

TP 5A - Génie Logiciel

Patrons de conception: Composite

Nicolas Delanoue

(1 séance de 1h20)

Objectif : Se familiariser avec le patron de conception *Composite*. Amélioration d'un code existant (notion de *refactoring*).

1 Cahier des charges et prototype

On souhaite faire un programme de gestion d'un stock de pièces. On considère deux types de pièces :

- Les pièces de base sont caractérisées par un prix unitaire et un nom. Leur prix est donné par le prix unitaire.
- Les pièces composites sont caractérisées par un nom, un prix d'assemblage et un ensemble de pièces qui peuvent être, soit des pièces composites, soit des pièces de base. Le prix d'une pièce composite résulte du calcul de la somme des prix des pièces la constituant à laquelle on ajoute le prix d'assemblage.

Le rôle du stock est de référencer chaque pièce créée et de fournir une méthode permettant de lister les pièces créées ainsi que leur prix.

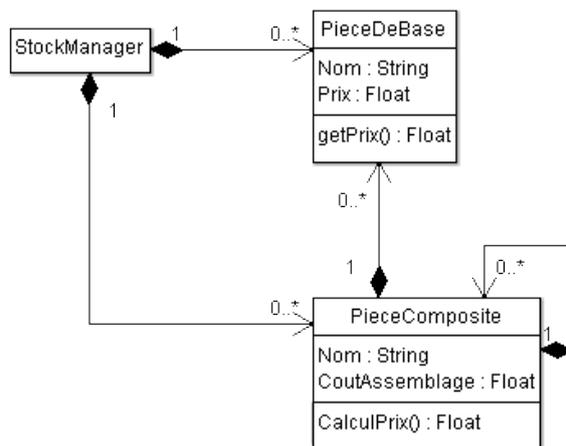


FIGURE 1 – Diagramme de classe UML du projet sujet.

Le code `TPComposite_eclipse_sujet` est un prototype du programme (voir la figure 1 pour la modélisation UML).

2 Travail à réaliser

Dans un premier temps, familiarisez-vous avec le code `TPComposite_eclipse_sujet`. On pourra représenter sous forme arborescente les relations qui existent entre les pièces déjà créées dans le `main`.

L'objectif est de faire évoluer la conception de ce programme afin de réduire sa complexité (le lanceur `Main` restant similaire). Pour cela, on propose de s'inspirer du patron de conception *Composite* (voir la figure 2). Le travail à réaliser est le suivant :

- Proposer un diagramme de classe UML.
- Ajuster le code en conséquence.
- On donnera les diagrammes de séquence avant et après refactoring pour le scénario de calcul du prix total du stock (méthode `getPrix()` de la classe `StockManager`).

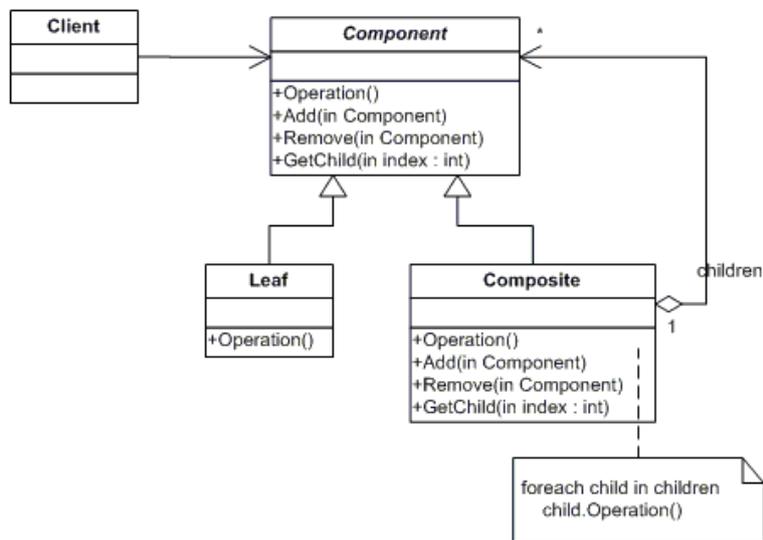


FIGURE 2 – Diagramme de classes du patron de conception *Composite*.