Practice Task 3:

This is an extension on the work from Practice 1. It includes my implementation of the Circle class, so you shouldn't look at this is you still want to work on Practice 1 on your own.

In this example we animate the shapes. There is an executable JAR file included in the project that you can run to see how it should work. There are two fields to the abstract class Shape, int xMotion and int yMotion. These are intended to control the motion. The ShapeOverlay class has a new method public void move() that shifts the positions of each Shape object according to it's xMotion and yMotion values. It draws the shape in it's existing location using the background color (to wipe out the old image) then updates x and y, then draws the shape in it's assigned color. The move function also includes some logic to change the direction of motion if a Shape object reaches the edge of the panel.

Right now, the “Move Once” button calls the ShapeOverlay's move() function, and you can see the shapes shift on screen.

Your task is to animate the shapes using a second thread.

Make the ShapeOverlay class implement Runnable. Note that I added a boolean class variable isRunning. You can use this to control the loop in your run() method that drives the animation. Your run method should repeatedly sleep for 20 milliseconds, then call move() to shift the shapes.

In the ApplicationFrame, you will need to add event handlers to the start and stop buttons. These should create and start the thread or kill it by switching the value of the Overlay's isRunning field.

This gives more practice with polymorphism and practice with multithreading and sleep() including the use of runnables. The graphics stuff might be interesting to look at too, just to make sure you understand how the code works.