

## MAPC / TD – Design Pattern Decorator

---

### Expansion d'un point de vente de boissons chaudes<sup>1</sup>

L'objectif de cet exercice est de proposer un modèle de conception pour répondre à l'expansion du point de vente de boissons chaudes, StarBuzz.

StarBuzz propose à la base 4 types de boissons : HouseBlend, DarkRoast, Decaf et Espresso. Chaque boisson dispose d'une description et peut calculer son prix. Il existe une large variété d'ingrédients additionnels pour les boissons, tels que lait, soja, mocha. Ces ingrédients modifient la description et le coût d'une boisson.

Initialement, la conception de StarBuzz pour son système de gestion des boissons est la suivante.

- Classe abstraite Beverage, qui offre l'interface getDescription() et cost().
- Les sous-classes HouseBlend, DarkRoast, Decaf et Espresso, qui redéfinissent getDescription() et cost().

Commentez les approches suivantes pour réaliser l'extension du système initial.

1. Ajout d'une sous-classe par combinaison d'ingrédients. Par exemple : EspressoWithSoyAndMocha, HouseBlendWithSoyAndMocha, DecafWithWhipAndSoy, HouseBlendwithMilkAndMocha, etc.
2. Gestion des ingrédients dans la classe mère Beverage : hasMilk(), setMilk(...), hasSoy(), setSoy(...). Comment gérer les boissons avec l'ingrédient doubleSoy représenté par un booléen ?
3. Utilisation du design pattern Decorateur

Implantez la troisième approche.

---

<sup>1</sup> Tiré du livre Head First Design Patterns