

TP3 Entrées/sorties

Développement Logiciel (L2-S4)

Vendredi 8 février 2013

Ce TP prend sa place (encore une fois) dans le stock de livres dans une bibliothèque. Il touche plus particulièrement l'ajout de nouveaux livres et la recherche de livres en utilisant une base de connaissances référençant les différents ouvrages possédés par la bibliothèque. Un livre possède un auteur, une maison d'imprimerie et un titre. Il est également utile de connaître pour chaque oeuvre le nombre d'exemplaires disponibles et possédés par la bibliothèque. Deux types d'ouvrages existent : les bandes dessinées et les romans. Les bandes dessinées possèdent également un dessinateur (exemple Uderzo pour astérix). Les romans ont un genre (policier, science-fiction, fantasy,...). Quatre grands genres sont utilisés dans la représentation : Amour, Aventure, Policier et Biographique. Dans la suite les genres seront représentés par un type énuméré.

Exercice 1 : Création des classes

1. Écrire les classes **Roman**, **BandeDessinees** et **Livre** ainsi que leurs constructeurs et accesseurs et les méthodes `toString`.
2. Écrire une méthode permettant d'incrémenter ou de décrémenter le nombre d'exemplaires disponibles d'un livre donné.

Exercice 2 : Lecture de la base de données Les oeuvres sont rangées dans le fichier `oeuvres.txt`. Ouvrez ce fichier. Dans cette base, les bandes dessinées sont sous la forme " BD\t titre\t Auteur\t Dessinateur\t Éditeur\t nombre libre\t nombre possédés ". Les romans sont sous la forme "Roman\t genre\t Titre\t Auteur\t Éditeur\t nombre libre\t nombre possédés ".

1. Écrire une méthode `void litfichier()` permettant d'afficher le contenu du fichier `oeuvres.txt`. Pour ceci vous pouvez utiliser la classe **FileReader** qui permet la lecture de fichier.
2. Créer une méthode `Livre creelivre(String ligne)` qui transforme une des lignes lues précédemment en un livre (une bande dessinée ou un roman). Pour ceci vous pouvez utiliser la fonction `split` de la classe **String** qui permet de séparer une String en plusieurs éléments suivant un motif.
3. Créer une méthode `ArrayList<Livre> parcourbase()` qui lit les lignes de `oeuvres.txt` et qui pour chaque ligne crée le livre associé et l'ajoute à l'ensemble donné en retour.
4. Affichez les résultats et vérifiez que c'est bien correct.

Exercice 3 : Création de livres Il est aussi demandé de pouvoir ajouter de nouveaux livres. Pour ceci le bibliothécaire utilise une interface texte dans laquelle le système lui pose des questions auxquels il répond. Par exemple, le système demande "Quel est le titre ?" et le bibliothécaire répond "moby dick".

1. Écrire la fonction `static Livre dialogue()` permettant un dialogue entre l'homme et la machine afin de créer un livre. Pour ceci vous pouvez utiliser **System.in** qui est un **InputStream**.
2. Écrire une fonction `ArrayList<Livre> ajoutelivre(ArrayList<Livre> leslivres)` créant un livre de cette manière et permettant de l'ajouter à l'ensemble des livres obtenus par la fonction `parcourbase()`. Notons qu'il se peut que le bibliothécaire rajoute des livres déjà possédés. Dans ce cas seul le nombre d'exemplaires de cet ouvrage doit changer.

3. Modifier votre fonction pour qu'elle puisse boucler si le bibliothécaire veut rajouter d'autres ouvrages. Cela est fait en posant la question " Voulez vous rajouter d'autres oeuvres ".

Exercice 4 : Mise en mémoire Après avoir obtenu les nouveaux livres, l'étape suivante consiste à les ranger. Pour ceci il faut mettre toutes les oeuvres de l'ArrayList obtenue dans le fichier oeuvre.txt.

1. Créer les fonctions *String BDtobase()* et *String Romantobase()* permettant de transformer les BD et roman dans le format de la base de connaissance.
2. Créer la fonction *void ajouteDansBase(ArrayList<Livre> leslivres)* permettant de mettre les romans et bandesdessinées dans le fichier en utilisant les fonctions de transformation précédentes et **BufferedWriter**.
3. Créer la fonction *void principaleajout()* qui appelle les fonctions des différents exercices afin d'ajouter des ouvrages dans la base. Elle commence par appeler *parcourebase* pour avoir les livres contenus dans la base puis *ajoutelivre* pour créer les livres et enfin *ajouteDansBase* pour tout mettre dans la base de connaissance.
4. Tester votre travail.

Exercice 5 : Aller plus loin Maintenant il faut de créer une méthode permettant de chercher certains livres dans la base en utilisant un système de dialogue. Un tel système peut commencer par exemple les questions " Quel type de livre souhaitez vous (BD ou roman) " (roman), " Quel critère voulez vous utiliser (auteur, type, titre) ? " (type), " Quel type de roman souhaitez vous ? " (roman) Vous êtes libre de la manière d'organiser ce dialogue mais à la fin le système doit afficher les livres intéressants ou " Je suis désolé aucun livre ne correspond à votre demande. ". Notons toutefois que les critères peuvent varier et par exemple être " les livres disponibles " ou " le titre contient astérix ".

1. Créer la méthode *ArrayList<Livre> filtre(ArrayList<Livre> leslivres, String critere, String valeur)* parcourant leslivres afin de n'avoir en sortie que ceux correspondant bien au critères. Par exemples si le critère est "auteur" et la valeur "Victor Hugo" renvoie tous les livres de victor hugo
2. Créer la fonction *void recherche()*. Elle commence par appeler *parcourebase()* afin d'avoir l'ensemble des listes connus par la bibliothèque. Puis à partir de cette liste demande à l'utilisateur son critère et sa valeur et à partir de ceux ci filtre l'ensemble des livres. Ces étapes sont faites jusqu'à ce que le client soit satisfait.