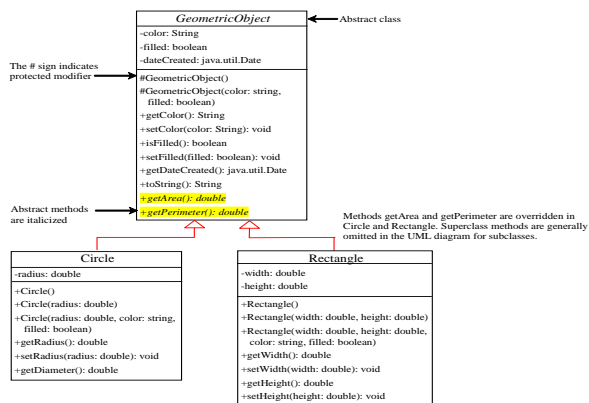


แบบฝึกหัดปฏิบัติการคาบที่ 8: Interface / Abstract

คำสั่ง

1. ให้ศึกษาหลักการ Interface / Abstract ต่อไปนี้

เราสามารถที่จะนิยามเมธอดต่างๆ ภายในคลาสขึ้นมาก่อนได้โดยยังไม่ต้องกำหนดคำสั่งต่างๆ ภายในเมธอดเหล่านั้น แต่จะให้มีการกำหนดคำสั่งภายในคลาสอื่นๆ ที่จะสืบทอด วิธีการนี้จะช่วยให้เราสามารถจะกำหนดเมธอดที่ไม่สามารถนิยามคำสั่งไว้ล่วงหน้าได้ ภาษาจาวาได้กำหนดคลาสลักษณะนี้ไว้สองแบบคือ คลาสแบบ Abstract และ อินเตอร์เฟส



คลาสแบบ Abstract

- คลาสที่มีคำว่า abstract นำหน้าจะหมายความว่าคลาสนั้นยังเป็นคลาสที่ไม่สมบูรณ์ โดยมีเมธอดแบบ abstract ซึ่งเป็นเมธอดที่ยังไม่สมบูรณ์อย่างน้อยหนึ่งเมธอดอยู่ในคลาส
- abstract method จะต้องอยู่ในคลาสที่เป็น abstract class
- ถ้าคลาสลูกของ คลาสแม่ที่เป็น abstract ไม่เขียนรายละเอียดของ abstract methods ทั้งหมด ตัวของคลาสลูกเองต้องกำหนดเป็น abstract
- คลาสลูกที่ไม่ได้เป็น abstract ที่สืบทอดมาจากคลาสแม่ที่เป็น abstract ทุก ๆ abstract methods จะต้องถูกเขียนรายละเอียดการทำงานแม้ว่าจะไม่ได้ใช้ใน subclass ก็ตาม
- ไม่สามารถสร้างวัตถุจาก abstract class ด้วยโอเปอเรเตอร์ new แต่สามารถใช้ abstract class เพื่อเป็น data type เช่น `GeometricObject[] geo = new GeometricObject[10];`
- สามารถที่จะสร้าง constructors ได้โดยสามารถถูกเรียกได้จาก constructors ของคลาสลูก เช่น constructors ของ `GeometricObject` สามารถถูกเรียกจากคลาส `Circle` และคลาส `Rectangle` ได้
- สามารถสร้าง abstract class ที่ไม่มี abstract methods เลยก็ได้ในกรณีนี้เราไม่สามารถสร้างวัตถุจากคลาสดังกล่าวโดยการใช้โอเปอเรเตอร์ new โดยคลาสดังกล่าวจะถูกใช้เป็น base class เพื่อสร้าง subclass ตัวใหม่
- คลาสลูกสามารถเป็น abstract ได้แม้ว่า superclass จะเป็นคลาสปกติ (concrete class) เช่น `abstract class GeometricObject extends Object`
- คลาสลูกสามารถโอเวอร์ไรด์เมธอดจากคลาสแม่เพื่อให้เป็น abstract method ก็ได้

อินเตอร์เฟส อินเตอร์เฟส คือโครงสร้างที่มีลักษณะคล้ายคลาส ส่วนประกอบของอินเตอร์เฟสจะมีเฉพาะ ค่าคงที่ constants และ abstract methods

- จุดประสงค์ของการสร้าง interface จะคล้าย abstract class แต่จุดประสงค์หลักจริง ๆ คือการกำหนดพฤติกรรม behavior สำหรับ objects เช่น `Comparable`, `Edible`, `Cloneable`
- interface เป็นคลาสพิเศษในภาษาจาวาโดย ทุกๆ interface จะถูกคอมไพล์เป็น bytecode เหมือนคลาสปกติ
- ไม่สามารถสร้างวัตถุจาก interface โดยการใช้ new แต่สามารถใช้สำหรับการประกาศชนิดข้อมูลให้กับตัวแปร

2. ให้ศึกษาและทดลองพิมพ์ Code ตัวอย่างการสร้างอินเตอร์เฟซจากตัวอย่างต่อไปนี้

```

1 public class House implements Cloneable, Comparable<House> {
2     private int id;
3     private double area;
4     private java.util.Date whenBuilt;
5     public House(int id, double area) {
6         this.id = id;
7         this.area = area;
8         whenBuilt = new java.util.Date();
9     }
10    public int getId() {
11        return id;
12    }
13    public double getArea() {
14        return area;
15    }
16    public java.util.Date getWhenBuilt() {
17        return whenBuilt;
18    }
19    @Override
21    public Object clone() {
22        try {
23            return super.clone();
24        }
25        catch (CloneNotSupportedException ex) {
26            return null;
27        }
28    }
29    @Override // Implement the compareTo method defined in Comparable
30    public int compareTo(House o) {
31        if (area > o.area)
32            return 1;
33        else if (area < o.area)
34            return -1;
35        else
36            return 0;
37    }
38 }

```

2.1 ให้อธิบายการทำงานของโปรแกรมต่อไปนี้

บรรทัดที่ 1

บรรทัดที่ 2- 4

บรรทัดที่ 5- 9

บรรทัดที่ 10-18

บรรทัดที่ 21-28

บรรทัดที่ 30-37

2.2 สร้างคลาสทดสอบ และเพิ่มคำสั่งต่อไปนี้ใน main() ผลลัพธ์จะมีผลลัพธ์อย่างไร

```

House house1 = new House(1, 1750.50);
House house2 = (House)house1.clone();

```

3 จงเขียน class ที่วาดกราฟจากฟังก์ชันต่อไปนี้

```

abstract class AbstractDrawFunction extends JPanel
{
    /**Polygon to hold the points*/
    private Polygon p = new Polygon();
    /**Default constructor*/
    protected AbstractDrawFunction ()
    {
        drawFunction();
        setBackground(Color.white);
    }
    /**Draw the function*/
    public abstract double f(double x);
    /**Obtain points for x coordinates 100, 101, ..., 300*/
    public void drawFunction()
    {
        for (int x = -100; x <= 100; x++)
        {
            p.addPoint(x+200, 200-(int)f(x));
        }
    }
    /**Paint the function diagram*/
    public void paintComponent(Graphics g)
    {
        super.paintComponent(g);
        // Draw x axis
        g.drawLine(10, 200, 390, 200);
        // Draw y axis
        g.drawLine(200,30, 200, 390);
        // Draw arrows on x axis
        g.drawLine(390, 200, 370, 190);
        g.drawLine(390, 200, 370, 210);
        // Draw arrows on y axis
        g.drawLine(200, 30, 190, 50);
        g.drawLine(200, 30, 210, 50);
        // Draw x, y
        g.drawString("X", 370, 170);
        g.drawString("Y", 220, 40);
        // Draw a polygon line by connecting the points in the polygon
        g.drawPolyline(p.xpoints, p.ypoints, p.npoints);
    }
}

```

ทดสอบคลาสโดยใช้ฟังก์ชันต่อไปนี้

- $f(x) = x^2$;
- $f(x) = \sin(x)$;
- $f(x) = \cos(x)$;
- $f(x) = \tan(x)$;
- $f(x) = \cos(x) + 5\sin(x)$;
- $f(x) = 5\cos(x) + \sin(x)$;
- $f(x) = \log(x) + x^2$;

สำหรับฟังก์ชันแต่ละฟังก์ชันให้สร้างคลาสที่ extends คลาส AbstractDrawFunction และ implements เมธอด `f(double x)`;

กำหนดให้คลาสทดสอบคือ

```
public class Test extends JFrame
{
    public Test()
    {
        getContentPane().setLayout(new GridLayout(1, 2, 5, 5));
        getContentPane().add(new DrawSine());
    }
    public static void main(String[] args)
    {
        Test frame = new Test();
        frame.setSize(400, 400);
        frame.setTitle("Exercise 10.10");
        frame.setVisible(true);
    }
}
```

4. [Application] จงใช้ Employee ที่ให้มาต่อไปนี้เป็นคลาสแม่

```
public abstract class Employee{
    private String firstname;
    private String lastname;
    private String id;
    public Employee(String firstname,String lastname,String id){
        this.firstname=firstname;
        this.lastname=lastname;
        this.id=id;
    }
    public abstract double earning();
    public abstract double bonus(int year);
}
```

4.1 จงสร้างคลาสลูกของ Employee ชื่อ SalariedEmployee ซึ่งมีรายละเอียดดังนี้

1. มี Instance variable เพิ่มจาก Employee ชื่อ salary สำหรับบันทึกค่าเงินเดือนของพนักงาน
2. มี Constructor ที่รับข้อมูลดังนี้

```
public SalariedEmployee(String firstname,String lastname,String id, double sal)
```

เพื่อส่งพารามิเตอร์สามตัวแรกไปยัง constructor ใน employee ให้ทำการกำหนดค่าเบื้องต้นให้กับ

firstname lastname id ส่วนพารามิเตอร์ตัวสุดท้ายใช้สำหรับกำหนดค่าให้กับ salary

3. ให้ method bonus (int year) มีการประมวลผลโดยรับพารามิเตอร์ year ซึ่งเป็นระยะเวลาการทำงานของพนักงานเข้ามา เพื่อทำการตรวจสอบว่าถ้าระยะเวลาเกิน 5 ปี จะคืนค่าโบนัสเป็น 12 เท่าของเงินเดือน ไม่เช่นนั้นจะคืนค่าโบนัสเป็น 6 เท่าของเงินเดือน

4. ให้ Method earning () มีการประมวลผลโดยนำเงินเดือนของพนักงานแต่ละคนมาทำการหักภาษี 5% ของเงินเดือน แล้ว return เงินเดือนที่หักภาษีแล้ว

4.2 จงสร้างคลาสลูกของ Employee ชื่อ ComEmployee ซึ่งมีรายละเอียดดังนี้

1. มี Instance variable เพิ่มจาก Employee ชื่อ grossSale สำหรับบันทึกยอดขายที่ได้รับ และ ComRate ที่มีชนิดข้อมูลเป็นเลขจำนวนจริง เพื่อบันทึกเปอร์เซ็นต์ของคอมมิชชั่นที่ได้
2. มี Constructor ที่รับข้อมูลดังนี้

```
public ComEmployee(String firstname,String lastname,String id, double sales, double percent)
```

เพื่อส่งพารามิเตอร์สามตัวแรกไปยัง constructor ใน employee ให้ทำการกำหนดค่าเบื้องต้นให้กับ firstname lastname id ส่วนพารามิเตอร์ sale และ percent ใช้สำหรับกำหนดค่าให้กับ grossSale และ ComRate

3. ให้ Method bonus (int year) มีการประมวลผลโดยรับพารามิเตอร์ year ซึ่งเป็นระยะเวลาการทำงานของพนักงานเข้ามา เพื่อทำการตรวจสอบว่าถ้าระยะเวลาเกิน 5 ปี จะคืนค่าโบนัสเป็น 6 เท่าของยอดขาย ไม่เช่นนั้นจะคืนค่าโบนัสเป็น 3 เท่าของยอดขาย

4. ให้ Method earning () มีการประมวลผลโดยนำยอดขาย ของพนักงานแต่ละคนมารวมกับค่าคอมมิชชั่น (Comission=grossSale*ComRate) แล้ว return ผลรวมที่ได้

4.3. จงสร้าง test class ชื่อ Final.java ซึ่งมีรายละเอียดต่อไปนี้

1.ให้สร้าง arrayList ขึ้นมา 1 arrayList เพื่อเก็บ ข้อมูลของพนักงานทุกประเภทไว้ด้วยกันให้เป็นพนักงานชนิด SalariedEmployee 2 คน และพนักงานชนิด ComEmployee 2 คน

2. ให้สร้าง method ชื่อ printEmp(ArrayList a) โดยรับพารามิเตอร์ที่เป็นข้อมูลชนิด ArrayList แล้วทำการคำนวณหา earning และ bonus ของแต่ละคนโดยใช้คำสั่งเดียวกันคือ r.earning() และ r.bonus() จากนั้นทำการสร้าง array ชื่อ arrayEarn สำหรับเก็บข้อมูล earning ของพนักงานทุกคน และสร้าง array ชื่อ arrayBonus สำหรับเก็บข้อมูลโบนัสของพนักงานทุกคน ท้ายสุดให้ method printEmp(ArrayList a) นี้ทำการพิมพ์รายงานที่มีรูปแบบดังด้านล่างนี้ออกทาง dialog box

Fisrt name Last name Earning Bonus

5. กำหนดให้ **GeometricObject.java** มีรายละเอียดดังนี้

```
public abstract class GeometricObject {
    private String color = "white";
    private boolean filled;
    protected GeometricObject() {
    }
    protected GeometricObject(String color, boolean filled) {
        this.color = color;
        this.filled = filled;
    }
    public String getColor() {
        return color;
    }
    public void setColor(String color) {
        this.color = color;
    }
    public boolean isFilled() {
        return filled;
    }
    public void setFilled(boolean filled) {
        this.filled = filled;
    }
    public abstract double getArea();
    public abstract double getPerimeter();
}
```

5.1 ให้สร้างคลาส Circle ที่ extend GeometricObject และ implements อินเตอร์เฟซ Comparable ให้ทำการ override เมธอด equals() ของคลาส Object และ implements เมธอด compareTo() เพื่อทำการเปรียบเทียบวงกลมจากรัศมีของวงกลม โดยที่วงกลมสองวงจะเท่ากันก็ต่อเมื่อมีรัศมีเท่ากัน วาดคลาสไดอะแกรมและเขียนโปรแกรมทดสอบคลาสดังกล่าว

5.2 ให้สร้างคลาส Rectangle ที่ extends GeometricObject และ implements อินเตอร์เฟซ Comparable ให้ทำการ override เมธอด equals() ของคลาส Object และ implements เมธอด compareTo() เพื่อทำการเปรียบเทียบ Rectangle จากพื้นที่ของ Rectangle โดยที่สี่เหลี่ยมจะเท่ากันก็ต่อเมื่อมีพื้นที่เท่ากัน วาดคลาสไดอะแกรมและเขียนโปรแกรมทดสอบคลาสดังกล่าว

5.3 ให้สร้างคลาส Octagon ที่ extends GeometricObject และ implements อินเตอร์เฟส Comparable และ Cloneable ให้ทำการ override เมธอด equals() ของคลาส Object เมธอด compareTo() เพื่อทำการเปรียบเทียบ Octagon จากพื้นที่ของ Octagon โดยที่รูปแปดเหลี่ยมจะเท่ากันก็ต่อเมื่อมีพื้นที่เท่ากัน โดยพื้นที่ของ Octagon คือ

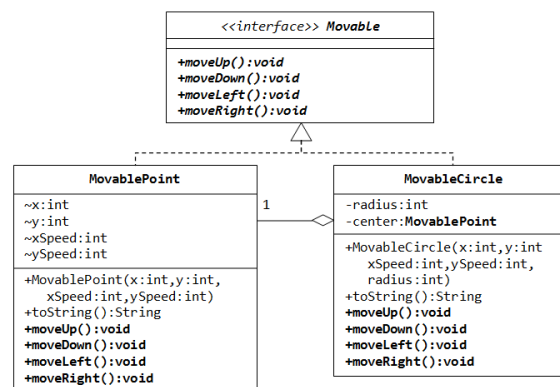
$$\text{area} = (2 + 4/\sqrt{2}) * \text{side} * \text{side}$$

วาดคลาสไดอะแกรมและเขียนโปรแกรมทดสอบคลาสดังกล่าวโดยสร้าง Object จากคลาส Octagon ที่มี side=5 ให้แสดงพื้นที่และเส้นรอบวง หลังจากนั้นให้สร้าง object ใหม่โดยการใช้เมธอด clone และเปรียบเทียบว่า object ทั้งสองดังกล่าวเท่ากันหรือไม่

5.4 สร้างคลาสที่ชื่อว่า ComparableCircle ที่สืบทอดมาจากคลาส Circle และ implements อินเตอร์เฟส Comparable วาด UML class diagram และ implements เมธอด compareTo() เพื่อทำการเปรียบเทียบวงกลมจากพื้นที่ของวงกลม เขียน Test class ที่ค้นหาวงกลมที่ใหญ่ที่สุดจาก 2 วงกลมที่เป็น object ของ ComparableCircle

6. Suppose that we have a set of objects with some common behaviors: they could move up, down, left or right. The exact behaviors (such as how to move and how far to move) depend on the objects themselves. One common way to model these common behaviors is to define an *interface* called Movable, with abstract methods moveUp(), moveDown(), moveLeft() and moveRight(). The classes that implement the Movable interface will provide actual implementation to these abstract methods.

Let's write two concrete classes - MovablePoint and MovableCircle - that implement the Movable interface.



7. [Algorithms] รถไฟใต้ดิน (subway)

พ.ศ. 2560 รัฐบาลได้ดำเนินโครงการก่อสร้างรถไฟใต้ดินซึ่งเป็นโครงการเมกะโปรเจกต์จนเสร็จสิ้น ทำให้กรุงเทพฯ กลายเป็นเมืองที่มีเครือข่ายรถไฟใต้ดินที่ใหญ่ที่สุดแห่งหนึ่งของโลก ประกอบด้วยเส้นทางรถไฟใต้ดินหลายร้อยสาย และสถานีอีกนับล้านสถานี คุณต้องการเดินทางโดยรถไฟใต้ดินจากสถานีหนึ่งไปยังอีกสถานีหนึ่ง โดยในระหว่างทาง สามารถทำการเปลี่ยนสายรถไฟได้โดยการไปลงที่บางสถานีแล้วขึ้นรถไฟใต้ดินสาย อื่นที่ผ่านสถานีนั่นต่อ แต่การเปลี่ยนสายรถไฟแต่ละครั้งก็ทำให้เสียเวลาเป็นอย่างมาก คุณจึงต้องการเดินทางโดยเปลี่ยนสายรถไฟให้น้อยครั้งที่ที่สุดเท่าที่จะทำได้

จงเขียนโปรแกรมเพื่อตอบคำถามทั้งหมด Q คำถามว่า การเดินทางจากสถานี A_i ไปยังสถานี B_i จะต้องทำการเปลี่ยนสายรถไฟอย่างน้อยกี่ครั้ง

ข้อมูลนำเข้า

บรรทัดแรกระบุจำนวนเต็ม N และ M ($2 \leq N \leq 1,000,000$; $1 \leq M \leq 500$) แทนจำนวนสถานีทั้งหมดและจำนวนสายของรถไฟใต้ดิน

อีก M บรรทัดต่อมา ในบรรทัดที่ $i+1$ ($1 \leq i \leq M$) ระบุจำนวนเต็มตัวแรกคือ S_i ($2 \leq S_i \leq 2,000$) แทนจำนวนสถานีที่รถไฟ

ใต้ดินสายที่ i ผ่าน และจำนวนเต็มอีก S_i จำนวนถัดมา ระบุหมายเลขของสถานีที่รถไฟใต้ดินสายดังกล่าวผ่าน เรียงตามลำดับจากปลายทางข้างหนึ่งไปจนถึงปลายทางอีกข้างหนึ่ง

บรรทัดถัดมา ระบุจำนวนเต็ม Q ($2 \leq Q \leq 1,000,000$) แทนจำนวนคำถามทั้งหมด

อีก Q บรรทัดถัดมา ในบรรทัดที่ $i+M+2$ ($1 \leq i \leq Q$) ระบุจำนวนเต็ม A_i และ B_i ($1 \leq A_i, B_i \leq N$) แสดงถึงคำถามที่ i สถานีแต่ละสถานีจะมีรถไฟใต้ดินผ่านไม่เกิน 20 สาย โดยที่บางสถานีอาจไม่มีรถไฟใต้ดินผ่านเลยแม้แต่สายเดียวก็ได้ นอกจากนี้เส้นทางของรถไฟใต้ดินแต่ละสายอาจผ่านบางสถานีมากกว่าหนึ่งครั้งก็ได้

ข้อมูลส่งออก

มีทั้งหมด Q บรรทัด ในบรรทัดที่ i ($1 \leq i \leq Q$) ให้พิมพ์จำนวนครั้งของการเปลี่ยนสายรถไฟที่น้อยที่สุดที่ต้องใช้ในการเดินทางจากสถานี A_i ไปยังสถานี B_i แต่ถ้าไม่สามารถเดินทางโดยรถไฟใต้ดินจากสถานี A_i ไปยังสถานี B_i ได้ ให้พิมพ์คำว่า impossible

ตัวอย่างข้อมูลนำเข้า	ตัวอย่างข้อมูลส่งออก
6 2 3 1 2 3 3 2 4 5 3 1 3 1 4 2 6	0 1 impossible
15 5 6 1 2 3 4 2 5 2 6 7 4 1 6 8 9 4 10 11 12 13 3 14 11 15 6 9 2 10 13 10 5 3 7 6 14 15 12	1 0 impossible 2 impossible 1