

بررسی الگوهای طراحی - State

توسط : محمد حسین زارع

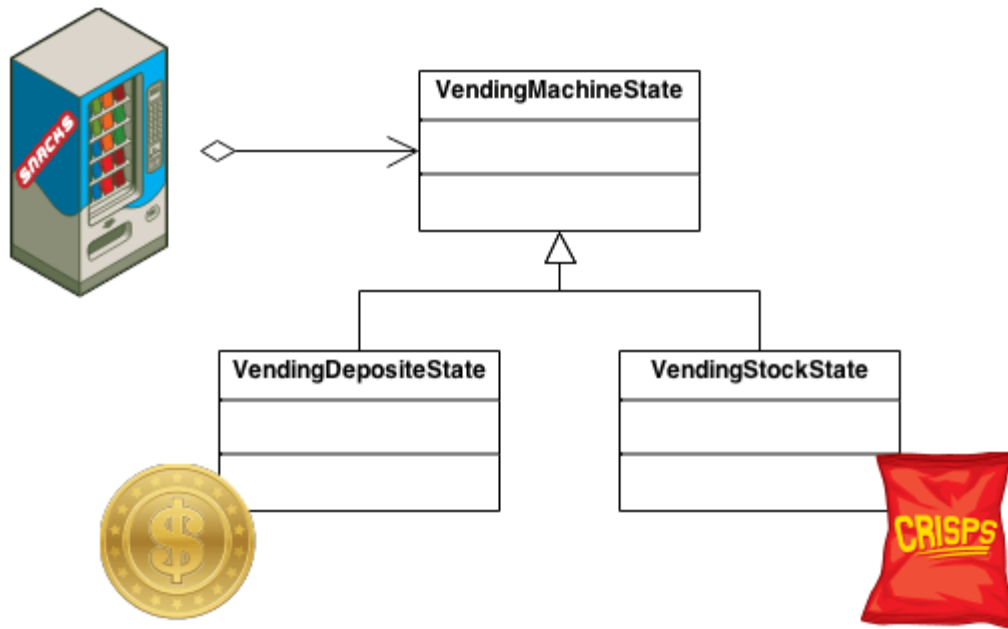
@zare88r : twitter
telegram

Join Us : @JavaLandCH

مفاهیم

- متمرکز کردن state behavior
- اشیائی برای نگهداری state ها
- **Open/Close Principle**
- نمونه ها :
- **Iterator** □
- **JSF** □

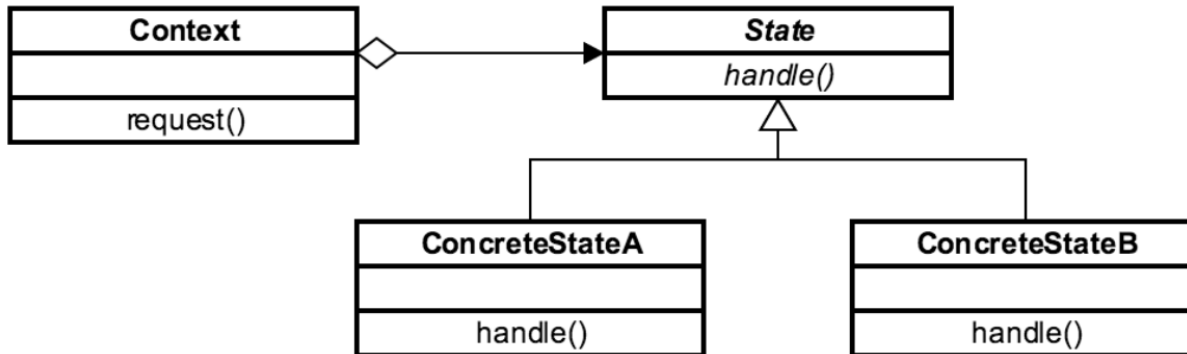
مفاهيم



طراحی

Context ▪

Abstract State – Concrete States ▪

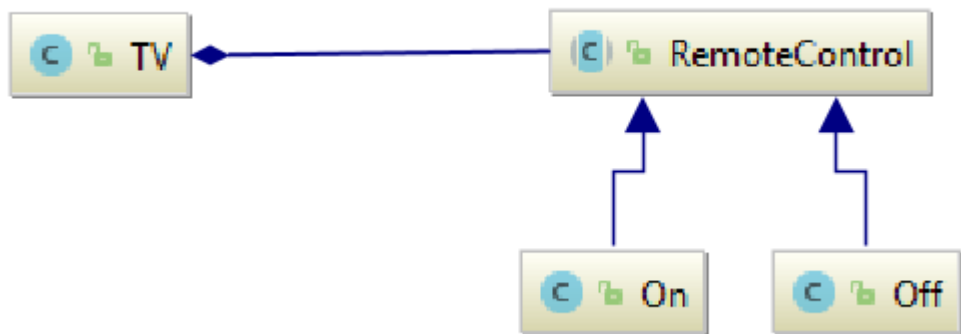


مثال 1

```
final static int ON = 0;
final static int OFF = 1;
int state = OFF;

public void pullChain() {
    if(state == ON) {
        System.out.println("Fan is already on");
    }
    else if (state == OFF) {
        System.out.println("Turning Fan on.");
        state = ON;
    }
}
```

یک نمونه



یک نمونه

```
public class Off extends RemoteControl {
    @Override
    public void pressSwitch(TV context) {
        System.out.println("I am Off .Going to be On now");
        context.setState(new On());
    }
}

public class On extends RemoteControl {
    @Override
    public void pressSwitch(TV context) {
        System.out.println("I am already On .Going to be Off now");
        context.setState(new Off());
    }
}
```

یک نمونه

```
public class TV {  
    private RemoteControl state;  
  
    public RemoteControl getState() {  
        return state;  
    }  
  
    public void setState(RemoteControl state) {  
        this.state = state;  
    }  
  
    public TV(RemoteControl state) {  
        this.state = state;  
    }  
  
    public void pressButton() {  
        state.pressSwitch(this);  
    }  
}
```


جمع بندی

- جلوگیری از پیچیدگی
- اضافه کردن state های دیگر ساده میشود
- نیاز به نوشتن کلاس های بیشتری دارد
- پیاده سازی شبیه به strategy دارد