Design Pattern Notes

* Memento
  + Use for tracking the history of an object
  + You have 3 classes
    - Originator – The primary class that you are tracking the state history of
    - Memento – a state of the object at that particular time
    - Caretaker – Also known as the history, will keep a list of memento’s to return should the developer require this
* State
  + When you want to track the state of a particular object without needing to constantly update that object
  + You can implement an interface that the primary class will use and call the same “use this object” function on it
  + That way you can just keep creating new tools/objects but the primary class does not need to change
  + Example is painting software and tracking what tool the user is currently using
    - You just need to have each tool inherit a “tool” interface that the primary class can call functions on
      * Click to start
      * Release to finish etc
* Iterator
  + When you want to iterate through something like a browser history
  + You don’t want to for, loop through the list in the main function because if you happen to change the list/array you would need to change the main functionality as well
  + To resolve this we can create an iterator the browser history can use that will have iterator functionality
  + This way we can iterate, and still change the list properties without needing to change our main functionality
  + Diagram

    Description automatically generated