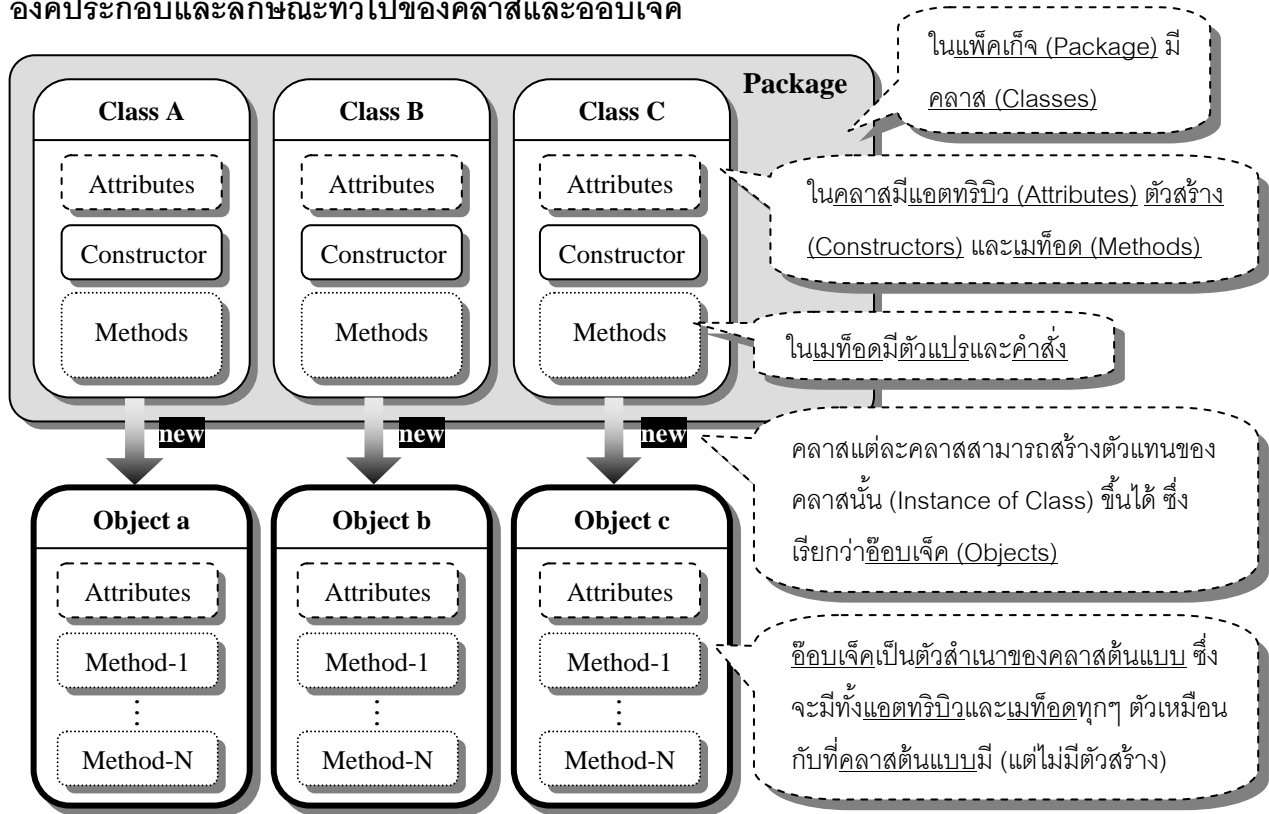


CHAPTER 12

คลาสและอ็อบเจ็กต์ (Classes and Objects)

1. ความรู้เบื้องต้นของคลาสและอ็อบเจ็กต์ (Introduction to Classes and Objects)

1. องค์ประกอบและลักษณะทั่วไปของคลาสและอ็อบเจ็กต์



2. นิยามและคำศัพท์ที่เกี่ยวข้องกับคลาสและอ็อบเจ็กต์

แพ็คเกจ (Package)	คือ สิ่งที่เกี่ยวข้องรวมคลาสและห่อหุ้มคลาสเอาไว้เป็นกลุ่มๆ (จะยังไม่กล่าวถึงมากนัก)
คลาส (Class)	คือ ต้นแบบของโปรแกรมหนึ่งๆ ซึ่งภายในประกอบไปด้วย แอตทริบิวต์ ตัวสร้าง และเมทอด
แอตทริบิวต์ (Attribute)	คือ ตัวแปรของคลาส หรือตัวแปรที่ประกาศอยู่ภายในคลาสแต่อยู่นอกเมทอดใดๆ
ตัวสร้าง (Constructor)	คือ ส่วนที่กำหนดค่าเริ่มต้นของคลาส ซึ่งจะกำหนดค่าเริ่มต้นให้กับแอตทริบิวต์ทุกตัวในคลาส
เมทอด (Method)	คือ ส่วนการทำงานย่อยของคลาส หรือโปรแกรมย่อยที่บ่งบอกการกระทำของคลาส
ตัวแปร (Variable)	คือ ตัวแปรของเมทอด หรือตัวแปรใดๆ ที่ประกาศภายในเมทอดใดเมทอดหนึ่ง
คำสั่ง (Statement)	คือ คำสั่งภายในเมทอด หรือคำสั่งใดๆ (if, for, ...) ที่ปรากฏภายในเมทอดใดเมทอดหนึ่ง
อ็อบเจ็กต์ (Object)	คือ ตัวแทนของคลาส ซึ่งจะมีทุกอย่างเหมือนกับคลาสดั้งเดิม (ยกเว้นตัวสร้างที่ไม่มี)

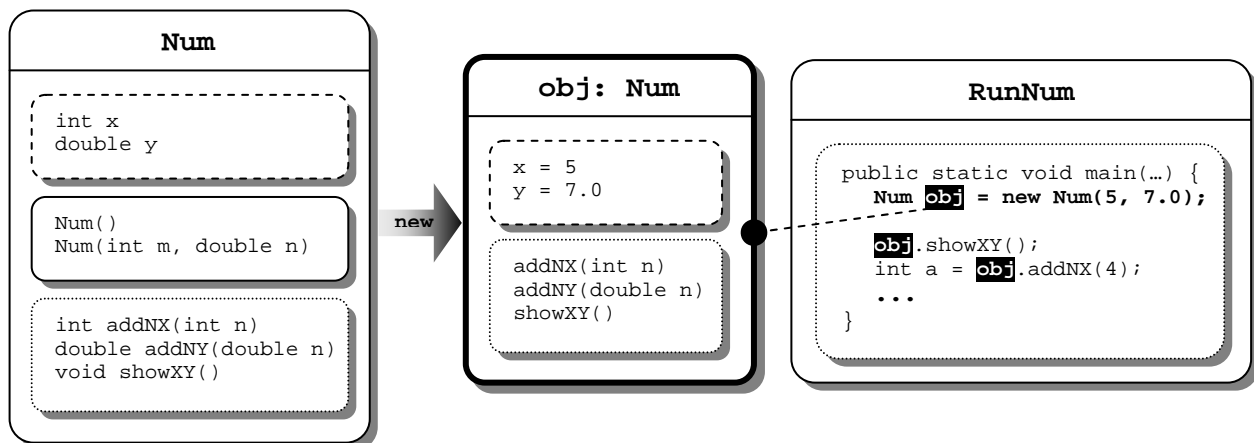
นำเสนอมุมมองในการเขียนโปรแกรมแบบใหม่ที่เรียกว่า การเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming)

โจทย์ข้อที่ 1 [ระดับง่าย] จงระบุส่วนประกอบต่างๆ ของคลาสจากโปรแกรมต่อไปนี้ (10 คะแนน)

1	public class Num {		
2	private int x;		
3	private double y;		
4	public Num() {		
5	x = 0; y = 0.0;		
6	}		
7	public Num(int m, double n) {		
8	x = m; y = n;		
9	}		
10	public int addNX(int n) {		
11	return n + x;		
12	}		
13	public double addNY(double n) {		
14	return n + y;		
15	}		
16	public void showXY() {		
17	System.out.println(x);		
18	System.out.println(y);		
19	}		
20	}		

1	public class RunNum {		
2	public static void main(String[] args){		
3	Num obj = new Num(5, 7.0);		
4	obj.showXY();		
5	int a = obj.addNX(4);		
6	double b = obj.addNY(13.0);		
7	System.out.println(a);		
8	System.out.println(b);		
9	}		
10	}		

จากโจทย์ข้อที่ 1 สามารถวาดเป็น Class Diagram เพื่อให้ประกอบความเข้าใจได้ดังนี้



โจทย์ข้อที่ 2 [ระดับง่าย] จงพิจารณาข้อความต่อไปนี้ถูก (✓) หรือผิด (✗) (15 คะแนน)

- ☐ คลาสทุกคลาสต้องมีเมทอดอย่างน้อย 1 เมทอดเสมอ (ไม่มีเมทอดไม่ได้)
- ☐ คลาสทุกคลาสต้องมีเมทอด main(...) เสมอ เพื่อเป็นตัวประสานเมทอดอื่นๆ
- ☐ แอดทริบิวของคลาสจะมีหรือไม่มีก็ได้ ซึ่งขึ้นอยู่กับกรอบแบบและการใช้งานคลาสนั้น
- ☐ คลาสสามารถมีแอดทริบิวเพียงส่วนเดียวได้โดยไม่ต้องมีเมทอด (ประกาศเพียงแอดทริบิว)

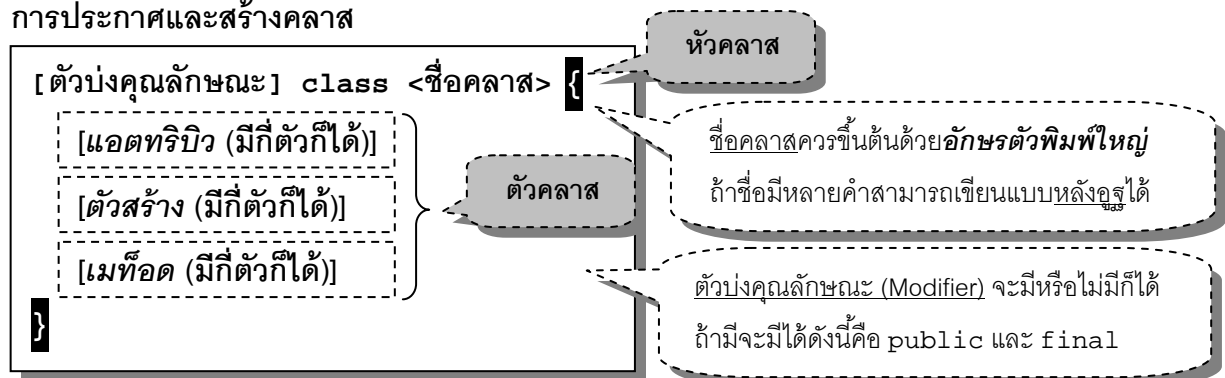
- 5) ☐ คลาสสามารถมีเมทอดเพียงส่วนเดียวได้โดยไม่ต้องมีแอตทริบิว
- 6) ☐ อ็อบเจ็คเป็นตัวแทนของคลาส และมีแอตทริบิวและเมทอดทุกตัวเหมือนกับคลาสนั้นมี
- 7) ☐ ตัวแปรที่ประกาศภายในเมทอดใดๆ สามารถเรียกใช้ได้เฉพาะเมทอดนั้น
- 8) ☐ ตัวแปรที่ประกาศภายนอกเมทอดใดๆ สามารถเรียกใช้ได้จากทุกๆ เมทอด
- 9) ☐ ตัวแปรที่ประกาศภายนอกเมทอดใดๆ คือแอตทริบิว หรือตัวแปรของคลาส
- 10) ☐ ตัวแปรที่ประกาศภายในเมทอดใดๆ คือตัวแปรทั่วไป หรือตัวแปรของเมทอด
- 11) ☐ แอตทริบิวสามารถประกาศชื่อซ้ำกันได้แต่จะต้องกำหนดค่าเริ่มต้นที่ต่างกัน
- 12) ☐ เมทอดสามารถประกาศชื่อซ้ำกันได้แต่จะต้องมีรายการพารามิเตอร์ที่ต่างกัน
- 13) ☐ ตัวสร้างจะปรากฏอยู่ในทุกอ็อบเจ็คที่สร้างจากคลาสดั้งแบบ
- 14) ☐ ตัวสร้างทำหน้าที่ในการกำหนดค่าเริ่มต้นให้กับแอตทริบิวทุกตัวเมื่อเริ่มสร้างอ็อบเจ็คใหม่
- 15) ☐ แอตทริบิวในคลาสดั้งแบบจะต้องกำหนดค่าเริ่มต้นเสมอเมื่อเริ่มสร้างคลาสดั้งแบบ

2. คลาส (Classes)

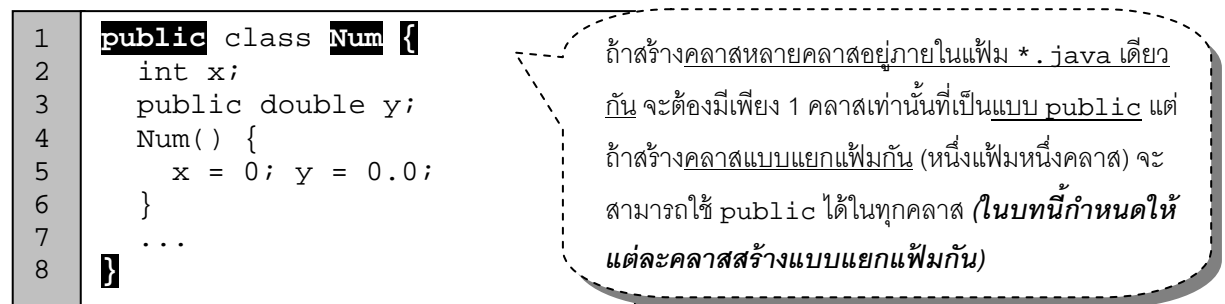
1. การประกาศและสร้างองค์ประกอบของคลาส

คลาสหนึ่งๆ มีองค์ประกอบ 3 ส่วนได้แก่ แอตทริบิว ตัวสร้าง และ เมทอด ซึ่งมีวิธีการประกาศและสร้างดังต่อไปนี้

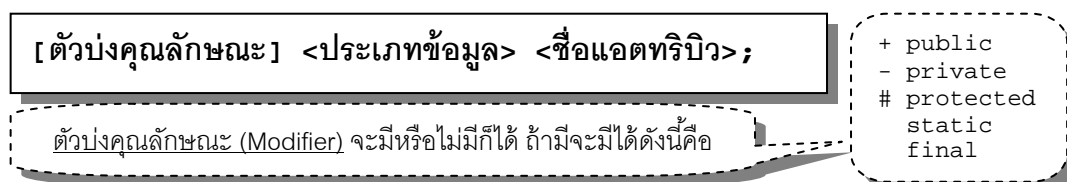
1) การประกาศและสร้างคลาส



ตัวอย่าง



2) การประกาศและสร้างแอตทริบิว ซึ่งจะคล้ายกับการประกาศและสร้างตัวแปรในบทที่ 2



ตัวอย่าง

```

1 public class Data {
2     public static int n = 1;
3     private double data;
4     ...
5 }

```

โดยทั่วไปแอตทริบิวต์ที่มี static กำกับอยู่ จะกำหนดค่าเริ่มต้นให้แอตทริบิวต์นั้นที่เมื่อเริ่มสร้างคลาสด้านแบบ แต่แอตทริบิวต์อื่นๆ ที่ไม่มี static กำกับ จะไม่มีการกำหนดค่าเริ่มต้นใดๆ ทั้งสิ้น

เมทอดทุกเมทอดภายในคลาสสามารถเรียกใช้งานแอตทริบิวต์ได้

3) การประกาศและสร้างตัวสร้าง

ตัวสร้าง (Constructor) คือ เมทอดที่มีชื่อเดียวกับชื่อคลาส (เมทอดที่ชื่อขึ้นต้นด้วยตัวอักษรพิมพ์ใหญ่)

[ตัวบ่งคุณลักษณะ] <ชื่อตัวสร้าง> ([พารามิเตอร์]) {

[กำหนดค่าเริ่มต้นให้กับแอตทริบิวต์ทุกตัว]

}

ตัวสร้างไม่มี Return Type

ตัวบ่งคุณลักษณะจะมีหรือไม่มีก็ได้ ถ้ามีจะมีได้แบบเดียวคือ public

ตัวอย่าง

```

1 public class Number {
2     public long x;
3     private int y;
4     public Number() {
5         x = 0L;
6         y = 0;
7     }
8     public Number(long a, int b) {
9         x = a;
10        y = b;
11    }
12    public Number(double k) {
13        x = y = (int) k;
14    }
15    ...
16 }

```

วิธีการเขียนตัวสร้าง คือ ให้ดูชื่อคลาสว่าชื่ออะไร ก็ให้ชื่อตัวสร้างเป็นชื่อเดียวกับชื่อคลาสนั้น

ตัวสร้างมีได้หลายแบบซึ่ง Overload กัน

ตัวสร้างถูกเรียกใช้ทุกครั้งเมื่อมีการสร้างอ็อบเจกต์ใหม่

คลาสใดที่ไม่มีตัวสร้าง คอมไพเลอร์จะใส่ตัวสร้างให้อัตโนมัติซึ่งเป็นตัวสร้างเริ่มต้น (Default Constructor) แต่ถ้ามีตัวสร้างอยู่แล้ว คอมไพเลอร์จะใช้ตัวสร้างนั้นในการทำงาน โดยไม่มีการใส่ตัวสร้างเริ่มต้นให้อีก

4) การประกาศและสร้างเมทอด จะเหมือนกับการประกาศและสร้างเมทอดในบทที่ 8

[ตัวบ่งคุณลักษณะ] <ประเภทข้อมูลที่ส่งกลับ> <ชื่อเมทอด> ([พารามิเตอร์]) {

[ตัวแปรและคำสั่ง]

}

ตัวเมทอด

หัวเมทอด

ตัวอย่าง

```

1 public class Operation {
2     ...
3     public int incrX(int x) {
4         x = x + 1;
5         return x;
6     }
7     ...
8 }

```

มอดดิฟายเออร์ของเมทอดมีได้ดังนี้คือ public, private, protected, static และ final

ดูรายละเอียดเพิ่มเติมในบทที่ 8 (ที่เรียนผ่านมา)

โจทย์ข้อที่ 3 [ระดับง่าย] จงเขียนโปรแกรมภาษาจาวาให้สมบูรณ์เพื่อสร้างคลาสชื่อ `Course` ซึ่งเป็นต้นแบบของวิชาใดๆ ในจุฬาฯ โดยประกอบไปด้วยรายละเอียดดังต่อไปนี้ (30 คะแนน)

```
public class Course {
```

```
    //แอตทริบิวต์ 3 ตัว (โดยไม่ต้องกำหนดค่าใดๆ ให้กับแอตทริบิวต์) มีดังต่อไปนี้ (3 คะแนน)
```

```
    //แอตทริบิวต์ชื่อ id แบบ private เพื่อเก็บรหัสวิชา (Course ID)
```

```
    //แอตทริบิวต์ชื่อ title แบบ public เพื่อเก็บชื่อวิชา (Course Title)
```

```
    //แอตทริบิวต์ชื่อ credit แบบ protected เพื่อเก็บจำนวนหน่วยกิต (Credit)
```

ให้จินตนาการว่าแอตทริบิวต์เหล่านี้มีค่าพร้อมแล้ว สามารถเรียกใช้ได้เลย

```
//ตัวสร้างแบบไม่มีพารามิเตอร์ใดๆ เพื่อกำหนดค่าเริ่มต้นให้กับแอตทริบิวต์ทั้ง 3 ตัวตามค่าเริ่มต้นของประเภทแอตทริบิวต์นั้นๆ (เช่น int เป็น 0, string เป็น "" เป็นต้น) (3 คะแนน)
```

```
//ตัวสร้างแบบมีพารามิเตอร์ 3 ตัว เพื่อกำหนดค่าเริ่มต้นให้กับแอตทริบิวต์ทั้ง 3 ตัวตามลำดับ (3 คะแนน)
```

```
//เมธอด setID(...) แบบ public ที่รับตัวเลขจำนวนเต็มผ่านทางพารามิเตอร์หนึ่งค่า เพื่อใช้สำหรับกำหนดค่าให้กับแอตทริบิวต์ id โดยไม่ต้องคืนค่ากลับ (3 คะแนน)
```

//เมื่อกด setTitle(...)แบบ public ที่รับข้อความผ่านทางพารามิเตอร์หนึ่งค่า เพื่อใช้สำหรับกำหนดค่าให้กับแอตทริบิว title โดยไม่ต้องคืนค่ากลับ (3 คะแนน)

//เมื่อกด setCredit(...)แบบ public ที่รับตัวเลขจำนวนจริงผ่านทางพารามิเตอร์หนึ่งค่า เพื่อใช้สำหรับกำหนดค่าให้กับแอตทริบิว credit โดยไม่ต้องคืนค่ากลับ (3 คะแนน)

//เมื่อกด getLevel(...)แบบ public เพื่อคืนค่าระดับการศึกษาที่ระบุในรหัสวิชา โดยให้ตัวเลขหลักที่ 3 จากทางขวามือของรหัสวิชาแทนระดับการศึกษา เช่น 2110191 ตัวเลข 1 แทนระดับการศึกษาเป็นต้น ซึ่งระดับการศึกษาประกอบไปด้วย 2 ระดับได้แก่ Undergraduate (ตัวเลข 1-4) และ Graduated (ตัวเลข 5 เป็นต้นไป) โดยกำหนดให้คืนค่าเป็นชื่อของระดับการศึกษานั้นๆ (3 คะแนน)

//เมื่อกด getFaculty(...)แบบ public เพื่อคืนค่ารหัสคณะที่เปิดสอนวิชาดังกล่าว โดยรหัสคณะคือตัวเลขสองหลักแรกของรหัสวิชาเริ่มจากทางซ้ายมือ เช่น 2110191 ตัวเลข 21 แทนรหัสคณะ เป็นต้น

(3 คะแนน)

//เมธอด `getDepartment(...)` แบบ `public` เพื่อคืนค่ารหัสภาควิชาที่เปิดสอนวิชาดังกล่าว โดยรหัสดังกล่าวคือตัวเลขสองหลักที่ 3-4 ของรหัสวิชาเริ่มจากทางซ้ายมือ เช่น 2110191 ตัวเลข 10 แทนรหัสของภาควิชา เป็นต้น (3 คะแนน)

//เมธอด `toString(...)` แบบ `public` เพื่อคืนค่ารายละเอียดของวิชาที่เก็บอยู่ในแอตทริบิวต์ตัวในรูปแบบของข้อความดังตัวอย่างเช่น 2110191 Innovative Thinking (3) เป็นต้น (3 คะแนน)

} //End of class

2. ขอบเขตของตัวแปร พารามิเตอร์ และแอตทริบิวต์

- 1) **ตัวแปร** คือ ตัวแปรที่ประกาศอยู่ในเมธอด โดยจะปรากฏอยู่ที่ตัวเมธอด (Body) ซึ่งถ้าตัวแปรใดประกาศในเมธอดไหนแล้วจะเรียกใช้งานได้แต่เพียงในเมธอดนั้นเท่านั้น (Local Variable)
- 2) **พารามิเตอร์** คือ ตัวแปรที่ประกาศอยู่ในเมธอด แต่จะปรากฏอยู่ที่หัวเมธอด (Head) จึงเรียกชื่อเสียใหม่ว่าพารามิเตอร์ ซึ่งพารามิเตอร์ใดๆ นั้นจะมีคุณสมบัติเช่นเดียวกับตัวแปร
- 3) **แอตทริบิวต์** คือ ตัวแปรที่ประกาศอยู่ในคลาสแต่อยู่นอกเมธอดใดๆ ซึ่งถ้าแอตทริบิวต์ประกาศขึ้นแล้วจะสามารถเรียกใช้งานได้จากทุกเมธอดภายในคลาสนั้น (Global Variable)

ตัวอย่าง

<pre> 1 public class Test { 2 private static int x = 10; 3 private static int y = 1; 4 public static int getX() { return x; } 5 public static int getX(int x) { return x; } 6 public static int incX(int y) { return x + y; } 7 public static void main(String[] args) { 8 int x = 5; 9 System.out.println(x); 10 System.out.println(y); 11 System.out.println(getX()); 12 System.out.println(getX(2)); 13 System.out.println(incX(x)); 14 } 15 }</pre>	<p>x และ y คือ แอตทริบิวต์</p> <p>getX() คืนค่าแอตทริบิวต์ x</p> <p>getX(int x) คืนค่าพารามิเตอร์</p> <p>ค่า x มาจากแอตทริบิวต์ x</p> <p>ค่า y มาจากพารามิเตอร์ y</p> <p>แสดงค่า x จากตัวแปร x</p> <p>แสดงค่า y จากแอตทริบิวต์ y</p> <p>ตัวแปร x ในเมธอด main(...) เป็นคนละตัวกับพารามิเตอร์ x ในเมธอด getX(int x)</p>	<p>→ 5</p> <p>→ 1</p> <p>→ 10</p> <p>→ 2</p> <p>→ 15</p>
--	--	--

4) ภาพรวมของตัวแปร พารามิเตอร์ และแอตทริบิว

```

public class TestClass {
    1 Attribute
    public int add( 2 Parameter ) {
        3 Variable
        return ... ;
    } //End of method
} //End of class

```

ตัวแปรกับพารามิเตอร์ประกาศชื่อซ้ำกันได้หรือไม่?

ตัวแปรกับแอตทริบิวประกาศชื่อซ้ำกันได้หรือไม่?

โจทย์ข้อที่ 4 [ระดับง่าย] จงแสดงผลลัพธ์จากโปรแกรมและตอบคำถามต่อไปนี้ (10 คะแนน)

```

1 public class Test {
2     private static int x = 5;
3     private static int y = 2;
4     public static void main(String[] args) {
5         int x = 9;
6         System.out.println(x);
7         System.out.println(getX());
8         System.out.println(showX());
9         y = 1;
10        System.out.println(y);
11        System.out.println(getY());
12        System.out.println(showY());
13        int y = 11;
14        System.out.println(y);
15        System.out.println(getY());
16        System.out.println(showY());
17    }
18    public static int getX() {
19        return x;
20    }
21    public static int getY() {
22        return y;
23    }
24    public static int showX() {
25        int x = 7;
26        return x;
27    }
28    public static int showY() {
29        int y = 12;
30        return y;
31    }
32 }

```

1) ค่า y ในบรรทัดที่ 10 คือแอตทริบิว y หรือตัวแปร y

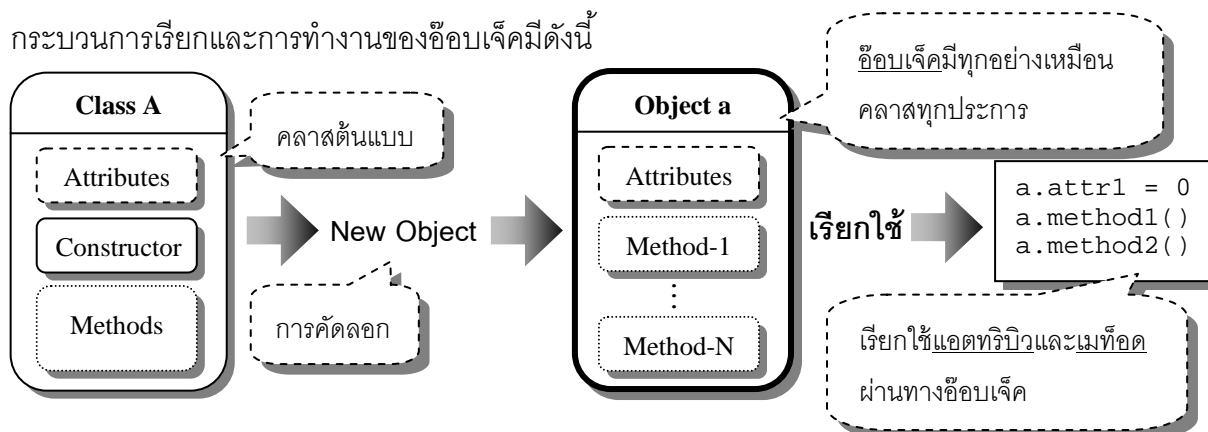
2) ค่า y ในบรรทัดที่ 14 คือแอตทริบิว y หรือตัวแปร y

3. อ็อบเจ็ค (Objects)

1. นิยามของอ็อบเจ็ค

- อ็อบเจ็คเป็นตัวแทนของคลาส ซึ่งใช้สำหรับอ้างอิงแอตทริบิวและเมธอดภายในคลาสนั้น โดยทั่วไปคลาสที่สร้างขึ้นเป็นเพียงคลาสต้นแบบที่ไม่สามารถใช้งานได้จริง ดังนั้นการสร้างอ็อบเจ็คก็เป็นเสมือนการคัดลอกคลาสต้นแบบเพื่อนำไปใช้งานนั่นเอง (การคัดลอกทำให้คลาสต้นแบบไม่เสียหาย)

- 2) กระบวนการเรียกและการทำงานของอ็อบเจกต์มีดังนี้



2. การประกาศและสร้างอ็อบเจกต์

- 1) การประกาศตัวแปรอ็อบเจกต์

```
<ชื่อคลาส> <ชื่ออ็อบเจกต์>;
```

เหมือนการประกาศตัวแปรทั่วไปเพียงแต่มีชนิดตัวแปรเป็นชื่อคลาส

เช่น Student std; / Car car; / Account obj;

- 2) การสร้างอ็อบเจกต์ (ใช้การ new)

```
<ชื่ออ็อบเจกต์> = new <ชื่อคลาส> (...);
```

เช่น std = new Student(52300121); / car = new Car(); / obj = new Account(101);

- 3) รวมขั้นตอน 2 ขั้นตอนในข้อ 1 และ 2 เข้าด้วยกัน (นิยมประกาศและสร้างอ็อบเจกต์แบบวิธีนี้)

```
<ชื่อคลาส> <ชื่ออ็อบเจกต์> = new <ชื่อคลาส> (...);
```

```
<ชื่อคลาส> <ชื่ออ็อบเจกต์> = new <ตัวสร้าง>;
```

ส่วนของ <ชื่อคลาส> (...) ก็คือตัวสร้าง (Constructor) ของคลาสนั่นเอง

เช่น Student std = new Student(52300121); / Account obj = new Account(101);

หมายเหตุ คำพารามิเตอร์ที่จะใส่เข้าไปภายในวงเล็บหลังชื่อคลาสหรือหลังชื่อตัวสร้างนั้น ให้พิจารณาจากคลาสต้นแบบว่ามีตัวสร้างแบบใดบ้างที่ประกาศขึ้นใช้งาน

- 4) ตัวอย่างการสร้างและใช้งานอ็อบเจกต์

```
1 public class Digit {
2     public int x;
3     Digit() {
4         x = 0;
5     }
6     Digit(int n) {
7         x = n;
8     }
9     public int getX() {
10        return x;
11    }
12 }
```

คลาสต้นแบบ

```
1 public class TestDigit {
2     public static void main(String[] args) {
3         Digit d = new Digit();
4         Digit e = new Digit(1);
5         Digit f = new Digit(2);
6         int digit1 = d.getX();
7         int digit2 = e.getX();
8         int digit3 = f.getX();
9     }
10 }
```

คลาสประมวลผล

แอตทริบิวต์ทุกตัวของอ็อบเจกต์จะถูกกำหนดค่าเริ่มต้นโดยตัวสร้าง

ภายในเมทอด main มีการเรียกใช้งานคลาสต้นแบบผ่านทางอ็อบเจกต์ โดยการเอาชื่ออ็อบเจกต์ + จุด + สิ่งที่จะเรียก

โจทย์ข้อที่ 5 [ระดับง่าย] จงหาผลลัพธ์จากการทำงานของโปรแกรมต่อไปนี้ (8 คะแนน)

```

1 public class Student {
2     public int id;
3     public String name;
4     Student() { id = 0; name = ""; }
5     Student(int i, String n) { id = i; name = n; }
6     public String getName() { return name; }
7     public int getID() { return id; }
8     public char calGrad(int score) {
9         if(score > 60) { return 'S'; }
10        else { return 'U'; }
11    }
12 }

```

คลาสต้นแบบ

```

1 public class TestStudent {
2     public static void main(String args[]){
3         Student y = new Student(101,"Taksin");
4         Student z = new Student(102,"Apisit");
5         Student x = new Student();
6         System.out.println(x.getID() + "," + x.getName());
7         System.out.println(y.getID() + "," + y.getName());
8         System.out.println((z.getID() + 1) + "," + z.getName());
9         System.out.println(y.getName()+ ":" + y.calGrad(49));
10        System.out.println(z.getName()+ ":" + z.calGrad(79));
11    }
12 }

```

คลาสประมวลผล

โจทย์ข้อที่ 6 [ระดับง่าย] จงเขียนคลาส TestNumber ให้สมบูรณ์เพื่อเรียกใช้งานคลาส Number โดยการสร้างอ็อบเจกต์ที่มีชนิดเป็น Number และเรียกใช้งานเมทอดตามเงื่อนไขต่อไปนี้ (10 คะแนน)

```

1 public class Number {
2     private double x;
3     private double y;
4     Number() { x = y = 0; }
5     Number(double a, double b) { x = a; y = b; }
6     public double add() { return x + y; }
7     public double sub() { return x - y; }
8     public double mul() { return x * y; }
9     public double div() { return x / y; }
10    public double mod() { return x % y; }
11 }

```

คลาสต้นแบบ

//ประกาศหัวคลาส TestNumber

import java.util.Scanner;

```

public static void main(String[] args) {
    Scanner kb = new Scanner(System.in);

```

//สร้างอ็อบเจกต์ชื่อ no1 ที่มีชนิดเป็นคลาส Number (เป็นตัวแทนของคลาส Number)

//รับค่าจำนวนจริงจากแป้นพิมพ์ 2 ค่าเพื่อใช้ในการสร้างอ็อบเจ็คชื่อ no2 ที่มีชนิดเป็นคลาส Number

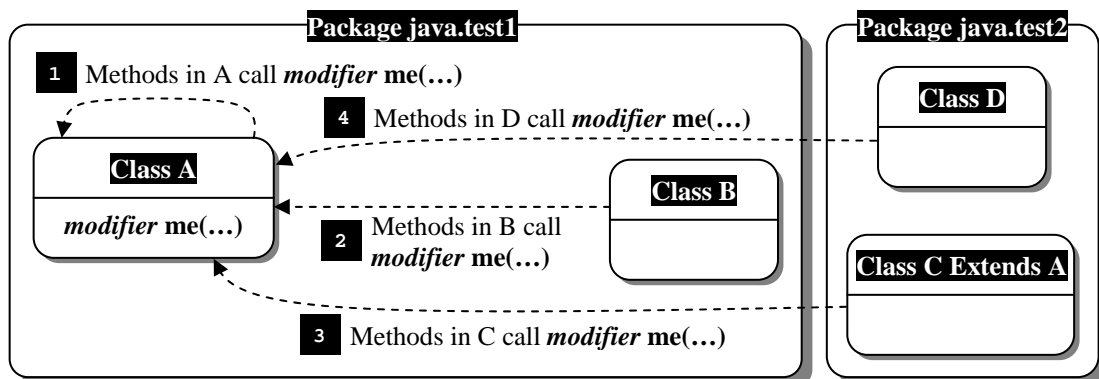
//แสดงผลการบวก การลบ การคูณ การหาร และการมีอด โดยเรียกเมทอดผ่านทางอ็อบเจ็ค no2

```
} //End of main
} //End of class
```

4. การกำหนดคุณสมบัติและการใช้งานคลาสและอ็อบเจ็ค

1. การกำหนดตัวบ่งคุณลักษณะให้กับแอตทริบิวต์และเมทอด

- 1) ตัวบ่งคุณลักษณะที่ใช้กำหนดขอบเขตการเรียกใช้แอตทริบิวต์และเมทอดได้แก่ **public**, **private** และ **protected** ซึ่งจะเป็นตัวบอกว่าแอตทริบิวต์และเมทอดจะถูกเรียกใช้งานจากที่ไหนได้บ้าง (Where)



- 1 หมายถึง เรียกใช้เมทอด **modifier me(...)** จากภายในคลาสเดียวกัน (คลาส A)
- 2 หมายถึง เรียกใช้เมทอด **modifier me(...)** จากต่างคลาสกันแต่อยู่ภายในแพ็คเกจเดียวกัน
- 3 หมายถึง เรียกใช้เมทอด **modifier me(...)** จากต่างแพ็คเกจกันแต่คลาสที่เรียกใช้เป็นสับคลาสกัน
- 4 หมายถึง เรียกใช้เมทอด **modifier me(...)** จากต่างแพ็คเกจกันหรือจากที่ใดๆ

สรุปเป็นตารางขอบเขตการเรียกใช้งานได้ดังนี้

รายละเอียดของ ตัวบ่งคุณลักษณะ	คลาสเดียวกัน แพ็คเกจเดียวกัน	ต่างคลาสกัน แพ็คเกจเดียวกัน	เป็นสับคลาสกัน (คลาสแม่ลูก)	ต่างคลาสกัน ต่างแพ็คเกจกัน
public me(...)	✓	✓	✓	✓
private me(...)	✓	✗	✗	✗
protected me(...)	✓	✓	✓	✗
me(...)	✓	✓	✗	✗

หมายเหตุ โดยทั่วไปแอตทริบิวต์นิยามประกาศเป็น **private** และเมทอดนิยามประกาศเป็น **public** แต่ถ้าไม่เช่นนั้นก็ให้ประกาศไปตามความต้องการหรือตามที่โจทย์กำหนด

- 2) ตัวบ่งคุณลักษณะที่ใช้กำหนดค่าประจำคลาสหรือประจำอ็อบเจ็คของแอตทริบิวต์และเมทอดได้แก่ **static** ซึ่งจะเป็นตัวบอกว่าแอตทริบิวต์และเมทอดจะถูกเรียกใช้งานอย่างไรได้บ้าง (How)

รายละเอียดของ ตัวบ่งคุณลักษณะ	เรียกใช้ผ่านทางคลาส	เรียกใช้ผ่านทางอ็อบเจ็ค
ค่าประจำคลาส (มี static)	✓	✓
ค่าประจำอ็อบเจ็ค (ไม่มี static)	✗	✓

โจทย์ข้อที่ 7 [ระดับง่าย] จงประกาศและสร้างแอตทริบิวต์หรือหวัเมทอดตามข้อกำหนดต่อไปนี้ (18 คะแนน)

- 1) แอตทริบิวต์ประจำอ็อบเจ็คชื่อ **data** ชนิดจำนวนจริง โดยสามารถเรียกใช้งานได้จากคลาสใดๆ

- 2) แอตทริบิวต์ประจำคลาสชื่อ **var** ชนิดจำนวนเต็ม โดยมีค่าเริ่มต้นเท่ากับ 10 และสามารถเรียกใช้งานได้จากภายในคลาสเดียวกันเท่านั้น

- 3) แอตทริบิวต์ประจำคลาสชื่อ **check** ชนิดค่าความจริง โดยสามารถเรียกใช้งานได้จากสับคลาส

- 4) แอตทริบิวต์ประจำอ็อบเจ็คชื่อ **stdName** เก็บรายชื่อนิสิตจำนวน 351 คน โดยสามารถเรียกใช้งานได้จากภายในคลาสเดียวกันเท่านั้น

- 5) แอตทริบิวต์ประจำคลาสชื่อ `m` เก็บเมตริกซ์ขนาดจำนวนเต็มขนาด 8×5 โดยสามารถเรียกใช้จากที่ใดก็ได้

- 6) เมธอดประจำอ็อบเจ็คชื่อ `show` เพื่อแสดงชื่อนิสิตและผลการเรียนเฉลี่ยที่รับจากพารามิเตอร์ชั้นบนจอภาพ โดยสามารถเรียกใช้งานได้เฉพาะคลาสที่ประกาศเมธอด `show` นี้เท่านั้น

- 7) เมธอดประจำคลาสชื่อ `search` เพื่อใช้ค้นหาตัวเลขจำนวนเต็ม `x` ที่รับเข้ามาว่ามีปรากฏอยู่ ณ ตำแหน่งใดในอาร์เรย์ `num` ที่รับเข้ามาเช่นกัน โดยสามารถเรียกใช้งานได้จากคลาสลูกหลาน

- 8) เมธอดประจำอ็อบเจ็คชื่อ `mulMatrix` เพื่อคำนวณค่าหาผลคูณของเมตริกซ์สองมิติขนาดจำนวนจริง 2 ตัวใดๆ โดยสามารถเรียกใช้จากที่ใดก็ได้

- 9) เมธอดประจำคลาสชื่อ `union` เพื่อใช้ยูเนียนเซตของจำนวนเต็ม 2 เซตใดๆ โดยเป็นแบบสาธารณะ

2. วิธีการเรียกใช้งานแอตทริบิวต์และเมธอด

- 1) **เรียกโดยตรง** เป็นการเรียกแอตทริบิวต์หรือเมธอดที่อยู่ภายในคลาสเดียวกัน (ตามหลักการเรียกเมธอดในบทที่ 8-11 ที่ผ่านมา) โดยจะพิจารณาเมธอดที่มี `static` และไม่มี `static` ดังเงื่อนไขต่อไปนี้

	เมธอดผู้ถูกเรียก	
	มี	ไม่มี
มี	✓	✗
ไม่มี	✓	✓

มี static เรียก มี static ได้
มี static เรียก ไม่มี static ไม่ได้
ไม่มี static เรียก มี static ได้
ไม่มี static เรียก ไม่มี static ได้

- 2) **เรียกผ่านคลาส** เป็นการเรียกแอตทริบิวต์หรือเมธอดประจำคลาส (มี `static`) ที่อยู่ต่างคลาสนั้น

- (1) **เรียกแอตทริบิวต์ผ่านคลาส** ซึ่งเรียกแอตทริบิวต์ประเภทนี้ว่า ตัวแปรประจำคลาส (Class Variable)

```
<ชื่อคลาส> . <ชื่อแอตทริบิวต์>;
```

ขึ้นต้นด้วยตัวพิมพ์ใหญ่ ลงท้ายไม่มีวงเล็บ

เช่น `Math.PI;` `Color.RED;`

- (2) เรียกเมทอดผ่านคลาส ซึ่งเรียกเมทอดประเภทนี้ว่า เมทอดประจำคลาส (Class Method)

```
<ชื่อคลาส> . <ชื่อเมทอด> (...);
```

ขึ้นต้นด้วยตัวพิมพ์ใหญ่ ลงท้ายมีวงเล็บ

เช่น `Math.pow(2,3); Time.start();`

- 3) เรียกผ่านอ็อบเจ็ค เป็นการเรียกแอตทริบิวหรือเมทอดประจำอ็อบเจ็ค (ไม่มี static) ที่อยู่ต่างคลาสิกกัน

- (1) เรียกแอตทริบิวผ่านอ็อบเจ็ค ซึ่งเรียกแอตทริบิวประเภทนี้ว่า ตัวแปรประจำอ็อบเจ็ค (Object Variable)

หรือ ตัวแปรตัวแทน (Instance Variable)

```
<ชื่ออ็อบเจ็ค> . <ชื่อแอตทริบิว>;
```

ขึ้นต้นด้วยตัวพิมพ์เล็ก ลงท้ายไม่มีวงเล็บ

เช่น `objStd.id; no.x; g.y;`

- (2) เรียกเมทอดผ่านอ็อบเจ็ค ซึ่งเรียกเมทอดประเภทนี้ว่า เมทอดประจำอ็อบเจ็ค (Object Method) หรือ เมทอดตัวแทน (Instance Method)

```
<ชื่ออ็อบเจ็ค> . <ชื่อเมทอด> (...);
```

ขึ้นต้นด้วยตัวพิมพ์เล็ก ลงท้ายมีวงเล็บ

เช่น `std.get(5230121); c.stop();`

จงตอบคำถาม

แอตทริบิวหรือเมทอดที่เป็นแบบ static จะสามารถเรียกใช้งานผ่านทางอ็อบเจ็คได้หรือไม่
ได้ → เราสามารถเรียกใช้ทุกอย่างผ่านทางอ็อบเจ็คได้ไม่ว่าจะเป็น static หรือไม่ก็ตาม.....

โจทย์ข้อที่ 8 [ระดับง่าย] จงใช้ตัวเลือกต่อไปนี้เพื่อระบุประเภทของแอตทริบิวหรือเมทอด (20 คะแนน)

(A) Class Variable (B) Class Method (C) Object Variable (D) Object Method

ข้อ	แอตทริบิวหรือเมทอด	คำตอบ
1.	<code>Math.sqrt(x)</code>	
2.	<code>p.colorCode(s)</code>	
3.	<code>Array.equals(a, b)</code>	
4.	<code>v.x</code>	
5.	<code>System.getProperties()</code>	
6.	<code>Integer.MAX_VALUE</code>	
7.	<code>kb.nextInt()</code>	
8.	<code>rectangle.setSize(w, h)</code>	
9.	<code>Math.PI</code>	
10.	<code>a.appendArrays(x,y)</code>	

ข้อ	แอตทริบิวหรือเมทอด	คำตอบ
11.	<code>std.grad</code>	
12.	<code>Sqt.borders(a,b)</code>	
13.	<code>in.readLine()</code>	
14.	<code>System.in</code>	
15.	<code>Math.random()</code>	
16.	<code>in.hasNext()</code>	
17.	<code>i.id_code</code>	
18.	<code>out.close()</code>	
19.	<code>String.format("%4d", x)</code>	
20.	<code>o.name</code>	

โจทย์ข้อที่ 9 [ระดับปานกลาง] จงหาผลลัพธ์จากการทำงานของโปรแกรมต่อไปนี้ (15 คะแนน)

```

1 public class Val {
2     public static int x;
3     public static int y;
4     public static String s;
5     public Val() { x = 0; y = 0; s = ""; }
6     public Val(int x, int y) { this.x = x; this.y = y; }
7     public static int getX() { return x; }
8     public static int getY() { return y; }
9     public static int getY(int y) { return y; }
10    public static String getS() { return s; }
11    public static String getS(String s) { return s; }
12 }

```

```

1 public class Test {
2     public static void main(String[] args) {
3         System.out.println(Val.x + "," + Val.y);
4         Val.x = 6; Val.y = 10;
5         System.out.println(Val.getX() + "," + Val.getY());
6         Val.s = "Good By A";
7         System.out.println(Val.getS());
8         System.out.println(Val.getS("See You F"));
9         Val v = new Val();
10        System.out.println(v.x + "," + v.y);
11        v.x = 5; v.y = 7;
12        System.out.println(v.getX() + "," + v.getY());
13        System.out.println(Val.getX() + "," + Val.getY());
14        Val w = new Val(9, 7);
15        System.out.println(w.getX() + "," + w.getY());
16        System.out.println(v.getX() + "," + v.getY());
17        v.s = "I love Java";
18        System.out.println(w.getS());
19    }
20 }

```

ข้อนี้แสดงให้เห็นถึงผลกระทบในการประกาศแอตทริบิวต์ให้เป็นแบบ static ซึ่งจะทำให้ค่าของแอตทริบิวต์นั้นเปลี่ยนแปลงแบบ Global

โจทย์ข้อที่ 10 [ระดับปานกลาง] จงเขียนคลาส Dice ซึ่งเป็นตัวแทนของลูกเต๋า n หน้าใดๆ เช่น ลูกเต๋า 6 หน้า ลูกเต๋า 12 หน้า เป็นต้น และเขียนคลาส TestDice เพื่อเรียกใช้เมทอดจากคลาส Dice (20 คะแนน)

```

public class Dice {
    //แอตทริบิวต์ประจำอ็อบเจ็คแบบ public ชื่อ face เพื่อเก็บจำนวนหน้าของลูกเต๋า
    //แอตทริบิวต์ประจำอ็อบเจ็คแบบ public ชื่อ value เพื่อเก็บหมายเลขหน้าหงายของลูกเต๋า

```

//ตัวสร้างแบบไม่มีพารามิเตอร์ เพื่อกำหนดจำนวนหน้าของลูกเต๋าเป็น 2 และกำหนดหน้าหงายของลูกเต๋าโดยการสุ่มค่า

//ตัวสร้างแบบมีพารามิเตอร์หนึ่งตัว เพื่อกำหนดจำนวนหน้าของลูกเต๋าตามตัวเลขที่ระบุทางพารามิเตอร์
และกำหนดหน้าหงายของลูกเต๋าโดยการสุ่มค่า

//ตัวสร้างแบบมีพารามิเตอร์สองตัว เพื่อกำหนดจำนวนหน้าของลูกเต๋าและกำหนดหน้าหงายของลูกเต๋า
ตามตัวเลขที่ระบุทางพารามิเตอร์ตามลำดับ

//เมทอดประจำอ็อบเจ็คชื่อ roll เพื่อใช้โยนลูกเต๋า ซึ่งจะสุ่มหมายเลขหน้าหงายของลูกเต๋ามีค่าตั้งแต่
1 จนถึงจำนวนหน้าที่เป็นไปได้ของลูกเต๋านั้น และเมื่อสุ่มหน้าหงายได้แล้ว ก็จะนำไปกำหนดค่าให้กับ
แอตทริบิวต์ value ที่เก็บหมายเลขหน้าหงายของลูกเต๋า โดยที่เมทอดนี้ไม่ต้องมีการคืนค่ากลับ

//เมทอดประจำอ็อบเจ็คชื่อ setValue เพื่อใช้ในการพลิกหน้าหงายของลูกเต๋าตามที่ต้องการ ซึ่งจะรับ
ค่าหมายเลขหน้าหงายของลูกเต๋ามาผ่านทางพารามิเตอร์ และจะนำไปกำหนดค่าให้กับแอตทริบิวต์ value
ที่เก็บหมายเลขหน้าหงายของลูกเต๋า โดยที่เมทอดนี้ไม่ต้องมีการคืนค่ากลับ

//เมทอดประจำอ็อบเจ็คชื่อ getFace เพื่อคืนค่าจำนวนหน้าของลูกเต๋า


```
//เมื่อกดปุ่มจําอับเจ็คชื่อ getValue เพื่อคืนค่าหมายเลขหน้าหายของลูกเต๋า
```

--

//เมื่อกดประจำอ็อบเจ็คชื่อ show เพื่อแสดงจำนวนหน้าและหมายเลขหน้าท้ายของลูกเต๋าที่ละบรรทัด

--

```
} //End of class
```

```
public class TestDice {
```

/* สร้างลูกเต๋าดังขึ้น 3 ลูก ในรูปของอ็อบเจ็คที่มีชื่อเป็น d1, d2 และ d3 ตามลำดับมีรายละเอียดดังนี้

ข้อบเจ็ด d1 เป็นลูกเต๋าที่ไม่ได้ระบุจำนวนหน้าและหมายเลขหน้าทาง

ข้อบเจ็ด ๔๒ เป็นลูกเต๋า ๑๓ หน้า ที่ไม่ได้ระบุหมายเลขหน้าหงาย

ข้อบเจ็ด d3 เป็นลูกเต้า 30 หน้า ที่มีหมายเลขหน้าหงายเป็น 9

พร้อมทั้งแสดงข้อมูลของลูกเต๋าทั้ง 3 ลูกขึ้นบนจอภาพ แล้วทำการโยนลูกเต๋าทั้ง 3 ลูกพร้อมๆ กัน 20 ครั้ง โดยในแต่ละครั้งให้แสดงหมายเลขหน้าหงายของลูกเต๋าทั้ง 3 ลูกขึ้นบนจอภาพ */



```
} //End of class
```

โจทย์ข้อที่ 11 [ระดับปานกลาง] จงเขียนคลาส RealNumber ซึ่งเป็นตัวแทนของจำนวนจริงใดๆ หนึ่งจำนวน และเขียนคลาส TestRealNumber เพื่อเรียกใช้เมทอดจากคลาส RealNumber (15 คะแนน)

```
public class RealNumber {
```

```
//แอตทริบิวต์ประจำอ็อบเจ็คแบบ public ชื่อ num เพื่อเก็บค่าจำนวนจริง
```

```
//ตัวสร้างแบบไม่มีพารามิเตอร์ เพื่อกำหนดค่าให้กับแอตทริบิวต์ num ตามค่า Default
```

```
//ตัวสร้างแบบมีพารามิเตอร์ เพื่อกำหนดค่าให้กับแอตทริบิวต์ num ตามค่าที่ระบุ
```

```
//เมทอดประจำอ็อบเจ็คชื่อ plus เพื่อรับค่าจำนวนจริงหนึ่งจำนวนเข้ามาทางพารามิเตอร์ แล้วนำมาหาผลบวกระหว่างแอตทริบิวต์ num กับค่าตัวเลขนั้นพร้อมทั้งคืนค่ากลับ
```

```
//เมทอดประจำอ็อบเจ็คชื่อ diff เพื่อรับค่าจำนวนจริงหนึ่งจำนวนเข้ามาทางพารามิเตอร์ แล้วนำมาหาผลต่างระหว่างแอตทริบิวต์ num กับค่าตัวเลขนั้นพร้อมทั้งคืนค่ากลับ
```

```
} //End of class
```

```
public class TestRealNumber {
    /* สร้างจำนวนจริง 1 จำนวนในรูปของอ็อบเจ็ค r ที่มีค่าเป็น 70 ประเภท RealNumber พร้อม
    คำนวณหาค่าผลบวกของตัวเลขนั้นกับ 15 และผลต่างของตัวเลขนั้นกับ 50 พร้อมทั้งแสดงผลลัพธ์
    ขึ้นบนจอภาพ*/

```

```
} //End of class
```

โจทย์ข้อที่ 12 [ระดับปานกลาง] จงสร้างคลาส Account ให้สมบูรณ์ ซึ่งเป็นคลาสที่ใช้ในการดำเนินการเกี่ยวกับระบบบัญชีธนาคาร โดยมีรายละเอียดดังต่อไปนี้ (10 คะแนน)

- 1) แอตทริบิวต์ชื่อ balance ซึ่งเป็น Object Variable เพื่อเก็บยอดเงินในบัญชี
- 2) คอนสตรัคเตอร์ 2 แบบเพื่อใช้ในการกำหนดค่าเริ่มต้นให้กับยอดเงินเมื่อเริ่มเปิดบัญชี
 - ถ้าไม่ได้ระบุจำนวนเงินเมื่อเริ่มต้นเปิดบัญชีให้ยอดเงินเป็น 0 บาท
 - ถ้ามีการระบุจำนวนเงินเมื่อเริ่มต้นเปิดบัญชีให้ยอดเงินเป็นไปตามค่าที่ระบุ
- 3) เมธอด deposit(...) ซึ่งเป็น Object Method เพื่อใช้ในการฝากเงินตามจำนวนเงินที่ระบุโดยไม่มีการคืนค่ากลับ
- 4) เมธอด withdraw(...) ซึ่งเป็น Object Method เพื่อใช้ในการถอนเงินตามจำนวนเงินที่ระบุและต้องตรวจสอบยอดเงินก่อนเสมอว่ามีเพียงพอในการถอนเงินหรือไม่ พร้อมทั้งคืนเงินที่ถอนด้วย
- 5) เมธอด getbalance(...) ซึ่งเป็น Object Method เพื่อคืนค่ายอดเงินคงเหลือในบัญชี

ให้สร้างคลาส `TestAccount` เพื่อเรียกใช้งานคลาส `Account` โดยให้เปิดบัญชีขึ้นมา 1 บัญชีพร้อมฝากเงินเริ่มต้นเปิดบัญชีจำนวน 2000 บาท หลังจากนั้นจึงทำการถอนเงิน 1500 บาท และฝากเพิ่มอีก 800 บาท ตามลำดับ (ในแต่ละกระบวนการให้แสดงยอดเงินในบัญชีหลังทำการรายการทุกครั้ง) (10 คะแนน)

โจทย์ข้อที่ 13 [ระดับปานกลาง] จงสร้างคลาส PiggyBank ให้สมบูรณ์เพื่อสร้างกระปุกออมสินลูกหมูซึ่งเก็บเฉพาะเหรียญ 1 บาท 2 บาท 5 บาท และ 10 บาท โดยมีรายละเอียดดังต่อไปนี้ (20 คะแนน)

```
public class PiggyBank {
```

- (1) แอตทริบิวต์ประจำอ็อบเจ็คแบบ public ชื่อ one, two, five และ ten ตามลำดับ เพื่อเก็บจำนวนของเหรียญแต่ละเหรียญ
- (2) แอตทริบิวต์ประจำอ็อบเจ็คแบบ public ชื่อ size เพื่อเก็บความจุของค่าเงินรวมสูงสุดที่สามารถเก็บได้ในกระปุก

- (3) ตัวสร้าง 2 แบบเพื่อกำหนดค่าเริ่มต้นของกระปุก

- 1) ถ้าไม่มีการกำหนดค่าใดๆ ให้แอตทริบิวต์ size เป็น 100 แอตทริบิวต์อื่นเป็น 0
- 2) ถ้ามีการกำหนดค่าให้กับแอตทริบิวต์ size ก็ให้ size เท่ากับค่าที่กำหนดมา ส่วนแอตทริบิวต์อื่นเป็น 0

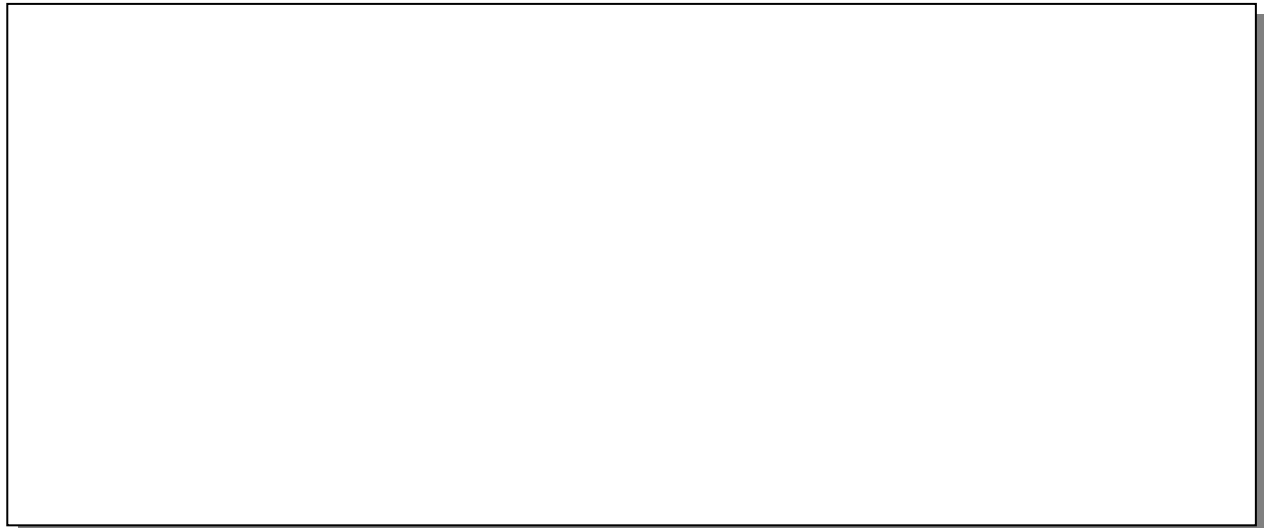
- (4) เมธอดประจำอ็อบเจ็คชื่อ clear(...) เพื่อทำการนำเหรียญออกจากกระปุกโดยไม่มีการคืนค่ากลับ (การนำเหรียญออกจากกระปุกคือให้จำนวนเหรียญทุกเหรียญเท่ากับ 0)

(5) เมท็อดประจำอ็อบเจ็คชื่อ `getTotal(...)` เพื่อคืนค่าเงินรวมทั้งหมดที่เก็บสะสมในกระปุก

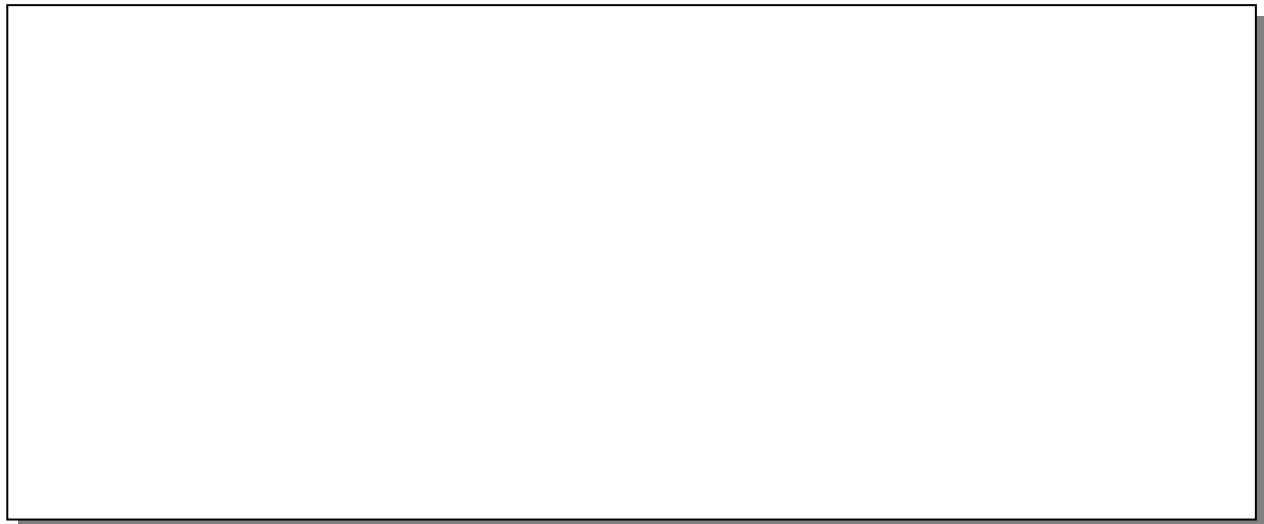
(6) เมท็อดประจำอ็อบเจ็คชื่อ `full(...)` เพื่อตรวจสอบว่ากระปุกเต็มหรือไม่ ถ้าเต็มจะคืนค่า `true` ถ้าไม่เต็มจะคืนค่า `false`

(7) เมท็อดประจำอ็อบเจ็คชื่อ `addOne(...)` เพื่อให้หยอดเหรียญ 1 บาท ตามจำนวนเหรียญที่ระบุลงในกระปุก โดยก่อนหยอดเหรียญจะต้องตรวจสอบก่อนว่ากระปุกออมสินเต็มหรือไม่

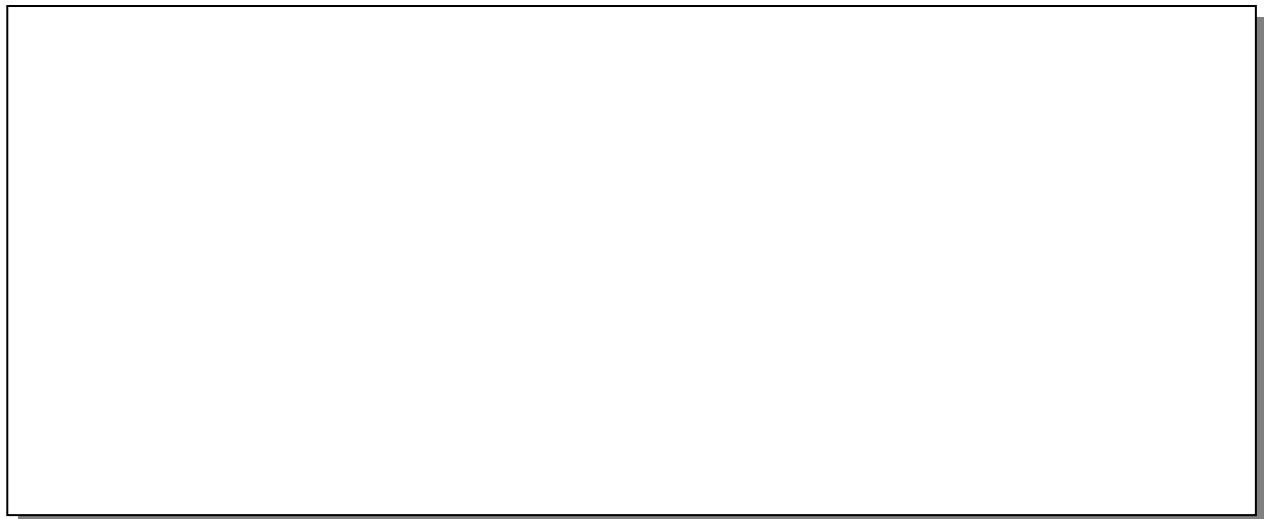
(8) เมท็อดประจำอ็อบเจ็คชื่อ `addTwo(...)` เพื่อให้หยอดเหรียญ 2 บาท ตามจำนวนเหรียญที่ระบุลงในกระปุก โดยก่อนหยอดเหรียญจะต้องตรวจสอบก่อนว่ากระปุกออมสินเต็มหรือไม่



(9) เมื่้อดประจำอับเจ็คชื่อ `addFive(...)` เพื่อใช้หยอดเหรียญ 5 บาท ตามจำนวนเหรียญที่ระบุลงในกระปุก โดยก่อนหยอดเหรียญจะต้องตรวจสอบก่อนว่ากระปุกออมสินเต็มหรือไม่



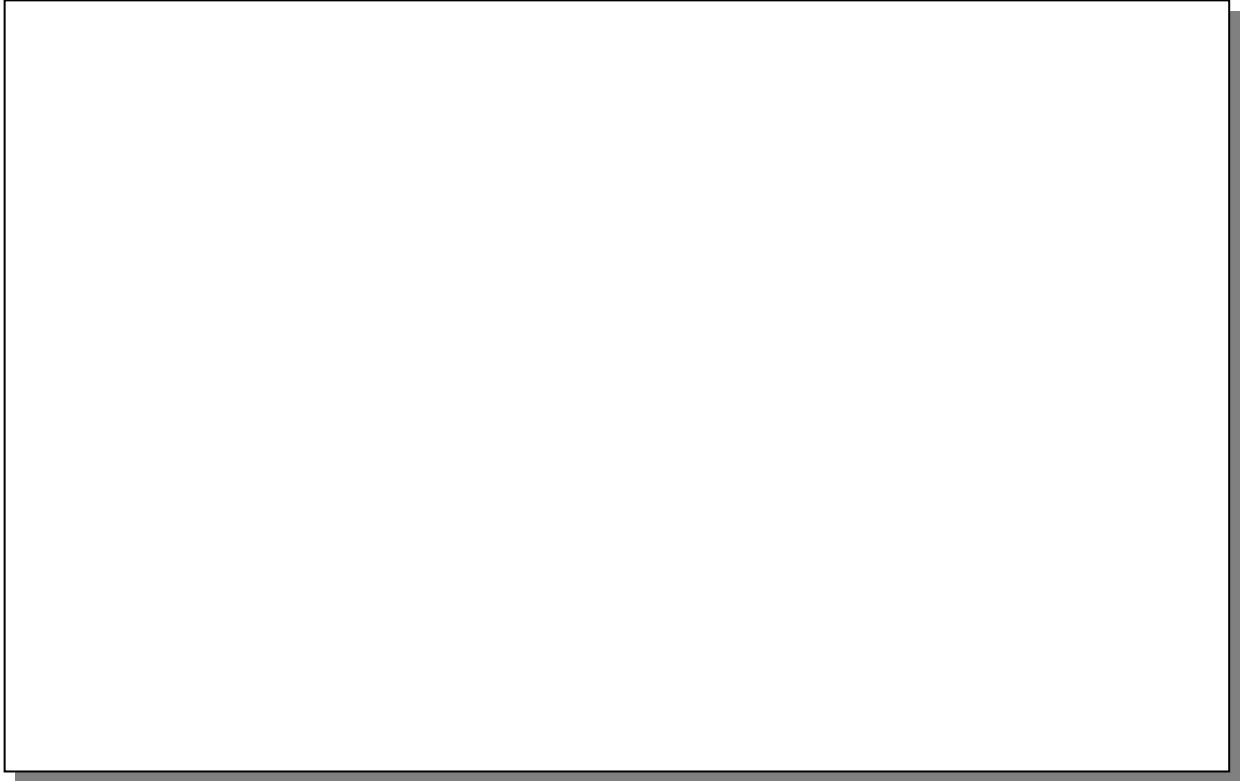
(10) เมื่้อดประจำอับเจ็คชื่อ `addTen(...)` เพื่อใช้หยอดเหรียญ 10 บาท ตามจำนวนเหรียญที่ระบุลงในกระปุก โดยก่อนหยอดเหรียญจะต้องตรวจสอบก่อนว่ากระปุกออมสินเต็มหรือไม่



`//End of class`

จงสร้างคลาส TestPiggyBank เพื่อเรียกใช้งานคลาส PiggyBank โดยสร้างกระปุกออมสินขนาดความจุ 500 บาท พร้อมทั้งมีการเก็บเหรียญสองบาท 34 เหรียญ และเหรียญสิบบาท 13 เหรียญ โดยในการเพิ่มเหรียญแต่ละครั้งให้แสดงค่าเงินรวมทั้งหมดในกระปุกให้เห็นทุกครั้ง

```
public class TestPiggyBank {  
    public static void main(String[] args) {
```



```
    } //End of main  
} //End of class
```