

Programmation Java - Master TTT

10 février 2011 - 2 heures

Les documents hors livres sont autorisés.

On prendra soin de bien préciser les visibilité des membres des classes, et de justifier les réponses.

Exercice n° 1

On veut gérer une liste d'ouvrages scientifiques. Ces ouvrages sont de deux types : les livres et les revues. Un livre est caractérisé par son titre et ses auteurs, alors qu'une revue est caractérisée par son titre et son numéro. De plus, chaque ouvrage a un identifiant (ISBN) codé par un entier long.

1. Ecrire une interface `Ouvrage` qui déclare une méthode `getTitre()` et une méthode `getISBN()` qui retourneront respectivement le titre et l'identifiant de l'ouvrage.
2. Ecrire une classe `Revue` qui implémente `Ouvrage` et telle que :
 - chaque objet `Revue` représentera le numéro d'une revue ;
 - le constructeur de la classe `Revue` a trois arguments, une chaîne de caractères (le titre), un entier (le numéro) et un entier long (l'identifiant) ;
 - la classe `Revue` a une méthode `getNumero()` qui retourne un entier indiquant le numéro de la revue ;
 - si on affiche un objet `Revue` avec `System.out.println`, on obtient son titre, suivi de son numéro ;
 - la classe `Revue` a une méthode statique `createRevue` sans argument qui demande à l'utilisateur d'entrer un titre, un numéro et un ISBN et qui retourne un objet `Revue` correspondant à ces informations.
3. On suppose qu'il existe de même une classe `Livre` qui implémente `Ouvrage` et qu'on ne demande pas d'écrire. Pensez-vous qu'il soit possible de mettre en commun les méthodes de la classe `Revue` et de la classe `Livre` qui sont identiques ? Comment faire ? Quelles sont ces méthodes ?
4. La liste d'ouvrages est sauvegardée dans un fichier texte dans lequel chaque ouvrage est décrit de la manière suivante :
 - une première ligne contient soit *L* s'il s'agit d'un livre, soit *R* s'il s'agit d'une revue ;
 - la deuxième ligne contient l'identifiant de l'ouvrage,
 - la troisième contient son titre,
 - la ou les ligne(s) suivante(s) contiennent soit le numéro de la revue, soit la liste des auteurs (un auteur par ligne) ;
 - la dernière ligne, avant de passer à l'ouvrage suivant contient uniquement le symbole #.

On suppose que la classe `Livre` possède un constructeur dont les arguments sont une chaîne de caractères (le titre), une liste `List<String>` contenant le nom des auteurs et un entier long (l'identifiant). Ecrire une méthode statique `chargeOuvrages` qui prend en argument le nom d'un fichier et retourne la liste des ouvrages décrits dans ce fichier. Si le fichier n'est pas au bon format, cette méthode lancera une exception `FichierOuvrageException`.

5. Définir le type d'exception `FichierOuvrageException`.

Exercice n° 2

1. On suppose qu'on veut écrire une classe `TriListes` qui contient des méthodes pour manipuler des listes d'entiers. Ecrire une méthode `estTrie` qui prend en argument une liste d'entiers (`List<Integer>` définie dans le package `java.util`), et qui retourne vrai si les éléments de la liste sont triés par ordre croissant (on pourra avoir deux fois le même élément dans la liste).
2. La méthode `estTrie` peut-elle être statique ?
3. Ecrire une méthode `fusion` qui prend en argument deux listes triées et retourne une liste (`LinkedList<Integer>`) contenant les éléments triés des deux listes :
 $\text{fusion}([1; 5; 6; 9; 12], [2; 7; 9; 11]) \mapsto [1; 2; 5; 6; 7; 9; 9; 11; 12]$.
4. Ecrire une méthode `split` qui prend en argument trois listes : une liste l quelconque, et deux listes r_1 et r_2 vides. La méthode `split` remplit la liste r_1 avec les éléments de l placés en position paire (0, 2, *etc.*) et r_2 avec les éléments de l placés en position impaire (1, 3, *etc.*) :
 $\text{split}(l = [4; 2; 9; 1; 4; 7; 3; 8], r_1 = [], r_2 = []) \mapsto r_1 = [4; 9; 4; 3], r_2 = [2; 1; 7; 8]$.
5. Pour trier une liste, on considère l'algorithme suivant :
 - si la liste est de longueur $\ell \leq 1$, elle est triée ;
 - sinon, on sépare la liste en deux listes (`split`), on trie récursivement chacune des deux listes, puis on les fusionne.Ecrire la méthode `tri` qui prend une liste en argument et retourne la liste triée qui contient les mêmes valeurs :
 $\text{tri}([4; 2; 9; 1; 4; 7; 3; 8]) \mapsto [1; 2; 3; 4; 4; 7; 8; 9]$.
6. Maintenant, on veut que la classe `TriListes` étende la classe `LinkedList<Integer>`.
 - a) Ecrire l'entête de la classe `TriListes`.
 - b) On veut modifier la méthode `estTrie` de façon à ce qu'on puisse écrire `l.estTrie()`, si `l` est un objet de type `TriListes`. Expliquer ce qu'il faut faire comme modifications par rapport à la version de `estTrie` du 1.