

แบบฝึกปฏิบัติ ครั้งที่ 4

เรื่อง

การเขียนโปรแกรมเชิงวัตถุเบื้องต้น

วัตถุประสงค์

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

1. ให้นักศึกษาเขียนผลลัพธ์จากโปรแกรมต่อไปนี้

```
public class Show{
    public void show(){
        System.out.println("Hello world");
    }
}

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

ผลลัพธ์

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

2. ให้นักศึกษาเขียนผลลัพธ์จากโปรแกรมต่อไปนี้

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

        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);
    }
}

public class MyAdd {
    public int AddTwo ( int a ){
        int result = a + 2;
        return result;
    }
}
```

2.1. ผลลัพธ์ที่ได้จากการรันโปรแกรมข้างต้น เมื่อผู้ใช้กรอก 10 ผ่านทางคีย์บอร์ด

Please insert number : 10
The result#1 is 12
The result#2 is 17
The result#3 is 34

2.2. ผลลัพธ์ที่ได้จากการรันโปรแกรมข้างต้น เมื่อผู้ใช้กรอก -5 ผ่านทางคีย์บอร์ด

Please insert number : -5
The result#1 is -3
The result#2 is 2
The result#3 is -11

3. ให้นักศึกษาเขียนผลลัพธ์จากโปรแกรมต่อไปนี้

```

import java.util.*;
public class Main {
    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();
        System.out.println("Please insert number2 : ");
        int num = sc.nextInt();

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

        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);
    }

    public class MyAdd2 {
        public int AddTwo ( int a ) {
            return AddNum ( a, 2 ) ;
        }
        public int AddNum ( int a, int num ){
            return (a + num) ;
        }
    }
}

```

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

```

Please insert number1 : 10
Please insert number2 : 5
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

```

Please insert number1 : -5
Please insert number2 : 10
The result#1 is -3
The result#2 is 10
The result#3 is -11

```

4. ให้นักศึกษาร่างคลาส **Student** ตามไดอะแกรมต่อไปนี้

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

โดยที่ **showGrade()** จะคำนวณหาค่าเกรดและแสดงผลทางจอภาพ "Your grade is [ค่าในตัวแปร **score**]" จากแอททริบิวต์ **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
```

5. 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
  
```

6. ให้นักศึกษาร่างเมธอดลงในคลาส **Fraction** เพิ่มเติม ดังนี้

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

```
public boolean myEquals(Fraction x){
    return topN*1.0/btmN == x.topN*1.0/x.btmN
}
```

6.2. เมธอด **LowestTermFrac()** เพื่อให้เศษส่วนตัวดังกล่าวเป็นเศษส่วนอย่างต่ำ

```
public void LowestTermFrac(){
    for (int num = topN; num > 0; num--) {
        if (btmN % num == 0) {
            topN, btmN = topN / num, btmN / num;
        }
    }
}
```

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
```

7. ให้นักศึกษาสร้างคลาสตามไต่อแกรม **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]
  
```

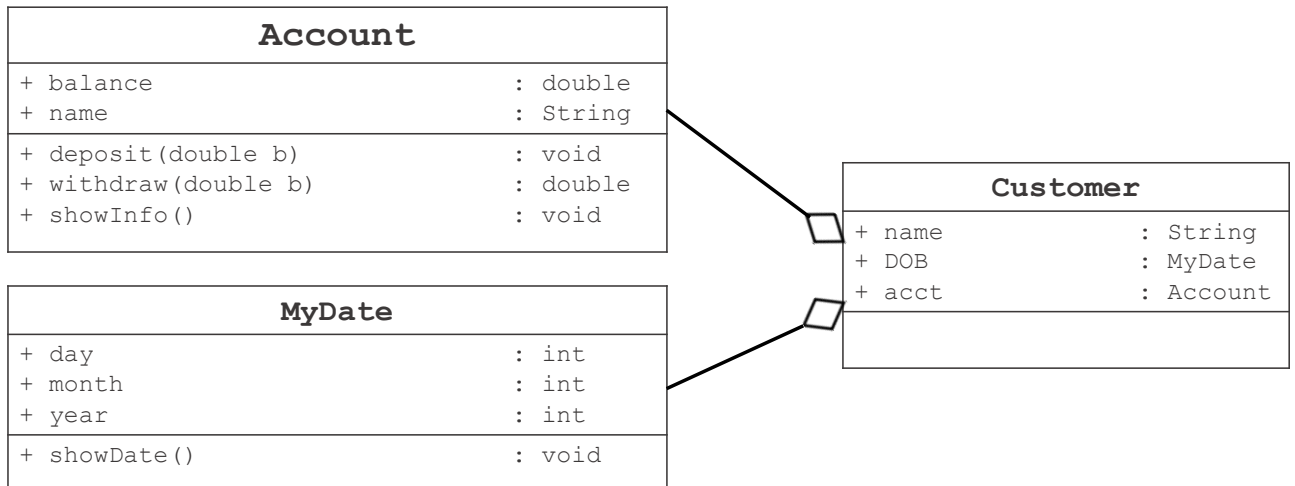
โค้ดสำหรับเรียกใช้เพื่อทดสอบความถูกต้องของคลาส **Cat** ที่นักศึกษาได้พัฒนาขึ้นมา

```
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
```


8. ให้นักศึกษาร่างคลาสตามไดอะแกรมต่อไปนี้



โดยกำหนดให้แต่ละคลาสมีรายละเอียดดังต่อไปนี้

คลาส Account

- เมธอด **deposit(...)** ทำหน้าที่เพิ่มค่าแอททริบิวต์ **balance** ตามค่าในตัวแปร **balance** ที่รับเข้ามา เมื่อตัวแปร **balance** มีค่ามากกว่าหรือเท่ากับ 0 แต่ถ้าไม่ใช่จะแสดงข้อความต่อไปนี้

The balance variable must be greater than or equal to zero.

- เมธอด **withdraw(...)** ทำหน้าที่ลดค่าแอททริบิวต์ **balance** ตามค่าในตัวแปร **balance** ที่รับเข้ามา จากนั้นจะดำเนินการคืนค่า (return) เงินที่ถอนออกมา เมื่อตัวแปร **balance** มีค่ามากกว่าหรือเท่ากับ 0 แต่ถ้าไม่ใช่จะคืนค่าเป็น 0 พร้อมแสดงข้อความต่อไปนี้

The balance variable must be greater than or equal to zero.

นอกจากนี้ ค่าแอททริบิวต์ **balance** ต้องไม่ติดลบ แต่ถ้าไม่ใช่ค่าแอททริบิวต์ **balance** จะไม่ถูกลดพร้อมคืนค่าเป็น 0 และแสดงข้อความต่อไปนี้

Your account balance is insufficient.

- เมธอด **showInfo(...)** ทำหน้าที่แสดงข้อความต่อไปนี้

In [ค่าแอททริบิวต์ **name**] account, there is a balance equal to [ค่าแอททริบิวต์ **balance**] baht.

คลาส MyDate

- เมธอด **showDate(...)** ทำหน้าที่แสดงข้อความต่อไปนี้

[ค่าแอททริบิวต์ **day**]/[ค่าแอททริบิวต์ **month**]/[ค่าแอททริบิวต์ **year**].

โค้ดสำหรับเรียกใช้เพื่อทดสอบความถูกต้องของคลาสที่นักศึกษาได้พัฒนาขึ้นมา

```
public class Main {  
    public static void main(String[] args) {  
        Customer c = new Customer();  
        c.name = "Taravichet";  
  
        MyDate dob = new MyDate();  
        dob.day = 8;  
        dob.month = 11;  
        dob.year = 2023;  
  
        Account acct = new Account();  
        acct.name = c.name;  
        acct.balance = 500;  
  
        c.DOB = dob;  
        c.acct = acct;  
  
        System.out.println("My name is " + c.name + ".");  
  
        c.acct.showInfo();  
        c.DOB.showDate();  
  
        c.acct.deposit(500);  
        c.acct.showInfo();  
        c.acct.withdraw(3000);  
        c.acct.showInfo();  
    }  
}
```

ตัวอย่างผลลัพธ์

```
My name is Taravichet.  
In Taravichet account, there is a balance equal to 500.0 baht.  
8/11/2023  
In Taravichet account, there is a balance equal to 1000.0 baht.  
Your account balance is insufficient.  
In Taravichet account, there is a balance equal to 1000.0 baht.
```

ให้นักศึกษาอธิบายความหมายของคำสั่งต่อไปนี้

```
c.DOB.showDate();  
  
c.acct.deposit(500);  
c.acct.showInfo();  
c.acct.withdraw(3000);  
c.acct.showInfo();
```