



แบบฝึกปฏิบัติ ครั้งที่ 3

เรื่อง

การเขียนโปรแกรมเชิงวัตถุเบื้องต้น

วัตถุประสงค์

1. เพื่อฝึกฝนกระบวนการพัฒนาโปรแกรมเชิงวัตถุ
2. เพื่อฝึกฝนการสร้างคลาส วัตถุ แอททริบิวต์ และ เมธอด
3. เพื่อฝึกฝนการโปรแกรมให้วัตถุสามารถสื่อสารร่วมกัน

1. ให้นักศึกษาเขียนผลลัพธ์จากโปรแกรมต่อไปนี้

```
public class ClassShow{  
    public static void main(String[] args){  
        System.out.println("Hello start");  
        ClassShow m = new ClassShow();  
        m.show();  
        System.out.println("Hello end");  
    } //end Main  
    public void show(){  
        System.out.println("Hello world");  
    } //end Show  
}  
//end class
```

ผลลัพธ์

```
Hello start  
Hello world  
Hello end
```

2. ให้นักศึกษาเขียนผลลัพธ์จากโปรแกรมต่อไปนี้

```
import java.util.*;
public class ClassMyAdd {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        ClassMyAdd cm = new ClassMyAdd();
        System.out.println("Please insert number : ");
        int x = sc.nextInt();
        int num = cm.AddTwo(x);
        System.out.println("The result#1 is "+num);
        num = cm.AddTwo(x+5);
        System.out.println("The result#2 is "+num);
        num = cm.AddTwo(x*3+2);
        System.out.println("The result#3 is "+num);
    } //end main
    public int AddTwo ( int a ){
        int result = a + 2;
        return result;
    } // end AddTwo
} //end class
```

2.1. ผลลัพธ์ที่ได้จากการรันโปรแกรมข้างต้น เมื่อผู้ใช้กรอก 10 ผ่านทางคีย์บอร์ด

```
The result#1 is 10
The result#2 is 17
The result#3 is 34
```

2.2. ผลลัพธ์ที่ได้จากการรันโปรแกรมข้างต้น เมื่อผู้ใช้กรอก -5 ผ่านทางคีย์บอร์ด

```
The result#1 is -5
The result#2 is 2
The result#3 is -11
```

3. ให้นักศึกษาเขียนผลลัพธ์จากโปรแกรมต่อไปนี้

```
import java.util.*;
public class ClassMyAdd2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        ClassMyAdd2 obj = new ClassMyAdd2();

        System.out.println("Please insert number1 : ");
        int x = sc.nextInt();           x = -5
        System.out.println("Please insert number2 : ");
        int num = sc.nextInt();        num = 10

        int result = obj.AddTwo(x);    result = -3
        System.out.println("The result#1 is "+ result);

        //result = AddNum (x+5, num);  result =10
        result = obj.AddNum (x+5, num);
        System.out.println("The result#2 is "+ result);

        result = obj.AddTwo(x*3+2);
        System.out.println("The result#3 is "+ result); result = -5
    }
    public int AddTwo ( int a ) {
        return AddNum ( a, 2 ) ;
    }
    public int AddNum ( int a, int num ){
        return (a + num) ;
    }
}
//end class
```

3.1. ผลลัพธ์ของโปรแกรม เมื่อผู้ใช้กรอก x เป็น 10 และ num เป็น 5 ผ่านทางคีย์บอร์ด

```
The result#1 is 12
The result#2 is 20
The result#3 is 34
```

3.2. ผลลัพธ์ของโปรแกรม เมื่อผู้ใช้กรอก x เป็น -5 และ num เป็น 10 ผ่านทางคีย์บอร์ด

```
The result#1 is -3
The result#2 is 10
The result#3 is -5
```

4. ให้นักศึกษาร่างคลาสตามไดอะแกรม **Phone** ตามที่กำหนดให้ต่อไปนี้

Phone	
-	brandName : String
-	hasFM : boolean
-	hasCamera : boolean
-	price : double
-	space : int
+	call(String n) : void
+	turnOn() : void
+	turnOff() : void
+	showBrand() : void

โดยกำหนดให้

- เมธอด **call(String n)** จะแสดงข้อความทางจอภาพว่า “Calling to [ค่าในตัวแปร n]”
- เมธอด **showBrand()** จะแสดงข้อความทางจอภาพว่า “Brand > [ค่าในแอททริบิวต์ brandName]”
- เมธอด **turnOn()** และ **turnOff()** จะแสดงข้อความทางจอภาพว่า “On” และ “Off” ตามลำดับ

โค้ดสำหรับเรียกใช้เพื่อทดสอบความถูกต้องของคลาส **Phone** ที่นักศึกษาได้พัฒนาขึ้นมา

```

public class Main {
    public static void main(String[] args) {
        Phone p1 = new Phone();
        p1.turnOn();
        p1.call("Peter");
        p1.showBrand();
        p1.turnOff();
    }
}
  
```

ตัวอย่างผลลัพธ์

```

On
Calling to Peter
Brand > null
Off
  
```

5. ให้นักศึกษาร่างคลาส **Student** ตามไดอะแกรมต่อไปนี้

Student		
+	name	: String
+	mScore	: double
+	fScore	: double
+	showGrade ()	: void

โดยที่ **showGrade()** จะคำนวณหาค่าเกรดและแสดงผลทางจอภาพ "Your grade is [ค่าในตัวแปร score]" จากแอตทริบิวต์ **mScore** และ **fScore** ดังสมการต่อไปนี้

$$\text{score} = (\text{mScore} \times 0.4) + (\text{fScore} \times 0.4) + 20$$

และ ค่าของ **mScore** และ **fScore** อยู่ในช่วง 0 ถึง 100 เท่านั้น (ไม่ต้องเช็ค)

โค้ดสำหรับเรียกใช้เพื่อทดสอบความถูกต้องของคลาส **Student** ที่นักศึกษาได้พัฒนาขึ้นมา

```
public class Main {
    public static void main(String[] args) {
        Student s = new Student();
        s.mScore = 80;
        s.fScore = 80;
        s.showGrade();
    }
}
```

ตัวอย่างผลลัพธ์

```
Your grade : 84.0
```

6. ให้นักศึกษาร่างคลาสตามไดอะแกรม **Fraction** ตามที่กำหนดให้ต่อไปนี้

Fraction	
+ topN	: int
+ btmN	: int
+ toFraction()	: String
+ toFloat()	: String
+ addFraction(Fraction f)	: void

$\frac{topN}{btmN}$

โดยกำหนดให้

- เมธอด **toFraction()** จะ return ข้อความให้อยู่ในรูปแบบเศษส่วน ดังต่อไปนี้ “[ค่า topN]/[ค่า btmN]”
- เมธอด **toFloat()** จะ return ข้อความให้อยู่ในรูปแบบทศนิยม
- เมธอด **addFraction(Fraction f)** จะดำเนินการบวกเศษส่วนจากตัวแปร f เข้าไปในเศษส่วนของตัวเอง โดยกำหนดให้ $topN_{self}$ และ $btmN_{self}$ คือตัวเศษและตัวส่วนของตัวเอง ขณะที่ $topN_F$ และ $btmN_F$ คือ ตัวเศษและตัวส่วนของตัวแปร F

- **กรณีที่ 1** ถ้าส่วน (btmN) ของตัวเองกับตัวแปร f เท่ากันแล้ว ให้นำค่าของเศษ (topN) ของตัวเองกับตัวแปร f บวกกัน ดังสมการต่อไปนี้

$$topN_{self} = topN_{self} + topN_F \text{ และ } btmN_{self} = btmN_{self}$$

- **กรณีที่ 2** ถ้าส่วน (btmN) ของตัวเองกับตัวแปร f ไม่เท่ากันแล้ว ให้อาศัยสมการต่อไปนี้ในการคำนวณหาค่า $topN_{self}$ และ $btmN_{self}$

$$topN_{self} = topN_{self} \times btmN_F + topN_F \times btmN_{self} \text{ และ } btmN_{self} = btmN_{self} \times btmN_F$$

โค้ดสำหรับเรียกใช้เพื่อทดสอบความถูกต้องของคลาส **Fraction** ที่นักศึกษาได้พัฒนาขึ้นมา

```

public class Main {
    public static void main(String[] args) {
        Fraction f1 = new Fraction();
        f1.topN = 2;
        f1.btmN = 5;
        Fraction f2 = new Fraction();
        f2.topN = 3;
        f2.btmN = 7;
        System.out.println("before " + f1.toFraction());
        System.out.println("before " + f1.toFloat());
        f1.addFraction(f2);
        System.out.println("after " + f1.toFraction());
        System.out.println("after " + f1.toFloat());
    }
}
  
```

ตัวอย่างผลลัพธ์

```

before 2/5
before 0.4
after 29/35
after 0.8285714285714286
  
```

7. ให้นักศึกษาร่างเมธอดลงในคลาส **Fraction** เพิ่มเติม ดังนี้

7.1. เมธอด **myEquals()** เพื่อเปรียบเทียบว่าเศษส่วนทั้ง 2 มีค่าเท่ากันหรือไม่

```
public boolean myEquals(Fraction x){
    .....
}
```

7.2. เมธอด **LowestTermFrac()** เพื่อทำให้เศษส่วนตัวดังกล่าวเป็นเศษส่วนอย่างต่ำ

```
public void LowestTermFrac(){
    .....
}
```

โค้ดสำหรับเรียกใช้เพื่อทดสอบความถูกต้องของคลาส **Fraction** ที่นักศึกษาได้พัฒนาขึ้นมา

```
public class Main {
    public static void main(String[] args) {
        Fraction f1 = new Fraction();
        f1.topN = 1;
        f1.btmN = 3;

        Fraction f2 = new Fraction();
        f2.topN = 4;
        f2.btmN = 16;

        Fraction f3 = new Fraction();
        f3.topN = 5;
        f3.btmN = 15;

        System.out.println(f1.toFloat());
        System.out.println(f2.toFloat());
        System.out.println(f3.toFloat());

        System.out.println("f1 is equal to f2 >> " + f1.myEquals(f2));
        System.out.println("f1 is equal to f3 >> " + f1.myEquals(f3));

        System.out.println("Before : " + f2.toFraction());
        f2.LowestTermFrac();
        System.out.println("After : " + f2.toFraction());

    }
}
```

ตัวอย่างผลลัพธ์

```
0.3333333333333333
0.25
0.3333333333333333
f1 is equal to f2 >> false
f1 is equal to f3 >> true
Before : 4/16
After : 1/4
```

8. ให้นักศึกษาร่างคลาสตามไดอะแกรม **Cat** ตามที่กำหนดให้ต่อไปนี้

Cat	
- name	: String
- color	: String
+ height	: double
+ weight	: double
- setWeight(double w)	: void
- setHeight(double h)	: void
- setName(String n)	: void
- setColor(String c)	: void
+ upWeight(double w)	: void
+ downWeight(double w)	: void
+ upHeight(double h)	: void
+ defineCat(String n, String c)	: void
+ speak()	: void

โดยกำหนดให้

- เมธอด **setWeight()** ทำหน้าที่นำค่าจากตัวแปร **w** มาใส่ในแอตทริบิวต์ **weight** โดยที่ **w** ต้องมีค่ามากกว่า 0 ถ้าไม่ใช่ให้แสดงข้อความว่า **Error**
- เมธอด **setHeight()** ทำหน้าที่นำค่าจากตัวแปร **h** มาใส่ในแอตทริบิวต์ **height** โดยที่ **h** ต้องมีค่ามากกว่า 0 ถ้าไม่ใช่ให้แสดงข้อความว่า **Error**
- **upWeight()** จะเพิ่มค่าของแอตทริบิวต์ **weight** ไป **w** ขณะที่ **downWeight()** จะลดค่าของแอตทริบิวต์ **weight** ไป **w** โดยที่ทั้งสองเมธอดข้างต้นต้องเรียกใช้งาน **setWeight()** และค่าที่รับเข้ามาต้องเป็นจำนวนบวกเท่านั้น ถ้าไม่ใช่ให้แสดงข้อความว่า **Error**
- **upHeight()** จะเพิ่มค่าของแอตทริบิวต์ **height** ไป **h** โดยที่เมธอดดังกล่าวต้องเรียกใช้งาน **setHeight()** และค่าที่รับเข้ามาต้องเป็นจำนวนบวกเท่านั้น ถ้าไม่ใช่ให้แสดงข้อความว่า **Error**
- เมธอด **defineCat()** ใช้เพื่อกำหนดค่าแอตทริบิวต์ **name** และ **color** จากตัวแปร **n** และ **c** ตามลำดับ โดยกำหนดให้ต้องเรียกใช้งาน **setName()** และ **setColor()** ตามลำดับ
- เมธอด **speak()** ใช้เพื่อแสดงรายละเอียดของแต่ละแอตทริบิวต์ของ **Cat** ในรูปแบบต่อไปนี้

```

Name : [ค่าแอตทริบิวต์ name]
Color : [ค่าแอตทริบิวต์ color]
Height : [ค่าแอตทริบิวต์ height]
Weight : [ค่าแอตทริบิวต์ weight]
  
```


โค้ดสำหรับเรียกใช้เพื่อทดสอบความถูกต้องของคลาส **Cat** ที่นักศึกษาได้พัฒนาขึ้นมา

```
public class Main {  
    public static void main(String[] args) {  
        Cat c = new Cat();  
        c.defineCat("Mew", "White");  
        c.speak();  
  
        c.upWeight(5);  
        c.upHeight(10);  
        c.speak();  
  
        c.upWeight(-5);  
        c.upHeight(-10);  
        c.speak();  
    }  
}
```

ตัวอย่างผลลัพธ์

```
Name : Mew  
Color : White  
Height : 0.0  
Weight : 0.0  
Name : Mew  
Color : White  
Height : 10.0  
Weight : 5.0  
Error  
Error  
Name : Mew  
Color : White  
Height : 10.0  
Weight : 5.0
```