Parker Petersen

UMLDrawer Improvements

For assignment 5 I decided to implement the State Pattern in my UMLDrawer program. The biggest improvement is in the MainForm. Before the improvements, I had a variable called SelectedTool that was a string that changed depending on what tool you had selected. In the mouseup event listener in the drawing panel I had a very long conditional statement that checked what mode the tool was in and then performed the appropriate action such as placing a class symbol or moving or deleting a symbol. It worked, but it was a very long function. It was over 100 lines of code and was a bit of an eyesore.

To improve the design, I made a new abstract class called SelectedToolState that held a Target Drawing and had an abstract method called trigger that would perform the appropriate action on the Drawing. I then created subclasses for each of the different modes that SelectedTool has such as Class, relationship, move, and delete. This allowed me to remove the extremely long conditional statement from the MainForm and replace it with only 2 lines of code.

The MainForm now holds a class member called ToolState that is a SelectedToolState. When a different tool is selected, the state changes. When the user uses a tool on the drawing, ToolState.trigger is called, and whichever state is currently active calls its unique implementation of trigger.

I also added in a new button that randomly assigns new colors to all of the classes and relationship lines in the drawing. A user is then able to undo or redo this action just as they would any other command.