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-307

Трофименко Є.Е.

**Прийняв:**

доцент кафедри ЕОМ

Іванов Ю. С.

Львів – 2023

**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. 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**

*EquationsApp.java*

package ki307.Trofymenko.Lab4;

/\*\*

\* Клас реалізації методу обчислення виразу заданого варіантом

\* **@version** 1.0

\*/

public class EquationsApp {

/\*\*

\* Конструктор за замовченням

\*/

public EquationsApp() {}

/\*\*

\* Вхідний метод програми

\* **@param** args визначення методу main

\*/

public static void main(String[] args) {

CalculateTheEquationInterface calc = new CalculateTheEquation();

System.***out***.println(" y=tg(4x)/x = " + calc.doCalculation(100));

System.***out***.println(" y=tg(4x)/x = " + calc.doCalculation(80));

System.***out***.println(" y=tg(4x)/x = " + calc.doCalculation(-5555));

System.***out***.println(" y=tg(4x)/x = " + calc.doCalculationWithInputInside());

}

}

*CalculateTheEquationInterface.java*

package ki307.Trofymenko.Lab4;

public interface CalculateTheEquationInterface {

double doCalculation(double variable);

double doCalculationWithInputInside();

}

*CalculateTheEquation.java*

package ki307.Trofymenko.Lab4;

import java.util.InputMismatchException;

import java.util.Scanner;

public class CalculateTheEquation implements CalculateTheEquationInterface {

*@Override*

public double doCalculation(double variable) {

try {

return Math.*tan*(variable);

} catch (ArithmeticException e){

System.***out***.println("Arithmetic exception: illegal value.");

}

return 0;

}

*@Override*

public double doCalculationWithInputInside() {

try {

Scanner inputScanner = new Scanner(System.***in***);

System.***out***.print("Enter value: ");

double value = inputScanner.nextDouble();

inputScanner.nextLine();

return Math.*tan*(4\*value) / (value);

} catch (ArithmeticException e){

System.***out***.println("Arithmetic exception: illegal value.");

}

catch (InputMismatchException e){

System.***out***.println("Input exception: illegal value.");

}

return 0;

}

}

**Результат виконання програми**

Изображение выглядит как текст, Шрифт, снимок экрана, число

Автоматически созданное описание

**Фрагмент згенерованої документації**

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, линия, чек

Автоматически созданное описание

**Відповіді на контрольні запитання**

1. Дайте визначення терміну «виключення».

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

1. У яких ситуаціях використання виключень є виправданим?

*Використовується при помилках введення, збоях обладнання, помилках, що пов’язані з фізичними обмеженнями комп’ютерної системи та помилках програмування.*

1. Яка ієрархія виключень використовується у мові Java?

*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.*

1. Як створити власний клас виключень?

*Для створення власного класу контрольованих виключень необхідно обов’язково успадкувати один з існуючих класів контрольованих виключень та розширити його новою функціональністю.*

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

*Приклад оголошення методу, що може генерувати виключення:*

*public int loadData(String fName) throws EOFException, MalformedURLException*

*{*

*...*

*}*

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

*Оголошувати слід лише всі контрольовані виключення. Якщо цього не зробити, то компілятор видасть повідомлення про помилку. Якщо метод оголошує, що він може генерувати виключення певного класу, то він може також генерувати виключення і його підкласів.*

1. 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.*

1. Розкрийте призначення та особливості роботи блоку try.

*Блок try в мові програмування Java використовується для визначення блоку коду, в якому можуть виникнути винятки.*

*Особливості роботи блоку try:*

1. *Весь код, який може викликати виняткову ситуацію, розміщується всередині блоку try.*
2. *Якщо виняток виникає всередині блоку try, виконання коду в блоку припиняється, і управління передається відразу до блоку catch.*
3. Розкрийте призначення та особливості роботи блоку catch.

*Блок catch використовується для обробки виняткових ситуацій, які виникають всередині блоку try. Кожен блок catch пов'язаний з конкретним типом винятку, і тільки той блок catch, який відповідає типу винятку, викликаному у блоку try, буде виконаний.*

*Особливості роботи блоку catch:*

1. *Блок catch вказує тип винятку, який він може обробити, і має доступ до об'єкта винятку для подальшої обробки.*
2. *У програмі може бути кілька блоків catch для обробки різних типів винятків.*
3. Розкрийте призначення та особливості роботи блоку finnaly.

*Блок finally використовується для визначення коду, який завжди виконується незалежно від того, чи виникла виняткова ситуація, чи ні. Блок finally дозволяє виконати очистку ресурсів або завершити виконання деяких операцій, навіть якщо виняток був викинутий і не був оброблений.*

*Особливості роботи блоку finally:*

1. *Код в блоку finally виконується завжди, навіть якщо виняток не виник.*
2. *Блок finally використовується для ресурсоємних операцій, таких як відкриття файлів або робота з мережею.*

**Висновок**

На данній лабораторній роботі я оволодів навиками використання механізму виключень при написанні програм мовою Java. Створив клас, що реалізує метод обчислення виразу заданого варіантом. Написав на мові Java та налагодив програму-драйвер для розробленого класу. При написанні програми застосував механізм виключень для виправлення помилкових ситуацій, що можуть виникнути в процесі виконання програми.