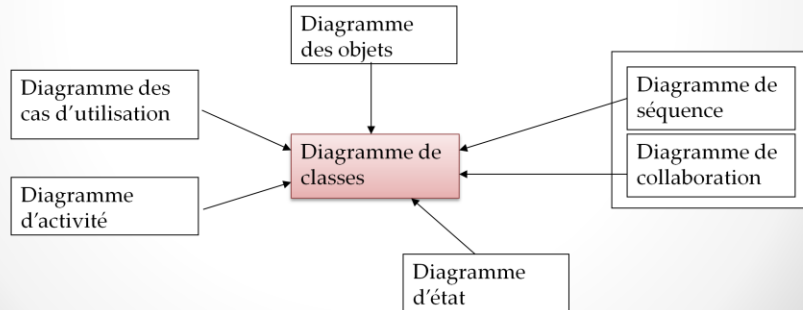


UML 2

- 3. La modélisation statique:
- Le diagramme de classes

Le diagramme de classes

- Représente les classes, leurs composants et la façon dont les classes sont liées les unes aux autres.
- Plusieurs autres diagrammes permettent de compléter le diagramme de classes.

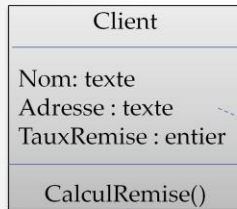


Les éléments de base

Une classe est une description abstraite d'un ensemble d'objets ayant les mêmes caractéristiques.

→ Les classes nécessaires dans un cas d'utilisation

→ Les classes associées à un scénario précis



Les attributs:

les types d'information contenus dans la classe.

En dehors du nom et du type de l'attribut, on peut également spécifier :

- La visibilité (+, -, #, ~)
- Un marqueur de dérivation
- Une opérateur d'affectation et valeur par défaut
- Des contraintes

Les opérations:

Des éléments de comportement contenu dans une classe.

Elles seront identifiées au moment de la conception.

● Carina Roels

● 3

Les attributs

❑ La visibilité d'un attribut peut être :

- **Public** + : l'accès est possible pour tous les objets de toutes les classes
- **Privé** - : l'accès n'est possible que depuis l'intérieur des objets. Autrement dit: seules les opérations de la classe peuvent manipuler l'attribut.
- **Protégé** # : L'accès est possible pour les objets d'une sous-classe. Dans un héritage, les sous-classes doivent avoir accès aux attributs de la superclasse.
- **Package** ~ : L'accès est possible pour les objets appartenant au même package.

❑ **Le marqueur de dérivation** / : indique que la valeur de l'attribut est calculée en utilisant d'autres valeurs et des règles/formules de calcul.

❑ **L'opérateur d'affectation et la valeur par défaut** permet d'affecter une valeur à certains attributs.

Cela facilite l'utilisation de l'attribut et fait en sorte que les attributs obligatoires aient une valeur, même si l'utilisateur n'en précise pas.

❑ **Les contraintes** permettent d'exprimer les règles nécessaires pour garantir l'intégrité de l'attribut. A chaque modification de la valeur de l'attribut, les règles devront être vérifiées. Ces règles sont généralement implémentées dans toutes les méthodes qui doivent modifier la valeur de l'attribut.

Les opérations sont généralement indiquées avec la syntaxe :

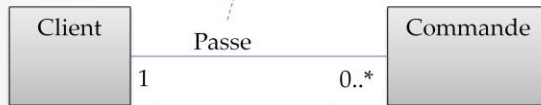
visibilité NomOpération (NomArgument {contraintes}, ...) : typeValeurRetour {contraintes}

Les arguments doivent être indiqués avec *nom:type*

Les éléments de base

Association :

Lien bidirectionnel (même si le verbe semble indiquer un sens) entre deux classes.



Multiplicités :

Indication du nombre d'objets qui peuvent participer à l'association. Ce nombre est exprimé en termes de nombre minimum et nombre maximum.

● Carina Roels

● 4

Les associations en UML sont représentées par des traits continus entre les classes. Pour plus de renseignements, il est utile de nommer les associations. Pour cela, le nom de ces dernières se met au milieu de la ligne qui symbolise l'association. Le sens de lecture de l'association peut être précisé au moyen des symboles "<" et ">".

Les **multiplicités** des associations indiquent combien d'objets de la classe considérée participent à la relation et peuvent être liés avec un objet de l'autre classe. UML définit la syntaxe suivante :

1	un et un seul
0..1	zéro ou un
M .. N	de M à N (où M < N)
*	plusieurs
0..*	de zéro à plusieurs
1..*	de un à plusieurs

Dans l'exemple, il est indiqué :

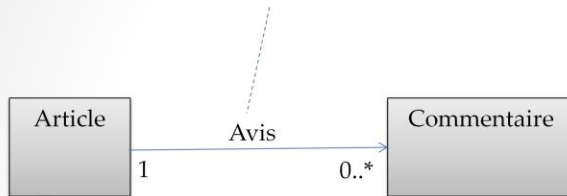
Qu'un client peut ne pas passer de commandes (multiplicité minimum = 0), mais peut également en passer plusieurs (multiplicité maximum = *),

Qu'une commande est passée par un et un seul client (multiplicité = 1).

Les associations navigables

Association navigable :

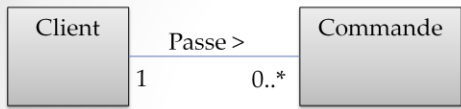
Lien unidirectionnel entre deux classes.



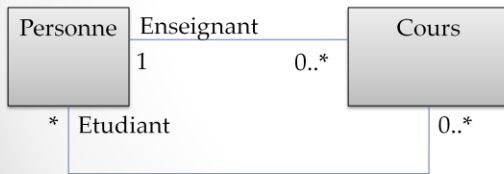
Exemple:

Connaissant un article on connaît les commentaires,
mais pas l'inverse..

Les précisions d'une association



Association en verbe active :
précision du sens de la lecture



Rôle:
précision de la fonction d'une classe pour une association donnée.
Utile en cas d'associations multiples entre deux classes ou dans des associations réflexives.

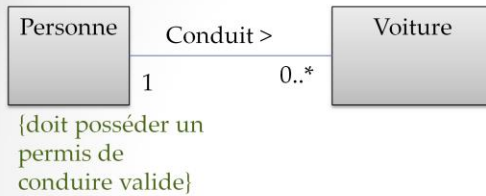
Le sens de la lecture peut avoir une importance pour la compréhension.

En cas de nombreuses modifications à un diagramme de classes, il est courant d'inverser la position des classes. Sans indication du sens de la lecture, on peut induire le lecteur en erreur.

Le rôle que joue une classe dans une association peut servir :

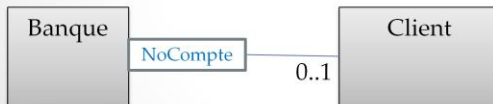
- A remplacer le nom de l'association, lorsque celui-ci n'est pas évident à trouver.
Par exemple : quel serait l'association entre un parent et un enfant?
- A indiquer une différence pour n associations entre les deux mêmes classes.
Comme le montre l'exemple : une personne peut être enseignant ou étudiants dans un cours. Il est possible d'indiquer le rôle à une extrémité d'une association ou aux deux.

Les précisions d'une association



La contrainte:

permet de préciser une règle qui doit être respectée pour que l'association puisse exister.



La qualification:

permet de sélectionner un sous-ensemble d'instances de l'autre classe

Une contrainte se place à l'extrémité d'une association, du côté de la classe à laquelle elle s'applique.

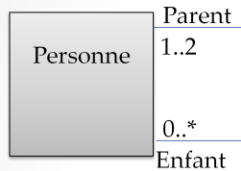
Il est possible d'indiquer des contraintes aux deux extrémités d'une association.

La qualification peut être vu comme 'une sorte d'index'.

Le qualificateur est placé du côté de la classe qui utilise la valeur pour la recherche. Normalement, le qualificateur est un attribut de la classe à l'autre extrémité de l'association (noms identiques).

Dans l'exemple, on indique qu'une banque utilise un numéro de compte pour retrouver un client. On doit donc retrouver l'attribut NoCompte parmi les attributs de la classe Client.

L'association réflexive



Association réflexive:
indication d'une association
entre plusieurs instances d'une
même classe.

Dans l'exemple :

- Une personne est parent de 0 à plusieurs autres personnes.
- Une personne est enfant d'une ou de 2 autres personnes.

Le rôle précisé sur l'association facilite la compréhension du modèle.

Une association est réflexive lorsqu'elle met en relation différents objets d'une même classe.

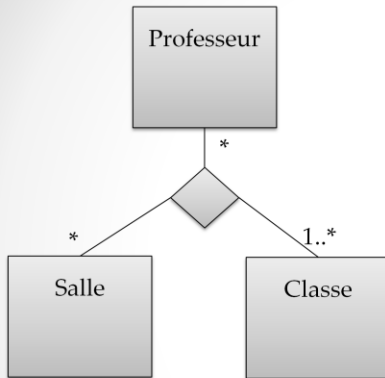
Lorsque le nom de l'association n'est pas évident à trouver, on peut indiquer les rôles à chaque extrémité de l'association.

Autre exemple :

Pour indiquer qu'un produit puisse être le résultat de l'assemblage d'autres produits on utiliserait :

- Une classe **produit**
- Une association réflexive
- Soit un nom d'association de type **composer**, soit des rôles de type **composé / composant**

L'association n-aire



Une association n-aire permet de relier plus de 2 classes.

L'exemple exprime qu'un professeur enseigne à des classes dans une salle définie.

Ces associations ne sont pas aussi facilement compréhensibles que les associations entre 2 classes.
Il est généralement préférable d'introduire d'autres éléments (exemple : ajout d'une classe d'association).

● Carina Roels

● 9

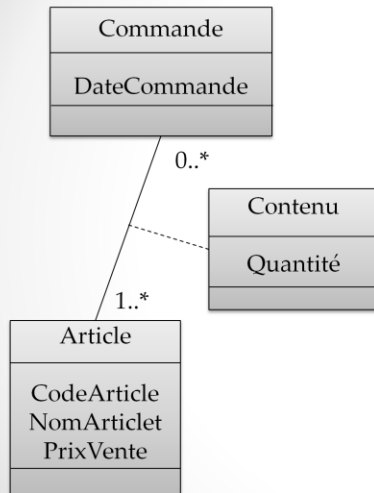
Les traits relient les classes à un losange central.

Le losange peut également accueillir une classe-association.

Une instance du losange =

Ensemble contenant une seule instance de chacune des classes liées.

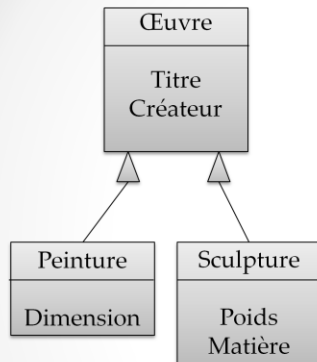
La classe d'association



La classe d'association permet de définir le lien qui existe entre les instances des autres classes de l'association.

Dans l'exemple, l'association exprime qu'une commande contient 1 voire plusieurs produits. Pour chaque article la quantité commandée peut être différente.

L'héritage



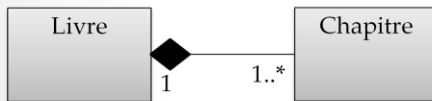
La spécialisation (démarche descendante)
La classe générique comporte un ensemble d'attributs/opérations commun à tous les objets de la classe. En identifiant des éléments particuliers pour un sous-ensemble d'objets, on crée des sous-classes (spécialisées).

La généralisation (démarche ascendante)
En observant des éléments communs pour des objets de classes différentes on peut décider de factoriser ces éléments dans une super-classe.

L'agrégation / la composition



L'agrégation permet d'indiquer que les objets des classes **peuvent être assemblés** pour créer des objets plus complexes.
Dans l'exemple
Un joueur **peut faire partie** d'une équipe.



La composition indique que les objets des classes dépendent les uns des autres.
Dans l'exemple, un chapitre **n'existe que** par rapport à un livre.

NOTE

Les multiplicités démontrent le choix entre l'agrégation et la composition.

● Carina Roels

● 12

L'agrégation est une forme particulière d'association.

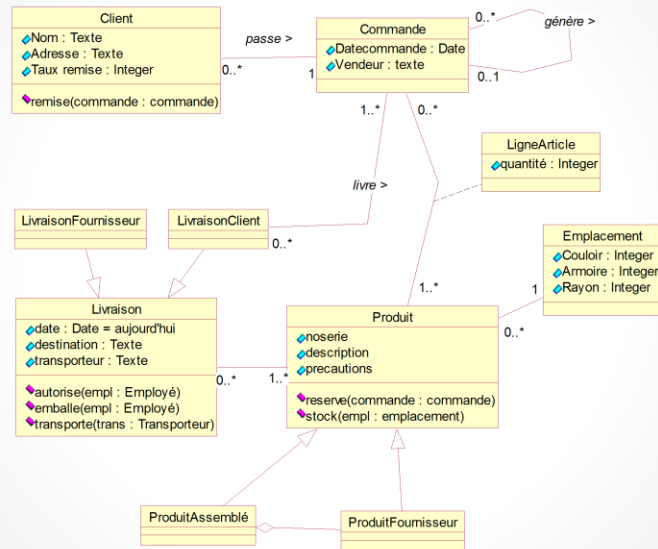
La composition est une agrégation forte.

La durée de vie des parties dépend de celle de l'agrégation. Autrement dit, les objets membres ne peuvent pas exister en dehors de l'agrégation.

Les exemples :

- Un joueur peut faire partie d'une équipe. Il peut également exister en dehors d'une équipe.
Lorsqu'on détruit un objet de type équipe, les joueurs continuent d'exister.
- Un chapitre existe forcément dans un livre. Il ne peut donc pas y avoir de chapitre en dehors d'un livre.
Par conséquent, lorsqu'on détruit un livre, on détruit également les chapitres.

Exemple d'un diagramme de classes



Le diagramme des objets

Aide à la compréhension d'un problème.
Vérification de la validité du diagramme de classes.

Diagramme de classe

- Modélise les règles concernant les types d'entités
- 1 classe = 3 compartiments (nom / attributs / opérations)
- Le nom de la classe est seul dans son compartiment
- Le compartiment des attributs : propriétés des attributs
- Les classes sont liées par des associations possédant un nom, des multiplicités, des contraintes, des rôles

Diagramme d'objets

- Modélise les faits réels
- 1 objet = 2 compartiments (nom / attributs)
- Nom d'objet : nom de classe Ex. 1234 :Commande
- Le compartiment des attributs : uniquement les valeurs pour les tests
- Les objets sont liés par de simples liens

Exemple de diagramme d'objets

