

صفحه اصلی	سرفصل آموزشی	مقالات	پرسش و پاسخ
-----------	--------------	--------	-------------

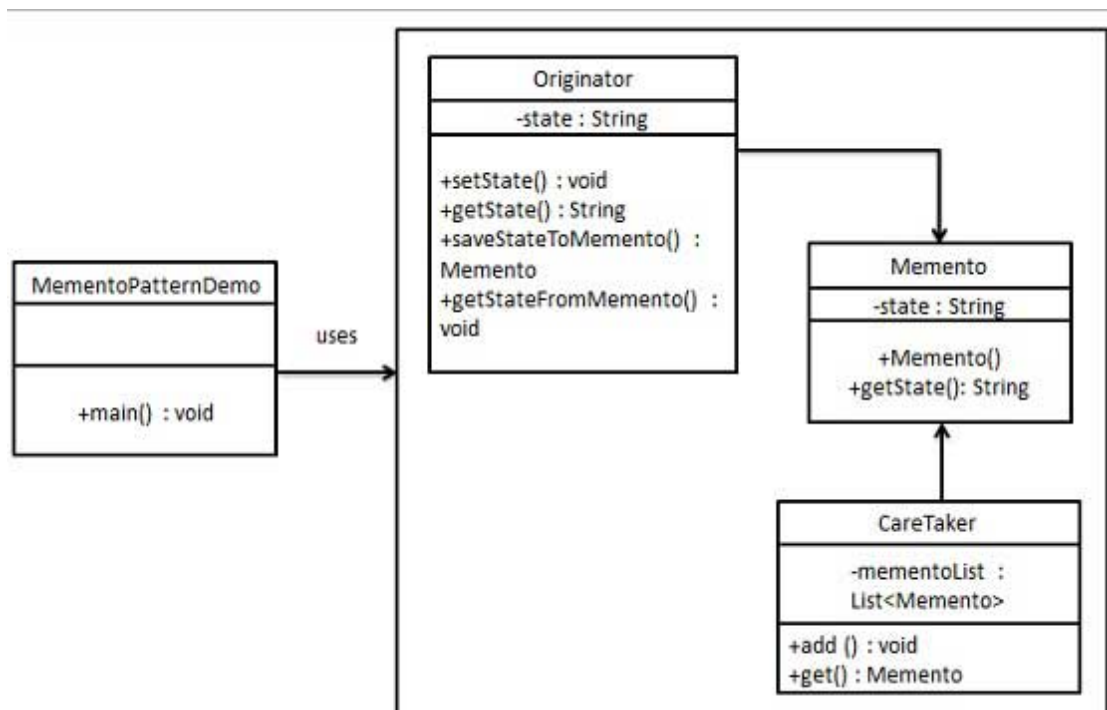
آموزش الگوهای طراحی

## Memento - الگوی Memento Design Pattern

الگوی Memento، برای برگرداندن وضعیت یک شیء به وضعیت قبلی استفاده میشه. الگوی Memento، زیرگروه الگوی رفتاری (behavioral) محسوب میشه.

### پیاده سازی

الگوی Memento از کلاس های actor استفاده می کنه. Memento حاوی وضعیتی از شیء است که قراره برگردانده بشه. کلاس Originator وضعیت ها را در اشیاء Memento ایجاد و ذخیره می کنه و شیء Caretaker، مسئول بازگرداندن وضعیت شیء از Memento است. کلاس های Originator، Memento و Caretaker را ایجاد می کنیم. MementoPatternDemo، در نسخه ی آزمایشی کلاسمان، از اشیاء CareTaker و Originator برای بازگرداندن وضعیت های اشیاء استفاده خواهیم کرد.



مرحله ی 1

ایجاد کلاس Memento.

Memento.java

```

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

```

## مرحله ی 2

ایجاد کلاس Originator.

Originator.java

```

public class Originator {
    ... private String state;
    ...
    ... public void setState(String state){
    ..... this.state = state;
    ... }
    ...
    ... public String getState(){
    ..... return state;
    ... }
    ...
    ... public Memento saveStateToMemento(){
    ..... return new Memento(state);
    ... }
    ...
    ... public void getStateFromMemento(Memento memento){
    ..... state = memento.getState();
    ... }
}

```

## مرحله ی 3

ایجاد کلاس CareTaker.

CareTaker.java

```

import java.util.ArrayList;
import java.util.List;

```

```

public 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);
... }
}

```

## مرحله ی 4

استفاده از اشیاء Originator و CareTaker.

MementoPatternDemo.java

```

public class MementoPatternDemo {
... 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());
... }
}

```

## مرحله ی 5

بررسی خروجی

Current State: . State. #4

First saved State: State #2  
Second. saved. State: . State. #3

در صورتی که سوال و یا نظری دارید، از بخش نظرات با ما در میان بگذارید.

برچسب ها: الگوهای طراحی جاوا الگوی Memento

الگوهای طراحی - الگوی Mediator درس قبلی

الگوهای طراحی - الگوی Observer درس بعدی

اشتراک مطلب در :

## خبرنامه Newsletters

در خبرنامه سافت اسکیل عضو شوید تا جدیدترین های سایت را بلافاصله در ایمیل خود دریافت کنید

عضویت

E-Mail

شما چه نظر و یا سوالی درباره این نوشته دارید؟

آدرس ایمیل

نام

پیغام مورد نظر خود را وارد کنید...

ارسال نظر

شبکه های

بخش های مختلف

دسترسی سریع

سرفصل های آموزشی

تبلیغات در سایت

مقالات تخصصی

درباره ما

پرسش و پاسخ

ارتباط با ما

تمامی حقوق  
باشد و استف  
قانون جرائم  
d

همکاری و استخدام