

Programmation Orientée Objet : JAVA

TP5

Héritage et polymorphisme

Pour la gestion d'une bibliothèque on nous demande d'écrire une application traitant des *documents* de nature diverse: des *livres*, des *revues*, des *dictionnaires*, etc. Les livres, à leur tour, peuvent être des *romans* ou des *manuels*.

Tous les documents ont un numéro d'enregistrement (un entier) et un titre (une chaîne de caractères). Les livres ont, en plus, un auteur (une chaîne) et un nombre de pages (un entier). Les romans ont éventuellement un prix littéraire (une chaîne de caractères), tandis que les manuels ont un niveau scolaire (un entier). Les revues ont un mois et une année (des entiers) et les dictionnaires ont une langue (une chaîne de caractères convenue, comme "anglais", "allemand", "espagnol", etc.). Tous les objets en question ici (*livres*, *revues*, *dictionnaires*, *romans* et *manuels*) doivent pouvoir être manipulés en tant que *documents*.

- A. Définissez les classes **Document**, **Livre**, **Roman**, **Manuel**, **Revue** et **Dictionnaire**, entre lesquelles existeront les liens d'héritage que la description précédente mentionne. Les classes appartenant à un paquetage *tp5.docs*.

Dans chacune de ces classes définissez :

- le constructeur qui prend autant d'arguments qu'il y a de variables d'instance dans les objets et qui se limite à initialiser ces dernières avec les valeurs des arguments,
- une méthode **public String toString()** retournant une description des objets sous forme de chaîne de caractères,
- des accesseurs (getters) permettant de consulter les valeurs des variables d'instance.

Écrivez une classe principale **TestDocuments** qui crée et affiche plusieurs documents de types différents.

- B. Une bibliothèque est une collection de documents. Elle peut être réalisée par un objet *ArrayList* (du package *java.util*) dont les éléments sont des *documents*. Définissez une classe **Bibliotheque** (appartenant à un paquetage *tp5.biblio*), avec une telle structure de données comme variable d'instance *privée* et les méthodes suivantes:

- **Bibliotheque()** - constructeur qui instancie la collection des documents.
- **int getNbrDocs()** - renvoie le nombre de documents dans la bibliothèque.
- **Document document(int i)** - renvoie le *i*^{ème} document.
- **boolean ajouter(Document doc)** - ajoute le document indiqué et renvoie **true** (ou renvoie **false** en cas d'échec).
- **boolean supprimer(Document doc)** - supprime le document indiqué et renvoie **true** (ou renvoie **false** en cas d'échec).
- **void afficherDocuments()** - affiche *tous* les ouvrages de la bibliothèque.
- **void afficherAuteurs()** - affiche la liste des auteurs de tous les ouvrages qui ont un auteur (au besoin, utilisez l'opérateur **instanceof**).

Ecrivez une classe principale **TestBibliotheque** qui crée une bibliothèque, y ajoute des documents de types différents puis teste les autres méthodes.