

METHODES LOGICIELLES (ET JAVA)

— TRAITEMENT D'UNE APPLICATION COMPLÈTE —

Cours de Centrale Pékin — 5A

Olivier ROUX

olivier.roux@ec-nantes.fr

2019/2020

TD



Traitement d'une application complète

- **Livre et Bibliothèque:**

L'objectif est de construire un programme autour d'une classe `Bibliothèque` avec des fonctionnalités de gestion d'objets, instances de deux classes de `Livres`.

Questions *(Livres et Bibliothèque)*

- Q.1 On souhaite construire une classe pour représenter un livre qui sera ensuite rangé dans une bibliothèque.

Cette classe **Livre** possède trois attributs qui donnent respectivement le titre du livre (**titre**), le nom de son auteur (**auteur**) et son numéro d'identification (**id**, assimilé à un élément de la classe des entiers).

Écrire la déclaration de la classe et de ses attributs. À partir de la question suivante, des accesseurs en lecture sont supposés avoir été écrits.

Questions *(Livres et Bibliothèque)*

- Q.2 Écrire un constructeur pour la classe **Livre** qui prend en paramètres son titre, le nom de son auteur et son numéro d'identifiant.
- Q.3 Écrire une méthode **affiche()** qui affiche les informations relatives à un livre selon le modèle suivant ("Au centre de la terre" étant un roman de Jules Verne qui porte le numéro 3):

`Au centre de la terre (Jules Verne) – 3`

Classe Livre *(Livres et Bibliothèque)*

```
class Livre {  
  
    protected String titre;  
  
    protected String auteur;  
  
    protected Integer id;  
  
    public Livre (String _titre, String _auteur, Integer _id){  
  
        titre = _titre;          //titre = new String(_titre);  
  
        auteur = new String(_auteur);  
  
        id = new Integer(_id);  }  
  
    public String get_titre() {  
  
        return titre;    }  
  
    public String get_auteur() {  
  
        return auteur;   }  
  
    public Integer get_id() {  
  
        return id;    }  
  
    public void affiche(){  
  
        System.out.println(titre + " (" + auteur + ") " + "--- " +id); }  
  
}
```

Classe Livre (*Livres et Bibliothèque*)

```
class Livre {  
  
    protected String titre;  
  
    protected String auteur;  
  
    protected Integer id;  
  
    public Livre (String _titre, String _auteur, Integer _id){  
        titre = _titre;          //titre = new String(_titre);  
        auteur = new String(_auteur);  
        id = new Integer(_id);    }  
  
    public String get_titre() {  
        return titre;    }  
  
    public String get_auteur() {  
        return auteur;    }  
  
    public Integer get_id() {  
        return id;    }  
  
    public void affiche(){  
        System.out.println(titre + " (" + auteur + ") " + "--- " + id); }  
  
}
```

Classe Livre *(Livres et Bibliothèque)*

```
class Livre {  
  
    protected String titre;  
  
    protected String auteur;  
  
    protected Integer id;  
  
    public Livre (String _titre, String _auteur, Integer _id){  
  
        titre = _titre;           //titre = new String(_titre);  
  
        auteur = new String(_auteur);  
  
        id = new Integer(_id); }  
  
    public String get_titre() {  
  
        return titre;    }  
  
    public String get_auteur() {  
  
        return auteur;   }  
  
    public Integer get_id() {  
  
        return id;    }  
  
    public void affiche(){  
  
        System.out.println(titre + " (" + auteur + ") " + "--- " +id); }  
  
}
```

Questions (suite) *(Livres et Bibliothèque)*

- Q.4 Certains livres comprennent des références bibliographiques, autrement dit une liste de références à d'autres livres. Construire une classe **LivreAvecBibliographie** prenant en compte cette particularité.
(On ne demande ici que la déclaration des attributs et un constructeur qui prend les trois mêmes paramètres qu'à la question 2 et qui initialise la liste des références bibliographiques à la liste vide).
- Q.5 Écrire une méthode (`ajouteRef(...)`) qui ajoute un livre passé en paramètre à la bibliographie d'un autre.
- Q.6 Écrire maintenant une méthode `affiche()` qui affiche pour un livre avec bibliographie : les informations relatives seront écrites comme demandé à la question Q.3.
Le nombre de références bibliographiques sera précisé à la ligne suivante selon le modèle suivant :

```
Au centre de la terre (Jules Verne) - 3  
Références bibliographiques - 24
```

Classe LivreAvecBibliographie (Q4)

```
class LivreAvecBibliographie extends Livre {  
    protected LinkedList<Livre> references;  
  
    public LivreAvecBibliographie(String _titre,String _auteur,Integer _id) {  
        super(_titre, _auteur, _id);  
        references = new LinkedList<Livre>(); // liste vide initialement  
    }  
  
    public LinkedList<Livre> get_references(){  
        return references;  
    }  
}
```

Discussion *(Livres et Bibliothèque)*

- Sur héritage et this et super :

"class LivreAvecBibliographie **extends** Livre"

"super(_titre, _auteur, _id);" exécute le constructeur de la classe mère

"return this.references;" retournerait les références de l'objet courant

"return super.titre;" retournerait le titre de l'objet de la classe mère
(exactement comme : "return this.titre; »)

Questions (suite) *(Livres et Bibliothèque)*

- Q.4 Certains livres comprennent des références bibliographiques, autrement dit une liste de références à d'autres livres. Construire une classe **LivreAvecBibliographie** prenant en compte cette particularité.
(On ne demande ici que la déclaration des attributs et un constructeur qui prend les trois mêmes paramètres qu'à la question 2 et qui initialise la liste des références bibliographiques à la liste vide).
- Q.5 Écrire une méthode (`ajouteRef(...)`) qui ajoute un livre passé en paramètre à la bibliographie d'un autre.
- Q.6 Écrire maintenant une méthode `affiche()` qui affiche pour un livre avec bibliographie : les informations relatives seront écrites comme demandé à la question Q.3.
Le nombre de références bibliographiques sera précisé à la ligne suivante selon le modèle suivant :

Au centre de la terre (Jules Verne) - 3
Références bibliographiques - 24

Classe LivreAvecBibliographie (Q5 et Q6)

```
class LivreAvecBibliographie extends Livre {  
  
    protected LinkedList<Livre> references;  
  
    public LivreAvecBibliographie(String _titre, String _auteur, Integer _id) {super(_titre, _auteur, _id);  
  
        references = new LinkedList<Livre>();}  
  
    public LinkedList<Livre> get_references(){return references;}  
  
    public void ajouteRef(Livre livre){  
  
        references.add(livre);  
  
    }  
  
    public void affiche() {  
  
        super.affiche();  
  
        System.out.println("\n" + "# Références biblio" + " : " +  
references.size() );  
  
    }  
  
}
```

Questions (suite) *(Livres et Bibliothèque)*

- Q.7 On s'intéresse maintenant à une bibliothèque numérique qui comprend une collection des **Livre** tels que nous les avons définis dans les questions précédentes. Écrire la définition de la classe **Bibliothèque**
 - écrire: attributs, (constructeur, accesseur,)
 - créer une méthode `ajouteLivre(...)` qui ajoute un livre dans la bibliothèque
 - créer une méthode `affiche(...)` qui permet d'afficher le contenu de la bibliothèque

Classe Bibliothèque (Q7)

```
class Bibliotheque {  
    protected LinkedList<Livre> lesLivres = new LinkedList<Livre>();  
  
    public void ajouteLivre(Livre _livre) {  
        lesLivres.add(_livre);  
    }  
  
    public void affiche() {  
        for (Livre unLivre : lesLivres) {  
            unLivre.affiche();  
        }  
    }  
}
```

Discussion (*Livres et Bibliothèque*)

- Sur les références comme *attribut* ou comme *héritage* dans la classe **Bibliotheque**

—> **2 solutions : Bibliotheque et Bibliotheque2**

- les références s'appellent **lesLivres** dans **Bibliotheque**
- et **this** dans **Bibliotheque2**

Classe Bibliothèque2 (Q7 bis)

```
class Bibliotheque2 extends LinkedList<Livre>{

    public void ajouteLivre(Livre _livre) {

        this.add(_livre);    }

    public void affiche() {

        for (Livre unLivre : this) {

            unLivre.affiche();

        }

    }

}
```

Questions (suite) *(Livres et Bibliothèque)*

- Q.8 Écrire une méthode **infoLivre(...)** pour cette classe **Bibliothèque** qui prend en argument un **titre de Livre** et qui affiche les informations associées au livre s'il existe dans la bibliothèque.

S'il s'agit d'un **LivreAvecBibliographie**, afficher également le nombre de références bibliographiques.

Classe Bibliothèque (Q8 bis)

```
// Dans la classe Bibliotheque2

// Méthode qui prend en argument un titre de Livre et qui affiche les
// informations associées au livre s'il existe dans la bibliothèque.

public void afficheLivre(String _titre) {

    for (Livre unLivre: this) {

        if (_titre.equals(unLivre.get_titre())) {

            unLivre.affiche();

        }

    }

}
```

Questions (suite) *(Livres et Bibliothèque)*

- Q.9 Écrire une méthode qui prend en paramètre un livre et qui renvoie le nombre de fois où celui-ci est cité dans les références bibliographiques des livres de la bibliothèque.

Classe Bibliothèque (Q9)

```
// Méthode qui prend en paramètre un livre et qui renvoie le nombre de fois où celui-ci
// est cité dans les références bibliographiques des ouvrages de la bibliothèque.

public int compte(Livre l) {

    int cpt = 0;

    for (Livre unLivre : lesLivres) {

        if ((unLivre instanceof LivreAvecBibliographie) ) {

            // indexOf(Object o) : Returns the index of the first occurrence
            // of the specified element in this list,
            // or -1 if this list does not contain the element.

            cpt=((LivreAvecBibliographie)unLivre).get_references().indexOf(l)== -1 ? cpt : cpt+1;

        }

    }

}
```

Questions (suite) *(Livres et Bibliothèque)*

- Q.10 Écrire une méthode qui calcule le nombre de citations d'un auteur, c'est-à-dire le nombre de fois où un auteur passé en paramètre est cité dans les références bibliographiques de la bibliothèque.

Classe Bibliothèque (Q10)

```
// Méthode qui calcule le nombre de citations d'un auteur, c'est-à-dire le nombre de fois
// où un auteur passé en paramètre est cité dans les références bibliographiques de la bibliothèque.

public int comptEcrivain(String ecrivain) {

    int cpt = 0;

    for (Livre unLivre: lesLivres) {

        if ((unLivre instanceof LivreAvecBibliographie) ) {

            for (Livre unAutreLivre: ((LivreAvecBibliographie) unLivre).get_references() ) {

                if (unAutreLivre.get_auteur().equals(ecrivain)) {      cpt ++;      }

            }

        }

    }

    return cpt;

}
```

Test d'exécution *(Livres et Bibliothèque)*

- Création de 3 livres de Jules Verne ("*Voyage au centre de la terre*", "*De la terre a la lune*" (avec 1 référence), "*Autour de la Lune*" (avec 2 références)),
- Introduction dans une Bibliothèque, et affichage du contenu de cette Bibliothèque,
- Comptage du nombre de fois où chacun des livres ci-dessus est cité dans des livres de la Bibliothèque,
- Comptage du nombre de citations de l'auteur Jules Verne.

Classe Test de l'application

```
class Test {  
  
    public static void main (String argv[]) {  
  
        Livre l1 = new Livre("Voyage au centre de la terre", "Jules Verne", 1864);  
  
        System.out.println("##### Livre l1 ##### »); l1.affiche();  
  
        LivreAvecBibliographie l2 = new LivreAvecBibliographie("De la terre a la lune", "Jules Verne", 1865);  
l2.ajouteRef(l1);  
  
        LivreAvecBibliographie l3 = new LivreAvecBibliographie("Autour de la Lune", "Jules Verne", 1870);  
l3.ajouteRef(l1); l3.ajouteRef(l2);  
  
        Bibliotheque bib = new Bibliotheque(); bib.ajouteLivre(l1); bib.ajouteLivre(l2);  
bib.ajouteLivre(l3);  
  
        System.out.println("##### Biblio ##### »); bib.affiche();  
  
        System.out.println("##### nombre de fois où l1, (resp. l2, l3) est cité dans les références  
bibliographiques des livres de la bibliothèque. #####");  
  
        System.out.println("l1 est cité " + bib.compte(l1) + " fois");  
  
        System.out.println("l2 est cité " + bib.compte(l2) + " fois »); System.out.println("l3 est cité " +  
bib.compte(l3) + " fois");  
  
        System.out.println("##### nombre de citations de l'auteur Jules Verne #####");  
  
        System.out.println("l'auteur Jules Verne est cité " + bib.comptecriv("Jules Verne") + "  
fois");  
  
    } }  
}
```


Résultat de l'exécution *(Livres et Bibliothèque)*

```
>>> java Test
```

```
##### Livre l1 #####
```

```
Voyage au centre de la terre (Jules Verne) --- 1864
```

```
##### Biblio #####
```

```
Voyage au centre de la terre (Jules Verne) --- 1864
```

```
De la terre a la lune (Jules Verne) --- 1865
```

```
# Références biblio : 1
```

```
Autour de la Lune (Jules Verne) --- 1870
```

```
# Références biblio : 2
```

```
##### nombre de fois où l1, (resp. l2, l3) est cité dans les références  
bibliographiques des ouvrages de la bibliothèque. #####
```

```
l1 est cité 2 fois
```

```
l2 est cité 1 fois
```

```
l3 est cité 0 fois
```

```
##### nombre de citations de l'auteur Jules Verne #####
```

```
l'auteur Jules Verne est cité 3 fois
```

Fin
