

# Programmation Orientée Objet : JAVA

## TP6

### Héritage, polymorphisme et classe abstraite

Une entreprise d'import/export souhaite automatiser le calcul du coût de transport des marchandises qui sont transportées sous forme de cargaisons. Chaque **Marchandise** est caractérisée par une *référence*, un *poids* et un *volume*.

Une **Cargaison** est par ailleurs réalisée par une collection de marchandises. Elle est également caractérisée par :

- la *distance* sur laquelle elle est transportée, donnée à la création de la cargaison sous forme d'un nombre entier de kilomètres.
- l'*encombrement* total de ces marchandises qui est soit le poids total, soit le volume total des marchandises, selon le type de transport utilisé.
- la *limite* d'encombrement total de ces marchandises à ne pas dépasser.

Le coût de transport se calcule différemment selon le type de cargaisons et dépend de la distance de transport et de l'encombrement des marchandises de la cargaison. On distingue trois types de cargaisons selon le moyen de transport utilisé. On précise qu'une cargaison ne peut contenir qu'un nombre limité de marchandises qui dépend d'un encombrement total de ces marchandises à ne pas dépasser. Les différents types de cargaison et leurs caractéristiques sont donnés par le tableau suivant:

Type	Encombrement	Coût	Limite
Fluviale	poids	distance $\times$ encombrement	300000
Routiere	poids	4 $\times$ distance $\times$ encombrement	40000
Aerienne	volume	10 $\times$ distance $\times$ encombrement	80000

Pour chacune de ces trois catégories de cargaison, on désire disposer des méthodes suivantes :

- **ajouter**: qui permet d'ajouter une marchandise dans une cargaison si cela est encore possible ( $\text{encombrement} \leq \text{limite}$ ).
  - **cout**: qui retourne le cout total du transport d'une cargaison.
1. Définissez la classe **Marchandise** et les classes permettant de gérer la hiérarchie des cargaisons. Votre solution doit être fondée sur la notion de *classes abstraites* et de *polymorphisme* et doit respecter le *principe d'encapsulation*.
  2. Ecrivez une classe **Principale** dans laquelle vous définirez une cargaison, que vous remplirez de marchandises de votre choix, puis dont vous afficherez le coût.