

# TP : Créez votre propre Café Deluxe avec le Pattern Decorator

## Contexte

Bienvenue dans le monde passionnant de la simulation de barista ! Vous venez d'ouvrir un café innovant, "Café Deluxe", où chaque boisson est unique. Votre tâche est de mettre en place un système permettant de composer des boissons, où chaque ajout d'ingrédient ou d'amélioration sur une boisson de base est représenté par un décorateur.

## Objectif

L'objectif de ce TP est de vous faire pratiquer le pattern Decorator en Java, en simulant un système où vous pouvez composer des boissons à l'aide d'ingrédients et d'améliorations. Vous allez créer des classes de base pour les boissons et des décorateurs pour chaque ingrédient ou amélioration.

## Tâches

### 1. Classe de base : Beverage

Créez une classe abstraite `Beverage` avec une méthode `getDescription()` et une méthode `cost()` qui renvoie le coût de la boisson.

### 2. Implémentations concrètes

Créez quelques classes concrètes de boissons de base telles que `Espresso`, `Latte`, et `FilterCoffee`. Chacune doit étendre `Beverage` et implémenter les méthodes `getDescription()` et `cost()`.

### 3. Classe de base pour les décorateurs : CondimentDecorator

-Créez une classe abstraite `CondimentDecorator` qui étend `Beverage`. Assurez-vous que cette classe permette aux décorateurs d'ajouter des ingrédients au `getDescription()`. `CondimentDecorator` doit également redéfinir `cost()`.

### 4. Implémentations des décorateurs

Créez plusieurs classes de décorateur, par exemple `Milk`, `Soy`, `Mocha`, `Whip` (crème fouettée). Chacune doit étendre `CondimentDecorator` et implémenter `getDescription()` et `cost()`.

### 5. Main Application : Simulateur de Café

Écrivez une classe `CafeSimulator` avec une méthode `main()` qui créera plusieurs combinaisons de boissons en ajoutant différents décorateurs et imprimera la description et le coût de chaque boisson.

## Exemple :

Imaginez que vous commandez un **Espresso** avec double **Mocha** et **Whip**, le programme devrait afficher une description correcte et le coût total.

## Bonus

- Ajoutez un système de taille pour les boissons (petit, moyen, grand) qui influence le coût.

- Implémentez un menu interactif en console où l'utilisateur peut choisir sa boisson et ses améliorations.