

COO - A214

Les diagrammes de classes

Ivan Logre

logre@i3s.unice.fr

Université de Nice Sophia Antipolis - IUT

D'après les slides de blay@unice.fr

<http://mireilleblayfornarino.i3s.unice.fr/>

Bibliographie



Voir sur le site web les autres cours.

UML Par la pratique (surtout dans sa dernière édition) (présent à la bibliothèque de l'IUT)

Méthodologie en Ingénierie du logiciel, Modélisation Orientée objet, M.Grimaldi – janvier 2010

Analyse du domaine

1



Détermination des concepts du domaine

La détermination des concepts s'effectue sur la base des cas d'utilisation par simple **analyse grammaticale** de la description textuelle.

D'une manière générale,
les **noms** représentent des **concepts** ou des **attributs**
les **verbes** représentent des **comportements** (opérations, méthodes)



**Parlez avec votre client
Capturez son métier**

Identifier les concepts

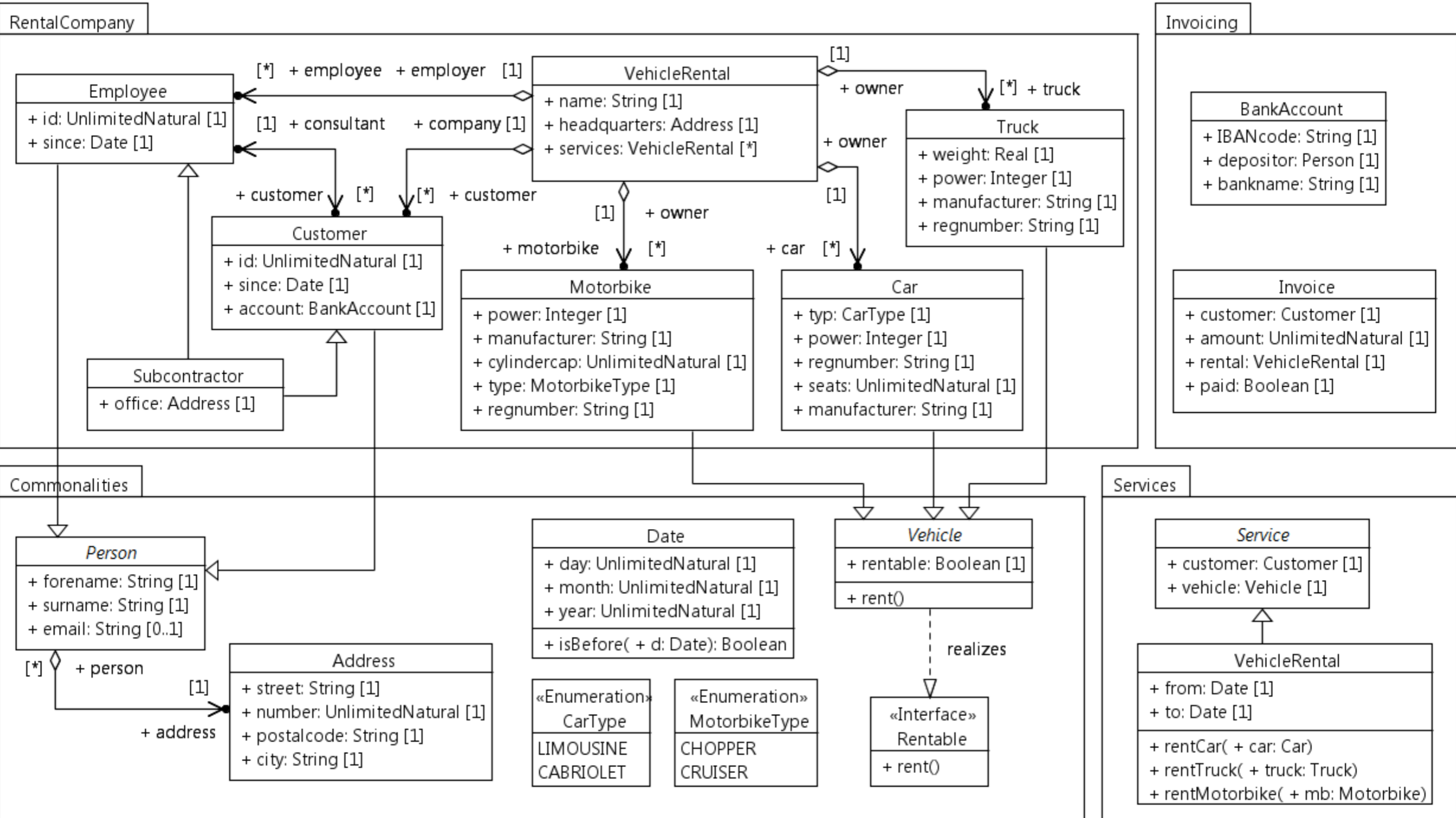
Règle du cartographe

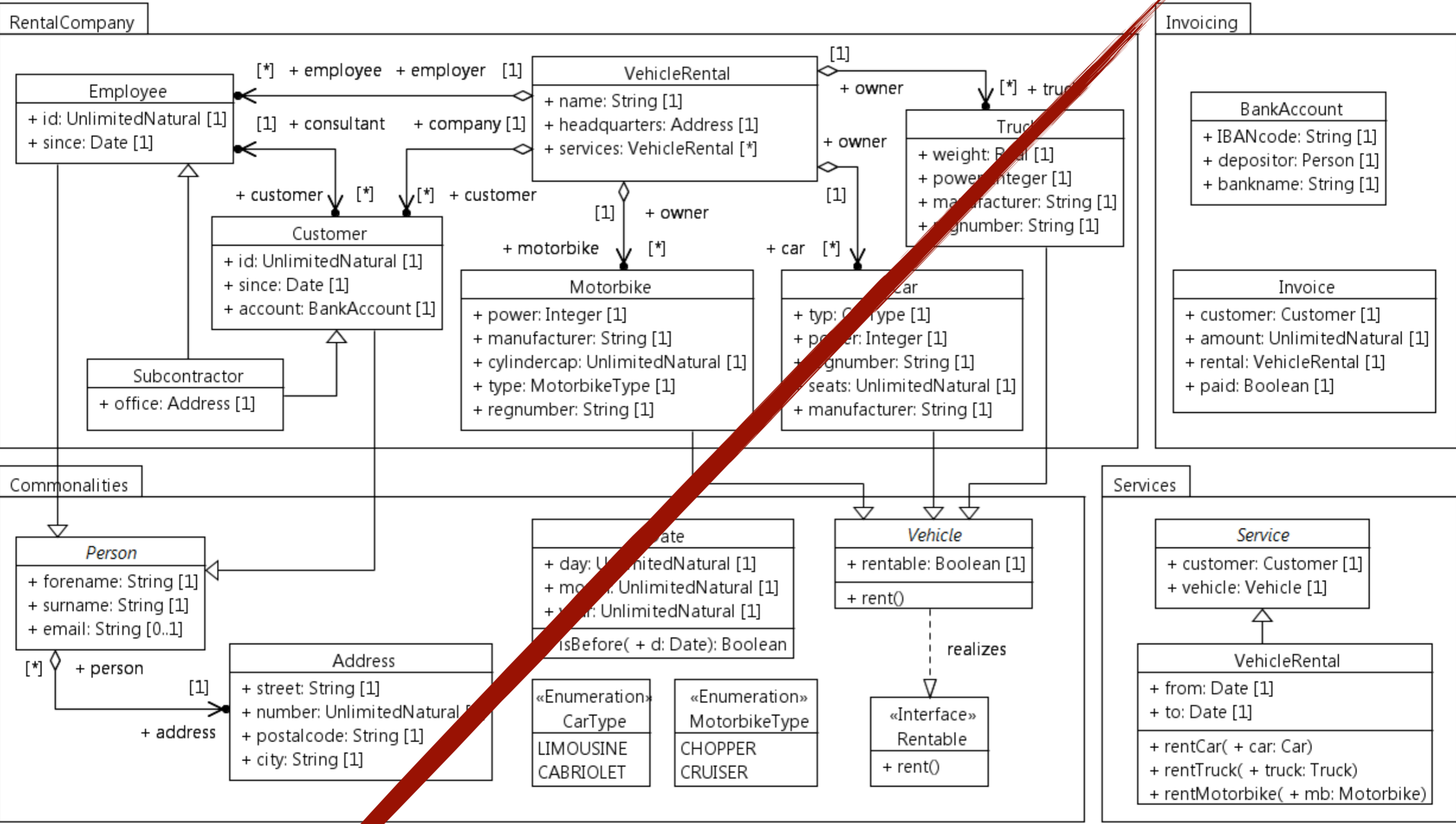
Le modèle du domaine se construit de la même façon qu'un cartographe dessine une carte :

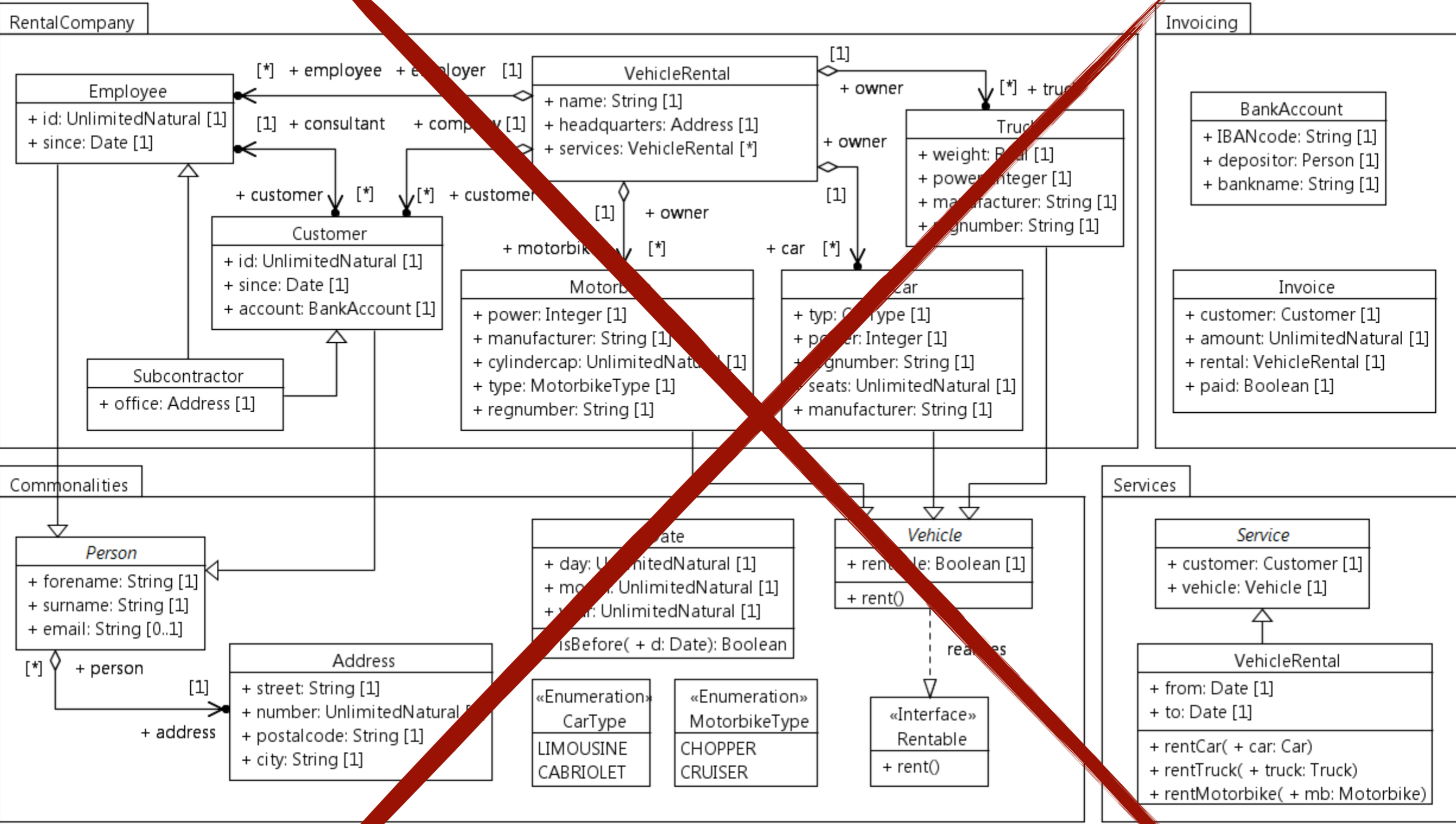
- ✓ En utilisant le **vocabulaire** du domaine étudié.
- ✓ En excluant les éléments **non pertinents** (pas les détails).
- ✓ En n'incluant pas d'éléments **inexistants** dans le domaine (pas d'interprétations).

Identifier les relations entre les concepts

- Déterminer les relations entre les concepts est essentiel.
 - Choix entre concept et attribut : Si un élément du domaine étudié est autre chose qu'un nombre ou un simple texte, alors il s'agit probablement d'un **concept** et non d'un **attribut**.
- > **Qu'est ce qu'un concept ? Qu'est-ce qu'un attribut ?**







Concepts du domaine

*Par l'exemple
et la définition*



La ludothèque

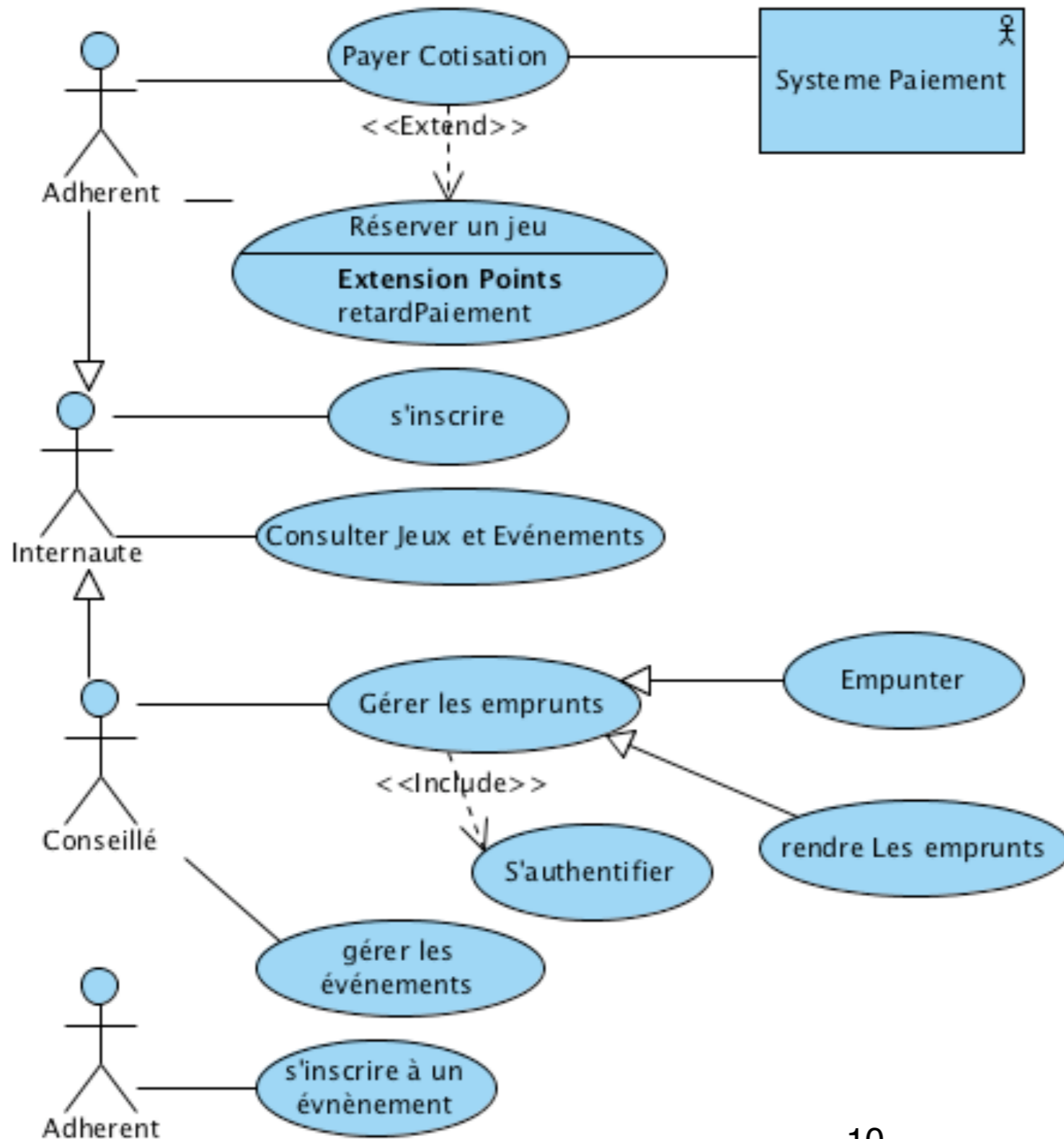
un domaine comme un autre...

- (1) Nous voulons informatiser une ludothèque pour favoriser la consultation des jeux proposés par la ludothèque.
- (2) Les adhérents peuvent emprunter des jeux en s'adressant à un conseiller qui enregistre l'emprunt.
- (3) Les jeux empruntés sont rendus à un conseiller....
- (4) Un adhérent peut réserver des jeux. Une réservation précise l'emprunteur, le jeu et la date de la demande de réservation. L'adhérent est averti quand le jeu revient en rayon.
- (5) Pour organiser un événement le conseiller spécialisé doit alors donner les informations suivantes : les jeux à tester, le nombre maximal et minimal de participants attendus, la date, et l'heure de début de l'événement.
- (6) Un adhérent peut s'inscrire pour participer à un événement à condition qu'il y ait encore de la place.
- (7) Un adhérent peut payer sa cotisation en ligne par un système de paiement externe

La ludothèque

Les cas d'utilisation

Ceci n'est pas "la" correction.



La ludothèque

Description d'un flot associé
à un cas d'utilisation

**Ceci n'est
toujours pas
"la" correction.**

Un conseiller enregistre l'emprunt d'un jeu pour un adhérent

- 1) Le conseiller s'authentifie;
- 2) Le conseiller saisit **l'identifiant** du jeu et de l'adhérent
- 3) Le système vérifie la **disponibilité** du jeu
- 4) Le système vérifie que la cotisation est bien payée
- 5) Le système vérifie que l'**adhérent n'a pas de pénalité impayée**
- 6) Le système enregistre **l'emprunt**.
- 7) Le système signale que l'emprunt est valide.

Une classe

"La classe est un concept abstrait qui permet de représenter toutes les entités d'un système. Une classe peut donc représenter une voiture, un devis, un utilisateur connecté, une structure de donnée ou tout autre élément devant être modélisé.

La classe est définie par son nom, ses attributs et ses opérations."

<http://www.uml-sysml.org/>

Une classe

"La classe est un concept abstrait qui permet de représenter toutes les entités d'un système. Une classe peut donc représenter une voiture, un devis, un utilisateur connecté, une structure de donnée ou tout autre élément devant être modélisé.

La classe est définie par son nom, ses attributs et ses opérations."

<http://www.uml-sysml.org/>



Je suis une boîte...
une **classe** !

Une classe, c'est une boîte.

Ivan Logre

Un attribut



J'ai des attributs,
je suis mieux définie.

Un attribut, c'est ce qui aide à définir une boîte.

Un attribut ne peut représenter qu'une **valeur primitive** (entier, texte, date, identificateur, matricule, . . .).

Un attribut ne peut représenter que des données **relatives au concept auquel il est associé**



Un attribut



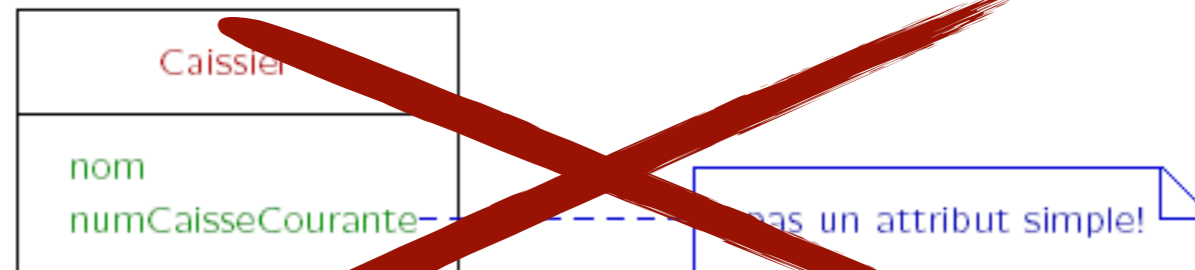
J'ai des attributs,
je suis mieux définie.

Un attribut, c'est ce qui aide à définir une boite.

Un attribut ne peut représenter qu'une **valeur primitive** (entier, texte, date, identificateur, matricule, ...).

Un attribut ne peut représenter que des données **relatives au concept auquel il est associé**

Incorrect



Correct



Une opération

la boîte

attribut 1
attribut 2

opération 1
opération 2

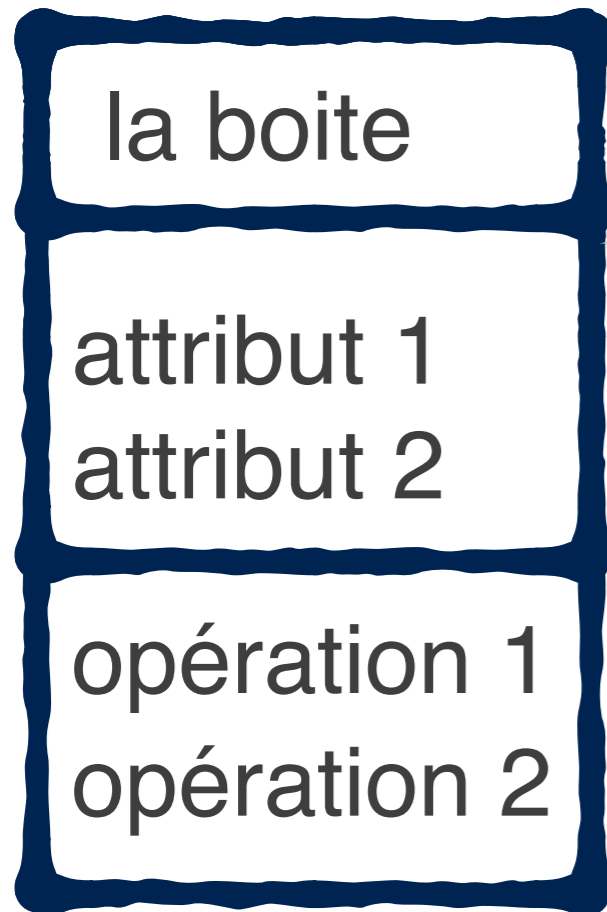
Maintenant je sais
faire des trucs !

Une opération, c'est ce que sait faire boîte.

le secrétaire

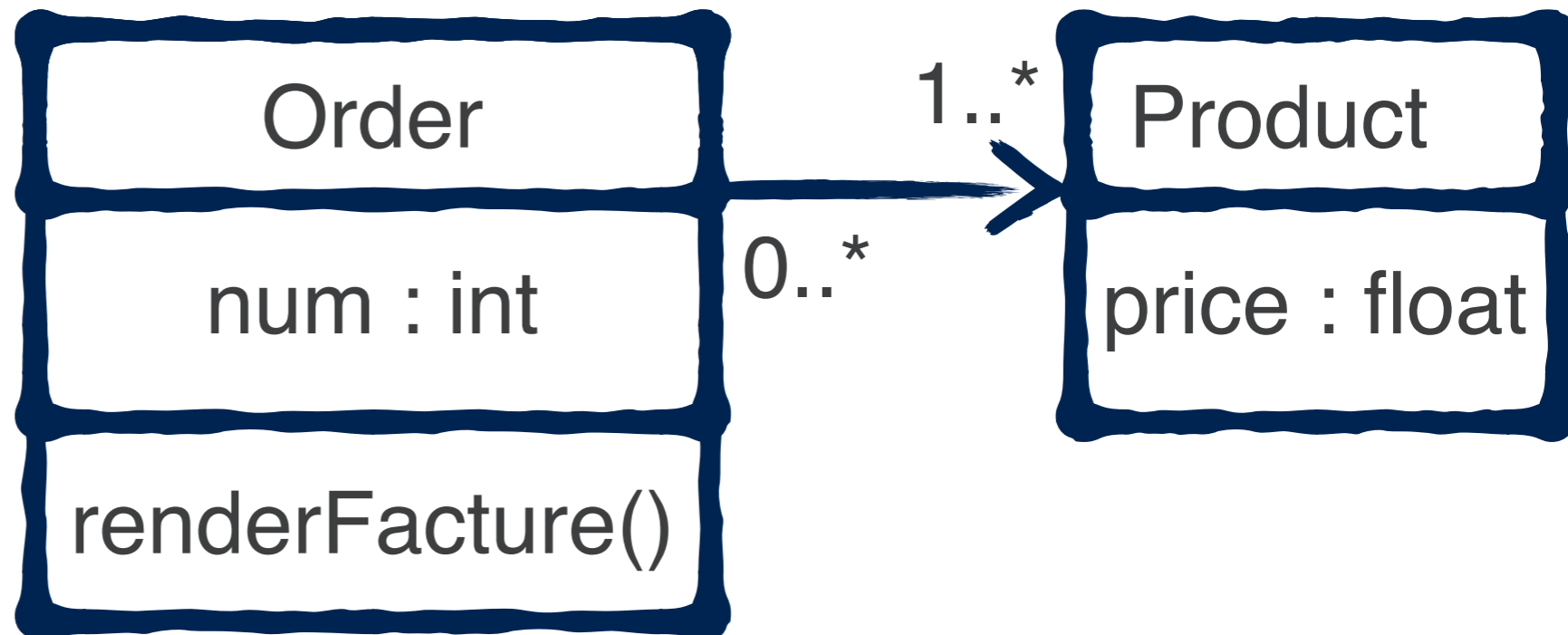
editFacture()

Une opération

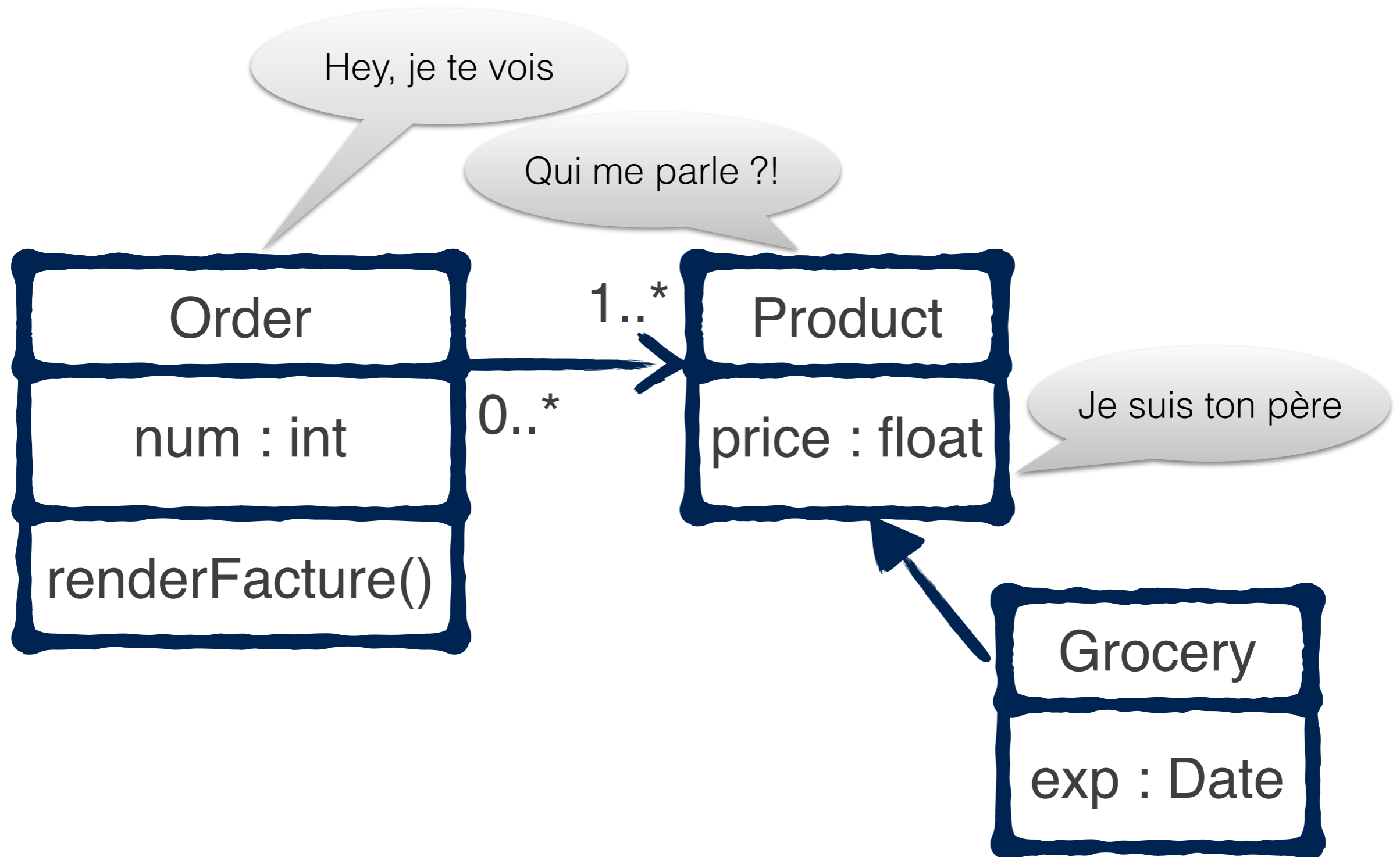


Maintenant je sais
faire des trucs !

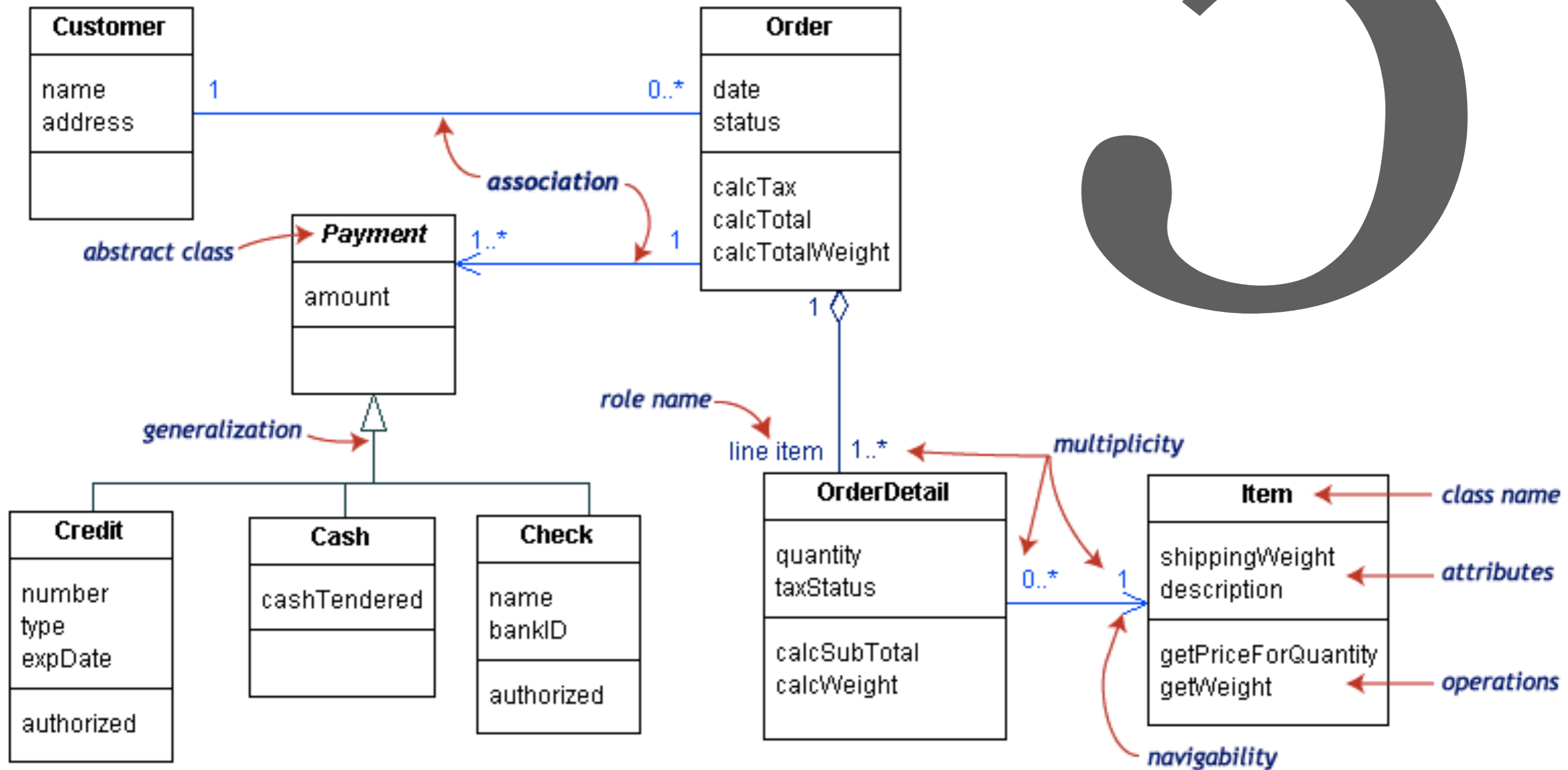
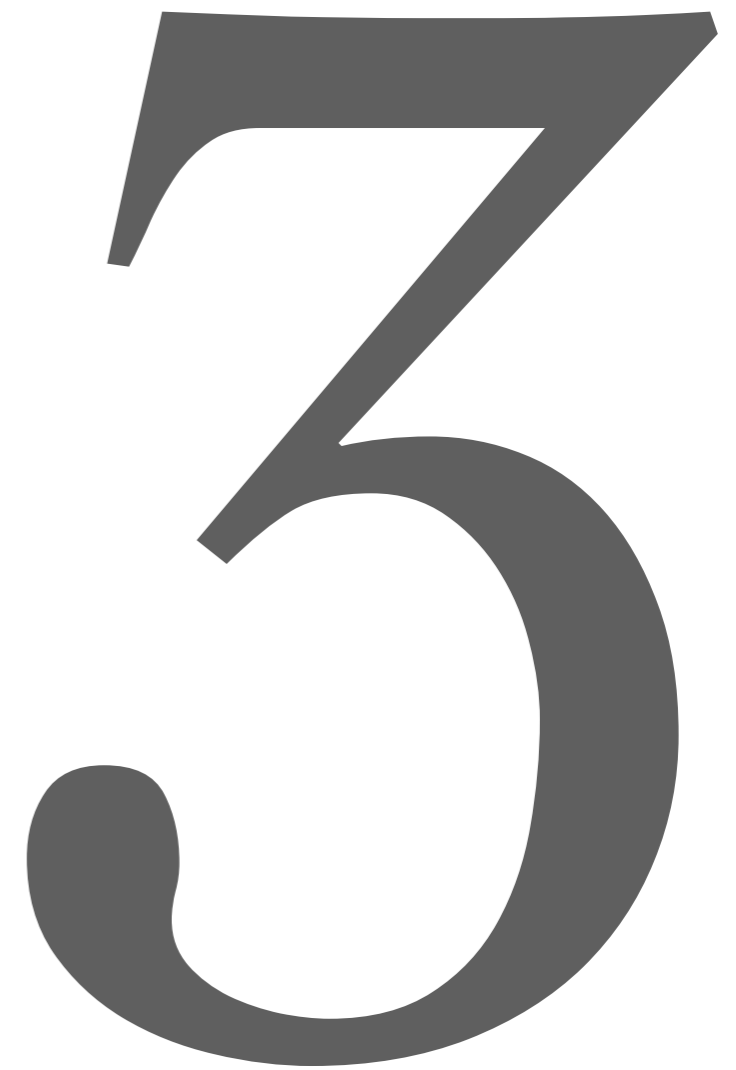
Une opération, c'est ce que sait faire boite.



Les relations

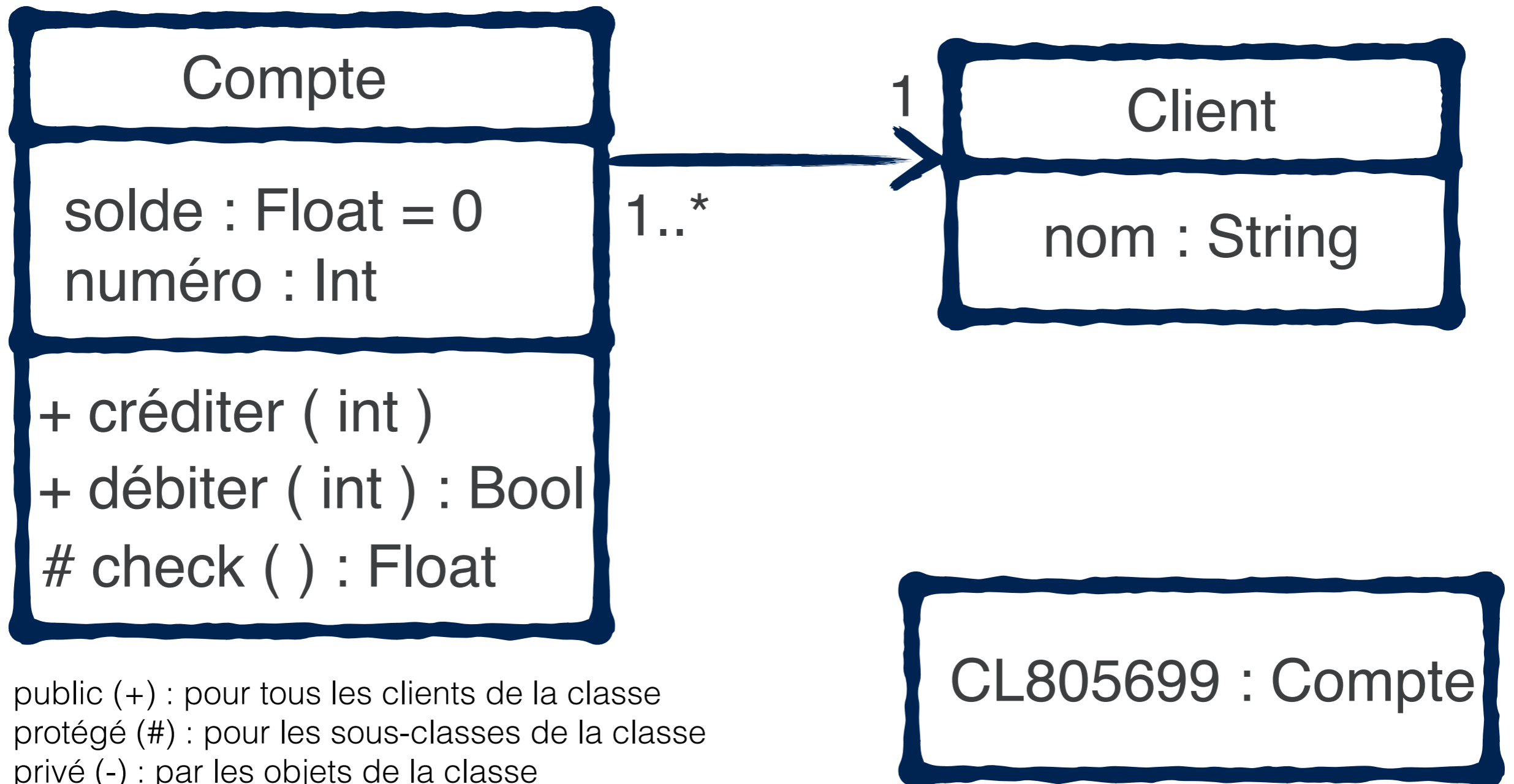


Les bases de la Notation



La notation en pratique

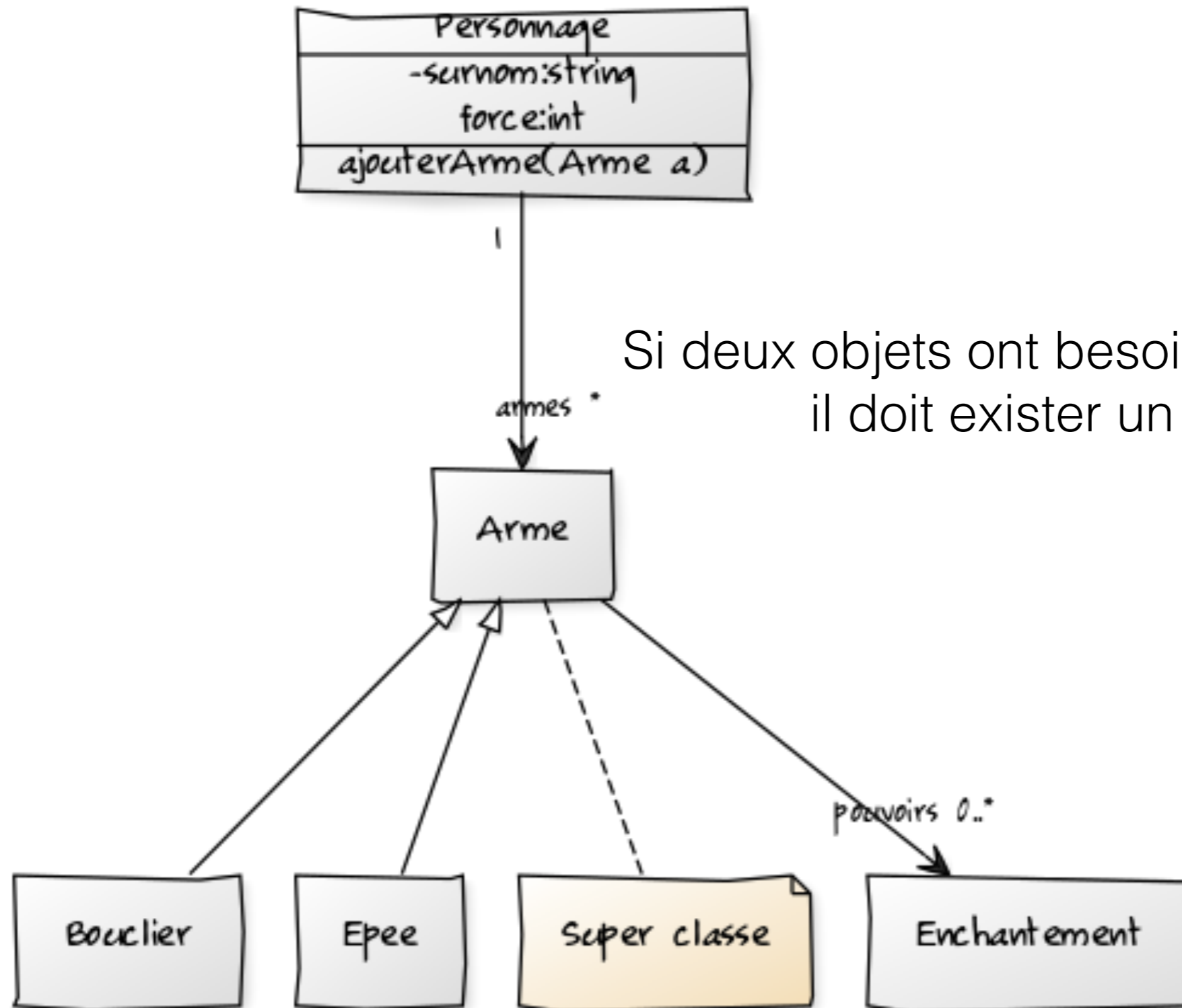
Pour classes et objets



public (+) : pour tous les clients de la classe
protégé (#) : pour les sous-classes de la classe
privé (-) : par les objets de la classe

Relations entre classes

Fournissent un chemin de communication entre objets.



Si deux objets ont besoin de se parler,
il doit exister un lien entre eux.

Associations

Nommage
Rôle
Multiplicité
Navigation

Les associations peuvent avoir des étiquettes de **nommage**

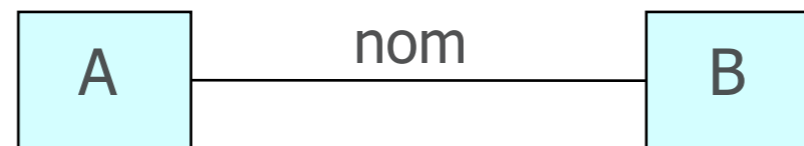
Les associations peuvent avoir des **rôles**

Les associations peuvent indiquer la **navigation**

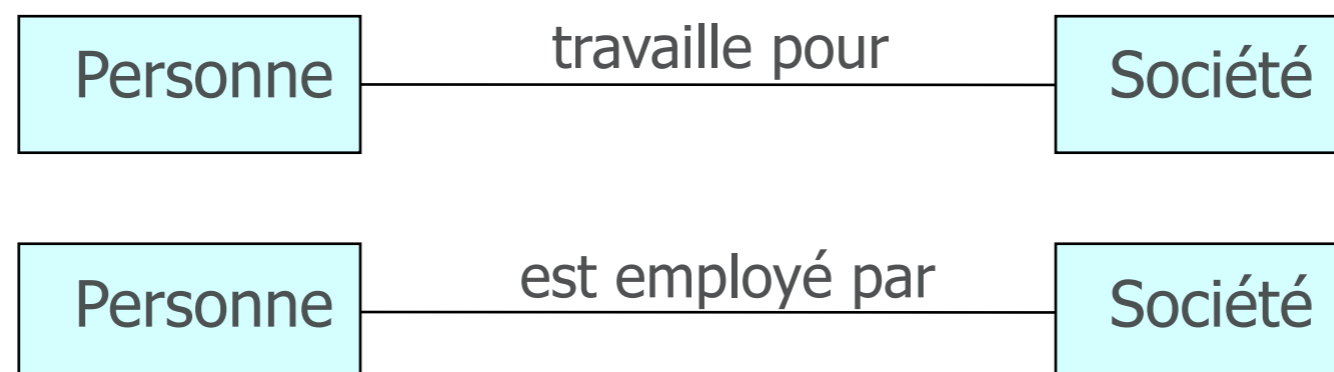
Les associations peuvent indiquer une **multiplicité**

Etiquette de nommage

Une association peut être nommée afin de faciliter la compréhension des modèles. Il est indiqué au milieu du lien symbolisant l'association



L'usage recommande de choisir comme nom d'une association une **forme verbale** active ou passive



Etiquette de nommage

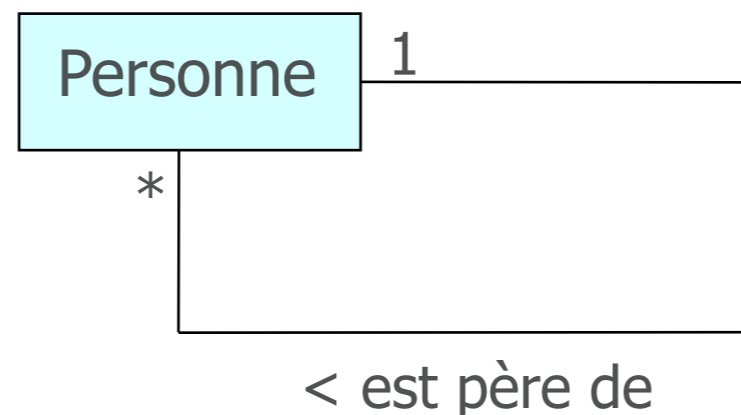
Nommage

Rôle
Multiplicité
Navigation

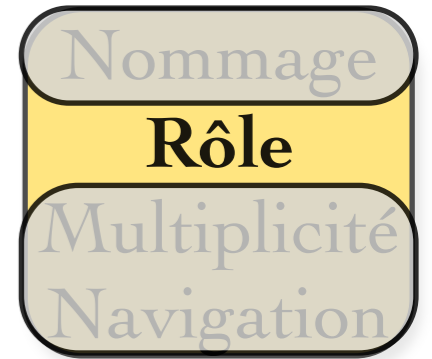
Autant que possible, respectez le sens de lecture !

En cas d'ambiguïté, ajouter '<' ou '>'

Exemples



Responsabilité des classes

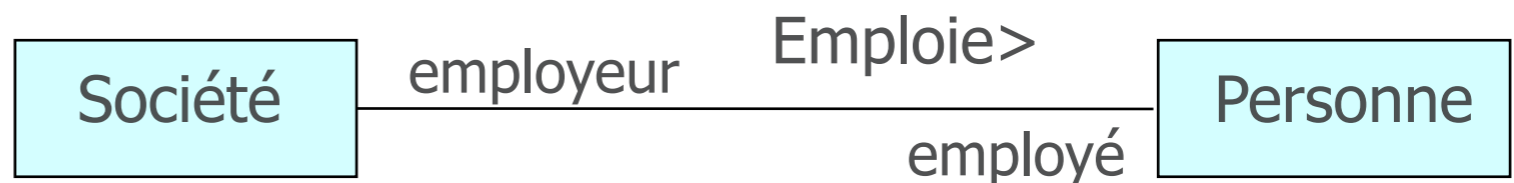


Attribuer à une extrémité un nom appelé **rôle** décrit comment une classe source voit une classe destination au travers de l'association

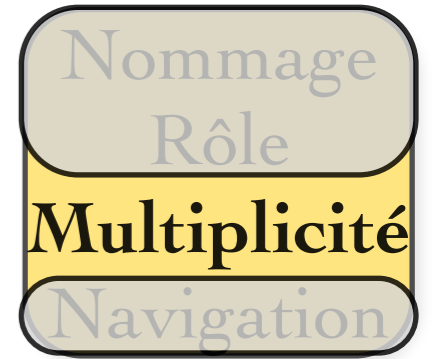
Le rôle est placé près de la fin de l'association et à côté de la classe à laquelle il est appliqué

L'utilisation des rôles est optionnelle (on la préfère au nommage de l'association)

Représentation et exemple

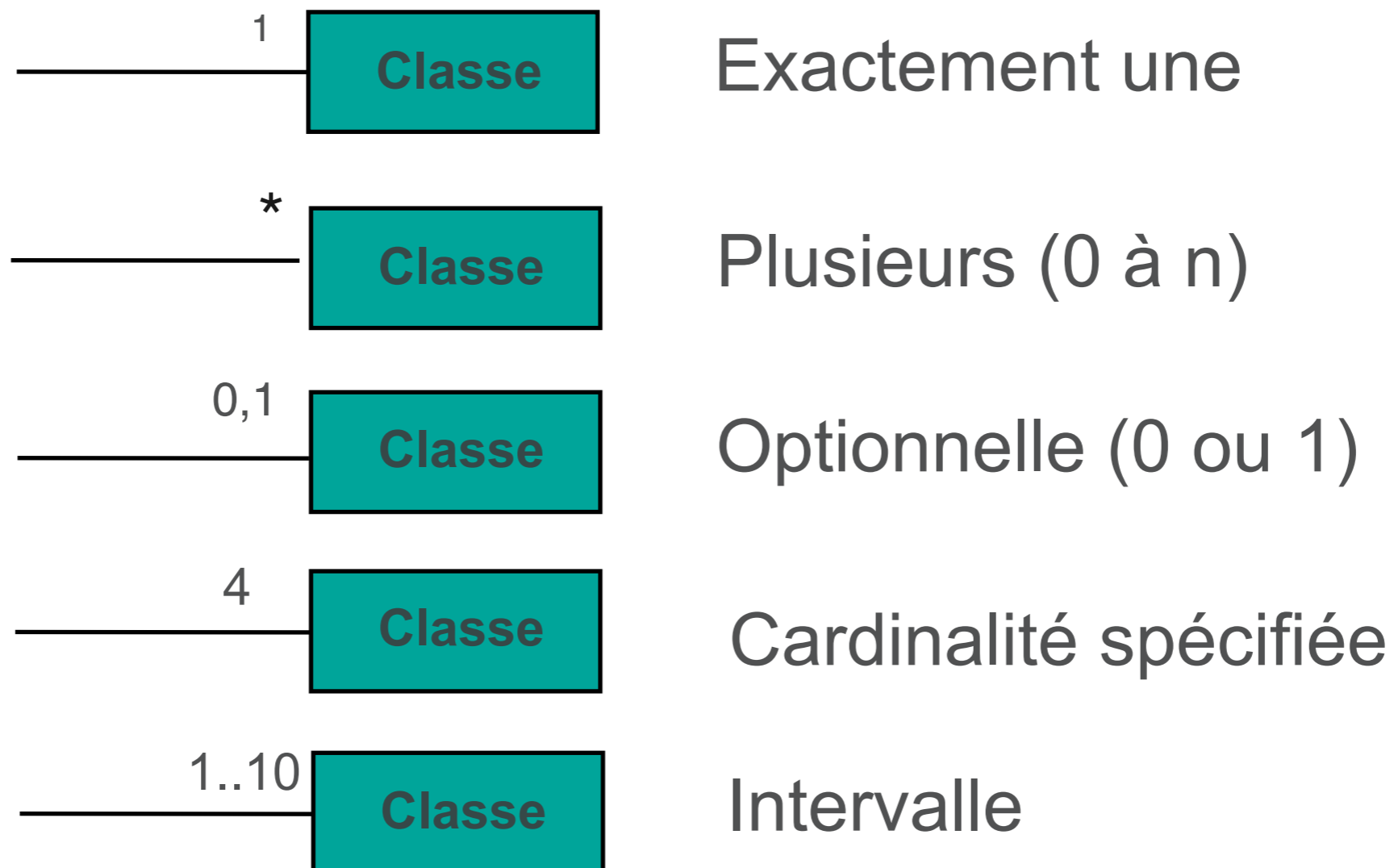


Quantification de l'association



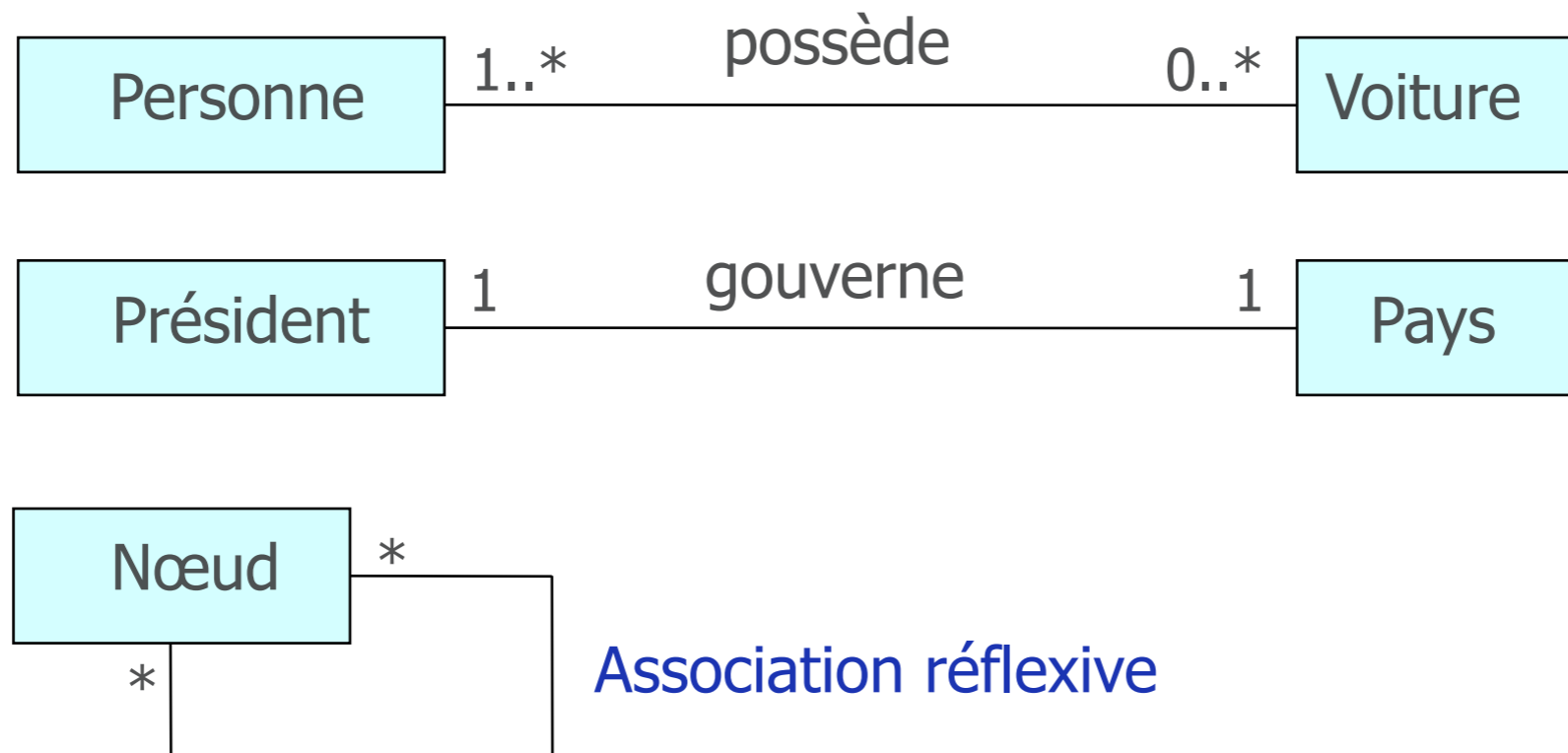
La multiplicité est définie par le nombre d'objets qui participent à une relation

- le nombre d'instances d'une classe reliées à UNE instance d'une autre classe
- deux multiplicités : une à chaque bout de la relation



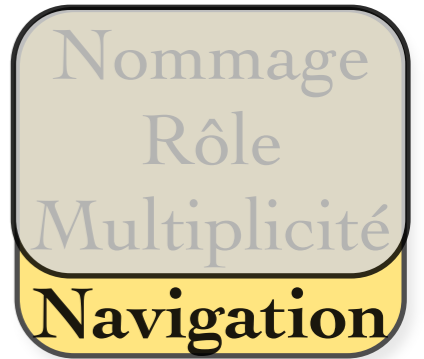
Quantification de l'association

Nommage
Rôle
Multiplicité
Navigation

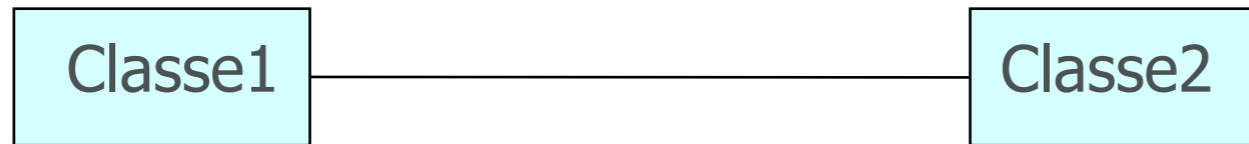


Un réseau informatique est composé de nœuds inter-connectés

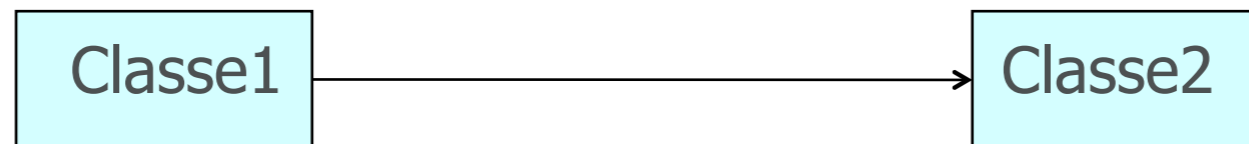
Quantification de l'association



Les objets de Classe2 sont accessibles à partir de ceux de Classe1 et vice-versa

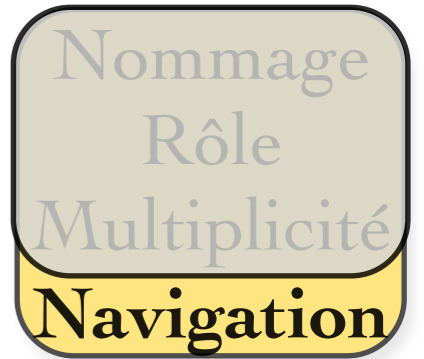


Si la navigation est restreinte, une flèche indique le sens de navigation
Les objets de la Classe 1 sont accessibles à la classe 2

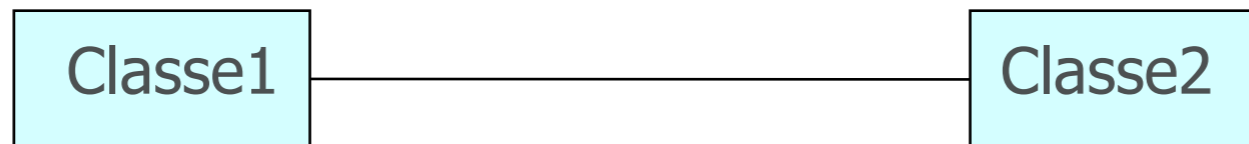


Pas de flèche => bidirectionnelle
Par défaut, mieux si unidirectionnelle.

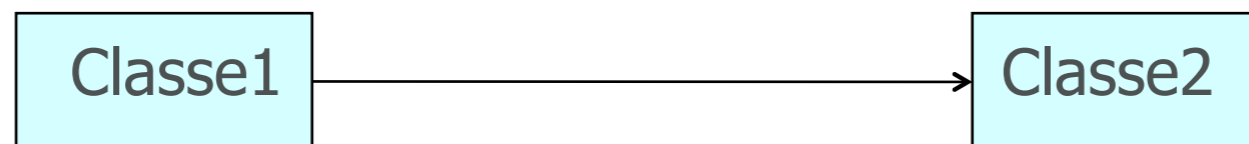
Quantification de l'association



Les objets de Classe2 sont accessibles à partir de ceux de Classe1 et vice-versa



Si la navigation est restreinte, une flèche indique le sens de navigation
Les objets de la Classe 1 sont accessibles à la classe 2



Pas de flèche => bidirectionnelle
Par défaut, mieux si unidirectionnelle.

4

Notations avancées

Les différentes relations



Association



Héritage



Agrégation

Héritage

L'héritage est une relation entre

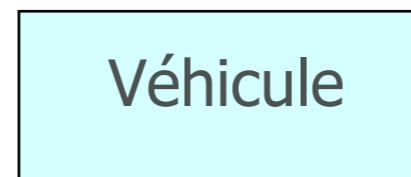
- une super-classe (classe de base)
- ses sous-classes (classes dérivées)

Deux manières d'identifier une relation d'héritage :

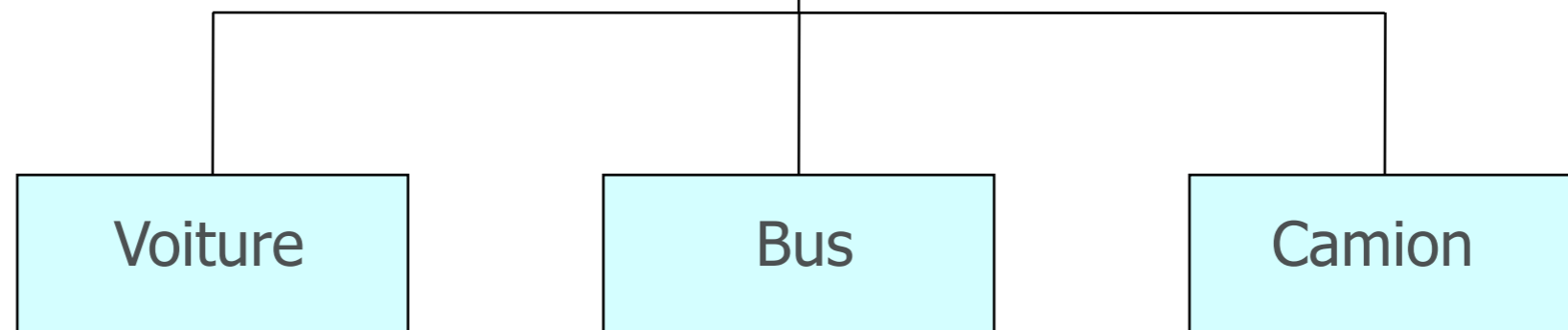
- Généralisation
- Spécialisation

Les éléments communs sont reportés au niveau le plus haut de la hiérarchie (attributs, comportements, relations)

Un Bus est un Véhicule.



Un Véhicule n'est pas un Bus.



Héritage

L'héritage est une relation entre

- une super-classe (classe de base)
- ses sous-classes (classes dérivées)

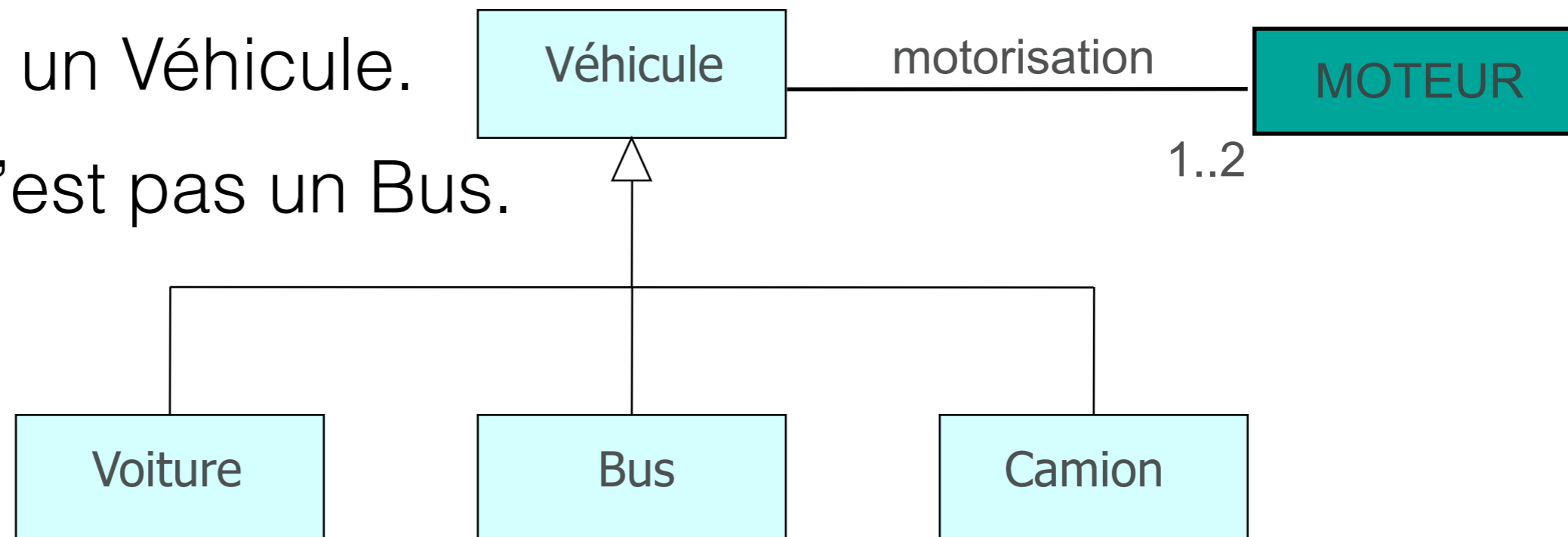
Deux manières d'identifier une relation d'héritage :

- Généralisation
- Spécialisation

Les éléments communs sont reportés au niveau le plus haut de la hiérarchie (attributs, comportements, relations)

Un Bus est un Véhicule.

Un Véhicule n'est pas un Bus.

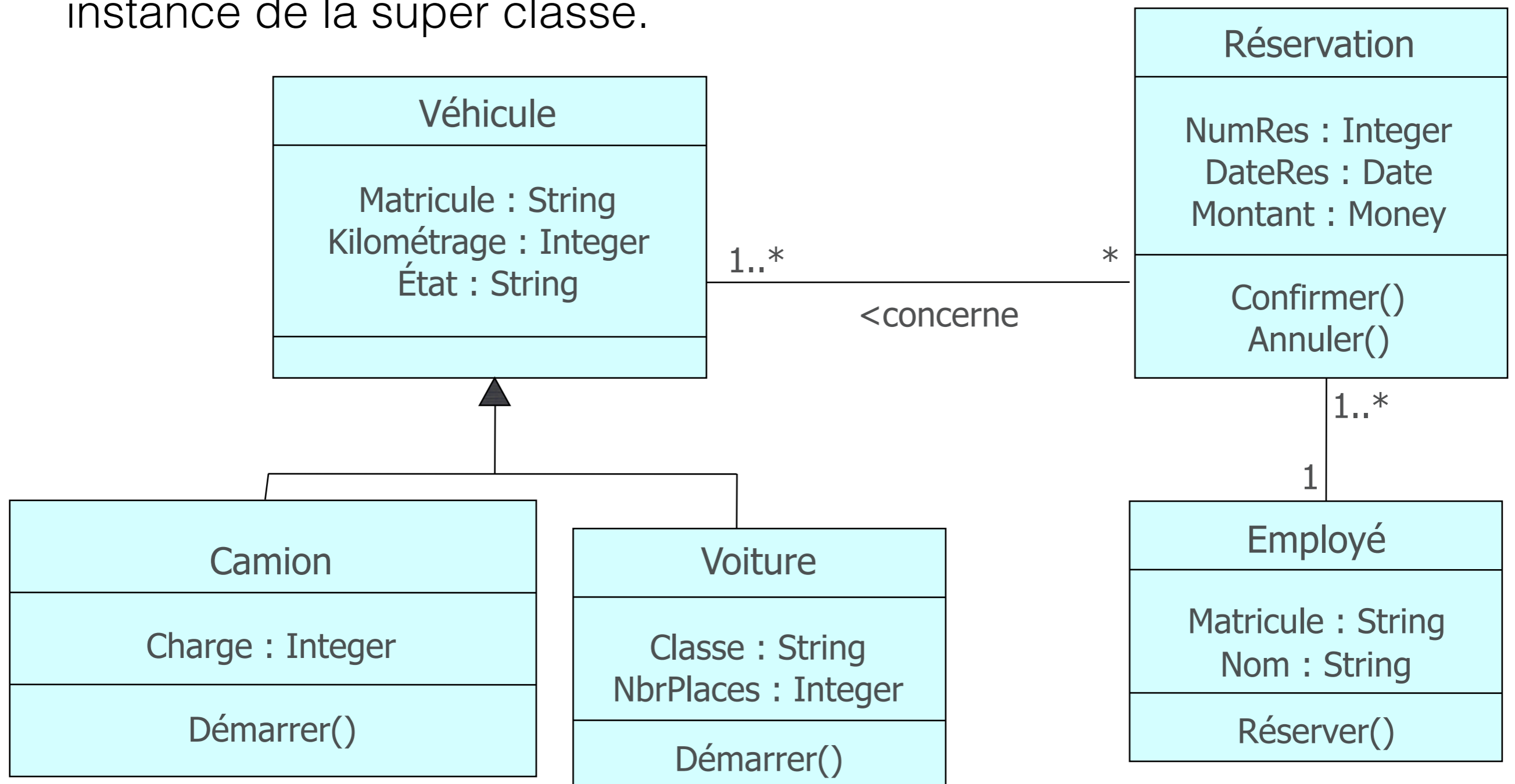


Héritage

L'enfant acquiert les propriétés **des** parents :

les attributs, les relations et les opérations

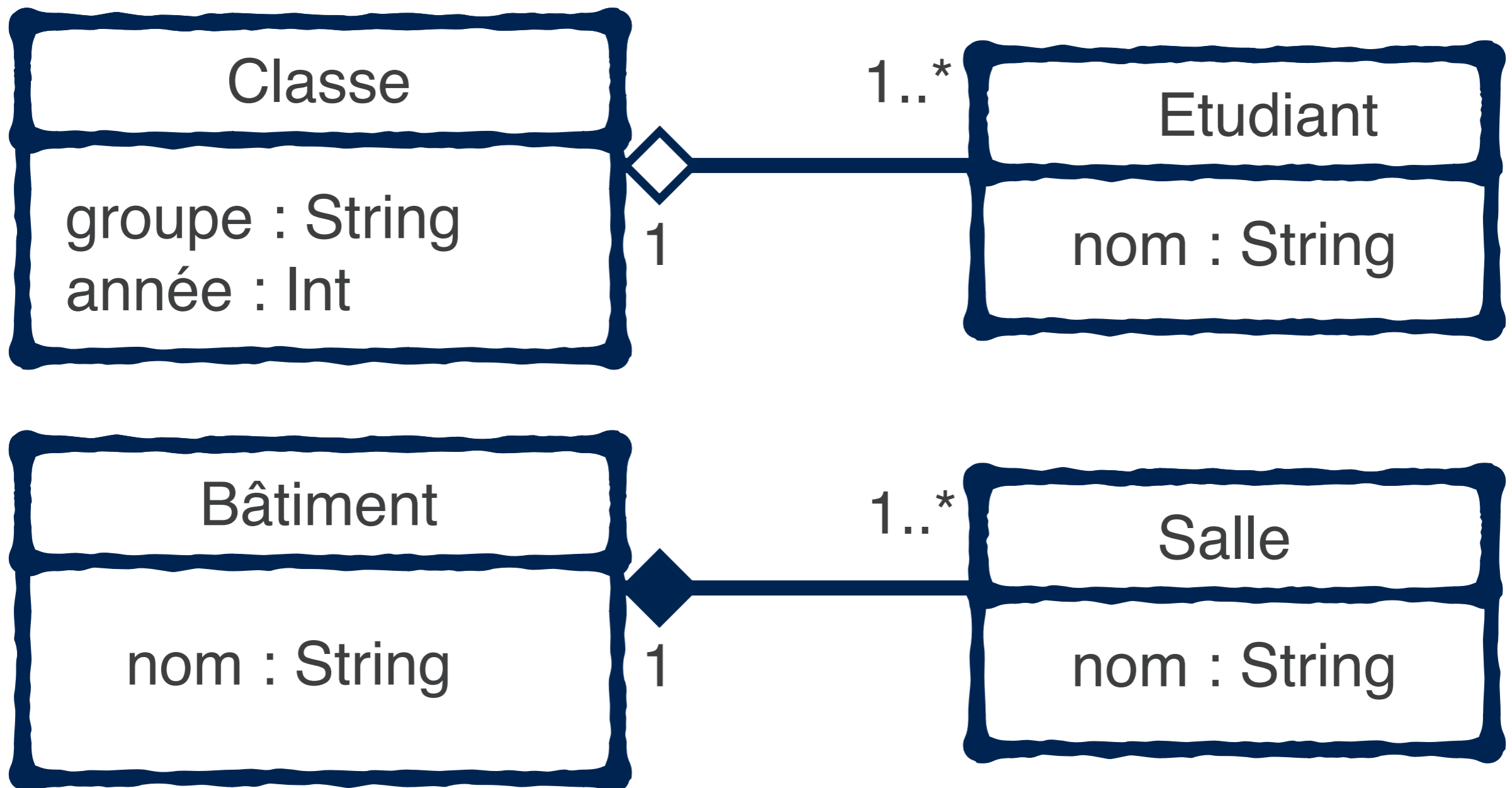
Il est possible de substituer une instance d'une sous-classe à une instance de la super classe.



Agrégation VS Composition

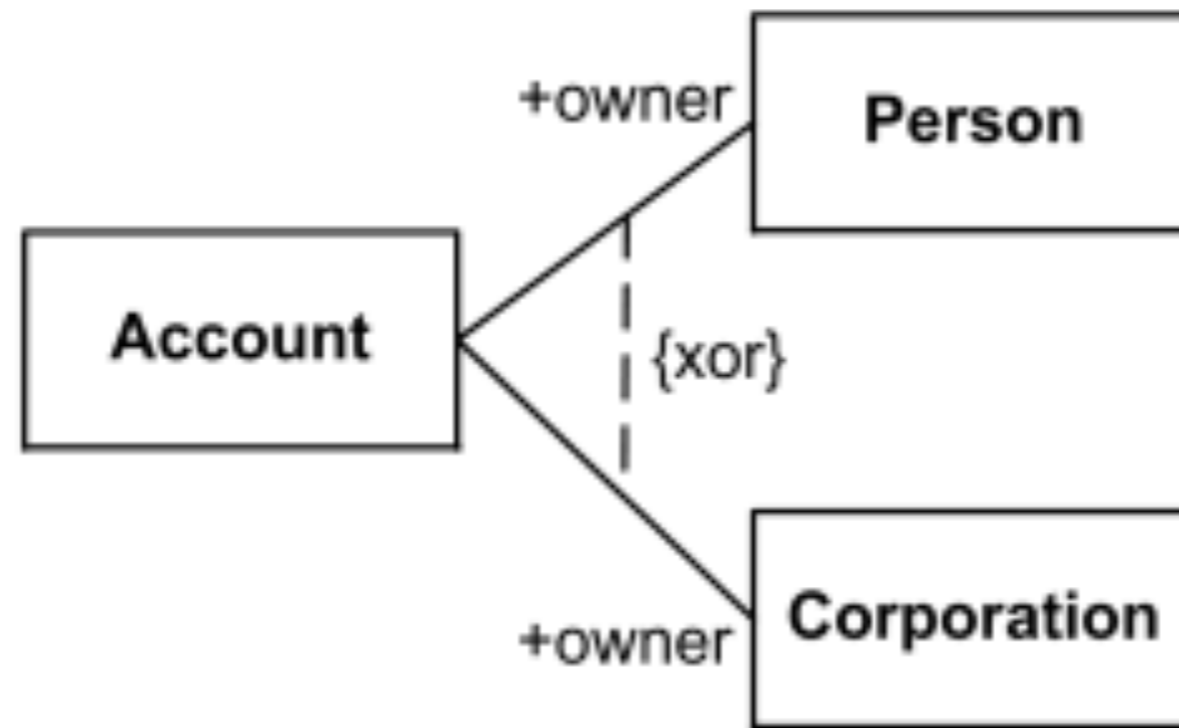
Définissent un contenant et un contenu. (Relation asymétrique)

Navigabilité et multiplicité automatique pour le contenant.



Contrainte exclusive

Définit une relation logique OU exclusif entre deux relations.



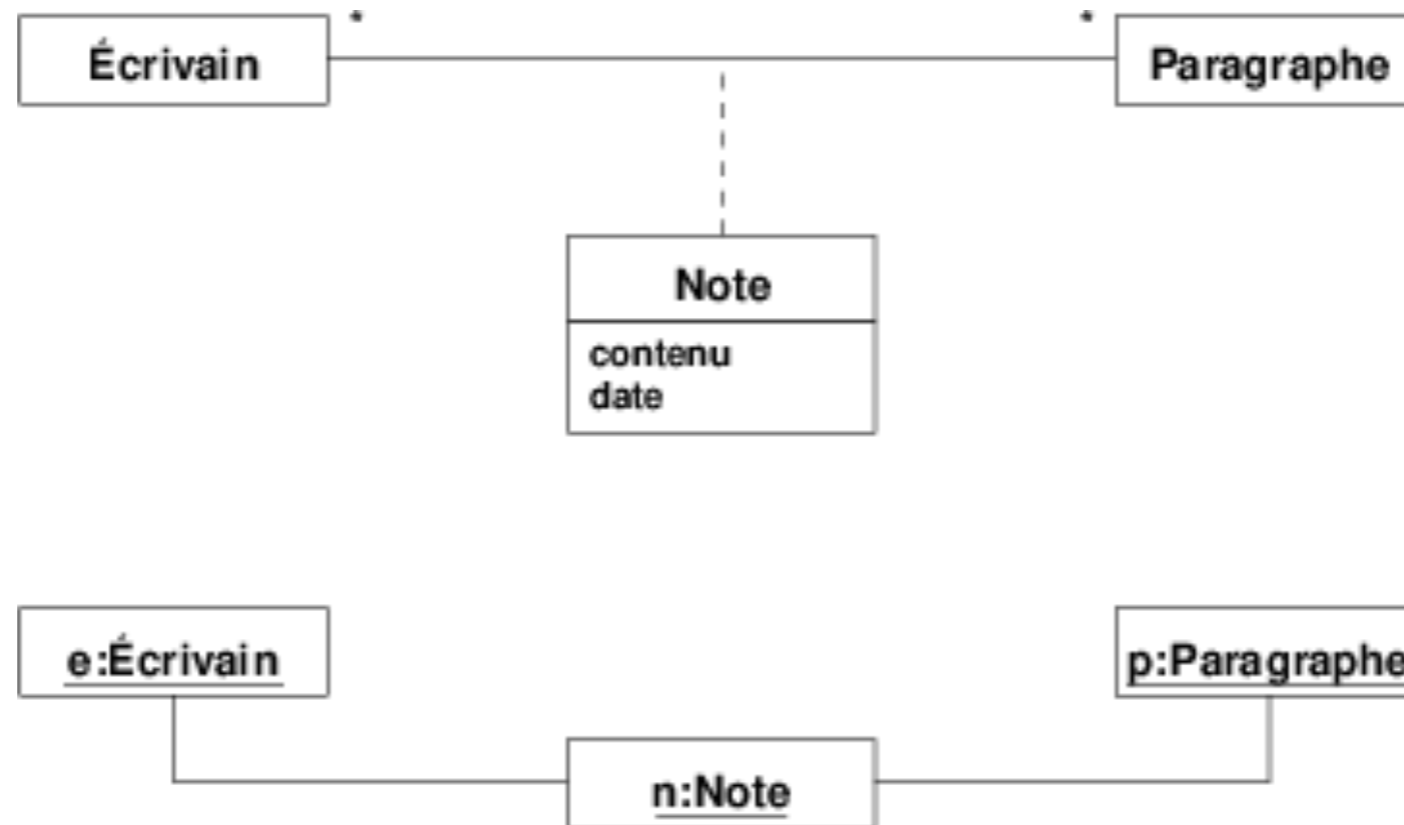
Exemple : le propriétaire d'un compte est soit une personne soit une entreprise.

Utile seulement si...?

<http://www.uml-diagrams.org/constraint.html>

Classe d'association

Définit une classe qui n'a de valeur que dans l'association de deux autres.



Exemple : à chaque paragraphe écrit par un écrivain il ajoute une note datée.

En implémentation, l'écrivain n'est pas directement lié au paragraphe.

C'est une forme contractée.