

Nassim MOROUCHE
Ramzy KERMAD
Vithursan SIVAKUMARAN

4 MOC
2020

MEMENTO

PRÉAMBULE

La présentation

Cette présentation a été faite avec peu de difficulté. Nous avons essayé d'expliquer au mieux le concept du patron de conception et avons donné un exemple concret d'utilisation. Nous espérons qu'à la fin vous aurez une vision plus claire à son propos.

Le sujet

Nous avons déjà utilisé ce pattern dans notre travail ou pour des projets personnels, nous n'avons donc pas eu de difficulté à le comprendre mais plutôt à vulgariser sa compréhension à ceux qui le verraient pour la première fois.

PRÉSENTATION DU DESIGN PATTERN

Qu'est ce que Memento ?

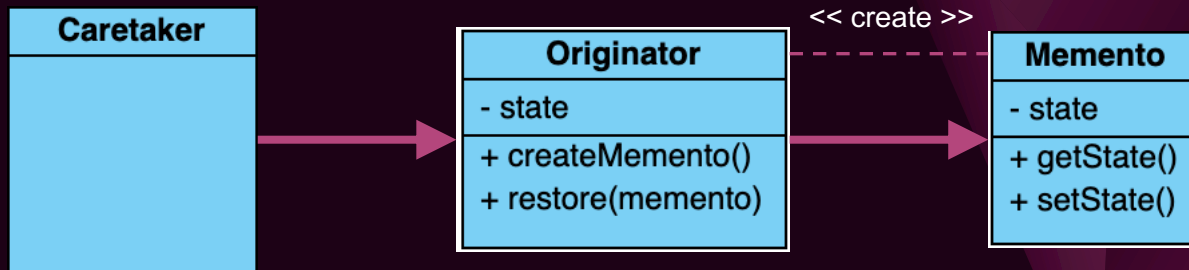
“

Patron de conception logiciel
Permet de restaurer un état précédent d'un objet

COMMENT FONCTIONNE-T-IL ?

- Le créateur
- Le gardien
- Un objet memento

STRUCTURE



EXEMPLE D'USAGE

```
class Originator {
    private String state;

    public void set(String state) {
        System.out.println("Originator: etat affecte a: "+state);
        this.state = state;
    }

    public Object saveToMemento() {
        System.out.println("Originator: sauvegarde dans le memento.");
        return new Memento(state);
    }

    public void restoreFromMemento(Object m) {
        if (m instanceof Memento) {
            Memento memento = (Memento)m;
            state = memento.getSavedState();
            System.out.println("Originator: Etat après restauration: "+state);
        }
    }

    private static class Memento {
        private String state;

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

EXEMPLE D'USAGE

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


EXEMPLE D'USAGE

```
class MementoExample {  
    public static void main(String[] args) {  
        Caretaker caretaker = new Caretaker();  
  
        Originator originator = new Originator();  
        originator.set("State1");  
        originator.set("State2");  
        caretaker.addMemento( originator.saveToMemento() );  
        originator.set("State3");  
        caretaker.addMemento( originator.saveToMemento() );  
        originator.set("State4");  
  
        originator.restoreFromMemento( caretaker.getMemento(1) );  
    }  
}
```

CONCLUSION

Complexité

- Gestion des stats à déterminer
- Créer un momento pour chaque stade avant modification

Modulable

- Facile à mettre en place grâce au pattern

Risque

- Se perdre dans la récupération des stats

MERCI!



Questions ?

CREDITS

Wikipedia:

➤ [https://fr.wikipedia.org/wiki/Mémento_\(patron_de_conception\)](https://fr.wikipedia.org/wiki/M%C3%A9mento_(patron_de_conception))

Pour la présentation :

- Presentation template by SlideCarnival