public static void main(String[] args) {
        out.print("Введіть ім'я файлу: ");
        Scanner in = new Scanner(System.in);
        String fName = in.nextLine();

        PrintWriter fout = null;

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

            Equations eq = new Equations();
```

```

        out.print("Введіть X: ");
        int x = in.nextInt();

        double result = eq.calculate(x);

        fout.print("Результат: " + result);
        fout.flush();
        fout.close();
        out.println("Результат: " + result);
    } catch (FileNotFoundException ex) {
        out.print("Причина винятку: Файл не знайдено");
    } catch (CalcException ex) {
        out.print(ex.getMessage());
    } finally {
        if (fout != null) {
            fout.close();
        }
    }
}
}

```

Файл CalcException.java:

```

package org.example;

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

/**
 * Клас <code>Equations</code> реалізує метод для обчислення
виразу ((sin(x) / cos(x)).
 * Автор: [Ваше_Прізвище]
 * Версія 1.0
 */
class Equations {
    /**
     * Метод обчислює вираз ((sin(x) / cos(x)).
     *
     * @param x Значення X, для якого проводиться обчислення.
     * @return Результат обчислення виразу ((sin(x) / cos(x)).
     * @throws CalcException якщо під час обчислення виникає
ВИНЯТОК.
     */
    public double calculate(int x) throws CalcException {
        double y, rad;
        rad = x * Math.PI / 180.0;

        try {
            y = Math.sin(rad) / Math.cos(rad);

```

```

        if (Double.isNaN(y) || Double.isInfinite(y) || x == 90
|| x == -90) {
            throw new ArithmeticException();
        }
    } catch (ArithmeticException ex) {
        if (rad == Math.PI / 2.0 || rad == -Math.PI / 2.0) {
            throw new CalcException("Причина винятку:
Некоректне значення X для обчислення тангенсу");
        } else if (x == 0) {
            throw new CalcException("Причина винятку: X = 0");
        } else {
            throw new CalcException("Невідома причина винятку
під час обчислення виразу");
        }
    }

    return y;
}
}

```

Результати роботи програми:

```

/Library/Java/JavaVirtualMachines/jdk-20.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA.app/Contents/lib/idea_rt.jar=60186:/Applications/IntelliJ I
Введіть ім'я файлу: Main.txt
Введіть X: 11
Результат: 0.2125565616700221

Process finished with exit code 0

```

EquationsApp.java ×		Main.txt ×		CalcException.java ×	
1	Результат: 0.2125565616700221				

Фрагмент згенерованої документації

The screenshot displays a web-based code editor and documentation viewer. The top section shows a file explorer for a project named 'doclab2'. The files are organized into a grid, including HTML files like 'allclasses-index.html', 'allpackages-index.html', 'help-doc.html', 'index.html', and 'index-files', as well as JavaScript files like 'member-search-index.js', 'module-search-index.js', 'package-search-index.js', 'script.js', 'search-page.js', 'search.html', 'search.js', 'tag-search-index.js', and 'type-search-index.js'. There are also CSS files like 'stylesheet.css' and a 'copy.svg' icon.

The bottom section shows the documentation for the 'EquationsApp' class. The package is 'org.example'. The class is 'EquationsApp', which extends 'Object'. The documentation includes a 'Constructor Summary' section with a table showing the constructor 'EquationsApp()'. It also includes a 'Method Summary' section with a table showing the method 'main(String[] args)'. The 'Methods inherited from class java.lang.Object' section lists methods like 'equals', 'getClass', 'hashCode', 'notify', 'notifyAll', 'toString', 'wait', and 'wait'.

Відповіді на контрольні запитання

1. Дайте визначення терміну «виключення».
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?

помилках введення, наприклад, при введенні назви неіснуючого файлу або Інтернет адреси з подальшим зверненням до цих ресурсів, що призводить до генерації помилки системним програмним забезпеченням;
збоях обладнання;
помилках, що пов'язані з фізичними обмеженнями комп'ютерної системи, наприклад, при заповненні оперативної пам'яті або жорсткого диску;
помилках програмування, наприклад, при некоректній роботі методу, читанні елементів порожнього стеку, виходу за межі масиву тощо.

3. Яка ієрархія виключень використовується у мові Java?

Всі виключення в мові Java поділяються на контрольовані і неконтрольовані та спадкуються від суперкласу Throwable

4. Як створити власний клас виключень?

Для створення власного класу виключень в Java, спадкуйте ваш клас від одного з існуючих класів контрольованих виключень, додайте конструктори та використовуйте його для генерації виключень у вашому коді.

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

```
public Return Type methodName(Parameters) throws ExceptionType {  
    // Код методу  
}
```

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

ті виключення, які можуть бути згенеровані з внутрішнього методу і які повинні оброблятися викликаючим кодом.

7. Як згенерувати контрольоване виключення?

Генерація контрольованих виключень відбувається за допомогою ключового слова throw після якого необхідно вказати об'єкт класу виключення який і є власне виключенням, що генерує метод

8. Розкрийте призначення та особливості роботи блоку try.

Блок try використовується для обгортання коду, який може генерувати виключення. Він служить для відстеження виключень під час виконання коду в блоку.

9. Розкрийте призначення та особливості роботи блоку catch.

Блок `catch` використовується для обробки виключень, які були сгенеровані в блоку `try`. Може бути кілька блоків `catch` для обробки різних типів виключень.

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

Блок `finally` використовується для виконання коду, який повинен виконатися завжди, незалежно від того, чи виникло виключення чи ні. Це корисно, наприклад, для звільнення ресурсів.

Висновок

У ході виконання даної лабораторної роботи, я отримав навички використання механізму виключень при написанні програм мовою Java. Я вивчив, як обробляти винятки та використовувати блоки `try`, `catch` і `finally` для забезпечення безпеки та надійності мого коду.