**วัตถุประสงค์ :**

1. เพื่อให้เข้าใจในหลักการของ Interface

1. โปรแกรม ResizeAndCompareShape.java จงใช้งาน Interface ที่กำหนดให้ต่อไปนี้

* จงสร้างสี่เหลี่ยมจัตุรัส และวงกลม ชนิดละ 3 ชิ้น โดยกำหนดให้ สี่เหลี่ยมจัตุรัสมีสีส้ม และ วงกลมมีสีม่วง และจัดเก็บโดยใช้ ArrayList หรือ Array แสดงข้อมูลของวัตถุทั้งหมด
* กำหนดให้ปรับขนาดสี่เหลี่ยมจัตุรัส 200% และ ปรับขนาดวงกลม 50% จากนั้นให้แสดงข้อมูลของวัตถุที่ทั้งหมด
* แสดงผลการเปรียบเทียบพื้นที่ของสี่เหลี่ยมจัตุรัส ชิ้นที่ 1 และชิ้นที่ 2
* แสดงผลการเปรียบเทียบพื้นที่ของวงกลม ชิ้นที่ 2 และชิ้นที่ 1

*<<interface>>*

*Comparable*

*+compareTo(object:Object):int*

*<<interface>>*

*Resizable*

*+resize(percent:int):void*

*<<abstract>>*

*Shape*

*+getArea():double* +setColor(c:String):void

+toString():String

#color:String

Square

+getArea():double

+toString():String

+getPerimeter():double

-width:int

Circle

+getArea():double

+toString():String

+getCircumference():double

-radius:int

class Square

1. Color: Orange width: 4

Area: 16 Perimeter: 16

2. Color: Orange width: 10

Area: 100 Perimeter: 40

3. Color: Orange width: 5

Area: 25 Perimeter: 20

========================

class Circle

1.Color: Violet radius: 3

Area: … Circumference: …

2.Color: Violet radius 6

Area: … Circumference: …

3.Color: Violet radius 10

Area: … Circumference: …

=====================================

After resize

1. Color: Orange width: 8

Area: 64 Perimeter: 32

2. Color: Orange width: 20

Area: 400 Perimeter: 400

3. Color: Orange width: 25

Area: 625 Perimeter: 100

========================

class Circle

1.Color: Violet radius: 1.5

Area: … Circumference: …

2.Color: Violet radius 3

Area: … Circumference: …

3.Color: Violet radius 5

Area: … Circumference: …

=====================================

Compare 2 Squares after resize

Square 1 is smaller than Square 2

Compare 2 Circle after resize

Circle 2 is bigger than Circle 1

2. โปรแกรม MovableLine.java จงใช้งาน Interface ที่กำหนดให้ต่อไปนี้

* สร้างเส้นตรง กำหนดให้ เส้นตรง 1 เส้น มาจากจุด 2 จุดใด ๆ โดยผู้ใช้สามารถระบุว่าจะสร้างกี่เส้น จากนั้นให้สุ่มตัวเลขพิกัด x,y สำหรับแต่ละจุด และแสดงข้อมูลและคำนวณความชันของเส้นตรง ดังตัวอย่างต่อไปนี้ 
  + สูตรคำนวณความชันเส้นตรง: Slope = (y1-y2) / (x1-x2)
  + สูตรคำนวณระยะห่างระหว่าง 2 จุด 
* เลื่อนเส้นตรง 1 ไปทางซ้าย เป็นระยะทาง 10 และด้านบนเป็นระยะทาง 20
* เลื่อนเส้นตรง 2 ไปทางขวา เป็นระยะทาง 30 และด้านล่างเป็นระยะทาง 100
* แสดงข้อมูลตำแหน่งของจุดต่าง ๆ หลังการดำเนินการข้างต้น

*<<interface>>*

*Movable*

Please enter number of line: 3

You need 6 Points to formed 3 Lines.

Line No. 1. From (5,7) to (10,12)

Slope is 1 Distant is 7.071068

Line No. 2. From (20,12) to (15,22)

Slope is -2 Distant is 11.18034

Line No. 3. From (30,10) to (24,20)

Slope is -1.666667 Distant is 11.6619

= = = = = = = = = = =

After Move Point

Line No. 1. From (-5,27) to (0,32)

Slope is 1 Distant is 7.071068

Line No. 2. From (50,-88) to (45,-78)

Slope is -2 Distant is 11.18034

Line No. 3. From (30,10) to (24,20)

Slope is -1.666667 Distant is 11.6619

*+moveUp(int y):void*

*+moveDown(int y):void*

*+moveLeft(int x):void*

*+moveRight(int x):void*

Point2D

-x : int

-y : int

+ Point2D ()

+ Point2D(x:int, y:int)

+getX():int

+getY():int

+setX(x:int):void

+setY(y:int):void

+toString():String