

Diagrammes de classes

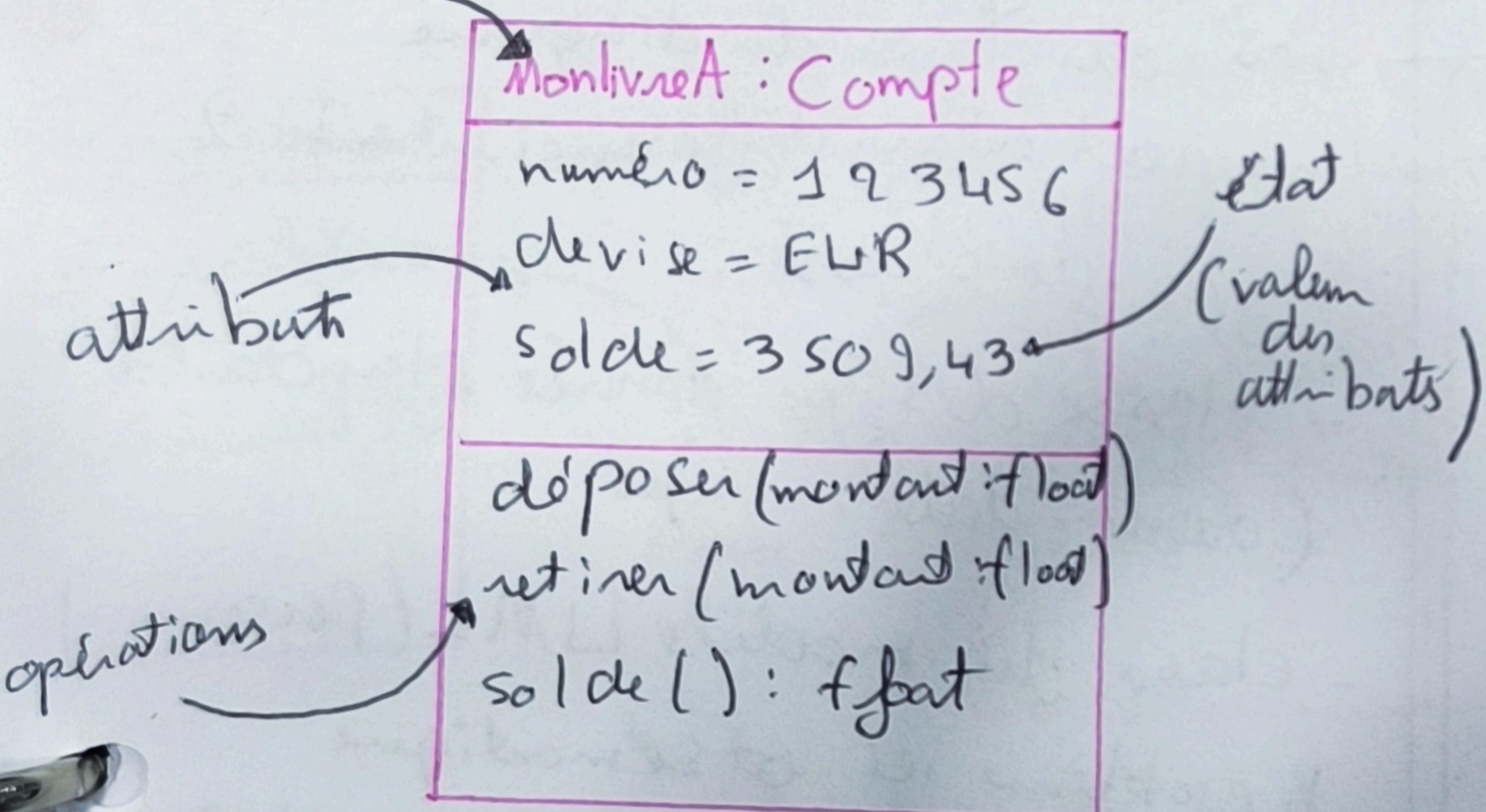
Objet et classe

Objet :

- Entité concrète ou abstraite du domaine d'application
- Décrit par : identité (adresse mémoire)
 - état (attributs)
 - comportement (opérations)

Soit l'instance suivante (l'instance) :

nom de l'objet

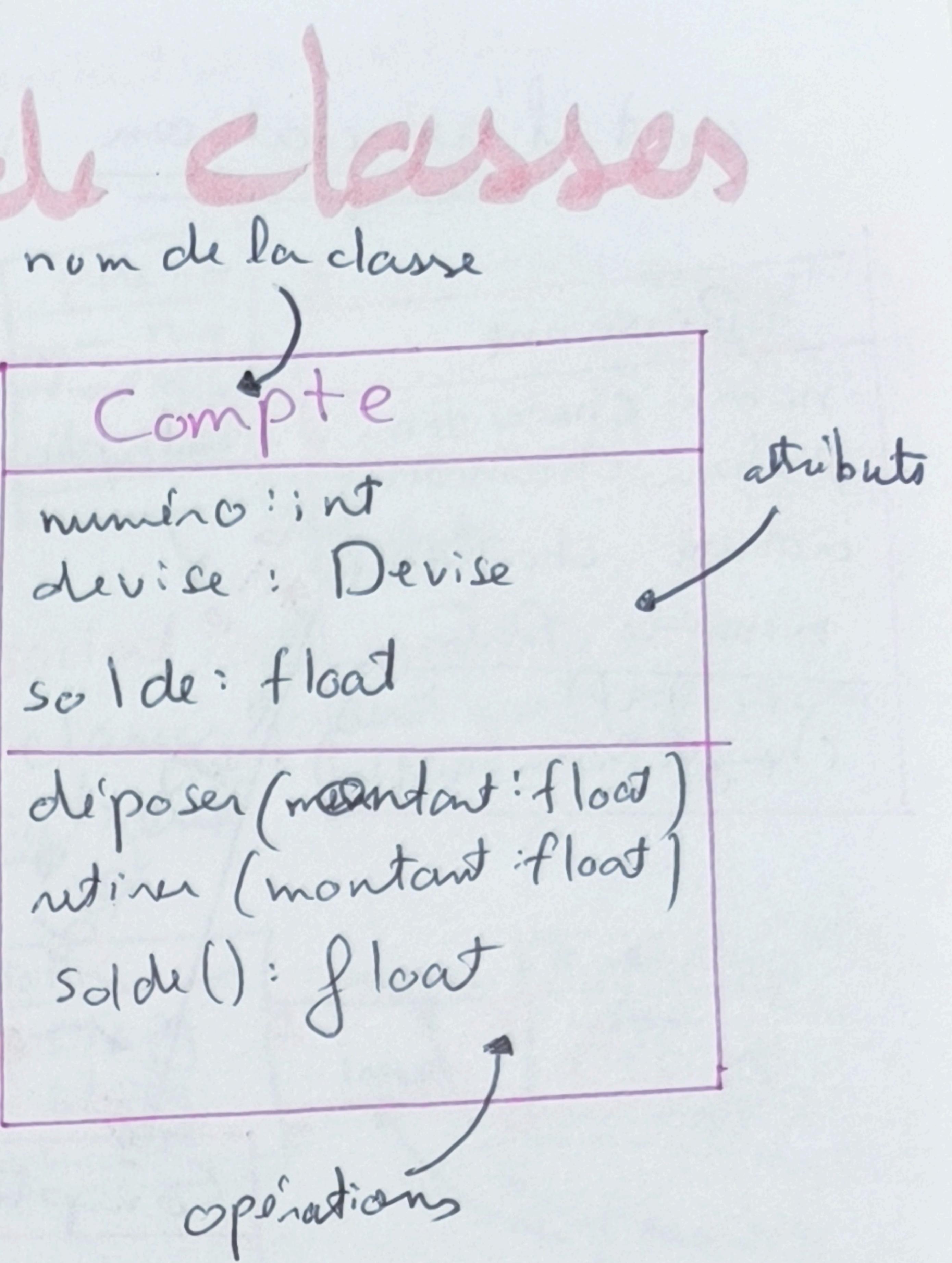


Classe :

une classe est un regroupement d'objets de même nature (mêmes attributs + mêmes opérations)

Objet = instance d'une classe

selon l'exemple précédent l'instance Monlivre A est une instance de la classe Compte suivante :



Attributs:

- Caractéristique partagée par tous les objets de la classe
- Associé à chaque objet une valeur (cas d'un attribut d'objet par attribut de classe)
- Deux objets peuvent avoir le même état d'attributs mais n'ont jamais la même identité + adresses mémoire

Opération:

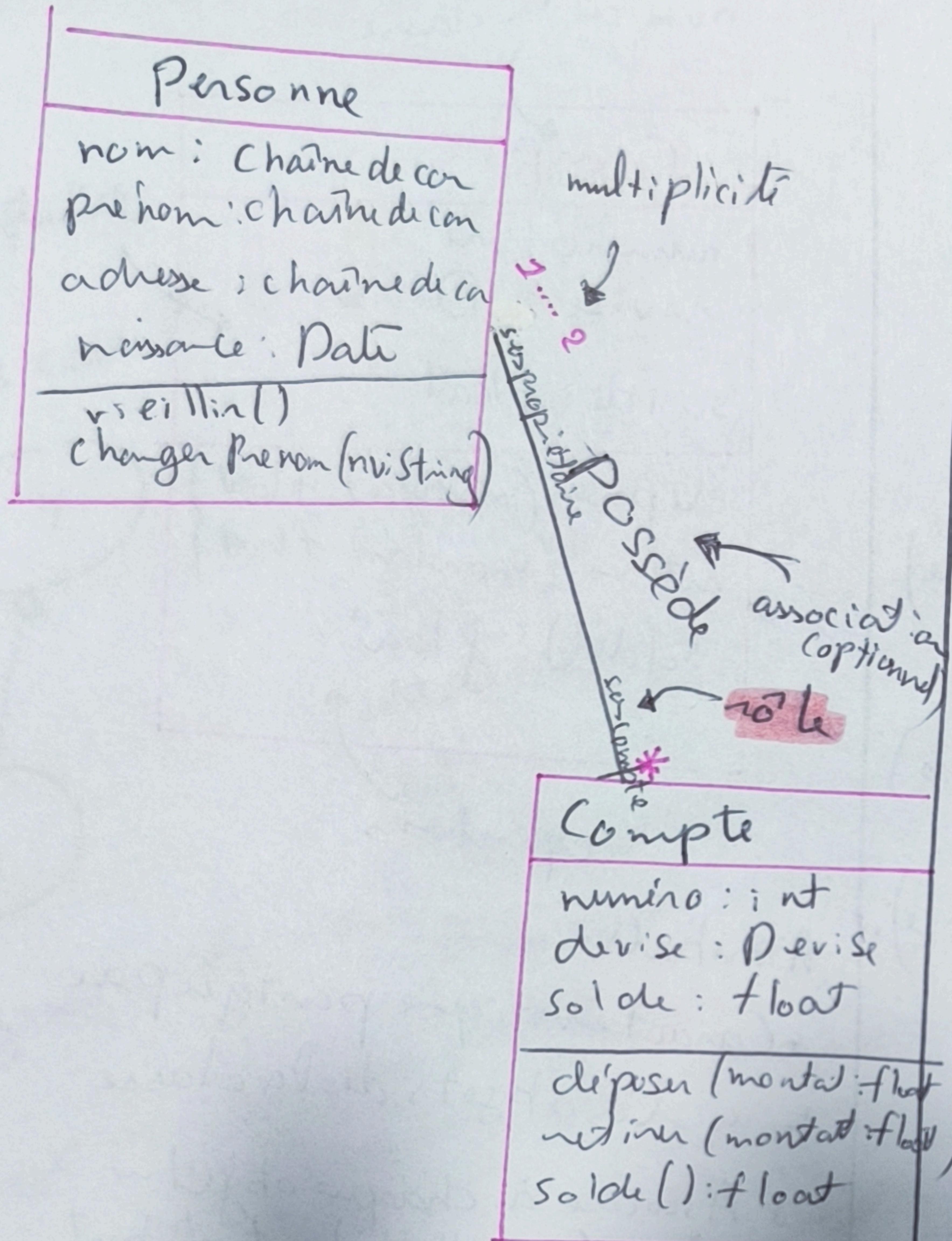
Service qui peut être demandé à tout objet

Comportement commun

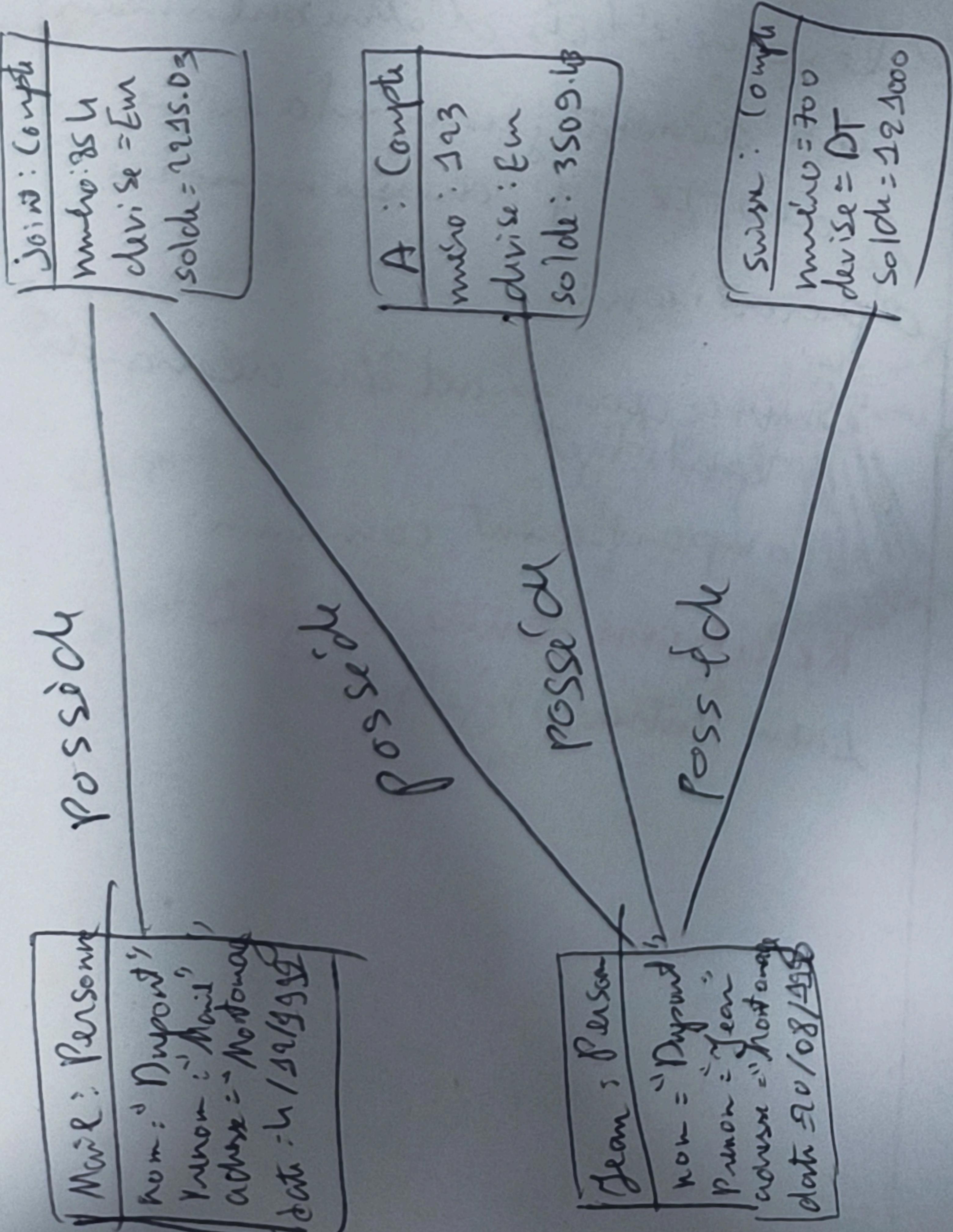
Relations entre objets

Lien entre objets

soit l'association :



donnant par exemple



=> Mr Dupont possède 3 comptes
dont 1 de ses comptes est aussi possédé par Mdm Marie

=> le compte joint possède 2 propriétaires

Attribut et association:

les types des attributs doivent être simple, primitif ou énuméré.

Signifie, pas d'attribut dont le type est une **classe du diagramme** mais il faut avoir une association vers cette classe du diagramme. On peut alors différencier entre 2 types de classe :

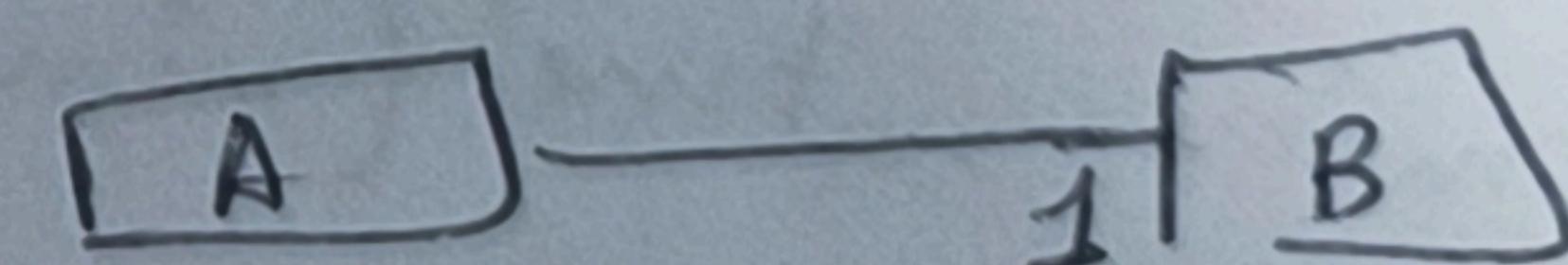
- classe de type donnée standard (devise, date ...)

- classe du modèle UML (Personne...)

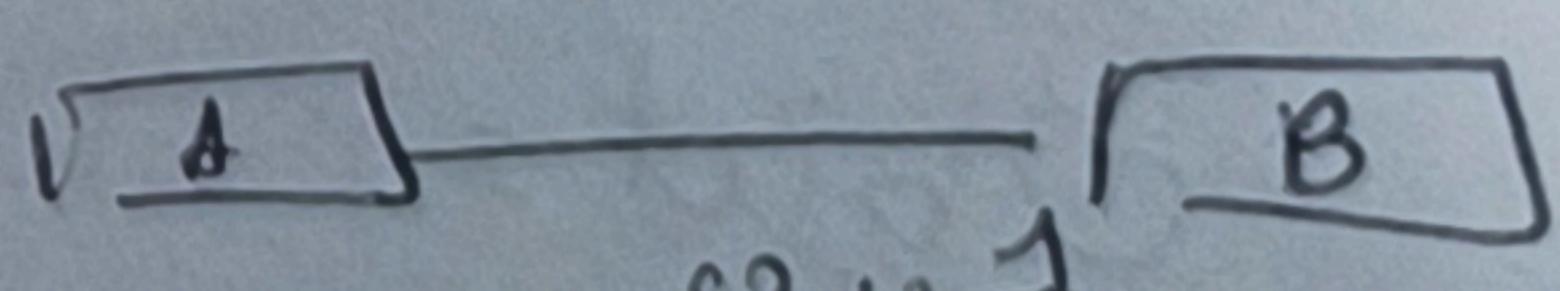
Le problème ici est sémantique : on choisit quelle classe appartient à quel catégorie.

Multiplicité:

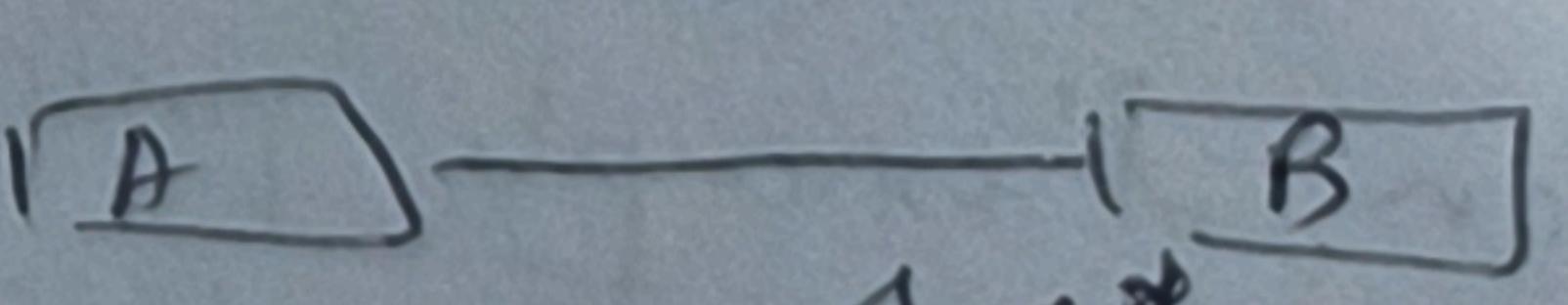
Exactement 1:



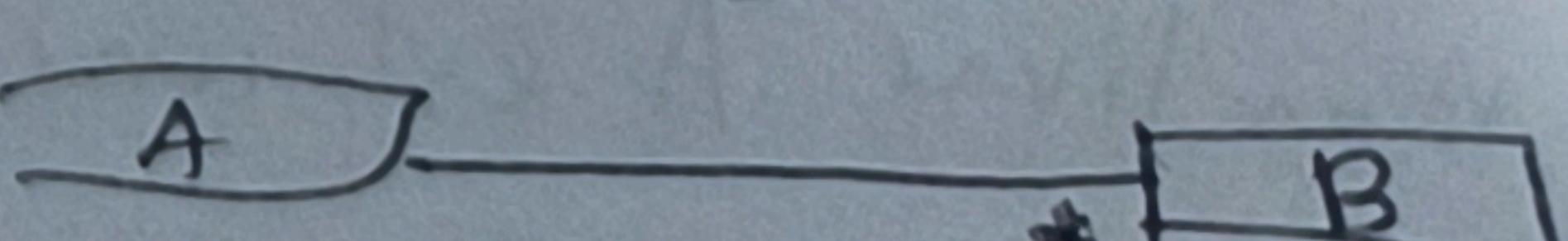
Plusieurs 1:



Un seul 1:

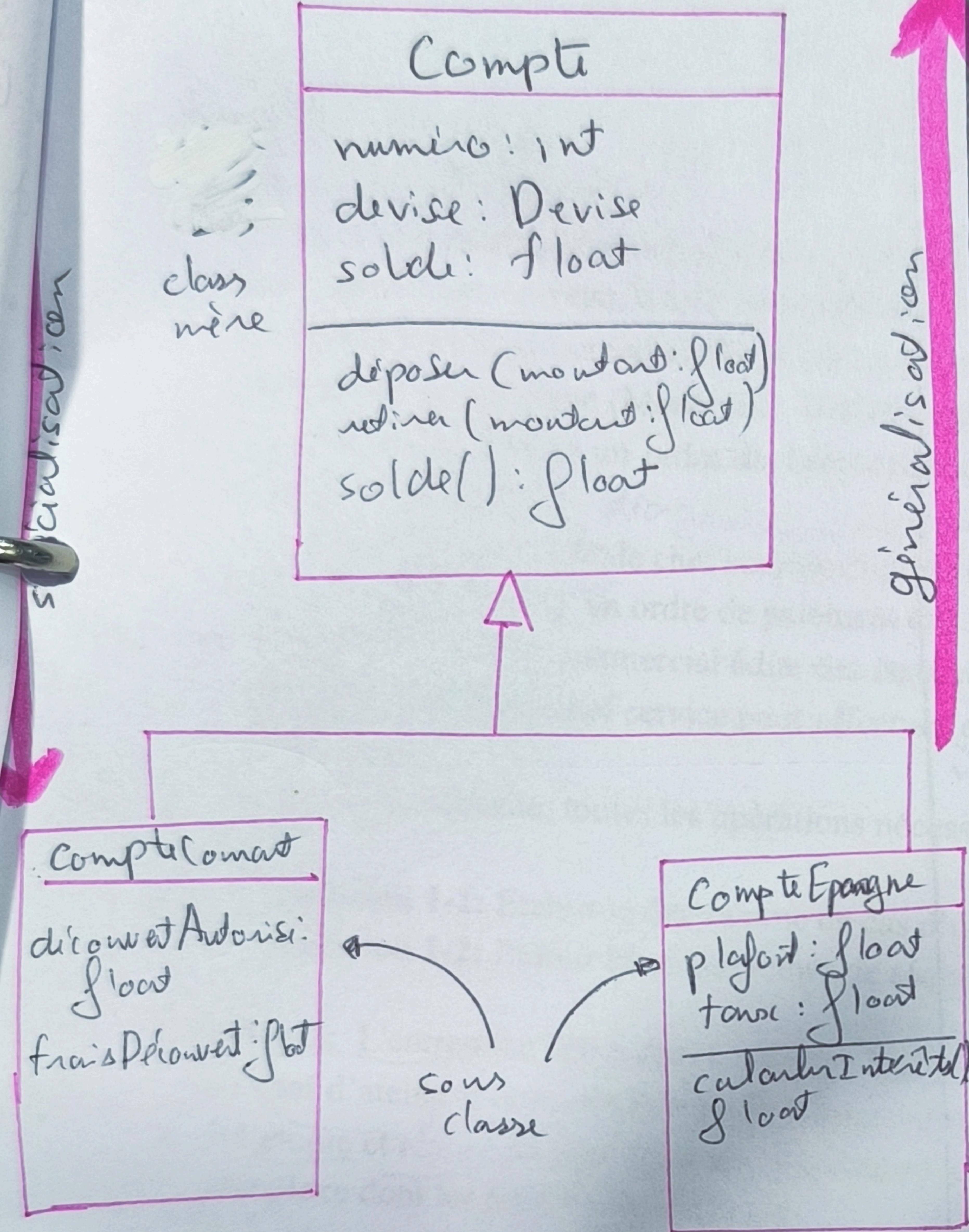


Combien :



arché de classes :
spécialisation / généralisation
soit le diagramme de classe

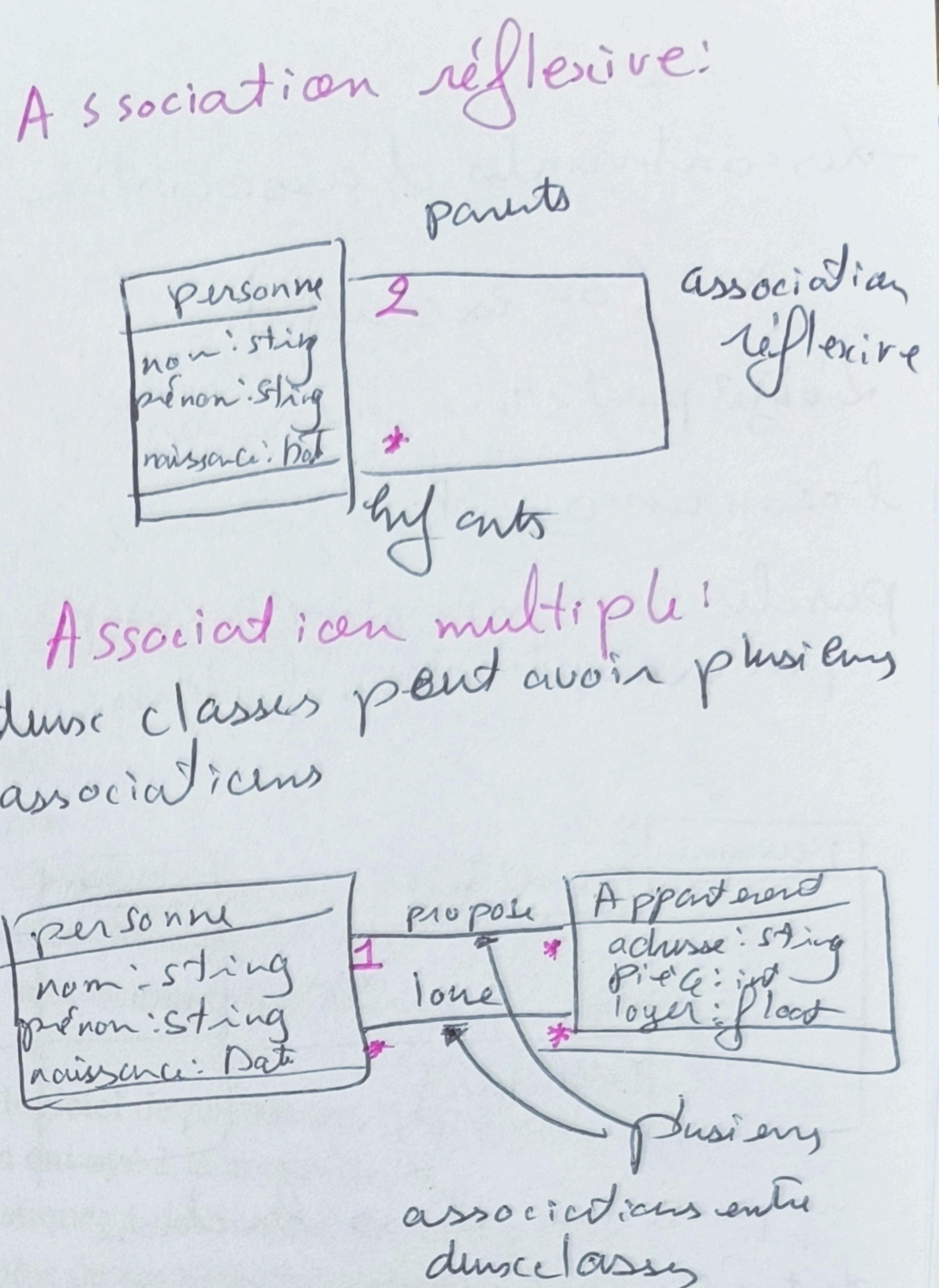
Survol



⇒ les sous classes CompteCompt et Compte Epargne héritent de la classe Compte

classe abstraite :

Classe sans instance car elle est une classe mère et elle est visible dans le cas comme la classe Compte car un compte n'existe pas en soi



Les contraintes :

on peut différencier entre 2 types :

- contraintes de collection :
 - ordered : les éléments doivent être ordonnés étiquetés
 - unique : chaque élément de la collection doit être unique
Une classe ne peut pas avoir 2 fois le même étiqueté
 - bag : collection non ordonnée et non unique, l'ordre n'a pas d'importance et on a des doublons
 - sequence : collection ordonnée et les doublons sont autorisés

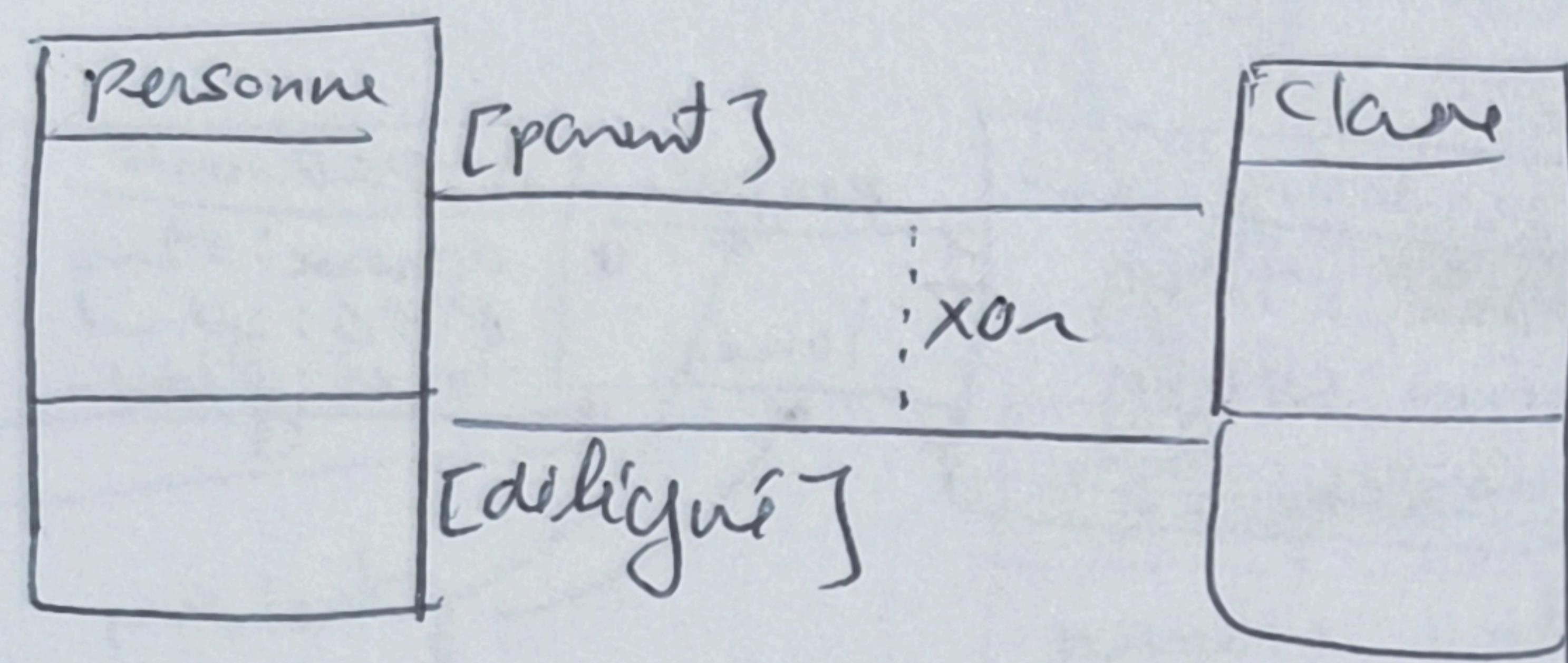
ordered = sequence + unique

- normalisation

- des contraintes d'association

- xor (ou exclusif)

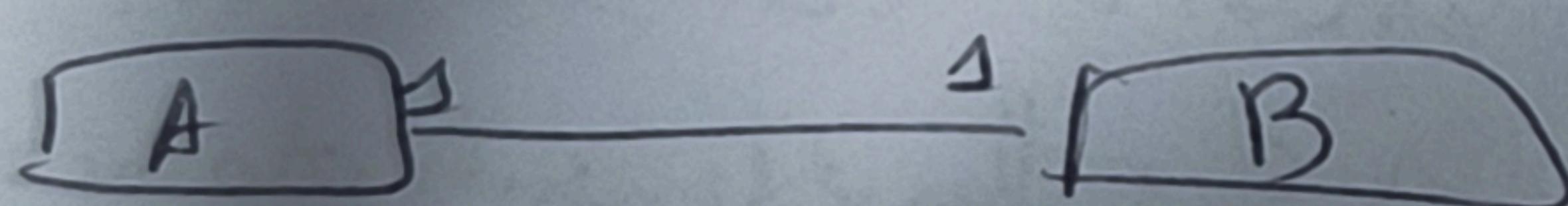
L'objet peut être lié selon
l'une ou la liaison 2
par les 2 en même temps comme
par exemple dans l'acceuse.



une personne dans la classe
doit être soit un parent soit un délégué
ne peut pas être les 2 en même temps

- contrainte de multiplicité:

Par exemple



la liaison doit exister