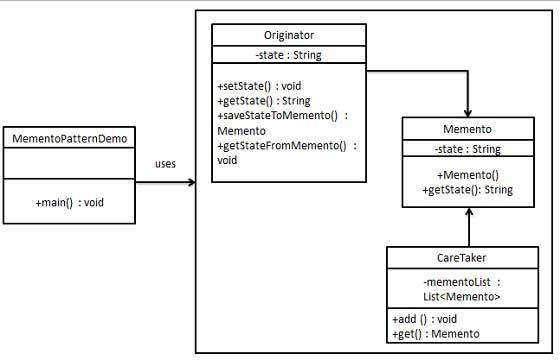
Memento pattern is used to restore state of an object to a previous state. Memento pattern falls under behavioral pattern category.

## **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.



/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Example:

return this.state;

}

public Memento saveStateToMemento() {

return new Memento(state);

}

public void getStateFromMemento(Memento memento) {

state = memento.getState();

}

}

/\*

\* create CareTaker class

\* CareTaker.java : restore object state from mememto

\*/

class CareTaker{

private List<Memento> mementoList = new ArrayList<Memento>();

public void add(Memento state) {

mementoList.add(state);

}

public Memento get(int index) {

return mementoList.get(index);

}

}

/\*

\* use Originator and careTaker class

\*/

public class TestMemento {

public static void main(String[] args) {

Originator originator = new Originator();

CareTaker careTaker = new CareTaker();

originator.setState("state #1");

originator.setState("state #2");

careTaker.add(originator.saveStateToMemento());

originator.setState("state #3");

careTaker.add(originator.saveStateToMemento());

originator.setState("state #4");

System.out.println("current state : " + originator.getState());

originator.getStateFromMemento(careTaker.get(0));

System.out.println("First saved state : "+originator.getState());

originator.getStateFromMemento(careTaker.get(1));

System.out.println("Second saved state : "+originator.getState());

}

}