Master bio-info : Java

Année 2008-2009

TD5

Exercice 1 Pour la gestion d'une bibliothèque, on vous demande d'écrire une application traitant des documents de nature diverse : des livres, qui peuvent être des romans ou des manuels, des revues, des dictionnaires, etc.

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, par exemple : Goncourt, Médicis, Femina, etc.), tandis que les manuels, en plus d'un auteur, 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, etc.) doivent pouvoir être manipulés en tant que documents.

- Proposez une hiérarchie pour les classes `Document`, `Livre`, `Roman`, `Manuel`, `Revue`, `Dictionnaire`. Dites dans quelle classe vous déclarez quel attribut.
- Écrivez un constructeur (ou plusieurs, si cela vous semble nécessaire) pour les classes `Document`, `Livre` et `Roman`. Comment faire pour être assuré que chaque document a un numéro d'enregistrement différent ?
Pour les autres classes, donnez juste le prototype du constructeur (la suite de ses arguments).
- Écrivez pour les classes `Document`, `Livre` et `Roman` une méthode public `String toString()` produisant une description sous forme de chaîne de caractères des objets.

La classe `Bibliothèque` Une bibliothèque sera représentée par un tableau de documents. Définissez une classe `Bibliothèque`, avec un tel tableau comme attribut et les méthodes et constructeur : Certaines cases du tableau peuvent être égales à `null`.

- `Bibliothèque(int capacité)` : constructeur qui crée une bibliothèque ayant la capacité (nombre maximum de documents) indiquée,
- `Bibliothèque()` : constructeur qui crée une bibliothèque ayant une capacité de 100 documents;
- `void afficherDocuments()` : affiche tous les ouvrages de la bibliothèque,
- `Document document(int i)` : renvoie le ième document,
- `boolean ajouter(Document doc)` : ajoute le document indiqué et renvoie `true` (`false` en cas d'échec),
- `boolean supprimer(Document doc)` : supprime le document indiqué et renvoie `true` (`false` en cas d'échec)
- `void afficherAuteurs()` : affiche la liste des auteurs de tous les ouvrages qui ont un auteur (au besoin, utilisez l'opérateur `instanceof`)

Exercice 2 On a les classes suivantes : (On remarquera qu'il serait plus logique de déclarer toutes ces méthodes de manière statique)

```
public class Automobiliste{
    public int vitesseEnVille(){
        return 50;
    }
}

public class AutomobilisteEcolo extends Automobiliste{
    public int vitesseEnVille(){
        return 40;
    }

    public String typeAuto(){
        return "hybride";
    }
}

public class AutomobilisteDelinquant extends Automobiliste{
    public int vitesseEnVille(){
        return 65;
    }

    public String typeAuto(){
        return "essence";
    }
}

public class Chauffard extends AutomobilisteDelinquant {
    public int vitesseEnVille(){
        return 90;
    }

    public String typeAuto(){
        return "4x4";
    }
}

public class AutomobilisteEcoloMilitant extends AutomobilisteEcolo {
    public String typeAuto(){
        return "electrique";
    }
}
```

Question 1 On a les affectations (valides) suivantes

```
AutomobilisteEcolo a1 = new AutomobilisteEcolo();
AutomobilisteEcolo a2 = new AutomobilisteEcoloMilitant();
Automobiliste a3 = new Chauffard();
```

Dites pour chacun des numéros de ligne suivant, si elle compile, si elle fonctionne à l'exécution, et ce qui est imprimé.

```
1:      System.out.println(a1.vitesseEnVille());
```

```
2:     System.out.println(a2.vitesseEnVille());
3:     System.out.println(a3.vitesseEnVille());
```

```
4:     System.out.println(a1.typeAuto());
5:     System.out.println(a2.typeAuto());
6:     System.out.println(a3.typeAuto());
```

Question 2 Dites pour chacun des numéros de ligne suivant, si elle compile et si elle fonctionne à l'exécution.

```
AutomobilisteDelinquant a4 = a3;
AutomobilisteEcoloMilitant a5 = a2;
a2 = a1;
a1 = (Automobiliste) a3;
```