



Analyse et Conception Orientées Objets

Cours 2 : Diagramme de classes

Diagramme de classes

- Une collection d'éléments de modélisation statiques qui montre la structure d'un modèle.
- Classes : nom, attributs, opérations
- Relations entre classes (de la plus faible jusqu'à la plus forte) :
 - Dépendance
 - Association
 - Agrégation
 - Composition
 - Généralisation

Classe : la visibilité des attributs

- Un attribut d'une classe peut être
 - **private** (-) : accessible que dans les opérations de la classe
 - **public** (+) : accessible dans toutes les opérations
 - **protected** (#) : accessible que dans les opérations de la classe et les classes dérivées
 - **package** (~) : accessible que dans les opérations des classes dans le même paquetage.

Example

Employee
-id : int -name : string -telephone : string -email : string = "no email" #password : string

Classe : l'exécution des opérations

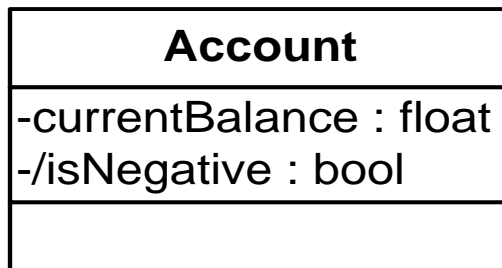
- Une opération d'une classe peut être
 - **private** (-) : *appelable* que par les opérations de la classe
 - **public** (+) : *appelable* par toutes les opérations
 - **protected** (#) : *appelable* que par les opérations de la classe et les classes dérivées
 - **package** (~) : *appelable* que par les opérations des classes dans le même paquetage.

Example

Employee
-ID[1] : int -Name[1] : string -Telephone[1] : string -Email[1..5] : string = "none" #Password[1] : string
+getID() : int +getName() : string +getTelephone() : string +getEmail(in index : int) : string +setTelephone(in phone : string) #setPassword(in pwd : string)

Classe : les attributs dérivés

- Un attribut peut-être constitué à partir d'autres attributs. On utilisera le symbole /
- Exemples :
 - longueur
 - largeur
 - /surface

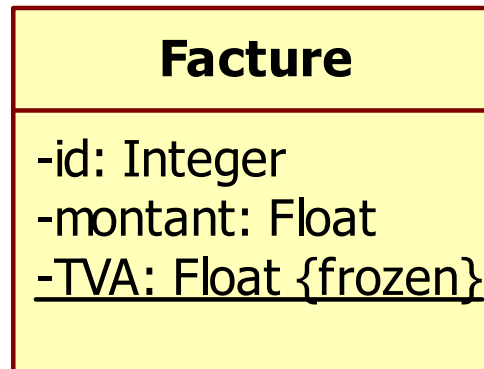


isNegative == true
iff
currentBalance < 0

Les attributs de classe

- Un **attribut de classe** est un attribut qui est commun à toutes les instances de la classe
- Graphiquement, un attribut de classe est souligné (équivalent à **static** en Java ou C++)
- Exemple : le taux de TVA dans la classe Facture

-



Les opérations de classe

- Une opération de classe est une opération qui est appellable sans passer par une instance
- Graphiquement, la méthode de classe est soulignée
- Exemples :
 - Les opérations qui fabriquent de nouveaux objets
 - Les opérations qui modifient les attributs de classes

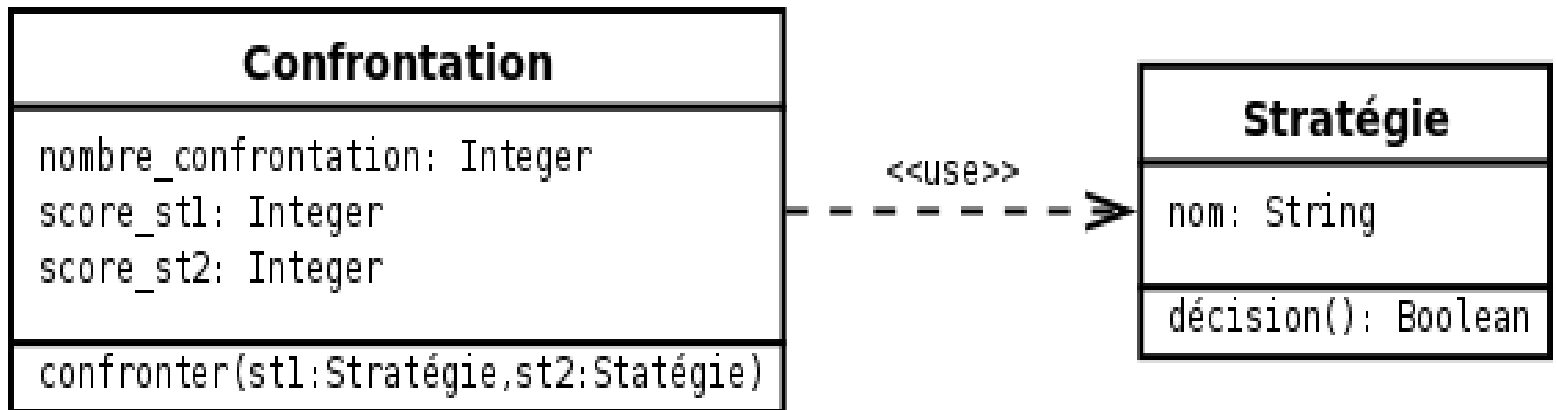
Facture
-id: Integer -montant: Float <u>-TVA: Float {frozen}</u>
<u>+creerFacture(id: Integer, montant: Float): Facture</u> <u>+getTVA(): Float</u> +getId(): Integer +getMontant(): Float

Relations entre classes

- De la plus faible jusqu'à la plus forte :
 - Dépendance
 - **Association**
 - **Agrégation**
 - **Composition**
 - **Généralisation**

Les relations de dépendance

- Une dépendance (**dependency**) est une relation faible entre classes
- Classe A **dépend** de la classe B si A **utilise** B dans une de ces opérations



Associations entre classes

- Une connexion sémantique entre deux classes
- Caractéristiques :
 - **navigabilité** : uni ou bidirectionnelle
 - **cardinalités** : précise le nombre d'instances qui participent à une relation.
 - **rôles** : spécifie la fonction d'une classe pour une association donnée.
 - **qualification** : restreint la portée de l'association à quelques éléments ciblés de la classe.
 - **classe d'association**

Association : navigation



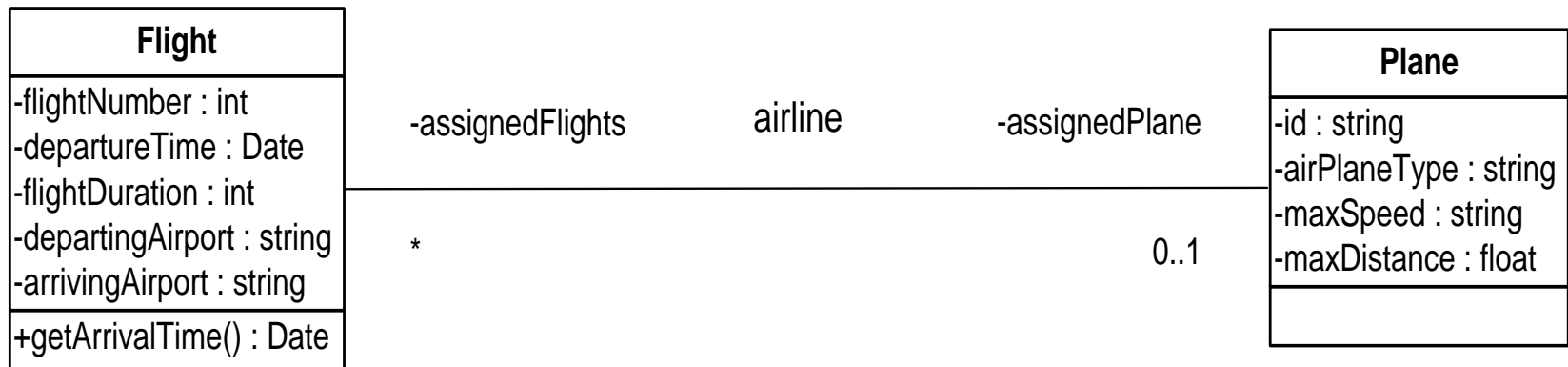
- La **flèche** indique la possibilité de **naviguer** de A à B
- La **navigabilité** de la classe A à la class B indique qu'une instance de la classe A peut accéder aux instances associées de la classe B.
- Quand l'association peut naviguer dans les deux directions, alors aucune flèche n'est dessinée.

Les rôles d'association



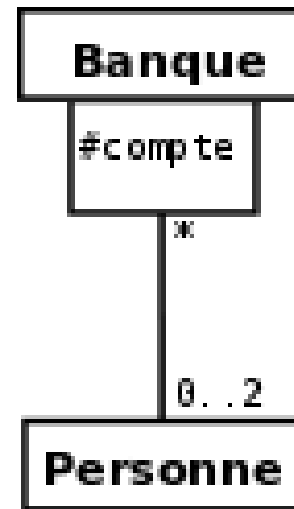
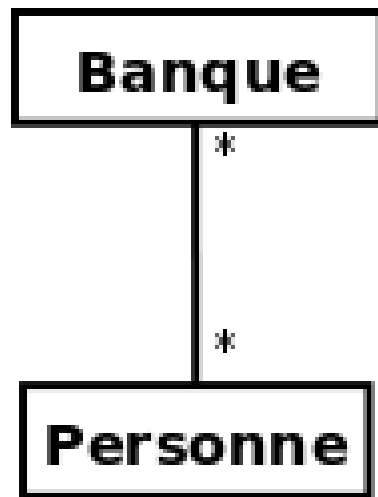
- employé et employeur sont les rôles (terminaisons) de l'association travailler pour

D'autres exemples

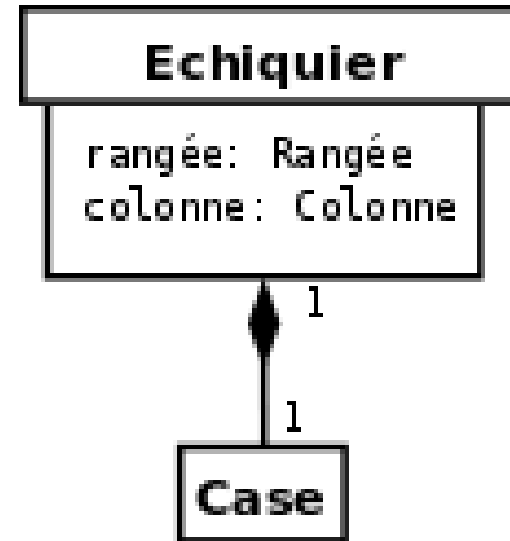
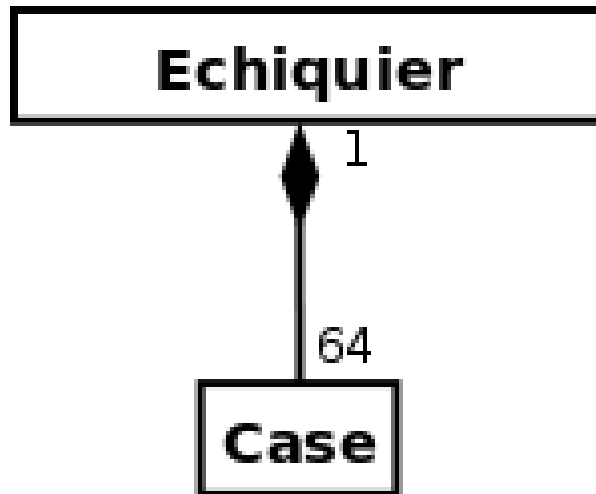


Qualification (association qualifiée)

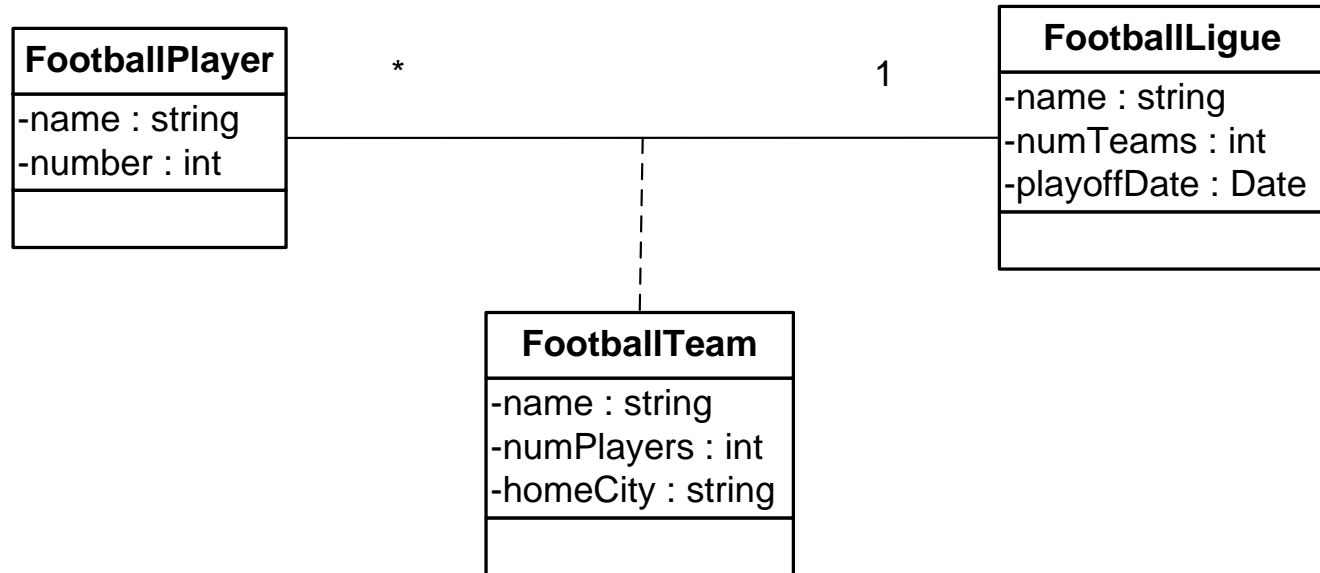
- Restreindre la portée de l'association à quelques attributs ciblés de la classe.



Un autre exemple de qualification

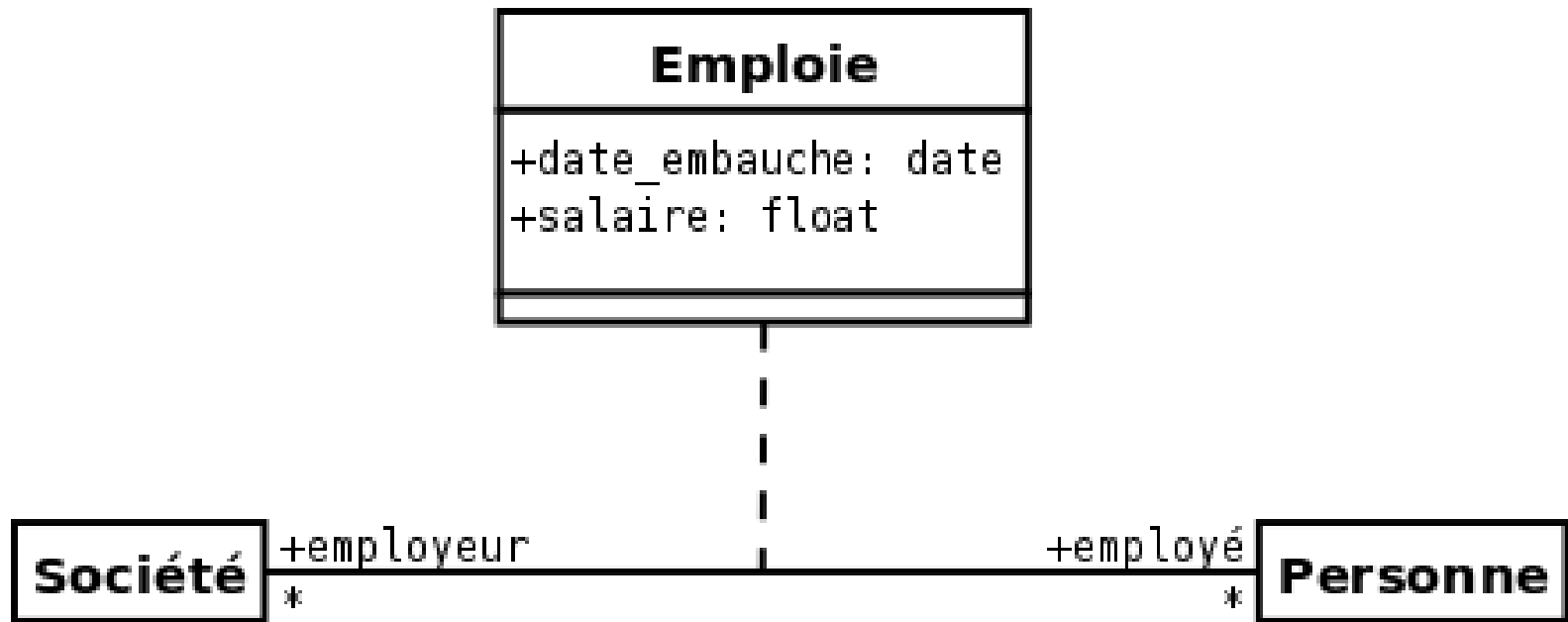


Les classes d'association

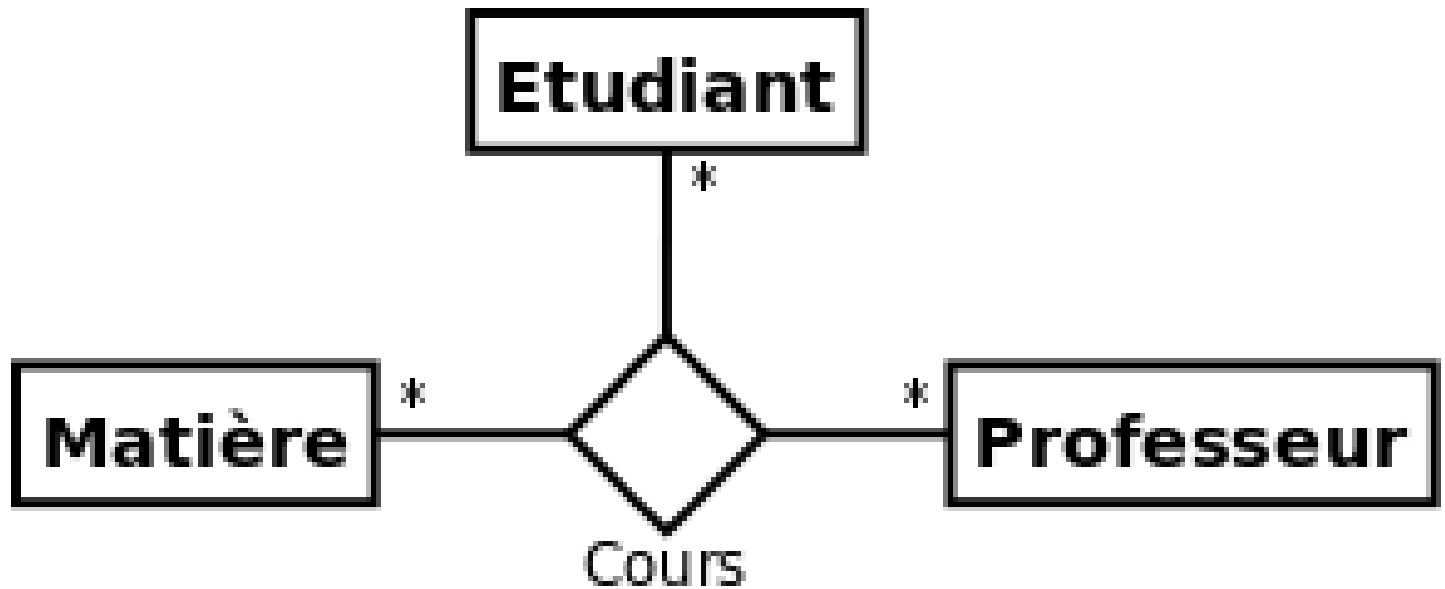


- Dans certaines associations, à chaque instance du couple, on associe un objet d'une autre classe. On dit que l'association est porteuse d'informations
- Exemple : Dans l'association qui relie la classe **FootballPlayer** à la classe **FootballLigue**, on définit la classe d'association **FootballTeam**

Un autre exemple de classe d'association



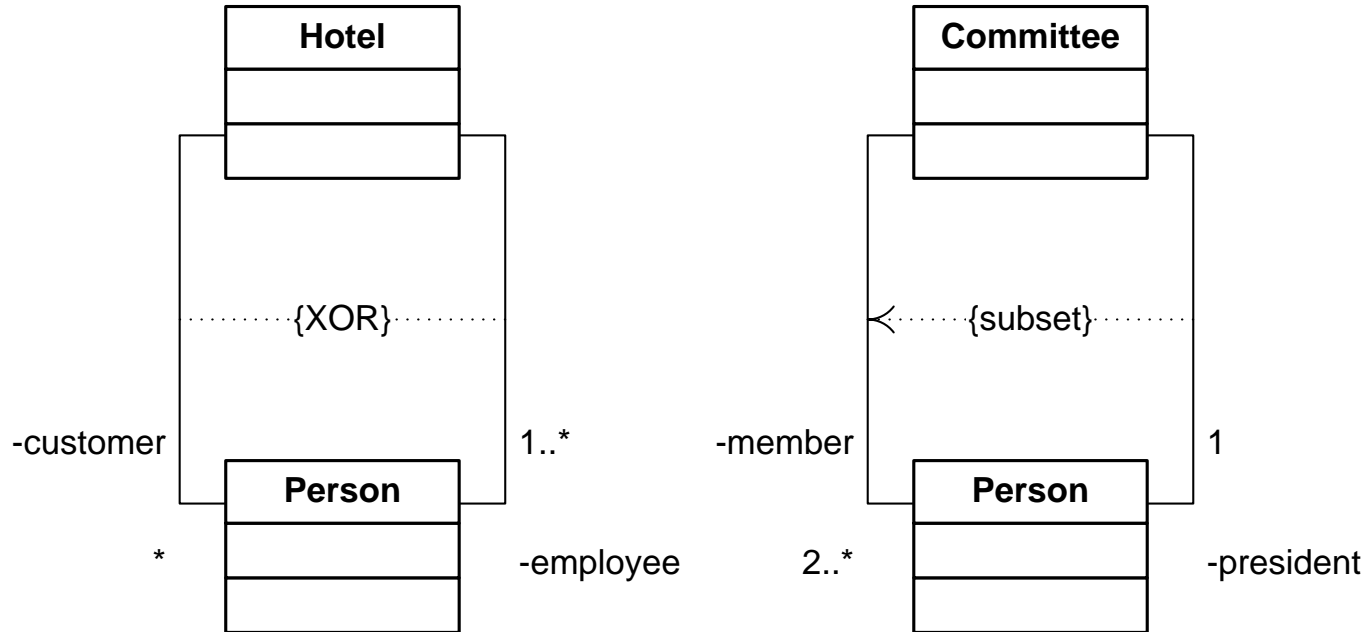
Association n-aire



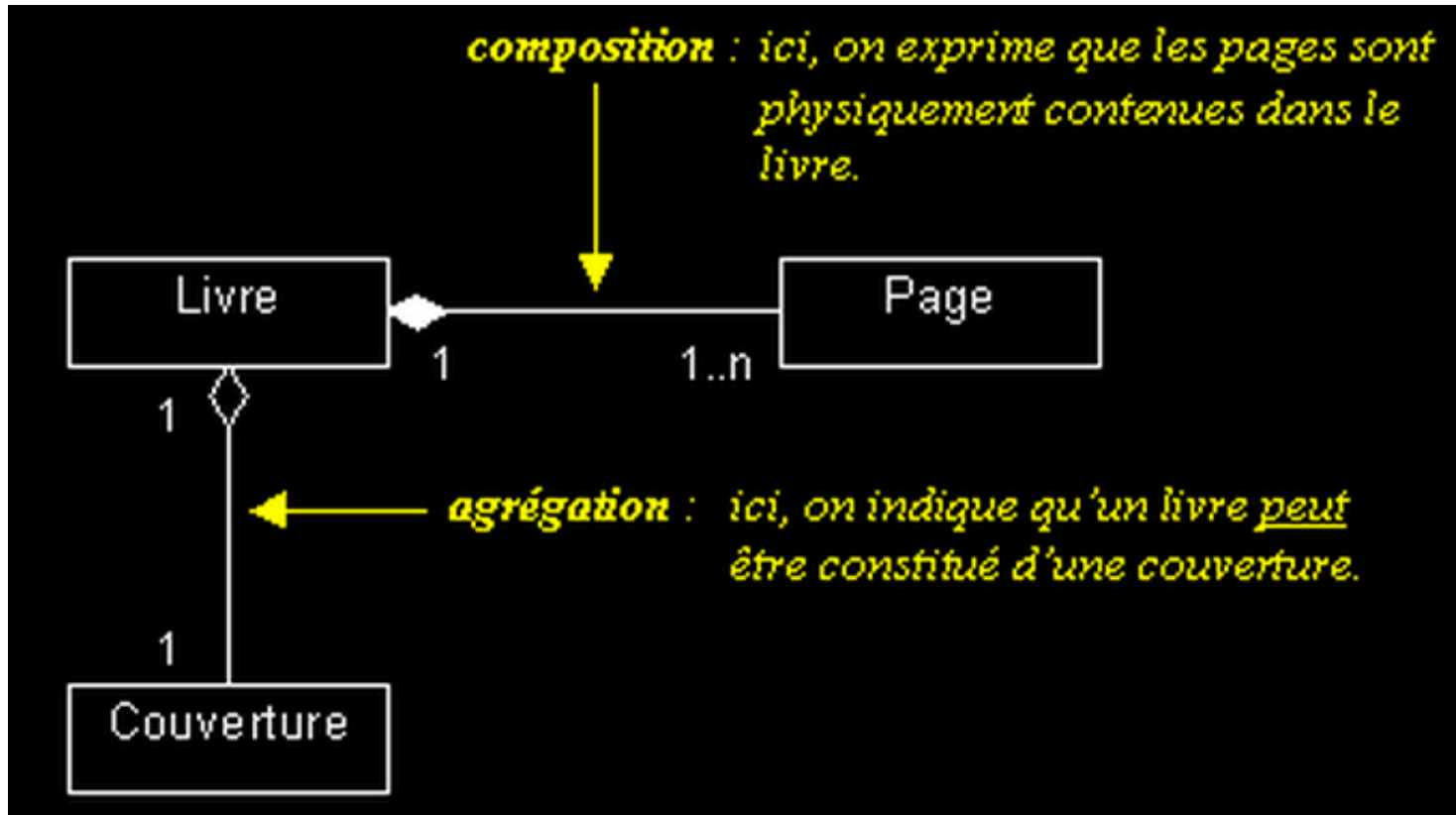
Contraintes d'association

- {XOR}
- {subsets <property_name>}
- {redefines <property_name>}
- {union}
- {ordered}
- {bag}
- {sequence} ou {seq}

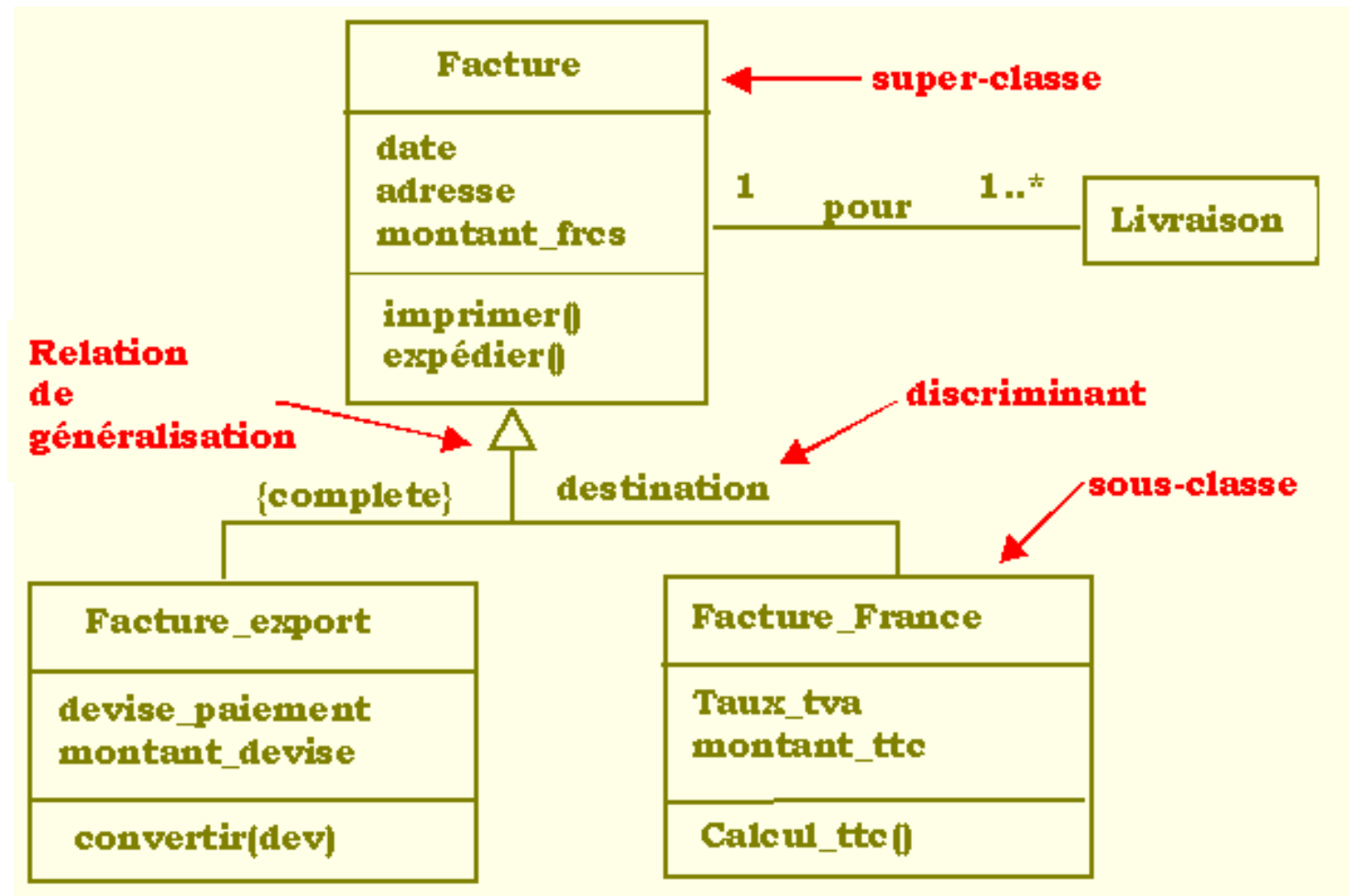
Contraintes d'association : exemples



Agrégation et Composition (uml.free.fr)



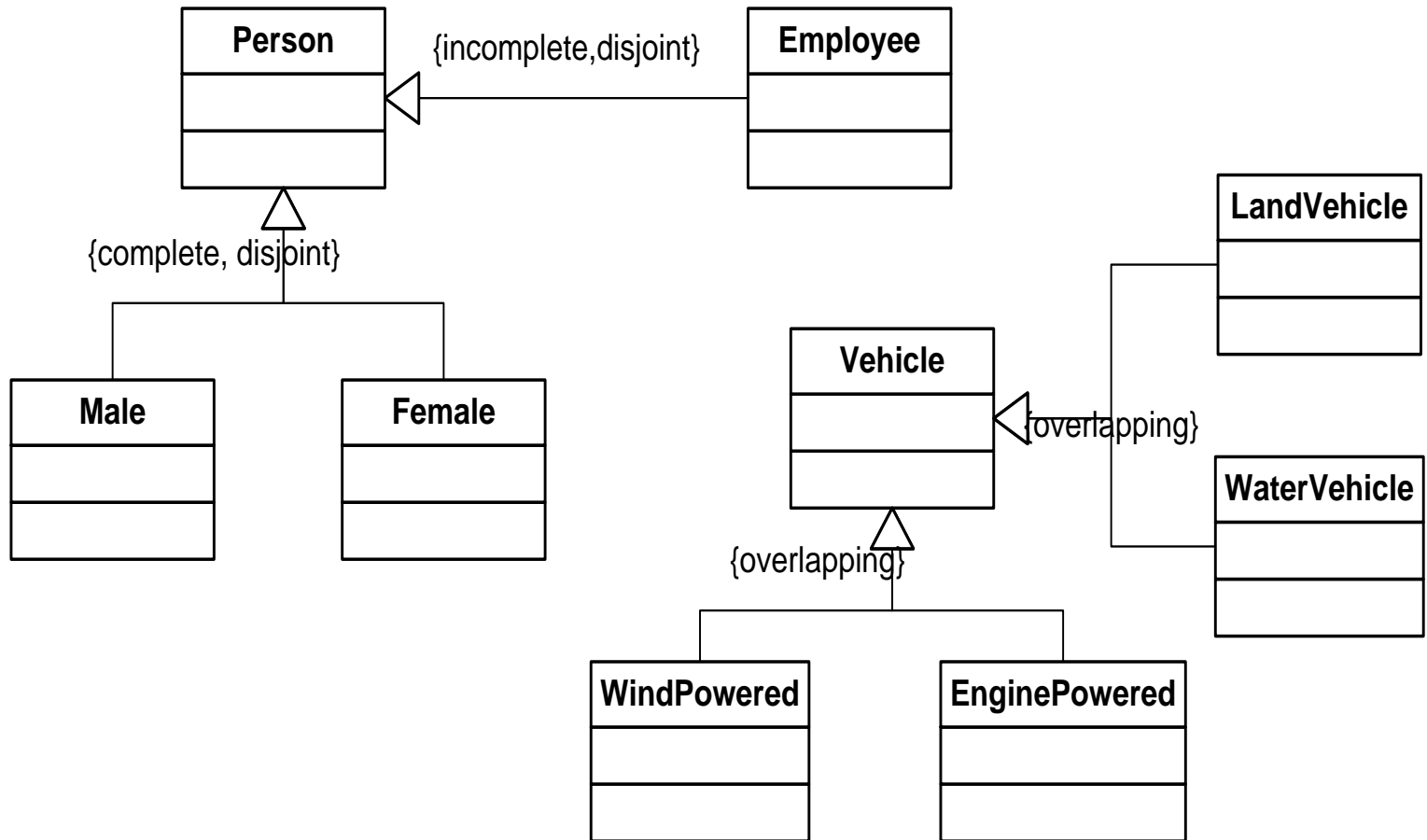
Généralisation



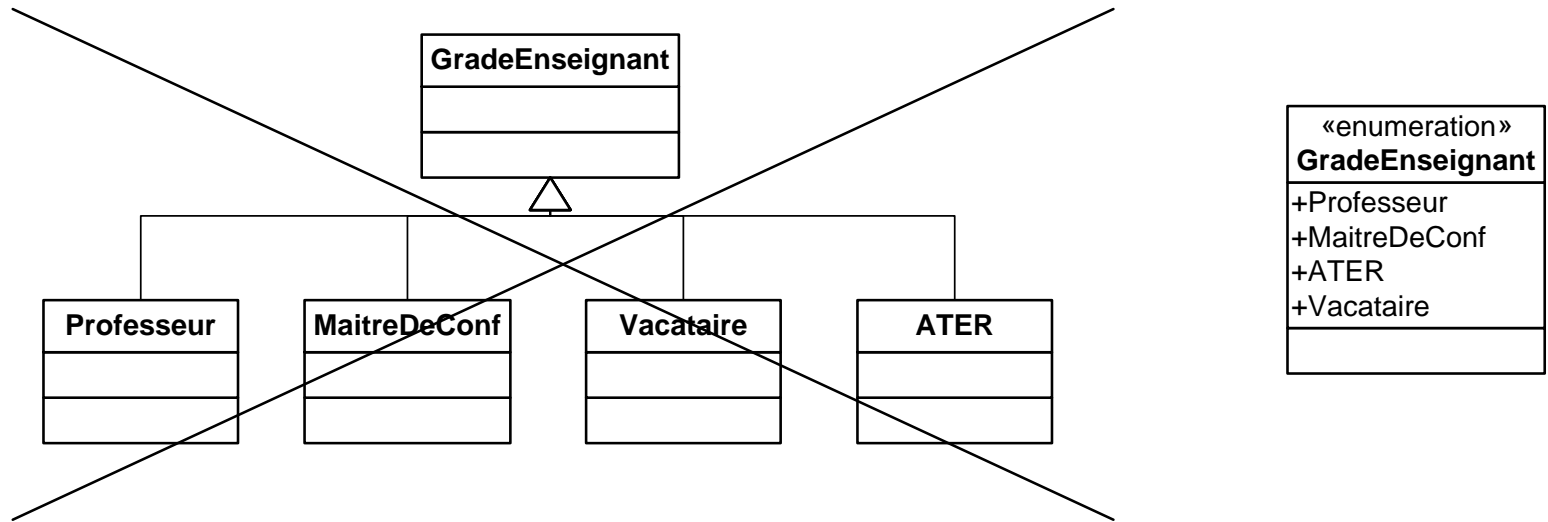
Contraintes de généralisation

- {complete, disjoint}
 - pas extensible, pas d'instance commune
- {incomplete, disjoint}
 - extensible, pas d'instance commune
- {complete, overlapping}
 - pas extensible, avec des instances communes
- {incomplete, overlapping}
 - extensible, avec des instances communes
- Par défaut : {incomplete, disjoint}

Contraintes de généralisation : exemples



Enumérations



- Enumérations sont des classes représentant des objets qui peuvent avoir un nombre fini des valeurs

Interface

- **Encapsulation** : un objet n'est accessible de l'extérieur qu'à travers ses opérations.
- La déclaration d'une opération d'un objet est constituée :
 - du nom de l'opération
 - ses paramètres
 - du retour

Cet ensemble est la **signature de l'opération**.

- L'ensemble des signatures des **opérations publiques** d'un objet est appelée **interface de l'objet**.

Interface

- L'interface est la vue externe d'un objet, elle définit les services accessibles (offerts) aux utilisateurs de l'objet.
 - Masquer les détails d'implémentation d'un objet (**encapsulation**)
 - Garantir l'intégrité des données contenues dans l'objet.

Interface : exemple

