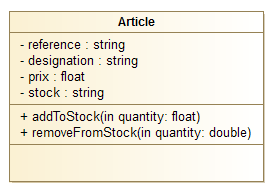
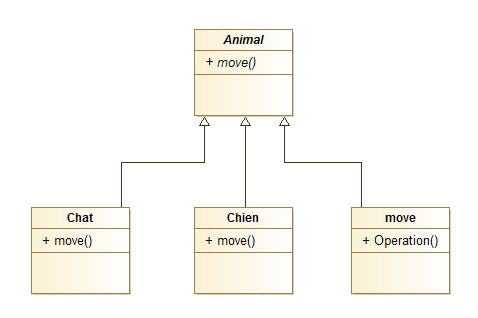
UML - Le diagramme de classe

# La classe

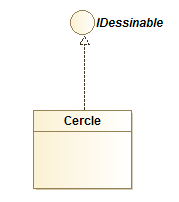


# Héritage



Note : *Animal* est une classe abstraite, son nom est *en italique*. De même, la méthode *move()* de la classe Animal est abstraite, son nom est *en italique*.

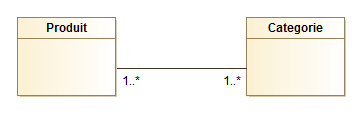
# Interface



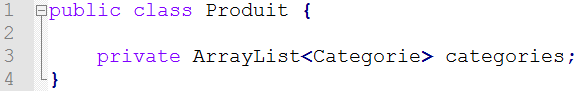
# Associations

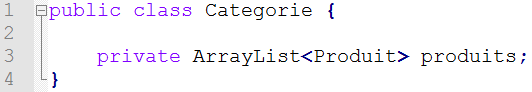
## Associations binaires

### Associations bidirectionnelles



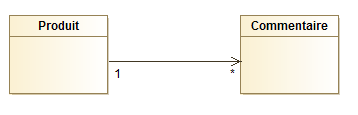
A éviter car elle complique le code.

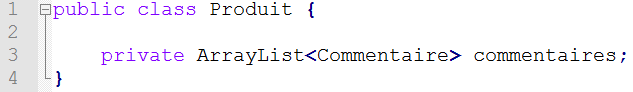




### Associations unidirectionnelles

#### Exemple n°1

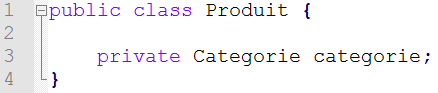




La classe Commentaire n’a pas connaissance de la classe Produit.

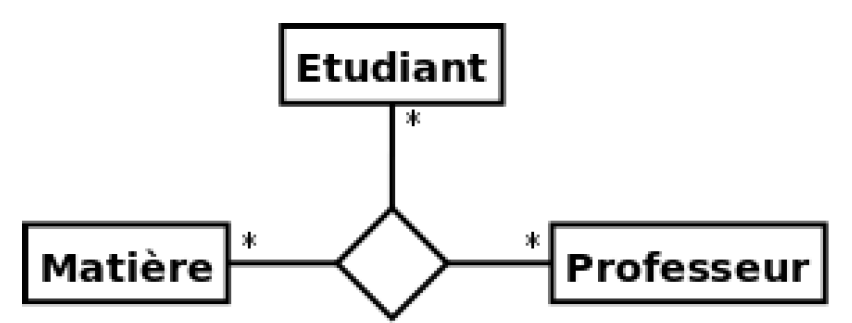
#### Exemple n°2



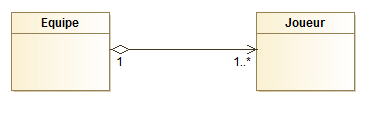


La classe Categorie n’a pas connaissance de la classe Produit.

## Associations n-aires

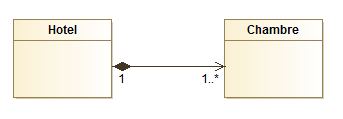


## Agrégation



En Java, l’agrégation se traduit comme une relation unidirectionnelle (voir sens de la flèche).

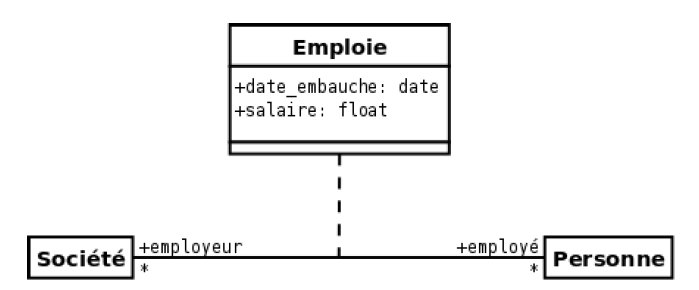
## Composition



En Java, la composition se traduit comme une relation unidirectionnelle (voir sens de la flèche).

**A noter** que la suppression du composite (ici Hotel) entraîne la suppression des composants (ici Chambre).

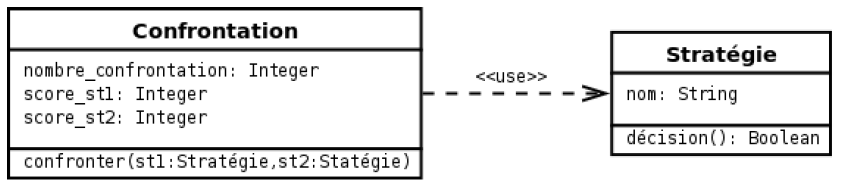
## Classe-association





Ces deux représentations sont équivalentes.

# Dépendance



Elle indique que la modification de la cible peut impliquer une modification de la source.

On utilise souvent une dépendance quand une classe en utilise une autre comme argument dans la signature d'une opération.