# ใบงานการทดลองที่ 12

# เรื่อง การใช้งานคำสั่ง try catch และ throw exception

**นางสาว เบญจภรณ**
**62543502007-2**

## 1. จุดประสงค์ทั่วไป

1.1. รู้และเข้าใจการใช้วัตถุ การทำหลายงานพร้อมกัน และการติดต่อระหว่างงาน

1.2. รู้และเข้าใจการจัดการกับความผิดปกติในการเขียนโปรแกรมเชิงวัตถุ

## 2. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

## 3. ทฤษฎีการทดลอง

3.1. Java Exception คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

**เป็นวิธีการจัดการข้อผิดพลาดที่เกิดขึ้นในขณะที่โปรแกรมทำงาน**

```
import java.util.Scanner;

public class TestException1 {

    public static void main (String[] args) {

        Scanner reader = new Scanner(System.in);

        int x;


        System.out.print("Enter number: ");

        x = reader.nextInt();

        System.out.println("Your number is " + x);

    }

}
```

นางสาว เบญจภรณ์ ไชยเสนา
62543502007-2 คอบ.คพ

3.2. คำสั่ง try มีลักษณะการทำงานอย่างไร?

```
Try {
        Jorhvsosd
        Int answer = j / s ;
}
```

3.3. คำสั่ง catch มีลักษณะการทำงานอย่างไร?

```
catch (InputMismatchException ex) {
        System.out.println("Exception occurred: " + ex);
    }
```

ตรวจจับ exception ที่จะเกิดขึ้นและจัดการกับมัน จากตัวอย่างด้านบน เราได้ปรับปรุงโปรแกรมให้สามารถจัดการกับ exception ได

3.4. คำสั่ง finally มีลักษณะการทำงานอย่างไร?

**เป็นคำสงวนใจภาษา Java ซึ่งเราสามารถใช้ได้กับการประกาศ ตัวแปร, method และ class ได้ด้วย โดยที่มันจะมีความหมายแตกต่างกันออกไปขึ้นอยู่กับว่าเราไปใช้ในการประกาศอะไร**

```
finally {
        System.out.println("bar's finally");
    }
```

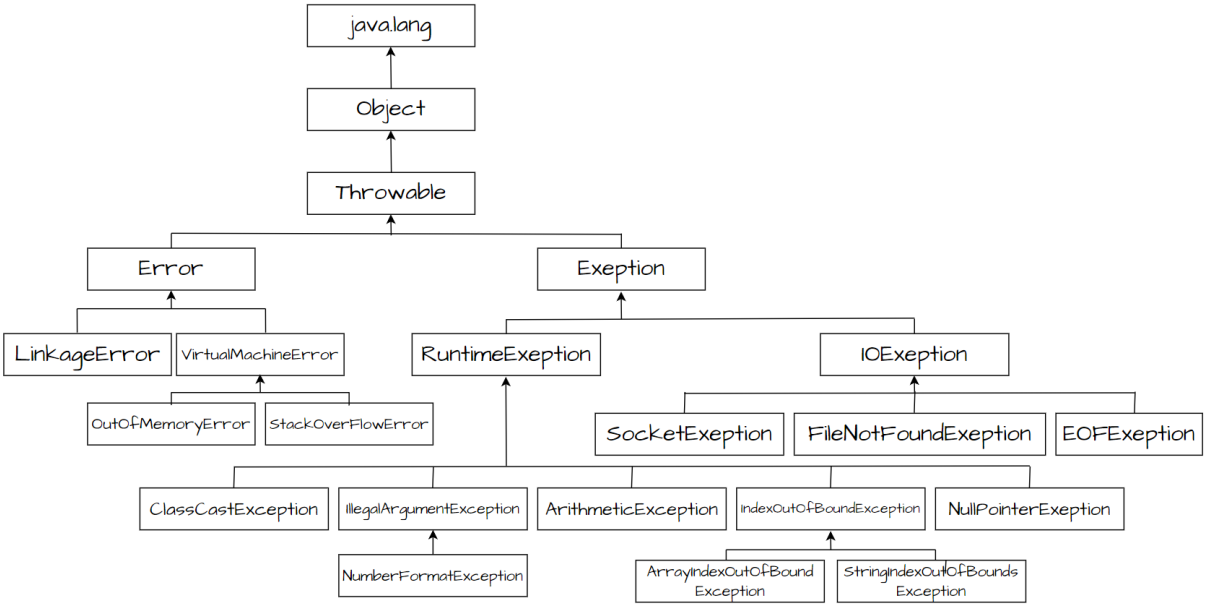
3.5. ลักษณะโครงสร้างของคำสั่ง try catch เป็นอย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ

**เป็นคำสั่งที่ใช้สำหรับกำหนดบล็อคเพื่อตรวจสอบและจัดการกับข้อผิดพลาดที่อาจจะเกิดขึ้นในโปรแกรม**

```
package com.java.myapp;
public class MyClass {
public static void main(String[] args) {
        try {
            int x = 200 ;
            int y = 0 ;
            int z = x / y;
            System.out.println(" x / y = " + z) ;
        } catch(Exception e) {
System.out.println("Error : " + e.getMessage());
        }
  }
}
```

## 4. ลำดับขั้นการปฏิบัติการ

4.1. จากผังงานต่อไปนี้ จงเขียนโค้ดโปรแกรมเพื่อแสดงตัวอย่างการจัดการความผิดปกติของคลาสการจัดการสิ่งผิดปกติจนครบทุกคลาส (เน้นเฉพาะส่วนของ Error และ Exception)




**ตัวอย่างโค้ดโปรแกรมการจัดการสิ่งผิดปกในส่วนของ Error**

```
try {
int[] array = new int[ 1000*1000*1000 ] ;

}catch( OutOfMemoryError e ) {
 System.out.println( " Array Size too large " );
}//end catch
try {
 Stack<Integer> st = new Stack<>();
}catch( StackOverflowError e ) {
 System.out.println( " Stack Overflow " );
}//end catch
```

**ตัวอย่างโค้ดโปรแกรมการจัดการสิ่งผิดปกในส่วนของ Exeption**

```
try {
 String[] strArray = new String[] { "John", "Snow" };
 ArrayList<String> strList = (ArrayList<String>) Arrays.asList(strArray);
 System.out.println("String list: " + strList);
}catch( ClassCastException e ) {
 System.out.println( " ClassCastException " );
}//end catch
try {
int a = Integer.parseInt(null);
}catch( NumberFormatException e ) {
 System.out.println( " NumberFormatException " );
}//end catch
try {
 int a = 5 / 0 ;
}catch( ArithmeticException e ) {
 System.out.println( " ArithmeticException " );
}//end catch
try {

int[] a = 5 / 0 ;

}catch( ArithmeticException e ) {

 System.out.println( " ArithmeticException " );

}//end catch
try {

int[] a = { 2 , 3 , 4 } ;
 System.out.println( a[10] );

}catch( ArrayIndexOutOfBoundsException e ) {

 System.out.println( " ArrayIndexOutOfBoundsException " );

}//end catch
try {

 String str = "Java Code Geeks!";
 CharSequence seq = str.subSequence(10, 20);
```

นางสาว เบญจภรณ์ ไชยเสนา
62543502007-2 คอบ.คพ

```
}catch( StringIndexOutOfBoundsException e ) {

 System.out.println( " StringIndexOutOfBoundsException " );

}//end catch
try {

 String a = null;
 System.out.println(a.toString());

}catch( NullPointerException e ) {

 System.out.println( " NullPointerException " );

}//end catch
try {

 client.sendMessage("hi");
 client.sendMessage("hi again");

}catch( SocketException e ) {

 System.out.println( " SocketException Connection Lost " );

}//end catch
try {

 FileReader reader = new FileReader("file.txt");

}catch( FileNotFoundException e ) {

 System.out.println( " SocketException " );

}//end catch
try {

 DataInputStream dis = new DataInputStream(new
FileInputStream("D:\\data.txt"));
while(true) {
char ch;
ch = dis.readChar() ;
 System.out.print(ch);
}
}catch( EOFException e ) {

 System.out.println( " EOFException " );

}//end catch
```

5. สรุปผลการปฏิบัติการ

การใช้ try เป็นส่วนหนึ่งที่ทำให้โปรแกรม exception ขึ้นได้ และสามารถทำ exception ให้ตรวจจับข้อมูลที่มีส่วนผิดพลาดได้

**6. คำถามท้ายการทดลอง**

6.1. เพราะเหตุใดการใช้ catch( Exception e ) ; จึงไม่เหมาะสมกับการจัดการสิ่งผิดปกติที่ดีที่สุด

**เพราะตัวโปรแกรมที่สร้างมามันอาจ ตรวจจับส่วนของ error ได้ ว่ามันจะ error ตรงไหน**

6.2. การจัดการสิ่งผิดปกติจากการตัวเลขต่างๆ ด้วยเลขศูนย์ ควรเลือกใช้วิธีใด?

**Catch( ArithmeticException e ){ }**

6.3. การจัดการสิ่งผิดปกติจากการเรียกใช้งาน Element เกินขนาดของอาเรย์ ควรเลือกใช้วิธีใด?

**Catch( ArrayindexOutOfBoundsException e ){ }**