

## Mise en oeuvre de patrons de conception

### Exercice 1

Un diagramme de classes correspondant à l'énoncé suivant vous est proposé au verso.

Des marchandises sont transportées sous la forme de cargaisons (cargaison fluviale, cargaison aérienne, ...); une cargaison est caractérisée par la distance sur laquelle elle est transportée.

Pour chaque cargaison, on pourra : 1) ajouter des marchandises si l'encombrement maximal n'est pas atteint, 2) rechercher si une marchandise existe dans la cargaison et sa place, 3) retirer une marchandise de la cargaison, 4) calculer le coût du transport, 5) visualiser une cargaison, etc.

Une cargaison ne peut réunir qu'un nombre limité de marchandises qui dépend d'un encombrement total de ces marchandises à ne pas dépasser. Cet encombrement est soit calculé suivant le poids total, soit suivant le volume total des marchandises. L'encombrement influe sur le calcul du coût de transport de la cargaison ; on distingue donc plusieurs types de cargaisons selon le moyen de transport utilisé.

cargaison	encombrement	coût	limite
fluviale	poids	distance × encombrement	encombrement ≤ 300'000
routière	poids	4 × distance × encombrement	encombrement ≤ 38'000
aérienne	volume	10 × distance × encombrement	encombrement ≤ 80'000
aérienneUrgente	volume	2 × coût cargaison aérienne	encombrement ≤ 80'000

1. Prenez le temps de lire et comprendre le diagramme UML proposé
2. Détaillez tous les appels de constructeurs/attributs/méthodes impliqués dans le scénario suivant : *on crée une cargaison aérienne, puis on lui ajoute 2 marchandises différentes ; finalement on calcule le coût* ; l'idée est de montrer les appels aux éléments de la classe elle-même, à différencier avec ceux de la super/sous-classe considérée.
3. Identifiez le(s) design pattern(s) mis en oeuvre dans la conception UML proposée.

### Exercice 2

On considère que les recettes de préparation d'un café et d'un thé sont données par le code ci-joint : il y a énormément de duplication de code : **il faut réviser la conception !!!!**

4. Proposez un diagramme de classes UML permettant de "minimiser" le code dupliqué.
5. Le détail des algorithmes de préparation du thé et du café est très similaire : quel design pattern va vous permettre d'éviter la duplication du code ? Proposez une nouvelle version du diagramme de classes.
6. Ajoutons une nouvelle boisson chaude : le cacao. 1) Il faut faire bouillir du lait et non de l'eau. 2) La préparation consiste à mélanger la poudre de cacao avec le lait. 3) On peut ajouter en supplément de la chantilly. Proposez une nouvelle version du diagramme de classes, évitant toujours toute duplication de code.

