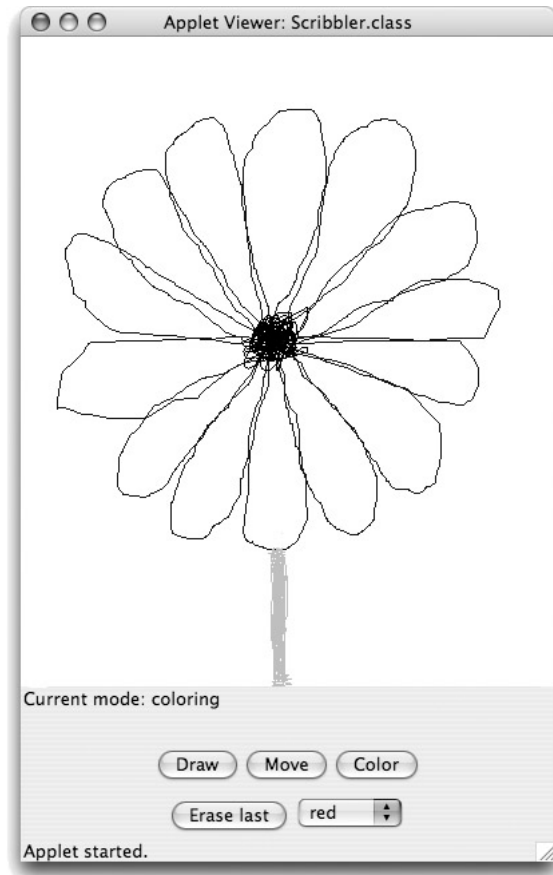


Recursion Lab

Scribbler

Objective: To gain more experience using recursion and recursive data structures.

This week, you will be implementing a simple drawing program we call “Scribbler.” A sample of what your program’s interface might look like is shown below.



The Scribbler program has three modes: Draw mode, Move mode, and Color mode. The modes behave as follows:

Draw mode: As the user drags the mouse around in the canvas, the program will draw lines following the path of the mouse.

Move mode: In move mode, the user can reposition a scribble by pressing the mouse while touching some part of the scribble and then dragging the scribble to its new location.

Color mode: Clicking on any scribble while in this mode should change its color to that selected in the color menu.

The program starts in Draw mode. Draw mode, Move mode, and Color mode are selected by pressing buttons, and the color used by Color mode is selected by choosing a color from a JComboBox filled with color names.

The program also has an “Erase Last” button that will erase the most recently drawn scribble. Pressing the button repeatedly will erase scribbles in the reverse order of which they were drawn.

How to Proceed We will provide you with a working but incomplete Scribble program as a starter for this week’s exercise. It supports only the Draw mode and a simplified Move mode that is capable of moving only the most recently drawn scribble. You will need to add code that will manage your scribbles to allow the various modes and the “Erase last” button to work correctly.

There are a number of step-by-step approaches you could take to complete this program, but it is important that you have a plan, and that you add functionality one step at a time. Here is one possible ordering of the tasks. We recommend that you develop and test your program incrementally – make sure you have a working implementation at each step before moving on to the next.

1. Implement a simplified Color mode. This will be similar to Move mode supported by the starter code, except that you set the color of the most recently drawn scribble instead of moving it. To do this, you will have to add a `setColor` method to the `ScribbleInterface` interface, and the `NonEmptyScribble` and `EmptyScribble` classes.
2. Implement a simplified Erase mode. Here, you respond to the “Erase last” button’s `actionPerformed` event by deleting the most recently drawn scribble from the canvas. For now, you will only be able to erase the most recently drawn scribble. A second button press will do nothing.
3. To implement the complete functionality for this program, you will need to keep track of a number of scribbles. You should do this by defining recursive classes to represent collections of scribbles. The interface for these classes will be called `ScribbleCollectionInterface`, and the classes implementing that interface will be named `EmptyScribbleCollection` and `NonEmptyScribbleCollection`.

Just as we do not know how many line segments will make up a scribble when we start to draw one, we will not know how many scribbles will be drawn and stored in a scribble collection. In your `Scribbler` class you will need a variable with type `ScribbleCollectionInterface` that will be initialized with an object created from `EmptyScribbleCollection`. You will add new scribbles to it using the constructor from `NonEmptyScribbleCollection` as they are drawn. Consider carefully what methods your scribble collection needs to support the functionality of the four modes. We have provided the skeleton of a `ScribbleCollectionInterface` interface, and `NonEmptyScribbleCollection` and `EmptyScribbleCollection` classes with our starter files.

- Draw mode needs to add a new scribble to the scribble collection after the dragging is done.
- The “Erase last” button needs to remove the most recently drawn scribble from the canvas and remove it from the scribble collection. This means there is now a new “most recently drawn” scribble, and a second press would remove that one, and so on. Be careful in the case when there are no scribbles left.
- Move mode and Color mode need to determine if a mouse click takes place on *any* of the scribbles on the canvas, not just the most recently drawn. This requires that your scribble collection classes be able to search through all of the scribbles until either a scribble is located that contains the click point, or it is determined that none of the existing scribbles contain the point. Once you have determined which scribble contains the click point, you can move or color that scribble as you did in the simplified versions of these modes.

Submitting Your Work Before turning in your work, be sure to double check both its logical organization and your style of presentation. Make your code as clear as possible and include appropriate comments describing major sections of code and declarations. As ever, please submit your work both on GitHub and on paper.