

## TP3

---

### Classes et héritage

---

**Exercice 1 – Les articles.** On souhaite réaliser la maquette d'un outil permettant de passer des commandes pour des fournitures de bureau. Ces articles peuvent être de types divers, représentés sous la forme d'objets :

- Les articles sont de type stylo ou ramette de papier.
- Un stylo est décrit par son libellé, sa référence, son prix unitaire et sa couleur.
- Une ramette de papier est décrite par son libellé, sa référence, son prix unitaire et le grammage du papier (en grammes par mètre carré).

Proposer une arborescence de classes pour représenter les fournitures, sachant que de nouveaux types d'articles doivent pouvoir être ajoutés dans l'application sans avoir besoin de réécrire tout le code.

**Exercice 2 – Catalogue.** Un catalogue, édité pour une année, référence la liste de tous les articles. Ecrire une classe Catalogue en utilisant la classe ArrayList du package java.util pour gérer une liste d'objets.

**Exercice 3 – Commande.** Une commande comporte un nom de client et une liste de lignes (c'est-à-dire de couples [article, quantité commandée]). Ecrire la classe Commande permettant de passer une commande, éventuellement d'y ajouter des lignes, de calculer le montant total de la commande et d'afficher les caractéristiques d'une commande. Ecrire une classe Appli comportant une méthode main() permettant de tester la classe Commande.

**Exercice 4 – Lot.** Un lot comporte une référence et est composé d'un ensemble d'articles. Il doit pouvoir faire l'objet d'une commande, de la même façon que tout autre article. Ecrire une classe Lot permettant de gérer les lots. Est-il nécessaire de modifier les classes Catalogue et Commande ?

**Exercice 5 – Tri home made.** On souhaite pouvoir effectuer des tris sur les catalogues, les commandes ou tout autre collection. Ecrire une classe ComparableArticle contenant les méthodes plusGrand(...) et plusCher(...). Ecrire une classe TriComparable, possédant une méthode de tri permettant de trier une collection d'objets Comparable en implémentant un tri à bulles.

**Exercice 6 – Tri.** Utiliser la classe ComparableArticle dans la classe Article, afin de pouvoir effectuer des comparaisons d'articles selon l'ordre lexicographique de leurs références ou leurs prix. En utilisant la méthode tri() de la classe TriComparable, écrire des méthodes triParRef() et triParPrix() permettant de trier un Catalogue, d'une part dans l'ordre lexicographique des références des articles qu'il contient et, d'autre part, par ordre croissant des prix des articles. Tester vos méthodes de tri.

---

### Flux

---

**Exercice 7 – Enregistrer et lire dans des fichiers.** Ecrire des méthodes sauver() et restaurer() permettant de sauvegarder sur fichier puis de restaurer un Catalogue. Ecrire des méthodes sauver() et restaurer() permettant de sauvegarder sur fichier puis de restaurer une Commande.

---

### Exceptions

---

Ajouter dans les articles une attribut et les méthodes associées permettant de savoir si un objet est disponible en stock. Créer une exception IndisponibleException. Si on essaie d'ajouter à une commande un objet non disponible, on émet une exception.