

Amazing Rectangles

In this assignment you will create a program that animates rectangles with specific behaviors on the screen. The rectangles will have the following behaviors:

- **FallingRectangle** - This rectangle should fall at a rate of 1px per time cycle and smoosh when it hits the bottom of the screen. The smooshing should be completed by increasing its width by 2px, reducing its height by 1px, and moving the rectangle 1px to the left and 1px down. This rectangle will eventually disappear off the screen. To help you get started, the helper code provides starter code for this rectangle but you will need to add logic to emulate the smooshing!
- **BouncingRectangle** - This rectangle moves at a rate of 1px per time cycle in a horizontally or vertically direction (randomly determined in the constructor) and bounces off the sides of the draw area.
- **ColoringRectangle** - This rectangle remains stationary, but changes its fill color each time cycle by reducing the Red, Green, and Blue (RGB) by 1 unit for any RGB value greater than 0. Note that 0 is the lowest value you can specify for a RGB value. This rectangle should eventually turn black.
- Create your own amazing rectangle!

Tasks for you to complete before you begin programming are outlined below.

- 1) Review the helper code closely. Do you have any questions? If so, write them down and we will discuss them.
- 2) Create a UML diagram using Violet to illustrate how `ColoringRectangle`, `FallingRectangle`, `BouncingRectangle`, `AbstractRectangle` and `DrawingAreaComponent` relate to one another. You do NOT need to supply the methods and instance fields within the classes on the UML diagram.
- 3) Explain how polymorphism is used in this program.
- 4) Explain how the program performs animation.
- 5) What are the benefits of using the abstract class, `AbstractRectangle`?
- 6) What modifications would be necessary to add a new shape to this program?