

## Héritage

### 🔧 Exercice 1 :

1. Ce code compile-t-il ? Si la réponse est non, expliquer les erreurs. Si la réponse est oui, qu'affiche-t-il ?

```

1  public class Livre{
2      private String titre;
3      private int nbEmprunt;
4      private static int nbEmpruntTotal;
5      private boolean disponible = true;
6
7      public Livre(String s){
8          this.titre = s;
9      }
10
11     public emprunt(){
12         disponible = false;
13     }
14
15     public static void main(String[] args){
16         Livre a = new Livre();
17         Livre b = Livre("Tintin et le vaccin interdit");
18
19         a.emprunt();
20         Livre.nbEmprunt++;
21         Livre.nbEmpruntTotal++;
22
23         b.emprunt();
24         Livre.nbEmprunt++;
25         Livre.nbEmpruntTotal++;
26
27         System.out.println(a.nbEmpruntTotal);
28     }
29 }
```

2. Ce code compile-t-il ? Si la réponse est non, expliquer les erreurs. Si la réponse est oui, qu'affiche-t-il ?

```

1  public class Livre{
2      protected String titre;
3      protected int dateParution;
4
5      public Livre(String s){
6          this.titre = s;
7      }
8
9      public void affiche(){
10         System.out.println("Je suis un livre de " + dateParution

```

```

11         + " ! Mon titre est " + this.titre + ".");
12     }
13 }

```

```

1  public class BD extends Livre{
2
3      public BD(String s){
4          super(s);
5      }
6
7      public void affiche(){
8          System.out.println("Je suis une BD de " + dateParution
9              + " ! Mon titre est " + this.titre + ".");
10     }
11 }

```

```

1  public class Main{
2      public static void main(String[] args){
3          Livre a = new Livre("Alice au pays du Covid");
4          BD b = new BD("Tintin et le vaccin interdit");
5
6          a.affiche();
7          b.affiche();
8      }
9  }

```

### 🔧 Exercice 2 : Où sont les erreurs dans ce code ?

```

1  public class Test {
2
3      public static void main(String[] args) {
4          final Livre l = new Livre();
5          l.setTitre("Mobidic");
6      }
7  }

```

```

1  public final class Livre {
2      private String titre;
3      private String auteur;
4      final private int cout;
5
6      public Livre() {
7          this("", "");
8      }
9
10     public Livre(String p_titre, String p_auteur) {
11         titre = p_titre ;
12         auteur = p_auteur ;
13         cout = 10;

```

```


14
15         final int i;
16         i = 1;
17
18         final int i2 = 2;
19         i2 = 5;
20     }
21
22     public void setTitre(String s) {
23         titre = s;
24     }
25
26     public final void afficher() {
27         System.out.println("Titre : " + titre + "\n Auteur : " + auteur );
28     }
29 }

```

```

1  public class BD extends Livre{
2      private String dessinateur;
3
4      public BD() {
5          super();
6          dessinateur = "";
7      }
8
9      public BD(String p_titre, String p_auteur, String p_dessinateur) {
10         super(p_titre, p_auteur);
11         dessinateur = dessinateur;
12     }
13
14     public void afficher() {
15         super.afficher();
16         System.out.println("\n Dessinateur : " + dessinateur );
17     }
18
19     public void afficher(String s) {
20         this.afficher();
21         System.out.println(s);
22     }
23
24
25 }

```

 **Exercice 3 :** On va continuer à utiliser la classe Box de la séance précédente. Dans une interface graphique, on peut noter qu'il y a des rectangles particuliers avec du texte, et qu'il y a des rectangles encore plus particuliers avec du texte et qui déclenche une action quand on clique dessus... Il semble que l'héritage soit bien utile ici !

1. Créez une classe TextBox qui n'est rien de plus qu'une Box à laquelle on ajoute du texte. Implémentez un constructeur qui prend en argument quatre **int** représentant les coordonnées des deux coins du TextBox et une chaîne de caractères qui représente le texte.

- 2 Implémentez une méthode **public** `String toString()`.
- 3 Qu'avez-vous choisi pour la visibilité de vos variables d'instance de la classe `Box` (par exemple pour les coordonnées des coins)?
- 4 Mettez les variables d'instance en `private` (sans rien changer d'autres). Si vous avez une erreur de compilation, quelle est-elle?  
Expliquez votre erreur et corrigez la.
- 5 Expliquez ci-dessous votre choix entre `private` et `protected`. Même si ce n'est pas la solution que vous privilégiez, on vous impose maintenant d'utiliser `private`. Modifiez votre code si besoin est.
- 6 Redéfinissez la méthode `equals` de la classe `Object`. Vérifiez que vous codez bien une redéfinition en utilisant l'annotation `@Override`. Vous pouvez tester votre méthode avec le code suivant.

```
1 Box b = new Box(0,10,10,0);
2 TextBox tb = new TextBox(0,10,10,0,"hello");
3 System.out.println(b.equals(tb) + " ?? " + tb.equals(b));
4 Box fb = new TextBox(0,10,10,0,"hello");
5 System.out.println(fb.equals(tb) + " ?? " + tb.equals(fb));
```

Que devrait être le résultat de l'exécution de ce code? Testez le!

- 7 Implémentez une méthode `clone()` pour dupliquer un élément.
- 8 On va placer ces lignes de code dans la méthode `main`.

```
1 Box a = new Box(100, 200, 300,100);
2 Box b = (Box) a.clone();
3 System.out.print(a==b);
4 System.out.print(" | "+a.equals(b));
5 Pixel p = b.getCoinHG();
6 p.setCoordonnees(0,200);
7 System.out.print(a);
```

Modifiez si besoin votre code pour que l'exécution affiche `false|true` (ce ne sont pas les mêmes `Box`, mais deux `Box` avec les mêmes coordonnées sont égales).

Le résultat de la ligne 7 est-il celui que vous attendiez? Si non, corrigez votre code.