

## TD n°6 - Correction

### Héritage

**Exercice 1** 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*. À 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. Définissez les classes `Document`, `Livre` et `Dictionnaire`. Définissez pour chacune un constructeur permettant d'initialiser toutes ses variables d'instances.

**Correction :**

```
public class Document {
    static private int numeroSuivant = 0;
    static private int getID() { return ++numeroSuivant; }
    private int numero;
    private String titre;

    public Document(String titre) {
        this.numero = getID();
        this.titre = titre;
    }
    public int getNumero(){
        return numero;
    }
    public String getTitre(){
        return titre;
    }
}
public class Dictionnaire extends Document {
    private int nbTomes;
    private String langue;

    public Dictionnaire(String titre, int nbTomes, String langue) {
        super(titre);
        this.nbTomes = nbTomes;
        this.langue = langue;
    }
    public int getNombreDeTomes() { return nbTomes; }
    public String getLangue() { return langue; }
}
public class Livre extends Document{
    private String auteur;
    private int nbPages;
```

```

    public Livre(String titre, String auteur, int nbPages) {
        super(titre);
        this.nbPages = nbPages;
        this.auteur = auteur;
    }
    public String getAuteur() { return auteur; }
    public int getNombreDePages() { return nbPages; }
}

```

2. Définissez une classe Bibliothèque réduite à une méthode main permettant de tester les classes précédentes (ainsi que les suivantes).

**Correction :** Par exemple :

```

public class Bibliotheque{
    public static void main(String[] argv){
        Document d = new Document("Configurer son compte utilisateur");
        Livre l = new Livre("La communication sous unix","J-M Rifflet",799);
        Dictionnaire di = new Dictionnaire("Man pages",4,"Anglais");
        System.out.println(d.getTitre() );
        System.out.println(l.getTitre() );
        System.out.println(di.getTitre() );
    }
}

```

3. Définissez la classe ListeDeDocuments permettant de créer une liste vide de documents, puis y adjoindre une fonction permettant d'ajouter un document.

**Correction :**

```

import java.util.Enumeration; // utile pour l'ajout de la question suivante
import java.util.Vector;
public class ListeDeDocuments {
    private Vector lesDocuments;
    public ListeDeDocuments() {
        lesDocuments = new Vector();
    }
    public void ajouteDocument(Document unDocument) {
        lesDocuments.add(unDocument);
    }
}

```

4. Dans la classe ListeDeDocuments définissez une méthode tousLesAuteurs() qui affiche la liste des numéros des documents de la liste avec, pour chacun, l'éventuel auteur.

**Correction :**

```

    public void tousLesAuteurs() {
        Enumeration e = lesDocuments.elements();
        Document unDocument;
        while( e.hasMoreElements() ) {
            unDocument = (Document)e.nextElement();
            System.out.print(unDocument.getNumero());
            if (unDocument instanceof Livre)
                System.out.println( " " + ((Livre)unDocument).getAuteur());
            else
                System.out.println() ;
        }
    }
}

```

5. Redéfinissez 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.

**Correction :** Dans `Document` :

```
public String toString() {
    return ("numero: "+numero+" titre: "+titre);
}
```

Dans `Livre` :

```
public String toString(){
    return (super.toString()+" auteur: "+getAuteur()+" pages: "+getNombreDePages());
}
```

Dans `Dictionnaire` :

```
public String toString() {
    return (super.toString()+" tomes: "+getNombreDeTomes()+" langue: "+getLangue());
}
```

Dans `ListeDeDocuments` :

```
public void tousLesDocuments() {
    Enumeration e = lesDocuments.elements();
    Document unDocument;
    while (e.hasMoreElements()){
        unDocument = (Document)e.nextElement();
        System.out.println(unDocument);
    }
}
```

6. Proposez quelques lignes de codes à ajouter à la classe `Bibliotheque` afin de tester la classe `ListeDeDocuments`.

**Correction :**

```
public class Bibliotheque{
    public static void main(String[] argv) {
        Document d = new Document("Configurer son compte utilisateur");
        Livre l = new Livre("La communication sous unix","J-M Rifflet" ,799);
        Dictionnaire di = new Dictionnaire("Man pages",4,"Anglais");
        System.out.println(d.getTitre());
        System.out.println(l.getTitre());
        System.out.println(di.getTitre());

        ListeDeDocuments ld = new ListeDeDocuments();
        ld.ajouteDocument(d);
        ld.ajouteDocument(l);
        ld.ajouteDocument(di);
        ld.tousLesAuteurs();
        ld.tousLesDocuments();
    }
}
```

**Exercice 2** Écrire trois classes `Figure`, `Carre`, et `Rectangle`, telles que :

1. `Figure` a des attributs abscisse et ordonnée, ainsi qu'une couleur (encodée par un entier).

2. Carre et Rectangle héritent de Figure, mais ont en plus une ou deux longueur pour les côtés.
3. Figure a un attribut privé Vector référençant **toutes les instances** de sa classe et de ses sous classes.
4. Figure a une méthode statique getInstances() renvoyant ce vecteur.
5. Carre et Rectangle redéfinissent cette méthode getInstances() de manière à ne récupérer que les instances qui correspondent à leur type.

**Correction :** On déduira aisément une solution pour la classe Rectangle d'une solution pour la classe Carre :

```
//Fichier Figure.java
import java.util.Vector ;
public class Figure {
    private static Vector instances = new Vector();
    private int abscisse;
    private int ordonnee;
    private int couleur;

    public Figure( int abscisse , int ordonnee , int couleur ) {
        this.abscisse = abscisse;
        this.ordonnee = ordonnee;
        this.couleur = couleur;
        instances.add(this);
    }
    public static Vector getInstances() { return instances ; }
    public String toString(){
        return (abscisse+" "+ordonnee+" "+couleur);
    }
}

public class Carre extends Figure {
    private int cote ;
    public Carre(int abscisse, int ordonnee, int couleur, int cote) {
        super(abscisse,ordonnee,couleur);
        this.cote = cote;
    }
    public static Vector getInstances() {
        int nCarre = 0;
        Vector instancesCarre = new Vector();
        Vector instances = Figure.getInstances();
        Enumeration e = instances.elements();
        Figure uneFigure;
        while(e.hasMoreElements()) {
            uneFigure = (Figure)e.nextElement();
            if (uneFigure instanceof Carre)
                instancesCarre.add(uneFigure);
        }
        return instancesCarre;
    }
    public String toString(){
        return (super.toString()+" "+cote);
    }
}
}
```

```

//Fichier Test.java
import java.util.Vector ;
import java.util.Enumeration ;
public class Test {
    public static void main(String[] argv) {
        Figure f1 = new Figure(1,1,1);
        Figure f2 = new Figure(2,2,2);
        Carre c3 = new Carre(3,3,3,3);
        Figure f4 = new Figure(4,4,4);
        Carre c5 = new Carre(5,5,5,5);

        System.out.println("Liste des figures" ) ;
        Enumeration e = Figure.getInstances().elements() ;
        while(e.hasMoreElements()){
            System.out.println(e.nextElement() ) ;
        }
        System.out.println("Liste des carres" ) ;
        e = Carre.getInstances().elements() ;
        while(e.hasMoreElements()){
            System.out.println(e.nextElement() ) ;
        }
    }
}

```