Discussion Section 5 (09/25/2008)

Project Ideas:

This group exercise is designed to improve your understanding of object oriented design. The example we chose is based on both your project and the material from discussion section last week. While it may seem very similar to things you are required to do for your project, please do not interpret this exercise as advocating a particular design for your project. Naturally, you are free to write your project however you see fit. However, we hope that exploring some of these ideas will help you develop a clear sense of how you want to structure your project.

In the project we need to create 5 kinds of basic pictures: rectangle, filled rectangle, circle, filled circle, and line. An arbitrary number of such pictures can be grouped, and groups are pictures as well. The pictures are defined with: (rect *X Y W H*), (filledrect *X Y W H*), (circ *X Y R*), (filledcirc *X Y R*), (line X_0, Y_0 X_1 Y_1), (group P_1 … P_n).

The input language allows to perform the following operation on a picture: (move *P X Y*), (rotate *P D*), (scale *P S*), (draw *P*).

(color *R G B*) is to set the color for the pictures created from now on. (linewidth *W*) is to set the line width. For a filled picture, color is the filled color; otherwise, it is the outline color. The line width setting is applied to outline pictures only.

1. Draw the hierarchy given class names.

Shape	FillRect
Point	OutlineCircle
AbstractRect	FillCircle
AbstractCircle	Line
OutlineRect	Group

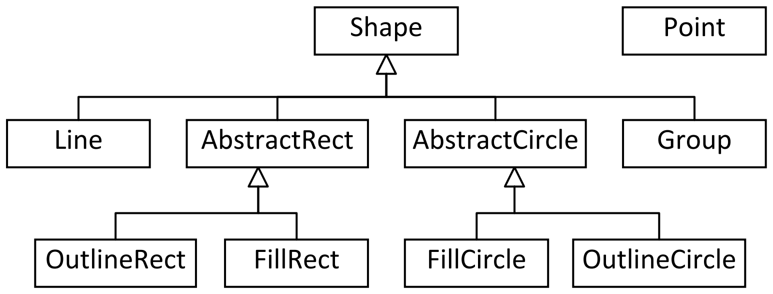
2. How do you enforce that all classes implement draw, move, rotate, scale?

3. Assign class variables.

4. Implement the draw methods for both the OutlineRect and FillRect class. The draw method should draw all 4 lines of the rectangle, followed by a command that determines whether or not the shape is filled. To draw a line, lets imagine we have a function "void draw(int x1, int y1, int x2, int y2)". Once the lines are drawn, specify a filled rectangle by the function "void fill()", otherwise specify an outlined rectangle using the function "void outline()". Examining the class Hierarchy, you should note the AbstractRect class. This is a good place to put methods that both OutlineRect and FillRect will use. In fact, if you structure your code correctly, you should be able to write both of these methods using just two function calls each (but this requires you to write some code in the abstract class).

5. Implement the move, scale, rotate, and draw methods for the Group class.

1. Draw the hierarchy given class names



2. How do you enforce that all classes implement move, rotate, scale, draw

Shape:	draw, move, rotate, scale (all abstract)
Group:	draw, move, rotate, scale
Line:	draw, move, rotate, scale
AbstractRect:	draw, move, rotate, scale
OutlineRect:	draw
FillRect:	draw
(same for circle)	

3. Assign class variables (Color is java.awt.Color)

Group:	ArrayList<Shape> children;
Line:	Point p1, p2; double linewidth; Color color;
AbstractRect:	Point p1, p2, p3, p4; Color color;
OutlineRect:	double linewidth;
AbstractCircle:	Point p; double r; Color color;
OutlineCircle:	double linewidth;