

 	TP1: 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

Partie 1 :

Exercice 1 : une première "vraie" classe

Voici le source de la classe Livre :

```
public class Livre {
// Variables
private String titre, auteur; private int nbPages

// Constructeur
public Livre(String unAuteur, String unTitre) {

auteur = unAuteur;

titre = unTitre; }

// Accesseur
public String getAuteur() {

return auteur; }

// Modificateur
public void setNbPages(int n) {

nbPages = nb; }
```

Corrigez quelques petites erreurs et ajoutez une méthode main() pour

Exercice 2 : accesseurs et modificateurs

Modifiez la classe Livre :

- Ajoutez un accesseur pour la variable titre et la variable nbPages.
- Ajouter un modificateur pour les variables auteur et titre.
- Changez le modificateur de nbPages : il ne devra changer le nombre de pages que si

on lui passe en paramètre un nombre positif, et ne rien faire sinon, en affichant un message d'erreur. On remarquera l'intérêt des variables private : on peut contrôler leur modification dans les éventuelles méthodes qui permettent de les modifier.

- Dans la méthode main(),
 - donnez le nombre de pages de chacun des 2 livres,
 - faites afficher ces nombres de pages,
 - calculez le nombre de pages total de ces 2 livres et affichez-le.

Exercice 3 : 2 classes dans 2 fichiers source pour une application

Ecrivez une classe TestLivre dans un fichier TestLivre.java. Cette classe a une seule méthode main() qui fait ce que fait la méthode main() de la classe Livre. Faites exécuter la méthode main() de la classe TestLivre.

- ☐ Créer 2 livres, ☐ Faire afficher les auteurs de ces 2 livres.

Exercice 4 : 2 classes dans un seul fichier source

Supprimez le fichier TestLivre.class.

Insérez la classe TestLivre dans le fichier Livre.java et supprimez la méthode main() de la classe Livre (si ce n'est déjà fait).

Compilez le fichier Livre.java. Vérifiez que le compilateur a créé 2 fichiers ".class" et faites exécuter la méthode main() de la classe TestLivre.

En général il vaut mieux avoir un fichier .java par classe. Vous terminerez donc ce TP avec les 2 classes Livre et TestLivre dans 2 fichiers séparés.

Exercice 5 : méthode toString()

1. Dans la classe Livre, ajoutez une méthode afficheToi() qui affiche une description du livre (auteur, titre et nombre de pages). Utilisez afficheToi() dans la méthode main() de TestLivre.
2. Ajoutez l'instruction System.out.println(livre) où livre désigne un des livres que vous avez créés. Vous essaierez de comprendre ce qui est affiché après avoir fait les 2 questions suivantes.
3. Ajoutez une méthode toString() qui renvoie une chaîne de caractères qui décrit le livre. Donnez à la méthode toString() le même profil que la méthode de même nom de la classe java.lang.Object (cherchez dans les API du JDK). Exécutez à nouveau la classe TestLivre. Voyez ce qui est affiché maintenant par l'instruction System.out.println(livre). Miracle ! println() utilise automatiquement la méthode

toString() de la classe de l'objet qu'il a à imprimer. Essayez de trouver une explication rationnelle en faisant la question suivante.

4. Il faut savoir chercher dans la documentation de l'API (javadoc). En partant de la classe java.lang.System et en cliquant sur les liens, retrouvez dans la documentation que System.out.println(objet) affiche ce que retourne la méthode toString de la classe de objet.
5. Modifiez la méthode afficheToi() pour utiliser toString().

Exercice 6 : les constructeurs

1. Enlevez (provisoirement) le seul constructeur de la classe Livre. Sans ajouter de nouveau constructeur, peut-on quand même créer un nouveau livre dans la méthode main ? Si c'est possible, créez un livre écrit par Victor Hugo et faites afficher son titre et son auteur.
2. Remettez le constructeur que vous avez enlevé. Est-ce que le code de la méthode main de la question précédente fonctionne toujours (testez) ?
3. Ajoutez 2 constructeurs pour avoir 3 constructeurs dans la classe :
 - un constructeur qui n'a pas de paramètre,
 - un qui prend en paramètre l'auteur et le titre du livre, ○ et l'autre qui prend en plus le nombre de pages.

Utilisez les 3 constructeurs (et éventuellement d'autres méthodes) pour créer 3 livres de 300 pages dans la méthode main de TestLivre.

Partie 2

Exercice :

Définissez une classe Individu composée d'un nom, d'une adresse, et d'un numéro de telephone. Vous écrirez aussi les méthodes constructeurs, affectation et consultation des champs et affichage.

Modifiez la classe Individu pour pouvoir :

- afficher le nombre d'objets créés,
- gérer la liste des individus créés,
- afficher la liste des individus créés,