

I used the Memento Software Design Pattern for this week's assignment 04, last week I did the Command Software Design Pattern for assignment 03. Based on my observation from the two software design pattern I found both interesting, I personally prefer the use of Command Software Design Pattern. I have come to fully understand its meaning and to implement redo-undo functionality. As I find it much more comfortable to undergo my workload using stacks, I had to create the undo and redo functions which I used for my assignment. What stack is, it's a subclass of vectors that implements a last-in, first-out stack. Stack only defines the default constructor. Which creates an empty stack, this works with undoing and redoing. If shape is added to the undo stack, this will make the Command class, the undo method will be a purpose. For the Memento, I added the undo-redo functionality using Lists and ArrayLists. The originator essentially holds a canvas and it manages it. You are able to add and remove shapes to from the canvas. The Memento class always allows each state to be snapshotted into canvas which that be saved so you can retrieve/undo an old state. Personally, I mainly prefer and would rather use stacks the next I do have to using Command Software Design Pattern again over Memento. I was able to integrate the undo and redo with no straightforwardness and it worked close to perfection. Unfortunately, I did not have time resource to fully understand and implement a redo to operate more than once using Memento.