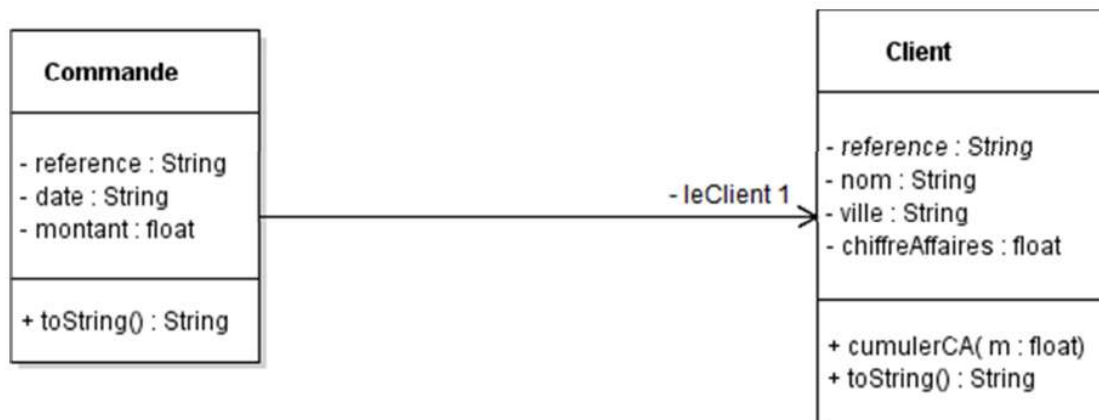


8) Traduire le lien de 1 à 1

On veut traduire le fait qu'une commande est passée par un client.

Ce lien se traduit de la façon suivante en diagramme de classes UML :



1 commande

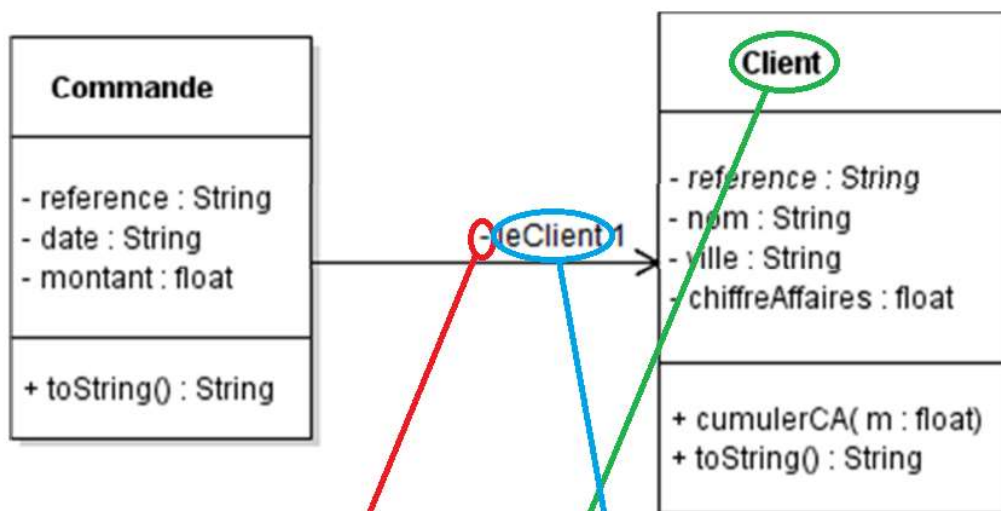
est passée par

1 client

(multiplicité)

Remarque : la flèche sur le lien indique le sens de navigabilité, dans ce cas une commande connaît un client mais un client ne connaît pas ses commandes.

En java



```

public class Commande{
    //Attributs privés
    private String reference;
    private String date;
    private float montant;
    private Client leClient;
    ...
}
  
```

Exercice 8 :

En utilisant la classe **Client** créée dans le chapitre précédent

1) Coder la classe **Commande** en respectant les règles de gestion suivantes :

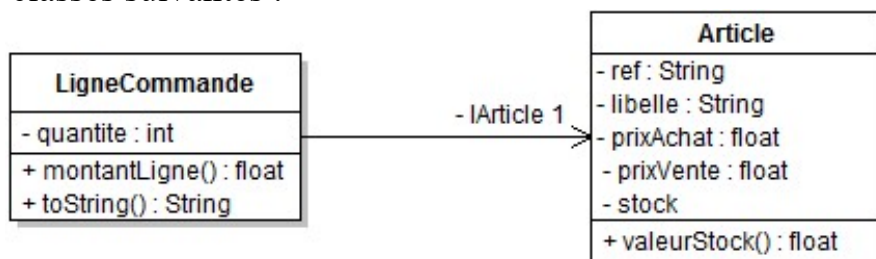
- Le constructeur de Commande valorise la référence, la date, le client et initialise le montant à 0 ;
- Une méthode **toString()** doit retourner la description de la commande y compris la référence, le nom et la ville du client ;
- Créer un accesseur en lecture pour tous les attributs privés et un accesseur en écriture pour l'attribut montant.

2) Tester ce scénario dans un programme de test :

- créer 1 client qui habite Colmar ;
- afficher les renseignements de ce client ;
- créer 2 commandes pour ce client ;
- affecter aux 2 commandes respectivement 1000 € et 500 € ;
- afficher les renseignements de ces 2 commandes ;
- modifier la ville du client qui déménage à Brest ;
- afficher les renseignements des 2 commandes, que peut-on en déduire ? ;
- modifier le montant de la 2^{ème} commande qui passe à 600 € ;
- réafficher les renseignements de la 2^{ème} commande pour vérifier le nouveau montant ;

Exercice 9 :

Soit les 2 classes suivantes :



- Ajouter au même projet les 2 classes (reprendre la classe Article déjà créée) ;

La méthode **toString()** de la classe LigneCommande devra renvoyer une chaîne de caractères sous la forme :

référence libellé prix de vente x quantité total ligne

Par exemple : V1 VTT 300 x 1 300