

软件体系结构

Zhenyan Ji

— Beijing Jiaotong University —

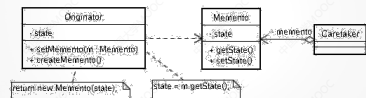
行为型模式--备忘录模式

备忘录模式

目的

- 在没有破坏封装性的前提下，捕获并外化对象的内部状态，以便该对象可恢复该状态。
- 需要将对象状态恢复到其以前状态（“undo”或“rollback”操作）。

备忘录模式



备忘录模式

Memento

- 存储Originator对象的内部状态。
- 不被originator外的其他对象访问，Memento类有两个有效的接口。Caretaker只能看见Memento的窄接口，只能将备忘录传递给其他对象。
- Originator可以看到宽接口，可以访问备忘录中所有有效数据，并将其还原到以前状态。理想情况下，只有制造了备忘录的Originator类可以访问备忘录的内部状态。

备忘录模式

Originator

- 创建含有对象当前状态快照的备忘录
- 使用备忘录恢复其内部状态

备忘录模式

» Caretaker

- 负责备忘录的安全管理
- 不对备忘录内容进行操作或检查。

备忘录模式:例子

```
class Memento {  
    private String state;  
    public Memento(String stateToSave) {  
        state = stateToSave;  
    }  
    public String getSavedState() {  
        return state;  
    }  
}
```

备忘录模式:例子

```
class Originator {  
    private String state;  
    public void set(String state) {  
        System.out.println("Originator: Setting state to " + state);  
        this.state = state;  
    }  
    public Memento saveToMemento() {  
        System.out.println("Originator: Saving to Memento.");  
        return new Memento(state);  
    }  
    public void restoreFromMemento(Memento m) {  
        state = m.getSavedState();  
        System.out.println("Originator: State after restoring from Memento: " + state);  
    }  
}
```

备忘录模式:例子

```
class Originator {  
    private String state;  
    public void set(String state) {  
        System.out.println("Originator: Setting state to " + state);  
        this.state = state;  
    }  
    public Memento saveToMemento() {  
        System.out.println("Originator: Saving to Memento.");  
        return new Memento(state);  
    }  
    public void restoreFromMemento(Memento m) {  
        state = m.getSavedState();  
        System.out.println("Originator: State after restoring from Memento: " + state);  
    }  
}
```

备忘录模式:例子

```
class Caretaker {  
    private ArrayList<Memento> savedStates = new ArrayList<Memento>();  
    public void addMemento(Memento m) {  
        savedStates.add(m);  
    }  
    public Memento getMemento(int index) {  
        return savedStates.get(index);  
    }  
}
```

备忘录模式

» 优点:

- 备忘录模式在不破坏封装性的前提下记录了对象的状态。
- 通过将状态的存储和恢复封装到Memento类中,实现了对Originator的简化。

» 缺点:

- 备忘录需要记录大量的状态信息,需要相当大的存储。