

approcheObjet

Gabriel Marie brisson

October 2023

1 Introduction

L'objectif de ce TD est d'appréhender les différences entre les Design Patterns. J'ai pu découvrir la modularité, la flexibilité et la maintenabilité que les Design Patterns peuvent m'apporter. Pour lancer le code, il faut choisir soit le sous-répertoire "design pattern Composite" ou "design pattern Decorator", puis lancer la commande en précisant le chemin jusqu'au main. Dans "design pattern Composite":

- "javac Builder/BuilderExample.java"
- "javac DesignPatternObserver/Apply.java"
- "javac Factory/Application.java"

Dans "design pattern Decorator":

- "javac Boulangerie.java"

2 Design Patterns Utilisés

2.1 Decorator Pattern

J'ai mis en œuvre le pattern Decorator. Comme les poupées russes, on emboîte les objets. Ceci est pratique pour vérifier l'emboîtement des ajouts.

2.2 Composite Pattern

Dans l'autre implémentation, j'ai dû coder le Design Pattern Composite pour améliorer la gestion des recettes. C'est comme des patchs que l'on rajoute à notre objet. La vérification de la cohérence des patchs est plus difficile à mettre en place, mais la création de l'objet est plus simple, ainsi que la gestion des ingrédients.

2.3 Builder Pattern

J'ai également utilisé le Design Pattern Builder pour la fabrication des gâteaux. L'intégration du Builder Pattern permet de créer des objets prédéfinis sans se soucier de leur implémentation. Ceci permet d'éviter de se prendre la tête. Cependant, je me rends compte qu'il aurait été préférable que je le rajoute à Decorator au lieu de Composite, car Decorator est plus complexe à créer. Les caractéristiques sont les suivantes :

- Interface pour la création d'objets.
- La création d'objets en étapes, avec un contrôle précis sur chaque étape.

2.4 Factory Pattern

Dans une autre approche, j'ai implémenté une version utilisant le Design Pattern Factory pour créer nos différentes recettes de gâteaux. Les caractéristiques sont les suivantes :

- Délègue la création d'objets aux sous-classes ou à des méthodes.
- Gestion des dépendances en évitant de créer directement des objets.

2.5 Observer Pattern

Pour la gestion de la boulangerie et la vente des gâteaux, nous avons mis en place le Design Pattern Observer. J'ai créé un objet de type Vendeur pour déclencher la fabrication de certains types de gâteaux lorsque le stock est inférieur à un seuil prédéfini.