

CHAPTER 13

คลาสและอ็อบเจกต์แบบซับซ้อน (Advanced Classes and Objects)

1. การรับค่าและคืนค่าประเภทข้อมูลแบบคลาส

1) เมทอดและตัวสร้างที่รับพารามิเตอร์เป็นอ็อบเจกต์ (พารามิเตอร์ที่มีประเภทข้อมูลเป็นคลาส)

```

1 public class Data {
2     public int x;
3     public Data() {
4         x = 0;
5     }
6     public Data(int n) {
7         x = n;
8     }
9     public Data(Data d) {
10        x = d.x;
11    }
12    public boolean isEqual(int n) {
13        return x == n;
14    }
15    public boolean isEqual(Data d) {
16        return x == d.x;
17    }
18 }

```

รับเข้ามาทั้งก้อนของอ็อบเจกต์ d แต่จะสนใจเพียงแค่แอตทริบิวต์ x ในอ็อบเจกต์ d เท่านั้น โดยจะเอาค่า x ที่อยู่ในอ็อบเจกต์ d หรือค่า d.x เข้าไปเก็บไว้ในแอตทริบิวต์ x ของคลาสต้นแบบ Data

รับเพียงแค่ int หนึ่งค่าในชื่อ n เข้ามาทางพารามิเตอร์

รับทั้งก้อนอ็อบเจกต์ชนิด Data หนึ่งก้อนในชื่อ d เข้ามาทางพารามิเตอร์ (แต่จะใช้งานเพียงแค่ค่า x ของอ็อบเจกต์ d เท่านั้น)

หมายเหตุ เรียกตัวสร้างแบบหมายเลข 1 ว่า ตัวสร้างเริ่มต้น (Default Constructor)

เรียกตัวสร้างแบบหมายเลข 2 ว่า ตัวสร้างรายละเอียด (Detail Constructor)

เรียกตัวสร้างแบบหมายเลข 3 ว่า ตัวสร้างคัดลอก (Copy Constructor)

2) เมทอดและตัวสร้างที่รับพารามิเตอร์ที่มีชื่อเหมือนกับแอตทริบิวต์ของคลาส

เมทอดหรือตัวสร้างใดก็ตามที่รับค่าพารามิเตอร์ที่มีชื่อเหมือนกับแอตทริบิวต์ของคลาสนั้น จะต้องใช้คีย์เวิร์ด **this** ระบุนำแอตทริบิวต์เพื่อไม่ให้เกิดความสับสนว่าค่าใดคือแอตทริบิวต์และค่าใดคือพารามิเตอร์ ดังตัวอย่าง

```

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

```

เกิดความสับสนขึ้น ว่า x ตัวใดคือแอตทริบิวต์และ x ตัวใดคือพารามิเตอร์

ในกรณีรับพารามิเตอร์ที่มีชื่ออื่นที่ไม่ใช่ชื่อ x สามารถเขียนเป็น **x = <พารามิเตอร์>** ได้ แต่ถ้าพารามิเตอร์มีชื่อเหมือนกับแอตทริบิวต์ x จะต้องเปลี่ยนเป็น **this.x = <พารามิเตอร์>**

คีย์เวิร์ด **this** สามารถใช้ได้กับแอตทริบิวต์เท่านั้น ไม่สามารถใช้ได้กับพารามิเตอร์

```

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

```

โจทย์ข้อที่ 1 [ระดับปานกลาง] จงแสดงผลลัพธ์ในคลาส TestNum ที่มีการสร้างและเรียกใช้ออบเจ็กต์จากคลาส Num ในรูปแบบต่างๆ พร้อมทั้งเติมเต็มส่วนของเมทอดในคลาส Num ให้สมบูรณ์ (16 คะแนน)

```
import java.util.Scanner;
public class TestNum {
    public static void main(String[] args) {
```

เรียกตัวสร้างใด
(ระบุหมายเลข)

ผลลัพธ์ที่แสดง
ขึ้นบนจอภาพคือ

```
Num n1 = new Num();
System.out.println(n1.x + "," + n1.y);
```

```
Num n2 = new Num(13, 2);
System.out.println(n2.x + "," + n2.y);
```

```
Num n3 = new Num(n2);
System.out.println(n3.x + "," + n3.y);
```

```
Num n4 = new Num(new Num(7, 11));
System.out.println(n4.x + "," + n4.y);
```

```
Num n5 = new Num(n4.y, n3.x);
System.out.println(n5.x + "," + n5.y);
```

```
n2 = new Num(new Num());
System.out.println(n2.x + "," + n2.y);
```

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

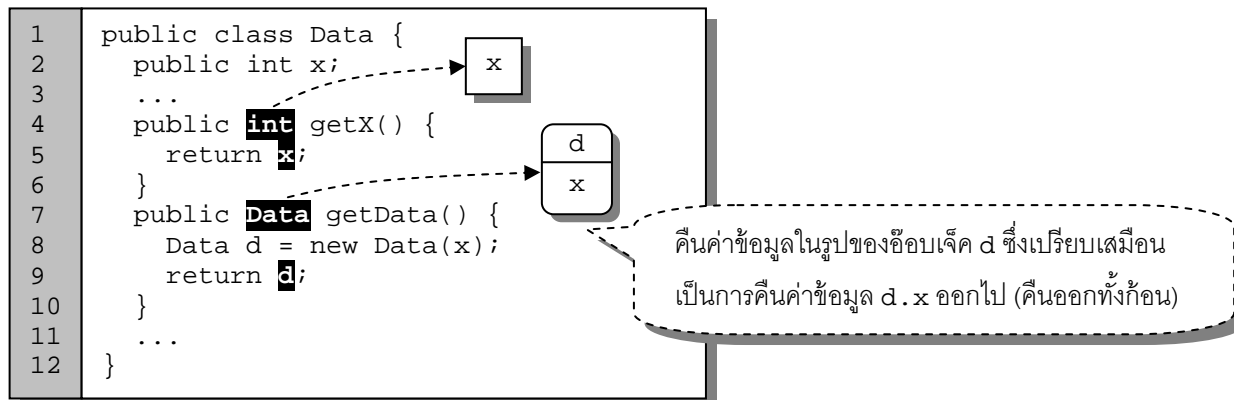
```
public class Num {
    public int x;
    public int y;
    Num() {
        x = y = 0;
    }
    Num(int x, int y) {
        this.x = x; this.y = y;
    }
    Num(Num n) {
        this.x = n.x; this.y = n.y;
    }
    public boolean isEqual(int x, int y) {
    }
    public boolean isEqual(Num n) {
    }
}
} //End of class
```

1

2

3

3) เมท็อดที่คืนค่าเป็นอ็อบเจ็ค (มีประเภทข้อมูลเป็นคลาส)



โจทย์ข้อที่ 2 [ระดับยาก] จงเขียนคลาส coin ให้สมบูรณ์ ซึ่งเป็นตัวแทนของเหรียญใดๆ ที่ประกอบไปด้วยมูลค่าของเหรียญ (Value) และหน้าของเหรียญ (Side) ที่เป็นหัว (H) หรือก้อย (T) พร้อมทั้งสร้างคลาส DemoCoin เพื่อสร้างและเรียกใช้อ็อบเจ็คจากคลาส Coin (20 คะแนน)

```
public class Coin {
```

```
//แอตทริบิวต์ side แบบ public เพื่อเก็บหน้าของเหรียญเช่น หัว "H" และ ก้อย "T"
```

```
//แอตทริบิวต์ value แบบ public เพื่อเก็บมูลค่าของเหรียญเช่น 1 บาท, 2 บาท, 5 บาท และ 10 บาท
```

```
//กำหนดให้แอตทริบิวต์ list เป็นรายการของหน้าที่เป็นไปได้ทั้งหมดของเหรียญ
```

```
private final String list[] = { "H", "T" };
```

```
//ตัวสร้างแบบไม่มีพารามิเตอร์เพื่อกำหนดให้หน้าของเหรียญเป็นหัว (H) และมูลค่าของเหรียญเป็น 0
```

```
//ตัวสร้างแบบมีพารามิเตอร์สองตัวที่เป็นสตริงและจำนวนเต็มเพื่อกำหนดหน้าและมูลค่าของเหรียญ
```

```
//ตัวสร้างแบบมีพารามิเตอร์หนึ่งตัวประเภท coin เพื่อกำหนดหน้าและมูลค่าของเหรียญ
```

//เมทอด flip(...) เพื่อเปลี่ยนหน้าของเหรียญโดยการสุ่ม (ควรเรียกใช้งานแอตทริบิว list)

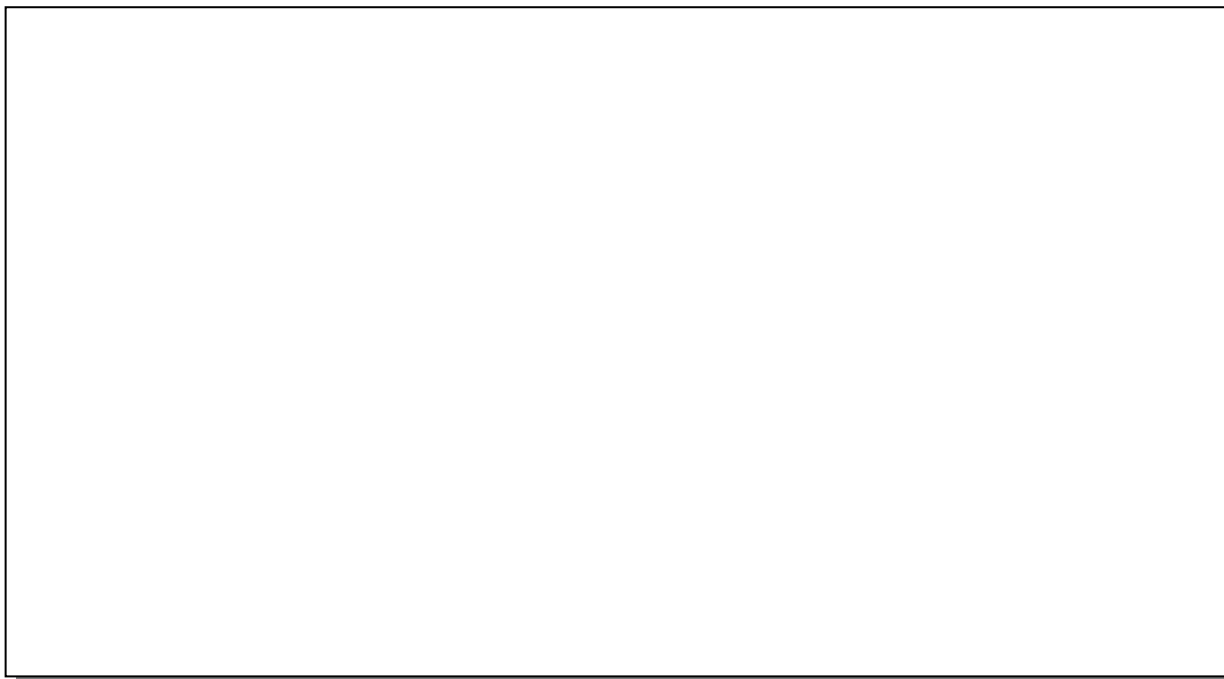
//เมทอด reverse(...) เพื่อกลับหน้าของเหรียญจากหัวเป็นก้อยหรือจากก้อยเป็นหัว

//เมทอด equals(...) เพื่อรับเหรียญอีกหนึ่งเหรียญเข้ามาทางพารามิเตอร์ แล้วตรวจสอบว่ามีหน้า
และมูลค่าของเหรียญนั้นเท่ากับเหรียญของคลาสนี้หรือไม่

} //End of class

/* คลาส DemoCoin เพื่อสร้างและเรียกใช้งานคลาส coin โดยให้สร้างเหรียญ 4 เหรียญที่ประกอบไปด้วยเหรียญ 1 บาทหน้าหัวจำนวน 2 เหรียญ (ต้องใช้ตัวสร้างคนละแบบกัน) และเหรียญ 5 บาทหน้าก้อยจำนวน 2 เหรียญ (ต้องใช้ตัวสร้างคนละแบบกัน) พร้อมสุ่มหน้าของเหรียญ 1 บาทเหรียญที่หนึ่งและกลับหน้าเหรียญที่สอง และสุ่มหน้าเหรียญ 5 บาทเหรียญที่หนึ่งและกลับหน้าเหรียญที่สอง โดยทำ 20 ครั้ง พร้อมตรวจสอบว่าเหรียญแต่ละคู่เท่ากันหรือไม่ แล้วแสดงรายละเอียดนั้นขึ้นบนจอภาพ */

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



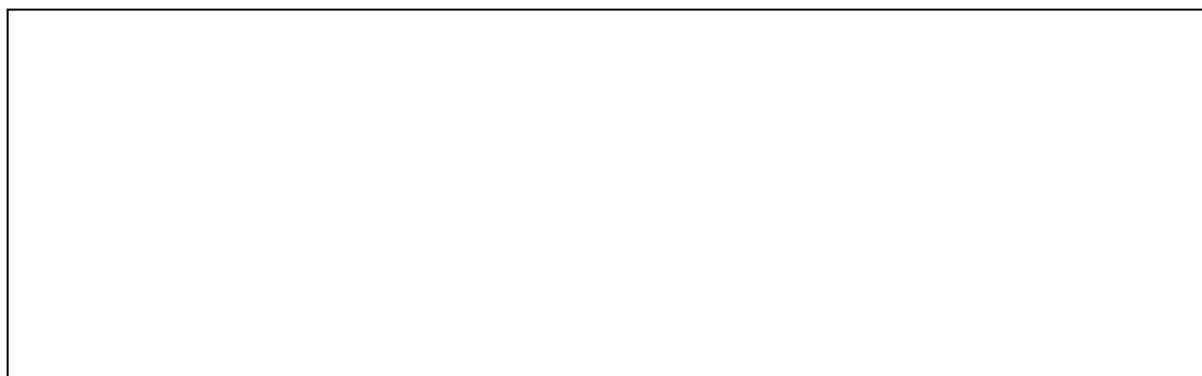
```
    } //End of main
} //End of class
```

โจทย์ข้อที่ 3 [ระดับยาก] จงเขียนโปรแกรมจาวาเพื่อสร้างคลาส `SetOfInteger` ที่ใช้แทนเซตของตัวเลขจำนวนเต็ม โดยจะเก็บไว้ในอาร์เรย์ที่เรียงจากน้อยไปหามากเสมอ และตัวเลขในอาร์เรย์ต้องไม่ซ้ำกันเลย โดยให้สร้างเมทอด และแอตทริบิวตามรายละเอียดที่กำหนดให้ (30 คะแนน)

```
public class SetOfInteger {
    //ประกาศแอตทริบิวแบบ public ที่เป็นอาร์เรย์ประเภทจำนวนเต็มชื่อ iset (ยังไม่ต้องระบุขนาด)
```



```
//กำหนดตัวสร้างแบบที่รับพารามิเตอร์ 1 ตัวที่เป็นอาร์เรย์ชนิดจำนวนเต็มเพื่อนำมากำหนดค่าเริ่มต้นให้กับแอตทริบิว iset โดยจะต้องมีการจัดเรียงจากน้อยไปมากและกำจัดสมาชิกตัวที่ซ้ำกันเสมอ ตัวอย่าง { 7, 5, 3, 5, 3 } จะได้เป็น { 3, 5, 7 } เป็นต้น (เรียกใช้เมทอดที่กำหนดไว้ให้)
    public SetOfInteger(int[] x) {
```



```
    } //End of constructor
```

//กำหนดตัวสร้างแบบที่รับพารามิเตอร์ 1 ตัวที่เป็นชนิด SetOfInteger เพื่อนำมากำหนดค่าเริ่มต้นให้กับแอตทริบิว iset

```
public SetOfInteger(SetOfInteger s) {
```



```
} //End of constructor
```

//กำหนดให้เมทอด sort(...) ใช้จัดเรียงข้อมูลแบบฟองจากน้อยไปมาก

```
public int[] sort(int x[]) {
    for (int i = x.length - 1; i >= 1; i--)
        for (int j = 0; j <= i; j++)
            if (x[j] > x[j + 1]) {
                int temp = x[j];
                x[j] = x[j + 1];
                x[j + 1] = temp;
            }
    }
    return x;
} //End of method
```

//กำหนดให้เมทอด removeDuplicatedMembers(...) ใช้กำจัดสมาชิกตัวที่ซ้ำกัน

```
public int[] removeDuplicatedMembers(int x[]) {
    int t[] = new int[x.length];
    int index = 0;
    for(int i = 0; i < x.length; i++) {
        if (i == 0) {
            t[index] = x[i];
            index++;
        } else {
            int j;
            for (j = 0; j < index; j++) {
                if (t[j] == x[i]) break;
            }
            if (j == index) {
                t[index] = x[i];
                index++;
            }
        }
    }
    int n[] = new int[index];
    for (int i = 0; i < n.length; i++) n[i] = t[i];
    return n;
} //End of method
```

//ให้เมทอด equals(...) ใช้ตรวจสอบว่าแอตทริบิว iset ในคลาสนี้ กับ iset จากอ็อบเจ็ค x ที่รับเข้ามาทางพารามิเตอร์เท่ากันหรือไม่

```
public boolean equals(SetOfInteger x) {
```

```
} //End of method
```

//ให้เมทอด toString(...) ใช้คืนค่าสมาชิกทุกตัวในแอตทริบิว iset ในรูปแบบของสตริง

```
public String toString() {
```

```
} //End of method
```

//ให้เมทอด addElement(...) ใช้เพิ่มจำนวนเต็มใดๆ หนึ่งค่าเข้าไปในแอตทริบิว iset โดยจะต้องขยายขนาดของแอตทริบิว iset และจัดเรียงและกํากับสมาชิกตัวที่ซ้ำใหม่อีกครั้งหนึ่ง

```
public void addElement(int a) {
```

```
} //End of method
```

//ให้เมทอด findElement(...) ใช้ค้นหาตำแหน่งของตัวเลขค่าใดค่าหนึ่งในแอตทริบิว iset โดยถ้าค้นเจอให้คืนค่าตำแหน่งที่เจอ ถ้าไม่เช่นนั้นให้คืนค่า -1

```
public int findElement(int a) {
```

```
} //End of method
```

//ให้เมทอด isSubset(...) ใช้ตรวจสอบว่าค่า iset ในอ็อบเจ็ค s เป็นสับเซต (Subset) ของแอตทริบิว iset ในคลาสนี้หรือไม่

```
public boolean isSubset(SetOfInteger s) {
```

```
} //End of method
```

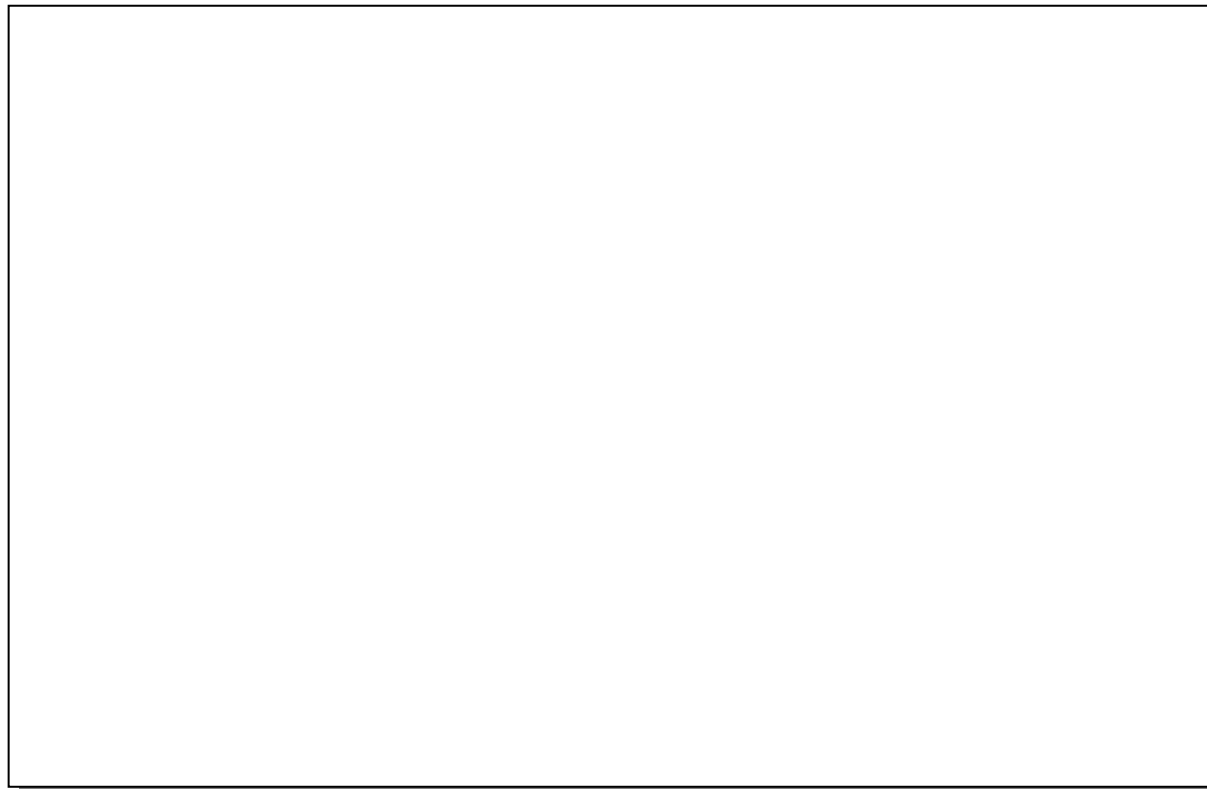
//ให้เมทอด union(...) ใช้คำนวณหาเซตที่เกิดจากการนำเอา iset ในคลาสนี้กับ iset ที่รับเข้ามาทางพารามิเตอร์มายูเนียนกัน (Union) แล้วคืนค่าคำตอบชนิด SetOfInteger

```
public SetOfInteger union(SetOfInteger s) {
```

```
} //End of method
```


//ให้เมทอด intersect(...) ใช้คำนวณหาเซตที่เกิดจากการนำเอา iset ในคลาสนี้กับ iset ที่รับเข้ามาทางพารามิเตอร์มาอินเตอร์เซกกัน (Intersection) แล้วคืนค่าคำตอบชนิด SetOfInteger

```
public SetOfInteger intersect(SetOfInteger s) {
```



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

2. การประกาศและสร้างอาร์เรย์ของอ็อบเจ็ค

1) วิธีการประกาศและสร้างอาร์เรย์ของอ็อบเจ็ค

ทำแบบเดียวกับการประกาศและสร้างอาร์เรย์แบบทั่วไป ซึ่งมีรายละเอียดดังนี้

(1) การประกาศและสร้างอาร์เรย์ของอ็อบเจ็คแบบ 1 มิติ

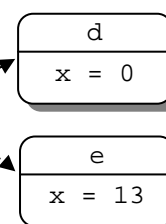
```
<ชื่อคลาส> <ชื่ออ็อบเจ็ค>[] = new <ชื่อคลาส>[ขนาด];
```

(2) การประกาศและสร้างอาร์เรย์ของอ็อบเจ็คแบบ 2 มิติ

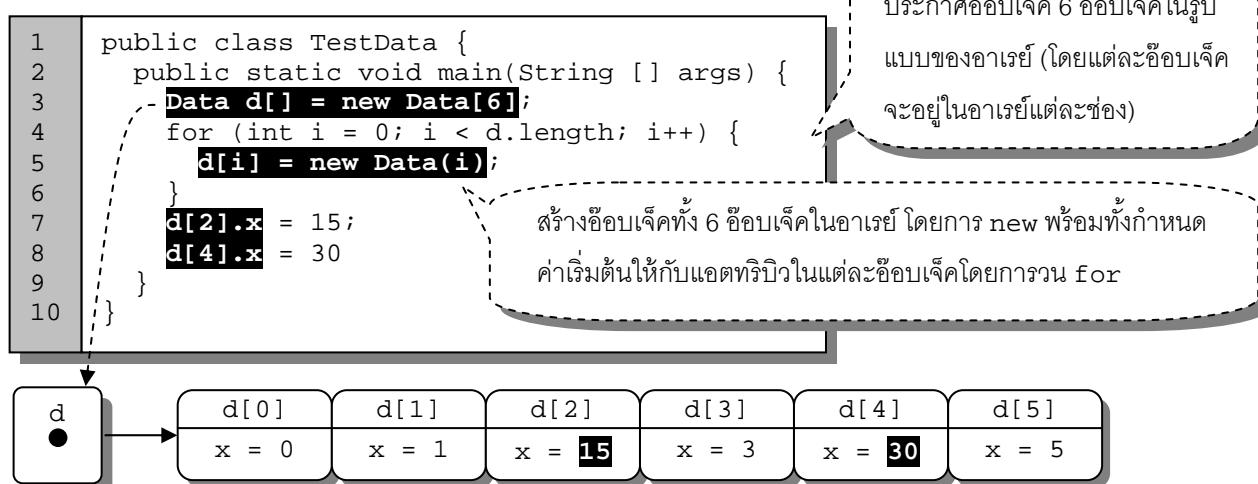
```
<ชื่อคลาส> <ชื่ออ็อบเจ็ค>[][] = new <ชื่อคลาส>[ขนาดแถว][ขนาดหลัก];
```

ตัวอย่างการประกาศและสร้างอ็อบเจ็คแบบทั่วไป (โดยอ้างอิงคลาส Data จากหน้าแรก)

```
1 public class TestData {
2     public static void main(String [] args) {
3         Data d = new Data();
4         Data e = new Data(13);
5         System.out.println(d.x + ", " + e.x);
6     }
7 }
```



ตัวอย่างการประกาศและสร้างอาร์เรย์ของอ็อบเจ็ค



โจทย์ข้อที่ 4 [ระดับยาก] จงสร้างคลาส `student` ที่ใช้แทนนิสิตหนึ่งคน ดังรายละเอียดต่อไปนี้ (30 คะแนน)

```
public class Student {
```

//แอตทริบิวต์ประจำอ็อบเจ็ค 3 ตัวแบบ private ได้แก่ (4 คะแนน)

- แอตทริบิวต์ `id` ชนิดจำนวนเต็มเพื่อเก็บเลขประจำตัวนิสิต
- แอตทริบิวต์ `name` ชนิดสตริงเพื่อเก็บชื่อนิสิต
- แอตทริบิวต์ `score` ชนิดจำนวนจริงซึ่งเป็นอาร์เรย์ที่เก็บคะแนนทุกรายวิชาที่ลงทะเบียนของนิสิต

//ตัวสร้างแบบไม่กำหนดพารามิเตอร์ (2 คะแนน)

//ตัวสร้างแบบ 3 พารามิเตอร์ ได้แก่ เลขประจำตัว ชื่อนิสิต และจำนวนวิชาที่ลงทะเบียน (2 คะแนน)

//ตัวสร้างแบบ 1 พารามิเตอร์ที่เป็นอ็อบเจกต์ของคลาส student (2 คะแนน)

//เมทอดประจำอ็อบเจกต์ 4 เมทอดได้แก่ (10 คะแนน)

- เมทอด `getID(...)` ที่ไม่รับพารามิเตอร์ เพื่อคืนค่าเลขประจำตัวนิสิต
- เมทอด `getName(...)` ที่ไม่รับพารามิเตอร์ เพื่อคืนค่าชื่อนิสิต
- เมทอด `getScore(...)` ที่ไม่รับพารามิเตอร์ เพื่อคืนค่าคะแนนทุกรายวิชาที่ลงทะเบียน
- เมทอด `setScore(...)` ที่รับค่าคะแนนทุกรายวิชาของนิสิตที่เป็นอาจารย์เข้ามาทางพารามิเตอร์ เพื่อกำหนดค่าให้กับแอตทริบิว `score` โดยเมทอดนี้ไม่มีการคืนค่ากลับ

} //End of class

จงสร้างคลาส `Freshmen` ที่มีเพียงเมทอด `main(...)` โดยมีรายละเอียดดังต่อไปนี้ (15 คะแนน)

```
import java.util.Scanner;
public class Freshmen {
    public static void main(String[] args) {
        Scanner kb = new Scanner(System.in);
        //กำหนดให้มีนิสิตทั้งหมด 800 คน ที่สร้างมาจากคลาส student (เป็นอาจารย์ของอ็อบเจกต์)
```

//กำหนดค่าเริ่มต้นให้กับเลขประจำตัวของนิสิตของแต่ละคนแบบอัตโนมัติ โดยเรียงลำดับตั้งแต่ 5300121 จนถึง 5380021 และกำหนดค่าเริ่มต้นให้กับชื่อนิสิตของแต่ละคน โดยการวนรับค่าจากแป้นพิมพ์ทีละชื่อ และกำหนดจำนวนวิชาที่นิสิตทุกคนลงทะเบียนเป็น 4 รายวิชาเท่ากันทุกคน

//เรียกใช้งานเมทอด `setScore(...)` ผ่านทางอ็อบเจกต์ของนิสิตแต่ละคน เพื่อใส่คะแนนทั้ง 4 รายวิชาให้กับนิสิตแต่ละคน โดยค่าคะแนนได้จากการสุ่มคะแนนในช่วง 0 ถึง 100

//แสดงเลขประจำตัวนิสิต ชื่อนิสิต และคะแนนทั้ง 4 รายวิชาของนิสิตแต่ละคนออกทางจอภาพ

//คำนวณหาคะแนนเฉลี่ยของแต่ละรายวิชา โดยเกิดจากการเอาคะแนนแต่ละรายวิชาของนิสิตทุกคนมารวมกันแล้วหารด้วยจำนวนนิสิตทั้งหมด พร้อมทั้งแสดงผลลัพธ์ออกทางจอภาพ

```

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

```

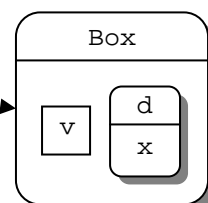
3. การประกาศและสร้างแอตทริบิวต์ที่มีประเภทข้อมูลแบบคลาส

1) วิธีการประกาศและสร้างแอตทริบิวต์ที่เป็นอ็อบเจ็ค (มีประเภทข้อมูลเป็นคลาส)

```

1 public class Box {
2     public double v;
3     public Data d;
4     public Box() {
5         v = 0.0;
6         d = new Data();
7     }
8     public Box(double a, Data b) {
9         v = a;
10        d = new Data(b);
11    }
12    public Box(Box b) {
13        v = b.v;
14        d = new Data(b.d);
15    }
16    ...
17 }

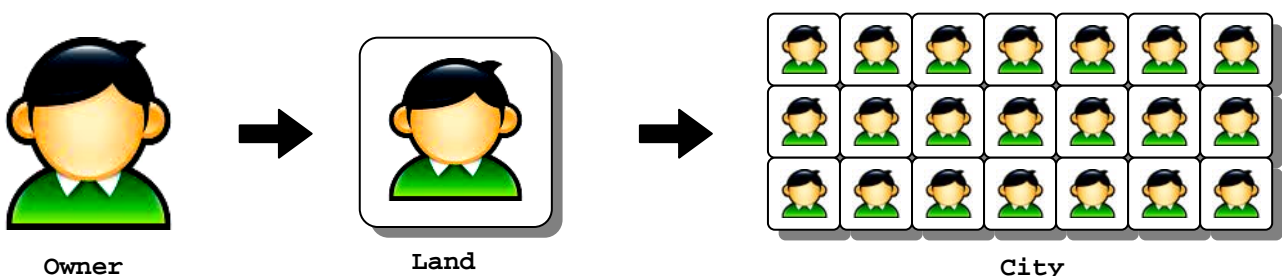
```



การกำหนดค่าเริ่มต้นให้กับแอตทริบิวต์ v ซึ่งเป็นประเภท double จะใช้เครื่องหมายเท่ากับ (แบบเดิมที่เคยทำ)

การกำหนดค่าเริ่มต้นให้กับแอตทริบิวต์ d ซึ่งเป็นอ็อบเจ็คประเภท Data จะใช้หลักการ new

โจทย์ข้อที่ 5 [ระดับยาก] จงเขียนโปรแกรมภาษาจาวาที่สมบูรณ์เพื่อสร้างเมือง (City) ที่มีพื้นที่เป็นรูปสี่เหลี่ยมผืนผ้าหรือสี่เหลี่ยมจัตุรัส โดยตัวเมืองนั้นเกิดจากการนำที่ดิน (Land) แปลงใดๆ มาเรียงต่อกันในรูปของตาราง ซึ่งที่ดินแต่ละแปลงจะมีเจ้าของที่ดิน (Owner) ครอบครองอยู่ โดยการทำงานของโปรแกรมจะแบ่งออกเป็น 3 คลาสได้แก่ คลาส Owner แทนเจ้าของที่ดินแต่ละคน คลาส Land แทนที่ดินแต่ละแปลง และคลาส City แทนเมืองหนึ่งเมือง โดยมีรายละเอียดดังนี้ (30 คะแนน)



1. คลาส Owner มีรายละเอียดดังนี้ (10 คะแนน)

```
public class Owner {
```

```
//แอตทริบิวต์ประจำอ็อบเจ็คแบบ private ชื่อ id เพื่อเก็บหมายเลขประจำตัว (ให้เป็นจำนวนเต็ม)
```

```
//แอตทริบิวต์ประจำอ็อบเจ็คแบบ private ชื่อ name เพื่อเก็บชื่อเจ้าของที่ดิน
```

```
//ตัวสร้างแบบ public ที่ไม่มีพารามิเตอร์ เพื่อกำหนดค่าเริ่มต้นให้กับแอตทริบิวต์ทั้งสองตัวให้เป็นค่าเริ่มต้นตามประเภทข้อมูลของแอตทริบิวต์นั้น (เช่น int เป็น 0 หรือ double เป็น 0.0 เป็นต้น)
```

```
//ตัวสร้างแบบ public ที่มีพารามิเตอร์ 2 ตัวที่เป็นจำนวนเต็มและสตริง เพื่อกำหนดค่าเริ่มต้นให้กับแอตทริบิวต์ทั้งสองตัวให้เป็นไปตามค่าที่ระบุ
```

```
//ตัวสร้างแบบ public ที่มีพารามิเตอร์ 1 ตัวที่เป็นอ็อบเจ็คชนิด Owner เพื่อกำหนดค่าเริ่มต้นให้กับแอตทริบิวต์ทั้งสองตัวให้เป็นไปตามค่าที่ได้จากอ็อบเจ็คนั้น
```

```
//เมธอดประจำอ็อบเจ็คแบบ public ชื่อ getID ที่ไม่มีพารามิเตอร์ เพื่อคืนค่าหมายเลขประจำตัว
```

//เมทอดประจำอ็อบเจ็คแบบ public ชื่อ getName ที่ไม่มีพารามิเตอร์ เพื่อคืนค่าชื่อเจ้าของที่ดิน

//เมทอดประจำอ็อบเจ็คแบบ public ชื่อ setID ที่รับค่าจากพารามิเตอร์ 1 ตัว เพื่อใช้กำหนดค่าให้กับหมายเลขประจำตัว (กำหนดค่าให้แอตทริบิวต์ id) โดยไม่มีการคืนค่ากลับ

//เมทอดประจำอ็อบเจ็คแบบ public ชื่อ setName ที่รับค่าจากพารามิเตอร์ 1 ตัว เพื่อใช้กำหนดค่าให้กับชื่อเจ้าของที่ดิน (กำหนดค่าให้แอตทริบิวต์ name) โดยไม่มีการคืนค่ากลับ

//เมทอดประจำอ็อบเจ็คแบบ public ชื่อ show ที่ไม่มีพารามิเตอร์ เพื่อแสดงค่าของแอตทริบิวต์ทั้งสองตัวออกทางจอภาพที่ละบรรทัด โดยไม่มีการคืนค่ากลับ

} //End of class

2. คลาส Land มีรายละเอียดดังนี้ (10 คะแนน)

public class Land {

//แอตทริบิวต์ประจำอ็อบเจ็คแบบ public ชื่อ n ที่เป็นอ็อบเจ็คของคลาส Owner (ประกาศอ็อบเจ็คจากคลาส Owner มาเป็นแอตทริบิวต์ในคลาส Land) เพื่อเป็นตัวแทนของเจ้าของที่ดินแปลงนี้

//แอตทริบิวต์ประจำอ็อบเจ็คแบบ public ชื่อ price เพื่อเก็บราคาที่ดิน

//ตัวสร้างแบบ public ที่ไม่มีพารามิเตอร์ เพื่อกำหนดค่าเริ่มต้นให้กับแอตทริบิวต์ทั้งสองตัวให้เป็นค่าเริ่มต้นตามประเภทข้อมูลของแอตทริบิวต์นั้น (แอตทริบิวต์ที่เป็นอ็อบเจ็กต์ให้ใช้การ new)

//ตัวสร้างแบบ public ที่มีพารามิเตอร์ 2 ตัวที่เป็นอ็อบเจ็กต์ชนิด Owner และจำนวนจริง เพื่อกำหนดค่าเริ่มต้นให้กับแอตทริบิวต์ทั้งสองตัวให้เป็นไปตามค่าที่ระบุ

//เมทอดประจำอ็อบเจ็กต์แบบ public ชื่อ setOwner ที่มีพารามิเตอร์ 1 ตัวคืออ็อบเจ็กต์ของคลาส Owner เพื่อใช้กำหนดค่าให้กับเจ้าของที่ดินแปลงนี้ (กำหนดค่าให้แอตทริบิวต์ n ที่เป็นอ็อบเจ็กต์) โดยไม่มีการคืนค่ากลับ

//เมทอดประจำอ็อบเจ็กต์แบบ public ชื่อ setPrice ที่มีพารามิเตอร์ 1 ตัวเพื่อใช้กำหนดค่าให้กับราคาที่ดินแปลงนี้ (กำหนดค่าให้แอตทริบิวต์ price) โดยไม่มีการคืนค่ากลับ

//เมทอดประจำอ็อบเจ็กต์แบบ public ชื่อ show ที่ไม่มีพารามิเตอร์ เพื่อแสดงค่าของแอตทริบิวต์ทั้งสองตัวออกทางจอภาพที่ละบรรทัด โดยไม่มีการคืนค่ากลับ

} //End of class

3. คลาส City ที่มีเมทอด main โดยมีรายละเอียดภายในเมทอด main ดังนี้ (10 คะแนน)

```
import java.util.Scanner;
public class City {
    public static void main(String[] args) {
        //สร้างเมืองที่เป็นอาเรียของอ็อบเจ็ค Land ที่ประกอบไปด้วยที่ดิน 3000 แปลง ขนาด 50 x 60
        (เป็นอาเรียสองมิติ)
```

//ภายในที่ดินแต่ละแปลงต้องมีเจ้าของที่ดินและราคาทุกแปลง โดยเริ่มต้นให้ที่ดินทุกแปลงยังไม่มีเจ้าของครอบครอง (กำหนดเจ้าของที่ดินให้เป็น Default) และสุ่มราคาที่ดินตั้งแต่ราคา 1 บาท จนถึง 10 ล้านบาทเพื่อกำหนดให้กับที่ดินแต่ละแปลง

//สร้างอ็อบเจ็คชื่อ ow1 ที่เป็นเจ้าของที่ดินชื่อ "Tukie" โดยมีเลขประจำตัวเป็น 1001 พร้อมแสดงข้อมูลของเจ้าของที่ดินคนนั้นบนจอภาพ

//สร้างอ็อบเจ็คชื่อ ow2 ที่เป็นเจ้าของที่ดินชื่อ "Annie" โดยมีเลขประจำตัวเป็น 1002 พร้อมแสดงข้อมูลของเจ้าของที่ดินคนนั้นบนจอภาพ

//กำหนดให้ Tukie ครอบครองที่ดินในแปลงตำแหน่งที่ (4, 9) พร้อมทั้งแสดงข้อมูลที่ดินแปลงนั้นออกทางจอภาพ

//เปลี่ยนราคาที่ดินที่ Tukie ครอบครองอยู่ให้เป็น 30 ล้านบาท พร้อมทั้งแสดงข้อมูลที่ดินแปลงนั้นออกทางจอภาพ

//เปลี่ยนให้ Annie เข้าไปครอบครองที่ดินแปลงที่ Tukie ครอบครองอยู่ พร้อมทั้งแสดงข้อมูลที่ดินแปลงนั้นออกทางจอภาพ

//กำหนดให้ Tukie เข้าครอบครองที่ดินแปลงใหม่ 50 แปลงที่อยู่ทางซ้ายสุดของเมือง (แถวที่ 1 ของเมือง) พร้อมกำหนดให้ราคาที่ดินทุกแปลงเป็น 50 ล้านบาท พร้อมทั้งแสดงข้อมูลของที่ดินแต่ละแปลงออกทางจอภาพ

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

โจทย์ข้อที่ 6 [ระดับยาก] โปรแกรมที่กำหนดให้ต่อไปนี้เป็นโปรแกรมที่ใช้คลาส `scanner` ในการรับค่าตัวเลขจำนวนเต็ม 1 ค่าจากแป้นพิมพ์ แล้วตามด้วยรับค่าสตริงอีก 1 ค่าจากแป้นพิมพ์ ซึ่งผลปรากฏว่าเกิดข้อผิดพลาดขึ้นกับตัวโปรแกรม โดยโปรแกรมเมื่อรับค่าตัวเลขจำนวนเต็มเรียบร้อยแล้ว จะไม่สามารถรับค่าสตริงต่อไปได้ เมื่อตรวจสอบสาเหตุก็พบว่า หลังจากรับค่าตัวเลขจำนวนเต็มแล้วตามด้วยการกดปุ่ม Enter จะมีอักขระบางตัวที่มองไม่เห็นเกิดขึ้น ซึ่งอักขระดังกล่าวได้ถูกเก็บเข้าไปในตัวแปรสตริงที่ตั้งใจจะรับค่าในตอนแรกแทน จึงทำให้โปรแกรมไม่สามารถรับค่าสตริงที่ต้องการได้

```
1 import java.util.Scanner;
2 public class Test {
3     public static void main(String[] args) {
4         Scanner kb = new Scanner(System.in);
5         System.out.print("Enter Int: ");
6         int a = kb.nextInt();
7         System.out.print("Enter Str: ");
8         String b = kb.nextLine();
9         System.out.println(a + "," + b);
10    }
11 }
```

ผลลัพธ์จริงที่ได้

Enter Int: 10
Enter Str: 10

ผลลัพธ์ที่ควรจะได้ (แต่ไม่ได้ตามนี้)

Enter Int: 10
Enter Str: Java
10,Java

จงสร้างคลาสขึ้นใหม่ชื่อ `sscaner` เพื่อทำการแก้ไขข้อผิดพลาดดังกล่าว โดยคลาส `sscaner` นี้จะทำงานคล้ายกับคลาส `scanner` โดยมีส่วนประกอบดังต่อไปนี้ (10 คะแนน)

```
import java.util.Scanner;
import java.io.InputStream;
public class SScanner {
    //แอตทริบิวต์ประจำอ็อบเจ็คชื่อ kb ที่มีชนิดเป็นแบบ Scanner
```

```
//ตัวสร้างที่รับพารามิเตอร์ 1 ตัวชื่อ in ที่มีชนิดเป็นแบบ InputStream เพื่อกำหนดค่าเริ่มต้นให้กับ
แอตทริบิวต์ kb
```

```
//เมทอดประจำอ็อบเจ็คชื่อ nextInt(...) ที่ไม่รับพารามิเตอร์ใดๆ โดยจะรับค่าตัวเลขจำนวนเต็ม
1 ค่าจากแป้นพิมพ์แล้วคืนค่าตัวเลขที่รับนั้น
```

```
//เมทอดประจำอ็อบเจ็คชื่อ nextLine(...) ที่ไม่รับพารามิเตอร์ใดๆ โดยจะรับค่าสตริง 1 ค่าจากแป้น
พิมพ์ ถ้าสตริงมีความยาวมากกว่า 0 ให้คืนค่าสตริงที่รับมา ถ้าไม่เช่นนั้นจะวนรับไปเรื่อยๆ
```

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

จงสร้างคลาสอีกหนึ่งคลาสชื่อ TestSScanner เพื่อเรียกใช้งานคลาส sscanner แล้วทำการรับค่าแบบเดิมอีกครั้งหนึ่ง เพื่อทดสอบว่าโปรแกรมสามารถทำงานได้อย่างไม่ผิดพลาดอีกต่อไป (5 คะแนน)

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

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

โจทย์ข้อที่ 7 [ระดับเทพ] นางสาวแอนนี่ (มอล) นิสิตทิ้งตงามที่สุดในภาคพื้นเอเชียตะวันออกเฉียงใต้ ได้จัดเตรียมของขวัญสำหรับแจกเหล่าบรรดาหนุ่มๆ ซึ่งมารุมจีบเธออย่างไม่ขาดสาย โดยของขวัญของเธอนั้นเป็นกระเช้าผลไม้ที่จัดตกแต่งอย่างสวยงาม แต่เนื่องจากการรุ่มร้อนและอัดอั้นในวิชาจาวาของเธอ จึงทำให้แอนนี่เขียนโปรแกรมภาษาจาวาเพื่อสร้างกระเช้าผลไม้ที่ใช้สำหรับใส่ผลไม้ขึ้นมา โดยประกอบไปด้วยคลาส 3 คลาสได้แก่ คลาส Fruit เพื่อแทนผลไม้ คลาส Basket เพื่อแทนกระเช้า และคลาส FruitBasket เพื่อใช้เป็นกระเช้าผลไม้ที่จัดตกแต่งอย่างสวยงามพร้อมใช้แจกให้กับบรรดากิ๊ก เอ้ย! บรรดาหนุ่มๆ ของเธอ โดยมีรายละเอียดของแต่ละคลาสดังต่อไปนี้ (50 คะแนน)

1. คลาส Fruit เพื่อแทนผลไม้หนึ่งผลใดๆ (10 คะแนน)

```
public class Fruit {
```

```
    //แอตทริบิวต์ประจำอ็อบเจ็คชื่อ name เพื่อเก็บชื่อผลไม้ เช่น ส้ม แอปเปิ้ล กล้วย เป็นต้น
```

```
    //แอตทริบิวต์ประจำอ็อบเจ็คชื่อ weigh เพื่อเก็บน้ำหนักของผลไม้
```

```
    //กำหนดให้แอตทริบิวต์ wh แทนน้ำหนักที่เป็นไปได้ทั้งหมดของผลไม้แต่ละผล
```

```
    private final double wh[] = { 0.1, 0.2, 0.3, 0.4, 0.5 };
```

```
    //ตัวสร้าง 4 แบบ ดังต่อไปนี้
```

- ตัวสร้างแบบไม่มีพารามิเตอร์ เพื่อกำหนดค่าเริ่มต้นให้กับแอตทริบิวต์ทุกตัว
- ตัวสร้างแบบหนึ่งพารามิเตอร์ที่เป็นสตริง เพื่อใช้กำหนดชื่อผลไม้ ส่วนน้ำหนักของผลไม้ นั้นจะไดจากการสุ่ม (แนะนำให้เรียกใช้แอตทริบิวต์ wh)

- ตัวสร้างแบบสองพารามิเตอร์ที่เป็นสตริงและจำนวนจริง เพื่อใช้กำหนดชื่อผลไม้ และน้ำหนัก ของผลไม้ตามลำดับ
- ตัวสร้างแบบหนึ่งพารามิเตอร์ประเภท Fruit เพื่อใช้กำหนดชื่อผลไม้ และน้ำหนักของผลไม้

```
//เมทอดประจำอ็อบเจ็คชื่อ showFruit เพื่อใช้แสดงรายละเอียดของผลไม้ขึ้นบนจอภาพ
```

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

2. คลาส Basket เพื่อแทนกระเช้า (20 คะแนน)

```
public class Basket {
    //แอตทริบิวต์ประจำอ็อบเจ็คชื่อ fruit ประเภท Fruit ซึ่งเป็นอาเรย์ที่ใช้สำหรับเก็บผลไม้
    //แอตทริบิวต์ประจำอ็อบเจ็คชื่อ count เพื่อบอกจำนวนของผลไม้ที่เก็บอยู่ในกระเช้า
```

//ตัวสร้างแบบไม่มีพารามิเตอร์เพื่อกำหนดให้กระเช้าสามารถเก็บจำนวนผลไม้ได้สูงสุด 10 ผล

//ตัวสร้างแบบหนึ่งพารามิเตอร์ชนิดจำนวนเต็ม เพื่อกำหนดให้กระเช้าสามารถเก็บจำนวนผลไม้ได้สูงสุดตามจำนวนเต็มที่ระบุ

//ตัวสร้างแบบหนึ่งพารามิเตอร์ชนิด Basket เพื่อกำหนดให้กระเช้ามีข้อมูลทุกอย่างเหมือนกับกระเช้าที่รับเข้ามา พร้อมทั้งนำผลไม้จากกระเช้าที่รับเข้ามาทางพารามิเตอร์มาใส่ในกระเช้านี้

//เมทอดประจำอ็อบเจ็คชื่อ isFull เพื่อตรวจสอบว่ากระเช้าเต็มหรือไม่ พร้อมทั้งคืนค่ากลับ

//เมทอดประจำอ็อบเจ็คชื่อ isEmpty เพื่อตรวจสอบว่ากระเช้าว่างหรือไม่ พร้อมทั้งคืนค่ากลับ

//เมทอดประจำอ็อบเจ็คชื่อ takeIn เพื่อรับผลไม้เข้ามา 1 ผลเพื่อนำไปเก็บในกระเช้า โดยก่อนจะเก็บลงไป จะต้องตรวจสอบก่อนว่ากระเช้าเต็มหรือไม่ ถ้าเต็มให้แสดงข้อความว่า "Basket Full" แต่ถ้าไม่เต็มก็ให้เก็บผลไม้ลงไปในกระเช้า พร้อมทั้งนับจำนวนผลไม้ที่อยู่ในกระเช้าด้วยว่ามีอยู่ที่ผล

//เมทอดประจำอ็อบเจ็คชื่อ takeOut เพื่อนำผลไม้ 1 ผลออกจากกระเช้า โดยก่อนจะนำออกไป จะต้องตรวจสอบก่อนว่ากระเช้าว่างหรือไม่ ถ้าว่างให้แสดงข้อความว่า "Basket Empty" แต่ถ้าไม่ว่างก็ให้นำผลไม้ออกจากกระเช้า พร้อมทั้งนับจำนวนผลไม้ที่อยู่ในกระเช้าด้วยว่ามีอยู่ที่ผล

//เมทอดประจำอ็อบเจ็คชื่อ getTotalWeigh เพื่อคืนค่าน้ำหนักรวมของผลไม้ทั้งหมดในกระเช้า

//เมื่้อดประจำอับเจ็คชื่อ showBasket เพื่อใช้แสดงรายละเอียดของผลไม้ในกระเช้าขึ้นบนจอภาพ

} //End of class

3. คลาส FruitBasket เพื่อใช้เป็นกระเช้าผลไม้ที่จัดตกแต่งอย่างสวยงาม (20 คะแนน)

import java.util.Scanner;

public class FruitBasket {

//เมื่้อดประจำคลาสชื่อ getFruitBasket เพื่อใช้สร้างกระเช้าผลไม้ 1 กระเช้าโดยจะต้องรับค่าชื่อผลไม้ชนิดใดชนิดหนึ่ง (ส้ม แอปเปิ้ล ฯลฯ) น้ำหนักของผลไม้ (เท่ากันทุกผล) และจำนวนของผลไม้ในกระเช้าผ่านทางพารามิเตอร์ พร้อมทั้งคืนค่ากระเช้าผลไม้ที่จัดเก็บเรียบร้อยแล้วกลับมา

//เมื่้อดประจำคลาสชื่อ showFruitBasket ที่รับกระเช้าเข้ามาทางพารามิเตอร์ เพื่อใช้แสดงค่าข้อมูลในกระเช้าขึ้นบนจอภาพ ซึ่งได้แก่ น้ำหนักรวมของกระเช้า (Total Weigh) จำนวนผลไม้ทั้งหมดในกระเช้า (Total Fruit) และรายละเอียดของผลไม้ในกระเช้าแต่ละผล (List of Fruit)

//เมื่อกด main เพื่อใช้ในการจัดเก็บและตกแต่งผลไม้ในกระเช้า

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

//สร้างกระเช้าผลไม้ขึ้นมา 3 กระเช้า (เป็นอาเรย์ที่เป็นประเภท Basket) โดยในการสร้างกระเช้า
นั้นให้เรียกใช้เมื่อกด getFruitBasket ซึ่งกระเช้าที่ 1 เป็นกระเช้าแอปเปิ้ล (Apple) น้ำหนักผล
0.12 กรัม จำนวน 25 ผล กระเช้าที่ 2 เป็นกระเช้าส้ม (Orange) น้ำหนักผลละ 0.1 กรัม จำนวน 30
ผล และ กระเช้าที่ 3 เป็นกระเช้าเมลอน (Melon) น้ำหนักผลละ 0.45 กรัม จำนวน 15 ผล

//สร้างกระเช้าผลไม้เปล่าๆ ขึ้นมาอีก 6 กระเช้า (เป็นอาเรย์ที่เป็นประเภท Basket) แล้วทำการ
หยิบผลไม้จาก 3 กระเช้าแรกที่ได้สร้างไว้ก่อนหน้านี้ลงไปในกระเช้าเปล่าทั้ง 6 กระเช้านั้น โดยให้
หยิบผลไม้จากกระเช้าที่ 1-3 เรียงตามลำดับไปเรื่อยๆ จนกว่ากระเช้าเปล่าจะเต็มทุกกระเช้า
พร้อมทั้งแสดงรายละเอียดของผลไม้ในแต่ละกระเช้าที่จัดเก็บเรียบร้อยแล้วขึ้นบนจอภาพโดยการ
เรียกใช้เมื่อกด showFruitBasket

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

โจทย์ข้อที่ 8 [ระดับเทพ] จงเขียนโปรแกรมภาษาจาวาให้สมบูรณ์เพื่อสร้างห้องติวจาวา ซึ่งก็คือห้อง 302 ตึก 1 คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย โดยจะประกอบไปด้วยคลาส 2 คลาส ได้แก่คลาส student ที่แทนนิสิตแต่ละคน และคลาส room ที่แทนห้องติวจาวา โดยภายในห้องติวจะมีนิสิตนั่งอยู่ข้างใน ซึ่งตำแหน่งที่นั่งของนิสิตแต่ละคนจะจัดให้ตามระดับความเกเรียน (Noob Levels) ของนิสิตคนนั้น ซึ่งจะมี 4 ระดับได้แก่ ระดับที่ 1 เกเรียน (Noob) ระดับที่ 2 เกเรียนเรียกพ่อ (Father Noob) ระดับที่ 3 เกเรียนเทพ (God Noob) ระดับที่ 4 เกเรียนเมพ (Hof Noob) โดยรายละเอียดการทำงาน และรายละเอียดของคลาสทั้งสองมีดังต่อไปนี้ (30 คะแนน)

1. คลาส student มีรายละเอียดดังนี้ (10 คะแนน)

```
public class Student {
```

```
    //แอตทริบิวต์ประจำอ็อบเจ็คแบบ private ชื่อ id เก็บเลขประจำตัวนิสิต
```

```
    //แอตทริบิวต์ประจำอ็อบเจ็คแบบ private ชื่อ name เก็บชื่อนิสิต
```

```
    //แอตทริบิวต์ประจำอ็อบเจ็คแบบ private ชื่อ noob เก็บระดับความเกเรียน
```

```
//ตัวสร้างแบบไม่มีพารามิเตอร์ เพื่อกำหนดค่าเริ่มต้นให้กับแอตทริบิวต์ทุกตัว
```

```
//ตัวสร้างแบบที่รับพารามิเตอร์ 3 ตัวเพื่อกำหนดค่าเริ่มต้นให้กับแอตทริบิวต์ทุกตัว
```

```
//ตัวสร้างแบบที่รับพารามิเตอร์เป็นชนิด student เพื่อกำหนดค่าเริ่มต้นให้กับแอตทริบิวต์ทุกตัว
```

//เมทอดประจำอ็อบเจ็คแบบ public ชื่อ getID ที่ไม่มีพารามิเตอร์ เพื่อคืนเลขประจำตัวนิสิต

//เมทอดประจำอ็อบเจ็คแบบ public ชื่อ getName ที่ไม่มีพารามิเตอร์ เพื่อคืนชื่อนิสิต

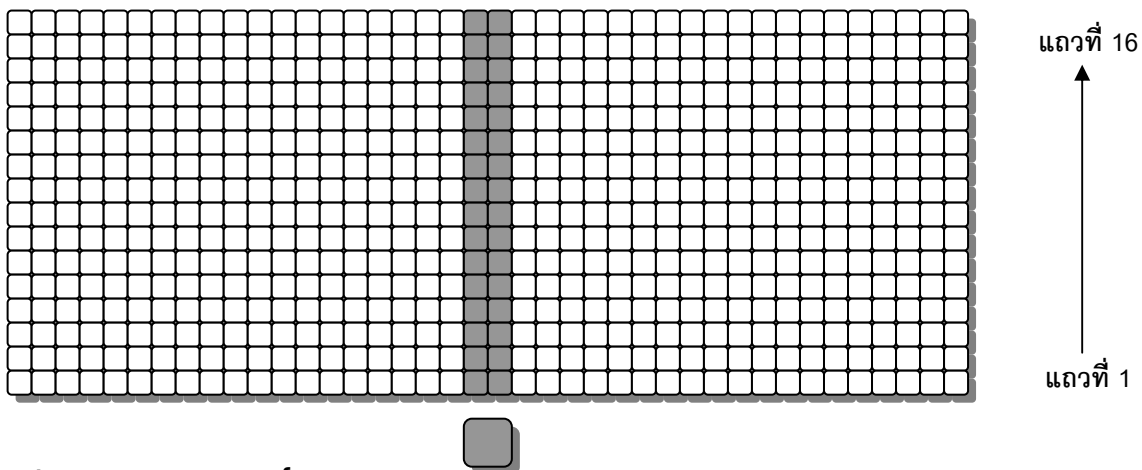
//เมทอดประจำอ็อบเจ็คแบบ public ชื่อ getNoob ที่ไม่มีพารามิเตอร์ เพื่อคืนระดับความเกรียน

//เมทอดประจำอ็อบเจ็คแบบ public ชื่อ set ที่รับพารามิเตอร์ชนิด student เพื่อนำมากำหนด ค่าให้กับแอตทริบิวต์ตัวให้เหมือนกับค่าพารามิเตอร์นั้น

//เมทอดประจำอ็อบเจ็คแบบ public ชื่อ show ที่ไม่มีพารามิเตอร์ เพื่อแสดงค่าของแอตทริบิวต์ตัว ในคลาสนี้ขึ้นบนจอภาพ ซึ่งประกอบไปด้วยเลขประจำตัว ตามด้วยชื่อ และตามด้วยระดับความเกรียน โดยถ้าระดับความเกรียนเป็น 1 ให้แสดงว่า "[1-Noob]" ถ้าระดับความเกรียนเป็น 2 ให้แสดงว่า "[2-Father Noob]" ถ้าระดับความเกรียนเป็น 3 ให้แสดงว่า "[3-God Noob]" ถ้าระดับความเกรียนเป็น 4 ให้แสดงว่า "[4-Hof Noob]" ถ้าไม่เช่นนั้นให้แสดงว่า "[Un- known]" โดยให้ทุกค่าแสดงอยู่ภายในบรรทัดเดียวกัน

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

2. คลาส Room แทนห้องติวจาวา ห้อง 302 ตึก 1 โดยกำหนดให้มี 16 แถว แถวละ 40 ที่นั่ง โดยใน 40 ที่นั่งนั้นจะนั่งได้จริงแค่ 38 ที่นั่งเพราะถูกเว้นไว้สำหรับช่องทางเดินตรงกลางห้อง โดยมีแผนผังห้องตามตัวอย่างที่กำหนดให้ และมีรายละเอียดของโปรแกรมดังต่อไปนี้ (20 คะแนน)



```
public class Room {
```

```
//แอตทริบิวต์ประจำคลาสแบบ public ชื่อ s ที่เป็นอาร์เรย์ 2 มิติ โดยมีชนิดเป็นแบบ student เพื่อใช้สำหรับสร้างที่นั่งให้กับห้องติวจาวา โดยกำหนดให้ห้อง 302 ตึก 1 มี 16 แถว แถวละ 40 ที่นั่ง
```

```
เมทอดประจำคลาสแบบ public void ชื่อ setSeat ที่รับพารามิเตอร์เป็นชนิด student (นิสิต 1 คน) เข้ามาเพื่อนำมาจัดสรรที่นั่งให้กับนิสิตผู้นั้น โดยให้พิจารณาระดับความเกียยณดังนี้
```

- (1) ถ้าระดับความเกียยณเป็น 1 ให้เลือกที่นั่งตั้งแต่แถวที่ 13-16
- (2) ถ้าระดับความเกียยณเป็น 2 ให้เลือกที่นั่งตั้งแต่แถวที่ 9-12
- (3) ถ้าระดับความเกียยณเป็น 3 ให้เลือกที่นั่งตั้งแต่แถวที่ 5-8
- (4) ถ้าระดับความเกียยณเป็น 4 ให้เลือกที่นั่งตั้งแต่แถวที่ 1-4

```
//โดยตำแหน่งที่นั่งนั้นให้จัดสรรเรียงตามลำดับที่นั่งว่างของแต่ละแถวไปเรื่อยๆ จนเต็มแถว และจะไม่มีการจัดสรรที่นั่งบริเวณทางเดินกลางห้อง (ปล่อยว่างเอาไว้)
```

//เมื่อกดประจำคลาสแบบ public void ชื่อ showRoom ที่ไม่รับพารามิเตอร์ เพื่อใช้แสดงแผนผังห้องตัวพร้อมกับระบุตำแหน่งของนิสิตที่กำลังนั่งอยู่ในห้อง ณ ขณะนั้น ตามที่ได้ถูกจัดสรรที่นั่งเอาไว้ โดยถ้าที่นั่งมีนิสิตนั่งให้แสดงตัวอักษรตัวแรกของชื่อนิสิตขึ้นบนแผนผังห้อง เช่น พ, ทร เป็นต้น แต่ถ้าที่นั่งว่างให้แสดงเป็นสัญลักษณ์ชาร์ป (#) ส่วนบริเวณทางเดินให้แสดงเป็นช่องว่าง

//เมื่อกดประจำคลาสแบบ public void ชื่อ showStudent ที่รับพารามิเตอร์เข้ามา 2 ตัว เป็นลำดับแถวและหลักของที่นั่ง เพื่อแสดงข้อมูลของนิสิตที่นั่งอยู่ ณ ตำแหน่งนั้น

//เมื่อกดประจำคลาสแบบ `public void` ชื่อ `showStudent` ที่ไม่รับพารามิเตอร์ เพื่อแสดงข้อมูลของนิสิตทุกคนที่อยู่ในห้องติว พร้อมทั้งระบุตำแหน่งที่นั่งของนิสิตแต่ละคน

//เมื่อกด `main(...)` เพื่อประมวลผลการจัดสรรห้องติวดังต่อไปนี้

- (1) กำหนดให้แอตทริบิวต์ `s` ที่เป็นอาร์เรย์ 2 มิติชนิด `student` ทุกช่อง มีค่าเริ่มต้นเป็นที่นั่งว่าง (แต่ละช่องมีเลขประจำตัวนิสิตเป็น 0 ชื่อเป็น "" และระดับความเกรียนเป็น 0
- (2) สร้างนิสิตจำนวน 5 คนได้แก่นิสิตชื่อ Akeudom, Phakhin, Taparb, Ong-ard และ Wongyos ตามลำดับ และมีระดับความเกรียนเป็น 1, 1, 2, 3, และ 4 ตามลำดับ โดยให้กำหนดเลขประจำตัวนิสิตได้อย่างอิสระ
- (3) จัดสรรที่นั่งในห้องติวให้กับนิสิตทั้ง 5 คน โดยเรียกใช้เมทอด `setSeat(...)`
- (4) แสดงแผนผังห้องติวเพื่อแสดงตำแหน่งที่นั่งของนิสิตทั้ง 5 คน โดยเรียกใช้เมทอด `showRoom(...)`
- (5) แสดงข้อมูลของนิสิตทุกคนในห้องติวโดยเรียกใช้เมทอด `showStudent(...)`

} //End of class