

MIN17201 - Programmation, GL, preuve - Sujet 1

Stéphane Lopes

Durée: 1h30

Contraintes techniques

- Le projet doit se construire avec *maven package* ou *gradle build*.
- La construction doit exécuter les tests unitaires.
- Les tests unitaires doivent être implémentés avec JUnit 4 ou 5.
- La clarté/lisibilité du code sera également un critère évalué.

Exercice

On veut proposer une application pour fabriquer des pizzas.

Une pizza possède obligatoirement un nom (*4 fromages, ...*) et une sauce (*tomate, crème fraîche, ...*). Une pizza est également composée d'ingrédients simples (*mozza., champignon, ...*) et de préparations.

Un ingrédient simple possède un nom et un coût. Une préparation possède également un nom et est composée d'ingrédients simples et d'autres préparations.

Le coût d'une préparation est la somme des coûts de ses ingrédients (simples et préparations). Le coût d'une pizza est la somme des coûts de ses ingrédients (simples et préparations). On veut pouvoir calculer le coût d'une pizza.

1. En vous appuyant sur le pattern *Composite*, proposer une implémentation pour les ingrédients en réalisant les tests suivants :
 - (a) instanciation d'un ingrédient simple *mozza. de coût 3*
 - (b) instanciation d'une préparation *reine composée de mozza. de coût 3 et de champignons de coût 5*
 - (c) calcul du coût de la préparation précédente (*reine*)
 - (d) il est impossible d'ajouter une préparation à elle-même
 - (e) il est impossible d'ajouter une préparation comme descendante d'elle-même
2. En vous appuyant sur le pattern *Builder*, proposer une implémentation pour les pizzas en réalisant les tests suivants :
 - (a) instanciation d'une pizza *margarita (sauce tomate, mozza.)*
 - (b) calcul du coût d'une pizza *reine supplément mozza. (sauce tomate, mozza., préparation reine)*