

Міністерство освіти і науки України  
Національний університет «Львівська політехніка»

Кафедра ЕОМ



## **Лабораторна робота №5**

з дисципліни: «Кросплатформенні засоби програмування»

на тему: «Виключення»

Варіант № 14

Виконала:  
ст.гр. КІ-34  
Олексій М. В.  
Прийняв:  
Іванов Ю.С.

**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:** 14.  $y = \cos(x) / \tan(2x)$

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:

```
public class EquationsApp {
    public static void main(String[] args) {
        CalculateTheEquationInterface calc = new CalculateTheEquation();
        System.out.println(" y=cos(x)/tg(2x) = " + calc.doCalculation(100));
        System.out.println(" y=cos(x)/tg(2x) = " + calc.doCalculation(70));

        System.out.println(" y=cos(x)/tg(2x) = " +
            calc.doCalculationWithInputInside());
        System.out.println(" y=cos(x)/tg(2x) = " +
            calc.doCalculationWithInputInside());
    }
}
```

### Код CalculateTheEquationInterface.java:

```
public interface CalculateTheEquationInterface {

    double doCalculation(double variable);
}
```

```

        double doCalculationWithInputInside();
    }

```

## Код CalculateTheEquation.java:

```

import java.util.InputMismatchException;
import java.util.Scanner;

public class CalculateTheEquation implements CalculateTheEquationInterface {
    @Override
    public double doCalculation(double variable) {
        try {
            return Math.cos(variable) / Math.tan(2*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.cos(value)/ Math.tan(2*value);
        } catch (ArithmeticException e){
            System.out.println("Arithmetic exception: illegal value.");
        }
        catch (InputMismatchException e){
            System.out.println("Input exception: illegal value.");
        }
        return 0;
    }
}

```

## Скріншоти програми:

```
<terminated> EquationsApp [Java Application] C:\Program Files
  y=cos(x)/tg(2x)  = -0.48106312460229117
  y=cos(x)/tg(2x)  = -0.12780459741811173
Enter value: 45
  y=cos(x)/tg(2x)  = -0.2632928429662439
Enter value: 230
  y=cos(x)/tg(2x)  = -0.1955395163223274
```

*Рис. 1. Результат роботи програми.*

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