

แบบฝึกปฏิบัติการครั้งที่ 11 : Abstract class and polymorphism1. จงสร้าง **abstract class** ชื่อ Taylor โดย

มี private instance variable คือ int k และ double x (ตามสูตรด้านล่าง)

มี constructor เพื่อกำหนดค่า k และ x ตามที่รับมาทางพารามิเตอร์

มี method

public int factorial(int n) คำนวณค่า factorial ของ n

public int getIteration() เพื่อคืนค่า k

public double getValue() เพื่อคืนค่า x

public abstract void printValue()

public abstract double getApprox()

และจงสร้างคลาส 3 คลาส ชื่อ Expo, Sine และ Cosine โดยทั้งสามคลาส extends จาก Taylor เพื่อคำนวณค่าประมาณของ  $e^x$ ,  $\sin(x)$  และ  $\cos(x)$  ตามลำดับจากสูตรต่อไปนี้

$$\exp(x) = \sum_{n=0}^k \frac{x^n}{n!}$$

$$\sin(x) = \sum_{n=0}^k \frac{(-1)^n x^{2n+1}}{(2n+1)!}$$

$$\cos(x) = \sum_{n=0}^k \frac{(-1)^n x^{2n}}{(2n)!}$$

โดยให้ทั้งสามคลาส Override method จาก Taylor ดังนี้

public abstract double getApprox() เพื่อคืนค่าประมาณที่คำนวณได้จากสูตรที่กำหนด

public abstract void printValue() เพื่อแสดงผลลัพธ์เป็นค่าที่คำนวณได้จากการเรียกใช้

method ของคลาส Math และที่คำนวณได้จากการเรียกใช้ method getApprox()

กำหนด main method ของคลาสทดสอบดังนี้

```
public static void main(String[] args) {  
    Expo exp = new Expo(7, 1);  
    exp.printValue();  
    Sine s = new Sine(7, Math.PI/4);  
    s.printValue();  
    Cosine cs = new Cosine(7, 1);  
    cs.printValue();  
}
```

ผลลัพธ์ที่ได้คือ

```
Value from Math.exp() is 2.718281828459045.  
Approximated value is 2.7182539682539684.  
Value from Math.sine() is 0.7071067811865475.  
Approximated value is 0.7071067811886981.  
Value from Math.cos() is 0.5403023058681398.  
Approximated value is 0.5403023050976684.
```

จงเพิ่มคำสั่งใน main method ต่อท้ายจาก code เดิมที่มีอยู่ โดยให้มีการสร้างตัวแปรแถวลำดับขนาด 3 ช่องให้มีชนิดเป็น Taylor แล้วสร้าง object จากคลาสทั้งสามเก็บเข้าไปในแถวลำดับ จากนั้น วนลูปเพื่อเรียกใช้ method printValue() ให้แสดงผลลัพธ์เป็นค่าที่ประมาณได้จากทั้ง 3 object

2. กำหนดคลาส Quadrilateral (สี่เหลี่ยมใดๆ) เป็น abstract superclass มี instance variable 4 ตัวเพื่อกำหนดจุดที่เป็นมุมทั้ง 4 ของสี่เหลี่ยม โดยแต่ละจุดสร้างขึ้นจาก Point class (ศึกษาคลาสนี้ได้จาก Java API) constructor ของคลาส Quadrilateral จะรับค่าคู่อันดับ x, y ทั้งหมด 4 จุด (รวม 8 ค่า) เพื่อกำหนดเป็นมุมของสี่เหลี่ยมทั้ง 4 มุม (โดยรับค่า x1, y1, x2, ... , y4) ตามตัวอย่าง

```
import java.awt.Point;
public abstract class Quadrilateral    {
    private Point a, b, c, d;

    public Quadrilateral(int x1, int y1, int x2, int y2,
int x3, int y3, int x4, int y4)    {
        a = new Point(x1, y1);
        b = new Point(x2, y2);
        c = new Point(x3, y3);
        d = new Point(x4, y4);
    }
}
```

/\* จงเขียน method getSortedDist () เพื่อคำนวณหาระยะห่างระหว่างคู่จุดต่าง ๆ ที่เป็นมุมของสี่เหลี่ยม เก็บลงในตัวแปรแถวลำดับ dis ขนาด 6 ช่อง แล้วเรียงค่าระยะห่างใน dis ให้มีลำดับจากน้อยไปมาก คืนค่าเป็นแถวลำดับชนิด double

**Hint :** การหาระยะทางระหว่าง 2 จุด สามารถเรียกใช้ method distance() ที่คลาส Point inherit จากคลาส java.awt.geom.Point2D

แต่การเรียงลำดับ ขอให้เขียน code เพื่อเรียงลำดับค่าในแถวลำดับเอง ไม่ใช่ sort() \*/

/\* จง override toString() เพื่อให้แสดงผลลัพธ์เป็นจุด (x,y) ของจุดมุมทั้ง 4 จุด \*/

```
// จงเขียน abstract method area ซึ่งคืนค่าชนิด double
    }
}
```

- จงสร้างคลาส Kite, Rectangle และ Square ซึ่งเป็นคลาสสี่เหลี่ยมรูปว่าว, สี่เหลี่ยมผืนผ้า และ สี่เหลี่ยมจัตุรัส ตามลำดับ ให้เป็น subclass ของ Quadrilateral โดยให้ในคลาสทั้งสาม
  - มี constructor รับค่าคู่อันดับ x, y ทั้งหมด 4 จุด (รวม 8 ค่า)
  - override abstract method **area()** เพื่อคำนวณหาพื้นที่ของแต่ละ object ที่สร้างขึ้นจากทั้ง 3 คลาส โดยใช้สูตรดังนี้
    - พท สี่เหลี่ยมจัตุรัส = ด้าน x ด้าน
    - พท สี่เหลี่ยมผืนผ้า = กว้าง x ยาว
    - พท สี่เหลี่ยมรูปว่าว = ผลคูณเส้นทแยงมุม / 2
- จากนั้นสร้างคลาสทดสอบดังนี้

```
public class QuadrilateralTester {
    public static void main(String[] args) {
        Square a1 = new Square(0,0,2,2,0,2,2,0);
        System.out.println(a1.toString());
        System.out.println("Area of square = "+a1.area());

        Square a2 = new Square(2,2,0,2,0,0,2,0);
        System.out.println(a2.toString());
        System.out.println("Area of square = "+a2.area());

        Rectangle b1 = new Rectangle(2,5,0,5,0,0,2,0);
        System.out.println(b1.toString());
        System.out.println("Area of rectangle = "+b1.area());

        Kite c1 = new Kite(0,0,1,0,0,1,4,4);
        System.out.println(c1.toString());
        System.out.println("Area of kite = "+c1.area());
    }
}
```

\*\*\* โปรแกรมที่สร้างขึ้นจะต้องคำนวณพื้นที่ได้ถูกต้องแม้ว่าจะมีการสลับลำดับของจุดของสี่เหลี่ยมดังแสดงในตัวอย่าง

ผลลัพธ์ที่ได้คือ

```
java.awt.Point[x=0,y=0]
java.awt.Point[x=2,y=2]
java.awt.Point[x=0,y=2]
java.awt.Point[x=2,y=0]
Area of square = 4.0
java.awt.Point[x=2,y=2]
java.awt.Point[x=0,y=2]
java.awt.Point[x=0,y=0]
java.awt.Point[x=2,y=0]
Area of square = 4.0
java.awt.Point[x=2,y=5]
java.awt.Point[x=0,y=5]
java.awt.Point[x=0,y=0]
java.awt.Point[x=2,y=0]
Area of rectangle = 10.0
java.awt.Point[x=0,y=0]
java.awt.Point[x=1,y=0]
java.awt.Point[x=0,y=1]
java.awt.Point[x=4,y=4]
Area of kite = 4.0000000000000001
```