Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра ЕОМ



**Лабораторна робота №4**

з дисципліни: «Кросплатформенні засоби програмування»

на тему: «Виключення»  
Варіант № 18

Виконала: ст. гр. КІ-305

Лесяк Х. В.

Прийняв:

Іванов Ю.С.

Львів – 2023

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

**Контрольні питання:**

1. Дайте визначення терміну «виключення».

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

2. У яких ситуаціях використання виключень є виправданим?

Відповідь: помилках введення, збоях обладнання, помилках, що пов’язані з фізичними обмеженнями комп’ютерної системи та помилках програмування

3. Яка ієрархія виключень використовується у мові Java?

Відповідь: 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 створити власний клас виключень?

Відповідь: заекстендити Error або Exception(або дочірні).

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

Відповідь: [access modifier] [return type] [name]([parameters]) throws Exception

{

…

}

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

Відповідь: 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.

7. Як згенерувати контрольоване виключення?

Відповідь:

1. throw new Exception();

2. Exception ex = new Exception(); throw ex;

8. Розкрийте призначення та особливості роботи блоку try.

Відповідь: Блок виконання коду.

9. Розкрийте призначення та особливості роботи блоку catch.

Відповідь: Блок перехоплення виключення.

10. Розкрийте призначення та особливості роботи блоку finally.

Відповідь: Блок гарантованого виконання коду після try-catch блоку.

**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.

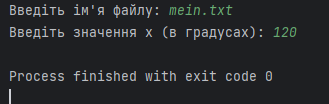
3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.

4. Дати відповідь на контрольні запитання.

**Код EquationsApp.java:**

package KI.Lesiak.Lab4;  
import java.io.File;  
import java.io.IOException;  
import java.io.PrintWriter;  
import java.util.Scanner;  
  
import static java.lang.System.*out*;  
public class EquationsApp {  
 public static void main(String[] args) {  
 try {  
 *out*.print("Введіть ім'я файлу: ");  
 Scanner in = new Scanner(System.*in*);  
 String fName = in.nextLine();  
 PrintWriter fout = new PrintWriter(new File(fName));  
  
 try {  
 try {  
 *out*.print("Введіть значення x (в градусах): ");  
 double x = in.nextDouble();  
 double result = *calculateExpression*(x);  
 fout.print("Результат обчислення для x = " + x + ": " + result);  
 }finally {  
 fout.flush();  
 fout.close();  
 }  
 } catch (IllegalArgumentException ex) {  
 *out*.print(ex.getMessage());  
 }  
 } catch (IOException ex) {  
 *out*.print("Помилка: " + ex.getMessage());  
 }  
 }  
  
 */\*\*  
 \* Метод обчислює вираз y = tg(x) / (sin(4x) - 2cos(x))  
 \*  
 \* @param x Значення x в градусах  
 \* @return Результат обчислення виразу  
 \* @throws IllegalArgumentException Якщо виникла помилка при обчисленні  
 \*/* public static double calculateExpression(double x) throws IllegalArgumentException {  
 double rad = Math.*toRadians*(x);  
  
 // Перевірка на ділення на нуль  
 double denominator = Math.*sin*(4 \* rad) - 2 \* Math.*cos*(rad);  
 double tolerance = 1e-10;  
 if (denominator < tolerance) {  
 throw new IllegalArgumentException("Ділення на нуль неможливе.");  
 }  
  
 double numerator = Math.*tan*(rad);  
 return numerator / denominator;  
 }  
  
}

**Скріншоти програми:**

****

**Зображення, що містить текст, знімок екрана, програмне забезпечення, монітор

Автоматично згенерований опис**

Рис. 1. Результат роботи програми.

**Висновок:** Я оволоділа навиками використання механізму виключень при написанні програм мовою Java.