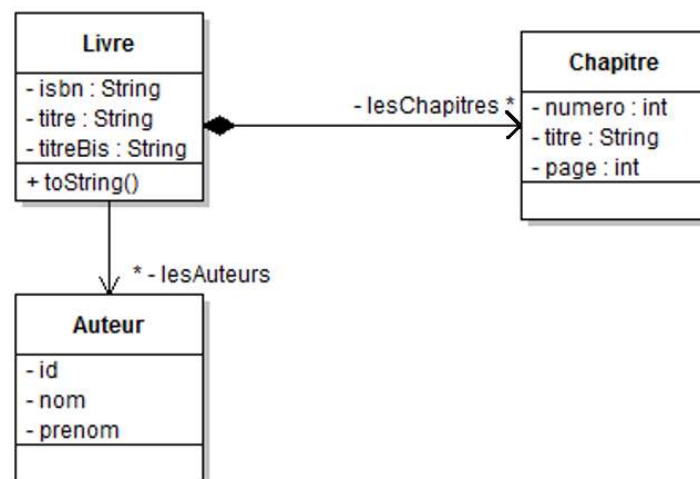


12) La composition

Le diagramme de classes suivant représente le fait qu'un livre contient des chapitres et a été écrit par des auteurs. Il contient un nouveau signe un **losange noir plein** qui indique le concept de la "**composition**". Cela signifie qu'un livre est "composé" de chapitres et que chaque chapitre ne peut exister sans son livre. Par ailleurs si je supprime un livre, il paraîtrait normal que les chapitres soient également supprimés. On dit que les objets Chapitre sont liés au "**cycle de vie**" de l'objet Livre auquel ils appartiennent. Ce qui ne sera pas le cas pour les objets Auteur qui auront leur propre vie en dehors des livres.



L'implémentation en java :

L'idée est non plus de fournir un objet *Chapitre* déjà créé à un objet *Livre*, mais de lui fournir de quoi créer un chapitre qui sera construit **dans** l'objet et ensuite inséré dans la collection *lesChapitres*.

La classe Chapitre (rien de nouveau)

```
public class Chapitre{
    //Propriétés privées
    private int numero;
    private String titre;
    private int page;

    //Constructeur
    public Chapitre(int n, String t, int p){
        this.numero = n;
        this.titre = t;
        this.page = p;
    }

    //Méthode publique
    @Override
    public String toString(){
        return "Chapitre " + this.numero + " - " + this.titre + "\t\t" + this.page;
    }
}
```

La classe Auteur (rien de nouveau)

```
public class Auteur{
    //Propriétés privées
    private int id;
    private String nom;
    private String prenom;

    //Constructeur
    public Auteur(int id, String n, string p){
        this.id = id;
        this.nom = n;
        this.prenom = p;
    }

    //Accesseurs
    public String getNom(){
        return this.nom;
    }

    public String getPrenom(){
        return this.prenom;
    }
}
```

La classe Livre (c'est ici que ça change...)

```
Public class Livre{
    //Propriétés privées
    private String isbn;
    private String titre;
    private String titreBis;
    private ArrayList<Chapitre> lesChapitres;
    private ArrayList<Auteur> lesAuteurs;

    //Constructeur
    public Livre(String isbn, String t, String tb){
        this.isbn = isbn;
        this.titre = t;
        this.titreBis = tb;
        this.lesChapitres = new ArrayList<Chapitre>();
        this.lesAuteurs = new ArrayList<Auteurs>();
    }

    //Méthodes publiques
    public void ajouterAuteur(Auteur unAuteur){
        this.lesAuteurs.add(unAuteur);
    }

    public void ajouterChapitre(int numero, String titre, int page){
        //Création à l'intérieur de Livre d'un chapitre (variable locale)
        Chapitre unChapitre = new Chapitre(numero, titre, page);

        //Insertion de ce chapitre dans la collection
        this.lesChapitres.add(unChapitre);
    }
}
```

Soit un livre très connu par les informaticiens analyste :



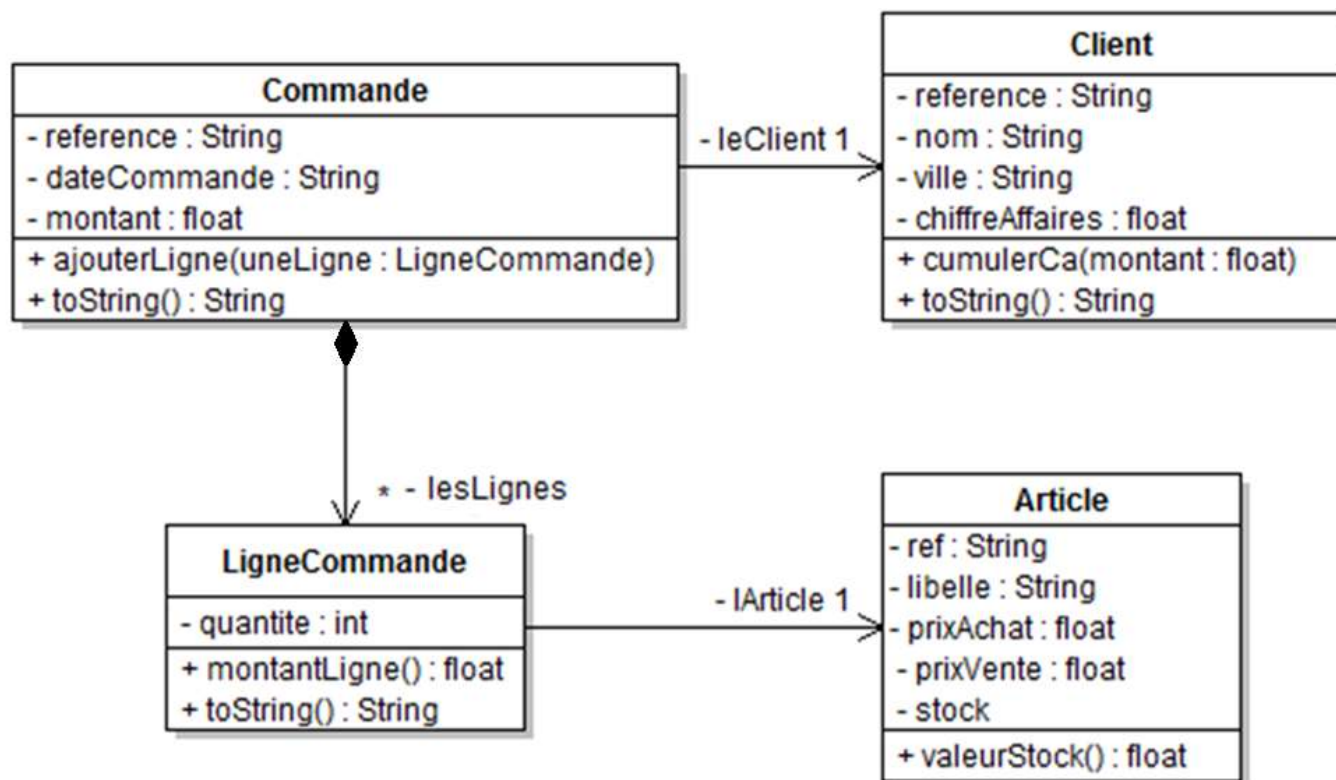
Chapitre 1 – Introduction : contexte d'apparition de MERISE	page 11
Chapitre 2 – Présentation générale de MERISE	page 17
Chapitre 3 – Le système d'information : approche fonctionnelle, sa modélisation	page 51
Chapitre 4 – Le système d'information : approche formelle	page 89
Chapitre 5 – Règles de construction des modèles de données et de traitement	page 165
...	
Chapitre 14 – Conclusion	page 322

Missions :

- Observez la différence entre les méthodes *ajouterAuteur()* et *ajouterChapitre()*,
- Programmer les 3 classes, la méthode *toString()* devra afficher un livre sous une forme proche de celle de l'exemple,
- Dans une classe de test insérer et afficher le livre de l'exemple.

13) Exercice 12 Revenons à notre commande...

La version codée ne représente pas exactement la réalité. En effet une ligne de commande ne peut pas exister sans sa commande ce qui était le cas jusqu'ici. Autre cas de figure, lors de la destruction d'une commande, il paraîtrait normal que les lignes de commande soient également supprimées. en UML :



En cas de suppression d'une commande les lignes de commandes associées seront également supprimées contrairement aux articles et aux clients qui auront leur propre vie en dehors des commandes.

Mission :

- A partir de la version précédente, créer un nouveau projet qui inclut le lien de composition,
- Modifier l'exemple de test en conséquence.