San Jose State University | CS 151 - OO Design | Spring 2019

Homework 6

We will practice the design patterns from sections 5.4 - 5.6 of the textbook.

1. Start with the `ch04/animation` project and the modification of it in Lab 8 that allows you to animate multiple shapes.
2. Add this method to `MoveableShape`:

```
/**
 * Yields the bounding rectangle of this shape.
 * @return the bounding rectangle
 */
Rectangle getBounds();
```

3. Implement the method in the `Car` and `MoveableIcon` classes.
4. Provide a decorator class `BoxedShape` with a constructor

```
public BoxedShape(MoveableShape shape, int gap)
```

that, when drawn, yields the original shape with a rectangle along its bounds if `gap` is zero, or with as many pixels between the bounds and the rectangle on each side as given by `gap`. It's a decorator, so you should be able to apply it twice:

```
new BoxedShape(new BoxedShape(new CarShape(...), 0), 5)
```

5. Use the Composite pattern to group multiple shapes into one. A `CompoundShape` draws all of its shapes, moves each of them, and has a bounding box that is the smallest rectangle containing all individual bounding boxes. Provide a constructor

```
public CompoundShape(MoveableShape... shapes)
```

Note the varargs parameter. You should be able to call

```
new CompoundShape(new BoxedShape(...), new CarShape(...), new MoveableIcon(...))
```

6. Right now, `AnimationTester` simply moves all moveable shapes in each timer tick. Suppose we want it to do something more sophisticated, like stopping shapes that reach the boundary. That would be a different strategy. Provide an interface `MoveStrategy` with an abstract method

```
void process(List<MoveableShape> shapes)
```

7. Provide a class `SimpleMoveStrategy` that does what's currently done in `AnimationTester`, and a class `BoundedMoveStrategy` that only moves shapes whose bounds are contained inside a `Rectangle` that is given in the constructor.
8. Rename `AnimationTester` to `Animation`. Turn the `main` method into a method

```
public static void show(List<MoveableShape> shapes, MoveStrategy strategy, int width, int height)
```

I will call that method from my test cases. Here is an example. Make sure that it compiles with your classes with no change.

```
import java.awt.Rectangle;
import java.util.ArrayList;
```

```
import java.util.List;

public class AnimationTester
{
   public static void main(String[] args)
   {
      final int CAR_WIDTH = 100;
      List<MoveableShape> shapes = new ArrayList<>();
      shapes.add(new BoxedShape(new CompoundShape(new CarShape(200, 20, CAR_WIDTH),
            new MoveableIcon("dog.png", 100, 10),
            new MoveableIcon("dog.png", 150, 100)), 0));
      Animation.show(shapes,
            new BoundedMoveStrategy(new Rectangle(0, 0, 500, 200)),
            600, 200);
   }
}
```

9. As before, you need to provide javadoc for all classes and methods, and use spaces, not tabs. Make at least three Git commits (and preferably more).