

## TD JAVA (HERITAGE, POLYMORPHISME ET INTERFACES)

### Exercice 1 (Comprendre les constructeurs)

```
public class Livre {  
  
    public static int nombreLivres;  
    private String auteur, titre, dateParution;  
    private int nombrePages;  
  
    public Livre(String auteur, String titre){  
        this.auteur = auteur; this.titre = titre;  
    }  
  
    public Livre(String auteur, String titre, String dateParution) {  
  
        this.auteur = auteur; this.titre = titre;  
        this.dateParution = dateParution;  
    }  
  
    public Livre(String auteur, String titre, String dateParution, int nombrePages)  
{  
  
        this.auteur = auteur;  
        this.titre = titre;  
        this.dateParution = dateParution;  
        this.nombrePages = nombrePages;  
    }  
}
```

On peut créer un objet Livre en ayant besoin seulement de son titre et du nom de l'auteur. D'où le premier constructeur. On peut créer un objet Livre en ayant besoin en plus de sa date de parution. D'où le deuxième constructeur. On peut créer un objet Livre en ayant besoin de en plus du nombre de pages.

- **Question 1** : comment peut-on améliorer l'écriture du 2ème et du 3ème constructeur? Cette amélioration aura pour but de diminuer le nombre d'instructions.
- **Question 2** : Dans quel constructeur doit-on placer l'instruction servant à compter le nombre de livres dans un programme? Quelle est cette instruction?
- **Question 3** : Ecrivez un petit programme qui permet de créer des livres tout en affichant le nombre de livres créés.

## Exercice 2 (Compte d'épargne)

### Description

Un compte d'épargne possède les mêmes propriétés qu'un compte de débit avec quelques spécificités.

Dans un compte d'épargne, le solde n'est jamais négatif. Et même, il est forcément supérieur à zéro. Lors de la création d'un compte d'épargne, la banque offre une somme de 10 euro au titulaire.

Un compte d'épargne rapporte de l'argent variable selon un taux qui dépend du type de compte. Le taux d'intérêt annuel est fixé par la banque lors de la création de ce compte.

Chaque fin d'année la banque fait le calcul du nouveau montant du solde de l'épargne en rajoutant les intérêts.

### Travail demandé

**A partir de la description précédente, élaborer une spécification d'une classe Java modélisant un compte d'épargne et créer une classe pour le test.**

## Exercice 3 (Compte comparables)

C'est une interface générique, comme l'indique le `<T>` mais il n'y a pas grand-chose à savoir sur le sujet pour l'utiliser. Il faudra, en implémentant l'interface, indiquer à la place du T le type des objets qu'on veut comparer entre eux.

L'interface `Comparable<T>` déclare une seule méthode :

```
public int compareTo(T o);
```

Quand cette méthode est implémentée, elle doit retourner une valeur strictement négative, nulle ou strictement positive selon que l'objet concerné (celui qui correspond à la référence `this`) est plus petit que l'objet `o`, égal à l'objet `o` ou plus grand que l'objet `o`.

On comparera les comptes selon leurs soldes.

### Travail demandé

**Il s'agit ici de compléter la classe `Compte` pour faire en sorte d'avoir une classe qui implémente l'interface `java.lang.Comparable<T>`.**

**Ensuite, de créer un classe qui s'appelle `Banque` qui contient une structure de type `ArrayList` où on stocke tous les comptes bancaires créés tout en gardant l'ordre naturel des comptes. (L'ordre naturel ici est selon le solde)**