Module M2103 – Java
TP 6 : Héritage et Polymorphisme

On désire désormais distinguer différents types de documents dans la bibliothèque, par exemple des livres, CDs, documents électroniques... Aussi, les différents membres de la bibliothèque ne peuvent tous emprunter le même nombre de documents. Dans le cadre de ce TP, nous développerons des classes spécifiques pour les différents types de documents et de membres de la bibliothèque.

Partie I

- **Tâche 1** – Réorganiser le programme développé de manière à avoir une classe `Livre` qui est un type spécifique de `DocBibliotheque`. Quelques attributs et méthodes resteront dans la classe `DocBibliotheque` tandis que d'autres devront être transférés dans la classe `Livre`. De plus, un objet `Livre` devra maintenir de l'information à propos du nom de son éditeur, son nombre de pages et ISBN. Vous écrirez une méthode toString() pour pouvoir afficher de l'information à propos d'objets `Livre` en les désignant dans des instructions *println*. Mettre à jour la classe test de manière à pouvoir instancier des objets `Livre` à la place d'objets `DocBibliotheque`.
- **Tâche 2** – Créer une nouvelle classe permettant de gérer un nouveau type de document de la bibliothèque appellée `CD`. Un CD est caractérisé par les informations suivantes : code d'archivage, titre, artiste (équivalent à l'auteur d'un livre), liste de morceaux. On supposera que cette information est fournie à l'instanciation.
- **Tâche 3** – Créer une autre classe, nommée `DocURL`, représentant un document électronique pouvant être catalogué. L'information relative à un document électronique est son code d'archivage, son URL (e.g. http://www.siteweb.com/), une courte description de son contenu, l'auteur ou l'entreprise ayant publié le document (si connu). La classe `DocURL` est une classe fille de la classe `DocBibliotheque`. Etant donné qu'un document électronique n'est pas un objet physique, il ne peut être emprunté, retourné ou réservé.
- **Tâche 4** – Faire en sorte que la classe CatalogueBibliotheque puisse stocker des objets `Livre`, `CD` et `DocURL`. Tester la classe avec les trois types de documents.
- **Tâche 5** – Ajouter les méthodes suivantes à la classe CatalogueBibliotheque :

  `compteLivres()` – fournit le nombre total de livres dans la bibliothèque.

  `compteCDs()` – fournit le nombre total de CDs dans la bibliothèque.

Partie II

Il existe en fait deux types de membres de la bibliothèque : Etudiant et Personnel. Les étudiants peuvent emprunter quatre livres au maximum alors que les membres du personnel peuvent en emprunter huit. Le but de cette partie est l'écriture de code qui permet de respecter cette contrainte.

- **Tâche 1** – Ajouter à la classe `MembreBibliotheque` un attribut permettant de déterminer le nombre de documents empruntés, deux méthodes pour incrémenter et décrémenter ce nombre ainsi qu'une méthode permettant de déterminer le nombre de documents empruntés par le membre.
- **Tâche 2** – Modifier l'implémentation des méthodes d'emprunt et de retour de documents dans la classe `CatalogueBibliotheque` de manière à ce que les méthodes de la tâche 1 soient appelées de manière appropriée. Par exemple, lorsque la méthode d'emprunt est appelée et que l'emprunt du document par le membre est validé, le nombre de documents empruntés par le membre doit être incrémenté.
- **Tâche 3** – Ecrire une méthode nommée *peutEmprunterAutreDocument* dans la classe `MembreBibliotheque` qui, pour l'instant, retournera le booléen True.
  Modifier la méthode d'emprunt de document dans la classe `CatalogueBibliotheque` de manière à ce que la méthode *peutEmprunterAutreDocument* soit appelée avant de décider si l'emprunt est réalisable. En d'autres termes, la méthode d'emprunt de la classe `CatalogueBibliotheque` vérifie désormais que le document est disponible pour l'emprunt et qu'il est permis au membre d'emprunter plus de documents.
- **Tâche 4** – Comme indiqué en préambule, les étudiants et membres du personnel peuvent emprunter un nombre different de documents. Pour respecter cette contrainte, vous utiliserez le polymorphisme.
  Développer deux nouvelles classes, nommées `MembreEtudiant` et `MembrePersonnel`, basées sur la classe `MembreBibliotheque`.
- **Tâche 5** – Modifier la classe de test de manière à instancier des objets `MembreEtudiant` et `MembrePersonnel` au lieu d'objets `MembreBibliotheque`.