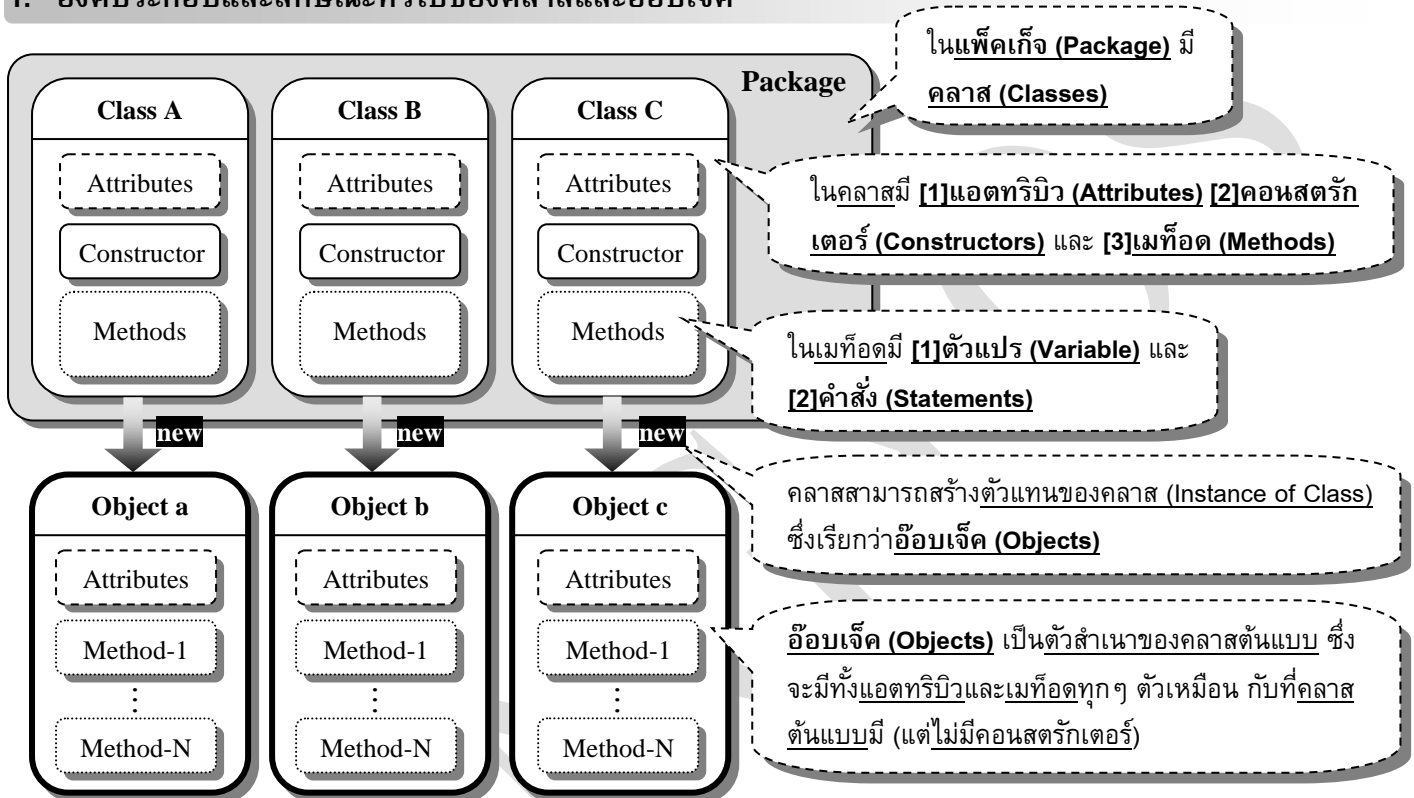


CHAPTER 3

คลาสและอ็อบเจกต์เบื้องต้น (Introduction to Classes and Objects)

1 หลักการของคลาสและอ็อบเจกต์ (Classes and Objects)

1. องค์ประกอบและลักษณะทั่วไปของคลาสและอ็อบเจกต์



2. นิยามและคำศัพท์ที่เกี่ยวข้องกับคลาสและอ็อบเจกต์

การเขียนโปรแกรมในบทที่ 3 นี้ จะเป็นมุมมองแบบใหม่ที่เรียกว่า **การเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming)** ซึ่งมีองค์ประกอบดังต่อไปนี้

คำศัพท์	นิยามและความหมาย
แพ็คเกจ (Package)	สิ่งที่เก็บรวบรวมคลาสและห่อหุ้มคลาสเอาไว้เป็นกลุ่มๆ (จะยังไม่กล่าวถึงมากนัก)
คลาส (Class)	ต้นแบบของโปรแกรมซึ่งภายในประกอบไปด้วย แอตทริบิวต์ คอนสตรัคเตอร์ และ
แอตทริบิวต์ (Attribute)	ตัวแปรของคลาส หรือตัวแปรที่ประกาศอยู่ภายในคลาสแต่อยู่นอกเมทอดใดๆ
คอนสตรัคเตอร์ (Constructor)	ส่วนที่กำหนดค่าเริ่มต้นของคลาส ซึ่งจะกำหนดค่าให้กับแอตทริบิวต์ทุกตัวในคลาส
เมทอด (Method)	ส่วนการทำงานย่อยของคลาส หรือโปรแกรมย่อยที่บ่งบอกการกระทำของคลาส
ตัวแปร (Variable)	ตัวแปรของเมทอด หรือตัวแปรใดๆ ที่ประกาศภายในเมทอดใดเมทอดหนึ่ง
คำสั่ง (Statement)	คำสั่งภายในเมทอด หรือคำสั่งต่างๆ ที่ปรากฏภายในเมทอดใดเมทอดหนึ่ง
อ็อบเจกต์ (Object)	ตัวแทนของคลาส ซึ่งจะมีทุกอย่างเหมือนกับคลาสต้นแบบมี

3. การแจกแจงส่วนประกอบของคลาสและอ็อบเจ็กต์

```

1 public class Num {
2     public int x;
3     public 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 static int addAB(int a, int b) {
11        return a + b;
12    }
13    public double addXY() {
14        return x + 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         double a = obj.addXY();
6         int b = Num.addAB(3, 5);
7         System.out.println(a);
8         System.out.println(b);
9     }
10 }

```

ผลลัพธ์ที่ได้

หมายเหตุ ในบทนี้จะเน้นการศึกษาและเรียกใช้งานคลาสต้นแบบที่กำหนดมาให้ โดยไม่เน้นการเขียนคลาสต้นแบบใหม่

4. การประกาศและสร้างคลาส

คลาสมือถือประกอบ 2 ส่วนใหญ่ได้แก่ หัวคลาส และ ตัวคลาส โดยมีรายละเอียดดังต่อไปนี้

```

1 public class Data {
2     private long x;
3     private String y;
4     public Data() {
5         x = 0L; y = "";
6     }
7     public long add(int n) {
8         return x + n;
9     }
10 }

```

ตัวบ่งคุณลักษณะ

ชื่อคลาส

หัวคลาส

ตัวคลาส

แอตทริบิวต์

คอนสตรัคเตอร์

เมทอด

คลาสที่กล่าวถึงในที่นี้คือคลาสต้นแบบ (Prototype Class) ซึ่งโดยทั่วไปจะไม่เมทอด main(...)

5. การประกาศและสร้างแอตทริบิวต์

การประกาศและสร้างแอตทริบิวต์จะคล้ายกับการประกาศและสร้างตัวแปรในบทที่ 2 โดยมีรายละเอียดดังนี้

```
1 public class Data {
2     private long x;
3     private String y;
4     ...
5 }
```

แอตทริบิวต์ คือ ตัวแปรที่ประกาศอยู่ในคลาสแต่อยู่นอกเมทอดใดๆ

โดยทั่วไปจะกำหนดให้แอตทริบิวต์เป็น private ไม่ใส่ static

6. การประกาศและสร้างคอนสตรักเตอร์

การประกาศและสร้างคอนสตรักเตอร์จะคล้ายกับการประกาศและสร้างเมทอด โดยมีรายละเอียดดังนี้

```
1 public class Data {
2     private long x;
3     private String y;
4     public Data() {
5         x = 0L; y = "";
6     }
7     public Data(long a, String b) {
8         x = a; y = b;
9     }
10    ...
11 }
```

ชื่อคลาส

แอตทริบิวต์

คอนสตรักเตอร์เริ่มต้น (Default Constructor)

คอนสตรักเตอร์รายละเอียด (Detail Constructor)

คอนสตรักเตอร์ (Constructor) คือ เมทอดที่มีชื่อเดียวกับชื่อคลาส ไม่มี Return Type ไม่ใส่ static สามารถ Overload ได้

7. การประกาศและสร้างเมทอด

การประกาศและสร้างเมทอดได้กล่าวไปอย่างละเอียดแล้วในบทที่ 6 ซึ่งกล่าวสรุปอีกครั้งได้ดังนี้

```
1 public class Data {
2     ...
3     public long add(int n) {
4         return x + n;
5     }
6     ...
7 }
```

โดยทั่วไปจะกำหนดให้คอนสตรักเตอร์และเมทอดเป็น public ไม่ใส่ static (ทั้งนี้เมทอดอาจใส่ static ได้ขึ้นกับการใช้งาน)

คลาสต้นแบบจะไม่มีเมทอด main(...)

8. การประกาศและสร้างอ็อบเจ็กต์

การประกาศและการเรียกใช้งานคลาสต้นแบบผ่านทางอ็อบเจ็กต์ทำได้ดังนี้

```
1 public class TestData {
2     public static void main(String[] args) {
3         Data a = new Data();
4         Data b = new Data(1L, "Bank");
5         System.out.println(a.add(5));
6     }
7 }
```

คลาสที่กล่าวถึงในที่นี้คือคลาสประมวลผล (Processing Class) ซึ่งโดยทั่วไปจะมีเมทอด main(...)

<ชื่อคลาส> <ชื่ออ็อบเจ็กต์> = new <ชื่อคลาส> (...);

<ประเภทข้อมูล> <ชื่อตัวแปร> = <ชื่ออ็อบเจ็กต์>.<แอตทริบิวต์ / เมทอด>;

เมื่อเริ่มสร้างอ็อบเจ็กต์ให้พิจารณาที่คอนสตรักเตอร์ของคลาสต้นแบบ

โจทย์ข้อที่ 1 [ระดับง่าย] จงพิจารณาการประกาศและสร้างองค์ประกอบของคลาสต่อไปนี้ผิด (✗) หรือถูก (✓) ตามหลักไวยากรณ์ของภาษาจาวาพร้อมทั้งบอกเหตุผลกำกับ (โจทย์แต่ละข้อจะต่อเนื่องกัน) (12 คะแนน)

- 1) ☐

```
public static class Grean {
    public String x;
    private boolean y;
    int z;
}
```
- 2) ☐

```
Grean() {
    x = "";
    y = false;
    z = 0;
}
```
- 3) ☐

```
public void Grean(String a, boolean b, int c) {
    x = a; y = b; z = c;
}
```
- 4) ☐

```
public static Grean(int m, int n) {
    x = m + "" + n;
    y = m == n;
    z = m + n;
}
```
- 5) ☐

```
public grean(double k) {
    x = k + "";
    y = k == 1.0;
    z = (int) k;
}
```
- 6) ☐

```
public Grean(long k) {
    x = k + "";
    y = k == 1;
    z = 0;
}
```
- 7) ☐

```
public int getZ() {
    return z;
}
```
- 8) ☐

```
public boolean getYorZ(int n) {
    if (n == 1) {
        return y;
    }
    return z;
}
```
- 9) ☐

```
public boolean isEqual(int n) {
    int z = getZ();
    boolean x = z == n;
    return x;
}
```
- 10) ☐

```
public static double getReal() {
    double d = getZ();
    return d;
}
```
- 11) ☐

```
public static String toString(int n) {
    String s = "n = " + n;
    return s;
}
```
- 12) ☐

```
public void setXandZ(String x, int z) {
    this.x = x;
    this.z = z;
}
```

ข้อที่ 1 เป็นส่วนของหัวคลาสและแอตทริบิวต์

ข้อที่ 2-6 เป็นส่วนของคอนสตรักเตอร์

ข้อที่ 7-12 เป็นส่วนของเมทอด

โจทย์ข้อที่ 2 [ระดับปานกลาง] จงพิจารณาการประกาศและเรียกใช้งานอ็อบเจกต์ต่อไปนี้ผิด (✗) หรือถูก (✓)
ตามหลักไวยากรณ์ของภาษาจาวาพร้อมทั้งบอกเหตุผลกำกับ (โจทย์แต่ละข้อจะต่อเนื่องกัน) (11 คะแนน)

```

1 public class Real {
2     public double x;
3     public Real() {
4         x = 0.0;
5     }
6     public Real(double n) {
7         x = n;
8     }
9     public double getX(){
10        return x;
11    }
12    public String showPowX(int n) {
13        double d = Math.pow(x, n);
14        return (int) d + "";
15    }
16 }
```

```

1 public class Login {
2     private int id;
3     private int pass;
4     public Login() {
5         id = 0; pass = 0;
6     }
7     public Login(int n, int p) {
8         id = n; pass = p;
9     }
10    public boolean isPass(int p) {
11        return pass == p;
12    }
13    public static int nextID(int n) {
14        return n + 1;
15    }
16 }
```

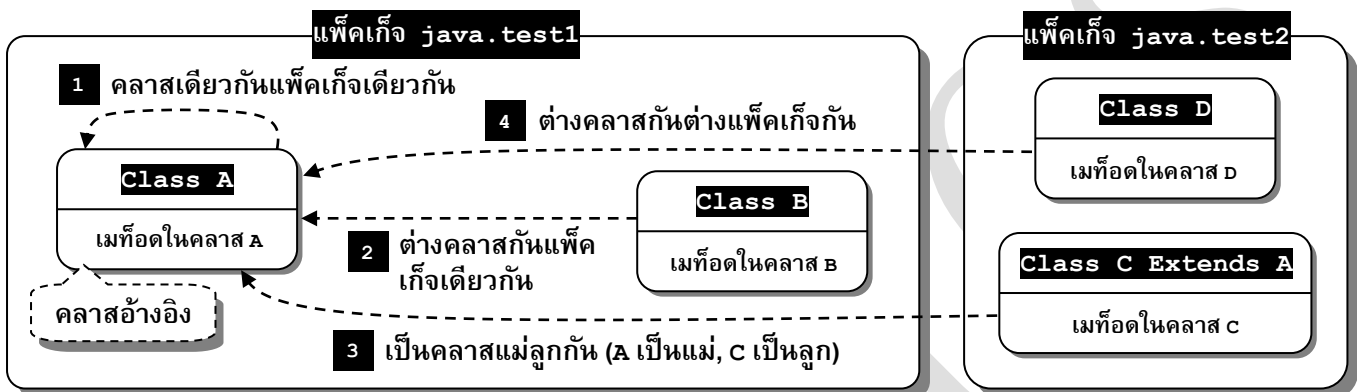
- 1) ☐ public class MyDemo {
 public static void Main(String[] args) {
- 2) ☐ Real a = new Real(1);
 Real b = new Real();
 Real c = new Real(1.0);
- 3) ☐ Real d = new Real(1, 1);
 Real e = new Real("1");
 Real f = new Real(5.0F);
- 4) ☐ System.out.println(a.getX());
 System.out.println(a.x);
 System.out.println(b.x);
- 5) ☐ int var = (int) c.x;
 System.out.println(f.showPowX(var));
 getHalf(c.getX());
- 6) ☐ Login u = new Login(1001, 1234567);
 Login v = new Login(1002, "abcdef");
 Login w = new Login();
- 7) ☐ Login x = new Login(1003);
 Login y = new Login(1004, 1005);
 Login z = new Login(6030126821, 1);
- 8) ☐ System.out.println(u.isPass(1234));
 System.out.println(Login.isPass(1234));
 System.out.println(Login.nextID(1234));
- 9) ☐ System.out.println(y.id + "," + y.pass);
 System.out.println(getHalf(y.nextID(13)));
 System.out.println(w.getHalf(Login.nextID(5)));
 showDemo("Hello Bank");
 new MyDemo().showDemo("Hello Bank");
 }
- 10) ☐ private static int getHalf(double n) {
 int x = (int) n / 2;
 return x;
 }
- 11) ☐ public void showDemo(String s) {
 System.out.println(s)
 }
}

2 คุณสมบัติและการใช้งานคลาสและอ็อบเจกต์

1. ขอบเขตการเรียกใช้งานแอตทริบิวต์และเมทอด

ตัวบ่งคุณลักษณะที่ใช้กำหนดขอบเขตการเรียกใช้แอตทริบิวต์และเมทอด ได้แก่ **public**, **private** และ **protected** ซึ่งจะเป็นตัวบอกว่าแอตทริบิวต์และเมทอดจะถูกเรียกใช้งานจากที่ไหนได้บ้าง (**Where**) โดยมีรายละเอียดดังนี้

- 1) **public** หมายถึง เป็นสาธารณะ สามารถเรียกใช้ได้จากทุกคลาส (คลาสใดๆ ทั่วทุกหนทุกแห่ง)
- 2) **private** หมายถึง เป็นส่วนตัว สามารถเรียกใช้ได้เฉพาะภายในคลาสเดียวกัน (คลาสของตัวเอง)
- 3) **protected** หมายถึง เป็นสิ่งที่ถูกปกป้องไว้ สามารถเรียกใช้ได้จากคลาสที่มีความสัมพันธ์กัน (คลาสลูกหลาน)



สรุปเป็นตารางขอบเขตการเรียกใช้งานได้ดังนี้

รายละเอียดของ ตัวบ่งคุณลักษณะ	คลาสเดียวกัน แพ็คเกจเดียวกัน	ต่างคลาสกัน แพ็คเกจเดียวกัน	เป็นสับคลาสกัน (คลาสแม่ลูก)	ต่างคลาสกัน ต่างแพ็คเกจกัน
แอตทริบิวต์หรือเมทอดเป็น public	✓	✓	✓	✓
แอตทริบิวต์หรือเมทอดเป็น private	✓	✗	✗	✗
แอตทริบิวต์หรือเมทอดเป็น protected	✓	✓	✓	✗
แอตทริบิวต์หรือเมทอดไม่ระบุมีดัดฟายเออร์	✓	✓	✗	✗

2. วิธีการเรียกใช้งานแอตทริบิวต์และเมทอด

ตัวบ่งคุณลักษณะที่ใช้กำหนดวิธีการเรียกใช้หรือกำหนดค่าประจำคลาสหรือค่าประจำอ็อบเจกต์ของแอตทริบิวต์และเมทอด ได้แก่ **static** ซึ่งจะเป็นตัวบอกว่าแอตทริบิวต์และเมทอดจะถูกเรียกใช้งานอย่างไรได้บ้าง (**How**) โดยรายละเอียดดังนี้

- 1) การเรียกใช้เมื่ออยู่ภายในคลาสเดียวกัน จะใช้วิธีการเรียกชื่อแอตทริบิวต์หรือเมทอดนั้นโดยตรงตามหลักการในบทที่ 6 ที่ผ่านมา ซึ่งจะพิจารณาแอตทริบิวต์หรือเมทอดที่มี **static** และไม่มี **static** ตามเงื่อนไขต่อไปนี้

		เมทอดผู้ถูกเรียก	
เมทอดผู้เรียก	static	ไม่มี	มี
	ไม่มี	✓	✓
	มี	✗	✓
	มี	✗	✓

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

2) การเรียกใช้เมื่ออยู่ต่างคลาสเดียวกัน จะใช้วิธีการเรียกด้วยชื่อคลาสหรือชื่ออ็อบเจกต์ ซึ่งจะพิจารณาแอดทริบิวต์หรือเมทอดที่มี static และไม่มี static ตามเงื่อนไขต่อไปนี้

(1) เรียกแอดทริบิวต์และเมทอดที่มี static โดยจะเรียกผ่านทางคลาส (Class)

```
<ชื่อคลาส> . <ชื่อแอดทริบิวต์>;
```

ขึ้นต้นพิมพ์ใหญ่ ลงท้ายไม่มีวงเล็บ เช่น Math.PI

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

ขึ้นต้นพิมพ์ใหญ่ ลงท้ายมีวงเล็บ เช่น Math.pow(2, 3)

(2) เรียกแอดทริบิวต์และเมทอดที่ไม่มี static โดยจะเรียกผ่านทางอ็อบเจกต์ (Object)

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

ต้องสร้างอ็อบเจกต์ก่อนการเรียกใช้งาน

```
<ชื่ออ็อบเจกต์> . <ชื่อแอดทริบิวต์>;
```

ขึ้นต้นพิมพ์เล็ก ลงท้ายไม่มีวงเล็บ เช่น obj.id, g.y

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

ขึ้นต้นพิมพ์เล็ก ลงท้ายมีวงเล็บ เช่น std.get(123)

โจทย์ข้อที่ 3 [ระดับง่าย] จงพิจารณาตัวบ่งคุณลักษณะของแอดทริบิวต์และเมทอดต่อไปนี้และทำเครื่องหมายผิด (✗) หรือถูก (✓) ตามคุณสมบัติที่ถูกกำหนดให้กับแอดทริบิวต์หรือเมทอดนั้น (10 คะแนน)

ข้อ	รายละเอียดของแอดทริบิวต์หรือหัวเมทอด	แอดทริบิวต์ (A) หรือเมทอด (M)	ขอบเขตการเรียกใช้				วิธีการเรียกใช้			
			คลาสเดียวกัน แอดแอคเดียวกัน	คลาสต่างกัน แอดแอคเดียวกัน	คลาสต่างกัน เป็นสับคลาสกัน	คลาสต่างกัน แอดแอคต่างกัน	เรียกโดยตรงด้วยเมทอดมี static	เรียกโดยตรงด้วยเมทอดไม่มี static	เรียกผ่านคลาส	เรียกผ่านอ็อบเจกต์
1.	public void setID(int i)									
2.	private int id									
3.	protected static int getID()									
4.	public static final String name = "A"									
5.	public double salary									
6.	protected double getSalary(int pos)									
7.	private static boolean status = true									
8.	public static boolean getStatus()									
9.	void setName(String n)									

ข้อ	รายละเอียดของแอตทริบิวหรือหัตถ์เมทอด	แอตทริบิว (A) หรือเมทอด (M)	ขอบเขตการเรียกใช้				วิธีการเรียกใช้			
			คลาสเดียวกัน แพ็คเก็ตเดียวกัน	คลาสต่างกัน แพ็คเก็ตเดียวกัน	คลาสต่างกัน เป็นสับคลาสกัน	คลาสต่างกัน แพ็คเก็ตต่างกัน	เรียกโดยตรงด้วยเมทอดมี static	เรียกโดยตรงด้วยเมทอดไม่มี static	เรียกผ่านคลาส	เรียกผ่านอ็อบเจกต์
10.	<code>private String toString()</code>									
11.	<code>public static int parseInt(String s)</code>									
12.	<code>protected String readLine()</code>									
13.	<code>private static final int MAX_VALUE</code>									
14.	<code>double speed</code>									
*15.	<code>private Date birthDay</code>									
16.	<code>protected double average(int x, int y)</code>									
17.	<code>public final int indexOf(String s)</code>									
18.	<code>public static void num</code>									
*19.	<code>private static Dice getNewDice(int a)</code>									
*20.	<code>protected double distanceTo(Point p)</code>									

โจทย์ข้อที่ 4 [ระดับง่าย] จงใช้ระบุประเภทของแอตทริบิวหรือเมทอดจากการเรียกใช้ต่อไปนี้ (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>s.indexOf("bank")</code>	
3.	<code>v.x</code>	
4.	<code>underTen(a, b, c, d)</code>	
5.	<code>System.getProperties()</code>	
6.	<code>Integer.MAX_VALUE</code>	
7.	<code>kb.nextInt()</code>	
8.	<code>circle.setPoint(x, y)</code>	
9.	<code>Color.RED</code>	
10.	<code>a.appendArrays(x, y)</code>	

ข้อ	แอตทริบิวหรือเมทอด	คำตอบ
11.	<code>noob.level</code>	
12.	<code>Triangle.borders(a, b)</code>	
13.	<code>stdin.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>x2018.name</code>	

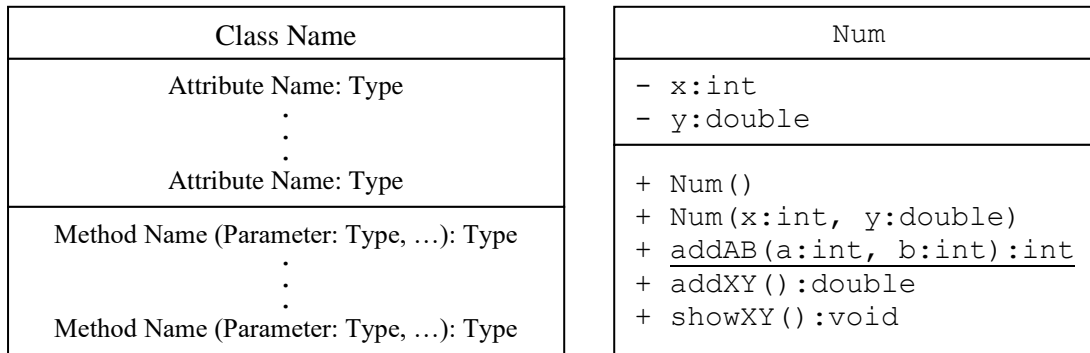
3

การนำเสนอและอธิบายองค์ประกอบของคลาสและอ็อบเจกต์

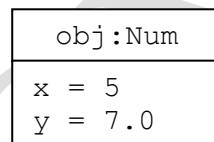
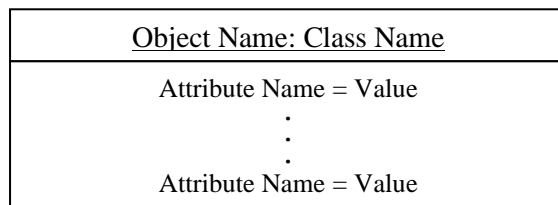
1. การนำเสนอและอธิบายองค์ประกอบของคลาสและอ็อบเจกต์ด้วยยูเอ็มแอล (UML)

การออกแบบคลาสและอ็อบเจกต์ในปัจจุบันนั้นจะนิยมเขียนอยู่ในรูปแบบของ แผนภาพคลาส (Class Diagram) และ แผนภาพอ็อบเจกต์ (Object Diagram) ซึ่งเป็นส่วนหนึ่งของ สัญลักษณ์ยูเอ็มแอล โดยมีรายละเอียดดังต่อไปนี้

1) แผนภาพคลาส (Class Diagram)



2) แผนภาพอ็อบเจกต์ (Object Diagram)



- 1) เครื่องหมาย + คือ public
- 2) เครื่องหมาย - คือ private
- 3) เครื่องหมาย # คือ protected
- 4) เครื่องหมาย ~ คือ package
- 5) การขีดเส้นใต้ คือ static

2. การนำเสนอและอธิบายองค์ประกอบของคลาสด้วยจาวาเอพีไอ (Java API)

ผู้พัฒนาภาษาจาวาได้ใช้ Java API ในการอธิบายรายละเอียดต่างๆ ของคลาสต้นแบบที่อยู่ในไลบรารี (Libraries) ที่จาวาส่งสร้างขึ้นโดยประกอบไปด้วย (1) คำอธิบายโดยสรุปของแอตทริบิวต์หรือฟิลด์ (Field Summary) (2) คำอธิบายโดยสรุปของคอนสตรัคเตอร์ (Constructor Summary) และ (3) คำอธิบายโดยสรุปของเมทอด (Method Summary) ซึ่งจากตัวอย่างของคลาส `Num` ที่กล่าวไว้ข้างต้น (หน้า 97) สามารถอธิบายเป็น Java API ได้ดังนี้ (ขออธิบายเป็นภาษาอังกฤษ)

Class Num

Field Summary

Modifier and Type	Field and Description
int	x The value of integer x
double	y The value of real number y

Constructor Summary

Constructor and Description
Num () Constructs a new Num that produces default values of field

Constructor Summary**Constructor and Description**

Num(int m, double n)

Constructs a new Num that produces detail values of field from parameter

Method Summary

Modifier and Type	Method and Description
static int	addAB(int a, int b) Return int value from the addition of parameter a and b
double	addXY() Return double value from the addition of field x and y
void	showXY() Show out the value of field x and y to the screen (monitor)

โจทย์ข้อที่ 5 [ระดับง่าย] จงศึกษารายละเอียด UML ของคลาส Real แล้วทำตามคำสั่งต่อไปนี้ (14 คะแนน)

Real
- num:double
+ Real() + Real(n:double) + add(n:double):double + sub(n:double):double + mul(n:double):double + div(n:double):double + equal(x:double, y:double):boolean

ไม่อนุญาตให้ใช้เครื่องหมายบวก ลบ คูณ หาร ในการประมวลผลกับโจทย์ข้อนี้

จาก UML ของคลาส Real จงประกาศและเรียกใช้งานแอตทริบิวต์หรือเมทอดให้ถูกต้อง

ข้อ	คำอธิบาย	คำสั่ง
1.	สร้างอ็อบเจกต์ชื่อ x โดยกำหนดค่าเริ่มต้นเป็น 0.0	
2.	สร้างอ็อบเจกต์ชื่อ y โดยกำหนดค่าเริ่มต้นเป็น 2.7	
3.	คำนวณและแสดงผลลัพธ์ของ $y + 5.7$	
4.	คำนวณและแสดงผลลัพธ์ของ $x - y * 9$	
5.	คำนวณและแสดงผลลัพธ์ของ $(x - 2) + (y / 3)$	
6.	เปรียบเทียบว่า $x + 4$ เท่ากับ $y * 2.5$ หรือไม่	
7.	เรียกใช้งานแอตทริบิวต์ num ผ่านอ็อบเจกต์ y	

โจทย์ข้อที่ 6 [ระดับง่าย] จงศึกษาการทำงานของคลาส `Circle` ซึ่งเป็นตัวแทนของวงกลมใด ๆ และเขียนคลาส `Circle` และคลาส `TestCircle` เพื่อประมวลผลโปรแกรมตามที่โจทย์กำหนดให้ (30 คะแนน)

Class Circle

Field Summary

Modifier and Type	Field and Description
double	radius The radius of circle
double	centerX The coordinate x of center position in circle
double	centerY The coordinate y of center position in circle

Constructor Summary

Constructor and Description
<code>Circle()</code> Creates an empty instance of Circle.
<code>Circle(double r)</code> Creates a new instance of Circle with a specified radius.
<code>Circle(double x, double y, double r)</code> Creates a new instance of Circle with a specified position and radius.

Method Summary

Modifier and Type	Method and Description
double	<code>getArea()</code> Return the area of circle.
double	<code>getCircumference()</code> Return the circumference of circle.
double	<code>distanceTo(double a, double b)</code> Return the distance between point of (centerX, centerY) and (a, b).
boolean	<code>isInCircle(double a, double b)</code> Return true if point (a, b) is in the circle, otherwise return false.
String	<code>toString()</code> Returns a string representation of this Circle object.

สร้างคลาส `TestCircle` โดยประกอบไปด้วยรายละเอียดดังต่อไปนี้

- 1) สร้างวงกลม `c1` ที่มีจุดศูนย์กลางเป็น (0, 0) และมีรัศมีเป็น 14.0 วงกลม `c2` ที่มีจุดศูนย์กลางเป็น (-9, 11) และมีรัศมีเป็น 8.5 พร้อมทั้งเรียกใช้เมทอดที่เหมาะสมเพื่อแสดงรายละเอียดขึ้นบนจอภาพ (4 คะแนน)
- 2) คำนวณหาพื้นที่วงกลมและเส้นรอบวงของวงกลม `c1` และ `c2` พร้อมแสดงผลลัพธ์บนจอภาพ (2 คะแนน)
- 3) คำนวณหาระยะห่างระหว่างจุด (12, 4) กับจุดศูนย์กลางวงกลม `c1` พร้อมแสดงผลลัพธ์บนจอภาพ (2 คะแนน)
- 4) ตรวจสอบว่าจุด (17, -16) อยู่ภายในวงกลม `c2` หรือไม่ พร้อมแสดงผลลัพธ์บนจอภาพ (2 คะแนน)

โจทย์ข้อที่ 7 [ระดับง่าย] จงสร้างคลาส Rectangle ตามรายละเอียดต่อไปนี้ และสร้างคลาส RectangleDemo ที่เป็นคลาสประมวลผลเพื่อเรียกใช้งานคลาส Rectangle (50 คะแนน)

คลาส Rectangle

- 1) แอตทริบิวต์ width แบบ private เพื่อใช้แทนความกว้างของสี่เหลี่ยม (2.5 คะแนน)
- 2) แอตทริบิวต์ height แบบ private เพื่อใช้แทนความสูงของสี่เหลี่ยม (2.5 คะแนน)
- 3) ตัวสร้างแบบที่ไม่มีพารามิเตอร์เพื่อกำหนดค่าให้กับแอตทริบิวต์ทุกตัวของคลาสนี้ตามค่าเริ่มต้น (Default) ของประเภทของแอตทริบิวต์แต่ละตัว เช่น int ให้เป็น 0 เป็นต้น (5 คะแนน)
- 4) ตัวสร้างแบบที่มีพารามิเตอร์ 2 ตัว เพื่อกำหนดค่าให้กับแอตทริบิวต์ทุกตัวของคลาสนี้ตามค่าและประเภทข้อมูลที่ควรจะเป็น (5 คะแนน)
- 5) เมท็อด getArea (...) แบบ public ที่ไม่มี static และไม่รับพารามิเตอร์ใด ๆ เพื่อคำนวณและคืนค่าพื้นที่ของสี่เหลี่ยม (5 คะแนน)
- 6) เมท็อด getBorder (...) แบบ public ที่ไม่มี static และไม่รับพารามิเตอร์ใด ๆ เพื่อคำนวณและคืนค่าความยาวเส้นรอบรูปสี่เหลี่ยม (5 คะแนน)
- 7) เมท็อด getDiagonal (...) แบบ public ที่ไม่มี static และไม่รับพารามิเตอร์ใด ๆ เพื่อคำนวณและคืนค่าความยาวเส้นทแยงมุมเส้นใดเส้นหนึ่งของสี่เหลี่ยม (5 คะแนน)
- 8) เมท็อด equals (...) แบบ public ที่ไม่มี static โดยรับพารามิเตอร์ที่เป็นความกว้างและความสูงของสี่เหลี่ยมหนึ่งรูปใด ๆ เข้ามา เพื่อตรวจสอบว่าสี่เหลี่ยมเท่ากันหรือไม่ (5 คะแนน)
- 9) เมท็อด toString (...) แบบ public ที่ไม่มี static และไม่รับพารามิเตอร์ใด ๆ เพื่อนำแอตทริบิวต์ทุกตัวของคลาสนี้มาเรียงต่อกันในรูปของสตริง 1 ชุดพร้อมทั้งคืนค่าสตริงนั้น (5 คะแนน)

คลาส RectangleDemo

สร้างเมท็อด main (...) เพื่อเรียกใช้งานเมท็อดต่าง ๆ ทุกเมท็อดในคลาส Rectangle (10 คะแนน)

โจทย์ข้อที่ 8 [ระดับง่าย] จงประกาศและสร้างแอตทริบิวต์ ตัวสร้าง (คอนสตรักเตอร์) และเมทอดของคลาสที่เป็นตัวแทนของสามเหลี่ยมหน้าจั่วใด ๆ ชื่อ Triangle ตามรายละเอียดที่กำหนดให้ต่อไปนี้ (40 คะแนน)

คลาส Triangle

- 1) แอตทริบิวต์ชื่อ base แบบ private เพื่อใช้แทนความยาวฐานของสามเหลี่ยมหน้าจั่ว (2.5 คะแนน)
- 2) แอตทริบิวต์ชื่อ height แบบ private เพื่อใช้แทนความสูงของสามเหลี่ยมหน้าจั่ว (2.5 คะแนน)
- 3) ตัวสร้างแบบที่ไม่มีพารามิเตอร์เพื่อกำหนดค่าให้กับแอตทริบิวต์ทุกตัวของคลาสนี้ตามค่าเริ่มต้น (Default) ของประเภทของแอตทริบิวต์แต่ละตัว เช่น int ให้เป็น 0 เป็นต้น (5 คะแนน)
- 4) ตัวสร้างแบบที่มีพารามิเตอร์ 2 ตัว เพื่อกำหนดค่าให้กับแอตทริบิวต์ทุกตัวของคลาสนี้ตามค่าและประเภทข้อมูลที่ควรจะเป็น (5 คะแนน)
- 5) เมทอด getArea(...) แบบ public ที่ไม่มี static และไม่รับพารามิเตอร์ใด ๆ เพื่อบำหนดและคืนค่าพื้นที่ของสามเหลี่ยมหน้าจั่ว (5 คะแนน)
- 6) เมทอด getHalfArea(...) แบบ public ที่ไม่มี static และไม่รับพารามิเตอร์ใด ๆ เพื่อบำหนดและคืนค่าพื้นที่ครึ่งหนึ่งของสามเหลี่ยมหน้าจั่ว (ให้เรียกใช้เมทอด getArea(...)) (5 คะแนน)
- 7) เมทอด getBorder(...) แบบ public ที่ไม่มี static และไม่รับพารามิเตอร์ใด ๆ เพื่อบำหนดและคืนค่าความยาวเส้นรอบรูปสามเหลี่ยมหน้าจั่ว (5 คะแนน)
- 8) เมทอด equals(...) แบบ public ที่ไม่มี static โดยรับพารามิเตอร์ที่เป็นความยาวฐานและความสูงของสามเหลี่ยมหน้าจั่วอีกหนึ่งรูป เพื่อตรวจสอบว่าสามเหลี่ยมหน้าจั่วเท่ากันหรือไม่ (5 คะแนน)
- 9) เมทอด toString(...) แบบ public ที่ไม่มี static และไม่รับพารามิเตอร์ใด ๆ เพื่อนำแอตทริบิวต์ทุกตัวของคลาสนี้มาเรียงต่อกันในรูปของสตริง 1 ชุดพร้อมทั้งคืนค่าสตริงนั้น (5 คะแนน)

คลาส TriangleDemo

สร้างเมทอด main(...) เพื่อเรียกใช้งานเมทอดต่าง ๆ ทุกเมทอดในคลาส Triangle (10 คะแนน)

โจทย์ข้อที่ 9 [ระดับปานกลาง] จงประกาศและสร้างแอตทริบิวต์ ตัวสร้าง (คอนสตรักเตอร์) และเมทอดของคลาสที่เป็นตัวแทนของหนังสือใด ๆ ชื่อ Book ตามรายละเอียดที่กำหนดให้ต่อไปนี้ (20 คะแนน)

คลาส Book

- 1) แอตทริบิวต์ชื่อ title แบบ private เพื่อเก็บชื่อหนังสือ (1 คะแนน)
- 2) แอตทริบิวต์ชื่อ author แบบ private เพื่อเก็บชื่อผู้แต่ง (ประกาศเป็นอาร์เรย์) (2 คะแนน)
- 3) แอตทริบิวต์ชื่อ price แบบ private เพื่อเก็บราคาหนังสือ (1 คะแนน)

- 4) แอดทริบิวต์ `year` แบบ `private` เพื่อเก็บปีที่พิมพ์หนังสือ (1 คะแนน)
- 5) ตัวสร้างแบบที่มีพารามิเตอร์ 4 ตัว เพื่อกำหนดค่าให้กับแอดทริบิวต์ทุกตัวของคลาสนี้ตามค่าและประเภทข้อมูลที่ควรจะเป็น (ชื่อผู้แต่งให้รับเข้ามาเป็นอาร์เรย์) (10 คะแนน)
- 6) เมทอด `getPriceAfterDiscount(...)` แบบ `public` ที่ไม่มี `static` เพื่อรับเปอร์เซ็นต์ส่วนลดเข้ามาทางพารามิเตอร์ พร้อมทั้งคำนวณและคืนค่าราคาหนังสือหลังจากหักส่วนลดออกไปแล้ว (5 คะแนน)

คลาส BookDemo

สร้างเมทอด `main(...)` เพื่อเรียกใช้งานเมทอดต่าง ๆ ทุกเมทอดในคลาส `Book` (10 คะแนน)

โจทย์ข้อที่ 10 [ระดับปานกลาง] จงสร้างคลาส `BankAccount` ที่สมบูรณ์ ซึ่งเป็นคลาสที่ใช้ในการจัดการกับบัญชีธนาคาร และสร้างคลาส `TestBankAccount` เพื่อใช้ประมวลผลกับคลาส `BankAccount` โดยแต่ละคลาสมีรายละเอียดตามแผนภาพยูเอ็มแอล (UML Diagram) ที่กำหนดให้ดังต่อไปนี้ (30 คะแนน)

BankAccount	sudseaw:BankAccount
<ul style="list-style-type: none"> - number:int - name:String - balance:double + <u>count = 0:int</u> 	<ul style="list-style-type: none"> name = "Sudseaw" balance = 50000.0
<ul style="list-style-type: none"> + BankAccount(name:String) + BankAccount(name:String, balance:double) + getNumber():int + getName():String + getBalance():double + withdraw(money:double):void + deposit(money:double):void + updateYearInterest(percent:double):void 	<p>โดยกำหนดให้แอดทริบิวต์ <code>number</code> เก็บหมายเลขบัญชี แอดทริบิวต์ <code>name</code> เก็บชื่อบัญชี แอดทริบิวต์ <code>balance</code> เก็บยอดเงินคงเหลือในบัญชี และแอดทริบิวต์ <code>count</code> เก็บจำนวนบัญชีทั้งหมดที่เปิดใช้ในธนาคาร และกำหนดให้นาย Sudseaw เปิดบัญชีเงินฝาก 50000 บาท โดยเมื่อเวลาผ่านไป 1 ปีได้ฝากเงินเพิ่มอีก</p>

20000 บาท และอีก 1 ปีหลังจากนั้นได้ถอนออกไป 10000 บาท จงแสดงจำนวนเงินในบัญชีของ Sudseaw เมื่อเวลาผ่านไป 5 ปี โดยกำหนดให้อัตราดอกเบี้ยคงที่ตลอดเวลาอยู่ที่ร้อยละ 1.75 ต่อปี

โจทย์ข้อที่ 11 [ระดับยาก] จงสร้างคลาส Dice ซึ่งเป็นตัวแทนของลูกเต๋า n หน้าใด ๆ เช่น ลูกเต๋า 6 หน้า ลูกเต๋า 12 หน้า เป็นต้น และเขียนคลาส TestDice เพื่อเรียกใช้เมทอดจากคลาส Dice (40 คะแนน)

คลาส Dice

- 1) ประกาศแอตทริบิว face เก็บจำนวนหน้าของลูกเต๋า (2.5 คะแนน)
- 2) ประกาศแอตทริบิว value เก็บหมายเลขหน้าหงายของลูกเต๋า (2.5 คะแนน)
- 3) สร้างคอนสตรัคเตอร์ 3 แบบได้แก่ Dice(), Dice(int f) และ Dice(int f, int v) (10 คะแนน)
- 4) สร้างเมทอดชื่อ roll เพื่อใช้โยนลูกเต๋า ซึ่งจะสุ่ม (Random) หมายเลขหน้าหงายของลูกเต๋ามีค่าตั้งแต่ 1 จนถึงจำนวนหน้าที่เป็นไปได้ของลูกเต๋านั้น และเมื่อสุ่มหน้าหงายได้แล้ว ก็จะนำไปกำหนดค่าให้กับแอตทริบิว value โดยที่เมทอดนี้ไม่ต้องมีการคืนค่ากลับ (4 คะแนน)
- 5) สร้างเมทอดชื่อ setValue เพื่อใช้ในการพลิกหน้าหงายของลูกเต๋าตามที่ต้องการ ซึ่งจะรับค่าหมายเลขหน้าหงายของลูกเต๋ามาทางพารามิเตอร์ และจะนำไปกำหนดค่าให้กับแอตทริบิว value โดยที่เมทอดนี้ไม่ต้องมีการคืนค่ากลับ (4 คะแนน)
- 6) สร้างเมทอดชื่อ getFace เพื่อคืนค่าจำนวนหน้าของลูกเต๋า (2 คะแนน)
- 7) สร้างเมทอดชื่อ getValue เพื่อคืนค่าหมายเลขหน้าหงายของลูกเต๋า (2 คะแนน)

คลาส TestDice

- 1) สร้างลูกเต๋า (อ็อบเจ็กต์) ชื่อ d1 เป็นลูกเต๋าที่ไม่ได้ระบุจำนวนหน้าและหมายเลขหน้าหงาย (2 คะแนน)
- 2) สร้างลูกเต๋า (อ็อบเจ็กต์) ชื่อ d2 เป็นลูกเต๋า 13 หน้า ที่ไม่ได้ระบุหมายเลขหน้าหงาย (2 คะแนน)
- 3) สร้างลูกเต๋า (อ็อบเจ็กต์) ชื่อ d3 เป็นลูกเต๋า 30 หน้า ที่มีหมายเลขหน้าหงายเป็น 9 (2 คะแนน)
- 4) เขียนเมทอด main(...) เพื่อใช้ในการโยนลูกเต๋าทันที 3 ลูกพร้อม ๆ กัน 2 ครั้ง โดยในแต่ละครั้งให้แสดงผลหมายเลขหน้าหงายของลูกเต๋าทันที 3 ลูกขึ้นบนจอภาพ (7 คะแนน)

โจทย์ข้อที่ 12 [ระดับยาก] จงเขียนคลาส Coin ให้สมบูรณ์ ซึ่งเป็นตัวแทนของเหรียญใด ๆ ที่ประกอบไปด้วยมูลค่าของเหรียญ (Value) และหน้าของเหรียญ (Side) ที่เป็นค่าหัว (H) หรือก้อย (T) พร้อมทั้งสร้างคลาส CoinDemo เพื่อสร้างและเรียกใช้งานเมทอดผ่านทางอ็อบเจ็กต์ของคลาส Coin (30 คะแนน)

คลาส Coin (20 คะแนน)

- 1) แอตทริบิว side เก็บหน้าของเหรียญ เช่น หัว "H" และ ก้อย "T"
- 2) แอตทริบิว value เก็บมูลค่าของเหรียญ เช่น 1 บาท 10 บาท เป็นต้น

- 3) คอนสตรัคเตอร์ 2 แบบได้แก่ `Coin()` และ `Coin(String s, int v)`
- 4) เมท็อดประจำอ็อบเจ็คชื่อ `flip(...)` เพื่อเปลี่ยนหน้าของเหรียญเป็นหัวหรือก้อยโดยการสุ่ม (Random)
- 5) เมท็อดประจำอ็อบเจ็คชื่อ `reverse(...)` เพื่อกลับหน้าของเหรียญจากหัวเป็นก้อยหรือจากก้อยเป็นหัว
- 6) เมท็อดประจำอ็อบเจ็คชื่อ `equals(...)` เพื่อรับหน้าและมูลค่าของเหรียญอีกเหรียญหนึ่งมาทางพารามิเตอร์ แล้วตรวจสอบว่าหน้าและมูลค่าของเหรียญนั้นเท่ากับเหรียญในคลาสนี้หรือไม่
- 7) เมท็อดประจำคลาสชื่อ `getHiLo(...)` เพื่อหน้าของเหรียญ 2 เหรียญใด ๆ เข้ามาทางพารามิเตอร์ โดยถ้าหน้าของเหรียญเป็นหัวหรือก้อยทั้งคู่ให้คืนค่า "Hi" ไม่เช่นนั้นให้คืนค่า "Lo"

คลาส `CoinDemo` (10 คะแนน)

สร้างเหรียญ (อ็อบเจ็ค) `c1` ที่เป็นเหรียญ 1 บาทหน้าหัว เหรียญ `c2` ที่เป็นเหรียญ 5 บาทหน้าหัว และเหรียญ `c3` ที่เป็นเหรียญ 5 บาทหน้าก้อย จากนั้นให้ทำการโยนเหรียญทั้ง 3 เหรียญพร้อม ๆ กันหนึ่งครั้ง โดยใช้การ `Flip` กับเหรียญ `c1` และ `c2` และใช้การ `Reverse` กับเหรียญ `c3` พร้อมทั้งแสดงผลการเท่ากันของเหรียญคู่ `c2` กับ `c3` ขึ้นบนจอภาพ และแสดงค่า `Hi-Lo` ของเหรียญคู่ `c1` กับ `c3` ขึ้นบนจอภาพ

โจทย์ข้อที่ 13 [ระดับยาก] จงสร้างคลาส `PiggyBank` ให้สมบูรณ์เพื่อเป็นตัวแทนของกระปุกออมสินลูกหมูซึ่งเก็บเฉพาะเหรียญ 1 บาท 2 บาท 5 บาท และ 10 บาท และสร้างคลาส `MyPiggyBank` เพื่อใช้สำหรับประมวลผลกับคลาส `PiggyBank` โดยมีรายละเอียดดังต่อไปนี้ (60 คะแนน)

คลาส `PiggyBank`

- 1) แอตทริบิวต์ประจำอ็อบเจ็คแบบ `private` ชื่อ `one`, `two`, `five` และ `ten` ตามลำดับเพื่อเก็บจำนวนของเหรียญแต่ละเหรียญที่เก็บในกระปุกหมู และแอตทริบิวต์ประจำอ็อบเจ็คแบบ `private` ชื่อ `size` เพื่อเป็นขนาดของกระปุกหมู ซึ่งเป็นค่าน้ำหนักรวมสูงสุดของเหรียญที่กระปุกหมูจะเก็บได้ (2 คะแนน)
- 2) ตัวสร้างแบบไม่มีพารามิเตอร์เพื่อกำหนดให้จำนวนเหรียญของทุกเหรียญเป็น 0 และขนาดของกระปุกหมูเป็น 500.0 กรัม (4 คะแนน)
- 3) ตัวสร้างแบบมีพารามิเตอร์หนึ่งตัวประเภทจำนวนจริง เพื่อกำหนดให้จำนวนเหรียญของทุกเหรียญเป็น 0 และขนาดของกระปุกหมูเป็นไปตามค่าที่รับเข้ามาทางพารามิเตอร์ (4 คะแนน)
- 4) เมท็อดประจำอ็อบเจ็คแบบ `public` ชื่อ `clear` ที่ไม่รับพารามิเตอร์ เพื่อกำหนดให้จำนวนเหรียญในกระปุกทุกเหรียญเป็น 0 (4 คะแนน)

- 5) เมท็อดประจำอ็อบเจ็คแบบ `public` ชื่อ `getMoneyTotal` ที่ไม่รับพารามิเตอร์ เพื่อคืนค่าจำนวนเงินทั้งหมดที่เก็บอยู่ในกระปุกหมู (ในกระปุกมีเงินรวมทั้งหมดกี่บาท) (4 คะแนน)
- 6) เมท็อดประจำอ็อบเจ็คแบบ `private` ชื่อ `getMass` ที่ไม่รับพารามิเตอร์ เพื่อคืนค่าน้ำหนักรวม ณ ปัจจุบันของเหรียญที่เก็บอยู่ในกระปุกหมู โดยที่เหรียญ 1 บาท 2 บาท 5 บาท และ 10 บาทหนักเหรียญละ 3.0 กรัม 4.4 กรัม 7.5 กรัม และ 8.5 กรัม ตามลำดับ (4 คะแนน)
- 7) เมท็อดประจำอ็อบเจ็คแบบ `public` ชื่อ `toString` ที่ไม่รับพารามิเตอร์ เพื่อคืนค่าข้อมูลของกระปุกหมูในรูปแบบของสตริง เช่น `One(9)`, `Two(18)`, `Five(3)`, `Ten(51)` เป็นต้น (4 คะแนน)
- 8) เมท็อดประจำอ็อบเจ็คแบบ `public` ชื่อ `isFull` ที่ไม่รับพารามิเตอร์ เพื่อตรวจสอบว่ากระปุกหมูเต็มหรือไม่ (คืนค่า `true` หรือ `false`) โดยให้พิจารณาค่าน้ำหนักของเหรียญทั้งหมดต้องไม่เกินขนาดของกระปุกหมู (4 คะแนน)
- 9) เมท็อดประจำอ็อบเจ็คแบบ `public` ชื่อ `isFull` ที่รับค่าของเหรียญ (1 บาท 2 บาท ฯลฯ) เข้ามาทางพารามิเตอร์ เพื่อตรวจสอบว่าเมื่อหยอดเหรียญค่านั้นลงไป ในกระปุกหมูแล้วจะทำให้กระปุกหมูเต็มหรือไม่ (คืนค่า `true` หรือ `false`) โดยให้พิจารณาเช่นเดียวกับเมท็อด `isFull` ก่อนหน้านี้ (6 คะแนน)
- 10) เมท็อดประจำอ็อบเจ็คแบบ `public` ชื่อ `addCoin` ที่รับค่าของเหรียญผ่านทางพารามิเตอร์ เพื่อใช้ในการหยอดเหรียญลงไป ในกระปุกหมูทีละเหรียญ โดยก่อนหยอดเหรียญจะต้องตรวจสอบก่อนเสมอว่าเหรียญที่หยอดใช้เหรียญที่ถูกต้องหรือไม่ ถ้าไม่ใช่ให้แสดงข้อความว่า `"Invalid Coin"` ขึ้นบนจอภาพ และต้องตรวจสอบด้วยว่ากระปุกหมูเต็มหรือไม่ ถ้าเต็มให้แสดงข้อความว่า `"Cannot Add"` ตามด้วยค่าของเหรียญ ขึ้นบนจอภาพ แต่ถ้าเงื่อนไขทั้งสองถูกต้องให้ทำการหยอดเหรียญลงไป ในกระปุกหมูพร้อมทั้งแสดงข้อความว่า `"Already Add"` ตามด้วยค่าของเหรียญขึ้นบนจอภาพ (8 คะแนน)

คลาส `MyPiggyBank`

กำหนดให้คลาส `MyPiggyBank` ใช้สำหรับประมวลผลคลาส `PiggyBank` โดยสร้างกระปุกหมูตามขนาดที่รับเข้ามาทางแป้นพิมพ์ แล้วทำการรวบรวมเหรียญจากทางแป้นพิมพ์และหยอดลงไป ในกระปุกหมูไปเรื่อย ๆ จนกว่ากระปุกหมูจะเต็ม โดยในการหยอดเหรียญแต่ละครั้งจะต้องแสดงรายละเอียดของกระปุกหมูขึ้นบนจอภาพเสมอ และเมื่อเหรียญในกระปุกเต็มแล้วให้ทำการถามผู้ใช่ว่า `"Renew Piggy Bank? (Y/N)"` ถ้าผู้ใช้ตอบ `"Y"` จะทำการเคลียร์กระปุกหมูใหม่พร้อมทั้งวนรับค่าเหรียญใหม่ แต่ถ้าผู้ใช้ตอบ `"N"` ให้จบการทำงาน (16 คะแนน)