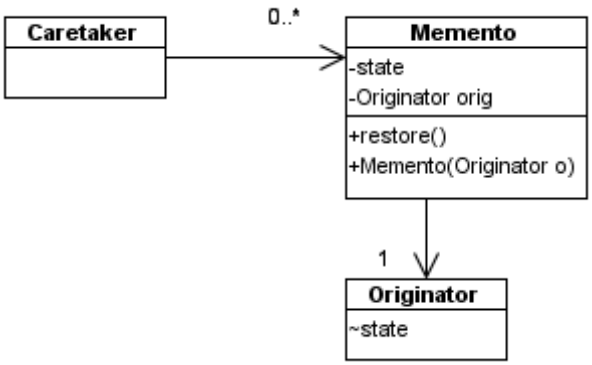
Supplementary Exercise on Memento Pattern

(a)



(b)

Originator
- Object whose state we want to save

Memento
- Object that saves the state of originator

Caretaker
- Object that manages the timing of saving states, saves the memento, and uses memento to restore state of originator

(c)

```
public class WatchMemento {

    private String modelNo;
    private String name;
    private int price;

    private Watch w;

    public WatchMemento(Watch w) {
        this.w = w;
        modelNo = w.modelNo;
        name = w.name;
        price = w.price;
    }

    public void restore() {
        w.setModelNo(modelNo);
        w.setName(name);
        w.setPrice(price);
    }
}
```

(d)

```
public class CareTaker {
    private Stack undoList;

    public CareTaker() {
        undoList = new Stack();
    }

    public void saveWatch(Watch w) {
        WatchMemento wm = new WatchMemento(w);
        undoList.push(wm);
    }

    public void undo() {
        WatchMemento wm = (WatchMemento)undoList.pop();
        wm.restore();
     }
}
```

(e)

```
public class TestMemento {

   public static void main(String[] args) {

        CareTaker ct = new CareTaker();
        Watch w = new Watch("A123", "Solar Watch", 3000);
         ct.saveWatch(w);
         w.setName("Next Solar Watch");
         ct.saveWatch(w);
         w.setPrice(3500);
         ct.undo();
         System.out.println(w);
  }
}
```