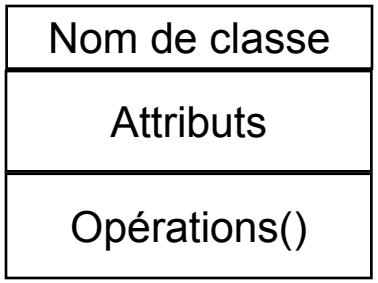
Diagrammes des Classes et d’objets

Le Diagramme de Classes (DCL)

- DCL: est une représentation de la structure statique d'un système, en terme de classes et des relations entre ces classes.

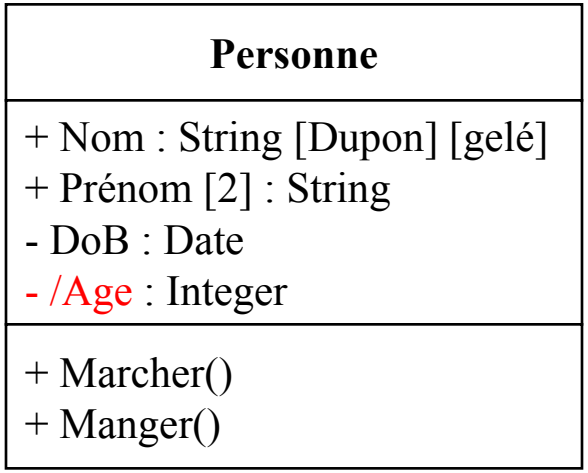
- Une classe: est une description abstraite d'un ensemble d'objets qui partagent les mêmes *Attributs*, *Opérations* et *Relations*

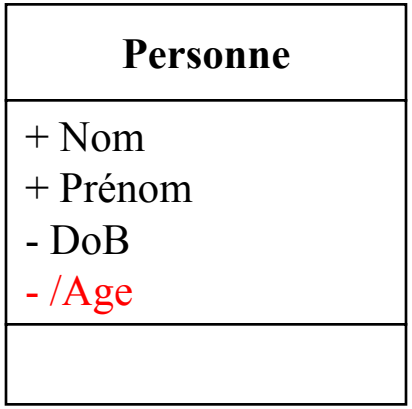
- Représentation d'une classe:

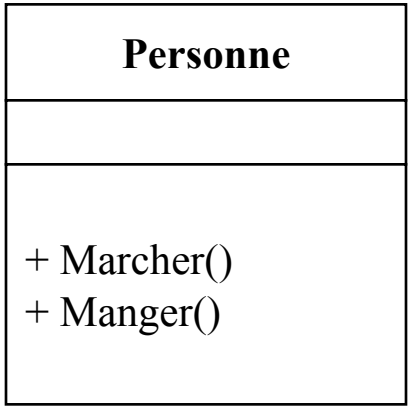


- Syntaxe de déclaration:
  - Attributs: Visibilité Nom_d'attribut [Multiplicité] : Type [Valeur initiale] [{Propriété}]
  - Operations: Visibilité Nom_d'opération ()

Exemples de classes





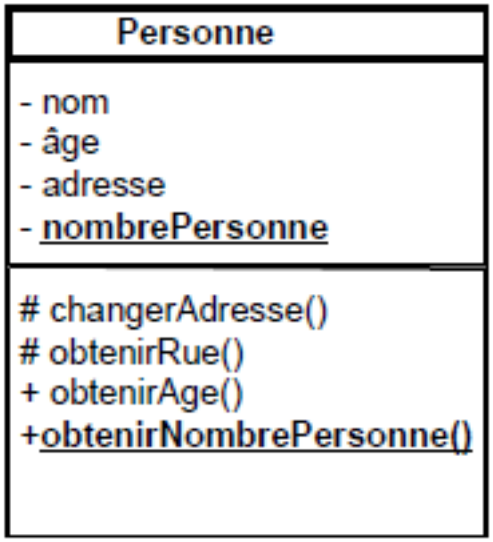


NB: l'attribut Age est dérivé

Les parties peuvent êtres supprimées pour alléger les diagrammes

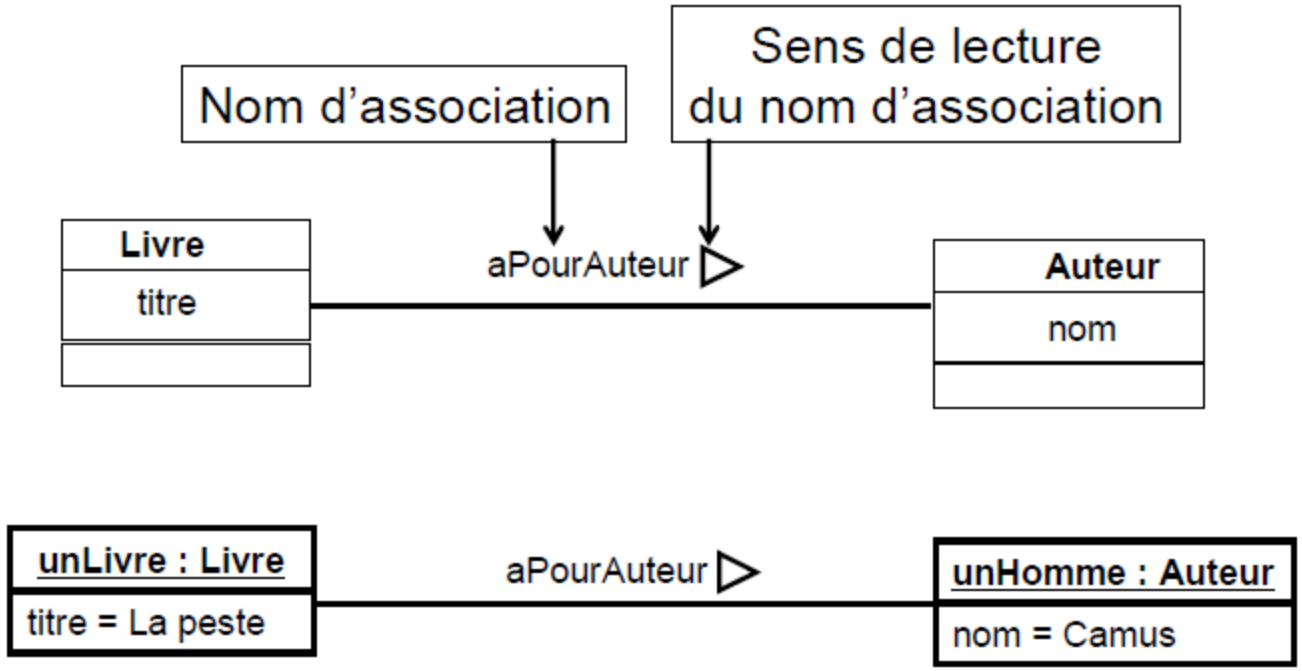
- **Protection des attributs et des opérations: l'encapsulation**
  - Privé (-): l'attribut (ou l'opération) est visible par sa classe seulement
  - Protégé (#): l'attribut (ou l'opération) est visible par sa classe et ses classes dérivées
  - Public (+): l'attribut (ou l'opération) est visible pour toutes les classes

- **Attributs et opérations de niveau Classe:**
  - **Attribut de niveau classe**
    - Attribut dont la valeur est constante pour toutes les instances (les objets)
    - Représentation : nomAttributSouligné
  - **Opération de niveau classe**
    - Opération qui ne s 'applique pas aux objets de la classe
    - Opération qui s 'applique à la classe
    - Représentation : nomOpérationSoulignée



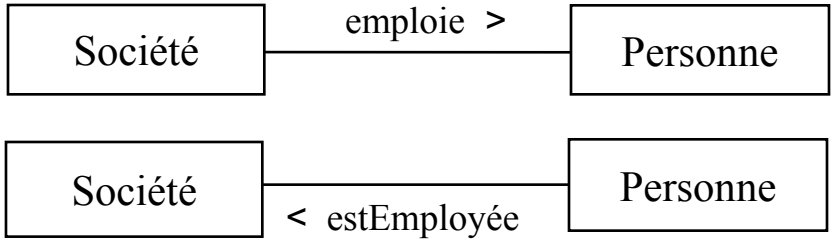
Relations entre classes: Les Associations

- Une Association précise que les objets d'une classe peuvent être reliés aux objets d'une autre classe.



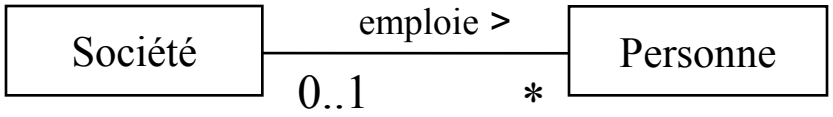
Nom d'association, nom de rôle, multiplicité

Le nom d'une association: le nommage des associations facilite la compréhension des modèles



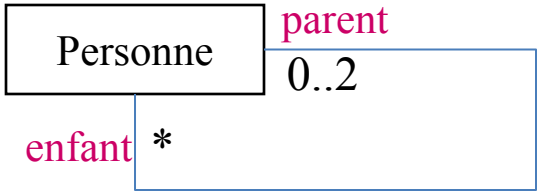
Multiplicité

1	Un et un seul
0..1	Zéro ou un
M..N	De M à N (entiers)
M	M (entier)
* ou 0..*	de zéro à plusieurs
1..*	d'un à plusieurs



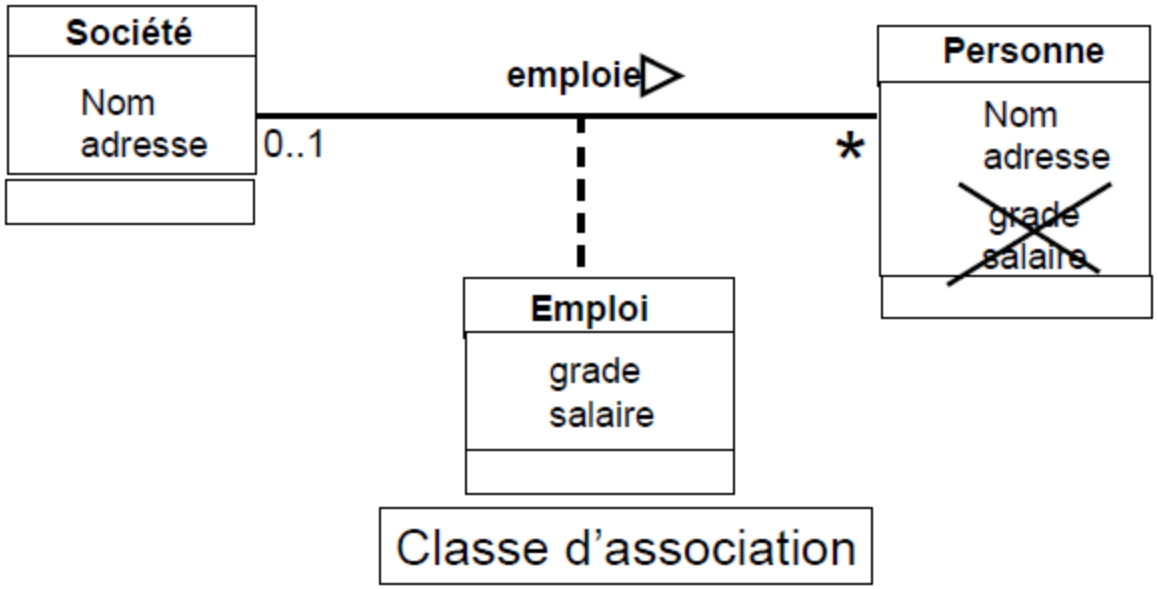
Rôle d'une association

- Décrit comment une classe participe dans une association
- Devient obligatoire lorsque plusieurs associations existent entre 2 classes ou quand l'association est réflexive



Les classes-associations

- Une association peut être représentée par une classe pour ajouter des attributs et des opérations

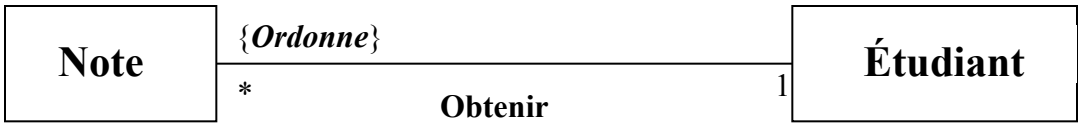


Les contraintes sur les associations

- Les contraintes permettent d'apporter plus de **précisions** à une association
- Peut être définie sur 1 ou plusieurs associations
- Notation **{expression}**

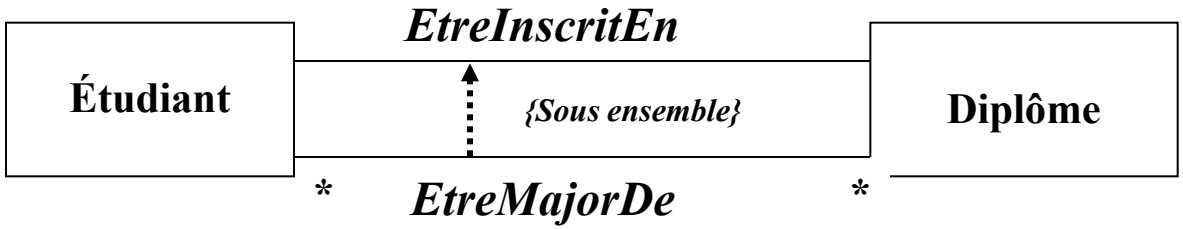
Exemple de contraintes:

- **{ordonnée}**: spécifie qu'il y a une relation d'ordre entre les objets d'une collection, mais ne spécifie pas comment ils sont ordonnés.

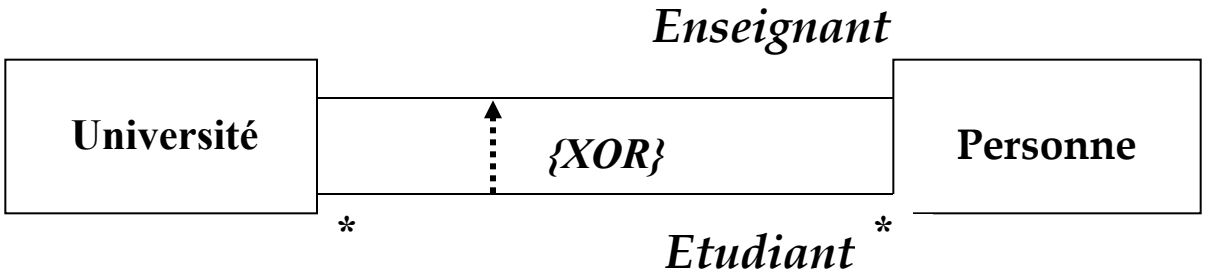


Les contraintes sur les associations

- **{sous-ensemble}:** une relation est incluse dans une autre.

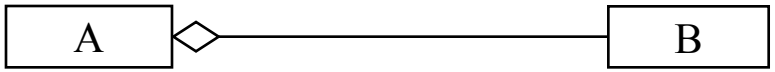


- **{ou-Excusif} ou {XOR}:** une seule association est valide pour un objet donné.

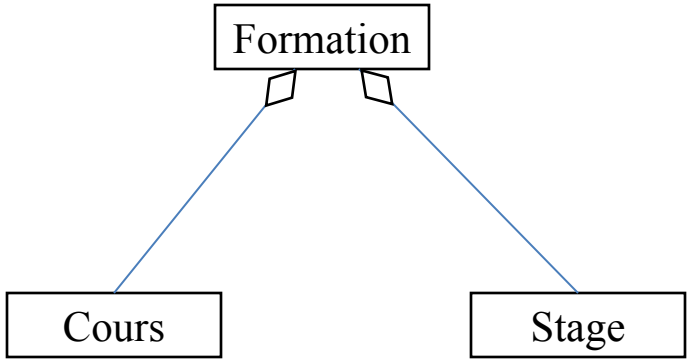


Relations entre classes: Agrégation

- L'agrégation est une association non symétrique, qui exprime un couplage fort et une relation de subordination.
- Elle représente une relation "*fait-partie-de"*
- Représentation:

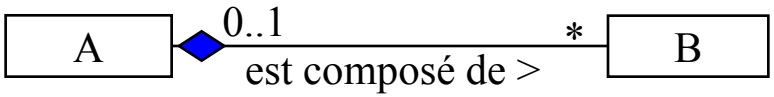


  - Les instances de B peuvent exister indépendamment de celles de A
  - Une instance de B peut éventuellement appartenir à plusieurs instances de A en même temps.

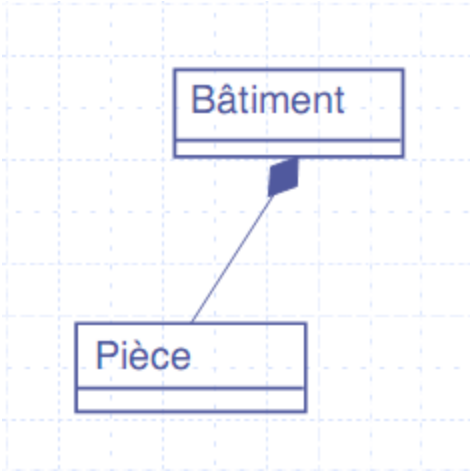


Relations entre classes: Composition

- La composition est une agrégation forte qui porte la sémantique "est composé de"
- Représentation:



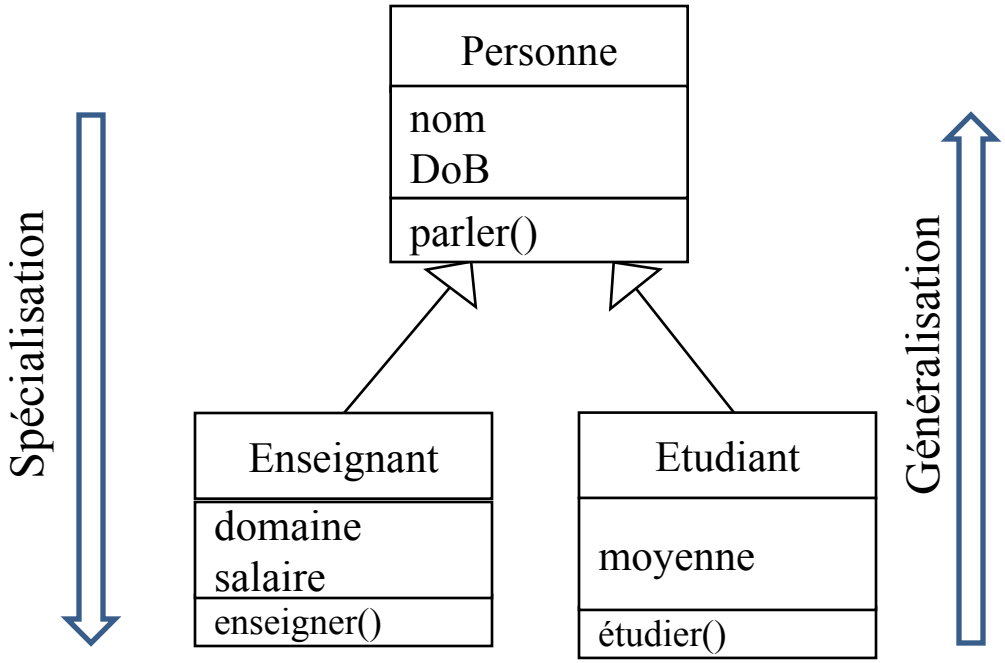
- A est composé de un ou plusieurs B
- Un B ne peut exister tout seul (dans le système)
- Un B n'appartient qu'à un seul A



Relations entre classes: Généralisation/Spécialisation

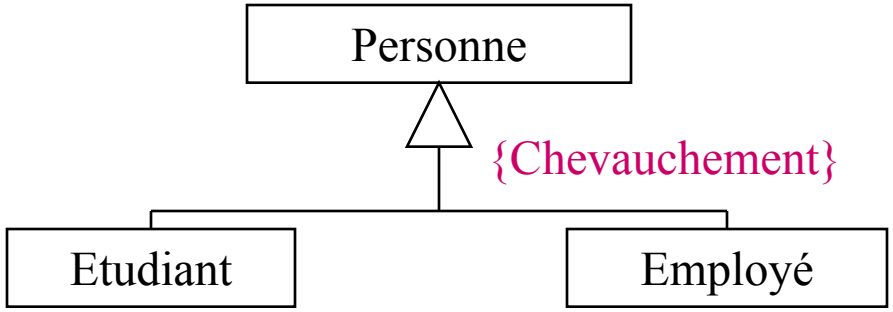
- La généralisation est une relation entre une classe et une ou plusieurs versions affinées de la classe (classes plus spécifiques).
- Permet une factorisation des attributs et des comportements communs à plusieurs classes

Enseignant *et* ***Etudiant*** sont des *spécialisations* (ou sous classes) de la classe ***Personne***.

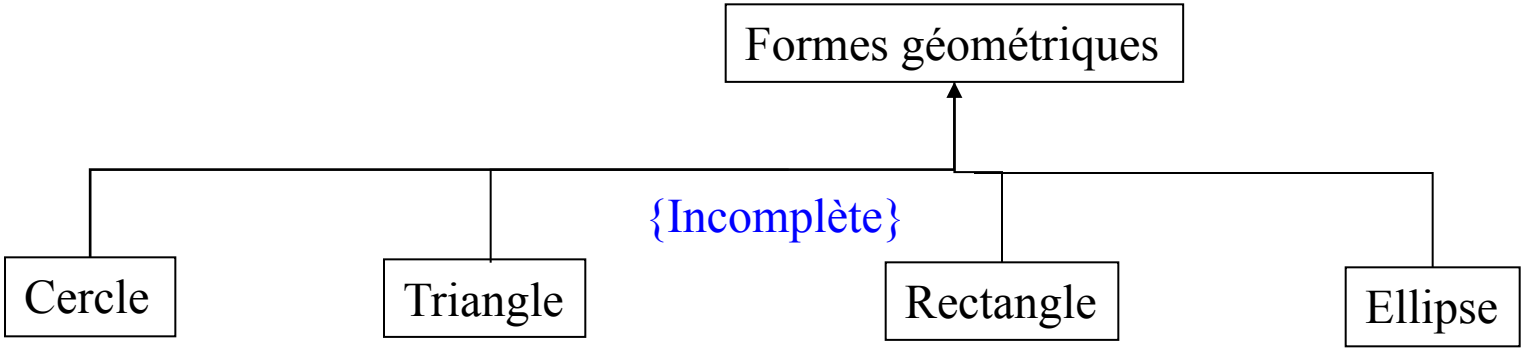


Contraintes sur les Généralisations

- {chevauchement} ou {inclusif} : 2 classes peuvent avoir, parmi leurs instances, des instances identiques,
- {Exclusif} : un objet est au plus une instance d'une seule sous classe

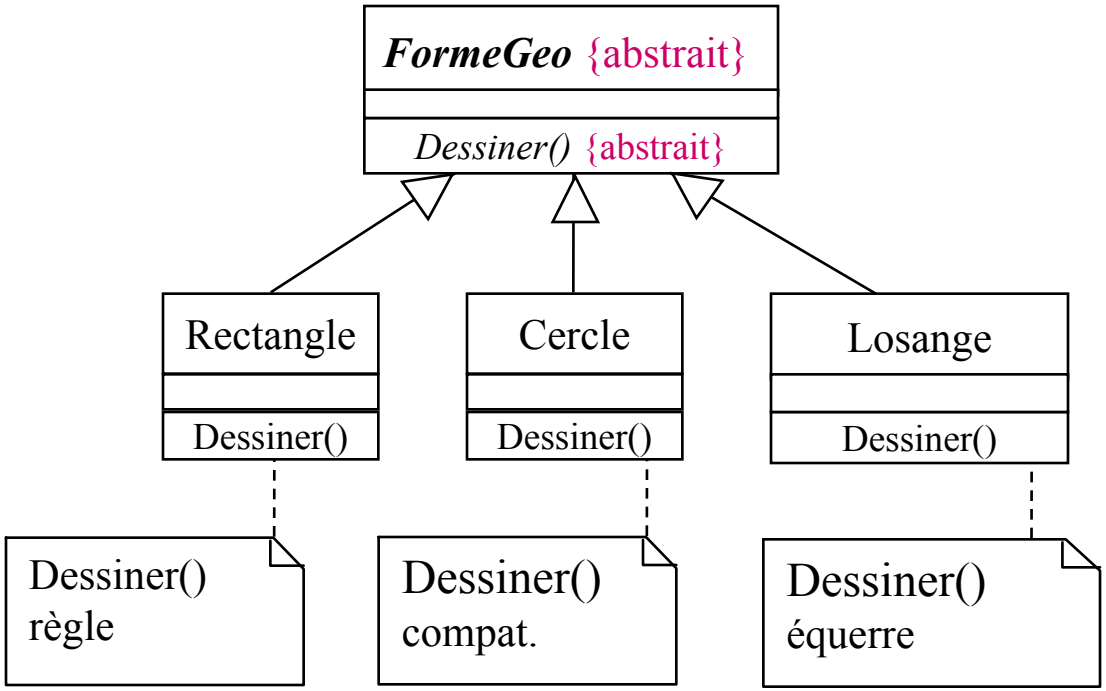


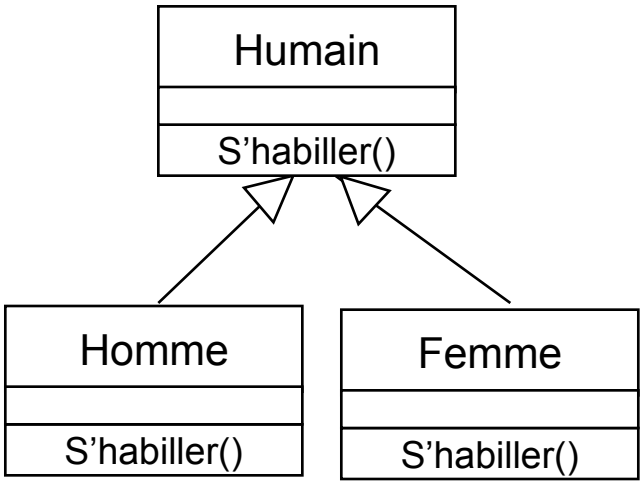
- *{Complète} : existence de toutes les sous-classes*
- *{Incomplète}* : généralisation extensible



Les classes abstraites

- Classe non instanciable,
- Classe de spécification générale: elle permet de déclarer le comportement que les sous-classes doivent implémenter.
- Représentation: nom de classe en *italique*, ou utilisation de <<abstrait>>





Le diagramme d’objets (DOB)

Diagramme d'Objets

Structure statique d'un système, en termes d'**objets** et de **liens** entre ces objets.

Un objet est une *instance* de classe et un lien est une instance de relation.

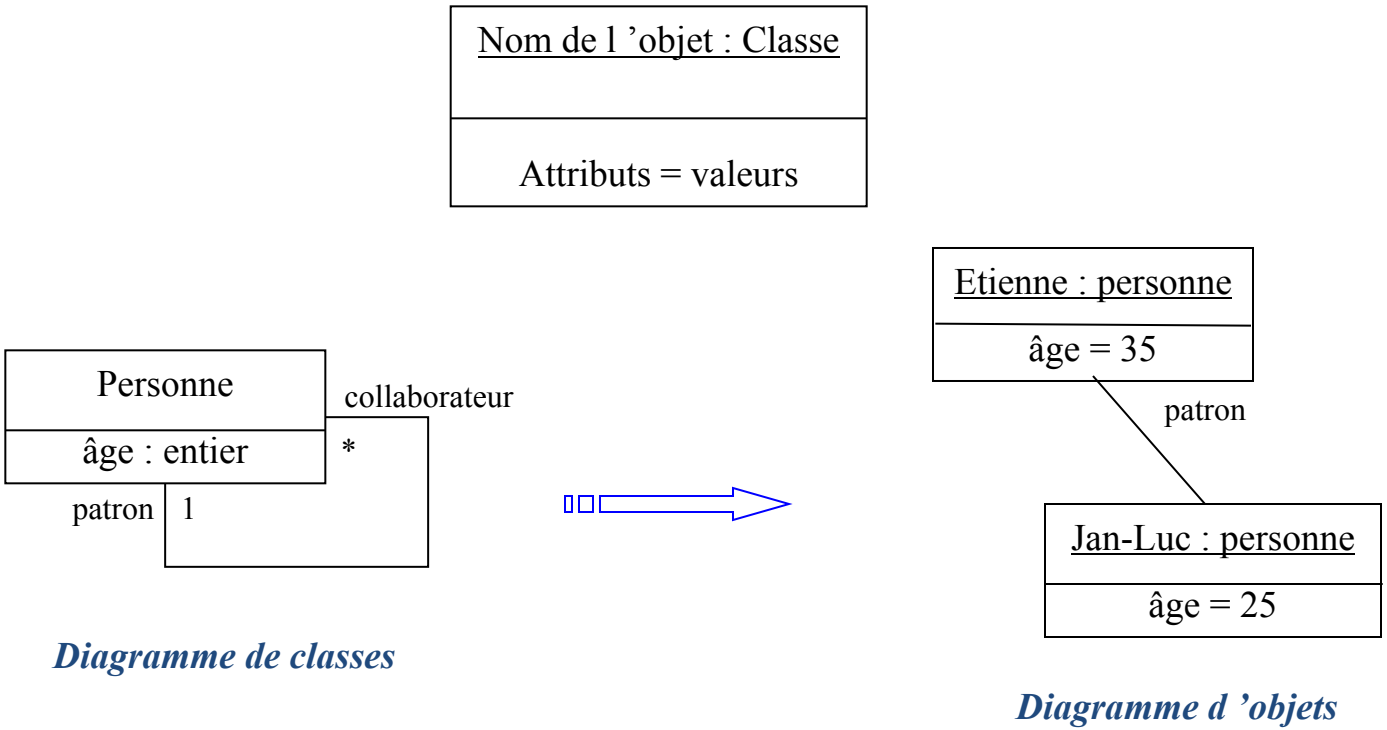
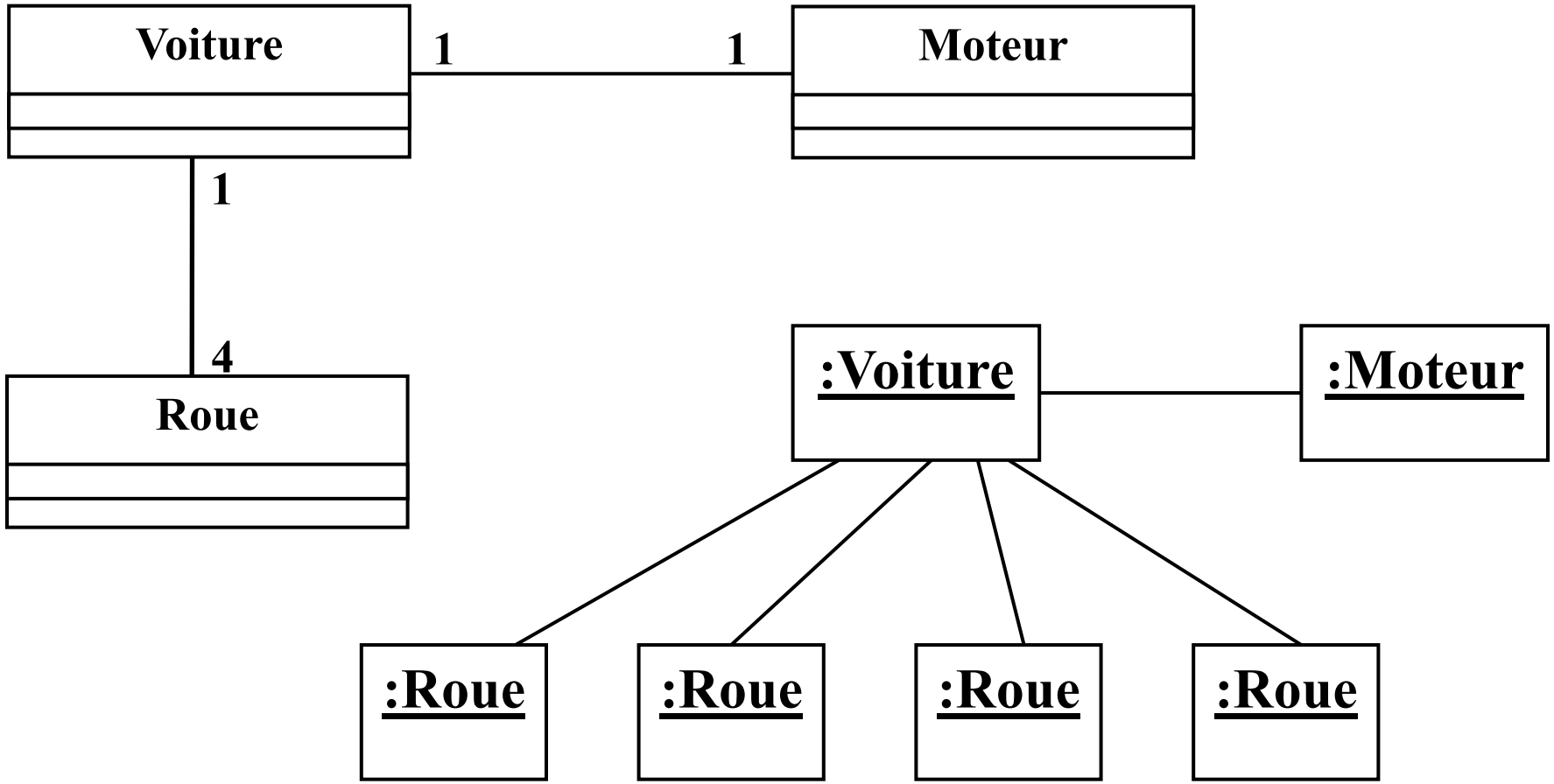
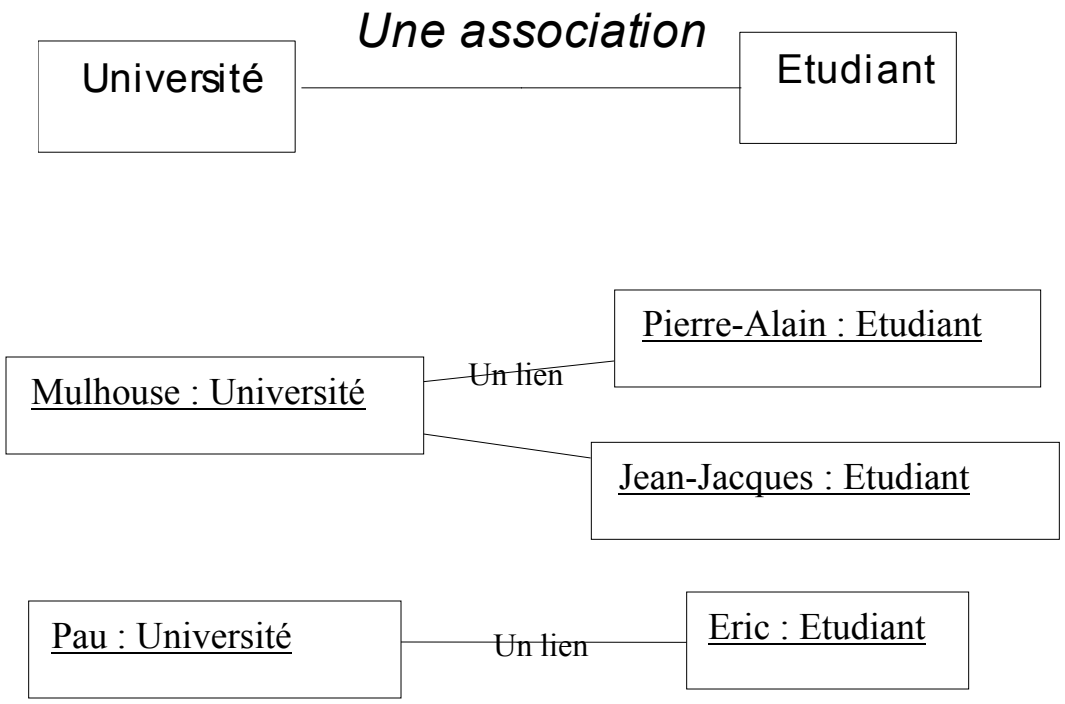


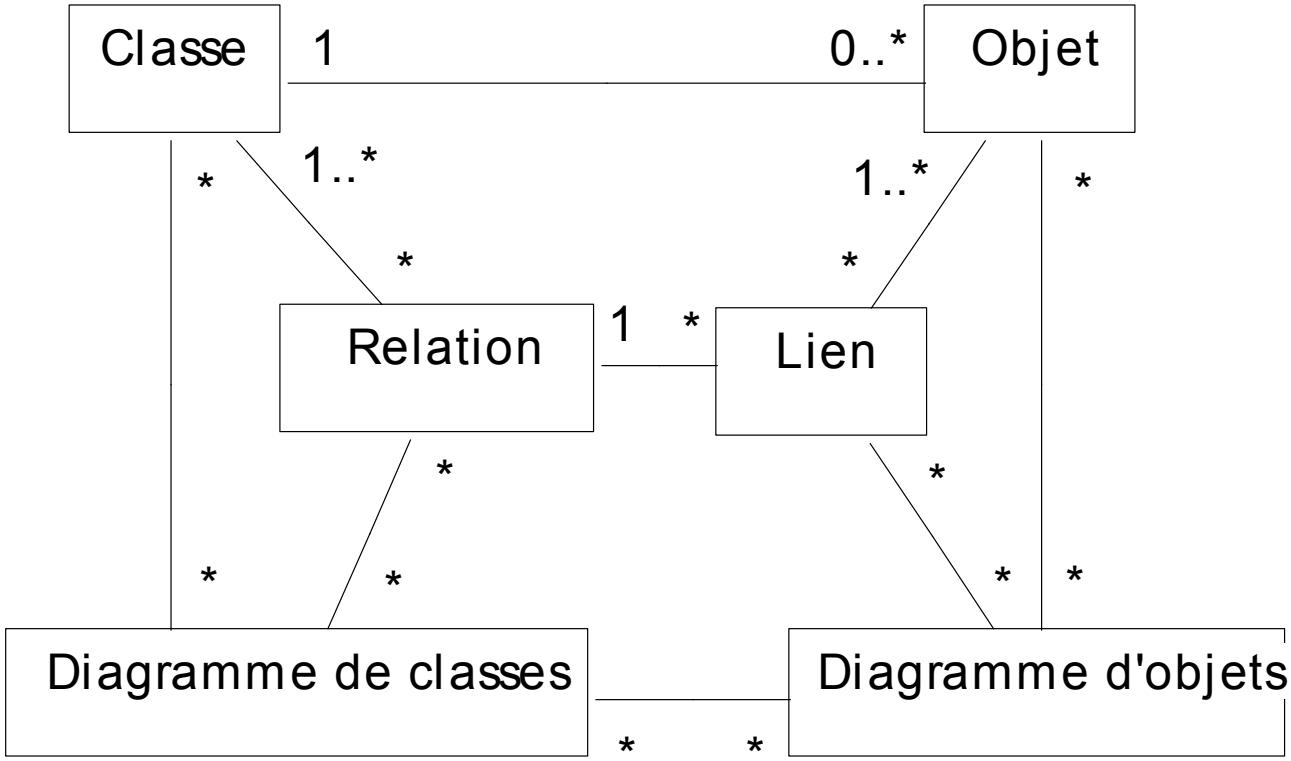
Diagramme de classes

Diagramme d 'objets


Voiture
1
1
Moteur
1
4
Roue
:Voiture
:Moteur
:Roue
:Roue
:Roue
:Roue


Une association
Université
Etudiant
Pierre-Alain : Etudiant
Mulhouse : Université
Un lien
Jean-Jacques : Etudiant
Pau : Université
Un lien
Eric : Etudiant

Correspondances entre DCL et DOB diagrammes



Correspondances entre DCL et DOB diagrammes

- Chaque objet est instance d'une classe
- Chaque lien est instance d'une relation
- Les liens relient les objets, les relations relient les classes
- Un lien entre deux objets implique une relation entre les classes des deux objets

Les diagrammes de classes