

TP Java n°5 : Gestion d'un commerce

IUP I.I.E.S

8 Mars

J. Carme

Objectifs du TP : Pratique de l'héritage, utilisation de Javadoc

Exercice 1: Première version du magasin

On désire programmer une mini-application de gestion de petit commerce.

Pour cela, on va écrire une classe `Produit` et une classe `Magasin`.

Dans notre commerce, chaque produit est caractérisé par son nom, son prix d'achat, son prix de vente, le nombre d'exemplaires en stock, et sa description.

A la création du produit, on fixe son nom, son prix d'achat et son prix de vente, c'est-à-dire que ces données sont fournies en argument au constructeur de la classe. Par défaut, la description du produit est "Pas de description", et la quantité de produits en stock est nulle.

La classe `Produit` dispose d'un certain nombre de méthodes, qui lui permettent d'afficher la description du produit, de l'éditer, d'augmenter ou de diminuer le nombre d'exemplaires en stock, ainsi que d'obtenir les valeurs des différents attributs.

Question 1.1: Écrivez la classe `Produit`

Un magasin se caractérise par son solde et par son stock de produits. Le stock de produit est représenté par un tableau d'objets "Produit".

Avant de pouvoir acheter ou vendre un produit, il faut l'avoir ajouté dans le stock. Pour cela, la classe `Magasin` dispose d'une méthode:

```
public void ajouterProduit(String nom, float prixAchat, float prixVente)
```

L'indice de la case du tableau dans laquelle on a stocké le produit devient alors la référence de ce produit. Pour acheter ou vendre ce produit, on utilise alors la référence comme argument des méthodes:

```
public void acheterProduit(int referenceProduit, int nombreExemplaires)
public void vendreProduit(int referenceProduit, int nombreExemplaires)
```

La classe `Magasin` dispose également des méthodes habituelles d'accès à ses attributs, et d'une méthode `Bilan` qui permet d'afficher un bilan du magasin.

Question 1.2: Écrivez la classe `Magasin`. Testez-la.

Exercice 2: Différents types de produits

Jusque là, les produits disposent d'une description qui est une simple chaîne de caractère. Cela peut-être insuffisant dans certains cas.

Par exemple, on aimerait avoir une classe `Livre` qui ait le même comportement que la classe `Produit`, mais qui dispose d'un attribut `auteur` et d'un attribut `editeur`.

Question 2.1: Écrivez la classe `Livre` qui hérite de la classe `Produit`. Certaines méthodes doivent être surchargées (c'est-à-dire réécrites)

De même, un cd se caractérise par son auteur, son interprète, et l'ensemble des titres de ses pistes.

Question 2.2: Écrivez la classe Cd qui hérite de la classe Produit.

Il va maintenant falloir faire les modifications nécessaires dans la classe `Magasin`, mais elles sont minimales. En effet, `Livre` étant une sous-classe de `Produit`, les objets de type `Livre` sont également de type `Produit`, donc le tableau de produits peut également contenir des livres ou des cds.

C'est à la création des objets que les choses changent. En plus de la méthode `ajouterProduit`, la classe `Magasin` doit également contenir une méthode `ajouterLivre` et une méthode `ajouterCd`.

Question 2.3: Modifier la classe Magasin pour tenir compte des nouveaux objets.

Pour utiliser pleinement votre classe `Magasin`, vous avez maintenant besoin, comme dans le tp sur la banque, d'une méthode d'interaction permettant de proposer à l'utilisateur d'ajouter un livre, un cd ou un produit standard, d'acheter, de vendre, d'obtenir ou d'éditer la description d'un produit déjà existant, d'afficher un bilan général.

Question 2.4: Écrivez la méthode interaction de la classe Magasin

Exercice 3: Recherche par mot

Dans l'exercice précédent, pour acheter, vendre, éditer ou afficher la description d'un produit existant, il fallait le spécifier par sa référence, c'est-à-dire par son indice dans le tableau. Ce n'est pas un moyen très pratique d'aller chercher les produits, surtout si il y en a un très grand nombre.

Nous allons écrire la méthode

```
public Produit rechercherProduit(String nom)
```

qui renvoie un objet `Produit` dont le nom est spécifié en argument, ainsi que la méthode

```
public Produit rechercherProduitParMot(String mot)
```

qui renvoie un objet `Produit` dont la description contient le mot spécifié en argument.

Pour écrire ces méthodes, vous allez avoir besoin d'un outil capable de comparer deux chaînes de caractères, et d'un outil capable de dire si une chaîne est incluse dans une autre. C'est la classe `String` elle-même qui est capable de faire cela. En effet, contrairement à ce que vous avez vu jusque là, `String` n'est pas un type simple comme `int`, mais une classe à part entière qui a ses attributs et ses méthodes.

Comme toute classe interne à Java, elle est précisément décrite dans la documentation de Java. La documentation de Java est normalement disponible sur votre machine, et elle est également disponible à l'adresse suivante: <http://java.sun.com/j2se/1.4.2/docs/api/overview-summary.html>.

Attention! La documentation de Java semble complexe et déroutante au premier abord. Pour obtenir les informations qui vous intéressent, allez dans l'index et cherchez la classe `String`. Vous pouvez également vous balader dans le reste de la doc pour vous familiariser avec elle.

Question 3.1: Écrivez les méthodes rechercherProduit et rechercherProduitParMot. Modifiez votre méthode interaction pour tenir compte de ces nouvelles méthodes.