

---

## Exercice 1 – Gestion d’un convoyeur

Nous nous intéressons à un système de convoyage dans lequel un chariot est utilisé pour transporter des marchandises d’un point à un autre. Les marchandises ne sont pas identiques : leur poids est variable.

### Question 1

Définissez une classe `AleaObjet` permettant de construire des marchandises à transporter dont le poids est tiré aléatoirement entre deux bornes `min` et `max`. Les bornes peuvent varier d’une marchandise à l’autre. La classe doit permettre d’accéder au poids et d’afficher les caractéristiques d’une marchandise.

Les marchandises à transporter font partie d’un stock de taille limitée (en nombre de marchandises). On veut pouvoir remplir le stock, tester s’il est vide ou en extraire un élément.

### Question 2

Définissez une classe `AleaStock` permettant de manipuler un stock de taille éléments de type `AleaObjet`.

Le chariot qui transporte les marchandises a une capacité limitée en poids (`poidsMax`) et en nombre (`nbMax`) de marchandises. Il est alimenté par un chargeur dont le rôle est d’extraire les marchandises du stock et des empiler sur le chariot tant que celui-ci n’est pas plein (on n’a atteint ni le poids maximum, ni le nombre de marchandises maximum). Lorsqu’un chargement est plein, le chargeur doit attendre que le chariot soit à nouveau prêt à recevoir des marchandises (ce qui suppose qu’il a été déchargé, nous ne nous intéressons pas à cette opération pour l’instant).

Ces opérations sont répétées jusqu’à ce que le stock soit vide. Lorsque le stock est vide, le chargement est considéré comme terminé, même si la capacité maximum du chariot n’est pas atteinte.

### Question 3

Écrivez un constructeur pour la classe `Chargeur`, et la boucle d’opérations du chargeur. Déduisez-en un ensemble d’opérations à implémenter dans la classe `Chariot`.

### Question 4

Écrivez la classe `Chariot`. On utilisera un objet de type `ArrayList` pour stocker les éléments présents dans le chariot. Que peut-on utiliser pour réaliser l’opération d’attente ?

### Question 5

Construisez une classe permettant de tester l’exécution correcte d’un chargement : le chargeur remplit le chariot et se met en attente tant que le chariot n’a pas été vidé (donc le programme ne se termine pas).

Nous souhaitons maintenant compléter notre système en regardant les opérations de déchargement : une fois que le chariot a été rempli, le déchargeur récupère toutes les marchandises qu’il contient. Il signale alors que le chariot est vide et prêt pour un nouveau chargement. Il doit répéter cette opération pour chaque chargement effectué (on ne gère pas pour l’instant la terminaison).

### Question 6

Écrivez la classe `Dechargeur` et déduisez-en les opérations à ajouter à la classe `Chariot`.

### Question 7

Complétez la classe `Chariot` et le programme de test.

### Question 8

Proposez une solution pour gérer la terminaison du déchargeur et implémentez-la.