

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра ЕОМ



Звіт

Лабораторна робота № 4

З дисципліни «Кросплатформні засоби програмування»

На тему: «Виключення»

Виконала: ст. гр. КІ-303

Кілик І.Р.

Перевірів:

доцент кафедри ЕОМ

Іванов Ю.С.

Львів 2025

ВИКЛЮЧЕННЯ

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

ТЕОРЕТИЧНІ ВІДОМОСТІ

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

- 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.

Ієрархія класів виключень

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 (див. рис. 1). Ієрархія класів, що спадкує клас Error, описує внутрішні помилки і ситуації, що пов'язані з браком ресурсів у системі підтримки виконання програм. Жоден об'єкт цього типу самостійно згенерувати неможна. При виникненні внутрішньої помилки можна лише відобразити повідомлення користувачу та спробувати коректно завершити виконання програми. Такі помилки є нечастими.

Ієрархія класів, що спадкує клас Exception поділяється на клас RuntimeException та інші. Виключення типу RuntimeException виникають

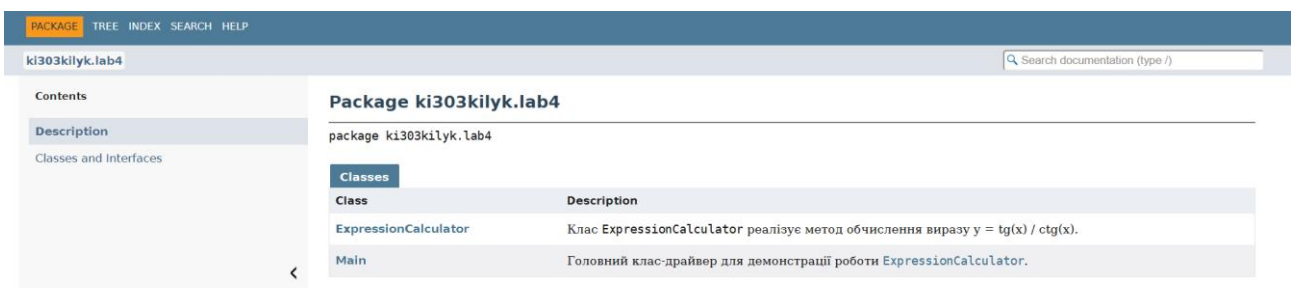
внаслідок помилок програмування. Всі інші помилки є наслідком непередбачених подій, що виникають під час виконання коректної програми, наприклад, помилок вводу/виводу. Класи, що спадкуються від `Error` та `RuntimeException`, відносяться до

неконтрольованих виключень. Всі інші класи відносяться до контрольованих виключень. Лише контрольовані виключення можуть бути згенеровані програмістом у коді програми явно за допомогою ключового слова `throw`. Для всіх контрольованих виключень компілятор перевіряє наявність відповідних обробників.

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.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 - $y = \text{tg}(x) / \text{ctg}(x)$



The screenshot shows the Java documentation for the package `ki303kilyk.lab4`. The left sidebar contains a 'Contents' section with 'Description' and 'Classes and Interfaces'. The main content area is titled 'Package ki303kilyk.lab4' and shows a table of classes.

Class	Description
<code>ExpressionCalculator</code>	Клас <code>ExpressionCalculator</code> реалізує метод обчислення виразу $y = \text{tg}(x) / \text{ctg}(x)$.
<code>Main</code>	Головний клас-драйвер для демонстрації роботи <code>ExpressionCalculator</code> .

Висновок: Завдяки виконанню даної лабораторної роюоти, я оволоділа навиками використання механізму виключень при написанні програм мовою Java.

Додатки

```
package ki303kilyk.lab4;

/**
 * Клас {@code ExpressionCalculator} реалізує метод обчислення виразу  $y = \operatorname{tg}(x) / \operatorname{ctg}(x)$ .
 * <p>
 * Виконує перевірку вхідних даних і може генерувати виключення при некоректних обчисленнях.
 * </p>
 *
 * @author
 * Кілик Ірина, KI-303
 * @version 1.0
 */
public class ExpressionCalculator {

    /**
     * Обчислює значення виразу  $y = \operatorname{tg}(x) / \operatorname{ctg}(x)$ .
     *
     * @param x аргумент виразу (в радіанах)
     * @return результат обчислення
     * @throws ArithmeticException якщо обчислення неможливе (ділення на нуль тощо)
     */
    public static double calculate(double x) throws ArithmeticException {
        double tan = Math.tan(x);
        double cot = 1 / tan;

        if (Double.isNaN(tan) || Double.isNaN(cot) || Double.isInfinite(tan) || Double.isInfinite(cot)) {
            throw new ArithmeticException("Некоректне значення тангенса або котангенса для x = " + x);
        }

        return tan / cot;
    }
}
```

```
package ki303kilyk.lab4;

import java.io.FileWriter;
import java.io.IOException;
```

```

import java.util.Scanner;

/**
 * Головний клас-драйвер для демонстрації роботи {@link ExpressionCalculator}.
 * <p>
 * Зчитує значення x з клавіатури, обчислює результат і записує його у файл.
 * </p>
 */
public class Main {

    /**
     * Точка входу в програму.
     *
     * @param args аргументи командного рядка (не використовуються)
     */
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Введіть значення x (у радіанах): ");

        try {
            double x = scanner.nextDouble();

            double y = ExpressionCalculator.calculate(x);
            System.out.println("Результат обчислення y = " + y);

            // Запис результату у файл
            try (FileWriter writer = new FileWriter("result_lab4.txt")) {
                writer.write("Вхідне значення x = " + x + "\n");
                writer.write("Результат y = tg(x)/ctg(x) = " + y + "\n");
                System.out.println("Результат записано у файл result_lab4.txt");
            }

        } catch (ArithmeticException e) {
            System.out.println("Помилка обчислення: " + e.getMessage());
        } catch (IOException e) {
            System.out.println("Помилка запису у файл: " + e.getMessage());
        } catch (Exception e) {
            System.out.println("Сталася непередбачена помилка: " + e.getMessage());
        } finally {
            scanner.close();
        }
    }
}

```