

Programmation orientée objet

TP n° 9 : Collections

Livres

Un livre est caractérisé par son auteur, son titre, son prix et son code ISBN. On représentera le code ISBN par une chaîne de caractères. Deux livres seront égaux s'ils partagent le même code ISBN. Tous les champs sont constants.

Définissez une classe **Book** qui respecte ces spécifications.

Inventaire

On appelle inventaire un ensemble d'éléments associés à une quantité. La classe **Inventory** représente un inventaire. Les opérations associées à la classe sont :

- **get(e)** indique la quantité associée à l'élément **e**. Cette méthode n'échoue jamais.
- **getAll()** renvoie la liste des éléments présents dans l'inventaire (sans préciser les quantités associées). Un élément est présent dans l'inventaire si la quantité associée est non nulle.
- **add(e, n)** ajoute dans l'inventaire l'élément **e** en quantité **n**. Si **n** est négatif, on lance une exception **IllegalArgumentException**. Si l'élément était déjà présent, on augmente la quantité associée.
- **remove(e, n)** retire de l'inventaire l'élément **e** en quantité **n**. Si **n** est négatif, on lance une exception **IllegalArgumentException**. Si on tente de retirer plus d'éléments que l'inventaire n'en contient, une exception **InventoryException** (à définir) est lancée sans modification de l'inventaire.

1. Définissez la classe **Inventory**.
2. Une entrée de l'inventaire est un couple composé d'un élément et de la quantité qui lui est associée. Définissez une classe **Entry** qui représente ce couple et rendez la classe **Inventory** itérable sur ses entrées. Il est possible de réaliser la suite du sujet sans répondre à cette question.

Librairie

Une librairie dispose d'un stock de livres qu'elle met en vente. On la représente par la classe **BookStore**. Il est possible de réapprovisionner le stock d'une librairie à l'aide de la méthode **restock**. À l'inverse, à chaque fois que des livres sont vendus, on appelle la méthode **sell**. Ces deux méthodes prennent en paramètre un inventaire de livres. Il est également possible d'obtenir la quantité disponible d'un livre et l'ensemble des livres en stock à l'aide de méthodes **get** et **getAll**. Les spécifications sont les mêmes que pour la classe **Inventory**.

Implémentez la classe **BookStore**.

Tri

On souhaite pouvoir trier les livres selon différents critères : par auteur, titre ou prix. Dans chacun des cas, le tri peut être croissant ou décroissant.

Créez une énumération **Criteria** qui contient les valeurs **AUTHOR**, **TITLE** et **PRICE**. L'énumération comporte une méthode **getComparator** qui prend un booléen en paramètre. Elle renvoie un comparateur de livre associé au critère courant. Si le booléen est vrai, le comparateur compare les livres par ordre croissant, sinon par ordre décroissant.