

Programming Fundamentals II

- Lap5:
- Inheritance and Polymorphism
 - Abstract Class and Method
 - Basic Interface

1.1 การใช้คำสั่ง super() และคำสั่ง super.

Lab 5.1 ให้นักศึกษาสร้าง class ดังต่อไปนี้ 1. Employee.java 2. Faculty.java

Employee.java เขียนโค้ดดังนี้

```
public class Employee
{
    public Employee(String s)
    {
        System.out.println(s);
    }
}
```

Faculty.java เขียนโค้ดดังนี้

```
public class Faculty extends Employee
{
    public Faculty()
    {
        System.out.println("Faculty's no-arg constructor is invoked");
    }
}
```

Lab 5.1 ให้นักศึกษาสร้าง class ชื่อ Lab51Super และให้เขียนโค้ดดังนี้

```
public class Lab51Super
{
    public static void main(String[] args)
    {
        new Faculty();
    }
}
```

จงหาสาเหตุของการเกิด error ของไฟล์ Faculty.java เกิดจากสาเหตุอะไร หลังจากนั้นให้ทำการแก้ไขไฟล์ Faculty.java ให้แสดงข้อความ Invoke Employee's constructor เป็นข้อความที่ส่งให้กับ public Employee(String s)

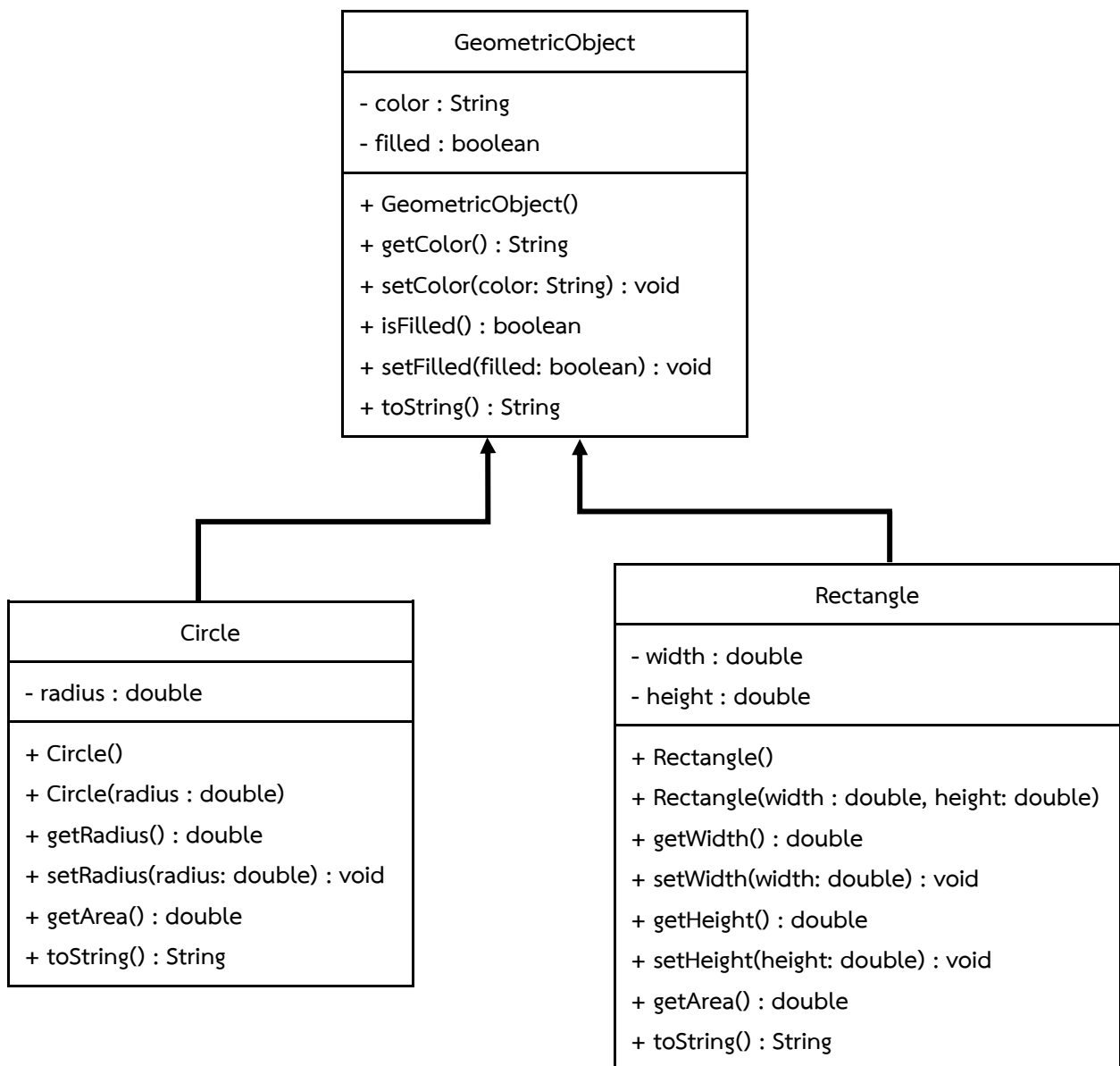
ผลการรัน ที่ต้องการ

```
Invoke Employee's constructor
Faculty's no-arg constructor is invoked
```

เขียนโค้ดที่แก้ไขที่ Faculty.java ให้ได้ผลลัพธ์ตามที่ต้องการ

```
public class Faculty extends Employee
{
    public Faculty()
    {
        _____
        System.out.println("Faculty's no-arg constructor is invoked");
    }
}
```

Lab 5.2 ให้นักศึกษาสร้าง Class ดังนี้ 1. GeometricObject 2. Circle และ 3. Rectangle โดยแต่ละ Class มีส่วนประกอบตาม Class Diagram ข้างล่าง โดยมีรายละเอียดดังต่อไปนี้



GeometricObject มีรายละเอียดดังนี้

- มี Attribute คือ สี (color) , ระบาย (filled)
- Default Attribute color=white, filled=false
- Methods คือ getter และ setter
- toString() มีการทำงานดังนี้

```

public String toString() {
    return "Color: "+color + ",isFilled: "+ filled;
}
  
```

Circle สืบทอดมาจาก GeometricObject มีรายละเอียดดังนี้

- มี Attribute คือ รัศมี (radius)
- Default Attribute radius =0.0
- Methods คือ getter และ setter
- toString() มีการทำงานดังนี้

```
public String toString() {
    return "Color: "+getColor() + ",isFilled: " + isFilled() +
    ",radius:"+radius;
}
```

- getArea() มีการทำงานดังนี้

```
public double getArea(){
    return Math.PI*radius*radius;
}
```

Rectangle สืบทอดมาจาก GeometricObject มีรายละเอียดดังนี้

- มี Attribute คือ ความกว้าง (width), ความสูง (height)
- Default Attribute width =0.0 , height = 0.0
- Methods คือ getter และ setter
- toString() มีการทำงานดังนี้

```
public String toString() {
    return "Color: "+getColor() + ",isFilled: " + isFilled()
    + ",width: " + width + ",height: " + height;
}
```

- getArea() มีการทำงานดังนี้

```
public double getArea(){
    return width*height;
}
```

Lab 5.2 ให้นักศึกษาสร้าง Class ชื่อ Lab52GeometricTest และให้เขียนโปรแกรมให้มีผลลัพธ์ออกมาดังนี้
ตัวอย่างผลลัพธ์

```
Color: white,isFilled: false,radius:5.0
Color: white,isFilled: false,width: 2.0,height: 4.0
```

เขียนโค้ด Lab52GeometricTest.java ให้ได้ผลลัพธ์ตามที่ต้องการ

```
public class Lab52GeometricTest
{
    public static void main(String[] args)
    {
        _____ // สร้าง Object

        _____ // toString Method

        _____ // สร้าง Object

        _____ // toString Method
    }
}
```

ให้นักศึกษาสังเกต Method toString() ของทั้ง 3 Class จะเห็นได้ว่า มี String บางส่วนซ้ำกัน ดังนั้นควรเขียนโค้ดเพิ่มเติมเฉพาะส่วนที่แตกต่างกันเท่านั้น ให้นักศึกษาทำการแก้ไข Method toString() ของ Circle และ Rectangle ต้องใช้คำสั่ง “super.” เข้าช่วยเท่านั้น โดยที่ผลการรัน Class Lab52GeometricTest.java ยังคงเหมือนเดิม

เขียนโค้ดที่แก้ไขที่ Method toString() ของ Class Circle

```
public String toString()
{

}
}
```

เขียนโค้ดที่แก้ไขที่ Method toString() ของ Class Rectangle

```
public String toString()
{

}
}
```

1.2 Polymorphism และ Dynamic binding

เราจะนำ Class จาก Class Diagram ข้างบน 3 Class มาพัฒนาต่อ ให้นักศึกษาสร้าง package แล้ว copy files ทั้ง 3 Class มาใส่ใน package ใหม่ โดยให้ตั้งชื่อ package ว่า “p53”

Lab 5.3 ให้นักศึกษาสร้าง class ว่า Lab53GeometricTest และเขียนโปรแกรมดังนี้

```
package p53;

public class Lab53GeometricTest
{
    public static void main(String[] args)
    {
        double area = 0.0;

        GeometricObject[] objs = {new Circle(5), new Rectangle(2,4)};

        for(GeometricObject obj : objs)
        {
            area += obj.getArea();
        }
        System.out.println("Total area = "+area);
    }
}
```

ทำการคอมไพล์ทั้ง 4 คลาสใหม่ จะพบว่าเกิด compile error จงหาสาเหตุ compile error บันทึกสาเหตุของการเกิด compile error

แล้วทำการแก้ไข compile error เพื่อให้สามารถรันคลาส Lab53GeometricTest ได้เป็นผลลัพธ์ของพื้นที่รวมของทั้ง 2 object

บันทึกการแก้ไขของ compile error

นิสิตแก้ไขที่ **Class** ชื่อ _____

แก้ไข เพิ่ม/ลด ส่วนไหน ให้บันทึกไว้ในกล่องข้อความนี้

ให้นิสิตเพิ่ม Method getDiameter() ลงใน Class Circle ดังนี้

```
public double getDiameter()
{
    return 2*Math.PI*radius;
}
```

Lab 5.4 เราต้องการเล่น Method getDiameter ดังนั้น ให้นิสิตสร้าง class ว่า Lab54GeometricTest และเขียนโปรแกรมดังนี้

```
package p53;

public class Lab54GeometricTest
{
    public static void main(String[] args)
    {
        double area = 0.0;

        GeometircObject[] objs = {new Circle(5), new Rectangle(2,4)};

        for(GeometircObject obj : objs)
        {
            if(_____) // ตรวจสอบ Object circle inx 0 ?
            {
                _____ // Casting

                double diameter = _____

                System.out.println("Diameter of circle = "+diameter);
            }
        }
    }
}
```

ผลลัพธ์ที่ได้จากการรันโปรแกรมข้างบน ให้บันทึกในกล่องข้อความด้านล่าง

1.3 วิธีการ overriding และ final

ให้นักศึกษาสร้าง package โดยให้ตั้งชื่อ package ว่า “p55” และให้นักศึกษาสร้าง class ดังต่อไปนี้ 1. A.java 2. B.java
A.java เขียนโค้ดดังนี้

```
package p55;

public class A
{
    public static void m1()
    {
        System.out.println("m1 in A");
    }

    public void m2()
    {
        System.out.println("m2 in A");
    }

    public final void m3()
    {
        System.out.println("m3 in A");
    }
}
```

B.java เขียนโค้ดดังนี้

```
package p55;

public class B extends A
{
    ----- (A) -----
    {
        ----- (B) -----
    }
}
```

ให้ใส่ code ต่อไปนี้ลง Class B ทีละ Method แล้วบันทึกผลการทดลองที่ได้ หลังจากนั้นให้ comment Method ที่ทดลองก่อนใส่ Method ถัดไป

Method	A	B
m1	public void m1()	System.out.println("m1 in B");
m2	void m2()	System.out.println("m2 in B");
m3	public void m3()	System.out.println("m3 in B");

บันทึกผลการทดลองที่ได้ลงในตารางข้างล่างนี้

กรณีใส่ Method	ผลการ compile และให้ระบุเหตุผลของการ error
m1	
m2	
m3	

1.4 วิธีการ Hiding fields, static method

เราจะนำ Class จาก package p55 ข้างบนมาแก้ไข ดังต่อไปนี้

A.java เขียนโค้ดดังนี้

```
package p55;

public class A
{
    public int i = 1;
    public static int j = 11;

    public static String m1()
    {
        return "A's static m1";
    }

    public String m2()
    {
        return "A's instance m2";
    }

    public String m3()
    {
        return "A's instance m3";
    }
}
```

B.java เขียนโค้ดดังนี้

```
package p55;

public class B extends A
{
    public int i = 2;
    public static int j = 12;

    public static String m1()
    {
        return "B's static m1";
    }

    public String m2()
    {
        return "B's instance m2";
    }
}
```

Lab 5.5 ให้นักศึกษาสร้าง Class ชื่อว่า Lab55HidingFields มีโค้ดดังต่อไปนี้

```
package p55;

public class Lab55HidingFields
{
    public static void main(String[] args)
    {
        A x = new B();

        // Access instance data field i
        System.out.println("(1)    x.i is " + x.i);           // (A)
        System.out.println("(2)  (B)x.i is " + ((B)x).i);      // (B)

        // Access static data field j
        System.out.println("(3)    x.j is " + x.j);           // (C)
        System.out.println("(4)  (B)x.j is " + ((B)x).j);      // (D)

        // Invoke static method m1
        System.out.println("(5)    x.m1() is " + x.m1());      // (E)
        System.out.println("(6)  (B)x.m1() is " + ((B)x).m1()); // (F)

        // Invoke instance method m2
        System.out.println("(7)    x.m2() is " + x.m2());      // (G)
        System.out.println("(8)    x.m3() is " + x.m3());      // (H)
    }
}
```

ก่อนจะทำการรัน นักศึกษาตรวจสอบดูก่อนว่าถ้ารันคลาส Lab55HidingFields แล้วจะได้ผลลัพธ์เป็นอะไร

ทำการรัน และบันทึกผลการทดลอง

บรรทัด	ผลที่คิดว่าจะเกิดขึ้นก่อนทำการ Run	ผลลัพธ์ที่ได้จากการ Run
A		
B		
C		
D		
E		
F		
G		
H		

1.5 Static block , Initialization block

ให้นักศึกษาสร้าง package โดยให้ตั้งชื่อ package ว่า “p56” และให้นักศึกษาสร้าง class ดังต่อไปนี้ 1. N.java 2. M.java
N.java เขียนโค้ดดังนี้

```
package p56;

class N
{
    N()
    {
        System.out.println("N's constructor body");
    }

    {
        System.out.println("N's instance initialization block");
    }

    static
    {
        System.out.println("N's static initialization block");
    }
}
```

M.java เขียนโค้ดดังนี้

```
package p56;

public class M extends N
{
    M()
    {
        System.out.println("M's constructor body");
    }

    {
        System.out.println("M's instance initialization block");
    }

    static
    {
        System.out.println("M's static initialization block1");
    }

    static
    {
        System.out.println("M's static initialization block2");
    }
}
```

Lab 5.6 ให้นักศึกษาสร้าง Class ชื่อว่า Lab56Block มีโค้ดดังต่อไปนี้

```
package p56;

public class Lab56Block
{
    public static void main(String[] args)
    {
        M obj = new M();
    }

    static
    {
        System.out.println("Lab56Block's static block");
    }
}
```

ก่อนจะทำการรัน นิสิตลองตรวจสอบดูก่อนว่าถ้ารันคลาส Lab56Block แล้วจะได้ผลลัพธ์เป็นอะไร
ผลลัพธ์ที่คิดก่อนจะทำการ Run

ผลลัพธ์เมื่อทำการรัน

นิสิตทดลองหาเหตุผล ที่ทำให้เกิดผลลัพธ์เช่นนี้

1.6 Abstract Class

ให้นักศึกษาสร้าง package ตั้งชื่อ package ว่า “p57” และให้นักศึกษาสร้าง class ดังต่อไปนี้

1. Student
2. UnderGraduate
3. Graduate

Student.java เขียนโค้ดดังนี้

```
package p57;

public abstract class Student
{
    private int score;

    public Student(int score)
    {
        this.setScore(score);
    }

    // Getter/Setter Score
    public int getScore()
    {
        return score;
    }
    public void setScore(int score)
    {
        this.score = score;
    }

    // Abstract Method
    public abstract String calculateGrade();
}
```

UnderGraduate.java เขียนโค้ดดังนี้

```
package p57;

public class UnderGraduate extends Student
{
    public UnderGraduate(int score)
    {
        super(score);
    }

    @Override
    public String calculateGrade()
    {
        int score = getScore();
        if(score >= 50)
        {
            return "PASS";
        }
        else
        {
            return "FALL";
        }
    }
}
```

ให้นักนิสิตสร้างคลาส Graduate สืบทอดจากคลาส Student และ ให้ implement method calculateGrade โดยหลักเกณฑ์คะแนนที่ผ่านต้องได้คะแนน score ≥ 70 จึงจะ return ค่าเป็น PASS นอกเหนือจากนั้นให้ return ค่าเป็น FAIL

```
package p57;

public class Graduate extends Student
{

}

}
```

Lab 5.7 ให้นักนิสิตสร้าง Class ชื่อว่า Lab57StudentTest มีโค้ดดังต่อไปนี้

```
package p57;

public class Lab57StudentTest
{
    public static void main(String[] args)
    {
        Student s1 = new Student(35);

        Student[] slist = {new UnderGraduate(50), new Graduate(50) };

        for(Student std : slist)
        {
            System.out.println(std.calculateGrade());
        }
    }
}
```

บันทึกสาเหตุเกิด Compile error

ให้แก้ไขโปรแกรมให้สามารถ Run ได้ หมายเหตุ : ห้ามแก้ไขที่คลาส Student ให้แก้ที่คลาส TestStudent เท่านั้น
หลังแก้ไข compile error แล้วให้ทำการรันคลาส TestStudent

บันทึกผลลัพธ์ที่ได้

1.7 Interface

ให้นักศึกษาคัดสร้าง package ตั้งชื่อ package ว่า “p58” และให้นักศึกษาคัดสร้าง class ดังต่อไปนี้

1. Rectangle, 2. ComparableRectangle, 3. Circle

Rectangle.java เขียนโค้ดดังนี้

```
package p58;
public class Rectangle
{
    private double width;
    private double height;
    public Rectangle(double width, double height){
        this.width = width;
        this.height = height;
    }
    public double getWidth(){
        return width;
    }
    public void setWidth(double width){
        this.width = width;
    }
    public double getHeight(){
        return height;
    }
    public void setHeight(double height){
        this.height = height;
    }
    public double getArea() {
        return width * height;
    }
}
```

ComparableRectangle.java เขียนโค้ดดังนี้

```
package p58;
public class ComparableRectangle extends Rectangle implements Comparable
{
    public ComparableRectangle(double width, double height)
    {
        super(width, height);
    }

    public int compareTo(Object o)
    {
        if (getArea() > ((ComparableRectangle)o).getArea())
            return 1;
        else if (getArea() < ((ComparableRectangle)o).getArea())
            return -1;
        else
            return 0;
    }
}
```

Circle.java เขียนโค้ดดังนี้

```
package p58;
public class Circle
{
    private double radius;
    public Circle(double radius) {
        this.radius = radius;
    }
    public double getRadius() {
        return radius;
    }
    public void setRadius(double radius) {
        this.radius = radius;
    }
    public double getArea() {
        return radius * radius * Math.PI;
    }
}
```

Class ComparableCircle เป็น Class ที่ implement interface Comparable โดยมี Method compareTo ไว้เปรียบเทียบ object ของคลาส ComparableCircle ดังนั้น นิติตจพัฒนา Class ComparableCircle ต่อไปนี้ให้สมบูรณ์เพื่อนำไปใช้งานโดย Class Max และ Class Lab58MaxTest ข้างล่าง

เขียนโค้ด ComparableCircle.java ที่นิติตพัฒนาลงกล่องข้อความด้านล่าง

```
package p58;
public class ComparableCircle extends Circle implements Comparable
{

}

}
```

Class Max.java มี Method max เพื่อไว้เปรียบเทียบระหว่าง Obj1 และ Obj2 ดัง source codes ต่อไปนี้

```
package p58;
public class Max
{
    /** Return the maximum of two objects */
    public static Comparable max (Comparable o1, Comparable o2) {
        if (o1.compareTo(o2) > 0)
            return o1;
        else
            return o2;
    }
}
```

Lab 5.8 ให้นิติตสร้าง Class ชื่อว่า Lab58MaxTest มีการสร้าง object จาก Class ComparableRectangle แล้วส่งต่อให้กับ method max ของ Class Max และนิติตจเพิ่ม source codes ส่วนเรียกใช้งาน Class ComparableCircle ที่พัฒนาขึ้น ลงใน Class Lab58MaxTest ต่อไปนี้

```
package p58;
public class Lab58MaxTest
{
    public static void main(String[] args)
    {
        ComparableRectangle rectangle1 = new ComparableRectangle(4, 5);    //20
        ComparableRectangle rectangle2 = new ComparableRectangle(3, 6);    //18
        Rectangle r = (Rectangle)Max.max(rectangle1, rectangle2);
        System.out.println(r.getArea());

        /* เพิ่ม Code เพื่อทดสอบ Class ComparableCircle ด้วยตนเอง */

        _____
        _____
        _____
        _____

    }
}
```

หลังจากพัฒนา Class Lab58MaxTest แล้วจงทำการรันและศึกษาการทำงานต่างๆ ของ Class ก่อนหน้าทั้งหมด

แบบทดสอบ Lab5

ข้อ1 ในงานชิ้นนี้เราจะสร้าง Invoice ขึ้น โดยให้ตัว Invoice มีรายการสินค้าได้หลายรายการ รายการสินค้าจะแยกเป็นคลาส Lineltem และ Product ไปต่างหาก นอกจากนี้ในแต่ละ Invoice ยังมีชื่อของลูกค้ากำกับด้วย ดังนั้นในงานนี้เราจะสร้างคลาสทั้งหมด 5 คลาสดังนี้

1. Invoice (ใบแจ้งรายการสินค้า)

ประกอบด้วยคุณสมบัติดังต่อไปนี้

1. id เป็น String แทนรหัสของใบแจ้งรายการสินค้า
 2. customer เป็น Customer แทนลูกค้าที่เป็นผู้ซื้อของ Invoice นี้
 3. items เป็นอาร์เรย์ของ Lineltem แทนรายการซื้อแต่ละรายการ และมีเมทอดดังต่อไปนี้
 4. Invoice(String id, Customer customer)
 5. void addItem(Product product, int quantity)
สร้าง Lineltem ใหม่จาก product และ quantity และใส่เข้าไปในรายการซื้อ
 6. String getId()
 7. Customer getCustomer()
 8. Lineltem getLineltem(int i) ส่งค่ากลับเป็นรายการซื้อลำดับที่ i โดยนับรายการแรกเป็นรายการที่ 0
 9. double getTotalPrice() ส่งค่ากลับเป็นราคารวมของทุกรายการซื้อ
 10. void print() แสดงข้อมูลของ Invoice ในรูปแบบดังตัวอย่างด้านล่าง
- *** กำหนดให้ใช้ ArrayList ในการเก็บ Lineltem แต่ละรายการ

2. Customer (ลูกค้า)

ประกอบด้วยคุณสมบัติดังต่อไปนี้

1. id เป็น String แทนรหัสลูกค้า
 2. firstName เป็น String แทนชื่อ
 3. lastName เป็น String แทนนามสกุล
- และมีเมทอดดังต่อไปนี้
4. Customer(String id, String firstName, String lastName)
 5. String getId()
 6. String getFirstName()
 7. String getLastName()

3. Lineltem (รายการซื้อ)

ประกอบด้วยคุณสมบัติดังต่อไปนี้

1. item เป็น Product แทนสินค้ารายการนั้น
 2. quantity เป็น int แทนจำนวนสินค้าในรายการ
- และมีเมทอดดังต่อไปนี้
3. Lineltem(Product product, int quantity) ถ้า quantity เป็นลบ ให้กำหนดให้เป็น 0
 4. Product getProduct()
 5. int getQuantity()
 6. double getTotalPrice() ส่งค่ากลับเป็นราคารวมของสินค้ารายการนี้

4. Product (สินค้า)

ประกอบด้วยคุณสมบัติดังต่อไปนี้

1. id เป็น String แทนรหัสสินค้า
2. name เป็น String แทนชื่อสินค้า
3. price เป็น double แทนราคาสินค้า

และมีเมทอดดังต่อไปนี้

4. Product(String id, String name, double price) ถ้า price เป็นลบ ให้กำหนดให้เป็น 0.0
5. String getId()
6. String getName()
7. void setPrice(double price) ถ้า price เป็นลบ ให้กำหนดให้เป็น 0.0
8. double getPrice()

5. InvoiceTest (ตัวทดสอบโปรแกรม)

ตัวทดสอบจะต้องสร้าง Invoice ขึ้นมาอย่างน้อย 3 Invoice ลูกค้า (Customer) อย่างน้อย 2 คน สินค้า (Product) อย่างน้อย 5 ชนิด Invoice มากกว่า 1 อันอาจเป็นของลูกค้าคนเดียวกันได้ และแต่ละ Invoice จะต้องมีการซื้ออย่างน้อย 3 รายการ และแสดงผลลัพธ์โดยการเรียกเมทอด print ของ Invoice แต่ละอันออกมา รูปแบบการแสดงผลของ Invoice แต่ละอันจะอยู่ในแบบดังตัวอย่างนี้ (อย่าลืมว่ามี Invoice 3 อัน เพราะฉะนั้นจะต้องมีการแสดงผลในลักษณะตามตัวอย่างทั้งหมด 3 ครั้งที่แตกต่างกัน)

ตัวอย่างผลลัพธ์ เมื่อทำการรันโปรแกรม

```
INVOICE: #<invoice id>
CUSTOMER: <customer name>

ITEMS:
1. <item name> x <item quantity> = <price>
2. ...
...

TOTAL: <total price>
```

หมายเหตุ1 คุณสมบัติทั้งหมดต้องเป็น private และให้เมทอดทั้งหมดเป็น public

หมายเหตุ2 สำหรับแบบฝึกหัด ให้ทำการ comment ในโปรแกรม ในส่วนของ statement หลักๆของโค้ดที่นิสิตเขียน ถ้าไม่มีการ comment จะถือว่าโค้ดไม่ครบสมบูรณ์