Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Звіт

до лабораторної роботи №5

З дисципліни: «Кросплатформенні засоби програмування»

На тему: «Виключення»

Виконав:

Студент групи КІ-34

Сенета Б.Р.

Прийняв:

Іванов Ю. С.

Львів 2022

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

Виконання роботи

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.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.

Завдання:

19. $y=ctg(x)/(\sin(2x) + 4\cos(x))$

Код програми:

Клас EquationsApp

```
import KI34.SENETA.LAB5.*;

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

public class EquationsApp {
    public static void main(String[] args) {
        try {
            out.print("Enter file name: ");
            Scanner in = new Scanner(System.in);
            String fName = in.nextLine();
            PrintWriter fout = new PrintWriter(new File(fName));
            try {
                try {
                    Equations eq = new Equations();
                    out.print("Enter X: ");
                    double result = eq.calculate(in.nextInt());
                    System.out.println("Result: " + result);
                    fout.print(result);
```

```
                } finally {
                    fout.flush();
                    fout.close();
                }
            } catch (CalculationException ex) {
                out.print(ex.getMessage());
            }
        } catch (FileNotFoundException ex) {
            out.print("Exception reason: Perhaps wrong file path");
        }
    }
}
```

Клас Equations

```
package KI34.SENETA.LAB5;

public class Equations {
    public double calculate(int x) throws CalculationException{double y,
        rad;
        rad = x * Math.PI / 180.0;try{
            y = (1.0 / Math.tan(x)) / (Math.sin(2 * rad) + 4 * Math.cos(rad));
            if(y == Double.NaN || y == Double.NEGATIVE_INFINITY||
                    y == Double.POSITIVE_INFINITY || x == 90 || x == -90){
                throw new ArithmeticException();
            }
        } catch (ArithmeticException ex) {
            // створимо виключення вищого рівня з поясненням причини
            // виникнення помилки
            if (rad==Math.PI/2.0 || rad==-Math.PI/2.0)
                throw new CalculationException("Exception reason: Illegal value of " +
                        "X for tangent calculation");
            else throw new CalculationException("Unknown reason of the exception" +
                        "during exception calculation");
        }
        return y;
    }
}
```

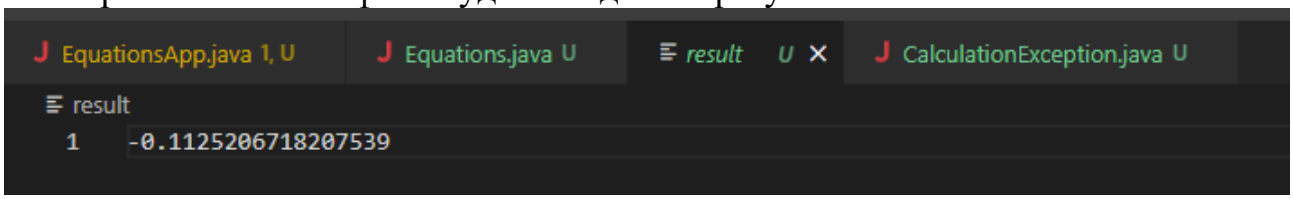
Клас CalculationException

```java
package KI34.SENETA.LAB5;

public class CalculationException extends ArithmeticException{
    public CalculationException(){
    }

    public CalculationException(String cause){
        super(cause);
    }
}
```

Результат роботи програми:

```
Enter file name: result
Enter X: 2
Result: -0.1125206718207539
```

Створено текстовий файл куди виводиться результат:



Успішно опрацьовує помилки:

```
Enter file name: Result.txt
Enter X: 90
Exception reason: Illegal value of X for tangent calculation
```

Відповіді на КЗ

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

2. 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.

3. 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.

4. Як правило, власні класи контрольованих виключень використовуються для конкретизації виключних ситуацій, що генеруються стандартними класами контрольованих виключень, з метою їх точнішого опрацювання. Для створення власного класу контрольованих виключень необхідно обов’язково успадкувати один з існуючих класів контрольованих виключень та розширити його новою функціональністю

5. public int loadData(String fName) throws EOFException, MalformedURLException { … }

6. Мова Java дає розробнику вибір або самому перехопити і опрацювати виключення у методі, або делегувати це право іншому методу. Як правило, слід перехоплювати лише ті виключення, які ви самі можете опрацювати, або, які були створені вами. Решту виключень слід передавати далі по ланцюгу викликаних методів.

7. 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.

8. – 9. – 10.

1. Якщо код у блоці try не генерує ніяких виключень, то програма спочатку повністю виконає блок try, а потім блок finally.

2. Якщо код у блоці try згенерував виключення, то подальше виконання коду в цьому блоці припиняється і відбувається пошук блоку catch тип у заголовку якого співпадає з типом виключення після чого виконується блок finally. Якщо виключення генерується у блоці catch, то після виконання блоку finally керування передається викликаючому методу.

3. Якщо виключення не опрацьовується у жодному з блоків catch, то виконується блок finally і керування передається викликаючому методу.

4. Якщо ж блоки finally і catch відсутні, то керування передається викликаючому методу.

Висновок: оволодів навиками використання механізму виключень при написанні програм мовою Java.