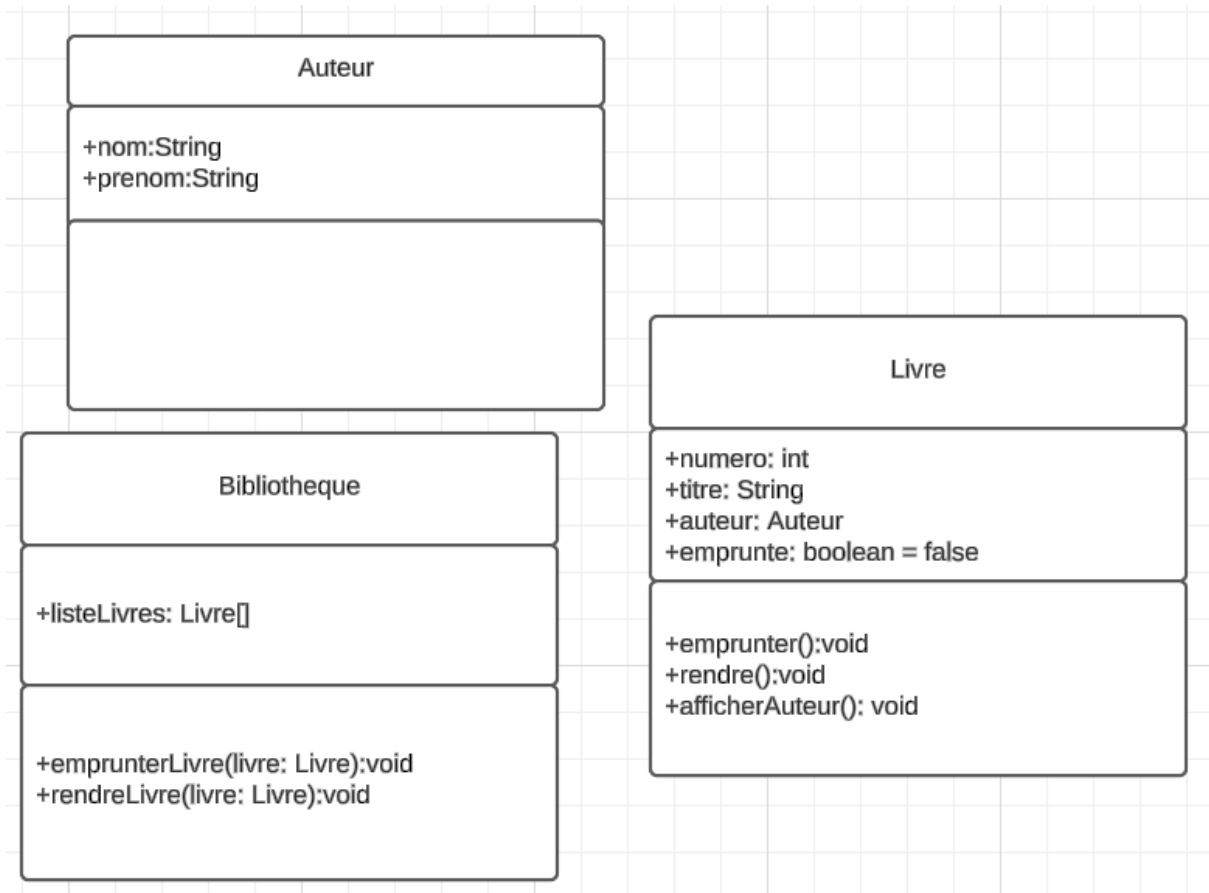


TP 4 - Partie 2

Objectif du TP : S'initier à des concepts avancés de programmation en JAVA

Essayez de reproduire le fonctionnement de cette bibliothèque.

Vous disposerez de toutes les indications du TP.



Vous préciserez le mot clé “public” devant vos attributs, vos classes et vos méthodes(ou fonction, cela inclut le constructeur) pour y avoir accès de partout dans votre dossier.

Exemples :

```
public class Livre {  
    public int num;  
    public void maMethode() {
```

- Un auteur est une classe représenté par les attributs “nom” de type String et “prenom” de type String.
- Un livre est une classe représentée par les attributs “numero” de type int, “titre” de type String, auteur de type Auteur et “emprunte” de type boolean. L’attribut emprunte représentera l’état du livre (true si le livre est déjà emprunté à un instant T, et sinon false)
- Un livre est représenté par la fonction emprunter() de type void et la fonction rendre() de type void.
- Une bibliothèque est représentée par l’attribut listeLivres de type Livre[]. Il s’agit d’un tableau qui comporte des objets de la classe Livre.
- Le constructeur de la bibliothèque est déjà créé. Vous pourrez vous en inspirer pour parcourir un tableau dans la suite de l’exercice.

Bien lire les informations ci-dessus avant de commencer à code

Étape 1 : Création de la classe Auteur.

1. Se rendre dans le fichier Auteur.java
2. Ajouter deux attributs : “titre” et “auteur”.
3. Créer le constructeur Auteur. Il prendra en paramètre deux variables : pNom de type String et pPrenom de type String. Le constructeur doit affecter la valeur pNom à l’attribut nom et pPrenom à l’attribut prenom. Le constructeur affiche également “Creation d’un nouvel auteur”

Étape 2 : Création de la classe Livre.

1. Se rendre dans le fichier Livre.java
2. Ajouter 4 attributs : “numero”, “titre”, “auteur”, “emprunte”.
3. Créer le constructeur Livre. Il prendra en paramètre quatre variables : pNumero de type int, pTitre de type String, pAuteur de type Auteur, pEmprunte de type boolean. Le constructeur doit affecter la valeur pNumero à l’attribut numero, pTitre à l’attribut titre, pAuteur à l’attribut auteur, pEmprunte à l’attribut emprunte. Le constructeur affiche également “Creation d’un nouveau livre”

4. Créer la méthode emprunter. La méthode emprunter ne prend aucun paramètre est de type void (donc ne retourne rien). La méthode emprunter accède à l'attribut emprunte de votre classe et regarde si sa valeur est égale à "true".
 - a. Si c'est le cas, cela veut dire que le livre est déjà à l'état emprunté. On affichera donc "Le livre [TITRE du livre] a deja ete emprunte".
 - b. En revanche, sinon, cela voudra dire que la valeur de emprunte est false. On affichera donc "Le livre [TITRE du livre] a bien ete emprunte". Pour rester cohérent, il faudra alors changer la valeur de l'attribut emprunte et la passer à true.
5. La méthode rendre fonctionne de la même manière. Elle ne prend aucun paramètre et ne retourne rien.
 - a. Si l'attribut emprunte est à true, on affichera "Le livre [TITRE] a ete rendu" et on passera la valeur de l'attribut emprunte à false.
 - b. Sinon, on affichera "Le livre [TITRE] n'est pas emprunte et ne peut donc pas etre rendu".
6. Écrire ensuite la méthode afficherAuteur qui ne prend pas de paramètre et ne retourne rien. Cette méthode affichera simplement incluant le titre du livre et son auteur. Exemple : L'auteur de [TITRE] est [NOM AUTEUR] [PRENOM AUTEUR]. L'attribut auteur est une instance de la classe Auteur. Vous pourrez donc aller récupérer ces informations grâce à auteur.nom ...

Étape 3 : Création de la classe Bibliothèque.

1. Le constructeur et l'attribut sont déjà renseignés. listeLivres et un attribut qui est initialisé avec une taille 10. La boucle for du constructeur permet de parcourir un tableau pListeLivres passé en paramètre et en même temps le tableau listeLivres. Il attribue automatiquement l'indice i de pListeLivres à listeLivres. Ainsi, le jeu de données dans Main.java est passé en paramètre lors de la création de la bibliothèque et le constructeur permet de remplir l'attribut listeLivres de tous les livres.

```
// Création des livres
Livre livre1 = new Livre(pNumero:1, pTitre:"Le Vieil Homme et la Mer", auteur1, pEmprunte:false);
Livre livre2 = new Livre(pNumero:2, pTitre:"Le Seigneur des Anneaux", auteur2, pEmprunte:false);
Livre livre3 = new Livre(pNumero:3, pTitre:"Harry Potter a l'ecole des sorciers", auteur3, pEmprunte:false);
Livre livre4 = new Livre(pNumero:4, pTitre:"Le Vieil Homme et les Enfants", auteur1, pEmprunte:false);
Livre livre5 = new Livre(pNumero:5, pTitre:"Le Hobbit", auteur2, pEmprunte:false);
Livre livre6 = new Livre(pNumero:6, pTitre:"Harry Potter et la Chambre des Secrets", auteur3, pEmprunte:false);
Livre livre7 = new Livre(pNumero:7, pTitre:"Pour qui sonne le glas", auteur1, pEmprunte:false);
Livre livre8 = new Livre(pNumero:8, pTitre:"Le Silmarillion", auteur2, pEmprunte:false);
Livre livre9 = new Livre(pNumero:9, pTitre:"Harry Potter et le Prisonnier d'Azkaban", auteur3, pEmprunte:false);
Livre livre10 = new Livre(pNumero:10, pTitre:"Le Soleil se leve aussi", auteur1, pEmprunte:false);

// Création de la liste de livres
Livre[] listeLivres = {livre1, livre2, livre3, livre4, livre5, livre6, livre7, livre8, livre9, livre10};

// Création de la bibliothèque
Bibliotheque bibliotheque = new Bibliotheque(listeLivres);
```

2. On passe aux choses sérieuses. Créez la méthode `emprunterLivre`. Cette méthode prend un livre en paramètre et ne retourne rien (type `void`). Le but de cette méthode est de parcourir le tableau `listeLivres` de son premier à son dernier index et de comparer chaque Livre du tableau au livre passé en paramètre (simple comparaison d'égalité du type `tab[i] == livre`). Si le livre est égal à un des livres de mon tableau, c'est qu'il existe dans la bibliothèque : vous écrirez dans ce cas l'instruction **`listeLivres[i].emprunter()`**; qui permet de lancer la méthode `emprunter` de la classe `Livre`.
3. De la même manière, la créez la méthode `rendreLivre` qui prend un livre en paramètre et ne retourne rien. Elle parcourt et compare de la même manière que `emprunterLivre`. Au lieu d'appeler la méthode `emprunter`, elle appellera la méthode `rendre` de la classe `Livre`.