

Aula Prática - 18 Padrões de Comportamento - Memento

Intenção

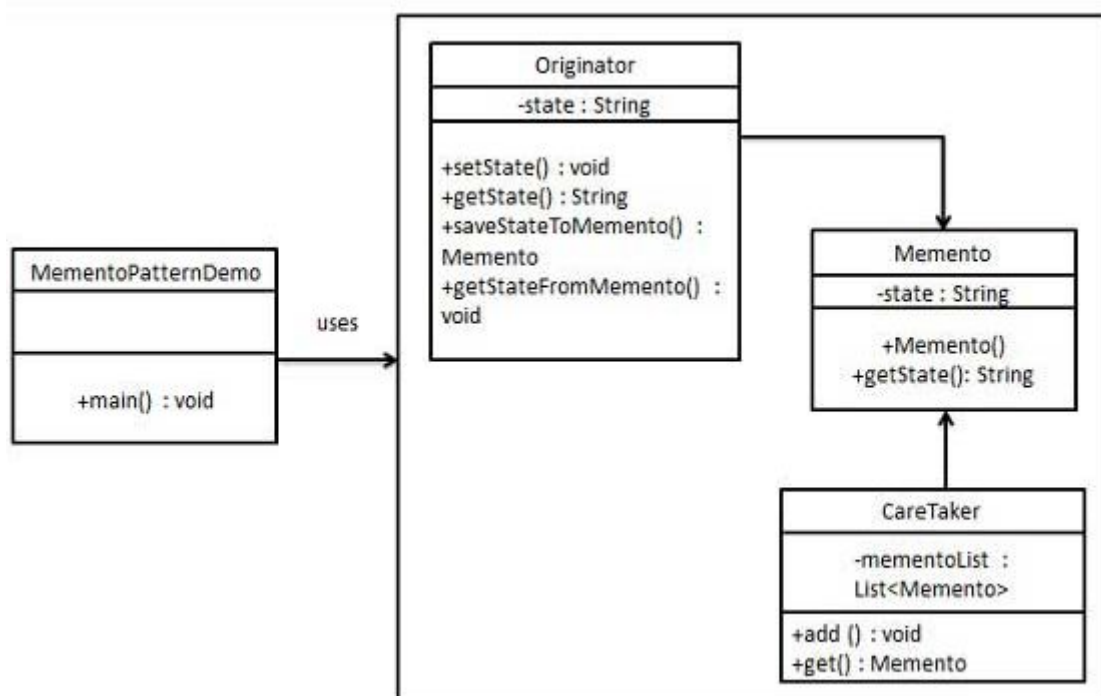
Sem violar o encapsulamento, capturar e externalizar o estado interno de um objeto para que possa ser restaurado posteriormente. Também conhecido como: Token.

Usar este padrão quando...

- O estado do objeto (ou de parte dele) deve ser armazenado para ser recuperado no futuro;
- Uma interface direta para obtenção de tal estado iria expor a implementação e quebrar o encapsulamento.

Vantagens e desvantagens

- Preserva o encapsulamento:
 - Retira do objeto original a tarefa de armazenar estados anteriores;
 - Caretaker não pode expor a estrutura interna do objeto, a qual tem acesso;
 - No entanto pode ser difícil esconder este estado em algumas linguagens.
- Pode ser caro:
 - Dependendo da quantidade de estado a ser armazenado, pode custar caro.



Passo 1

Crie a classe Memento.

Memento.java

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

Passo 2

Criar classe Originador

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();  
    }  
}
```

Passo 3

Criar classe 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);  
    }  
}
```

Passo 4

Use os objetos CareTaker e Originator

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());  
    }  
}
```

Passo 5

Teste sua implementação!