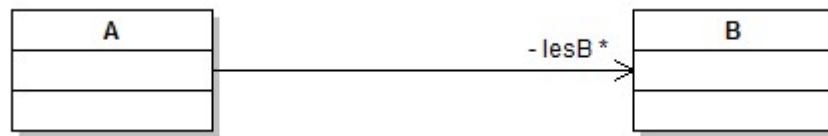


## 10) Traduire les liens de 1 à plusieurs

*L'implémentations se cache dans les détails !*

### a) Lien simple



Implémentation :

La flèche (navigabilité) signifie que l'on n'implémente le lien que du côté A. Sans flèche il faudrait l'implémenter des 2 côtés.

```
public class A{

    //Attributs Etape 1 : déclaration de la collection
    private ArrayList<B> lesB;

    //Constructeur
    public A(){ Etape 2 : instanciation de la collection
        this.lesB = new ArrayList<B>();
    }

    //Méthodes Etape 3 : création d'une méthode pour ajouter des objets dans la collection
    public void ajouterB(B b){
        this.lesB.add(b);
    }
}
```

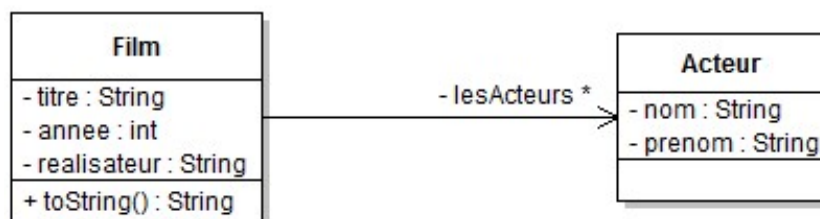
Utilisation :

```
B b;
b = new B();

A a;
a = new A();

a.ajouterB(b);
```

### Exercice 11 :



La méthode ***toString()*** affiche la description d'un film de cette façon :

AMERICAN BEAUTY (1999)  
 De Sam Mendez  
 Avec :  
     Kevin Spacey  
     Annette Bening  
     Thora Birch  
     Wes Bentley

### **Mission :**

Programmer les classes Acteur et Film. Dans un programme de test, créer le film fourni en exemple et afficher sa description.

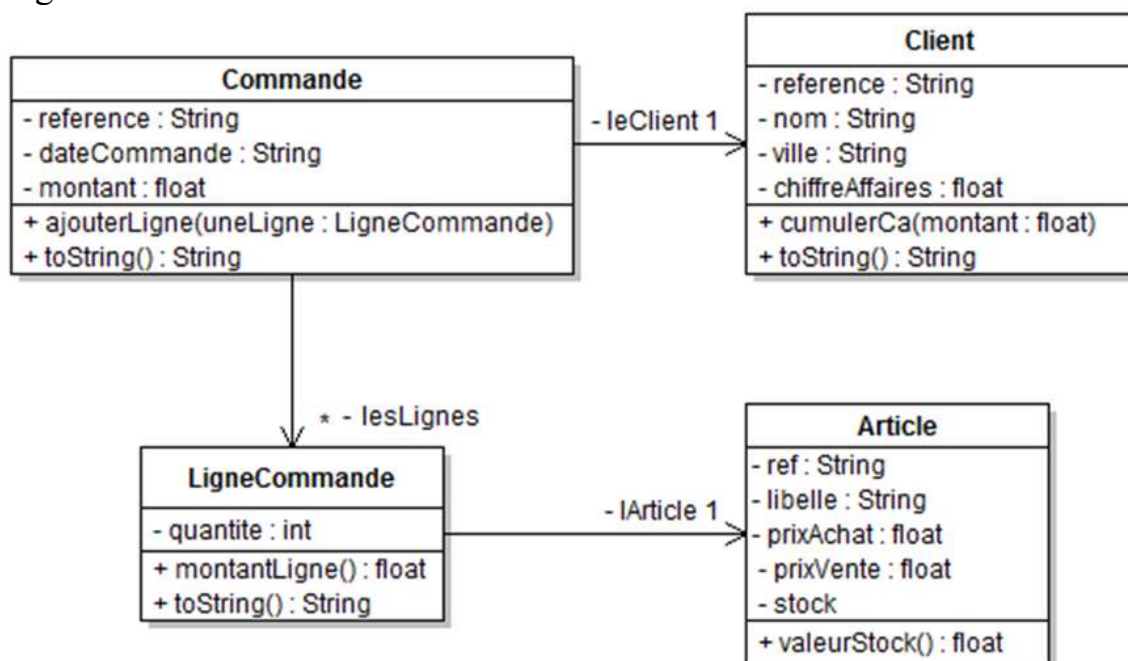
A ce stade nous pouvons maintenant faire la différence entre les classe métiers et les classes techniques.

**Classe métier** : Une classe métier est une classe qui est au cœur des objectifs d'une application (par exemple une application de gestion commerciale va utiliser des objets client, commande, article...).

**Classe technique** : une classe technique est fournie pour réaliser certaines tâches qui ne sont pas en relation directe avec le cœur du métier (accès à une base de donnée, gestion d'une collection, envoi d'un mail...). Les classes techniques n'apparaissent généralement pas sur le diagramme de classes.

## Exercice 12 :      Revenons à notre Commande...

Diagramme de classes UML :



↓ 1 commande comporte 0,1, ou plusieurs LigneCommande (multiplicité \*)

- Compléter la classe **Commande** afin de traduire le lien "lesLignes" ;
- ajouter une méthode `ajouterLigne(uneLigne)` qui permet :
  - de rajouter une ligne de commande à la collection `lesLignes`,
  - de cumuler le montant de la ligne au montant de la commande et au chiffre d'affaires du client,
- modifier la méthode ***toString()*** pour afficher en plus les lignes de commande sous la forme :

Commande					
Référence : 1234					
Date : 12/02/2020					
Reference Client : c1					
Nom : Coltrane					
V1	VTT	300	x	1	300
V2	VTC	250	x	2	500
Total					800
Nouveau chiffre d'affaires du client : 1000					

## Programmation Orientée Objets

Voici une représentation possible de la mémoire pour la commande 1234 :

