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 pouvant contenir `taille` marchandises. Nous nous plaçons dans le cas où les `taille` marchandises sont ajoutées au stock avant toute manipulation extérieure. À part le remplissage du stock, les autres opérations possibles sont un test permettant de savoir si le stock est vide et une opération d'extraction d'une unique marchandise.

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. Un ou plusieurs chargeurs transfèrent les marchandises du stock dans le chariot.

Un chargeur répète une séquence qui consiste à extraire une marchandise du stock et à la poser sur le chariot si celui-ci n'est pas plein (on n'a atteint ni le poids maximum, ni le nombre de marchandises maximum). Nous faisons l'hypothèse qu'aucune marchandise n'est plus lourde que la capacité du chariot. Si le chargeur ne peut pas déposer sa marchandise, il inidque que le chariot est plein et doit attendre que le chariot ait été vidé (nous ne modéliserons pas les déplacements) avant de poursuivre. Un chargeur termine son exécution lorsqu'il n'y a plus de marchandise dans le stock.

Le déchargement sera traité par un unique déchargeur. Celui-ci extrait un à un les objets du chariot jusqu'à ce qu'il ait été entièrement vidé, ce qui permet le démarrage d'un nouveau chargement.

Question 3
Quelles sont les variables d'instances nécessaires pour décrire le chariot ? Dans quel cas un chargeur peut-il se trouver bloqué ? Dans quel cas le déchargeur peut-il se trouver bloqué ? Déduisez-en les mécanismes de synchronisation à mettre en place.

Question 4
Écrivez un constructeur pour la classe `Chargeur`, et la boucle d'opérations du chargeur. Quelles sont les difficultés liées à la présence de plusieurs chargeurs ?

Question 5
Écrivez une première version de la classe `Chariot` contenant les variables et méthodes nécessaires pour un remplissage du chariot. Nous suggérons l'utilisation d'un objet de type `ArrayList` pour stocker les marchandises présentes dans le chariot.

Le déchargement consiste à extraire un à un les objets du chariot. Le déchargeur exécute une suite infinie d'opérations de déchargement.

Question 6
Écrivez la classe `Dechargeur` et complétez la classe `Chariot` pour permettre les opérations du déchargeur.

Question 7
Écrivez une classe principale permettant de tester un système avec deux chargeurs et un déchargeur.

Question 8

Proposez une solution pour traiter la terminaison du déchargeur, dans laquelle le code du déchargeur ne dépend pas du nombre d'instances de chargeurs.