



Command Design Pattern: The client is the DrawRect class. The invoker is the ActionListener. Specific command is called when user click a command button, or move the rectangle around. Different from strategy pattern, command pattern have concrete classes that are not alternative to each other. The drawing program is allowed to have multiple commands of different types when user draws.

There are three types of general commands: undo, redo, execute. All the concrete commands are the subclass of an abstract Command class that contains the three abstract methods mentioned above. The drawingProgram has two types of Commands: StyleCommand that changes color or size, MoveCommand and NewRectCommand. While both MoveCommand and NewRectCommand are concrete command, StyleCommand is still an abstract class. Abstract class StyleCommand implement redo() and undo() methods and implement execute() as a skeleton method that call doCommand(). doCommand() is implemented in every subclass.

The skeleton method execute() requires its subclass to define doCommand() method. This shows template method strategy.

CommandHistory class only contains static methods. Because there's only one command history, there's no need to create multiple CommandHistory object. Thus all the methods are static. Because each concrete command class only need one command object in the program, to store command history, the old command need to be clone before being stored. Otherwise, the object will be kept updated. Thus Command need to implement Cloneable interface.