An object or system goes through changes

* E.g. a bank account gets deposits and withdrawals

There are different ways of navigating those changes

One way is to record every change (command) and teach a command to undo itself

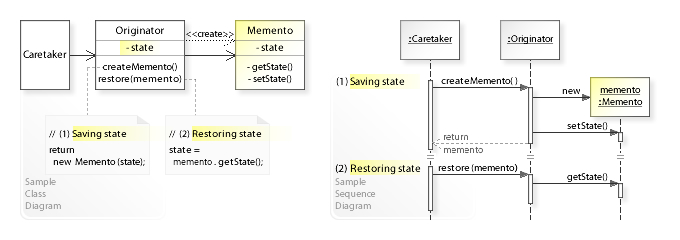
Another is to simply save the snapshot of the system

Memento is a token/handle representing the system state. Let us roll back to the state when the token was generated. May or may not directly expose state information.

Memento provides the ability to restore an object to its previous state (undo via rollback)

**GOFDefinition**  
Without violating encapsulation, capture and externalize an object’s internal state so that the object can be restored to this state later.

**Class Diagram**



**Design participants**

The memento pattern has three participants.

1. **Originator** – is the object that knows how to create and save its state for future. It provides methods createMemento () and restore (memento).
2. **Caretaker** – performs an operation on the Originator while having the possibility to rollback. It keeps track of multiple mementos. Caretaker class refers to the Originator class for saving (createMemento ()) and restoring (restore (memento)) originator’s internal state.
3. **Memento** – the lock box that is written and read by the Originator, and shepherded by the Caretaker. In principle, a memento must be in [immutable](https://howtodoinjava.com/java/basics/how-to-make-a-java-class-immutable/) object so that no one can change its state once created.

**Memento vs Command**

Command captures all information needed to perform certain action (not necessarily to *undo* this action). Your moves are basically commands.

Memento is a way to store state so that it is restoreable. Assume you'd have a class like GameState which represents the current state of the game. You'd be implementing Memento if your GameState would have methods like GameStateBackup createBackup() and restoreFromBackup(GameStateBackup).

Consider a game of chess where you'd want to be able to revert last x moves.

One way to do it would be to record all moves. You could then either "undo" moves. Or simple "replay" the first n-x moves. That would be the Command approach.

Another way would be to save the last x states of the game (and be able to restore them). This is the Memento approach.

You could actually use both patterns together. For instance, when implementation of "undo" would not be feasible you could record the state of the game before/after each move to make moves undoable.