Memento Design Pattern

Memento pattern is used to restore state of an object to a previous state.

**Known to use**

* Undo and restore operations in most software.
* Transactions are operations on the database that occur in an atomic, consistent, durable, and isolated fashion. A transaction can contain multiple operations on the database; each operation can succeed or fail, however a transaction guarantees that if all operations succeed, the transaction would commit and would be final. And if any operation fails, then the transaction would fail and all operations would rollback and leave the database as if nothing has happened. This mechanism of rolling back uses the memento design pattern.

**Related Pattern**

Command Pattern - Commands can use mementos to maintain state for undoable operations.

**Implementation**

Memento pattern uses three actor classes. Memento contains state of an object to be restored. Originator creates and stores states in Memento objects and Caretaker object is responsible to restore object state from Memento. We have created classes *Memento*, *Originator* and *CareTaker*.

*MementoPatternDemo*, our demo class, will use *CareTaker* and *Originator*objects to show restoration of object states.

**Memento**

* Stores internal state of the Originator object. The state can include any number of state variables.
* The Memento must have two interfaces, an interface to the caretaker. This interface must not allow any operations or any access to internal state stored by the memento and thus honors encapsulation. The other interface is to the originator and allows the originator to access any state variables necessary to for the originator to restore previous state.

**Originator**

* Creates a memento object capturing the originators internal state.
* Use the memento object to restore its previous state.

**Caretaker**

* Responsible for keeping the memento.
* The memento is opaque to the caretaker, and the caretaker must not operate on it.

