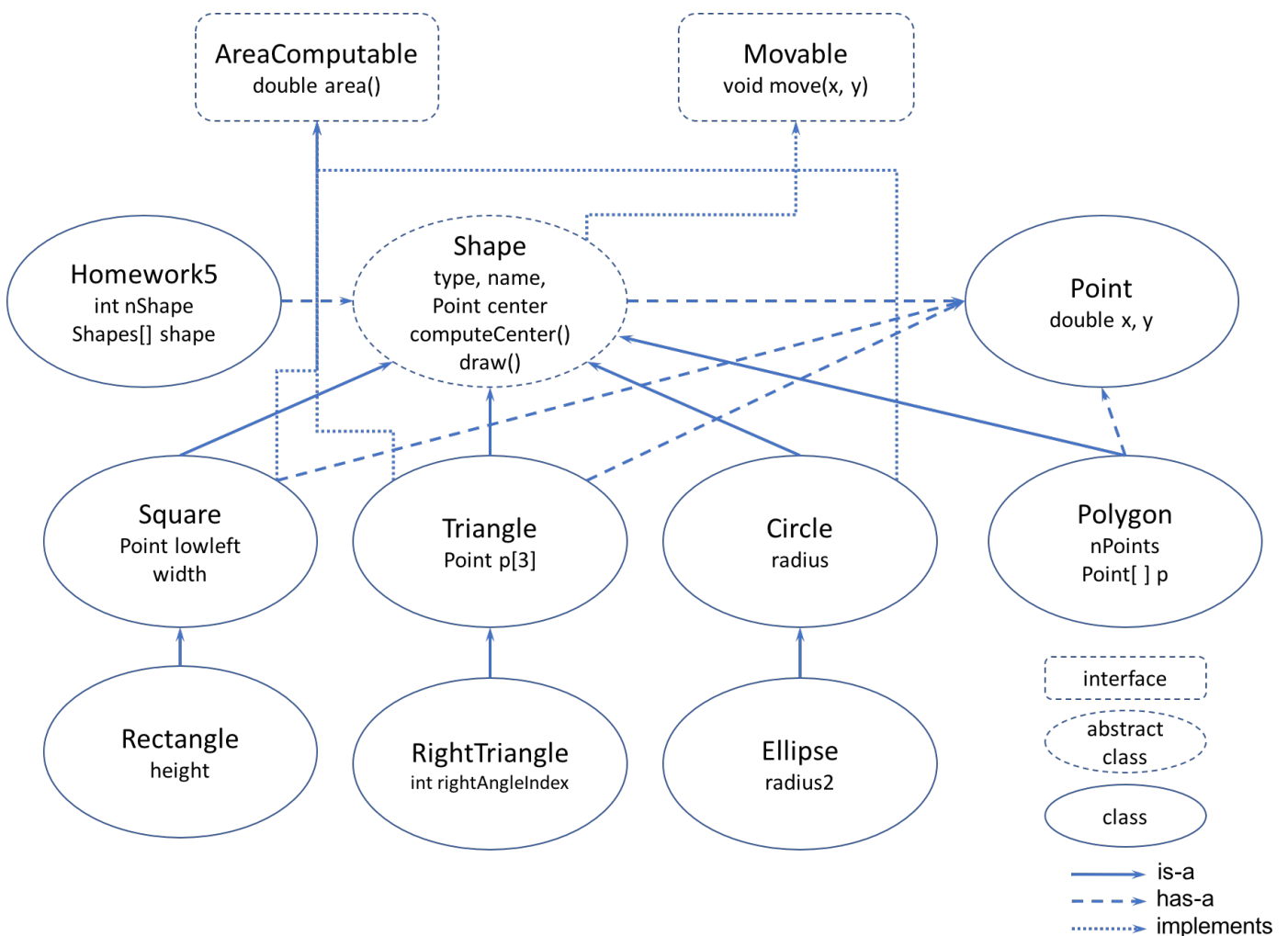


Homework 5

- 학번_hw5 (ex. 2018993906_hw5) 을 이름으로 하는 project 를 Eclipse 에서 새로 만들 것. Navigator 에서 project 명에 우클릭하면 나오는 메뉴에서 Export > General > Archive File 을 선택하여 zip 으로 project 전체를 묶어 제출함. 압축 파일 이름은 학번_hw5.zip 이 됨. (ex. 2018993906_hw5.zip) 제출 형식 따르지 않으면 감점 있음.
- Due: 11 월 30 일 (목) 23:00. 이후 제출은 받지 않음.

[Shape as Abstract Class + Interfaces] 이번 Homework 은 In-Class Activity #9 의 확장이다. 다음과 같은 class 와 interface 들을 디자인하고 구현한다. 주석을 달아 이해가 쉽게 하도록 함.



- 1) Homework5 class 는 unnamed package, 나머지 class 와 interface 들은 모두 Shapes package 에 속한다. (뒤에 설명할 IKDraw2D 와 IKDraw2DList 도 Shapes package 에 속함)
- 2) Shape 은 abstract class 로 abstract method 두 개를 가지는데, void computeCenter() 와 void draw(IKDraw2DList) 이다.
- 3) Square 는 lowleft (lower left corner point - Point type) 을 가진다.
- 4) Polygon 은 vertex 수가 자유로운 다각형이며, AreaComputable 이 아니다.

- 5) 모든 member variable 들은 protected 접근제어자로 정의하고, class 들은 public 으로 한다.
- 6) Shape 의 자손들과 Point 는 모두 두 개 이상의 constructor 를 가지며, 필요시 super(...)를 call 해야 한다. Shape 도 두개 constructor 가 있음.
- 7) Shape 과 그의 자손들, Point 는 모두 toString method 를 overriding 한다. 특히 Shape 자신을 제외하고 그의 자손들은 toString 에서 모두 super.toString()을 사용해야 하고, Point 의 toString 을 사용할 곳이 몇 군데 있음에 유의한다.
- 8) Rectangle 과 Square 의 constructor 에 주어지는 x, y 는 center 가 아니라 lower-left corner point 이다. 그래도 super(...) 를 call 해야 한다.
- 9) Triangle 과 RightTriangle 의 constructor 에는 center 가 주어지지 않고 세개 vertex 좌표만 주어진다. 역시 super(...)를 call 해야 한다.
- 10) Polygon 의 constructor 에는 nPoints 와 Point 배열이 주어진다. super(...)를 call 한다.
- 11) set~, get~ method 들은 필요한 경우에만 코딩하도록 한다.
- 12) Movable interface 에 void move(Point dp) method 가 있는데, shape 의 center.x += dp.x; center.y += dp.y 같이 주어진 dp 만큼 shape 을 이동시키는 것이다. 각 도형마다 move 에서 더 해야 할 일이 달라질 수 있다. 유의해서 overriding 한다.
- 13) 모든 value 들은 index 등을 제외하고는 모두 double 로 한다.

Homework5.java 에 있는 main method 에서 먼저 아래와 같은 출력이 나오도록 shape 들을 만들어 print 한다. (데이터를 똑 같이 할 것, In-Class 보다 area, lowerleft 등이 추가되어 프린트 함)

```
SQUARE (SQUARE1) center:(200.0,200.0) lowerleft:(100.0,100.0) width:200.0 area:40000.0
RECTANGLE (REC1) center:(150.0,250.0) lowerleft:(50.0,50.0) width:200.0 area:80000.0 height:400.0
TRIANGLE (TRI1) center:(233.33333333333334,316.6666666666667) vertex coord.:(150.0,150.0) (130.0,380.0) (420.0,420.0) area:67500.0
R-TRIANGLE (R-TRI1) center:(326.3333333333333,110.0) vertex coord.:(260.0,170.0) (260.0,80.0) (459.0,80.0) area:17910.0 right angle index:1
CIRCLE (CIRCLE1) center:(135.0,210.0) radius:70.0 area:15393.804002589986
ELLIPSE (ELLIPSE1) center:(355.0,140.0) radius:80.0 area:10053.096491487338 radius2:40.0
POLYGON (POLYGON1) center:(196.0,216.0) nPoints: 5 (150.0,220.0) (230.0,110.0) (270.0,160.0) (170.0,320.0) (160.0,270.0)
```

그리고 나서 모든 shape 을 (50, 50) 만큼 move 한 후 다시 print 한다.

```
After moving (50, 50)
SQUARE (SQUARE1) center:(250.0,250.0) lowerleft:(150.0,150.0) width:200.0 area:40000.0
RECTANGLE (REC1) center:(200.0,300.0) lowerleft:(100.0,100.0) width:200.0 area:80000.0 height:400.0
TRIANGLE (TRI1) center:(283.3333333333333,366.6666666666667) vertex coord.:(200.0,200.0) (180.0,430.0) (470.0,470.0) area:67500.0
R-TRIANGLE (R-TRI1) center:(376.3333333333333,160.0) vertex coord.:(310.0,220.0) (310.0,130.0) (509.0,130.0) area:17910.0 right angle index:1
CIRCLE (CIRCLE1) center:(185.0,260.0) radius:70.0 area:15393.804002589986
ELLIPSE (ELLIPSE1) center:(405.0,190.0) radius:80.0 area:10053.096491487338 radius2:40.0
POLYGON (POLYGON1) center:(246.0,266.0) nPoints: 5 (200.0,270.0) (280.0,160.0) (320.0,210.0) (220.0,370.0) (210.0,320.0)
```

마지막으로 모든 도형의 현재 상태를 2D graphics 로 그려본다. 2D graphics 를 위한 최소한의 class 들인 IKDraw2D 와 IKDraw2DList 를 배포한다. 이 두 파일의 내용은 고칠 수 없고, 있는 그대로 사용하도록 한다. main 에서 그릴 때 아래와 같이 하면 된다. 윈도우의 사이즈는 모든 도형이 다 그려지도록 충분한 크기로 잡는다. (아래: 500 x 500)

```
IKDraw2DList draw2DList = new IKDraw2DList();

for (int i = 0; i < nShape; i++) {
    shape[i].draw(draw2DList);
}

IKDraw2D draw2D = new IKDraw2D(500, 500, draw2DList);
draw2D.drawAll();
```

각 shape 의 draw method 는 예를 들면 Ellipse 의 draw 는 아래와 같이 구현한다. IKDraw2DList class 에 보면 각 shape 종류마다 method 들이 구현되어 있는데, 그것을 그냥 아래와 같이 call 한다.

```
public void draw(IKDraw2DList draw2DList) {
    draw2DList.drawEllipse(this);
}
```

다음은 2D 그래픽으로 그려진 window 를 캡처한 것이다.

