TD 3

Quelques classes et interfaces de la bibliothèque des collections Java

On souhaite gérer des commandes de fournitures de bureau.

I. Les articles

Deux types d'articles seront proposés : les stylos et les ramettes de papier.
Un stylo est caractérisé par sa référence, son libellé, son prix unitaire et sa couleur.
Une ramette de papier est caractérisée par sa référence, son libellé, son prix unitaire et le grammage du papier (en g/m^2).

- Proposer une arborescence de classes pour représenter les fournitures.
- Ecrire le code de ces classes en respectant le principe d'encapsulation. Prévoir les accesseurs nécessaires. Redéfinir la méthode `public String toString()` permettant de consulter l'état de l'objet sous forme d'une chaîne de caractères.

II. Le catalogue

Un catalogue, édité pour une année, représente l'ensemble de tous les articles.

Ecrire la classe `Catalogue` caractérisée par une année et un ensemble d'articles que l'on veut pouvoir trier suivant leur référence. On ne doit pas avoir deux fois le même article dans le catalogue. On pourra utiliser l'interface `SortedSet<E>` du package `java.util`. La classe `TreeSet<E>` implémente cette interface. Vérifiez dans la javadoc les contraintes d'utilisation de cette interface et de cette classe.

A la création, le catalogue sera vide.

Fournir les méthodes permettant d'ajouter un article, de supprimer un article. Redéfinir la méthode `public String toString()` qui retournera une chaîne de caractère contenant l'année puis l'ensemble des articles triés suivant leur référence.

III. La commande

Une commande comporte un nom de client et une liste de lignes, c'est à dire de couples [article, quantité commandée].

Ecrire une classe `Commande` permattant de créer une commande pour un client puis d'y ajouter des lignes (article , quantité). On voudra calculer le montant total de la commande et afficher les caractéristiques de la commande.

Pour représenter les paires d'information de type clé, valeur, on pourra utiliser l'interface `Map<K,V>` du package `java.util`. La classe `HashMap<K,V>` implémente cette interface.

IV. Gestion des lots

Un lot comporte une référence et est composée d'articles. Il doit pouvoir faire l'objet d'une commande, de la même façon que tout autre article.

Ecrire la classe `Lot` permettant de gérér les lots. Est-il nécessaire de modifier les classes `Catalogue` et `Commande`?