

Behavioral - Memento Design Pattern

Memento Design Pattern, elimizdeki mevcut nesnenin herhangi bir T anındaki durumunu kayda alarak, sonradan oluşabilecek değişiklikler üzerine tekrardan o kaydı elde etmemizi sağlayan bir desendir. Burada mevcut nesnenin özel bir halinden bahsetmemiz mümkündür. O hal ilgili tasarım kalıbı sayesinde sonradan da elde edilebilecektir. Memento Class - Sadece POJO class'ıdır ve içerisinde sadece state'i getter ile tutar. Originator Class - İçerisinde private şeklinde String state tutar ve state'i set(içerisinde setter bulunur) eder. save methodu state'i memento olarak kaydeder. restore methodu da undo işlemi için kullanılır. CareTaker - Birden fazla memento noktasını takip etmek için kullanılır. Save pointleri saklar.

//Memento

```
class Memento {  
    private String state;  
    public Memento(String state) {  
        this.state = state;  
    }  
    public String getState() {  
        return state;  
    }  
}
```

//CareTaker

```
class CareTaker {  
    private ArrayList<Memento> mementoList = new ArrayList<>();  
    public void addMementoToMementoList(Memento memento) {  
        mementoList.add(memento);  
    }  
    public Memento getMemento(int index) {  
        return mementoList.get(index);  
    }  
    public int getCount() {  
        return mementoList.size();  
    }  
}
```

//Origin

```
class Origin {  
    private String state;  
    public void setState(String state) {  
        System.out.println("Originator set to state : " + state);  
        this.state = state;  
    }  
    public Memento save() {  
        System.out.println("Originator saving to memento");  
        return new Memento(state);  
    }  
    public void restore(Memento memento) {  
        this.state = memento.getState();  
        System.out.println("Originator restoring to memento : " + state);  
    }  
}
```

```
class Test {  
    public static void main(String[] args) {  
        CareTaker careTaker = new CareTaker();  
        Origin origin = new Origin();  
        origin.setState("State 1");  
        origin.setState("State 2");  
        careTaker.addMementoToMementoList(origin.save());  
        System.out.println(careTaker.getCount());  
        origin.setState("State 3");  
        careTaker.addMementoToMementoList(origin.save());  
        System.out.println(careTaker.getCount());  
        origin.restore(careTaker.getMemento(1));  
    }  
}
```