Travaux Dirigés
Mise à niveau en JAVA
héritage, polymorphisme
–M1–

Exercice 1: Analyse d'un programme Java

Considérons le programme Java suivant :

```
package Ex1;
public class Livre {
        protected String titre, auteur, proprietaire ;
        protected int nb_page ;
        double prix ;

        public Livre(String t, String a, double p, int nb){
         titre = t ;
         auteur = a ;
         prix = p ;
         proprietaire = "" ;
         nb_page = nb ;
         }
        public void Afficher() {
         System.out.println("Titre : " + titre) ;
         System.out.println("Auteur : " + auteur) ;
         System.out.println("Prix : " + prix) ;
         System.out.println("Nombre de pages : " + nb_page);
         if ( this.Est_neuf() ) {
         System.out.println("Aucune proprietaire" ) ;
         } else {
         System.out.println("Proprietaire: "+proprietaire);
         }
         System.out.println() ;
         }
        public boolean Est_neuf() {
         if ( proprietaire =="" ) return true ;
         else return false ;
         }
        public void Acheter(String nom) {
         proprietaire = nom ;
         }
    }
```

```java
public class BD extends Livre {
    private boolean encouleur ;
    public BD(String t,String a,double p,int nb, boolean c){
     super(t,a,p,nb) ;
     encouleur = c ;
     }
    }
```

```java
public class Album extends Livre {
    boolean page_coloree[];
    public Album(String t, String a, double p, int n){
     super(t,a,p,n) ;
     page_coloree = new boolean[n];
     int i ;
     for (i=0 ; i<100 ; i++)
     page_coloree[i] = false ;
     }
    public void Colore(int num_page){
     if((page_coloree[num_page] == false) && !Est_neuf()){
     page_coloree[num_page] = true ;
     } else {
     System.out.println("page deja coloriee" ) ;
     }
     }}
```

```java
public class Test {
public static void main(String[] args) {

 Livre l1 = new Livre("Le petit prince","St Exupéry",10.40, 50) ;
 Livre l2 = new Livre("Contes","Grimm",14.40,254) ;
 l1.Afficher() ;
 l1.Acheter("moi") ;
 l1.Afficher() ;
 l1.prix = 0.0 ;
 l2.Acheter("lui") ;
 l2.Afficher();

 BD b1 = new BD("Lucky Luke","Morris",10.40, 45, true);
 BD b2 = new BD("Tintin","Herge",200.40, 45, false) ;
 b1.Acheter("moi");
 b1.Afficher() ;
 b2.Afficher() ;

 Album a1 = new Album("Dora","Dora", 3.5,300) ;
 a1.Afficher() ;
 a1.Colore(23) ;
 a1.Acheter("moi");
 a1.Colore(23) ;
 }
}
```

1. Expliquez les informations que le programme va afficher lors de son exécution.
2. Dans la classe Livre, l'attribut prix a été défini sans règle d'encapsulation (public, private...) Comment Java va-t-il l'interpréter ? Comment pouvez-vous le tester ou comment est-il testé dans le programme ?
3. Expliquez comment on teste dans ce programme si un livre est neuf ou non.

4. Décrivez la hiérarchie de classe décrite dans ce programme et expliquer le processus d'appel entre les constructeurs.

5. Expliquez comme ce programme gère le fait de colorer une page d'un album à colorer.

Exercice 2 : Héritage de classe et constructeurs

1. Un site internet est spécialisé dan la vente de livres pour enfant. Ces livres sont soit des bandes dessinées, soit des albums à colorer. Un livre est défini par son titre, son auteur, son prix et son nombre de pages. Les bandes dessinées sont soit en couleur soit en noir & blanc alors l'utilisateur a la possibilité de colorer une page d'un album présenté. Proposez, implémentez et testez sous Eclipse une solution à ce problème.

2. Le site web veut donner la possibilité aux utilisateurs de revendre un livre et de s'échanger deux bandes dessinées si elles ont un prix équivalent. Modifier le programme précédent pour prendre en compte ces fonctions supplémentaires.

3. Enfin, le site web veut étendre son offre d'œuvre culturelles à des films (DVD) qui sont définis eux aussi par un titre, un auteur et un prix mais avec en plus une information sur la durée du film. Comment modifier la hiérarchie de classe pour intégrer ces modifications ? Programmez-le.

Exercice 3 : Héritage de classe et polymorphisme

Vous devez développer pour une chaîne de télévision un logiciel permettant de gérer la programmation journalière d'émissions télévisuelles. Pour simplifier le problème, toutes les heures ou les durées sont représentées par des valeurs entières (13h, 15h, 22h...). Une émission peut être de trois types : divertissement, fiction et reportage. Une émission de type divertissement dure obligatoirement 2 heures, possède un nom et est systématiquement présentée par un animateur. Une fiction se définit par le nom du film, l'année de sa réalisation, le nom du réalisateur et s'il s'agit ou non d'une rediffusion. Enfin un reportage est défini par un nom, un thème (fixé parmi les trois thèmes : information, animalier, culturel) et une durée.

1. Proposez une solution fondée sur les héritages entre classe pour représenter toutes les émissions possibles. Donnez pour chaque classe, la liste de ses attributs et les paramètres de ses constructeurs. Comment coderiez-vous le fait que le thème d'un reportage soit prédéfini ?

2. Implantez cette solution sous Eclipse en Java et testez tout d'abord les classes que vous avez imaginées en instantiant différents objets de votre choix (avec les constructeurs) pour chacune de celles-ci.

3. La programmation d'une émission dans la journée dépend du type d'émission mais se traduit par le fait de lui fixer une heure de début de diffusion et de calculer l'heure de fin.
   a. Les divertissements durent systématiquement 2 heures, mais on ne peut les programmer qu'entre 18h et 23h.
   b. Les fictions qui ne sont pas des rediffusion ne se programment qu'en début de soirée, c'est-àdire qu'à 21h, alors qu'une rediffusion peut se programmer n'importe quand dans la journée.
   c. Enfin, les reportages ne se programment qu'à des heures creuses (14h –18h et 0h-6h) et s'ils ont une durée inférieure, égale à 1heure. Proposez une solution fondée sur la notion de classe abstraite et de polymorphisme permettant de décrire la programmation ou non n'importe quelle émission à une heure donnée. Comment initialiser efficacement ces heures de début et de fin de diffusion.

4. Définissez enfin un programme télé comme un ensemble fini (tableau) d'émissions hétérogènes que vous remplirez d'émission de votre choix, programmez à une heure de votre choix. Décrivez puis implémentez les algorithmes vous permettant :
   a. Affichez la liste des émissions programmées dans la journée
   b. Testez s'il y a une superposition de programmation.
   c. Affichez heure par heure les émissions programmées pour vérifier que tous les créneaux horaires ont bien été remplis.