

Design Pattern – Labo 3 – 19.04.2015

Auteurs

Horia Mut / Quentin Jeanmonod / Lukas Bitter

But

Appliquer le composite, decorator et state patterns à un programme lié à la création d'un genre de menu d'achats.

Description

Le programme affiche le contenu d'un panier de fruits. Le panier est rempli à fur et à mesure par l'utilisateur en ajoutant soit des différents fruits soit un autre panier. Panier dans un panier, illogique, mais Box in à Box, ça c'est du possible.

Utilisation des design patterns

Le code a été écrit de façon à ce que le pattern soit visible dans chaque cas.

Composite

Pour gérer les paniers et les fruits sans une explosion de classes il faut les gérer comme étant du même type. Les CComposite (panier) comme les CLeaf (fruit) sont du type CComponent.

La classe CComponent est une classe virtuelle pure qui définit une interface pour dire les classes spécialisées. Dans notre cas, nous avons spécialisé la classe CLeaf avec les classes CBanana, CApple et CStrawberry pour respecter une structure cohérente, mais il n'est pas nécessaire de faire cela pour respecter le Design Pattern, en effet les Fruits sont directement des Composants, donc peuvent dériver directement de CComponent.

Decorator

Le Decorator doit décorer un CComponent. Dans ce cas, il est une spécialisation de cette classe virtuelle qui redéfinit la méthode Print et contient un attribut de plus, le char définissant la décoration.

La classe CDecorator définit les méthodes et donne leur implémentation, mais le pattern nous demande de la spécialiser afin de pouvoir imbriquer les décorations une dans l'autre. Ceci est fait avec les classes CDecoratorStar, CDecoratorHash etc.

Vu que le CDecorator agit sur un CComponent et est lui-même un CComponent, il est possible de décorer un CLeaf (fruit) tout autant qu'un CComposite (panier).

State

Le design pattern state est utilisé pour changer dynamiquement le comportement d'une classe et la faisant changer son état à chaque itération ou interaction du user, ou un évènement quelconque.

Dans notre cas, les états changent quand l'utilisateur écrit dans la console et valide son entrée.

L'état passé est remplacé par un nouvel état différent et l'ancien est effacé, ainsi de suite jusqu'à ce que l'objet possède un état final.