

Université de Msila – Département  
d'Informatique

# Génie logiciel (GL)

(3ème année SI&ISIL)

Dr BOUNIF M-E

- 6. Le diagramme Classe
- **1. Les liens entre classes**
- **Les associations entre les classes**

Une association est une relation entre deux classes (association binaire) ou plus (association n-aire), qui décrit les connexions structurelles entre leurs instances. Une association indique donc qu'il peut y avoir des liens entre des instances des classes associées. Dans la pratique, la très grande majorité des associations sont binaires et les associations quaternaires et au-delà ne sont quasiment jamais utilisées.

Une association porte un nom. Comme pour une classe, ce nom est significatif des occurrences de l'association.

## 2. La représentation des associations entre les classes

La représentation graphique d'une association binaire consiste en un trait continu entre les deux classes dont elle associe les instances. Ces classes se situent aux extrémités de l'association. La figure 6.12 illustre la représentation d'une association binaire. Le nom de l'association est indiqué au-dessus du trait.

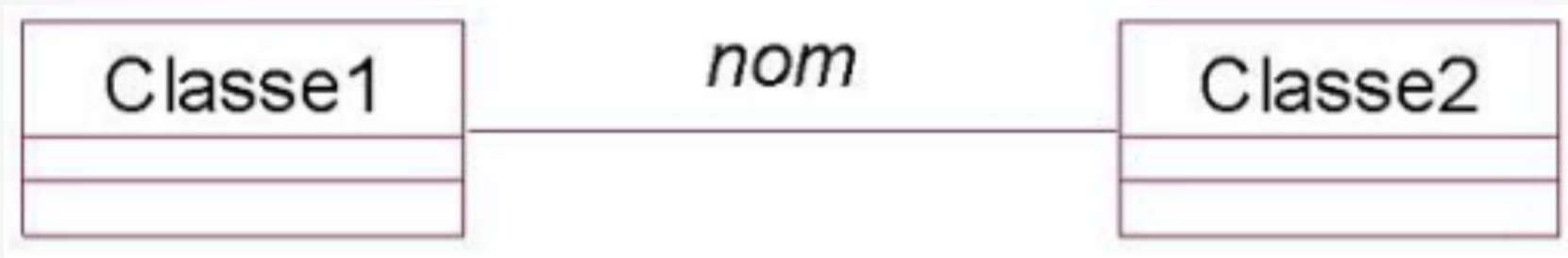


Figure 6.12 Association binaire entre classes

Il est possible de faire précéder le nom d'une association du signe < ou de le faire suivre par le signe > pour indiquer le sens de lecture du nom vis à vis du nom des classes. Si l'association est située sur un axe vertical, il est possible de faire précéder le nom par ^ ou v.

Chaque extrémité d'une association peut également être nommée. Ce nom est significatif du rôle que jouent les instances de la classe correspondante dans l'association. Un rôle a la même nature qu'un attribut dont le type serait la classe située à l'autre extrémité. Par conséquent, il peut être public ou encapsulé de façon privée, protégée ou pour le paquetage. Lorsque les rôles sont précisés, il n'est souvent plus nécessaire d'indiquer le nom de l'association car celui-ci est fréquemment le même que l'un des rôles.

La figure 6.13 illustre la représentation d'une association binaire en montrant les rôles.

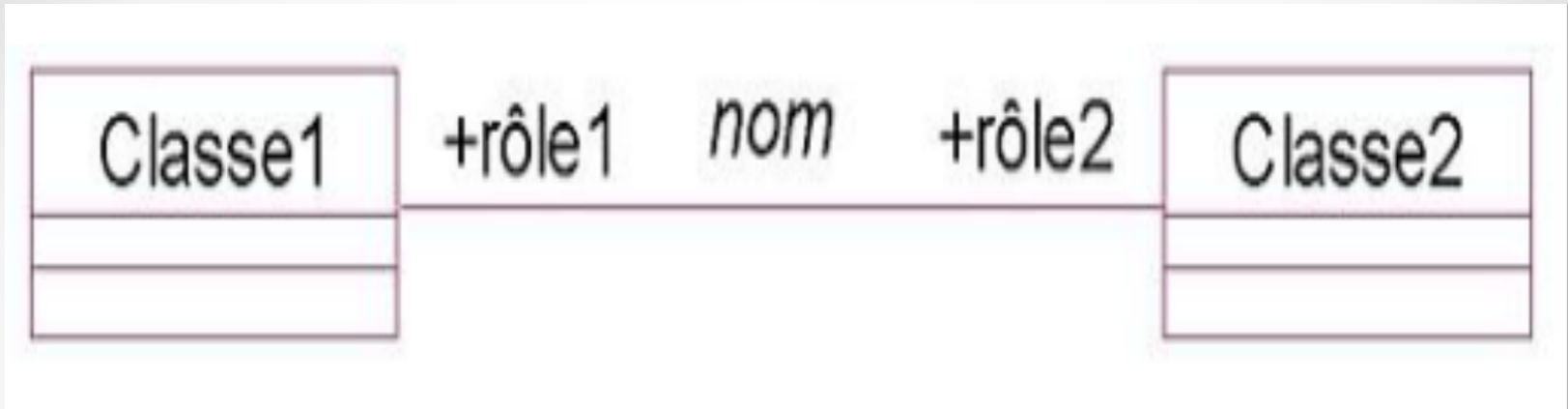


Figure 6.13 Rôles d'une association binaire

Exemple

La figure 6.14 illustre la représentation graphique des associations en exemple.

Pour ces associations, soit le nom de l'association, soit ses rôles ont été indiqués.

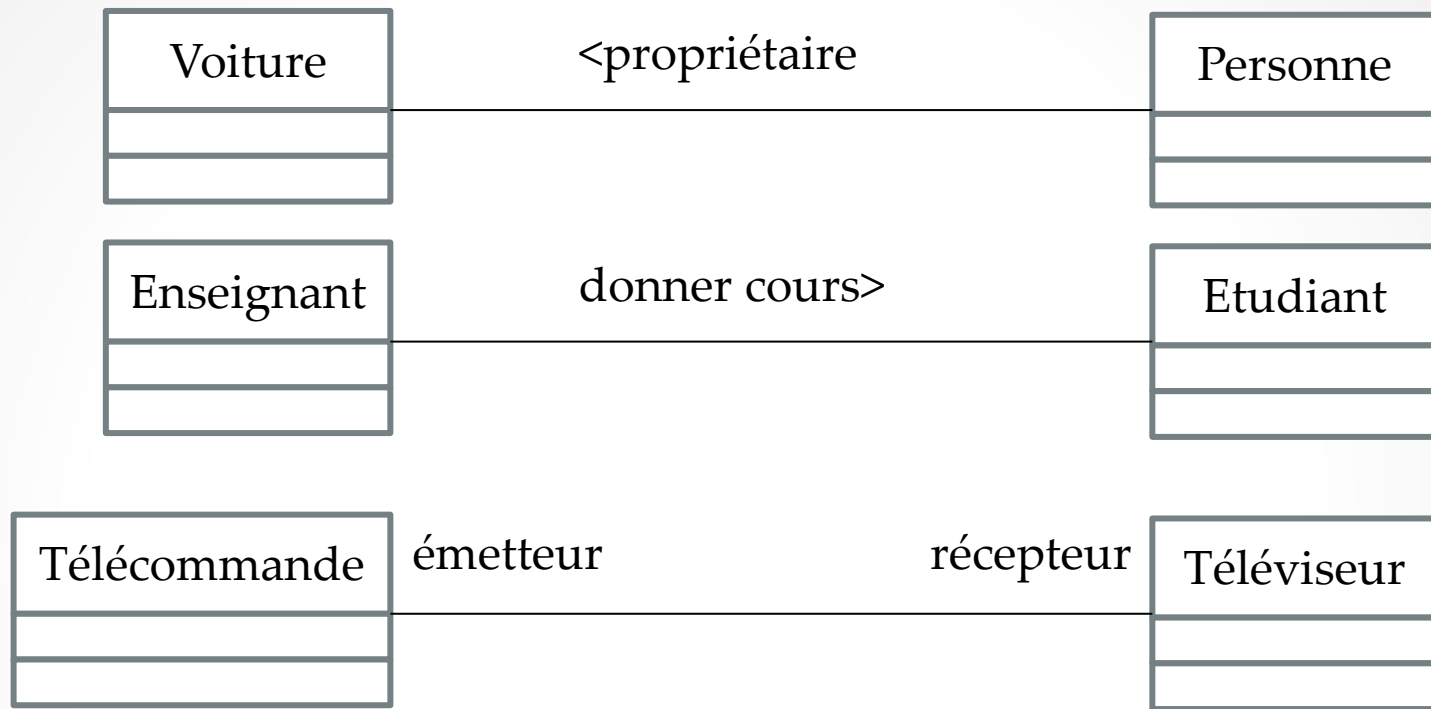


Figure 6.14 Exemples d'associations binaires

La représentation graphique d'une association ternaire et au-delà consiste en un losange qui relie les différentes classes. La figure 6.15 illustre la représentation d'une association ternaire.

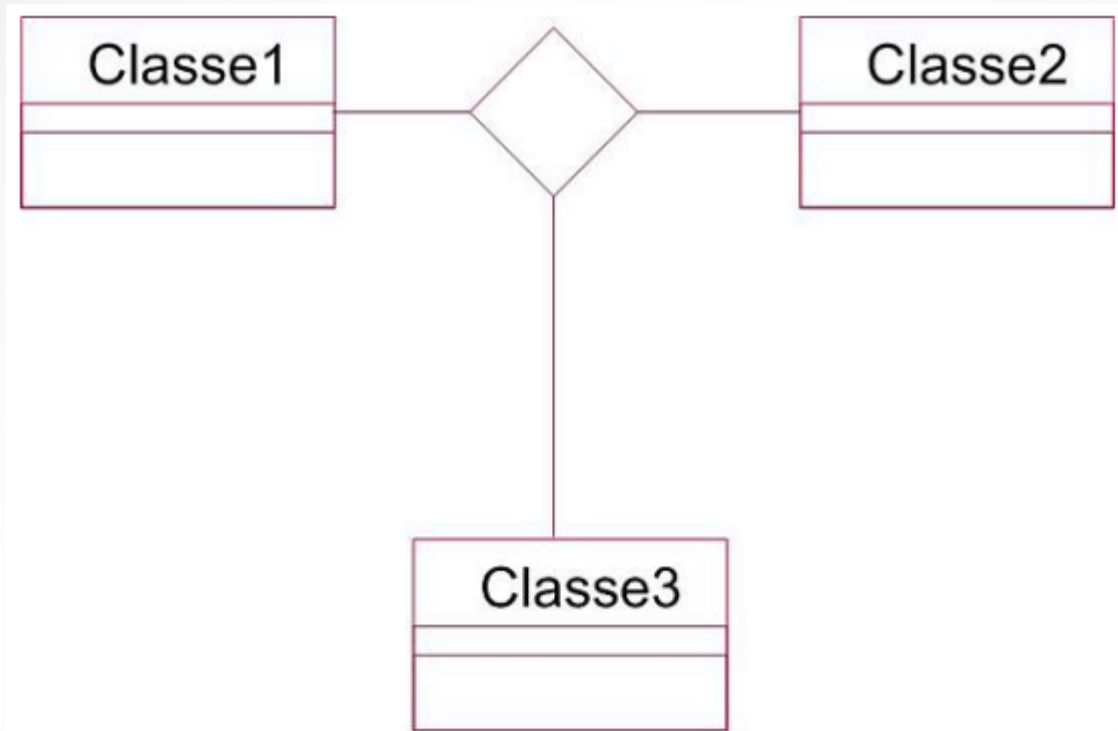


Figure 6.15 Association ternaire entre classes

Exemple

L'association qui relie les classes Étudiant, Enseignant et Module est illustrée à la figure 6.16. Chacune de ses Occurrences constitue un triplet (prof,etud , GL ).

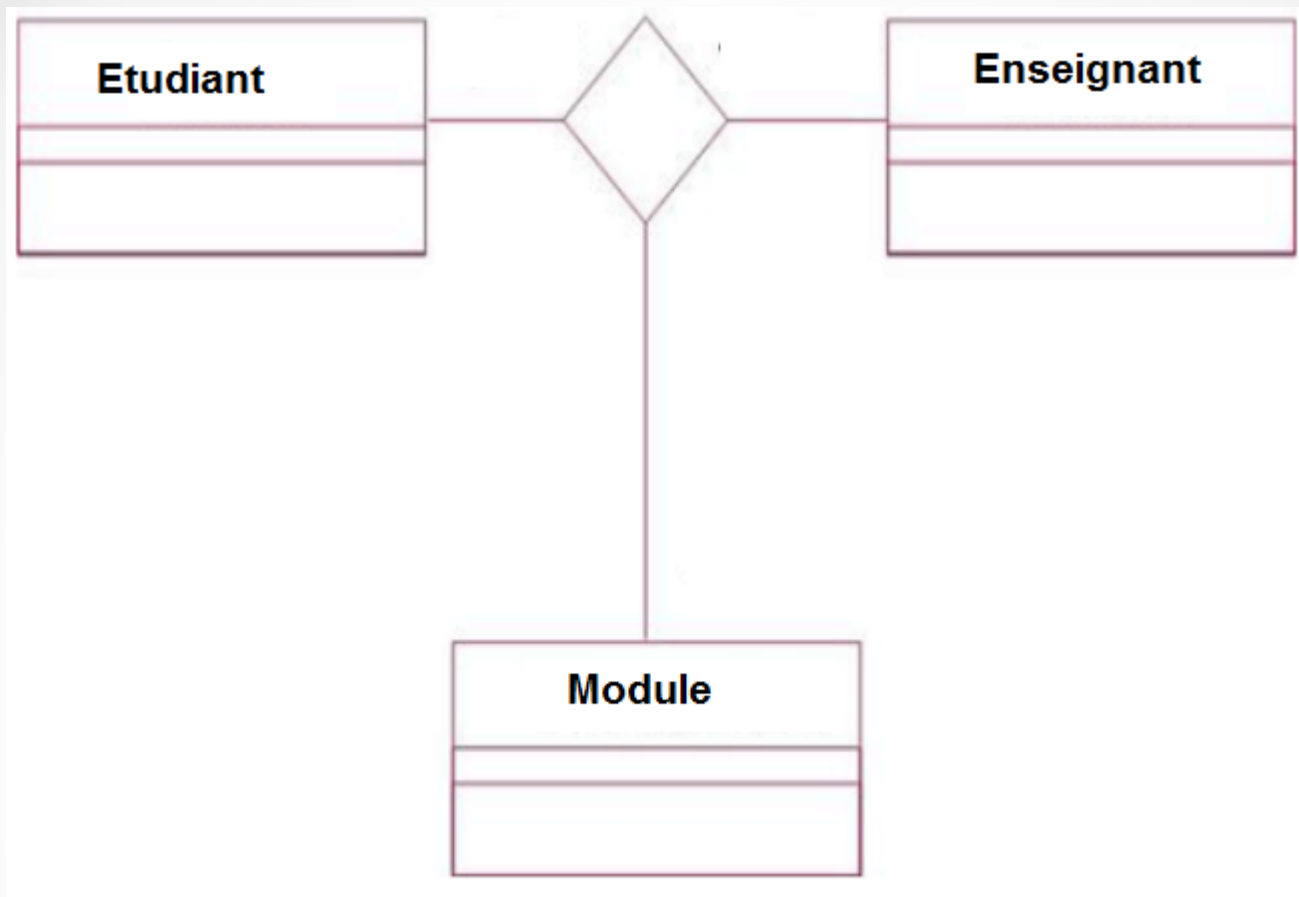


Figure 6.16 Exemples d'association ternaire



### 3. La cardinalité des associations

La cardinalité située à une extrémité d'une association indique à combien d'instances de la classe située à cette extrémité, une instance de la classe située à l'autre extrémité est liée.

Il est possible de spécifier, à une extrémité d'une association, la cardinalité minimale et la cardinalité maximale pour indiquer un intervalle de valeurs auquel doit toujours appartenir la cardinalité. La syntaxe de spécification des cardinalités minimales et maximales est décrite dans le tableau ci-après.

Spécification	Cardinalités
0..1	zéro ou une fois
1	une et une seule fois
*	de zéro à plusieurs fois
1..*	de un à plusieurs fois
M..N	entre M et N fois
N	N fois

En l'absence de spécification explicite des cardinalités minimales et maximales, celles-ci valent 1.

## 4. Navigation

Les associations ont par défaut une navigation bidirectionnelle, c'est à dire qu'il est possible de déterminer les liens de l'association depuis une instance de chaque classe d'origine. Une navigation bidirectionnelle est plus complexe à réaliser par les développeurs ; il convient de l'éviter dans la mesure du possible.

Spécifier le seul sens de navigation utile lors des phases de la modélisation proche du passage au développement se fait en dessinant l'association sous forme d'une flèche.

Exemple

Dans le cadre particulier d'un élevage de chevaux, il est utile de connaître les chevaux que cet élevage possède mais le contraire n'est pas nécessaire.

La figure 6.17 illustre l'association résultant avec le sens de navigation adapté.

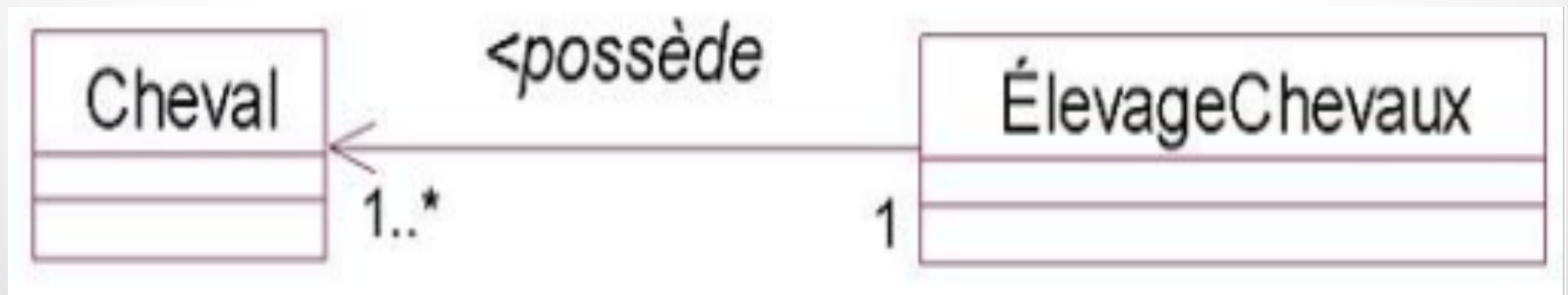


Figure 6.17 Exemple de navigation

## 5. Associer une classe avec elle-même

La même classe peut se trouver à chaque extrémité d'une association. Il s'agit alors d'une association réflexive, reliant entre elles les instances d'une même classe.

Dans ce cas, il est préférable de nommer le rôle joué par la classe à chaque extrémité de l'association.

Comme nous le verrons dans les exemples, une association réflexive sert principalement à décrire au sein de l'ensemble des instances d'une classe :

- des groupes d'instances ;
- ou une hiérarchie au sein des instances.

Pour les experts, dans le premier cas, il s'agit d'une association représentant une relation d'équivalence et dans le second, d'une association représentant une relation d'ordre.

Exemple

Pour pouvoir passer les épreuves de sélection d'un concours hippique international, un cheval doit avoir gagné d'autres concours préalables. Il est donc possible de créer une association entre un concours et ses concours préalables. La figure 6.18 illustre cette association. Cette association crée une hiérarchie au sein des concours.

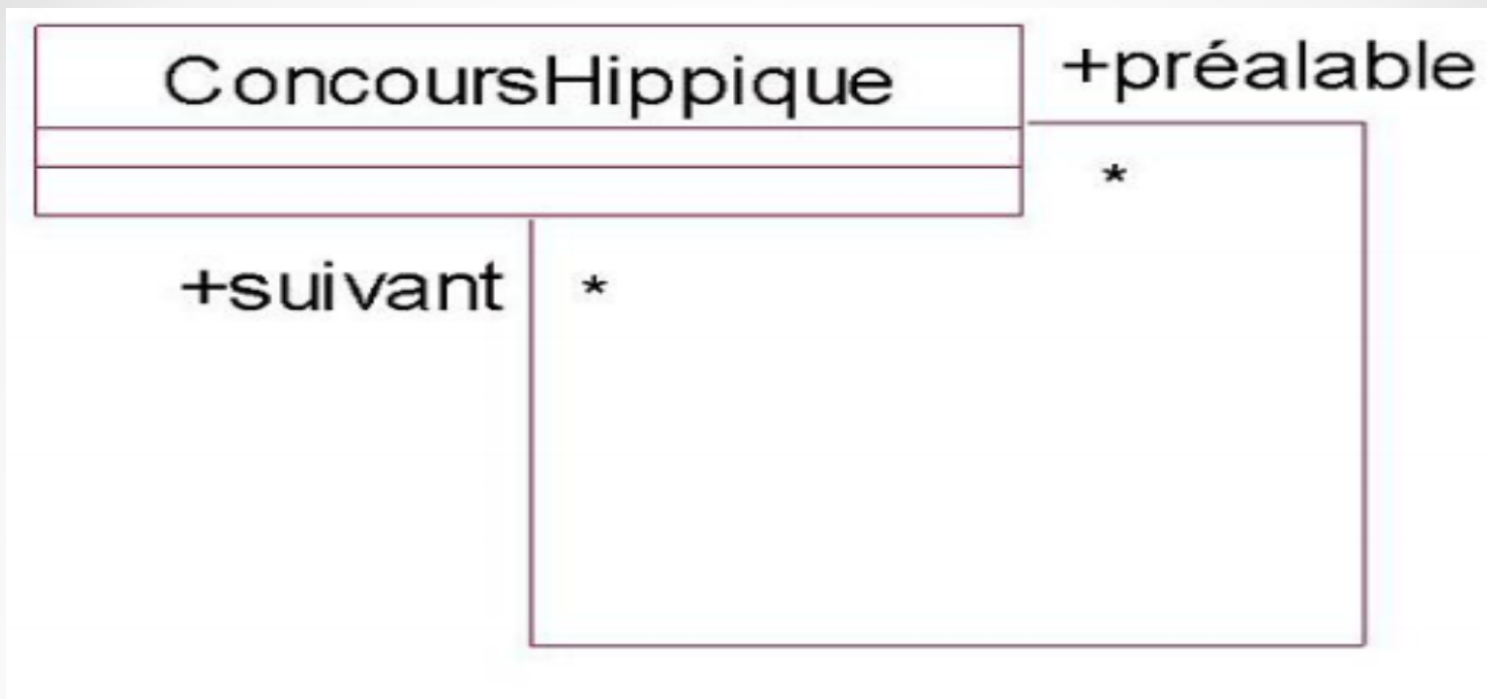


Figure 6.18 Association réflexive entre concours hippiques

### Exemple

La figure 6.19 illustre l'association entre les chevaux qui se trouvent dans le même élevage. Cette association crée des groupes au sein de l'ensemble des instances de la classe Cheval ,chaque groupe correspondant à un élevage.

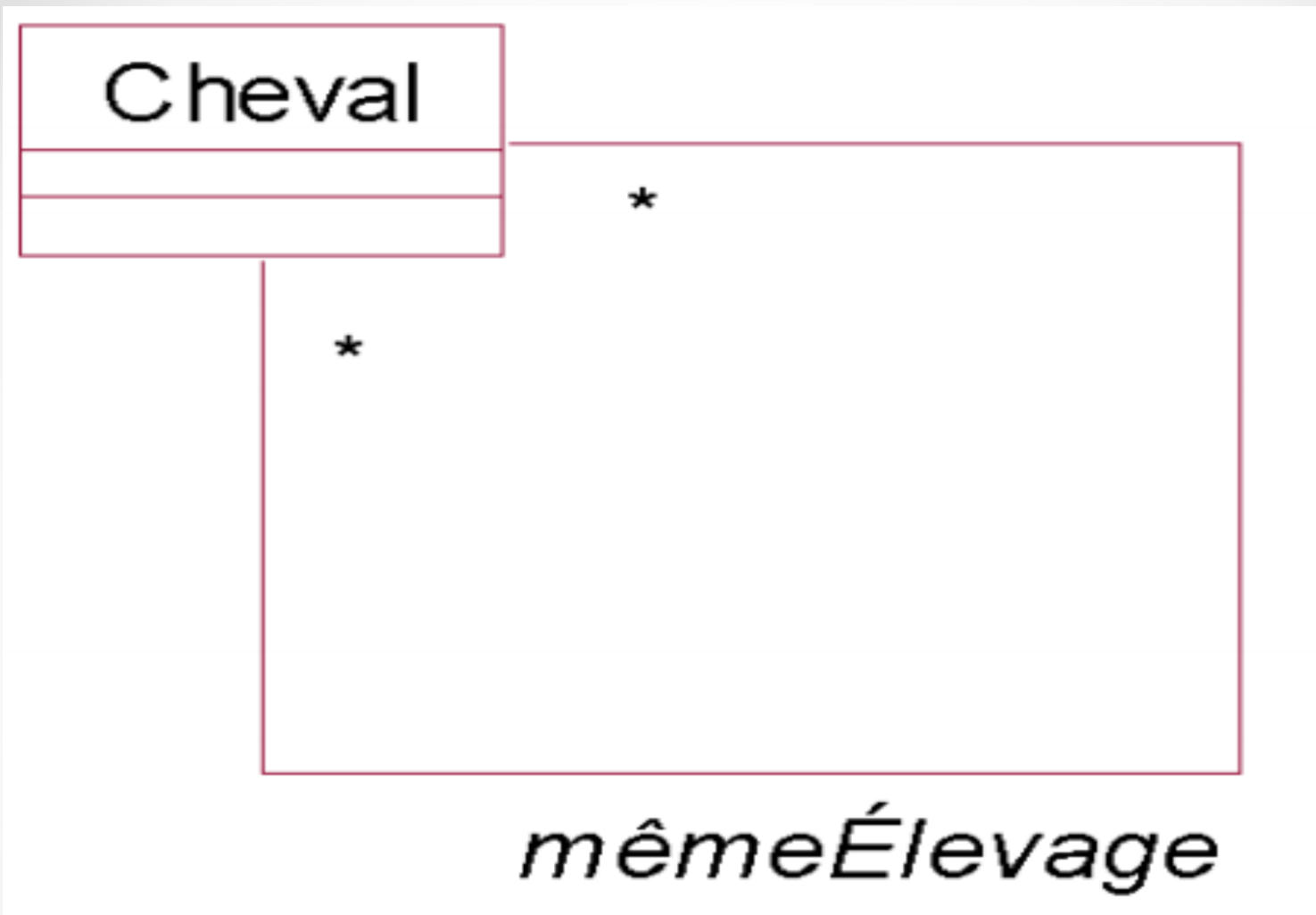


Figure 6.19 -  
Association réflexive qui relie les chevaux d'un même élevage

## 6. Les classes associations

Les liens entre les instances des classes peuvent porter des informations. Celles-ci sont spécifiques à chaque lien. Dans ce cas, l'association qui décrit de tels liens reçoit le statut de classe, dont les instances sont des occurrences de l'association. Comme toute autre classe, une telle classe peut être dotée d'attributs, d'opérations, et être reliée au travers d'associations à d'autres classes. La figure 6.20 représente graphiquement une classe association. Celle-ci est reliée à l'association par un trait pointillé.

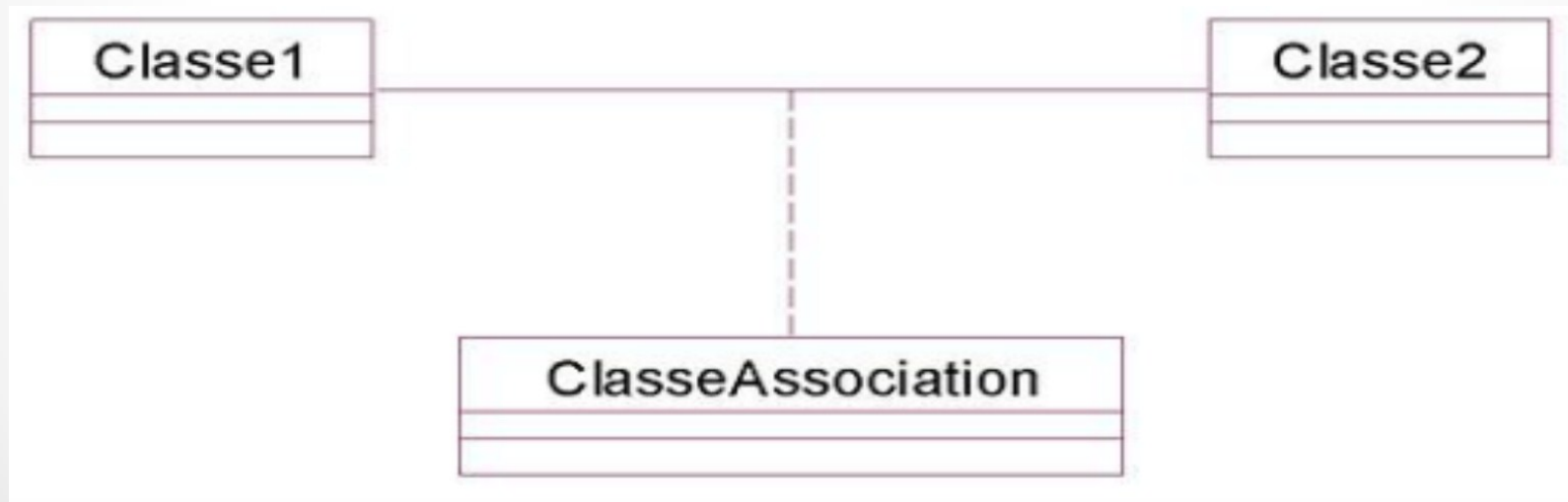


Figure 6.20 Représentation graphique d'une classe association

## Exemple

Quand un client achète un produit, il convient de spécifier la quantité de produits acquis par une classe association, ici la classe Acquisition (voir figure 6.21).

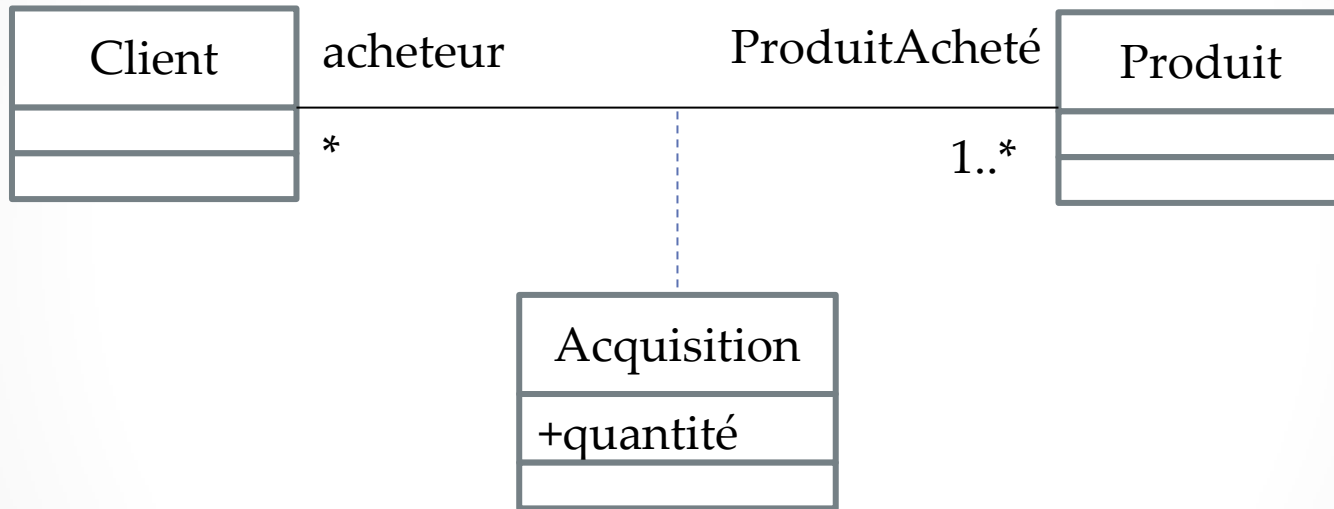


Figure 6.21 Classe association Acquisition

## 7. Les objets composés

La composition forte est la forme de composition telle que les composants sont une partie de l'objet composé. Chaque composant ne peut ainsi être partagé entre plusieurs objets composés. La cardinalité maximale, au niveau de l'objet composé, est donc obligatoirement de un.

La suppression de l'objet composé entraîne la suppression de ces composants.

Exemple Un cheval est composé d'un cerveau. Le cerveau n'est pas partagé. La mort du cheval entraîne la mort de son cerveau. Il s'agit donc d'une association de composition. Celle-ci est illustrée à la figure 6.22.



Figure 6.31 Association de composition entre un cheval et son cerveau



## **b. La composition faible ou agrégation**

La composition faible, appelée plus couramment agrégation, impose beaucoup moins de contraintes aux composants que la composition forte. Dans le cas de l'agrégation, les composants peuvent être partagés par plusieurs composés (de la même association d'agrégation ou de plusieurs associations distinctes d'agrégation) et la destruction du composé ne conduit pas à la destruction des composants.

L'agrégation se rencontre plus fréquemment que la composition. Lors d'une première phase de modélisation, il est possible d'utiliser seulement l'agrégation puis, plus tard, de déterminer quelles associations d'agrégation sont des associations de composition.

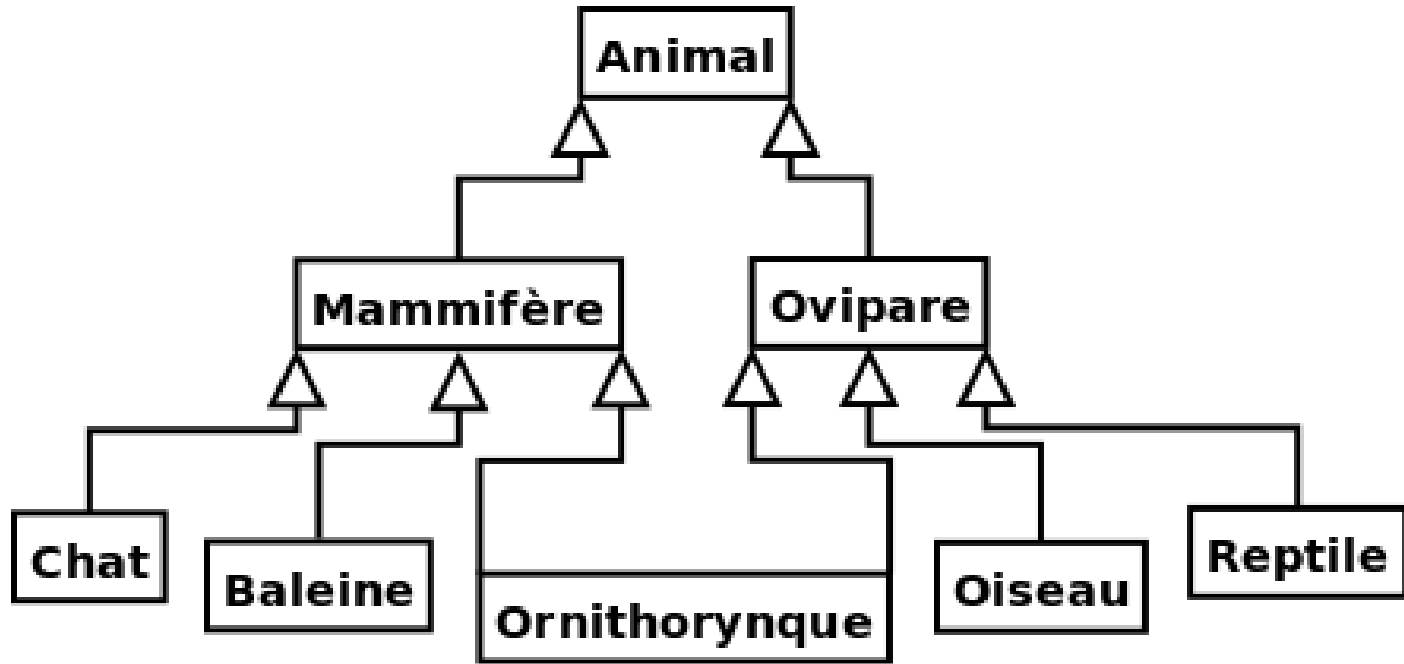


### c. Différences entre composition et agrégation

Le tableau suivant résume l'ensemble des différences entre agrégation et composition.

	<b>Agrégation</b>	<b>Composition</b>
Représentation	losange transparent	losange noir
Partage des composants par plusieurs associations	oui	non
Destruction des composants lors de la destruction du composé	non	oui
Cardinalité au niveau du composé	quelconque	0..1 ou 1

- **L'héritage** La généralisation décrit une relation entre une classe générale (sur classe) et une classe spécialisée (sous-classe). La classe spécialisée est intégralement cohérente avec la classe de base, mais comporte des informations supplémentaires (attributs, opérations, associations). Un objet de la classe spécialisée peut être utilisé partout où un objet de la classe de base est autorisé.
- Dans le langage UML, ainsi que dans la plupart des langages objet, cette relation de généralisation se traduit par le concept d'héritage. On parle également de relation d'héritage. Ainsi, l'héritage permet la classification des objets (figure 6.32).
- Le symbole utilisé pour la relation d'héritage ou de généralisation est une flèche avec un trait plein dont la pointe est un triangle fermé désignant le cas le plus général



- Figure 6.32 Hiérarchie de classes

- Les propriétés principales de l'héritage sont :
- la classe enfant possède toutes les caractéristiques de ses classes parents, mais elle ne peut accéder aux caractéristiques privées de cette dernière ;
- une classe enfant peut redéfinir (même signature) une ou plusieurs méthodes de la classe parent. Sauf indication contraire, un objet utilise les opérations les plus spécialisées dans la hiérarchie des classes ;
- toutes les associations de la classe parent s'appliquent aux classes dérivées ;
- une instance d'une classe peut être utilisée partout où une instance de sa classe parent est attendue. Par exemple, en se basant sur le diagramme de la figure 6.32, toute opération acceptant un objet d'une classe *Animal* doit accepter un objet de la classe *Chat* ;
- une classe peut avoir plusieurs parents, on parle alors d'héritage multiple ( la classe *Ornithorynque* de la figure 6.32). Le langage C++ est un des langages objet permettant son implémentation effective, le langage Java ne le permet pas.

## Expression de contraintes sur la relation d'héritage

UML offre quatre contraintes sur la relation d'héritage entre une surclasse et ses sous classes :

- La contrainte {incomplete} signifie que l'ensemble des sous classes est incomplet et qu'il ne couvre pas la surclasse ou encore que l'ensemble des instances des sous classes est un sous ensemble de l'ensemble des instances de la surclasse.
- La contrainte {complete} signifie au contraire que l'ensemble des sous classes est complet et qu'il couvre la surclasse.
- La contrainte {disjoint} signifie que les sous classes n'ont aucune instance en commun.
- La contrainte {overlapping} signifie que les sous classes peuvent avoir une ou plusieurs instances en commun.