

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

Хмільовський С. Р.

Прийняв:

доцент кафедри ЕОМ

Іванов Ю. С.

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

Завдання (варіант № 22)

1. Створити клас, що реалізує метод обчислення виразу заданого варіантом ($y = \tan(4x)/x$). 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. 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.

Вихідний код програми:

Файл EquationSolver.java

```
//package KI306.Khmilovskiy.Lab4;

import java.io.FileWriter;
import java.io.IOException;
import java.io.BufferedWriter;

/**
 * The {@code EquationSolver} class provides methods for solving the equation
 * and saving the result to a file.
 *
 * The equation is defined as:  $y = \tan(4x) / x$ 
 *
 * @author Khmilovskiy Stanislav
 * @version 1.0
 * @since 1.0
 */
public class EquationSolver {
    /**
     * Calculates the value of the expression  $y = \tan(4x) / x$ .
     */
}
```

```

*
* @param x The value of the variable x.
* @return The result of the expression.
* @throws ArithmeticException If x is equal to 0, division by zero is not
allowed.
*/
    public static double calculateExpression(double x) throws
ArithmeticException {
        if (x == 0) {
            throw new ArithmeticException("Ділення на нуль неможливе!");
        }
        // Обчислення виразу:
        double result = Math.tan(4 * x) / x;
        return result;
    }

/**
 * Saves the result of the expression calculation to a file.
 *
 * @param result The result of the expression calculation.
 * @param x The value of the variable x.
 * @param fileName The name of the file to save the result.
 * @throws IOException If an error occurs while writing to the file.
 * @throws IllegalArgumentException If the fileName is null or empty.
 */
    public static void saveResultToFile(double result, double x, String
fileName) throws IOException {
        // Перевірка аргументів на коректність:
        if (fileName == null || fileName.isEmpty()) {
            throw new IllegalArgumentException("Неправильне ім'я файлу!");
        }
        try (BufferedWriter writer = new BufferedWriter(new
FileWriter(fileName))) {
            writer.write("Вираз: y=tg(4x)/x\n");
            writer.write("Значення змінної \"x\": " + x + "\n");
            writer.write("\nРезультат обчислення виразу: " + result);

        } catch (IOException e) {
            // Обробка помилки запису до файлу:
            System.err.println("Помилка під час запису до файлу: " +
e.getMessage());
            throw e; // Прокидання винятку далі для обробки на вищому рівні
        }
    }
}

```

Файл EquationSolverApp.java

```

//package KI306.Khmilovskiy.Lab4;

import java.io.IOException;
import java.util.Scanner;

/**
 * The {@code EquationSolverApp} class is a program that allows users to input a
value for the variable 'x',
 * calculate the result of the expression  $y = \tan(4x) / x$  using the {@link
EquationSolver} class, and save the result to a file.
 *
 * Users are prompted to enter a double value for 'x', and the program
calculates the result of the expression
 * and saves it to a file named "result.txt". Error handling is provided for
input validation and file operations.
 *
 * @author Khmilovskiy Stanislav

```

```

* @version 1.0
* @since 1.0
*/
public class EquationSolverApp {
    /**
     * The main method of the application.
     *
     * @param args The command-line arguments (not used).
     */
    public static void main(String[] args) {
        try {
            //4 test:
            //double x = 2.0;
            Scanner scanner = new Scanner(System.in);
            System.out.print("Введіть значення \"x\" [double]: ");

            double x = scanner.nextDouble();
            double result = EquationSolver.calculateExpression(x);
            System.out.println("Результат обчислення виразу: " + result);

            EquationSolver.saveResultToFile(result, x, "result.txt");
            System.out.println("Результат записано у файл \"result.txt\"");

        } catch (java.util.InputMismatchException e) {
            System.err.println("Помилка: Ви ввели нечислові значення.");
        } catch (ArithmeticException e) {
            System.err.println("Помилка: " + e.getMessage());
        } catch (IOException e) {
            System.err.println("Помилка під час запису до файлу: " +
e.getMessage());
        }
    }
}

```

Результат виконання програми:

```

Run EquationSolverApp x
D:\Software\JDK 20\bin\java.exe" --javaagent:D:\Software\IntelliJ IDEA Community Edition 2023.2\lib\idea_rt.jar=55151:D:\Software\IntelliJ IDEA Community Edition 2023.2\bin" -i
Введіть значення "x" [double]: 0,96
Результат обчислення виразу: 0.8745515253041173
Результат записано у файл "result.txt"
Process finished with exit code 0

```

Вміст файлу result.txt:

```

result: Блокнот
Файл Редагування Формат Вигляд Довідка
Вираз: y=tg(4x)/x
Значення змінної "x": 0.96

Результат обчислення виразу: 0.8745515253041173

```

Фрагмент згенерованої документації JavaDoc:

PACKAGE

CLASS

TREE

INDEX

HELP

SUMMARY: NESTED | FIELD | CONSTR | METHOD

DETAIL: FIELD | CONSTR | METHOD

SEARCH

Class EquationSolver

java.lang.Object[Ⓜ]
EquationSolver

public class EquationSolver
extends Object[Ⓜ]

The EquationSolver class provides methods for solving the equation and saving the result to a file. The equation is defined as: $y = \tan(4x) / x$

Since:
1.0

Constructor Summary

Constructors

| Constructor | Description |
|------------------|-------------|
| EquationSolver() | |

Method Summary

All MethodsStatic MethodsConcrete Methods

| Modifier and Type | Method | Description |
|-------------------|---|---|
| static double | calculateExpression(double x) | Calculates the value of the expression $y = \tan(4x) / x$. |
| static void | saveResultToFile(double result, double x, String [Ⓜ] fileName) | Saves the result of the expression calculation to a file. |

Відповіді на контрольні запитання:

1. Дайте визначення терміну «виключення».

Виключення (або Exception) - це подія або об'єкт, що виникає під час виконання програми і може порушити нормальний хід виконання програми. Виключення вказують на помилку або несподівану ситуацію, яка вимагає обробки.

2. У яких ситуаціях використання виключень є виправданим?

Виключення слід використовувати, коли програма стикається з помилкою або непередбачуваною ситуацією, і ви хочете обробити цю ситуацію, щоб уникнути аварійного завершення програми. Це може включати в себе роботу з файлами, мережами, даними користувача тощо.

3. Яка ієрархія виключень використовується у мові Java?

У мові Java виключення представлені в ієрархічній структурі, починаючи з класу “java.lang.Throwable”. Ця ієрархія розділена на дві основні гілки: “java.lang.Error” (помилки, які не можна відновити) і “java.lang.Exception” (загальні виключення, які можна обробляти). Від “Exception” походять багато підкласів, наприклад, “RuntimeException”, “IOException”, “SQLException”, і т. д.

4. Як створити власний клас виключень?

Для створення власного класу виключень у Java вам потрібно створити клас, який успадковується від java.lang.Exception або від одного з його підкласів. Ваш

клас повинен мати конструктор і може містити додаткові поля та методи, що допомагатимуть при обробці виключення.

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

```
public void myMethod() throws MyCustomException {  
}
```

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

Виключення вказуються у заголовках методів за допомогою throws, якщо метод може викидати ці виключення. Це допомагає іншим програмістам розуміти, які виключення можуть виникнути при виклику методу і обробляти їх відповідно.

7. Як згенерувати контрольоване виключення?

```
throw new MyCustomException("Власне виключення");
```

8. Розкрийте призначення та особливості роботи блоку try.

Блок “try” використовується для обрамлення коду, в якому можуть виникати виключення. Якщо виключення виникає в блоці “try”, воно може бути оброблене в одному або кількох блоках “catch”. Якщо виключення не виникає, блок “catch” не виконується.

9. Розкрийте призначення та особливості роботи блоку catch.

Блок “catch” використовується для обробки виключень, які виникли в блоці “try”. Кожен блок “catch” вказує клас виключення, який він обробляє, і містить код обробки.

10. Розкрийте призначення та особливості роботи блоку finally.

Блок “finally” використовується для виконання коду, який завжди повинен виконуватися, незалежно від того, чи виникло виключення, чи ні. Це дозволяє вам виконати завершальні дії, такі як закриття ресурсів, навіть у випадку помилки.

Висновок:

На даній лабораторній роботі я отримав навички використання механізму виключень при написанні програм мовою Java.