

Programming Fundamentals II

- Lap10:
- Exception
 - Try catch finally and throws

Exception

Lab 10.1 ให้นิสิตสร้าง Package และสร้าง Class ชื่อ Lab101Exception.java นำไปใส่ใน Package ตามที่ได้สร้างไว้ เติมโค้ดให้สมบูรณ์และเขียนโปรแกรมตามตัวอย่างโค้ดด้านล่าง

```
public class Lab101Exception
{
    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;
    }
}
```

ให้นิสิตเขียนผลลัพธ์ที่ได้ ลงด้านล่าง

```
PS D:\coding_lab\java\6530300295\week10\first> java -classpath "..bin\" first.Lab101Exception
Exception in thread "main" java.lang.ArithmeticException: / by zero
    at first.MathDevider1.devider(Lab101Exception.java:12)
    at first.Lab101Exception.main(Lab101Exception.java:5)
```

โปรแกรมด้านบน Lab101 เกิด Exception ประเภทไหน

RuntimeException - ArithmeticException เกิด error เพราะ ตัวหารมีค่าเป็น 0

จงอธิบายแนวคิดในการแก้ไขปัญหาดังกล่าว

เขียน exception try-catch จับ error ไม่ให้โปรแกรม error

Try Catch and Finally

Lab 10.2 ให้นักศึกษาสร้าง Class ชื่อ Lab102TryCatchFinally.java นำไปใส่ใน Package ตามที่ได้สร้างไว้

Lab102TryCatchFinally.java

```
public class Lab102TryCatchFinally
{
    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

```
PS D:\coding_lab\java\6530300295\week10\first> java -classpath "..bin\" first.Lab102TryCatchFinally
java.lang.ArithmeticException: / by zero
Print from catch
Print from finally
0
```

ให้นักศึกษาทดลองเอา Block finally ออก รันได้ไหม ผลลัพธ์เป็นเช่นไร

รันได้

```
PS D:\coding_lab\java\6530300295\week10\first> java -classpath "..bin\" first.Lab102TryCatchFinally
java.lang.ArithmeticException: / by zero
Print from catch
0
```

ให้นักศึกษาทดลองเอา Block catch ออก รันได้ไหม ผลลัพธ์เป็นเช่นไร

รันไม่ได้ เพราะ try ไม่สามารถรันได้ ถ้าไม่มี catch หรือ finally อย่างใดอย่างหนึ่ง

```
PS D:\coding_lab\java\6530300295\week10\first> javac -d "..bin\" *.java
Lab102TryCatchFinally.java:13: error: 'try' without 'catch', 'finally' or resource declarations
    try{
    ^
1 error
```

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

```
PS D:\coding_lab\java\6530300295\week10\first> java -classpath "..bin\" first.Lab102TryCatchFinally
Print from try
Print from finally
2
```

Throw

Lab 10.3 ให้นักเขียนสร้าง Class ชื่อ Lab103Throw.java นำไปใส่ใน Package ตามที่ได้สร้างไว้

Lab103Throw.java

```
public class Lab103Throw
{
    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);

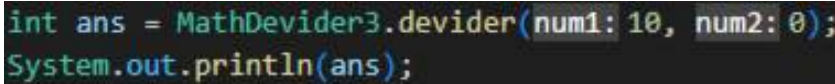
        return result;
    }

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

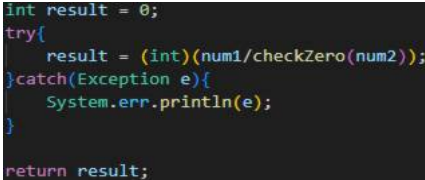
ให้นักเขียนผลลัพธ์ที่ได้ ลงด้านล่าง

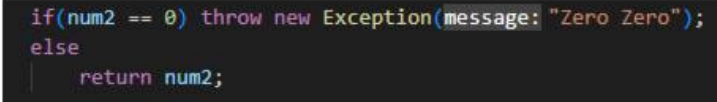
ให้นักศึกษาทดลองเขียนโปรแกรมที่ดักจับ Exception จากโปรแกรมด้านบน มาใส่ในกล่องข้อความด้านล่าง

```
public class Lab103Throw
{
    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
    {
        
    }
}
```

Homework Lab10

ในงานนี้เราจะปรับปรุงจากงาน HW#6 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() ซึ่งส่งค่ากลับในรูปสตริง "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() ซึ่งส่งค่ากลับในรูปสตริง "Amount: xx Installments: xx" และต่อด้วย Payment ของแต่ละงวดที่ได้จ่ายไปแล้ว ซึ่งสุดท้ายอาจจะออกมาในรูปดังนี้

```
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)

HomeWork 10#

```
1  /*
2   * Written by Nititorn Kijprasopchok
3   * ID: 6530300295
4   */
5  package HW6;
6  import java.util.ArrayList;
7
8  public class Invoice{
9      //instance variables
10     private String id;
11     private Customer customer;
12     private Payment payment;
13     private ArrayList<LineItem> items = new ArrayList<LineItem>();
14
15     //Invoice constructor
16     public Invoice(String id, Customer customer){
17         this.id = id;
18         this.customer = customer;
19     }
20
21     //method for adding an Items
22     public void addItem(Product product, int quantity){
23         this.items.add(new LineItem(product, quantity));
24     }
25
26     //method for get id
27     public String getId(){
28         return this.id;
29     }
30
31     //method for get customer
32     public Customer getCustomer(){
33         return this.customer;
34     }
35
36     //method for get lineitem
37     public LineItem getLineItem(int i){
38         return this.items.get(i);
39     }
40
41     //method for get total price
42     public double getTotalPrice(){
43         double sum = 0;
44         for(int i = 0; i < this.items.size(); i++){
45             sum += this.items.get(i).getTotalPrice();
46         }
47         return sum;
48     }
49
50     public void setPayment(Payment payment){
51         this.payment = payment;
52     }
53
54     public Payment getPayment(){
55         return payment;
56     }
57
58     //print method
59     public void print(){
60         try{
61             System.out.println("INVOICE: #" + getId());
62             System.out.println("CUSTOMER: " + getCustomer().getFirstName() + " " + getCustomer().getLastName());
63             System.out.println("ITEMS:");
64             for(int i = 0; i < this.items.size(); i++){
65                 System.out.println((i+1) + ". " + this.items.get(i).getProduct().getName() + " x " + this.items.get(i).getQuantity() + " = " + this.items.get(i).getTotalPrice());
66             }
67             System.out.println("TOTAL: " + getTotalPrice() + "\n");
68             System.out.print("Payment: ");
69             if(payment.getClass().isAssignableFrom(CashPayment.class)){
70                 System.out.println("Cash Payment");
71                 System.out.println(payment.toString());
72             }else if(payment.getClass().isAssignableFrom(CreditPayment.class)){
73                 System.out.println("Credit Payment");
74                 System.out.println(payment.toString());
75             }else if(payment.getClass().isAssignableFrom(PlanPayment.class)){
76                 System.out.println("Plan Payment");
77                 System.out.println(payment.toString());
78             }else{
79
80             }
81             System.out.print("=====\n");
82         }catch(NullPointerException e){
83             System.out.println("-");
84             System.out.println("=====\n");
85         }
86     }
87 }
88
```

```

1  /*
2  * Written by Nititorn Kijprasopchok
3  * ID: 6530300295
4  */
5  package HW6;
6
7  public abstract class Payment {
8      private double amount;
9
10     public Payment(double amount){
11         if(amount < 0){
12             this.amount = 0;
13         }
14         else{
15             this.amount = amount;
16         }
17     }
18
19     public double getAmount(){
20         return amount;
21     }
22 }
23

```

```

1  /*
2  * Written by Nititorn Kijprasopchok
3  * ID: 6530300295
4  */
5  package HW6;
6
7  public class CreditPayment extends Payment{
8      private String cardNumber;
9
10     public CreditPayment(double amount, String cardNumber){
11         super(amount);
12         this.cardNumber = cardNumber;
13     }
14
15     public String getCardNumber(){
16         return cardNumber;
17     }
18
19     public String toString(){
20         return String.format("Amount: %.2f Card Number: %s", super.getAmount(), getCardNumber());
21     }
22 }
23

```

```

1  /*
2  * Written by Nititorn Kijprasopchok
3  * ID: 6530300295
4  */
5  package HW6;
6
7  public class CashPayment extends Payment{
8      private double tendered;
9
10     public CashPayment(double amount, double tendered){
11         super(amount);
12         this.tendered = tendered;
13     }
14
15     public double getTendered(){
16         return tendered;
17     }
18
19     public double getChange(){
20         return getTendered() - super.getAmount();
21     }
22
23     public String toString(){
24         return String.format("Amount: %.2f Tendered: %.2f Change: %.2f", super.getAmount(), getTendered(), getChange());
25     }
26 }
27

```

```

1  /*
2  * Written by Nititorn Kijprasopchok
3  * ID: 6530300295
4  */
5  package HW6;
6
7  public class PlanPayment extends Payment{
8      private Payment[] payments;
9
10     public PlanPayment(double amount, int numberOfInstallments){
11         super(amount);
12         this.payments = new Payment[numberOfInstallments];
13     }
14
15     public void setInstallment(int installment, Payment payment){
16         try{
17             if(payment.getClass().isAssignableFrom(PlanPayment.class)){
18                 new Exception("");
19             }else{
20                 this.payments[installment] = payment;
21             }
22         }catch(Exception e){
23             System.err.println(e.toString());
24         }
25     }
26
27     public Payment getInstallment(int installment){
28         return payments[installment];
29     }
30
31     public String toString(){
32         String text = String.format("Amount: %.2f Installment: %d \n", super.getAmount(), payments.length);
33         try{
34             for(int i = 0; i < payments.length; i++){
35                 text += "Installment " + i + ": " + payments[i].toString() + "\n";
36             }
37         }catch(Exception e){
38             return text;
39         }
40
41         return text;
42     }
43 }
44

```



```

1  /*
2   * Written by Nititorn Kijprasopchok
3   * ID: 6530300295
4   */
5  package HW6;
6
7  public class InvoiceTest {
8      public static void main(String[] args){
9          //creating customer object
10         Customer customer1 = new Customer("1", "Nititorn", "Kijprasopchok");
11         Customer customer2 = new Customer("2", "John", "Kater");
12         Customer customer3 = new Customer("3", "Tom", "Lolan");
13         Customer customer4 = new Customer("4", "Jacky", "Chan");
14
15         //creating invoice object
16         Invoice invoice1 = new Invoice("01", customer1);
17         Invoice invoice2 = new Invoice("02", customer2);
18         Invoice invoice3 = new Invoice("03", customer3);
19         Invoice invoice4 = new Invoice("04", customer4);
20
21         //creating product object
22         Product product1 = new Product("001", "Table", 1000.0);
23         Product product2 = new Product("002", "Chair", 250.0);
24         Product product3 = new Product("003", "Sofa", 2000.0);
25         Product product4 = new Product("004", "Bed", 3000.0);
26         Product product5 = new Product("005", "TV", 5000.0);
27
28         //adding items for first invoice
29         invoice1.addItem(product1, 1);
30         invoice1.addItem(product2, 4);
31         invoice1.addItem(product5, 1);
32
33         //adding items for second invoice
34         invoice2.addItem(product3, 1);
35         invoice2.addItem(product4, 1);
36         invoice2.addItem(product5, 1);
37
38         //adding items for thrid invoice
39         invoice3.addItem(product1, 2);
40         invoice3.addItem(product2, 8);
41         invoice3.addItem(product3, 2);
42         invoice3.addItem(product4, 1);
43         invoice3.addItem(product5, 1);
44
45         //adding items for fourth invoice
46         invoice4.addItem(product1, 4);
47         invoice4.addItem(product2, 16);
48         invoice4.addItem(product3, 4);
49         invoice4.addItem(product4, 3);
50         invoice4.addItem(product5, 3);
51
52         //set invoice payment
53         invoice1.setPayment(null);
54         invoice2.setPayment(new CashPayment(invoice2.getTotalPrice(), 1000));
55         invoice3.setPayment(new CreditPayment(invoice3.getTotalPrice(), "6530300295"));
56         invoice4.setPayment(new PlanPayment(invoice4.getTotalPrice(), 10));
57         ((PlanPayment) invoice4.getPayment()).setInstallment(0, new CreditPayment(invoice4.getTotalPrice()/10, "6530300295"));
58         ((PlanPayment) invoice4.getPayment()).setInstallment(1, new CashPayment(invoice4.getTotalPrice()/10, invoice4.getTotalPrice()/10));
59         ((PlanPayment) invoice4.getPayment()).setInstallment(2, new CreditPayment(invoice4.getTotalPrice()/10, "6530300295"));
60
61         //print invoice from print method
62         invoice1.print();
63         invoice2.print();
64         invoice3.print();
65         invoice4.print();
66     }
67 }
68 }
69

```

```

PS D:\coding_lab\java\6530300295\week10\HW6> java -classpath "..bin\" HW6.InvoiceTest
INVOICE: #01
CUSTOMER: Nititorn Kijprasopchok
ITEMS:
1. Table x 1 = 1000.0
2. Chair x 4 = 1000.0
3. TV x 1 = 5000.0
TOTAL: 7000.0

Payment: -
=====

INVOICE: #02
CUSTOMER: John Kater
ITEMS:
1. Sofa x 1 = 2500.0
2. Bed x 1 = 3000.0
3. TV x 1 = 5000.0
TOTAL: 10500.0

Payment: Cash Payment
Amount: 10500.00 Tendered: 1000.00 Change: -9500.00
=====

INVOICE: #03
CUSTOMER: Tom Lolan
ITEMS:
1. Table x 2 = 2000.0
2. Chair x 8 = 2000.0
3. Sofa x 2 = 5000.0
4. Bed x 1 = 3000.0
5. TV x 1 = 5000.0
TOTAL: 17000.0

Payment: Credit Payment
Amount: 17000.00 Card Number: 6530300295
=====

INVOICE: #04
CUSTOMER: Jacky Chan
ITEMS:
1. Table x 4 = 4000.0
2. Chair x 16 = 4000.0
3. Sofa x 4 = 10000.0
4. Bed x 3 = 9000.0
5. TV x 3 = 15000.0
TOTAL: 42000.0

Payment: Plan Payment
Amount: 42000.00 Installment: 10
Installment 0: Amount: 4200.00 Card Number: 6530300295
Installment 1: Amount: 4200.00 Tendered: 4200.00 Change: 0.00
Installment 2: Amount: 4200.00 Card Number: 6530300295
=====

```