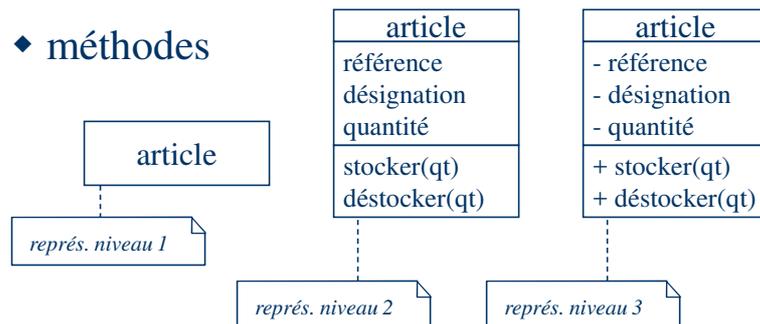


Représentations d'une classe

- ◆ nom et caractéristiques générales
- ◆ attributs
- ◆ méthodes



Attributs d'une classe

- ◆ Les attributs (niveaux 2 et 3) ont un type et éventuellement un espace de nommage

<<espace de nommage>><étiquette><nom>:<type>

désignation : String

qtéEnStock : int

- ◆ Les attributs de classe sont soulignés ou précédés de l'étiquette \$

taux1TVA : float

\$taux1TVA : float

Méthodes d'une classe

- ◆ Les méthodes (niveaux 2 et 3) ont un type et des paramètres typés

```
<<espace de nommage>><étiquette>  
<nom>(<liste de paramètres typés>):<type retourné>
```

```
ACommander() : boolean  
stocker(qtéAjoutée : int) : void
```

- ◆ Les méthodes de classe sont soulignées ou précédées de l'étiquette \$

```
bissextile(année : int) : boolean
```

Accessibilité aux membres

- ◆ L'accessibilité (niveau 3) des membres est indiquée par une étiquette
 - + pour public : tout objet a accès à ce membre
 - pour private : seuls les objets de la classe ont accès à ce membre
 - # pour protected : les objets de la classe et des classes dérivées ont accès à ce membre

Liaisons entre classes

- ◆ Dépendance (dépend de = utilise)



- ◆ Association

- non orientée



- orientée



Liaisons entre classes...

- ◆ Lien entre un ensemble et ses éléments

- agrégation



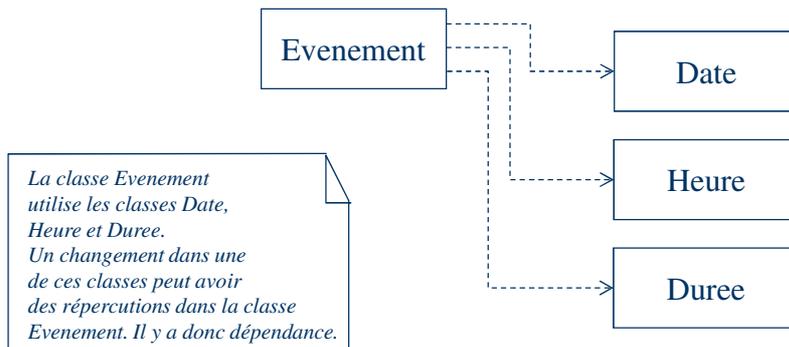
- composition



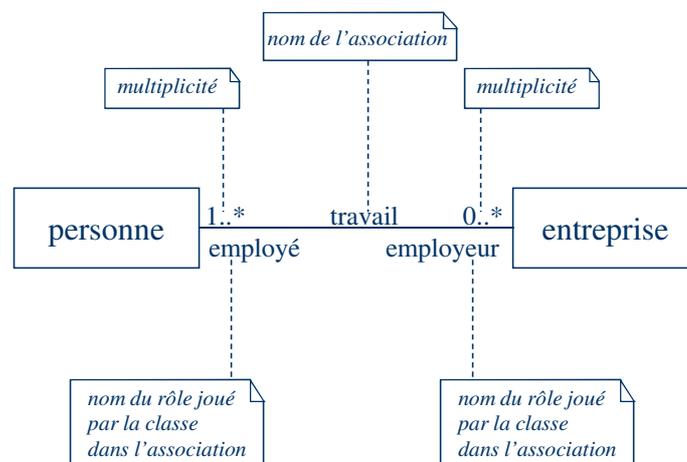
- ◆ Héritage (généralisation-spécialisation)



Dépendance entre classes



Association entre classes

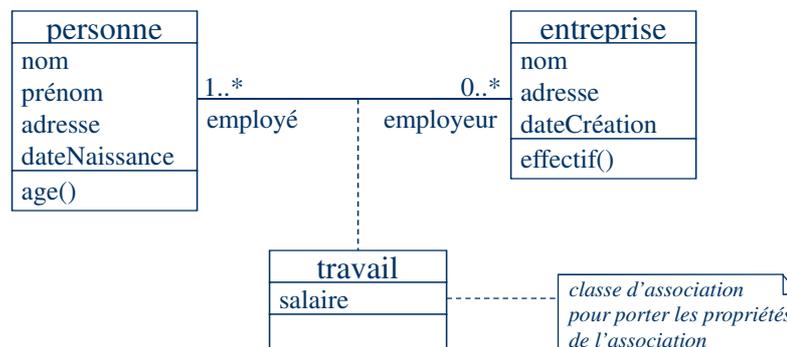


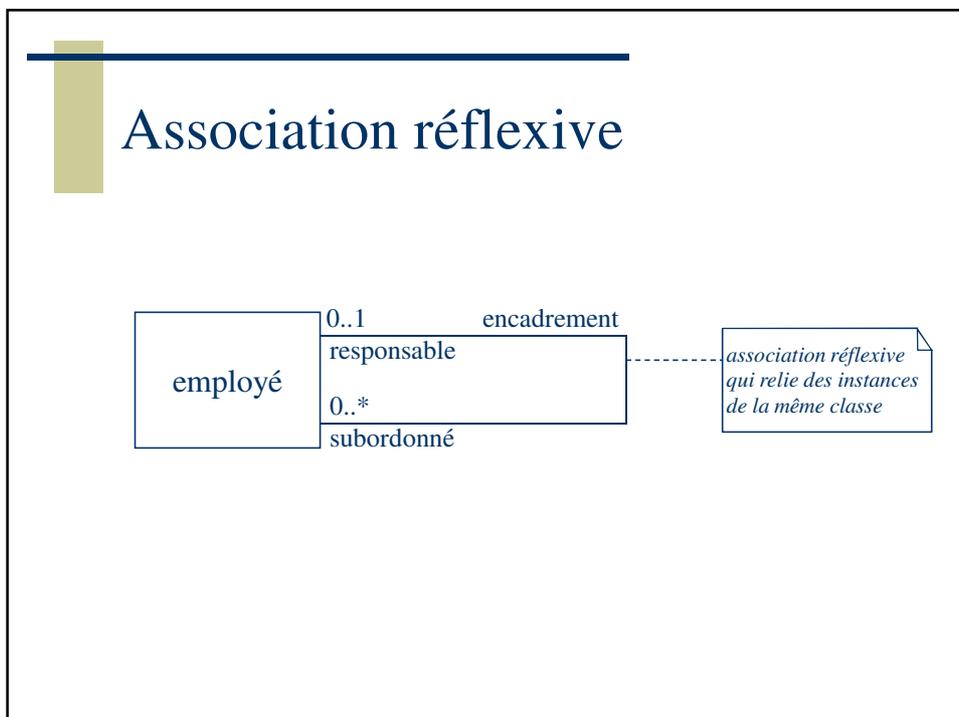
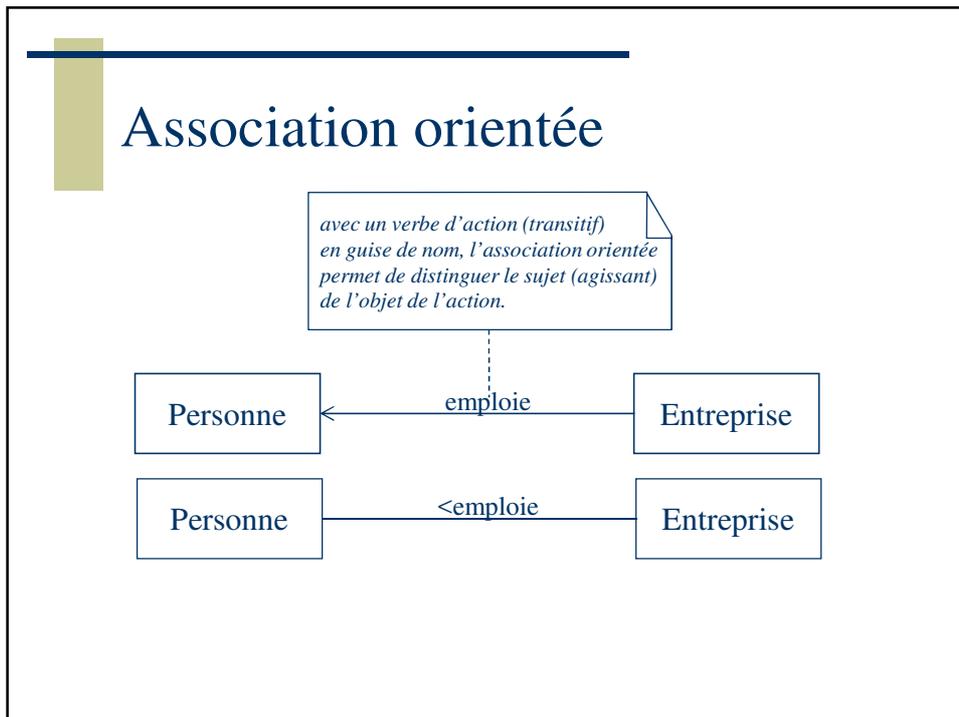
Multiplicités d'association

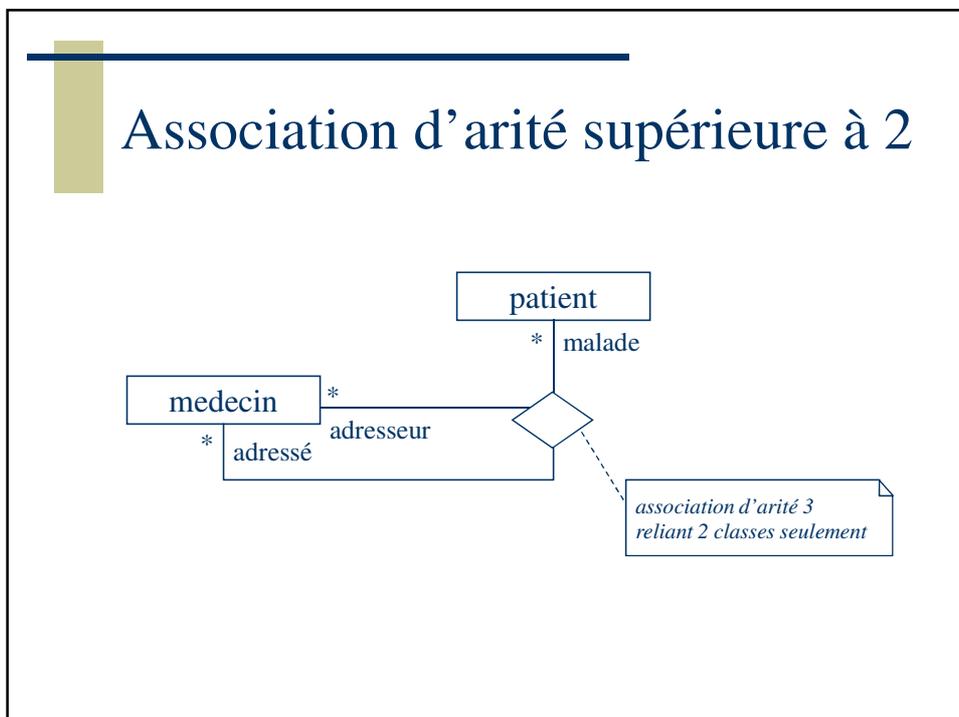
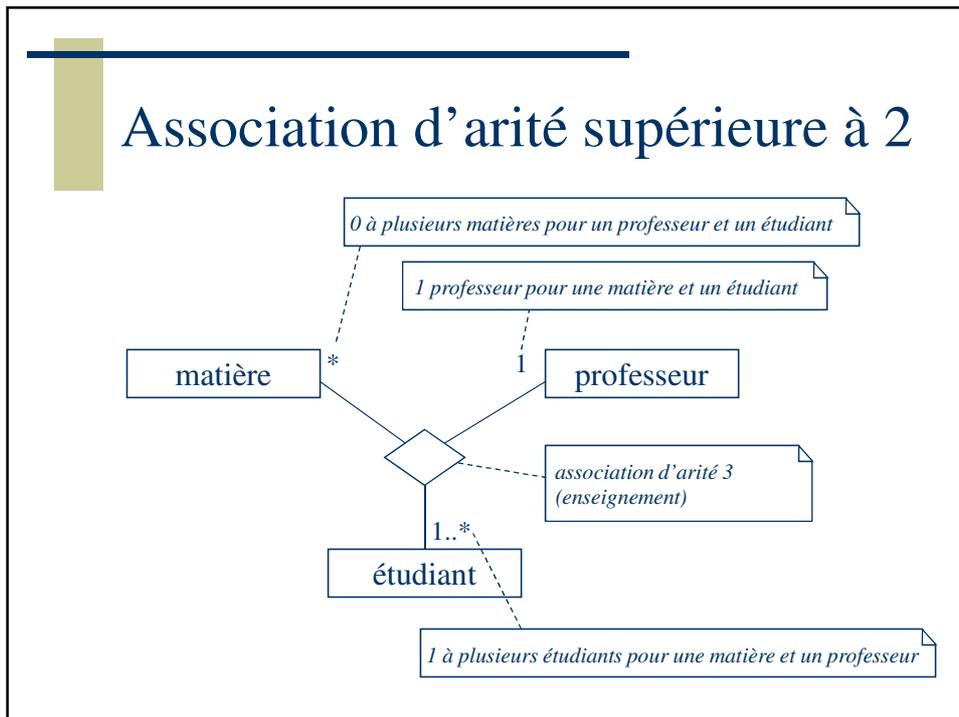
nombre d'instances
pouvant être liées par une extrémité d'association
à une instance pour chaque autre extrémité d'association

- 0..1 *0 à 1 instance*
- 0..* ou * *0 à plusieurs instances*
- 1 *1 instance exactement*
- 1..* *1 à plusieurs instances*
- n..m *n à m instances*
- n..* *n à plusieurs instances*

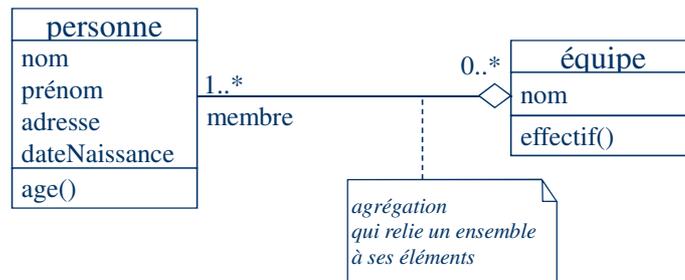
Classe d'association





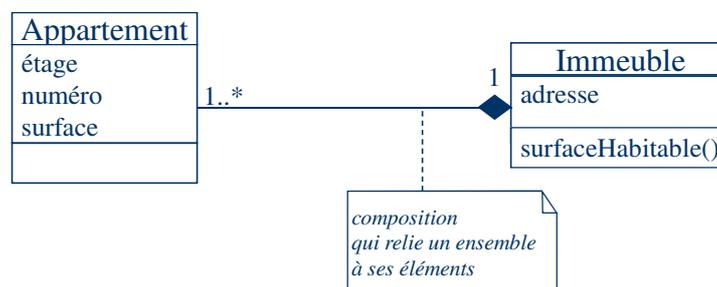


Agrégation



Une personne peut être membre de plusieurs équipes, éventuellement aucune.
 Une équipe compte au moins un membre.
 Les personnes membres d'une équipe demeurent même si l'équipe est « dissoute ».

Composition

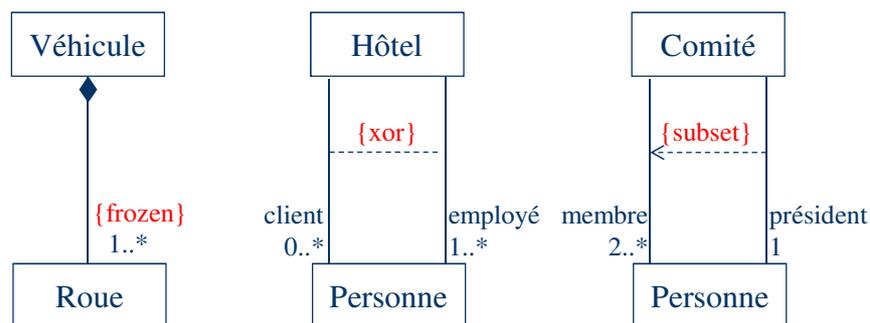


Un appartement est constituant (composant) d'un seul immeuble.
 Un immeuble compte au moins un appartement.
 Les appartements n'existent ni avant, ni après l'existence de l'immeuble qui les contient (ils disparaissent avec lui à sa démolition).

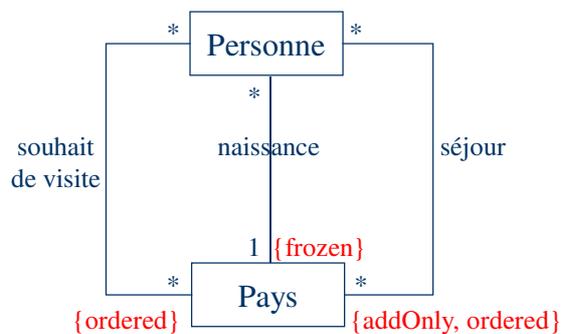
Contraintes

- ◆ {frozen}
la multiplicité est invariante ou l'association ne peut changer
- ◆ {xor}
les associations s'excluent mutuellement
- ◆ {subset}
l'ensemble des éléments liés par une association est inclus dans l'ensemble des éléments liés par une autre association
- ◆ {ordered}
les éléments associés sont ordonnés (préférence, priorité...)
- ◆ {addOnly}
l'ensemble des éléments liés ne peut que croître

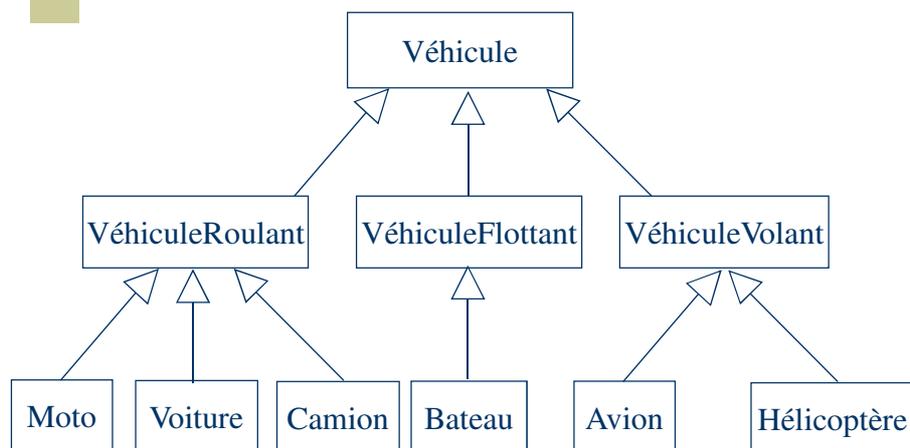
Exemples de contraintes



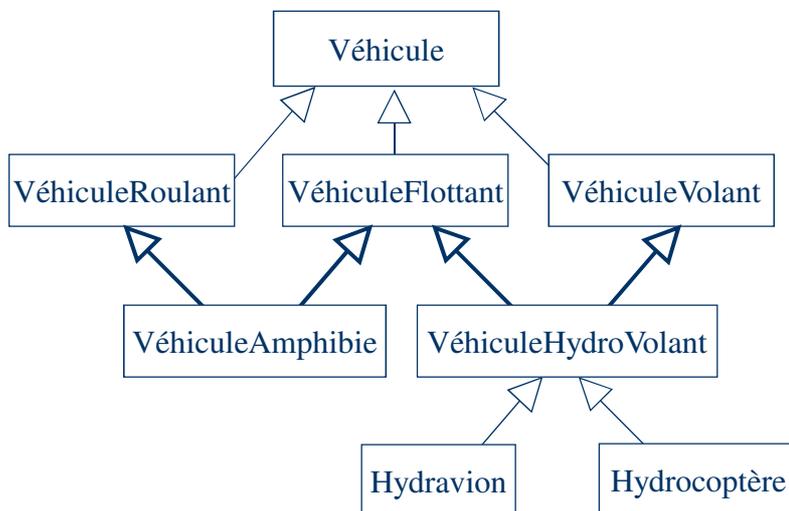
Exemples de contraintes ...



Exemple d'héritage



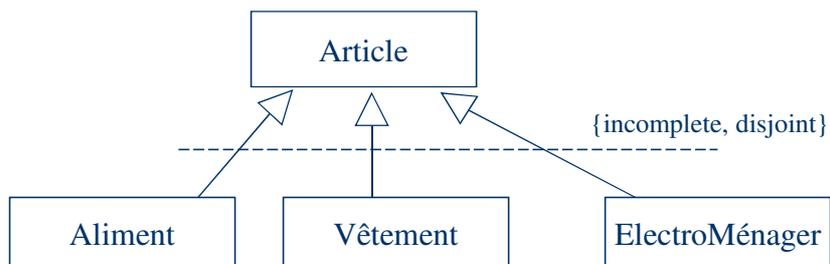
... héritage multiple



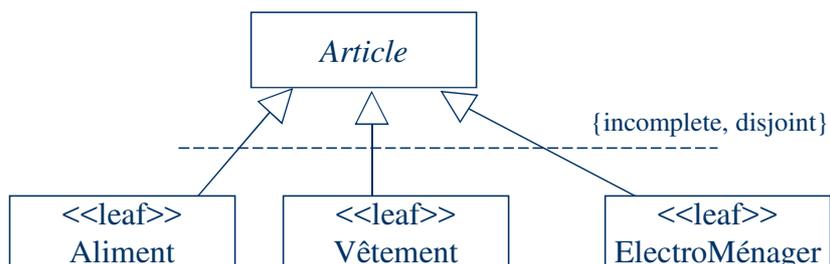
Contraintes et héritage...

- ◆ {complete}: il n'y a pas d'autres classes dérivées
- ◆ {incomplete} : les classes dérivées ne sont pas toutes représentées
- ◆ {overlapping} : un objet de ce type peut être instance de plusieurs classes dérivées de la classe de base
- ◆ {disjoint} : une instance de ce type ne peut être instance que d'une seule des classes dérivées de la classe de base
- ◆ {partition} = {complete, disjoint}

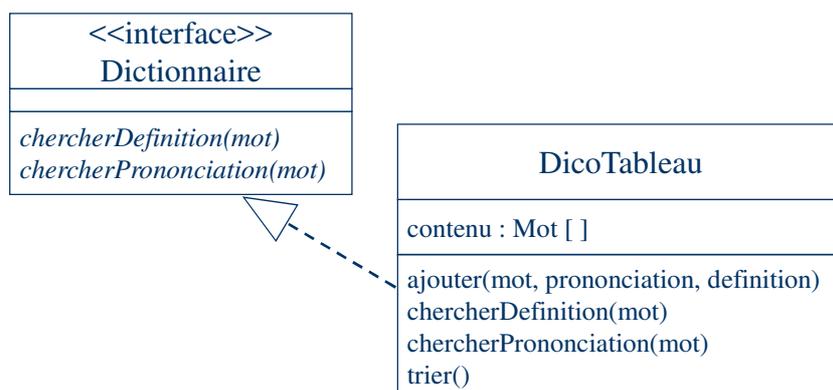
Héritage et contraintes



Héritage et stéréotypes



Implantation d'une interface



Implantation d'interfaces

