UML

Diagramme des classes

F.Roda 2009

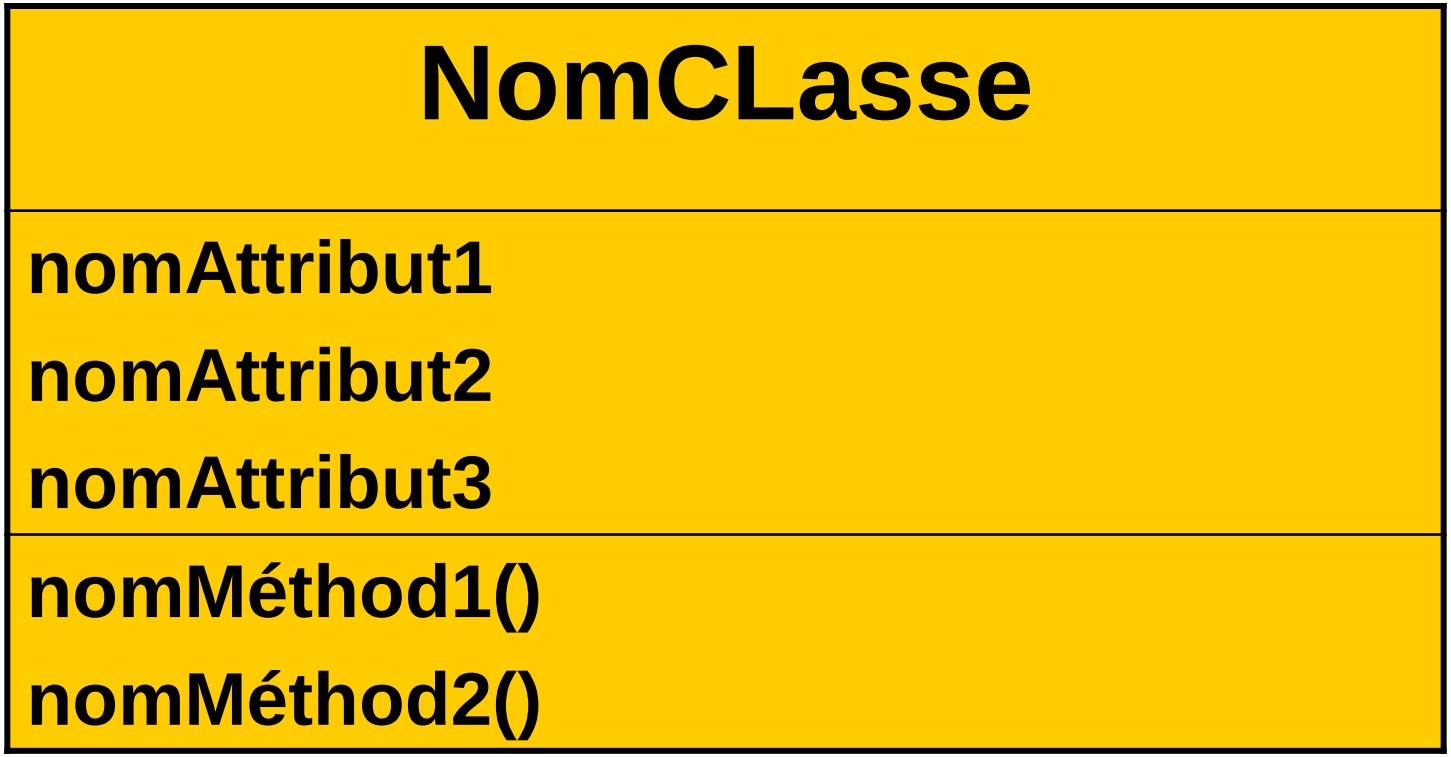
Introduction

- Le diagramme des classes contient principalement des classes
- Une classe contient des attributs et des opérations
- Le diagramme des classes n'indique pas comment utiliser les opérations
- C'est une description statique

Définition

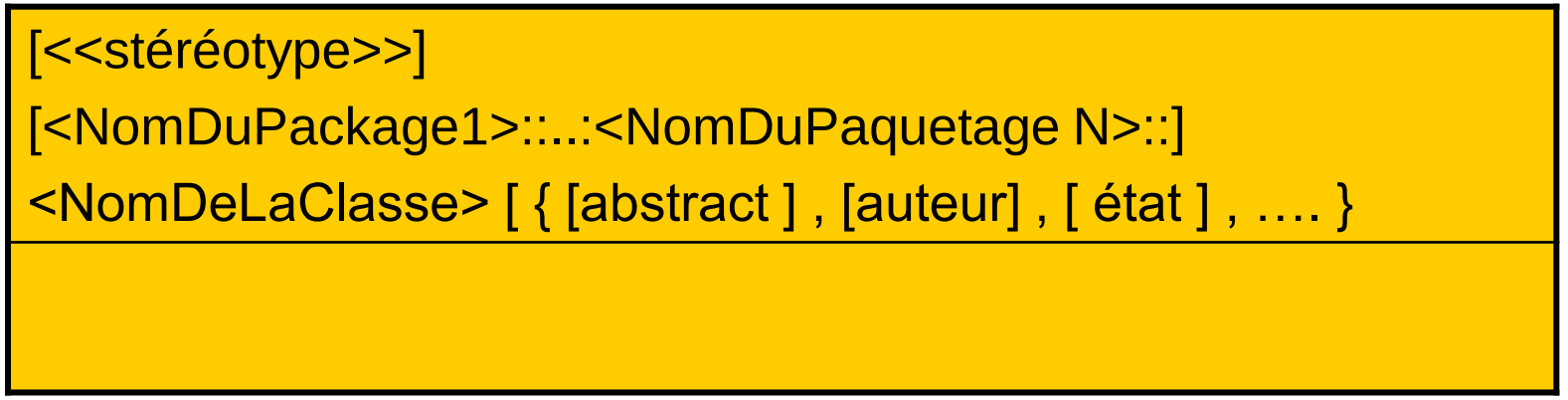
- Une classe est une description d'un ensemble d'objets ayant une sémantique, des attributs, des méthodes et des relations en commun.
- Un objet est une instance d'une classe

Représentation simplifiée d'une classe



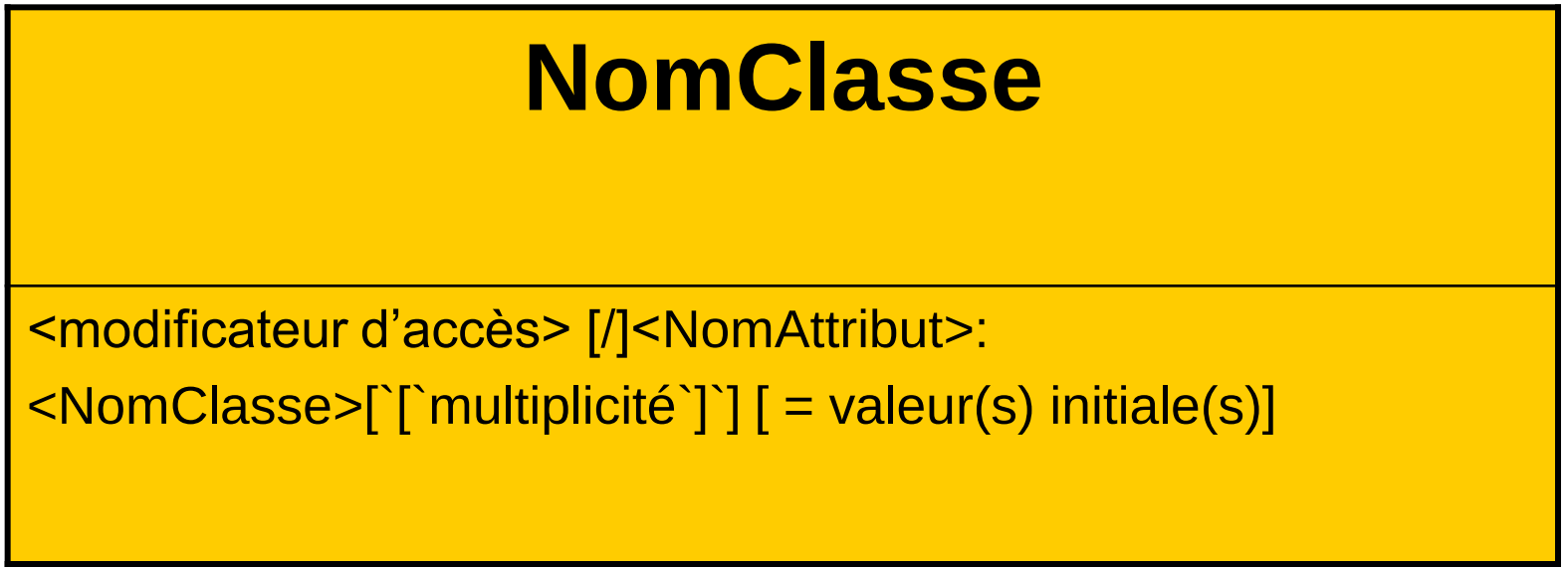
Nom de la classe

- Le nom d'une classe est au singulier
- Il est constitué d'un nom commun
- Ce nom est significatif de l'ensemble des objets constituant la classe
- Il représente la nature des instances d'une classe



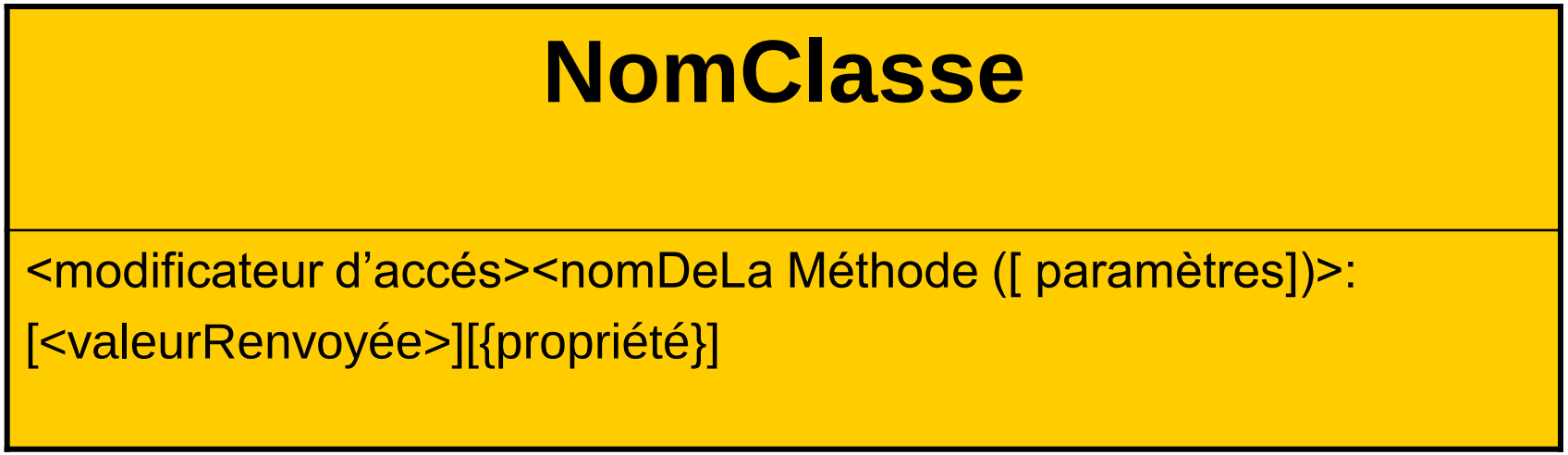
Attributs

- Ceux-ci contiennent l'information portée par un objet.
- L'ensemble des attributs forme la structure de l'objet



Méthodes

- Celles-ci correspondent aux services offerts par l'objet
- Elles peuvent modifier la valeur des attributs
- L'ensemble des méthodes forme le comportement de l'objet

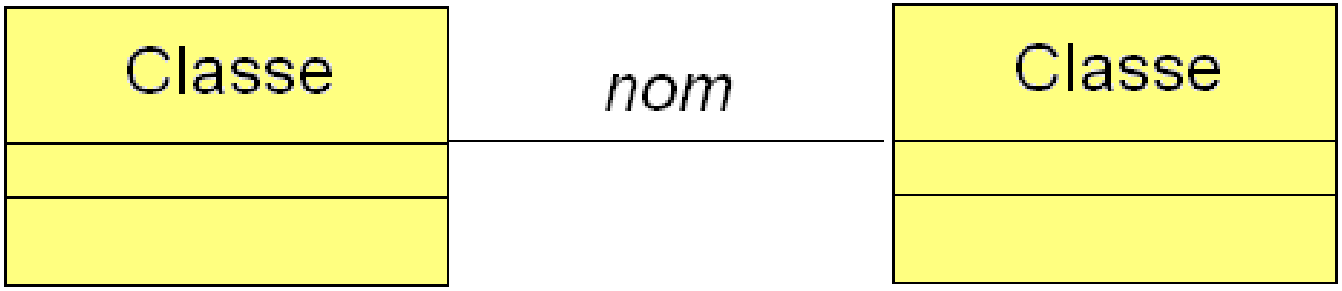


Encapsulation

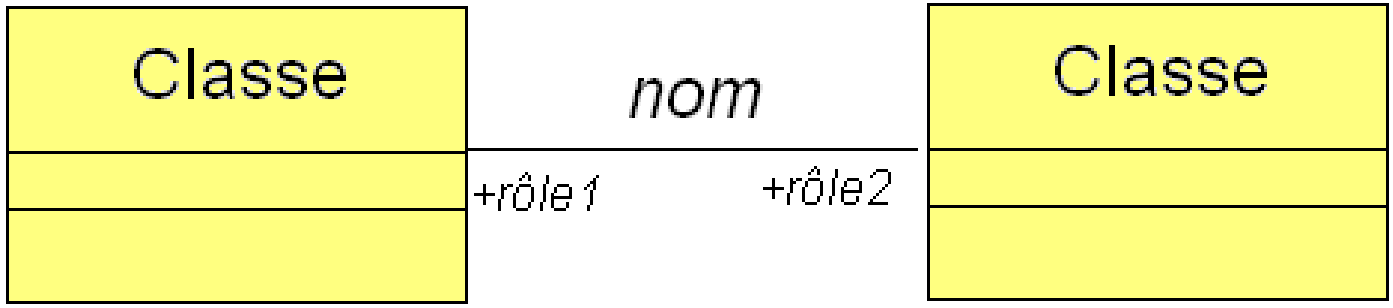
- Elle permet de définir les droits d'accès aux propriétés d'une classe.
- UML définit quatre niveaux d'encapsulation d'une propriété d'une classe

public	+	Élément non encapsulé visible par tous
protégé	#	Élément encapsulé visible dans le sous-classes de la classe
privé	-	Élément encapsulé visible seulement dans la classe
paquetage	~	Élément encapsulé visible seulement dans le s classes même paquetage

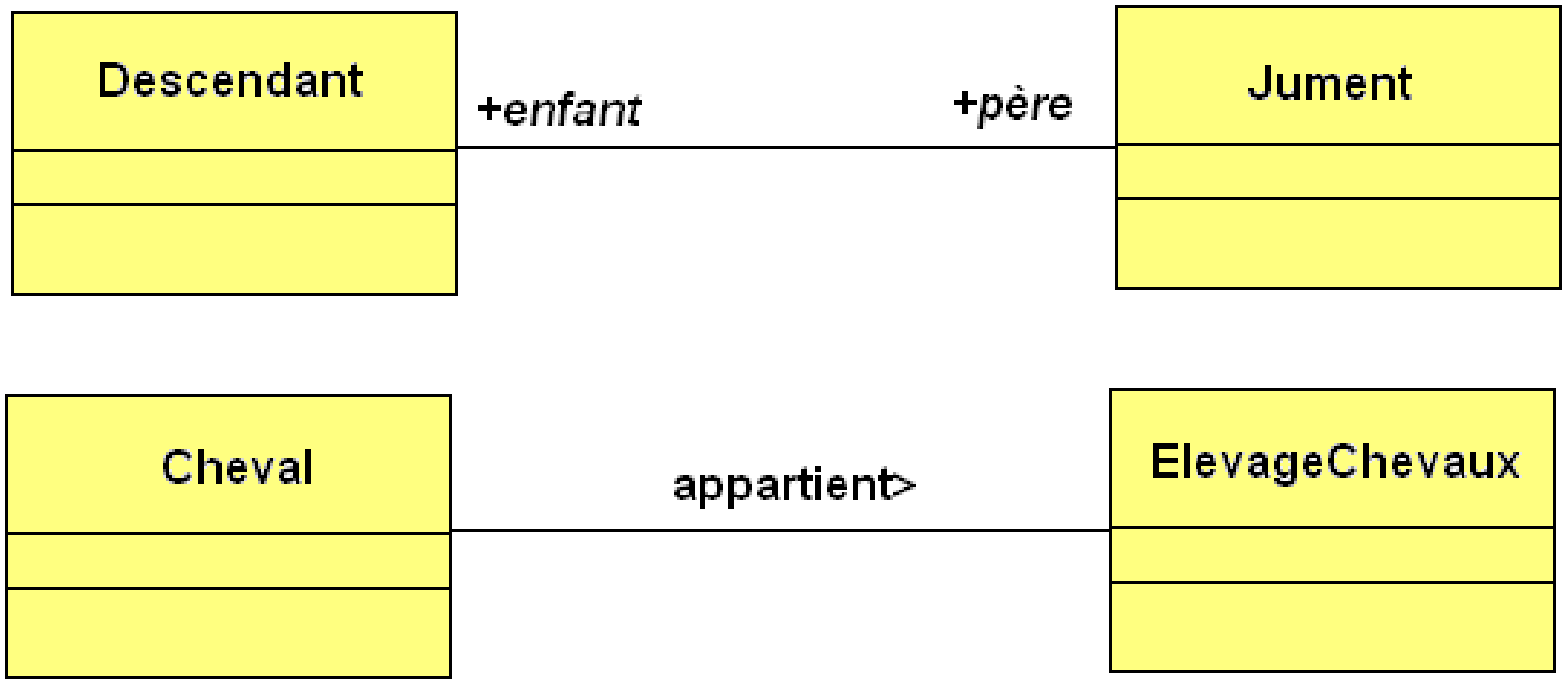
Relation entre classes: associations



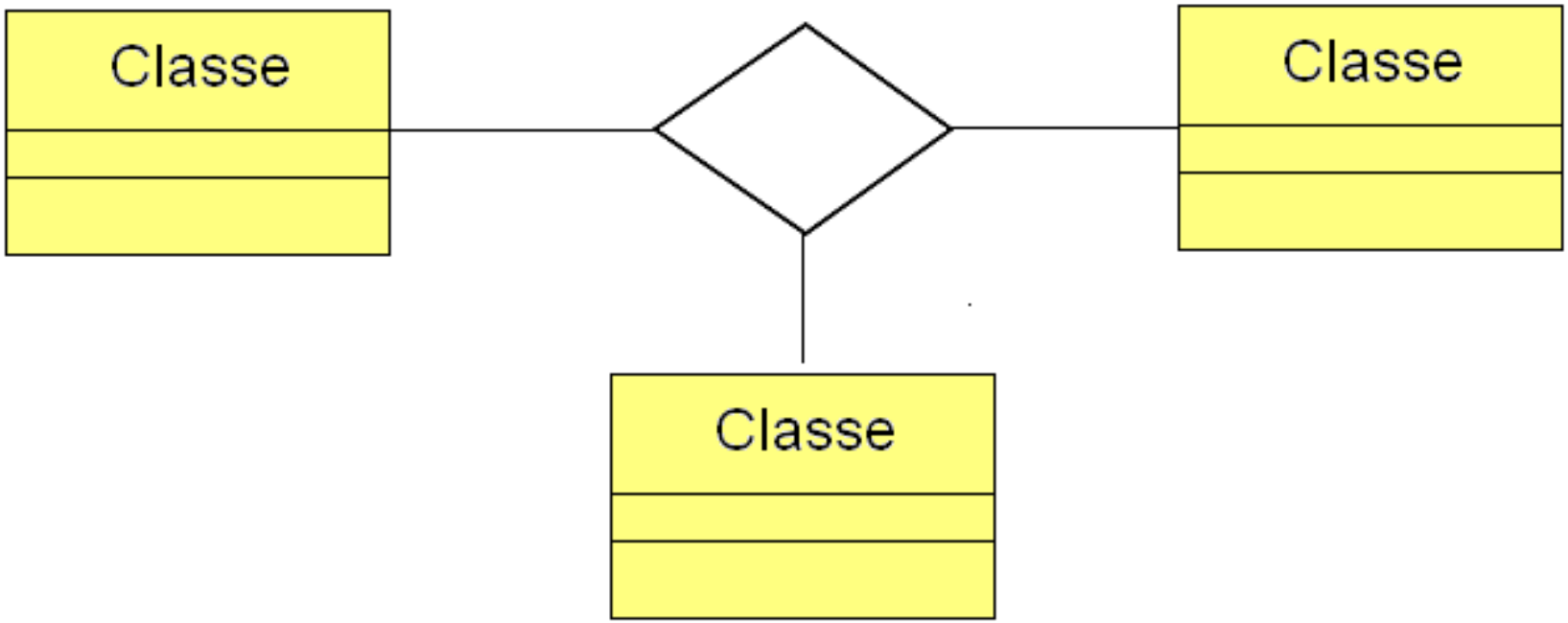
Rôle



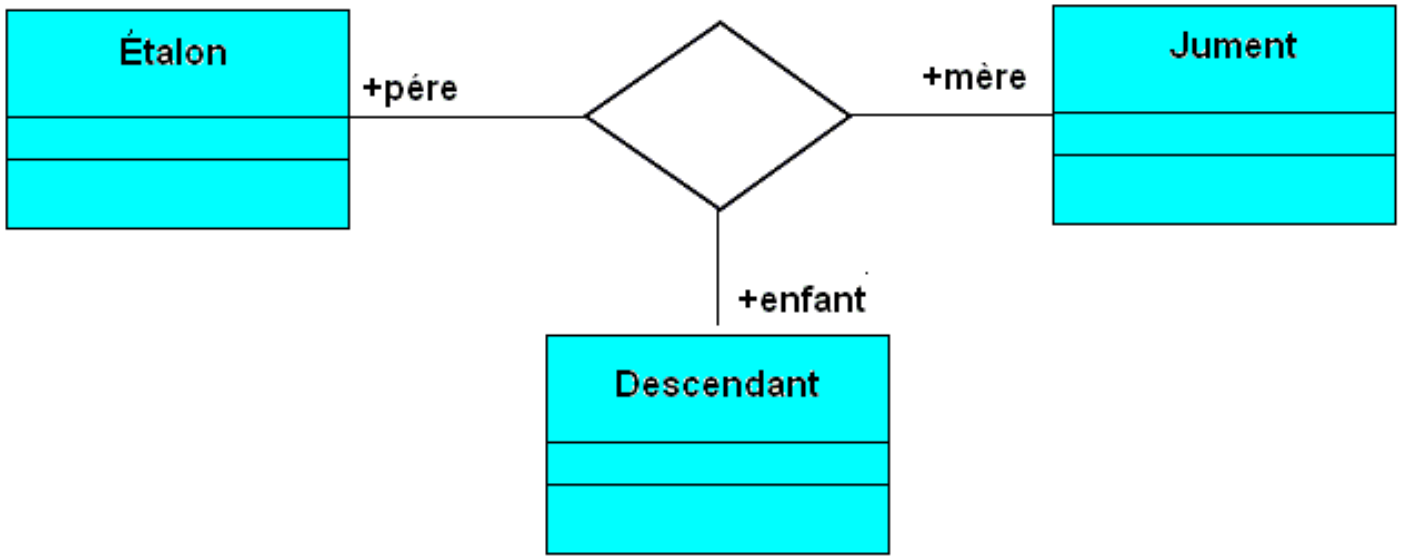
Exemple


Descendant
+enfant
+père
Jument
Cheval
appartient>
ElevageChevaux

Association Ternaire



Exemple



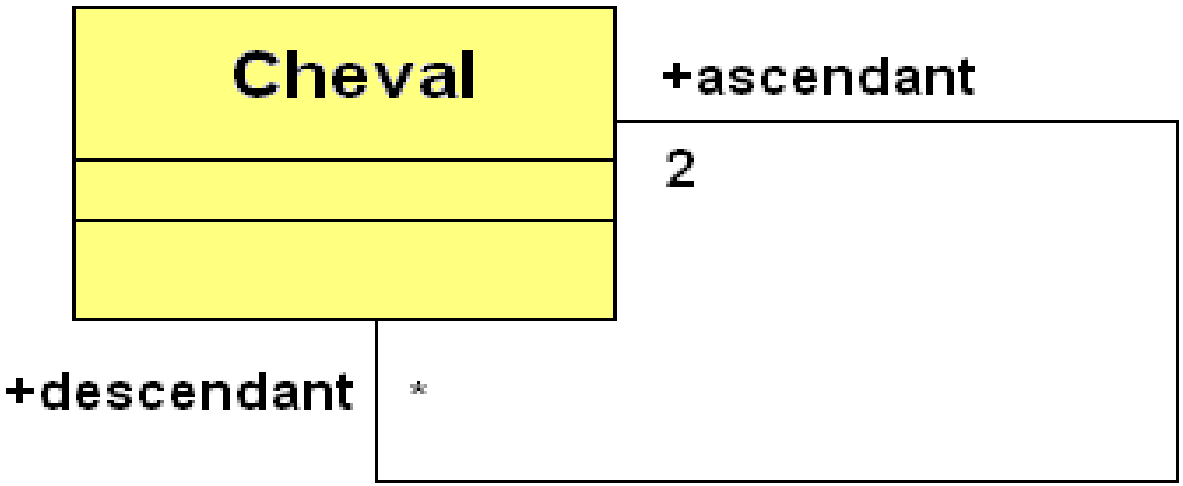
Cardinalité des associations

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

Navigation

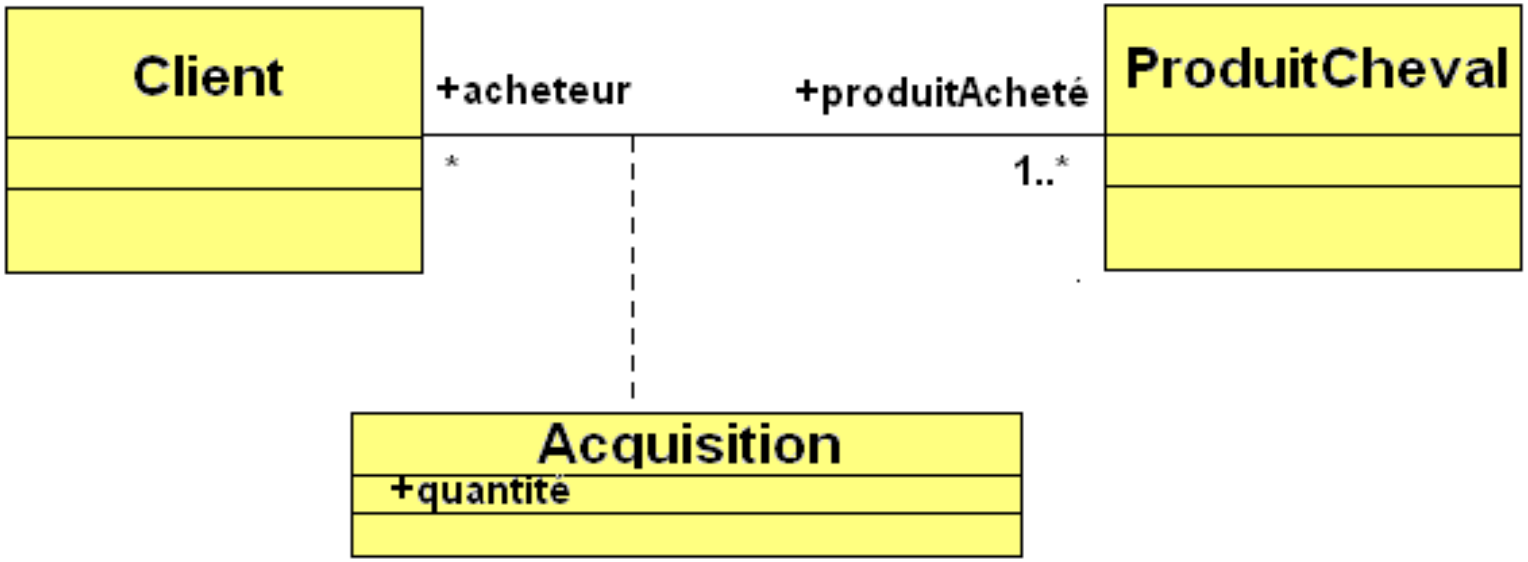
- Spécifier le sens de navigation utile se fait en dessinant l'association sous forme d'une flèche

Association réflexive



Les classes-association

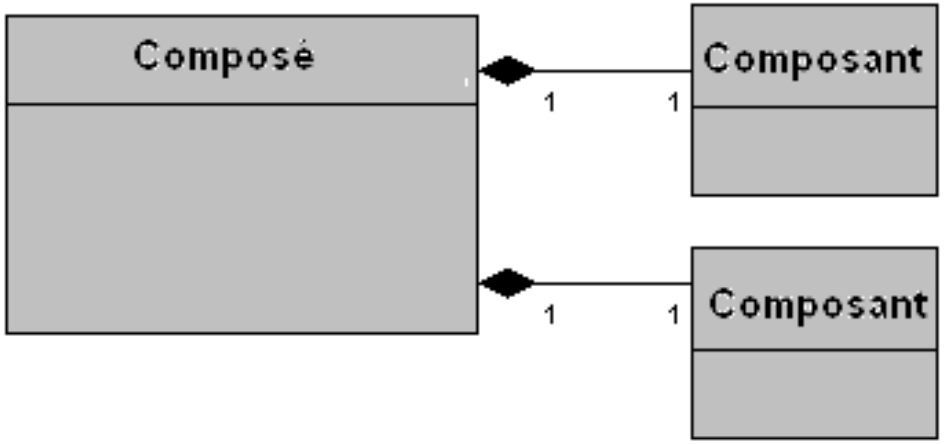
- Le liens entre les instances de classe peuvent porter des informations
- Dans ce cas, l'association qui décrit de tels liens reçoit le statut de classe

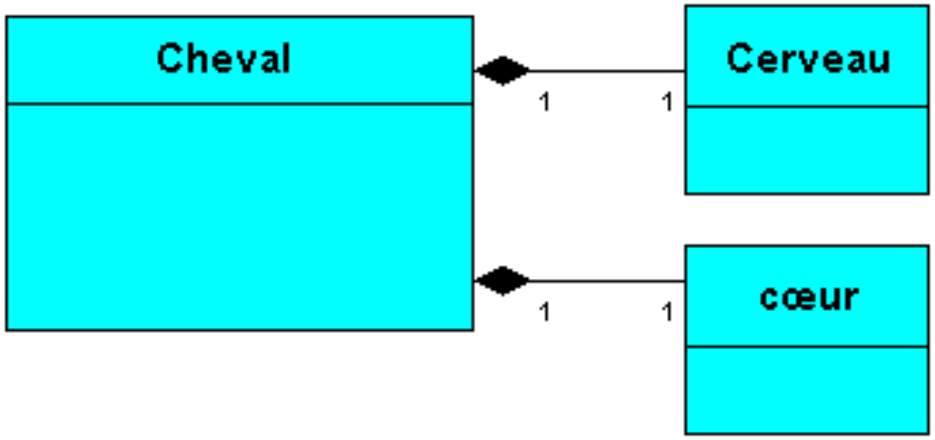


Objets composés: Association forte ou Composition

- 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 obligatoirement de un
- La suppression de l'objet composé entraine la suppression de ces composants

Composition

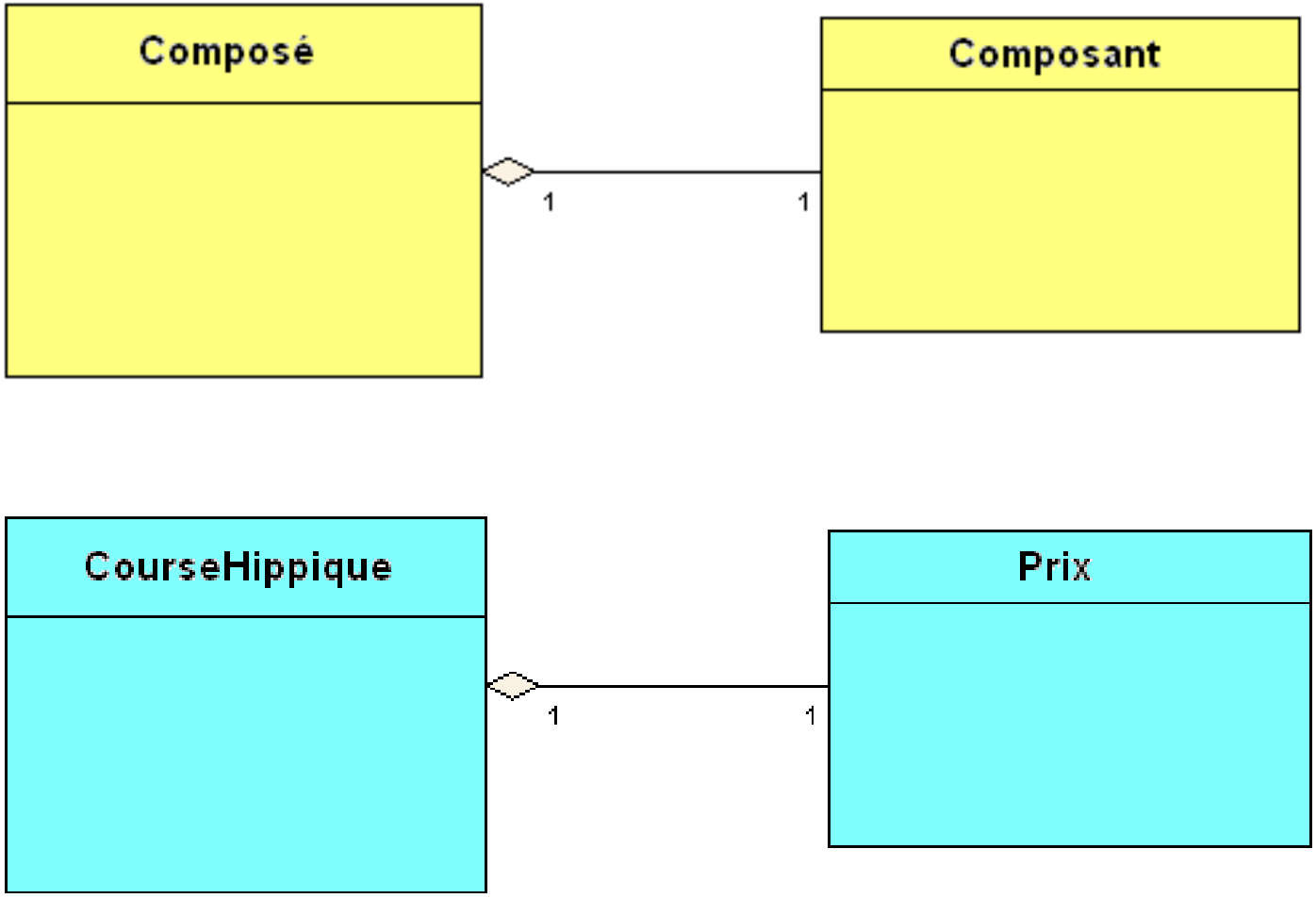




Association faible ou Agrégation

- Les composants peuvent être partagés par plusieurs composés
- La destruction du composé ne conduit pas à la destruction des composants
- Il est possible d'utiliser seulement l'agrégation puis, plus tard, de déterminer quelles associations d'agrégation sont des associations de composition

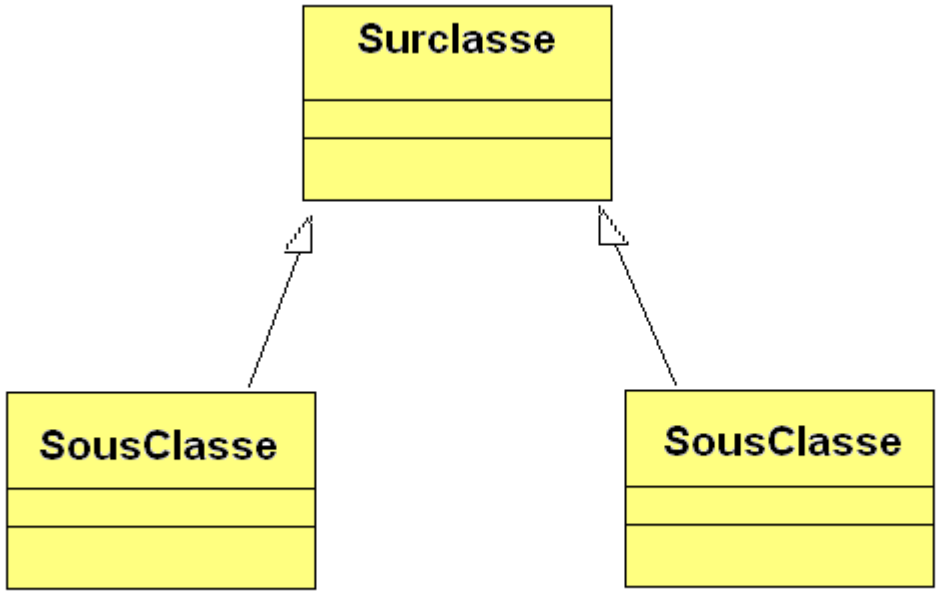
Association



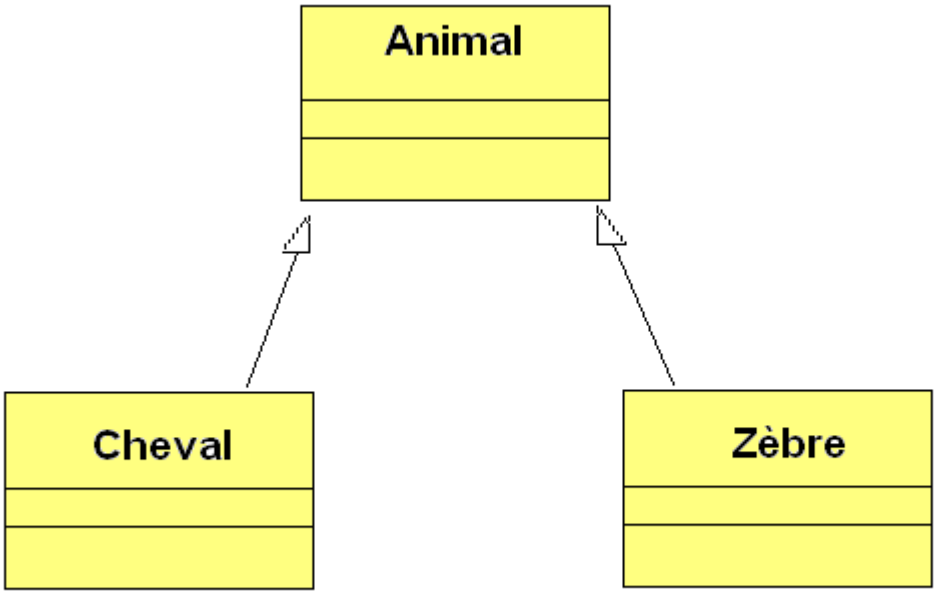
Généralisation - spécialisation

- Une classe est plus spécifique qu'une autre si toutes ses instances sont également instances de cette autre classe
- La classe plus spécifique est dite sous-classe de l'autre classe
- Cette dernière, plus générale, est dite sur-classe

Généralisation



Généralisation



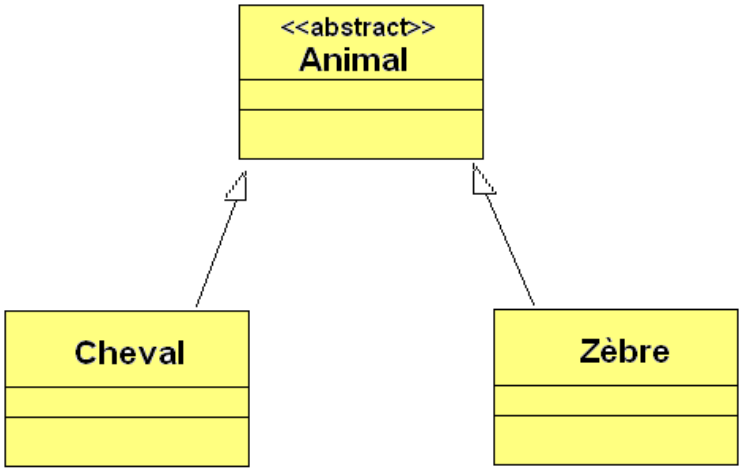
Héritage

- Les instances d'une classe sont aussi instances de ses surclasses
- Elles profitent des attributs et des méthodes introduits au niveau de leur surclasses

Classes concrètes et abstraites

- Une classe concrète possède des instances
- Elle constitue un modèle complet d'objet: tous les attributs et méthodes sont complètement décrits
- Une classe abstraite ne peut pas posséder d'instance directe car elle ne fournit pas une description complète
- Elle a pour vocation de posséder des sous classes concrètes et sert à factoriser des attributs et méthodes communs à ses sous classes.
- Une classe est représentée par le stéréotype <<abstract>>

Concrètes et abstraites



Interface

- Une interface est une classe totalement abstraite: sans attributs et dont toutes les méthodes sont publiques
- L'implantation des méthodes est réalisée par une ou plusieurs classes concrètes, sous classes de l'interface
- La relation d'héritage qui existe entre l'interface et une sous classe d'implantation est appelée relation de réalisation
- Elle est représentée par un trait pointillé

Réalisation