

Programming Fundamentals II

- Lap9:
- Exception
 - Try catch finally and throws
-

1.1 Exception

Lab 9.1 ให้นักศึกษาสร้าง Package และสร้าง Class ชื่อ Lab91Exception.java นำไปใส่ใน Package ตามที่ได้สร้างไว้ เติมโค้ดให้สมบูรณ์และเขียนโปรแกรมตามตัวอย่างโค้ดด้านล่าง

```
public class Lab91Exception
{
    public static void main(String[] args)
    {
        int ans = MathDevider1.devider(10, 0);
        System.out.println(ans);
    }
}

class MathDevider1
{
    public static int devider(int num1, int num2)
    {
        int result = (int) (num1/num2);
        return result;
    }
}
```

ให้นักศึกษาเขียนผลลัพธ์ที่ได้ ลงด้านล่าง

โปรแกรมด้านบน Lab91 เกิด Exception ประเภทไหน

จงอธิบายแนวคิดในการแก้ไขปัญหาดังกล่าว

1.2 Try Catch and Finally

Lab 9.2 ให้นักศึกษาสร้าง Class ชื่อ Lab92TryCatchFinally.java นำไปใส่ใน Package ตามที่ได้สร้างไว้

Lab92TryCatchFinally.java

```
public class Lab92TryCatchFinaaly
{
    public static void main(String[] args)
    {
        int ans = MathDevider2.devider(10, 0);
        System.out.println(ans);
    }
}

class MathDevide2
{
    public static int devider(int num1, int num2)
    {
        int result = 0;
        try
        {
            result = (int) (num1/num2);
            System.out.println("Print from try");
        }
        catch(ArithmeticException e)
        {
            System.out.println(e);
            System.out.println("Print from catch");
        }
        finally
        {
            System.out.println("Print from finally");
        }
        return result;
    }
}
```

ให้นักศึกษาเขียนผลลัพธ์ที่ได้ ลงด้านล่าง

ให้นักศึกษาทดลองเอา Block finally ออก รันได้ไหม ผลลัพธ์เป็นเช่นไร

ให้นักศึกษาทดลองเอา Block catch ออก รันได้ไหม ผลลัพธ์เป็นเช่นไร

ให้นักเขียนให้กลับเหมือนเดิม และทดลองเปลี่ยนบรรทัดนี้เป็น `int ans = MathDevider.devider(10, 5);` จงเขียนผลลัพธ์

1.3 Throw

Lab 9.3 ให้นักสร้าง Class ชื่อ Lab93Throw.java นำไปใส่ใน Package ตามที่ได้สร้างไว้

Lab93Throw.java

```
public class Lab93Throw
{
    public static void main(String[] args) throws Exception
    {
        int ans = MathDevider3.devider(10, 0);
        System.out.println(ans);
    }
}

class MathDevider3
{
    public static int devider(int num1, int num2) throws Exception
    {
        int result = (int) (num1/num2);

        return result;
    }

    public static int checkZero(int num2) throws Exception
    {
        if(num2 == 0) throw new Exception("Zero Zero");
        else
            return num2;
    }
}
```

ให้นักเขียนผลลัพธ์ที่ได้ ลงด้านล่าง

ให้นักศึกษาทดลองเขียนโปรแกรมที่ดักจับ Exception จากโปรแกรมด้านบน มาใส่ในกล่องข้อความด้านล่าง

```
public class Lab93Throw
{
    public static void main(String[] args) throws Exception
    {

    }
}

class MathDevider3
{
    public static int devider(int num1, int num2) throws Exception
    {

    }

    public static int checkZero(int num2) throws Exception
    {

    }
}
```

แบบทดสอบ Lab9

ในงานนี้เราจะปรับปรุงจากงาน HW#5 เรื่อง Invoice โดยเพิ่มส่วนของการชำระเงินเข้ามาด้วย จะมี การปรับปรุงคลาส Invoice และเพิ่มคลาสอีก 4 คลาส คือ 1.Payment 2.CashPayment 3.CreditPayment 4.PlanPayment โดยมีรายละเอียดดังนี้

1. Invoice

เพิ่มคุณสมบัติดังต่อไปนี้

- payment เป็น Payment แทนการชำระเงิน ถ้ายังไม่ชำระให้มีค่าเป็น null

เพิ่มเมทอดดังต่อไปนี้

- void setPayment(Payment payment)
- Payment getPayment()
- void print() ปรับปรุงให้มีการแสดงการชำระเงินของ Invoice ด้วยถ้าได้มีการชำระแล้ว

2. Payment เป็น abstract class

ประกอบด้วยคุณสมบัติดังต่อไปนี้

- amount เป็น double แทนยอดที่ต้องชำระ

มีเมทอดดังต่อไปนี้

- Payment(double amount) ถ้า amount เป็นลบ ให้กำหนดให้เป็น 0
- double getAmount()

3. CashPayment เป็น subclass ของ Payment

ประกอบด้วยคุณสมบัติดังต่อไปนี้

- tendered เป็น double แทนยอดที่รับมา

และมีเมทอดดังต่อไปนี้

- CashPayment(double amount, double tendered)
- double getTendered()
- double getChange() คำนวณเป็นยอดเงินทอน
- String toString() ซึ่งส่งค่ากลับในรูปแบบสตริง "Amount: xx Tendered: xx Change: xx"

4. CreditPayment เป็น subclass ของ Payment

ประกอบด้วยคุณสมบัติดังต่อไปนี้

- cardNumber เป็น String แทนหมายเลขบัตรเครดิต

และมีเมทอดดังต่อไปนี้

- CreditPayment(double amount, String cardNumber)
- String getCardNumber()
- String toString() ซึ่งส่งค่ากลับในรูปแบบสตริง "Amount: xx Card Number: xx"

5. PlanPayment เป็น subclass ของ Payment

ประกอบด้วยคุณสมบัติดังต่อไปนี้

- payments เป็นอาร์เรย์ของ Payment แทนการจ่ายเงินแต่ละงวด

และมีเมทอดดังต่อไปนี้

- PlanPayment(double amount, int numberOfInstallments)
- void setInstallment(int installment, Payment payment) อนุญาตให้ payment เป็นได้เฉพาะ CashPayment หรือ CreditPayment เท่านั้น ไม่อนุญาตให้เป็น PlanPayment
- Payment getInstallment(int installment)
- String toString() ซึ่งส่งค่ากลับในรูปแบบสตริง "Amount: xx Installments: xx" และต่อด้วย Payment ของแต่ละงวดที่ได้จ่ายไปแล้ว ซึ่งสุดท้ายอาจจะออกมาในรูปแบบดังนี้

```
Amount: 10000.00 Installments: 10
Installment 0: Amount: 1000.00 Card Number: 1234567887654321
Installment 1: Amount: 1000.00 Tendered: 1000.00 Change: 0.00
Installment 2: Amount: 1000.00 Card Number: 1234567887654321
```

6. InvoiceTest

ปรับปรุงตัวทดสอบให้มีการสร้าง Invoice ออกมา 4 แบบ แบบแรกยังไม่จ่ายเงิน แบบที่สองจ่ายเป็นเงินสด แบบที่สามจ่ายด้วยบัตรเครดิต และแบบที่สี่จ่ายเป็นแบบผ่อน โดยงวดที่ผ่อนชำระต้องมีทั้งแบบที่ชำระด้วยเงินสดและด้วยบัตรเครดิต สุดท้ายให้แสดงผล Invoice ทั้ง 4 ออกมา (ซึ่งต้องแสดงรายละเอียดการชำระเงินด้วย)