

UE Conception Orientée Objet

Transport

Exercice 1 : Transport de marchandises

On souhaite modéliser en JAVA le calcul de coûts de transport de marchandises (*goods*) :

Goods
#weight : int
#volume : int
+Goods(weight : int, volume : int)
+getWeight() : int
+getVolume() : int

Les marchandises sont transportées sous la forme de cargaisons (*shipment*). Les fonctionnalités publiques des cargaisons utiles pour ce sujet sont :

**add** qui permet d’ajouter une marchandise dans cette cargaison si cela est encore possible, dans le cas contraire une exception `IllegalStateException` est levée.

**cost** qui retourne, sous la forme d’un nombre entier d’euros, le coût total du transport de cette cargaison.

**allGoods** qui fournit la liste des marchandises présentes dans la cargaison.

Une cargaison est également caractérisée par la distance sur laquelle elle est transportée. Ce renseignement est communiqué à la construction de la cargaison sous la forme d’un nombre entier de kilomètres. On précise qu’une cargaison ne peut réunir qu’une quantité limitée de marchandises (*quantity*) : cette quantité est exprimée soit en termes de poids (*weight*) soit de volume (*volume*), selon le type de transport utilisé. Ce dernier influe aussi sur le calcul du coût de transport de la cargaison qui, de la même façon, dépend de la quantité des marchandises de la cargaison. On distingue donc plusieurs types de cargaisons selon le moyen de transport utilisé :

Type de cargaison	Quantité	Coût	Limite
Fluviale ( <i>fluvial</i> )	poids	distance × √quantité	quantité ≤ 300000
Routière ( <i>road</i> )	poids	4 × distance × quantité	quantité ≤ 38000
Aérienne ( <i>air</i> )	volume	10 × distance + 4 × quantité	quantité ≤ 80000
Aérienne urgente ( <i>urgent</i> )	volume	2 × coût d’une cargaison aérienne	quantité ≤ 80000

- Q 1 . Identifiez les caractéristiques communes des types de transport.
- Q 2 . Proposez une hiérarchie de classes réunissant les caractéristiques communes et séparant les caractéristiques différentes.
- Q 3 . Une fois cette hiérarchie établie, donnez le code de la méthode `add` qui s’applique pour
  - les cargaisons fluviales,
  - les cargaisons aériennes.

A la lecture de ces codes, la séparation des caractéristiques communes et des caractéristiques différentes est-elle bien faite ?

Si ce n’est pas le cas faites un nouvelle proposition pour la hiérarchie précédente.

- Q 4 . On souhaite tester les spécifications suivantes valables pour tous les types de cargaison :
  1. après l’ajout réussi d’une marchandise, cette marchandise se trouve bien parmi les marchandises de la cargaison et la quantité de la cargaison a été correctement augmentée.
  2. l’exception `IllegalStateException` est déclenchée lorsque l’on essaie d’ajouter une marchandise qui excède la quantité disponible restante.
  3. le coût de la cargaison est valide après un ajout de marchandise réussi.
  4. le coût de la cargaison n’est pas modifié après un ajout impossible.

Proposez des classes de test qui permettent de vérifier que ces spécifications sont vérifiées par chacune des implémentations des différents types de cargaison.

Le cas échéant, les tests de ces spécifications pour un nouveau type de cargaison devraient être faciles à mettre en œuvre.

- Q 5 . (plus) Codez l’ensemble des classes et tests de ce sujet.