

CHAPTER

4

คลาสและอ็อบเจกต์ขั้นสูง (Advanced Classes and Objects)

1 หลักการของคลาสและอ็อบเจกต์เบื้องต้น

หลักการของคลาสและอ็อบเจกต์ที่เคยเรียนผ่านมามีบทที่ 2 จะมีการมีคลาส 2 แบบโดยมีรายละเอียดดังนี้

- 1) **คลาสต้นแบบ (Prototype Class)** ที่ประกอบไปด้วยองค์ประกอบ 3 ส่วนได้แก่ แอตทริบิวต์ (Attribute) คอนสตรัคเตอร์ (Constructor) และ เมทอด (Method)
- 2) **คลาสประมวลผล (Processing Class)** ที่ประกอบไปด้วย เมทอด main และภายในจะมีการ สร้างอ็อบเจกต์ (Object) เพื่อเรียกใช้งาน คลาสต้นแบบ ที่สร้างขึ้น

```

1 public class BigData {
2     •-----•
3     private int x;
4     private int y;
5     •-----•
6     public BigData(int a, int b) {
7         x = a; y = b;
8     }
9     •-----•
10    public int add(int n) { return x + n; }
11    public static int sum(int a, int b) {
12        return a + b;
13    }
14    •-----•
15 }

1 public class TestBigData {
2     public static void main(String[] args) {
3         BigData a = new BigData(1, 3);
4         BigData b = new BigData(6, 4);
5         BigData c = ...
6         System.out.println(a.add(20));
7         System.out.println(BigData.sum(5, 3));
8     }
9 }
  
```

2 การรับค่าและคืนค่าประเภทข้อมูลแบบคลาส

1. **คอนสตรัคเตอร์ที่รับพารามิเตอร์เป็นอ็อบเจกต์ (พารามิเตอร์ที่มีประเภทข้อมูลเป็นคลาส)**

คอนสตรัคเตอร์ในคลาสหนึ่งๆ สามารถมีได้หลายแบบซึ่ง Overload กัน แต่โดยทั่วไปนิยมเขียนกัน 3 แบบได้แก่

- 1) **คอนสตรัคเตอร์เริ่มต้น (Default Constructor)** เป็นคอนสตรัคเตอร์ที่ไม่รับพารามิเตอร์ใดๆ เลย [หมายเลข 1]
- 2) **คอนสตรัคเตอร์รายละเอียด (Detail Constructor)** เป็นคอนสตรัคเตอร์ที่รับจำนวนพารามิเตอร์เท่ากับจำนวนของแอตทริบิวต์ และมีประเภทข้อมูลของพารามิเตอร์แต่ละตัวสอดคล้องกับประเภทข้อมูลของแอตทริบิวต์ [หมายเลข 2]

3) **คอนสตรัคเตอร์คัดลอก (Copy Constructor)** เป็นคอนสตรัคเตอร์ที่รับพารามิเตอร์เป็นอ็อบเจ็กต์ของคลาสต้นแบบใดๆ (อ็อบเจ็กต์จากคลาสตัวเองหรือคลาสอื่น) [หมายเลข 3]

```

1 public class Data {
2     public int x;
3     public Data() {
4         1 x = 0;
5     }
6     public Data(int n) {
7         2 x = n;
8     }
9     public Data(Data d) {
10        3 x = d.x;
11    }
12    ...
13 }

```

รับเพียงแค่ `int` หนึ่งค่าในชื่อ `n` เข้ามาทางพารามิเตอร์

รับเข้ามาทั้งก้อนของอ็อบเจ็กต์ `d` แต่จะสนใจเพียงแค่แอตทริบิวต์ `x` ในอ็อบเจ็กต์ `d` เท่านั้น โดยจะเอาค่า `x` ที่อยู่ในอ็อบเจ็กต์ `d` หรือค่า `d.x` เข้าไปเก็บไว้ในแอตทริบิวต์ `x` ของคลาสต้นแบบ `Data`

2. เมธอดที่รับพารามิเตอร์เป็นอ็อบเจ็กต์ (พารามิเตอร์ที่มีประเภทข้อมูลเป็นคลาส)

เมธอดสามารถรับพารามิเตอร์ได้ทั้งประเภทข้อมูลแบบดั้งเดิม (Primitive Data Type) เช่น `int`, `double`, `boolean` ดังเมธอดหมายเลข 1 และประเภทข้อมูลแบบคลาสหรือพารามิเตอร์เป็นอ็อบเจ็กต์ ดังเมธอดหมายเลข 2

```

1 public class Data {
2     public int x;
3     ...
4     public boolean isEq1(int n) {
5         1 return x == n;
6     }
7     public boolean isEq2(Data d) {
8         2 return x == d.x;
9     }
10 }

```

รับเพียงแค่ `int` หนึ่งค่าในชื่อ `n` เข้ามาทางพารามิเตอร์

รับทั้งก้อนอ็อบเจ็กต์ชนิด `Data` หนึ่งก้อนในชื่อ `d` เข้ามาทางพารามิเตอร์ (แต่จะใช้งานเพียงแค่ค่าแอตทริบิวต์ `x` ของอ็อบเจ็กต์ `d` เท่านั้น)

3. เมธอดที่คืนค่าเป็นอ็อบเจ็กต์ (มีประเภทข้อมูลเป็นคลาส)

เมธอดสามารถคืนค่าข้อมูลที่มีประเภทข้อมูลแบบดั้งเดิม (Primitive Data Type) ได้ เช่น `int`, `double`, `boolean` ดังเมธอดหมายเลข 1 และคืนค่าประเภทข้อมูลแบบคลาสหรือคืนค่าเป็นอ็อบเจ็กต์ได้ ดังตัวอย่างเมธอดหมายเลข 2

```

1 public class Data {
2     public int x;
3     ...
4     public int getX() {
5         1 return x;
6     }
7     public Data getData() {
8         2 Data d = new Data(x);
9         return d;
10    }
11 }

```

คืนค่าข้อมูลประเภท `int`

คืนค่าข้อมูลในรูปของอ็อบเจ็กต์ `d` ซึ่งเปรียบเสมือนเป็นการคืนค่าข้อมูล `d.x` ออกไป (คืนออกทั้งก้อน)

4. คอนสตรัคเตอร์และเมธอดที่รับพารามิเตอร์ที่มีชื่อเหมือนกับแอตทริบิวต์

คอนสตรัคเตอร์หรือเมธอดใดก็ตามที่รับค่าพารามิเตอร์ที่มีชื่อเหมือนกับแอตทริบิวต์ของคลาสนั้น จะต้องใช้คีย์เวิร์ด `this` ระบุหน้าแอตทริบิวต์ดังนี้ `this.<แอตทริบิวต์> = <พารามิเตอร์>` ทั้งนี้เพื่อไม่ให้เกิดความสับสนดังตัวอย่างต่อไปนี้

```

1 public class Data {
2     public int x;
3     ...
4     public Data(int x) {
5         x = x; //Error
6     }
7     ...
8 }

```

เกิดความสับสนขึ้นว่า x ตัวใดคือแอตทริบิวต์ และ x ตัวใดคือพารามิเตอร์

```

1 public class Data {
2     public int x;
3     ...
4     public Data(int x) {
5         this.x = x;
6     }
7     ...
8 }

```

ก็เวิร์ด this สามารถใช้ได้กับแอตทริบิวต์เท่านั้น

โจทย์ข้อที่ 1 [ระดับปานกลาง] จงพิจารณาคำสั่งการสร้างและเรียกใช้งานอ็อบเจกต์จากคลาส `PairNumber` ต่อไปนี้เพื่อระบุว่าคำสั่งดังกล่าวมีการประมวลผล ณ ส่วนหมายเลขใดในคลาส `PairNumber` (20 คะแนน)

```

1 public class PairNumber {
2     public int x, y;
3     1 public PairNumber() { x = y = 0; }
4     2 public PairNumber(int x, int y) { this.x = x; this.y = y; }
5     3 public PairNumber(PairNumber p) { this.x = p.x; this.y = p.y; }
6     public boolean isEqual(int x, int y) {
7     4         return this.x == x && this.y == y;
8     }
9     public boolean isEqual(PairNumber p) {
10    5         return x == p.x && y == p.y;
11    }
12    public static boolean compareTo(PairNumber p, PairNumber q) {
13    6         return p.x == q.x && p.y == q.y;
14    }
15    public static PairNumber downPair(PairNumber p) {
16    7         return new PairNumber(p.x - 1, p.y - 1);
17    }
18 }

```

กำหนดให้คำสั่งแต่ละข้อตั้งแต่ข้อ 1-10 ทำงานต่อเนื่องกัน

ข้อ	คำสั่งในการสร้างหรือเรียกใช้งานอ็อบเจกต์	ระบุหมายเลข	ผลลัพธ์ที่ได้
1.	<code>PairNumber a = new PairNumber();</code> <code>System.out.println(a.x + "," + a.y);</code>		
2.	<code>PairNumber b = new PairNumber(13, 2);</code> <code>System.out.println(b.x + "," + b.y);</code>		
3.	<code>PairNumber c = new PairNumber(b);</code> <code>System.out.println(c.x + "," + c.y);</code>		
4.	<code>PairNumber d = new PairNumber(new PairNumber(7, 11));</code> <code>System.out.println(d.x + "," + d.y);</code>		
5.	<code>b = new PairNumber(new PairNumber());</code> <code>System.out.println(b.x + "," + b.y);</code>		
6.	<code>boolean flag1 = b.isEqual(new PairNumber(c));</code> <code>System.out.println(flag1);</code>		
7.	<code>boolean flag2 = a.isEqual(b.x, b.y);</code> <code>System.out.println(flag2);</code>		
8.	<code>PairNumber e = PairNumber.downPair(new PairNumber());</code> <code>System.out.println(e.x + "," + e.y);</code>		
9.	<code>boolean flag3 = PairNumber.compareTo(c, d);</code> <code>System.out.println(flag3);</code>		
10.	<code>System.out.print(PairNumber.compareTo(a, e));</code> <code>System.out.print(a.isEqual(e));</code>		

โจทย์ข้อที่ 2 [ระดับปานกลาง] จงพิจารณาคลาส Rectangle ต่อไปนี้ ซึ่งเป็นต้นแบบของรูปสี่เหลี่ยมมุมฉากใด ๆ เพื่อใช้ในการสร้างเมทอดตามเงื่อนไขที่กำหนดให้ (25 คะแนน)

คลาส Rectangle

- 1) ให้แอตทริบิวต์ width และ height แทนความกว้างและความยาวของสี่เหลี่ยม และมีตัวสร้าง 2 แบบ

```
private double width, height;
public Rectangle(double x, double y) { width = x; height = y; }
public Rectangle(Rectangle r) { width = r.width; height = r.height; }
```
- 2) เมทอดประจำอ็อบเจ็คชื่อ isEqual(...) ที่รับพารามิเตอร์เป็นอ็อบเจ็คชนิด Rectangle ซึ่งแทนสี่เหลี่ยมใด ๆ เพื่อใช้ตรวจสอบการเท่ากันของความกว้างและความสูงของสี่เหลี่ยม (5 คะแนน)
- 3) เมทอดประจำอ็อบเจ็คชื่อ differentArea(...) ที่รับพารามิเตอร์เป็นอ็อบเจ็คชนิด Rectangle เพื่อหาผลต่างของพื้นที่ของรูปสี่เหลี่ยมรูปหนึ่งกับสี่เหลี่ยมของคลาสนี้ (5 คะแนน)
- 4) เมทอดประจำคลาสนี้ชื่อ isAreaEqual(...) ที่รับพารามิเตอร์ 2 ตัวที่เป็นอ็อบเจ็คชนิด Rectangle ซึ่งแทนสี่เหลี่ยม 2 รูปใด ๆ เพื่อใช้ตรวจสอบการเท่ากันของพื้นที่ของสี่เหลี่ยม (5 คะแนน)
- 5) เมทอดประจำคลาสนี้ชื่อ transformToSquare(...) ที่รับพารามิเตอร์ 1 ตัวชนิด Rectangle เพื่อทำการสร้างและคืนค่าสี่เหลี่ยมใหม่ที่เป็นสี่เหลี่ยมจัตุรัส โดยที่มีพื้นที่เท่ากับสี่เหลี่ยมเดิม (5 คะแนน)

คลาส TestRectangle

- 1) สร้างสี่เหลี่ยม 2 รูปได้แก่ a ขนาด 8 x 8 และ b ขนาด 6 x 11 (1 คะแนน)
- 2) แสดงค่าการเปรียบเทียบความเท่ากันของสี่เหลี่ยมทั้งสอง (1 คะแนน)
- 3) แสดงค่าการเปรียบเทียบความเท่ากันของขนาดพื้นที่ของสี่เหลี่ยมทั้งสอง (1 คะแนน)
- 4) แสดงค่าความแตกต่างของขนาดพื้นที่ของสี่เหลี่ยมทั้งสอง (1 คะแนน)
- 5) เปลี่ยนสี่เหลี่ยม b ให้เป็นสี่เหลี่ยมจัตุรัส (1 คะแนน)

โจทย์ข้อที่ 3 [ระดับยาก] จงเขียนโปรแกรมจาวาให้สมบูรณ์เพื่อสร้างคลาส Circle ซึ่งเป็นตัวแทนของวงกลมหนึ่งวง และสร้างคลาส TestCircle เพื่อใช้สำหรับเรียกใช้งานเมทอดต่าง ๆ ของคลาส Circle โดยให้สร้างเมทอดและแอตทริบิวต์ตามรายละเอียดที่กำหนดให้ต่อไปนี้ (30 คะแนน)

คลาส Circle

- 1) กำหนดให้แอตทริบิวต์ x และ y เก็บค่าพิกัดจุดศูนย์กลางวงกลม และแอตทริบิวต์ r เก็บค่ารัศมีวงกลม และกำหนดให้ตัวสร้างเริ่มต้น (Default Constructor) และตัวสร้างรายละเอียด (Detail Constructor) ดังนี้

```
private double x, y, r;
public Circle() {
    this.x = this.y = this.r = 0.0;
}
public Circle(double x, double y, double r) {
    this.x = x; this.y = y; this.r = r;
}
```

- 2) ตัวสร้างคัดลอก (Copy Constructor) ที่รับพารามิเตอร์เป็นอ็อบเจกต์ชนิด Circle (4 คะแนน)
- 3) เมทอดประจำอ็อบเจกต์ชื่อ distanceTo(...) ที่รับพารามิเตอร์เป็นอ็อบเจกต์ของวงกลมใด ๆ หนึ่งวง เพื่อใช้หา ระยะห่างระหว่างจุดศูนย์กลางของวงกลมที่รับเข้ามาทางพารามิเตอร์กับวงกลมนี้ พร้อมทั้งคืนค่ากลับ (4 คะแนน)
- 4) เมทอดประจำอ็อบเจกต์ชื่อ isInCircle(...) ที่รับพารามิเตอร์เป็นอ็อบเจกต์ของวงกลมใด ๆ หนึ่งวง เพื่อใช้ตรวจสอบว่าวงกลมที่รับมาทางพารามิเตอร์รั้งนั้นอยู่ภายในวงกลมนี้หรือไม่ พร้อมทั้งคืนค่ากลับ (4 คะแนน)
- 5) เมทอดประจำอ็อบเจกต์ชื่อ isIntersected(...) ที่รับพารามิเตอร์เป็นอ็อบเจกต์ของวงกลมใด ๆ หนึ่งวง เพื่อใช้ตรวจสอบว่าวงกลมที่รับมาทางพารามิเตอร์รั้งนั้นมีพื้นที่ซ้อนทับส่วนใดส่วนหนึ่งกับวงกลมนี้หรือไม่ พร้อมทั้งคืนค่ากลับ (4 คะแนน)
- 6) เมทอดประจำอ็อบเจกต์ชื่อ equals(...) ที่รับพารามิเตอร์เป็นอ็อบเจกต์ของวงกลมใด ๆ หนึ่งวง เพื่อใช้ตรวจสอบว่าวงกลมที่รับมาทางพารามิเตอร์รั้งนั้นเป็นวงกลมเดียวกับวงกลมนี้หรือไม่ (จุดศูนย์กลางเดียวกันและรัศมียาวเท่ากัน) พร้อมทั้งคืนค่ากลับ (4 คะแนน)

คลาส TestCircle

- 1) สร้างวงกลม a (0.0, 0.0, 10.0), b (-1.0, 2.0, 5.0), c (6.0, -4.0, 15.0) และ d (เหมือน c) (2 คะแนน)
- 2) แสดงระยะห่างระหว่างจุดศูนย์กลางของวงกลม a และ b ขึ้นบนจอภาพ (2 คะแนน)
- 3) แสดงคำตอบเพื่อตรวจสอบว่าวงกลม d อยู่ภายในวงกลม b หรือไม่ (true หรือ false) (2 คะแนน)
- 4) แสดงคำตอบเพื่อตรวจสอบว่าวงกลม c ซ้อนทับกับวงกลม a หรือไม่ (true หรือ false) (2 คะแนน)
- 5) แสดงคำตอบเพื่อตรวจสอบว่าวงกลม d และวงกลม c เท่ากันหรือไม่ (true หรือ false) (2 คะแนน)

3 การประกาศและสร้างอาร์เรย์ของอ็อบเจ็กต์

1. วิธีการประกาศและสร้างอาร์เรย์ของอ็อบเจ็กต์ คล้ายกับการประกาศและสร้างอาร์เรย์แบบทั่วไป ซึ่งมีขั้นตอนดังนี้

1) การประกาศและสร้างอาร์เรย์ของอ็อบเจ็กต์แบบ 1 มิติ

```
<ชื่อคลาส> <ชื่ออ็อบเจ็กต์>[] = new <ชื่อคลาส>[ขนาด];
```

2) การประกาศและสร้างอาร์เรย์ของอ็อบเจ็กต์แบบ 2 มิติ

```
<ชื่อคลาส> <ชื่ออ็อบเจ็กต์>[][] = new <ชื่อคลาส>[ขนาดแถว][ขนาดหลัก];
```

2. ตัวอย่างการประกาศ สร้าง และใช้งานอาร์เรย์ของอ็อบเจ็กต์

ตัวอย่างการประกาศและสร้างอ็อบเจ็กต์แบบทั่วไป (ถ้าต้องการอ็อบเจ็กต์หลายอ็อบเจ็กต์ก็ต้อง new หลายครั้ง)

<pre> 1 public class Data { 2 public int x; 3 public Data() ... 4 public Data(int n) ... 5 public Data(Data d) ... 6 } </pre>	<pre> 1 public class TestData { 2 public static void main(String [] ar) { 3 Data d = new Data(); 4 Data e = new Data(13); 5 } 6 } </pre>
---	--

e
x = 13

d
x = 0

ตัวอย่างการประกาศและสร้างอาร์เรย์ของอ็อบเจ็กต์

<pre> 1 public class TestData2 { 2 public static void main(String [] ar) { 3 Data d[] = new Data[8]; 4 for (int i = 0; i < d.length; i++) { 5 d[i] = new Data(9); 6 } 7 d[2].x = 15; 8 d[4].x = 30; 9 } 10 } </pre>	<p>ประกาศอ็อบเจ็กต์ 8 อ็อบเจ็กต์ในรูปแบบของอาร์เรย์ (โดยแต่ละอ็อบเจ็กต์จะอยู่ในอาร์เรย์แต่ละช่อง)</p> <p>จะต้อง new สองครั้งคือ (1) new อาร์เรย์ (2) new อ็อบเจ็กต์ที่อยู่ในแต่ละช่องของอาร์เรย์ (โดยการวน for)</p>
--	---

d

d[0]
x = 9

d[1]
x = 9

d[2]
x = 15

d[3]
x = 9

d[4]
x = 30

d[5]
x = 9

d[6]
x = 9

d[7]
x = 9

โจทย์ข้อที่ 4 [ระดับปานกลาง] จงเขียนส่วนของโปรแกรมในคลาส MakeBubble ให้สมบูรณ์ เพื่อใช้สำหรับสร้างลูกโป่งซึ่งมาจากคลาส Bubble โดยมีรายละเอียดดังต่อไปนี้ (20 คะแนน)

กำหนดให้คลาส Bubble มีรายละเอียดดังนี้

<pre> 1 public class Bubble { 2 private int size; 3 public Bubble(int s) { 4 size = s; 5 } 6 public void setSize(int s) { 7 size = s; 8 } 9 } </pre>	<p>size เป็นขนาดของลูกโป่ง ซึ่งประกอบด้วยขนาดเล็ก (size = 1) ขนาดกลาง (size = 2) และขนาดใหญ่ (size = 3)</p> <p>setSize(...) ใช้ในการเปลี่ยนขนาดของลูกโป่ง</p>
--	---

คลาส MakeBubble

- 1) สร้างลูกโป่งที่มีขนาดกลาง (size = 2) จำนวน 100 ลูก (สร้างอาร์เรย์ 1 มิติชื่อ x ที่มีชนิดเป็น Bubble ขนาด 100 ช่อง) (5 คะแนน)
- 2) เพิ่มขนาดของลูกโป่ง 30 ลูกแรกให้เป็นขนาดใหญ่ (size = 3) (5 คะแนน)
- 3) สร้างลูกโป่งที่มีขนาดเล็ก (size = 1) วางเรียงกันเป็นแถว 5 แถว โดยแต่ละแถวมีลูกโป่งจำนวน 20 ลูก (สร้างอาร์เรย์ 2 มิติชื่อ y ที่มีชนิดเป็น Bubble ขนาด 5 x 20) (5 คะแนน)
- 4) เพิ่มขนาดของลูกโป่งแถวที่ 2 ทุกลูกให้เป็นขนาดกลาง (size = 2) (5 คะแนน)

โจทย์ข้อที่ 5 [ระดับยาก] จงเขียนคลาส `Freshmen` ให้สมบูรณ์เพื่อใช้สร้างนิสิตชั้นปีที่ 1 จำนวน 800 คน โดยนิสิตแต่ละคนจะสร้างมาจากคลาส `Student` ซึ่งมีรายละเอียดในการประมวลผลดังต่อไปนี้ (30 คะแนน)

กำหนดให้คลาส `Student` มีรายละเอียดดังนี้

```

1 public class Student {
2     private int id;
3     private String name;
4     private double score[];
5     public Student(int i, String n, int x) {
6         id = i;
7         name = n;
8         score = new double[x];
9     }
10    public Student(Student s) {
11        id = s.id;
12        name = s.name;
13        setScore(s.score);
14    }
15    public int getID() {
16        return id;
17    }
18    public String getName() {
19        return name;
20    }
21    public double[] getScore() {
22        return score;
23    }
24    public void setScore(double scr[]) {
25        score = new double[scr.length];
26        for (int i = 0; i < scr.length; i++)
27            score[i] = scr[i];
28    }
29    public String toString() {
30        String s = id + "\t" + name + "\t";
31        for (int i = 0; i < score.length; i++)
32            s += score[i] + "\t";
33        return s;
34    }
35 }

```

แอตทริบิวต์ id เก็บเลขประจำตัวนิสิต

แอตทริบิวต์ name เก็บชื่อนิสิต

แอตทริบิวต์ score เก็บคะแนนทุกวิชาที่นิสิตลงทะเบียน

Student (เลขประจำตัว, ชื่อ, จำนวนวิชาที่ลงทะเบียน)

คืนค่าเลขประจำตัวของนิสิต

คืนค่าชื่อของนิสิต

คืนค่าคะแนนทุกวิชาของนิสิต

กำหนดค่าคะแนนในแต่ละวิชาให้กับนิสิต

คืนค่าข้อมูลนิสิตในรูปแบบของสตริง

คลาส Freshmen

- 1) กำหนดให้มีนิสิตทั้งหมด 800 คน ที่สร้างจากคลาส student (อาเรียซ์ของอ็อบเจ็ค) (2 คะแนน)
- 2) กำหนดค่าเริ่มต้นให้กับเลขประจำตัวของนิสิตของแต่ละคนแบบอัตโนมัติ โดยเรียงลำดับตั้งแต่ 60001 จนถึง 60800 และกำหนดค่าเริ่มต้นให้กับชื่อนิสิตของแต่ละคนโดยการวนรับค่าจากแป้น พิมพ์ทีละชื่อ และกำหนดจำนวนวิชาที่ลงทะเบียนของนิสิตเป็น 4 วิชาทุกคน (8 คะแนน)
- 3) เรียกใช้งานเมทอด `setScore(...)` ผ่านทางอ็อบเจ็คของนิสิตแต่ละคน เพื่อสุ่มคะแนนทั้ง 4 วิชาให้กับนิสิตแต่ละคน พร้อมทั้งแสดงข้อมูลของนิสิตแต่ละคนออกทางจอภาพ (10 คะแนน)
- 4) คำนวณหาคะแนนเฉลี่ยของแต่ละวิชา โดยการเอาคะแนนแต่ละวิชาของนิสิตทุกคนมารวมกันแล้วหารด้วยจำนวนนิสิตทั้งหมด พร้อมทั้งแสดงผลลัพธ์ออกทางจอภาพ (10 คะแนน)

โจทย์ข้อที่ 6 [ระดับยาก] จงเขียนคลาส `Noob` ให้สมบูรณ์เพื่อใช้สร้างต้นแบบของเกรียนหนึ่งคน โดยประกอบไปด้วยรายละเอียดดังต่อไปนี้ (25 คะแนน)

- 1) กำหนดให้แอตทริบิวต์ `name` และ `level` แทนชื่อของเกรียนและระดับความเกรียนของตามลำดับ และกำหนดให้ตัวสร้างเริ่มต้น (Default Constructor) ตัวสร้างรายละเอียด (Detail Constructor) และตัวสร้างคัดลอก (Copy Constructor) เป็นดังนี้

```
public String name;
public int level;
public Noob() { name = ""; level = 0; }
public Noob(String a, int b) { name = a; level = b; }
public Noob(Noob n) { name = n.name; level = n.level; }
```

- 2) เมทอดประจำคลาสชื่อ `findNoob(Noob n[], String s)` เพื่อค้นหาเกรียนตามชื่อที่ระบุจากกลุ่มเกรียน ถ้าค้นเจอให้คืนค่าเกรียนคนนั้น แต่ถ้าไม่เจอให้คืนค่า Default ของเกรียน (10 คะแนน)
- 3) เมทอดประจำคลาสชื่อ `findGroupOfNoob(Noob n[], int x)` เพื่อค้นหาเกรียนที่มีระดับความเกรียนตามที่ระบุ พร้อมทั้งคืนค่าเกรียนทุกคนที่ค้นเจอ (15 คะแนน)

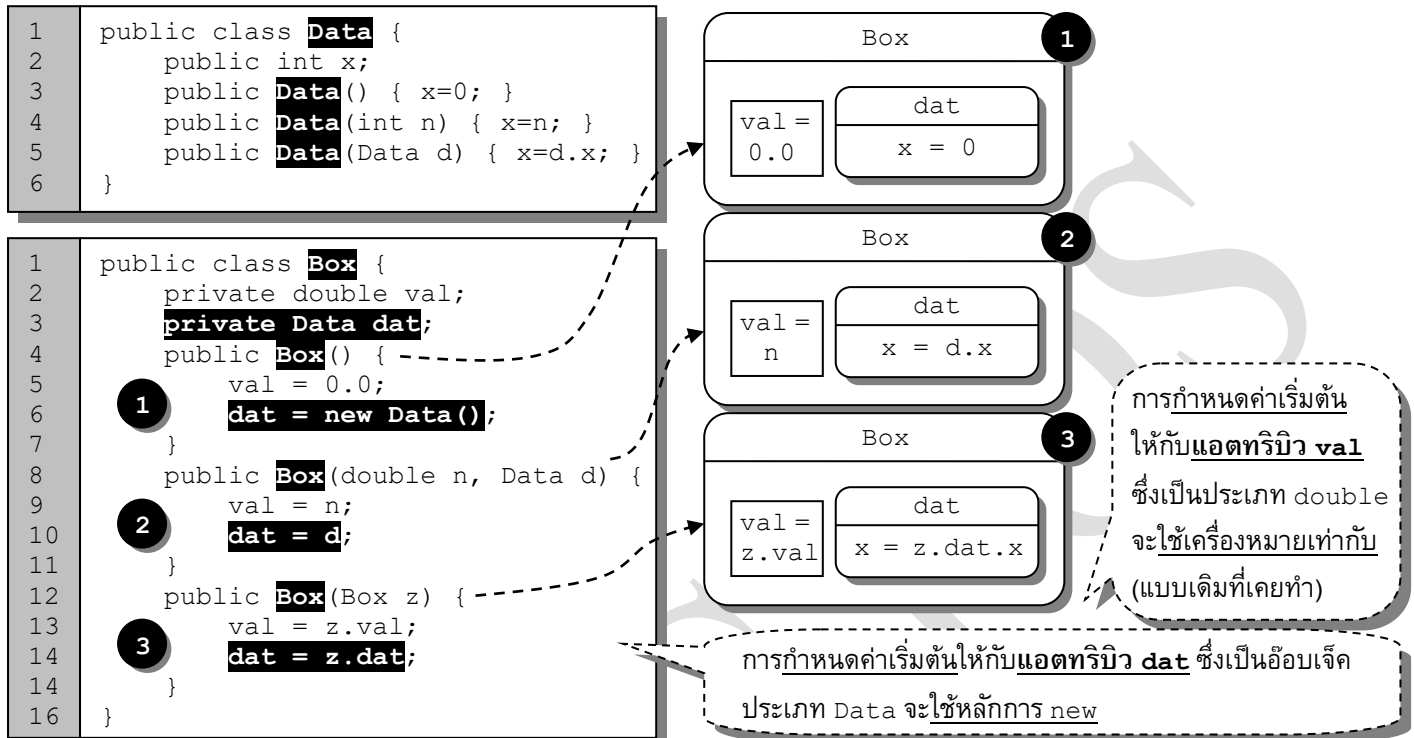
จากคลาส `Noob` จงสร้างคลาส `TestNoob` เพื่อประมวลผลคลาส `Noob` และเรียกใช้เมทอดต่าง ๆ จากคลาส `Noob` ให้ครบทุกเมทอด

4

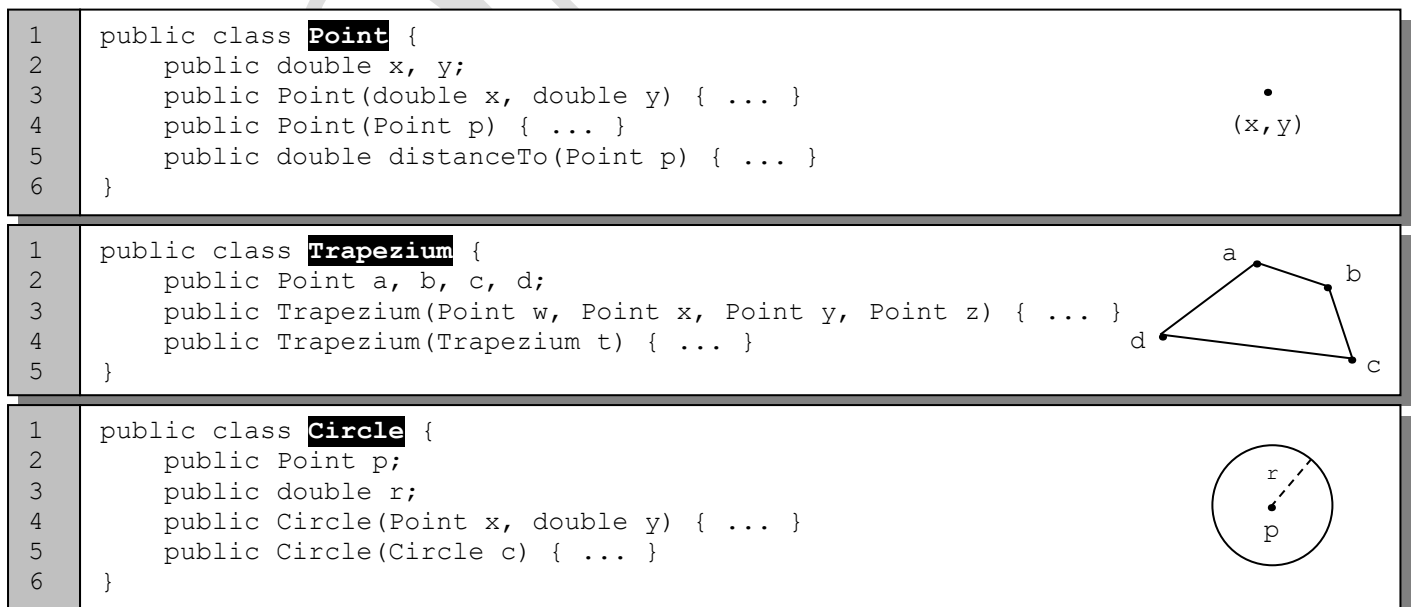
การประกาศและสร้างแอตทริบิวต์ที่มีประเภทข้อมูลแบบคลาส

1. วิธีการประกาศและสร้างแอตทริบิวต์ที่เป็นอ็อบเจกต์ (มีประเภทข้อมูลเป็นคลาส)

เราสามารถสร้างแอตทริบิวต์ที่เป็นอ็อบเจกต์ของคลาสอื่นได้ โดยกระทำเหมือนการประกาศแอตทริบิวต์ทั่วไป ดังตัวอย่าง



โจทย์ข้อที่ 7 [ระดับยาก] จงศึกษาคลาส Point ที่เป็นตัวแทนของจุด คลาส Trapezium ที่เป็นตัวแทนของสี่เหลี่ยมด้านไม่เท่า และคลาส Circle ที่เป็นตัวแทนของวงกลม เพื่อนำมาใช้ในการสร้างเมทอดต่าง ๆ ตามที่โจทย์ต้องการ (20 คะแนน)



1) เขียนเมทอดประจำคลาสชื่อ `getTrapeziumBorder` ที่รับพารามิเตอร์เป็นอ็อบเจกต์ของคลาส `Trapezium` เพื่อคืนค่าความยาวรอบรูปสี่เหลี่ยม (10 คะแนน)

- 2) เขียนเมทอดประจำคลาสชื่อ `isTrapeziumInCircle` ที่รับอ็อบเจ็คของคลาส `Trapezium` และอ็อบเจ็คของคลาส `Circle` เพื่อใช้ตรวจสอบว่าสี่เหลี่ยมนี้้อยู่ภายในวงกลมหรือไม่ ซึ่งสี่เหลี่ยมจะอยู่ภายในวงกลมได้ก็ต่อเมื่อจุดมุมทั้ง 4 ของสี่เหลี่ยมต้องอยู่ภายในวงกลม (10 คะแนน)

โจทย์ข้อที่ 8 [ระดับยาก] จงเขียนคลาส `Book` ให้สมบูรณ์เพื่อใช้สร้างต้นแบบของหนังสือหนึ่งเล่มใด ๆ โดยประกอบไปด้วยรายละเอียดดังต่อไปนี้ (40 คะแนน)

```

1 public class Author {
2     public String name;
3     public int age;
4     public Author() { ... }
5     public Author(String n, int a) { ... }
6 }

```

คลาส Book

- กำหนดให้แอตทริบิวต์ `title`, `author` และ `price` แทนชื่อหนังสือ ชื่อผู้แต่ง และราคาตามลำดับ และตัวสร้างรายละเอียด (Detail Constructor) และตัวสร้างคัดลอก (Copy Constructor) เป็นดังนี้

```

public String title;
public Author[] author;
public double price;
public Book(String t, Author a[], double p) { ... }
public Book(Book b) { ... }

```
- เมทอดประจำคลาสชื่อ `countBookByAuthorName(Book b[], String n)` เพื่อนับจำนวนหนังสือทั้งหมดที่แต่งโดยผู้แต่งตามชื่อที่ระบุ (10 คะแนน)
- เมทอดประจำคลาสชื่อ `getFirstAuthorByBookTitle(Book b[], String t)` เพื่อคืนค่าข้อมูลของผู้แต่งคนที่หนึ่งที่แต่งหนังสือตามชื่อหนังสือที่ระบุ ถ้าไม่พบหนังสือให้คืน Default ผู้แต่ง (10 คะแนน)
- เมทอดประจำคลาสชื่อ `getBookWhichManyAuthor(Book b[])` เพื่อคืนค่าหนังสือทุกเล่มที่มีจำนวนผู้แต่งตั้งแต่ 3 คนขึ้นไป (10 คะแนน)
- เมทอดประจำคลาสชื่อ `showBookByAuthor(Book b[], Author a[])` เพื่อแสดงชื่อหนังสือทุกเล่มพร้อมด้วยราคาตามรายการรายชื่อผู้แต่งแต่ละคน (ดูตัวอย่างผลลัพธ์) (10 คะแนน)

Penporn Rukjung
Wongyos Keardsri
Apimahakarn Tumrong

Basic Java 200
Green Java 300
The ...

Programming with C++ 225
Green C++ 299 Green PHP 250

โจทย์ข้อที่ 9 [ระดับยาก] จงสร้างคลาส Room และคลาส Hotel ตามรายละเอียดดังต่อไปนี้ (55 คะแนน)

คลาส Room

ทำหน้าที่เป็นคลาสต้นแบบของห้องพักหนึ่งห้องใด ๆ ของโรงแรม โดยมีรายละเอียดดังต่อไปนี้

- 1) แอตทริบิวต์ประจำอ็อบเจกต์ชนิดสตริงชื่อ code เพื่อเก็บเบอร์/รหัส/หมายเลขของห้องพัก (1 คะแนน)
- 2) แอตทริบิวต์ประจำอ็อบเจกต์ชนิดจำนวนจริงชื่อ price เพื่อเก็บราคาของห้องพัก (1 คะแนน)
- 3) แอตทริบิวต์ประจำอ็อบเจกต์ชนิดจำนวนเต็มชื่อ status เพื่อเก็บสถานะของห้องพัก (ห้องว่างแทนด้วย 0 ห้องถูกจองแทนด้วย 1 และห้องมีคนพักแทนด้วย 1) (1 คะแนน)
- 4) คอนสตรัคเตอร์เริ่มต้น (Default Constructor) เพื่อกำหนดค่าเริ่มต้นให้กับแอตทริบิวต์ทุกตัว (4 คะแนน)
- 5) คอนสตรัคเตอร์รายละเอียด (Detail Constructor) เพื่อกำหนดค่าเริ่มต้นให้กับแอตทริบิวต์ทุกตัว (4 คะแนน)
- 6) คอนสตรัคเตอร์คัดลอก (Copy Constructor) เพื่อกำหนดค่าเริ่มต้นให้กับแอตทริบิวต์ทุกตัว (4 คะแนน)
- 7) เมธอดประจำอ็อบเจกต์ชื่อ getCode (...) เพื่อคืนค่าหมายเลขของห้องพัก (4 คะแนน)
- 8) เมธอดประจำอ็อบเจกต์ชื่อ getPrice (...) เพื่อคืนค่าราคาของห้องพัก (4 คะแนน)
- 9) เมธอดประจำอ็อบเจกต์ชื่อ getStatus (...) เพื่อคืนค่าสถานะของห้องพัก (4 คะแนน)
- 10) เมธอดประจำอ็อบเจกต์ชื่อ setStatus (...) ที่รับจำนวนเต็ม 1 ค่าเข้ามาทางพารามิเตอร์เพื่อใช้กำหนดค่าให้กับสถานะของห้องพัก (4 คะแนน)
- 11) เมธอดประจำอ็อบเจกต์ชื่อ toString (...) คืนค่าข้อมูลของห้องพัก (ประกอบไปด้วย เลขห้อง ราคา สถานะ) ในรูปแบบของสตริง (4 คะแนน)

คลาส Hotel

ทำหน้าที่เป็นคลาสประมวลผล โดยคลาสนี้จะเป็นตัวแทนของการสร้างโรงแรมใด ๆ หนึ่งแห่ง ซึ่งประกอบไปด้วยรายละเอียดดังต่อไปนี้ (30 คะแนน)

- 1) เมธอด main (...) เพื่อประมวลผลดังนี้ (20 คะแนน)
 - (1) สร้างโรงแรมที่เป็นอาเรียชนิด Room ประกอบไปด้วย 10 ชั้น โดยแต่ละห้องมีชั้นละ 8 ห้อง
 - (2) กำหนดให้ชั้นที่ 1-5 มีหมายเลขห้องเป็น A1-A8, B1-B8, C1-C8, D1-D8, E1-E8 ตามลำดับ มีราคาห้องเป็น 1000 บาทและมีสถานะเป็นห้องว่าง
 - (3) กำหนดให้ชั้นที่ 6-8 มีหมายเลขห้องเป็น F1-F8, G1-G8, H1-H8 ตามลำดับ มีราคาห้องเป็น 1500 บาทและมีสถานะเป็นห้องว่าง
 - (4) กำหนดให้ชั้นที่ 9-10 มีหมายเลขห้องเป็น I1-I8, J1-J8 ตามลำดับ มีราคาห้องเป็น 2000 บาทและมีสถานะเป็นห้องว่าง

(5) ทำการสร้างระบบบริหารจัดการโรงแรมโดยระบบจะวน Loop ไปเรื่อย ๆ เพื่อรับค่าหมายเลขเมนูที่เป็นจำนวนเต็มจากผู้ใช้ โดยถ้าผู้ใช้ใส่ 1 คือการแสดงผลผังห้องพักในโรงแรม ใส่ 2 คือการทำให้ห้องพักว่าง ใส่ 3 คือจองห้องพัก ใส่ 4 คือเข้าพักในห้องพักใส่ 9 คือออกจากระบบและจบการทำงานใส่เลขอื่น ๆ นอกเหนือจากนี้ให้ระบบแจ้งว่าหมายเลขเมนูผิดพลาดพร้อมทั้งวนรับค่าเมนูใหม่

2) เมทอดประจำคลาสชื่อ `showRoom(...)` ที่รับอาร์เรย์ของอ็อบเจกต์ชนิด `Room` เข้ามาทางพารามิเตอร์ เพื่อใช้สำหรับแสดงผลผังห้องพักขึ้นบนจอภาพดังตัวอย่างเช่น (5 คะแนน)

[A1,1000.0,0]	[A2,1000.0,0]	[A3,1000.0,0]	[A4,1000.0,0]	[A5,1000.0,0]	[A6,1000.0,0]	[A7,1000.0,0]	[A8,1000.0,0]
[B1,1000.0,0]	[B2,1000.0,0]	[B3,1000.0,0]	[B4,1000.0,0]	[B5,1000.0,0]	[B6,1000.0,0]	[B7,1000.0,0]	[B8,1000.0,0]
[C1,1000.0,0]	[C2,1000.0,0]	[C3,1000.0,0]	[C4,1000.0,0]	[C5,1000.0,0]	[C6,1000.0,0]	[C7,1000.0,0]	[C8,1000.0,0]
[D1,1000.0,0]	[D2,1000.0,0]	[D3,1000.0,0]	[D4,1000.0,0]	[D5,1000.0,0]	[D6,1000.0,0]	[D7,1000.0,0]	[D8,1000.0,0]
[E1,1000.0,0]	[E2,1000.0,0]	[E3,1000.0,0]	[E4,1000.0,0]	[E5,1000.0,0]	[E6,1000.0,0]	[E7,1000.0,0]	[E8,1000.0,0]
[F1,1500.0,0]	[F2,1500.0,0]	[F3,1500.0,0]	[F4,1500.0,0]	[F5,1500.0,0]	[F6,1500.0,0]	[F7,1500.0,0]	[F8,1500.0,0]
[G1,1500.0,0]	[G2,1500.0,0]	[G3,1500.0,0]	[G4,1500.0,0]	[G5,1500.0,0]	[G6,1500.0,0]	[G7,1500.0,0]	[G8,1500.0,0]
[H1,1500.0,0]	[H2,1500.0,0]	[H3,1500.0,0]	[H4,1500.0,0]	[H5,1500.0,0]	[H6,1500.0,0]	[H7,1500.0,0]	[H8,1500.0,0]
[I1,2000.0,0]	[I2,2000.0,0]	[I3,2000.0,0]	[I4,2000.0,0]	[I5,2000.0,0]	[I6,2000.0,0]	[I7,2000.0,0]	[I8,2000.0,0]
[J1,2000.0,0]	[J2,2000.0,0]	[J3,2000.0,0]	[J4,2000.0,0]	[J5,2000.0,0]	[J6,2000.0,0]	[J7,2000.0,0]	[J8,2000.0,0]

3) เมทอดประจำคลาสชื่อ `updateStatus(...)` ที่รับอาร์เรย์ของอ็อบเจกต์ชนิด `Room` รับสตริงที่เป็นหมายเลขห้อง และรับจำนวนเต็มที่เป็นสถานะใหม่ของห้องนั้นเข้ามาทางพารามิเตอร์ เพื่อใช้สำหรับเปลี่ยนสถานะของห้องพักตามหมายเลขห้องที่ระบุ (5 คะแนน)

โจทย์ข้อที่ 10 [ระดับยาก] จงเขียนโปรแกรมภาษาจาวาที่สมบูรณ์เพื่อสร้างเมือง (City) ที่มีพื้นที่เป็นรูปสี่เหลี่ยมผืนผ้า โดยตัวเมืองนั้นเกิดจากการนำที่ดิน (Land) แปลงใด ๆ มาเรียงต่อกันในรูปของตาราง ซึ่งที่ดินแต่ละแปลงจะมีเจ้าของที่ดิน (Owner) ครอบครองอยู่ โดยการทำงานของโปรแกรมจะแบ่งออกเป็น 3 คลาสได้แก่ คลาส `Owner` แทนเจ้าของที่ดินแต่ละคน คลาส `Land` แทนที่ดินแต่ละแปลง และคลาส `City` แทนเมืองหนึ่งเมือง โดยมีรายละเอียดดังนี้ (35 คะแนน)



Owner



Land



City

กำหนดให้คลาส **Owner** มีรายละเอียดดังต่อไปนี้

```

1 public class Owner {
2     private int id;
3     private String name;
4     public Owner() {
5         id = 0; name = "";
6     }
7     public Owner(int i, String n) {
8         id = i; name = n;
9     }
10    public Owner(Owner own) {
11        id = own.id; name = own.name;
12    }
13    public int getID() { return id; }
14    public String getName() { return name; }
15    public void setID(int i) { id = i; }
16    public void setName(String n) { name = n; }
17    public void show() {
18        System.out.println("ID : " + id);
19        System.out.println("Name : " + name);
20    }
21 }

```

แอตทริบิวต์ **id** ใช้เก็บหมายเลขประจำตัวเจ้าของที่ดิน
แอตทริบิวต์ **name** ใช้เก็บชื่อเจ้าของที่ดิน

คอนสตรักเตอร์ 3 แบบเพื่อใช้กำหนดค่าเริ่มต้นให้กับเจ้าของที่ดิน

เมทอด **set** และ **get** เพื่อใช้กำหนดค่าและคืนค่าข้อมูลของเจ้าของที่ดิน

เมทอด **show** เพื่อใช้แสดงข้อมูลของเจ้าของที่ดินขึ้นบนจอภาพ

คลาส Owner

- 1) แอตทริบิวต์ประจำอ็อบเจ็คแบบ **private** ชื่อ **owner** ที่มีชนิดเป็น **Owner** (แอตทริบิวต์นี้เป็นอ็อบเจ็คของคลาส **Owner**) เพื่อเป็นตัวแทนของเจ้าของที่ดินแปลงนี้ (1 คะแนน)
- 2) แอตทริบิวต์ประจำอ็อบเจ็คแบบ **private** ชื่อ **price** เพื่อเก็บราคาที่ดิน (1 คะแนน)
- 3) คอนสตรักเตอร์เริ่มต้นเพื่อกำหนดค่าให้กับแอตทริบิวต์ทั้งสองตัวให้เป็นค่าเริ่มต้นตามประเภทข้อมูลของแอตทริบิวต์นั้น (แอตทริบิวต์ที่เป็นอ็อบเจ็คให้ใช้การ **new**) (2 คะแนน)
- 4) คอนสตรักเตอร์รายละเอียดที่มีพารามิเตอร์ 2 ตัวที่เป็นอ็อบเจ็คชนิด **Owner** และจำนวนจริง เพื่อกำหนดค่าเริ่มต้นให้กับแอตทริบิวต์ทั้งสองตัวให้เป็นไปตามค่าที่ระบุ (2 คะแนน)
- 5) เมทอดประจำอ็อบเจ็คแบบ **public** ชื่อ **setOwner** ซึ่งมีพารามิเตอร์ 1 ตัวที่เป็นอ็อบเจ็คของคลาส **Owner** เพื่อใช้กำหนดค่าให้กับเจ้าของที่ดินแปลงนี้ (กำหนดค่าให้กับแอตทริบิวต์ **owner** ที่เป็นอ็อบเจ็ค) โดยไม่คืนค่ากลับ (2 คะแนน)
- 6) เมทอดประจำอ็อบเจ็คแบบ **public** ชื่อ **setPrice** ที่มีพารามิเตอร์ 1 ตัวเพื่อใช้กำหนดค่าให้กับราคาที่ดินแปลงนี้ (กำหนดค่าให้กับแอตทริบิวต์ **price**) โดยไม่คืนค่ากลับ (2 คะแนน)
- 7) เมทอดประจำอ็อบเจ็คแบบ **public** ชื่อ **show** ที่ไม่มีพารามิเตอร์ เพื่อแสดงค่าของแอตทริบิวต์ทั้งสองตัวออกทางจอภาพที่ละบรรทัด โดยไม่มีการคืนค่ากลับ (2 คะแนน)

คลาส City (เป็นคลาสประมวลผลที่มีเมทอด **main(...)**)

- 1) สร้างเมืองที่เป็นอาร์เรย์สองมิติของอ็อบเจ็ค **Land** ชื่อ **land** ที่ประกอบไปด้วยที่ดินจำนวน 3000 แปลง โดยมีขนาดเป็น 50 x 60 (1 คะแนน)
- 2) ภายในที่ดินแต่ละแปลงต้องมีเจ้าของที่ดินและราคาทุกแปลง โดยเริ่มต้นให้ที่ดินทุกแปลงยังไม่มีเจ้าของครอบครอง (กำหนดเจ้าของที่ดินให้เป็น **Default**) และสุ่มราคาที่ดินตั้งแต่ราคา 1 บาทจนถึง 10 ล้านบาท เพื่อกำหนดให้กับที่ดินแต่ละแปลง (4 คะแนน)

- 3) สร้างอ็อบเจ็คชื่อ ow1 ที่เป็นเจ้าของที่ดินชื่อ "Tukie" โดยมีเลขประจำตัวเป็น 1001 พร้อมแสดงข้อมูลของเจ้าของที่ดินคนนั้นขึ้นบนจอภาพ (2 คะแนน)
- 4) สร้างอ็อบเจ็คชื่อ ow2 ที่เป็นเจ้าของที่ดินชื่อ "Annie" โดยมีเลขประจำตัวเป็น 1002 พร้อมแสดงข้อมูลของเจ้าของที่ดินคนนั้นขึ้นบนจอภาพ (2 คะแนน)
- 5) กำหนดให้ Tukie เข้าครอบครองที่ดินในแปลงตำแหน่งที่ (4, 9) พร้อมทั้งแสดงข้อมูลที่ดินแปลงนั้นออกทางจอภาพ (2 คะแนน)
- 6) เปลี่ยนราคาที่ดินที่ Tukie ครอบครองอยู่ให้เป็น 30 ล้านบาท พร้อมทั้งแสดงข้อมูลที่ดินแปลงนั้นออกทางจอภาพ (2 คะแนน)
- 7) เปลี่ยนให้ Annie เข้าไปครอบครองที่ดินแปลงที่ Tukie ครอบครองอยู่ (Annie เห็นว่าราคาที่ดินแปลงนี้ขึ้นเลยอยากเป็นเจ้าของ) พร้อมทั้งแสดงข้อมูลที่ดินแปลงนั้นออกทางจอภาพ (2 คะแนน)
- 8) กำหนดให้ Tukie เข้าครอบครองที่ดินแปลงใหม่ 50 แปลงที่อยู่ทางซ้ายสุดของเมือง พร้อมกำหนดให้ราคาที่ดินทุกแปลงเป็น 50 ล้านบาท และแสดงข้อมูลของที่ดินแต่ละแปลงออกทางจอภาพ (4 คะแนน)
- 9) กำหนดให้ Annie เข้าครอบครองที่ดินในแถวที่ทุกแถวทุกแปลงที่ยังไม่มีผู้ใดครอบครอง พร้อมทั้งแสดงข้อมูลของที่ดินแต่ละแปลงออกทางจอภาพ (4 คะแนน)

NOTE