Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра «Електронних обчислювальних машин»



**Звіт**

з лабораторної роботи № 4

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

**на тему:** «Виключення»

**Виконав:**

студент групи *КІ-306*

*Ярмола Юрій*

**Прийняв:**

доцент кафедри ЕОМ

Іванов Ю. С.

**Львів – 2023**

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

**Завдання ( Варіант 30 y=cos(x)/(x+2ctg(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.

**Вихідний код програми**

ExpressionCalculator.java

package KI306.Yarmola.Lab4;

import java.io.FileWriter;

import java.io.IOException;

import java.io.PrintWriter;

/\*\*

\* Клас ExpressionCalculator виконує обчислення виразу та запис результатів у файл.

\* Вираз: y = cos(x) / (x + 2 \* ctg(x))

\*/

public class ExpressionCalculator {

private double x;

/\*\*

\* Конструктор для створення об'єкта ExpressionCalculator зі значенням x.

\*

\* @param x Значення x, для якого буде обчислюватися вираз.

\*/

public ExpressionCalculator(double x) {

this.x = x;

}

/\*\*

\* Обчислює вираз y = cos(x) / (x + 2 \* ctg(x)).

\*

\* @return Результат обчислення виразу.

\* @throws ArithmeticException Виникає, якщо виникає помилка при діленні на нуль у виразі.

\*/

public double calculateExpression() throws ArithmeticException {

double ctgX = 1 / Math.tan(x);

if (Math.abs(ctgX) < 1e-6) {

throw new ArithmeticException("Ділення на нуль у виразі.");

}

return Math.cos(x) / (x + 2 \* ctgX);

}

/\*\*

\* Записує результат обчислення виразу у файл "result.txt".

\*

\* @param result Результат обчислення виразу.

\* @throws IOException Виникає, якщо виникає помилка при записі у файл.

\*/

public void saveResultToFile(double result) throws IOException {

try (PrintWriter writer = new PrintWriter(new FileWriter("result.txt"))) {

writer.println("Результат обчислення виразу: " + result);

}

}

}

Main.java

import KI306.Yarmola.Lab4.ExpressionCalculator;

import java.io.IOException;

import java.util.Scanner;

/\*\*

\* Головний клас програми для обчислення виразу та збереження результату у файл.

\*

\* @author Yurii Yarmola

\* @since 1.0

\*/

public class Main {

/\*\*

\* Точка входу в програму.

\*

\* @param args Масив аргументів командного рядка.

\*/

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Введіть значення x: ");

double x = scanner.nextDouble();

ExpressionCalculator calculator = new ExpressionCalculator(x);

try {

double result = calculator.calculateExpression();

calculator.saveResultToFile(result);

System.out.println("Результат обчислення: " + result);

} catch (ArithmeticException e) {

System.err.println("Помилка обчислення: " + e.getMessage());

} catch (IOException e) {

System.err.println("Помилка при записі до файлу: " + e.getMessage());

}

}

}

**Результат виконання програми**

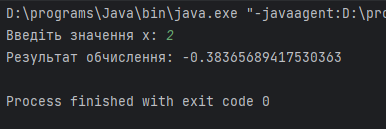
****

Рис.1 – Скріншот виконання програми

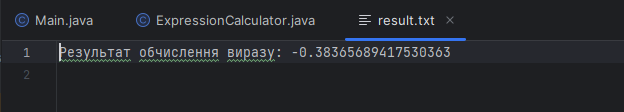
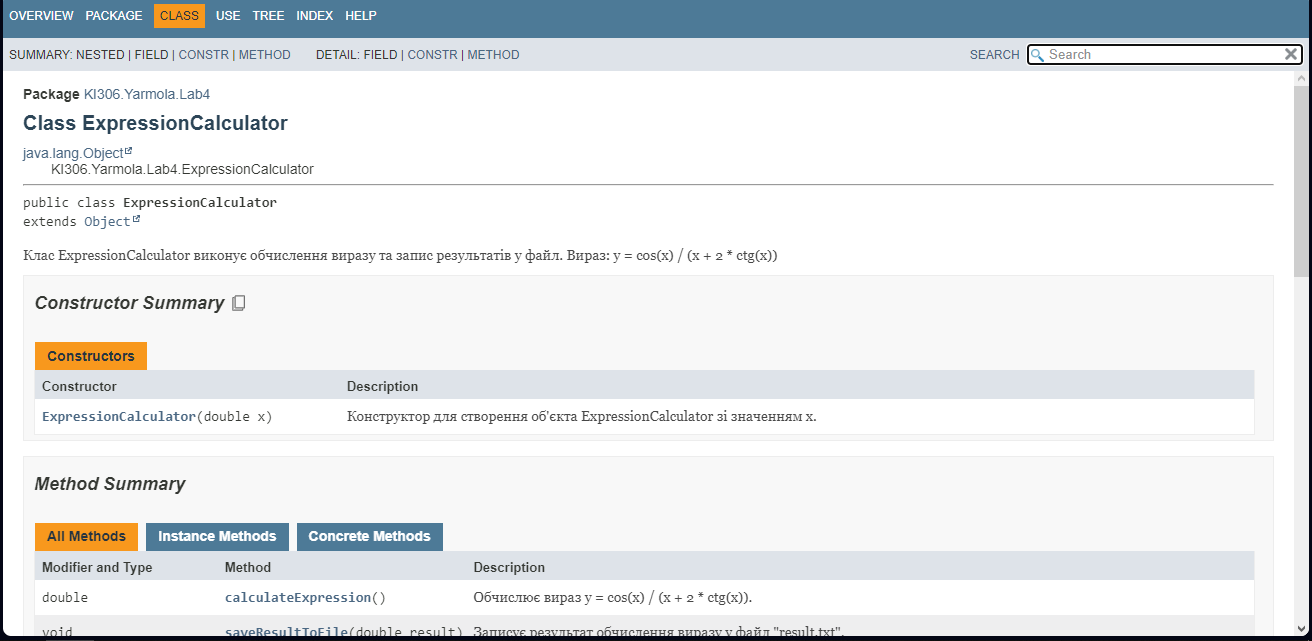
****

Рис.2 – Скріншот документу згенерованого програмою

**Фрагмент згенерованої документації**



**Відповіді на контрольні запитання**

1. Дайте визначення терміну «виключення».

Виключення - це подія або стан, який виникає в програмі і може призвести до порушення нормального ходу програми.

1. У яких ситуаціях використання виключень є виправданим?

Обробка помилок, які виникають під час виконання програми.

Повідомлення користувачам або іншим частинам програми про помилки або непередбачені ситуації.

Виправлення помилок і забезпечення стійкості програми.

1. Яка ієрархія виключень використовується у мові Java?

У мові Java існує ієрархія виключень, де базовий клас - java.lang.Throwable. Вона поділяється на дві основні групи: перевірювані (checked) та неперевірювані (unchecked) (підкласи RuntimeException). Кожен виняток є об'єктом певного класу і може бути викинутий за допомогою оператора throw.

1. Як створити власний клас виключень?

Для створення власного класу виключень потрібно створити підклас класу Exception або RuntimeException (або їх підкласи) і перевизначити конструктори, які вказують повідомлення про помилку.

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

returnType methodName(parameters) throws ExceptionType1, ExceptionType2 {

// код методу

}

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

У заголовках методів слід вказувати ті виключення, які можуть бути викинуті методом. Це включає в себе всі перевірювані виключення (checked exceptions) та будь-які інші виключення, які не є підкласами RuntimeException.

1. Як згенерувати контрольоване виключення?

if (someCondition) {

throw new SomeException("Повідомлення про помилку");

}

1. Розкрийте призначення та особливості роботи блоку try.

Блок try використовується для визначення коду, який потребує обробки виключень. Всі інструкції в блоку try виконуються, і якщо виникає виключення, вони переключаються до відповідного блоку catch.

1. Розкрийте призначення та особливості роботи блоку catch.

Блок catch використовується для обробки виключень, які були викинуті у блоку try. Кожен catch блок обробляє певний тип виключення.

1. Розкрийте призначення та особливості роботи блоку finally.

Блок finally використовується для визначення коду, який завжди буде виконуватися, незалежно від того, чи сталася помилка, чи ні. Він використовується для виконання завершальних дій, таких як закриття ресурсів (наприклад, файлів), навіть якщо виникають виключення.

**Висновок**

Ознайомився з виключеннями у мові Java. Написав клас розв’язання виразу та клас драйвер з використанням виключень.