TP5 : Héritage et Polymorphisme

Enseignants : Madhi Kamel, Majdoub Manel

Groupes : Prepa A02
Objectif : utiliser l'héritage et le polymorphisme

Exercice 1

1. On considère la classe Etudiant définie par :
   - les attributs privés suivants : nom, prénom, nombre de notes et un tableau de notes
   - les méthodes suivantes : Saisie, Affichage, Moyenne, Admis, Exaequo (Moyenne retourne la moyenne des notes, Admis et Exaequo retournent un résultat booléen, sachant que Exaequo compare la moyenne avec celle d'un autre étudiant).

   Définir la classe Etudiant en java.
2. On considère la classe EtudiantTerminal qui représente les étudiants en terminale
   - un étudiant de ce type possède un attribut supplémentaire par rapport aux autres étudiants : note de mémoire.
   - Les méthodes associées à cette classe sont : Saisie, Affichage, Moyenne, Admis, Exaequo.

   Définir la classe EtudiantTerminal en java.
3. On considère la classe TableEtudiants représentant un tableau d'étudiants pouvant contenir différents types d'étudiants (en maîtrise ou non). Les méthodes implémentées par cette classe sont : Ajout d'un étudiant donné en paramètre, Suppression d'un étudiant dont le rang est donné en paramètre, Nombre d'étudiants admis, Affichage de tous les étudiants du tableau.

   Définir la classe TableEtudiants en java.
4. Ecrivez un programme java permettant de créer un tableau d'étudiants et permettant de :
   - ajouter des étudiants dans le tableau
   - supprimer un étudiant
   - calculer le nombre d'étudiants admis
   - afficher la totalité des étudiants

Exercice 2

On modélise une application devant servir à l'inventaire d'une bibliothèque. Elle devra traiter des documents de nature diverse : des livres, des dictionnaires, et autres types de documents qu'on ne connaît pas encore précisément mais qu'il faudra certainement ajouter un jour (articles, bandes dessinées...).

Tous les documents possèdent un numéro d'enregistrement et un titre. A chaque livre est associé, en plus, un auteur et un nombre de pages, les dictionnaires ont, eux, pour attributs supplémentaires une langue et un nombre de tomes.

On veut manipuler tous les articles de la bibliothèque au travers de la même représentation : celle d'un document.

1. Définir les classes Document, Livre et Dictionnaire. Définir pour chacune un constructeur permettant d'initialiser toutes ses variables d'instances.
2. Définir une classe Bibliothèque réduite à une méthode main permettant de tester les classes précédentes (ainsi que les suivantes).
3. Définir la classe ListeDeDocuments permettant de créer une liste vide de documents, puis y adjoindre une fonction permettant d'ajouter un document. Utiliser la structure Vecteur (java.util.Vector).
4. Dans la classe ListeDeDocuments Définir une méthode tousLesAuteurs() qui affiche la liste des numéros des documents de la liste avec, pour chacun, l'éventuel auteur.
5. Redéfinir la méthode toString() dans la classe Document ainsi que dans les classes Livre et Dictionnaire et qui renvoie une chaîne de caractères décrivant un document, un livre ou un dictionnaire. Ajoutez alors dans la classe ListeDeDocuments une méthode tousLesDocuments() qui affiche consécutivement la description de tous les documents.
6. Tester la classe ListeDeDocuments.