

 	TP2: PROGRAMMATION ORIENTÉE OBJET JAVA	LEI (S5)
	<ul style="list-style-type: none"> ➤ Manipulation des classes et des objets ➤ Utilisation de méthodes statiques 	PAR MR : M.MOUKHAFI

On reprend la partie 1 du TP 1

Exercice 1 : Contrôle des variables private par les modificateurs

1. Ajoutez un prix aux livres (nombre qui peut avoir 2 décimales de type Java double) avec 2 méthodes `getPrix` et `setPrix` pour obtenir le prix et le modifier. Ajoutez au moins un constructeur qui prend le prix en paramètre. Testez.

Si le prix d'un livre n'a pas été donné, la description du livre (`toString()`) devra indiquer "Prix pas encore donné". Attention, un livre peut être gratuit (0 est une valeur possible pour un prix) ; la valeur -1 indiquera que le prix n'a pas encore été donné.

2. On bloque complètement les prix : un prix ne peut être donné qu'une seule fois et ne peut être modifié ensuite (une tentative pour changer le prix ne fait qu'afficher un message d'erreur).

Récrivez la méthode `setPrix` (et autre chose si besoin est). Vous ajouterez une variable booléenne `prixFixe` (pour "prix fixé") qui indiquera que le prix ne peut plus être modifié.

Faut-il écrire une méthode "`setPrixFixe`" pour modifier la variable booléenne ? Ajoutez une méthode "`isPrixFixe`" qui renvoie vrai si le prix a déjà été fixé.

Exercice 2 : Comparaison de 2 livres en utilisant une méthode d'instance

Dans la classe `Livre` écrivez une méthode d'instance `compare` pour comparer 2 livres sur leur nombre de pages.

compare prend un livre en paramètre et elle renvoie 0 si le livre a le même nombre de pages que l'instance qui reçoit le message, 1 si l'instance courante ("this") a plus de pages que le paramètre et -1 sinon.

Pour tester, vous utiliserez un code semblable au suivant :

```
System.out.print("L'auteur du plus gros livre est ");  
String auteurPlusGrosLivre; // Le code pour trouver le livre qui a le plus de pages  
(entre 2 livres) //
```

Ecrivez 2 versions : une avec if et l'autre avec switch . . .
System.out.println(auteurPlusGrosLivre);

Exercice 3 : Comparaison de 2 livres en utilisant une méthode de classe

Dans la classe Livre, écrivez une méthode de classe compare2 pour comparer 2 livres (passés en paramètres) sur leur nombre de pages.

Exercice 4 : Délégation (une nouvelle classe Comptable)

Cet exercice montre comment des instances peuvent collaborer pour obtenir le résultat voulu. Pour simplifier, vous vous placerez dans le cas où les prix des livres sont bloqués : le prix d'un livre ne peut être donné qu'une seule fois (dans un constructeur ou dans un modificateur) et ne peut être modifié ensuite.

1. Créez une classe Comptable qui possède une méthode de signature "void comptabiliser(Livre l)". Une instance de cette classe permettra de calculer le prix total de tous les livres qu'on lui aura passé par la méthode "comptabiliser". Vous mettrez la classe Comptable dans un fichier à part (pas dans le fichier Livre.java). Testez votre nouvelle classe dans la méthode main de TestLivre. 2 comptables seront créés. Ils comptabiliseront chacun quelques livres. A la fin, on fera afficher le total des prix ainsi comptabilisés par chacun des 2 comptables.

2. Vous ne devez pas modifier la classe Comptable pour cette question. Pour simplifier l'utilisation de la classe Livre, on veut cacher la classe Comptable aux "clients" de la classe Livre. Les livres vont faire enregistrer automatiquement leur prix par un comptable (un même comptable pour tous les livres) : appelez directement la méthode comptabiliser depuis la méthode setPrix (et depuis les constructeurs si nécessaire). Pour cela vous devrez utiliser un this explicite. A tout moment on peut demander (à qui ?) le total des prix comptabilisés. Testez.