

## ใบงานการทดลองที่ 12

### เรื่อง การใช้คำสั่ง try catch และ throw exception

#### 1. จุดประสงค์ทั่วไป

- 1.1. รู้และเข้าใจการใช้วัตถุ การทำงานหลายงานพร้อมกัน และการติดต่อระหว่างงาน
- 1.2. รู้และเข้าใจการจัดการกับความผิดปกติในการเขียนโปรแกรมเชิงวัตถุ

#### 2. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

#### 3. ทฤษฎีการทดลอง

- 3.1. Java Exception คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

เป็นวิธีการจัดการข้อผิดพลาดที่เกิดขึ้นในขณะที่โปรแกรมทำงาน

- 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 ที่เกิดขึ้นและจัดการกับมัน จากตัวอย่างด้านบน เราได้ปรับปรุงโปรแกรมให้สามารถจัดการกับ exception ได้

- 3.4. คำสั่ง finally มีลักษณะการทำงานอย่างไร?

เป็นคำสั่งงานในภาษา Java ซึ่งเราสามารถใช้ได้กับการประกาศ ตัวแปร, method และ class ได้ด้วย

โดยที่มันจะมีความหมายแตกต่างกันออกไปขึ้นอยู่กับว่าเราไปใช้ในการประกาศอะไร

```
finally {
```

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

```
}
```

- 3.5. ลักษณะโครงสร้างของคำสั่ง try catch เป็นอย่างไร? อธิบายพร้อมยกตัวอย่างประกอบ

เป็นคำสั่งที่ใช้สำหรับกำหนดบล็อกเพื่อตรวจสอบและจัดการกับข้อผิดพลาดที่อาจเกิดขึ้นในโปรแกรม

```
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());
```

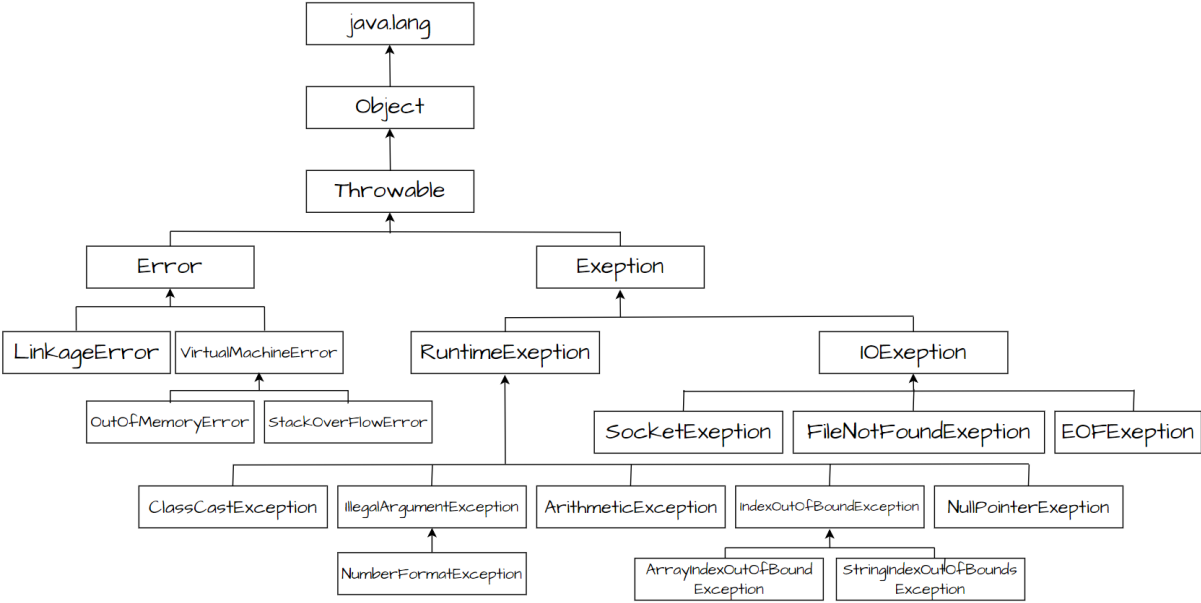
```
        }
```

```
    }
```

```
}
```

4. ลำดับขั้นการปฏิบัติการ

4.1. จากผังงานต่อไปนี้ จงเขียนโค้ดโปรแกรมเพื่อแสดงตัวอย่างการจัดการความผิดปกติของคลาสการจัดการสิ่งผิดปกติจนครบทุกคลาส (เน้นเฉพาะส่วนของ Error และ Exception)



ตัวอย่างโค้ดโปรแกรมการจัดการสิ่งผิดปกติในส่วนของ Error

ตัวอย่างโค้ดโปรแกรมการจัดการสิ่งผิดปกติในส่วนของ Exception

5. สรุปผลการปฏิบัติการ

## 6. คำถามท้ายการทดลอง

6.1. เพราะเหตุใดการใช้ `catch( Exception e )` ; จึงไม่เหมาะกับการจัดการสิ่งผิดปกติที่สุด  
เพราะตัวโปรแกรมที่สร้างมานั้นอาจ ตรวจจับส่วนของ error ได้ ว่ามันจะ error ตรงไหน

---

---

---

6.2. การจัดการสิ่งผิดปกติจากการตัวเลขต่างๆ ด้วยเลขศูนย์ ควรเลือกใช้วิธีใด?  
`Catch( ArithmeticException e ){ }`

---

---

---

6.3. การจัดการสิ่งผิดปกติจากการเรียกใช้งาน Element เกินขนาดของอาเรย์ ควรเลือกใช้วิธีใด?  
`Catch( ArrayindexOutOfBoundsException e ){ }`

---

---

---