

TP 8

Exercice 1 : Memento

Reprendre le code sur moodle de l'exemple du Memento sur le choix des options pour l'achat en ligne d'un véhicule neuf. Le diagramme de classes est dans le cours mais vous pouvez le refaire.

A faire :

- Compléter la classe Utilisateur.java et faire tourner l'application.
- Expliquer en détails ce qui se passe à l'exécution de vos messages pour le chariot et pour le memento

Exercice 2 : Mediator

Reprendre le code de l'exemple du lava-linge sur moodle.

Parmi les classes il y a des classes support qui n'interviennent pas dans le pattern :

- Sensor sert à contrôler la température, elle est utilisée par Heater (chauffage)
- Motor : démarrage et rotation du tambour
- SoilRemoval utilisé par la classe Machine, propose 3 niveaux

Il y a deux interfaces principales :

- MachineMediator spécifie les appels d'opérations d'un objet à un autre
- Colleague dispose d'une méthode pour définir le mediateur pour la classe du collègue concret.

Les classes Colleague

- La classe Button est une classe collègue qui contient une référence à un médiateur. L'utilisateur appuie sur le bouton qui appelle la méthode `press()` de cette classe qui, à son tour, appelle la méthode `start()` de la classe médiatrice concrète. Cette méthode `start()` du médiateur appelle la méthode `start()` de la classe machine au nom de la classe Button.
- La classe Machine ci-dessus, qui contient une référence au médiateur, possède la méthode `start()` qui est appelée en appuyant sur le bouton par la classe médiateur, comme indiqué ci-dessus. La méthode possède une méthode `open()` du médiateur qui à son tour appelle la méthode `open()` de la classe Valve afin d'ouvrir la valve de la machine.
- La classe Heater. La méthode `on()` allume le radiateur et règle la température selon les besoins. Il vérifie également si la température est atteinte comme requis, il met `off()` la méthode. Le contrôle de la température et l'arrêt du chauffage se font via le médiateur. Après s'être éteint, il appelle la méthode `wash()` de la classe Machine via le médiateur pour démarrer le lavage.

Il y a deux classes de médiateur concret qui modélisent les programmes de la machine : CottonMediator et DenimMediator.

A faire :

- Le diagramme de classes complet est un peu touffu. Tâcher de le faire en laissant tomber les classes support. Vous pouvez le simplifier encore pour mieux visualiser la mise en œuvre du pattern Mediator en ne considérant qu'un médiateur concret.
- Compiler le programme et le faire tourner
- Analyser la trace résultat pour le CottonMediator (1ère partie de la méthode main de TestMediator)

Exercice 3 : Adapter

Nous avons réalisé un petit calculateur (la classe Calculateur) et nous avons utilisé une bibliothèque qui nous a fourni une opération add qui prend deux entiers et retourne la somme des deux. Nous l'avons donc utilisé pour implémenter l'addition dans notre calculateur.

A faire :

1. Dessiner ce premier diagramme de classes
2. Maintenant nous voulons faire évoluer notre calculateur mais on s'aperçoit que le calcul a changé et que l'opération add qu'elle fournit prend maintenant deux nombres décimaux (flottant). Nous ne voulons pas modifier l'interface de notre calculateur car il faudrait changer tout le code client qui utilise l'addition.
Proposez une solution utilisant le patron Adapteur :
 1. donner le rôle des classes dans le pattern,
 2. faire les modifications sur le diagramme de la première question.

Dépôt sur moodle :

Faire une archive par exercice. Vous pouvez les regrouper pour un dépôt unique.

- Exercice 1 :
 - la classe Utilisateur.java
 - les explications de l'exécution
- Exercice 2 :
 - le diagramme de classes
 - l'analyse de la trace
- Exercice 3 :
 - Le diagramme final
 - les explications sur le pattern Adapter