

แบบฝึกหัดปฏิบัติการคาบที่ 2: Class and Object

1. ให้ศึกษาและทดลองพิมพ์ Code ตัวอย่างการสร้างคลาสจากตัวอย่างต่อไปนี้

ต้องการสร้างคลาสของเลขเชิงซ้อนที่ประกอบไปด้วยส่วนจริง(r) และส่วนจินตภาพ (i) เช่น $3+5i$

- โดยตัวเลขเชิงซ้อน ตัวจะถือว่าเป็น 1 Object
- การทำงานของตัวเลขเชิงซ้อนที่เราจะสามารถเรียกเพื่อให้งานทำได้คือ
 - การบวก (add) โดยการส่งตัวเลขเชิงซ้อนอีก ตัวเข้ามาบวก 1
 - และการแสดงผลตัวเลข (print) โดยนำส่วนจริงและส่วนจินตภาพมาแสดงผลในรูปแบบ $r+ci$

เมื่อต้องการสร้างตัวเลขเชิงซ้อนขึ้นมา 1 ตัวจะต้องระบุแบบของตัวเลขที่จะสร้าง(Type/class) และตามด้วยตัวแปรอ้างอิงวัตถุ (ตัวแปร) ที่จะเก็บที่อยู่ของวัตถุจริง ๆ และตามด้วยการสร้างวัตถุในหน่วยความจำที่ประกอบด้วยแอตทริบิวต์และเมธอดตามที่กำหนดไว้ในคลาส โดยการใช้คำสั่ง new และตามด้วย Constructor ที่ทำหน้าที่กำหนดค่าเริ่มต้นให้วัตถุ เช่น

`Complex a = new Complex(1.0, 2.0);`

เมื่อสร้างวัตถุด้วยคำสั่งดังกล่าวจะมีรูปแบบการเก็บข้อมูลในหน่วยความจำคือ

```
// ComplexTest.java
class Complex {
    private double r, i;
    Complex(double r, double i) {
        this.r = r; this.i = i;
    }
    Complex(Complex c) {
        this(c.r, c.i);
    }
    public void add(Complex c) {
        r += c.r;
        i += c.i;
    }
    public void print() {
        System.out.println(r + "+ i" + i);
    }
}
class ComplexTest {
    public static void main(String args[]) {
        Complex a = new Complex(1.0, 2.0);
        Complex b = new Complex(3.0, 4.0);
        Complex c = new Complex(a);
        c.add(b);
        c.print();
    }
}
```

จากโปรแกรมดังกล่าวให้เพิ่มความสามารถของ Object ให้สามารถลบ คูณ และหารได้

2. จงหาข้อผิดพลาดของส่วนของโปรแกรมต่อไปนี้ โดยวงกลมตำแหน่งที่ผิด เขียนหมายเลขกำกับแต่ละตำแหน่งและอธิบายเหตุผลด้านล่างว่าเหตุใดตำแหน่งดังกล่าวจึงผิด ถ้าไม่มีข้อผิดพลาดให้ตอบว่าไม่มีข้อผิดพลาด

ส่วนของโปรแกรม	ผลลัพธ์
<pre>public class ShowErrors{ public static void main(String[] args){ ShowErrors t= new ShowErrors(5); } }</pre>	1. ไม่ใช้สร้าง Constructor ให้ใช้ให้รับ parameter ให้เขียน
<pre>public class ShowErrors{ public static void main(String[] args){ ShowErrors t= new ShowErrors(); t.x(); } }</pre>	1. ไม่ใช้ function ชื่อ x ใน class ShowErrors
<pre>public class ShowErrors{ public void method1(){ Circle c; System.out.println("What is radius "+ c.getRadius()); c=new Circle; } }</pre>	1. ไม่ใช้สร้าง class Circle ไว้ประกาศไว้ก่อน 2. Circle c ยังไม่สร้างจึงจะเรียกใช้ Method คำนวณไม่ได้ 3. new () ควร Circle
<pre>public class ShowErrors{ public static void main(String[] args){ C c = new C(5.0); System.out.println(c.value); } } class C{ int value=2; }</pre>	1. ไม่ใช้สร้าง Constructor ให้ใช้ให้รับ parameter ให้เขียน 2. เรียก C.value ไม่ได้อีกเพราะ ประกาศ C ยังไม่ได้

3. จงอธิบายการทำงานของโปรแกรมต่อไปนี้

<pre>public class Test{ public static void main(String[] args){ Count myCount = new Count(); int times=0; for (int i=0; i<100 ;i++) increment(myCount, times); System.out.println("count is "+myCount.count); System.out.println("times is "+times); } public static void increment(Count c, int times){ c.count++; times++; } } public class Count{ public int count; public Count(int c){ count =c; } public Count(){ count =1; } }</pre>	
แสดงผลลัพธ์จากโปรแกรม count is 101 times is 0	

3.1 ข้อแตกต่างของการส่งค่าพารามิเตอร์ของ Primitive type และการส่งค่าพารามิเตอร์ของ reference type จากโปรแกรมต่อไปนี้ Primitive type จะเก็บค่าของ value เปรียบกับ reference type จะอ้างถึง object ที่เก็บค่าของ value ไว้

3.2 ระบุจุดที่มีการส่งค่าพารามิเตอร์ของ Primitive type และการส่งค่าพารามิเตอร์ของ reference type

- จุดที่ 1 คือแบบ reference type คือ Count c
- จุดที่ 2 คือ Primitive type คือ int times

4. จงแสดง output จากโปรแกรมต่อไปนี้พร้อมอธิบายการทำงานในแต่ละบรรทัด

ส่วนของโปรแกรม	ผลลัพธ์
<pre> public class Test { public static void main(String[] args) { int[] a = {1, 2}; <i>// array type int ชื่อ a มี 2 ตัวคือ 1, 2</i> swap(a[0], a[1]); <i>// a[0] คือ 1 และ a[1] คือ 2 ใช้ Function swap</i> System.out.println("a[0] = " + a[0] + " a[1] = " + a[1]); <i>// a[0] และ a[1] แต่ไม่ได้มีการเปลี่ยนค่าของ array</i> } public static void swap(int n1, int n2) { int temp = n1; <i>// ค่า n1 คือ 1 เก็บไว้ใน temp</i> n1 = n2; <i>// ค่า n2 แทนที่ n1 ตอนนี้ n1 มีค่าเป็น 2</i> n2 = temp; <i>// ค่า temp แทนที่ n2 ตอนนี้ n2 มีค่าเป็น 1</i> } } </pre>	<p>a[0] = 2 a[1] = 1</p>
<pre> public class Test { public static void main(String[] args) { T t1 = new T(); <i>// สร้าง t1 type class T</i> T t2 = new T(); <i>// สร้าง t2 type class T</i> System.out.println("t1's i = " + t1.i + " and j = " + t1.j); System.out.println("t2's i = " + t2.i + " and j = " + t2.j); } class T { static int i = 0; <i>// i เป็น static จะเก็บค่าไว้ที่เดียวใน class</i> int j = 0; <i>// j เป็น instance จะเก็บค่าไว้ที่เดียวใน object</i> T() { <i>// constructor Default</i> i++; <i>// i จะเพิ่ม 1 ในการทำงานที่ 1 ครั้ง ตอนนี้ i = 1</i> j = 1; <i>// j = 1 ตอนสร้าง object</i> } } } </pre>	<p>t1's i = 1 and j = 1 t2's i = 2 and j = 1</p>

(ฝึกเขียนโปรแกรมตาม Concept OOP)

5. [Algebra: solve 2x2 equation] ให้สร้างคลาส **LinearEquation** สำหรับการแก้สมการระบบสมการเชิงเส้น ตัวแปรได้ตามสมการต่อไปนี้

$$\begin{matrix} ax + by = e \\ cx + dy = f \end{matrix} \quad x = \frac{ed - bf}{ad - bc} \quad y = \frac{af - ec}{ad - bc}$$

โดยภายในคลาสประกอบด้วยส่วนประกอบต่อไปนี้

- ตัวแปรประเภท Private a, b, c, d, e, และ f.
- Constructor ที่มีการรับค่า arguments สำหรับ a, b, c, d, e, and f.
- get methods สำหรับตัวแปร a, b, c, d, e, and f.
- method ชื่อ isSolvable() ที่คืนค่า true ถ้า $ad - bc$ ไม่ใช่ 0.
- method ชื่อ getX() and getY() ที่คืนค่าผลลัพธ์ของสมการ

หลังจากนั้นให้เขียนโปรแกรมเพื่อทดสอบโดย รับค่า a b c d e f จากผู้ใช้หลังจากนั้นคำนวณผลลัพธ์ตัวอย่าง

Enter a b c d e f: 9.0 4.0 3.0 -5.0 -6.0 -21

x is -2.0 and y is 3.0

ตัวอย่าง:

Enter a b c d e f: -1.0 2.0 2.0 4.0 4.0 5.0

The equation has no solution

HW

6. (APPLICATION) จงเขียนโปรแกรมเพื่อแสดงรายละเอียดของคลาส Account ที่ประกอบด้วยสมาชิกต่อไปนี้

- ตัวแปร private ชนิดข้อมูล int ชื่อ id สำหรับเก็บหมายเลขบัญชี
- ตัวแปร private ชนิดข้อมูล double ชื่อ balance สำหรับเก็บยอดเงินคงเหลือ
- ตัวแปร private ชนิดข้อมูล double ชื่อ annualInterestRate สำหรับเก็บอัตราดอกเบี้ย
- ตัวแปร private ชนิดข้อมูล Date ชื่อ dateCreated สำหรับเก็บวันที่ที่บัญชีถูกสร้าง
- constructor ที่ไม่มี argument สำหรับการสร้างบัญชีแบบ default
- constructor ที่มี argument สำหรับการสร้างบัญชีแบบระบุเลขที่บัญชี และยอดเงินเริ่มต้น
- accessor method(get) และ mutator method(set) สำหรับตัวแปร id, balance, annualInterestRate, dateCreated
- เมธอด getMonthlyInterestRate() ที่คืนอัตราดอกเบี้ยรายเดือน
- เมธอด getMonthlyInterest() ที่คืนดอกเบี้ยรายเดือน
- เมธอด withdraw() ที่ถอนเงินตามจำนวนที่ระบุ
- เมธอด deposit() ที่ฝากเงินตามจำนวนที่ระบุ

วาดคลาสไดอะแกรมและเขียนส่วนของ Client สำหรับเรียกใช้คลาส Account โดยสร้างอ็อบเจกต์ของบัญชีเลขที่ (ID) 1122 ยอดเงินเปิดบัญชีคือ 20000 และอัตราดอกเบี้ยคือ 4.5 % หลังจากนั้นให้ใช้ withdraw method สำหรับถอนเงิน 2500 บาท และใช้ deposit method สำหรับฝากเงิน 3000 บาท และโปรแกรมสามารถแสดงยอดเงินคงเหลือและอัตราดอกเบี้ยรายเดือนได้

HW

7. เขียนโปรแกรมที่รับค่าจากผู้ใช้ที่ประกอบด้วยพิกัดของจุดศูนย์กลาง x, y ความกว้าง ความสูงของสี่เหลี่ยมและเช็คว่สี่เหลี่ยมรูปที่สองอยู่ในสี่เหลี่ยมรูปแรก หรือซ้อนทับกับรูปแรก ดังแสดงในภาพ

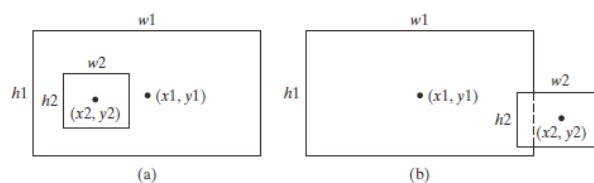


FIGURE 3.11 (a) A rectangle is inside another one. (b) A rectangle overlaps another one.

Here are the sample runs:

```
Enter r1's center x-, y-coordinates, width, and height: 2.5 4 2.5 43
Enter r2's center x-, y-coordinates, width, and height: 1.5 5 0.5 3
r2 is inside r1
```

8.

(Geometry: *n*-sided regular polygon) In an *n*-sided regular polygon, all sides have the same length and all angles have the same degree (i.e., the polygon is both equilateral and equiangular). Design a class named **RegularPolygon** that contains:

- A private **int** data field named **n** that defines the number of sides in the polygon with default value **3**.
- A private **double** data field named **side** that stores the length of the side with default value **1**.
- A private **double** data field named **x** that defines the *x*-coordinate of the polygon's center with default value **0**.
- A private **double** data field named **y** that defines the *y*-coordinate of the polygon's center with default value **0**.
- A no-arg constructor that creates a regular polygon with default values.
- A constructor that creates a regular polygon with the specified number of sides and length of side, centered at **(0, 0)**.
- A constructor that creates a regular polygon with the specified number of sides, length of side, and *x*-and *y*-coordinates.
- The accessor and mutator methods for all data fields.
- The method **getPerimeter()** that returns the perimeter of the polygon.

■ The method **getArea()** that returns the area of the polygon. The formula for computing the area of a regular polygon is $Area = \frac{n \times s^2}{4 \times \tan\left(\frac{\pi}{n}\right)}$.

Draw the UML diagram for the class and then implement the class. Write a test program that creates three **RegularPolygon** objects, created using the no-arg constructor, using **RegularPolygon(6, 4)**, and using **RegularPolygon(10, 4, 5.6, 7.8)**. For each object, display its perimeter and area.

9. [Maximum Prime Number] จงเขียนโปรแกรมแบบ OOP โดยมีคลาส เมธอด และแอทริบิวต์ ที่อ่านตัวเลขจำนวนจริง หาตัวเลขที่เป็นเลขจำนวนเฉพาะและมีความมากที่สุดจากกลุ่มของตัวเลขดังกล่าว โดยให้ตัวเลข Input จบด้วยเลข 0 ตัวอย่างเช่น ถ้าป้อนตัวเลขต่อไปนี้ 3 5 2 5 5 0 โปรแกรมจะหาตัวเลขที่เป็นเลขจำนวนเฉพาะและมีความมากที่สุดคือ 5 ถ้าหากไม่มีข้อมูลที่เป็นจำนวนเฉพาะในข้อมูลที่ให้มาเลยให้แสดงค่า 1-เช่น

ข้อมูลนำเข้า รับข้อมูลเลขจำนวนเต็มบวก n ตัว

ข้อมูลส่งออก ตัวเลขที่มีความมากที่สุดและจำนวนครั้งของการปรากฏ

ข้อมูลนำเข้า	ข้อมูลส่งออก
3 5 2 5 5 0	5
0 111 59 53 24 9 8 4 2 103	103
4 8 6 10 12 14 0	-1