```java
abstract class GraphObject {
        public abstract void draw();
        public abstract void translate(double x, double y);
        public abstract void rotate(double theta);
        public abstract void scale(double factor);
}

class rect extends GraphObject {
        private int length, width;
        public rect(int l, int w)
                {l = length; w = width;};

        public void draw() {
                System.out.println("Drawing rectangle now");
        }

        public void translate(double x, double y) {
   System.out.println("Translating rectangle by " + x +", " + y +".");
        }

        public void rotate(double theta)
{System.out.println ("Rotating rectangle by " + theta + " degrees.");}

        public void scale(double factor)
{System.out.println ("Scaling rectangle by a factor of " + factor + ".");}


}

class circ extends GraphObject {
        private double radius, x, y;
        public circ (double xx, double yy, double rr) {
                x = xx;
                y = yy;
                radius = rr;}

       public void draw() {
                System.out.println("Drawing circle now");
        }

        public void translate(double x, double y) {
   System.out.println("Translating circle by " + x +", " + y +".");
        }

        public void rotate(double theta) {
                System.out.println("(No need to rotate a circle)");}

        public void scale(double factor)
{System.out.println ("Scaling circle by a factor of " + factor + ".");}


}

class Client {
        public static void main (String argv[]) {

        GraphObject o[] = new GraphObject[2];

        o[0] = new rect(1, 2);
        o[1] = new circ(1, 3, 3.2);
```

```
        for (int count = 0; count < o.length; count++) {
        o[count].draw();
        o[count].translate(1 + 0.5*count, 2.25+count);
        o[count].rotate(10+2*count);
        o[count].scale(1.5+count);

        }
}
}
```