



Langage UML
Diagramme de classes

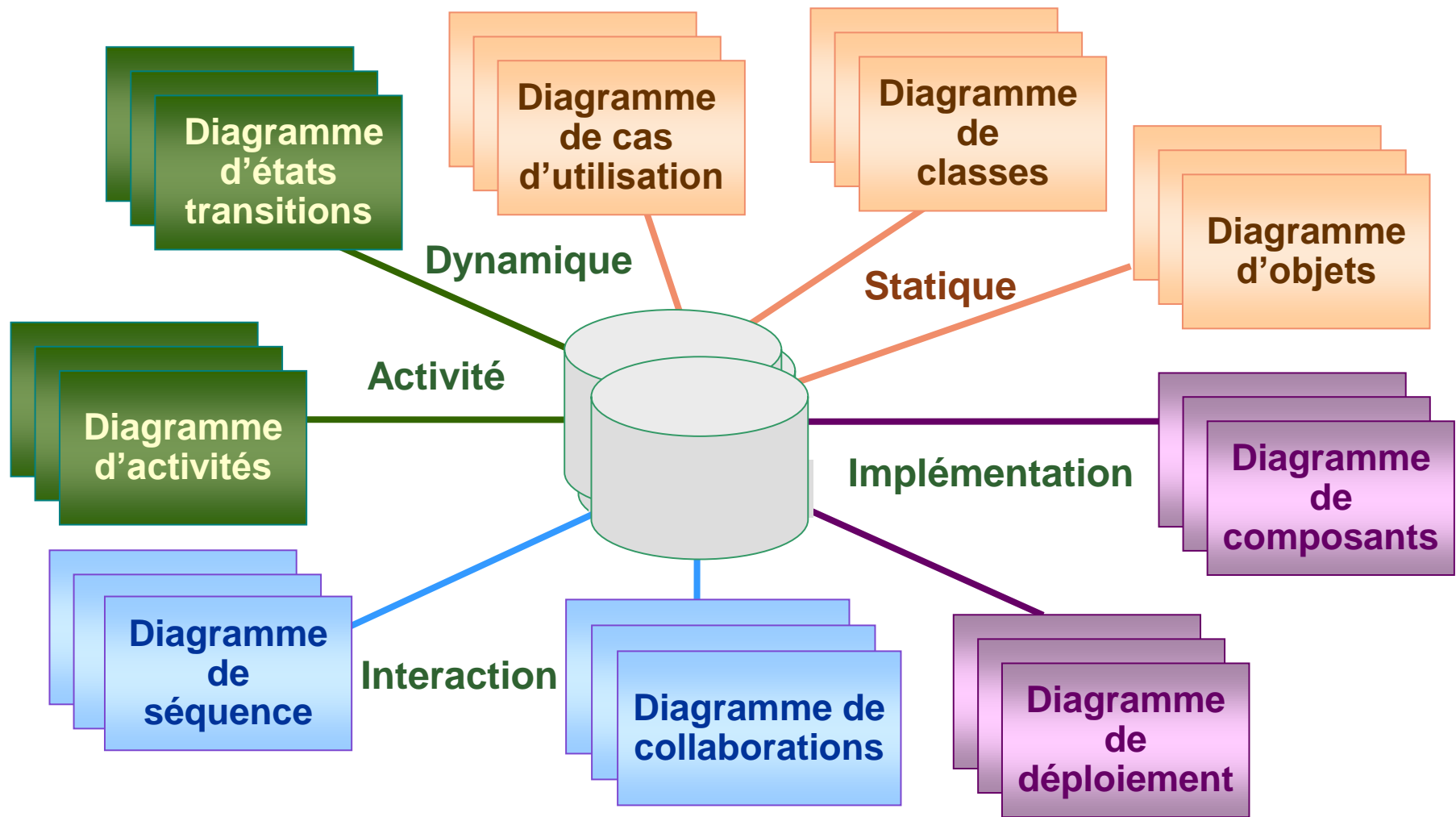
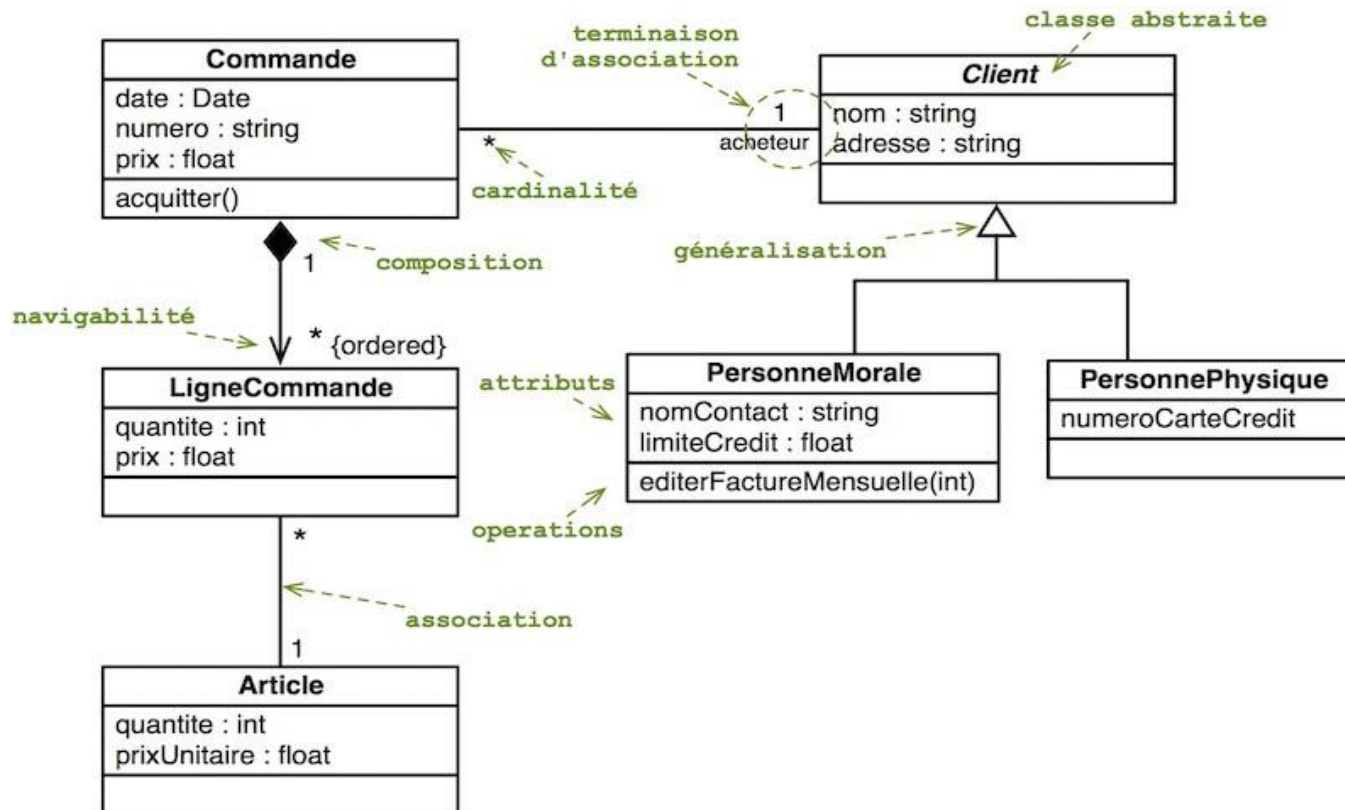




Diagramme de Classes

- Un **diagramme de classes** ou d'objets est un graphe d'éléments connectés par des relations ou des associations qui représente la **vue structurale** d'un SI
- Un diagramme de classes est une vue graphique de la structure **statique**
- Un diagramme d'objet est un cas particulier d'un diagramme de classes (Image dans le temps).





Les classes et les objets (Instances)

- Une classe représente la structure commune d'un ensemble d'objets, c'est un **modèle** pour la création des objets.
- C'est un type **abstrait** caractérisé par des propriétés (attributs et méthodes) communes à un ensemble d'objets.
- Une classe modélise des objets matériels et immatériels d'un SI
- Classe = attributs + méthodes + instances (Objets)
- Une classe est représentée par un **rectangle** qui contient une chaîne de caractères correspondant au nom de la classe.
- Ce rectangle peut être séparé en trois parties (nom, attributs, opérations).
- Le nom de la classe doit commencer par un caractère alphabétique et ne pas contenir le caractère '::'



Classes

Nom DeLaClasse
-nomAttribut1 -nomAttribut2: type -nomAttribut3: type = valeur
+nomOperation1() #nomOperation2(parametre1) -nomOperation3(parametre2: type, parametre3: type) #nomOperation4(): typeRetour -nomOperation5(parametre2: type, parametre3: type): typeRetour2

Employé

Personne
+ <u>nom</u> : string +Prénom : string #id : string nbPerson : integer /Nom complet : string
+constructor() +afficherAttribut() : void



Attributs

- Une classe peut contenir des attributs
- La syntaxe d'un attribut est :

visibilité nom : type

- La visibilité est:

'+' pour public

'#' pour protected

'-' pour private

- UML définit son propre ensemble de types

Integer, real, string, ...

- Un attribut peut être un attribut de classe, il est alors souligné.
- Un attribut peut être dérivé, il est alors préfixé par le caractère `/'



Attributs

Companie
url [3] : string nom : string

Personne
+ <u>nom</u> : string +Prénom : string #id : string nbPerson : integer /Nom complet : string



Opérations

- Une opération est un service qu'une instance de la classe peut exécuter.
- La syntaxe d'une opération est :

visibilité nom(param) : return.

Compagnie

url [3] : string

nom : string

+faireProfit():real

+getTravailEmp(): [*] Employee

Employé

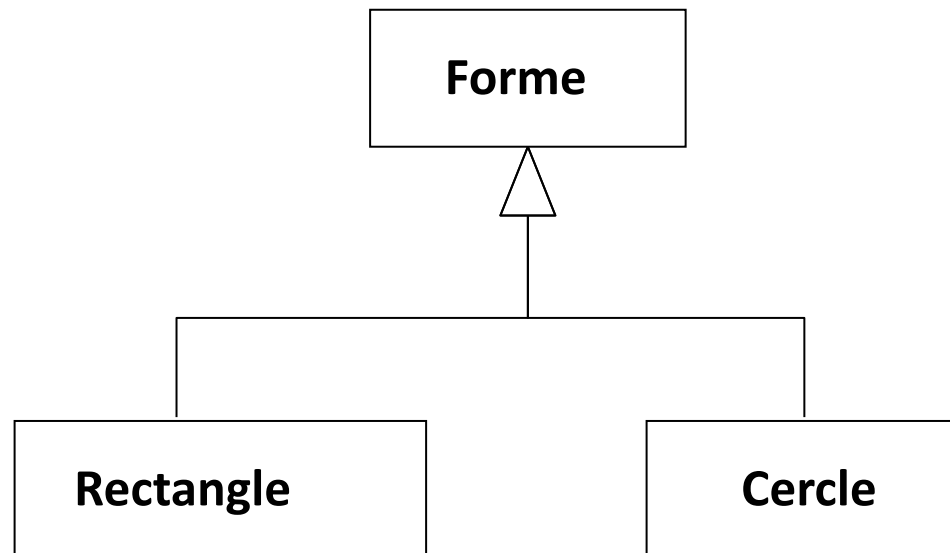
+FinTravail():boolean

+DebTravail(work:string):boolean



Héritage

- L'héritage est une relation entre un élément plus général et un élément plus spécifique.
- L'héritage existe entre des classes, des packages, ...
- L'héritage multiple est possible en UML





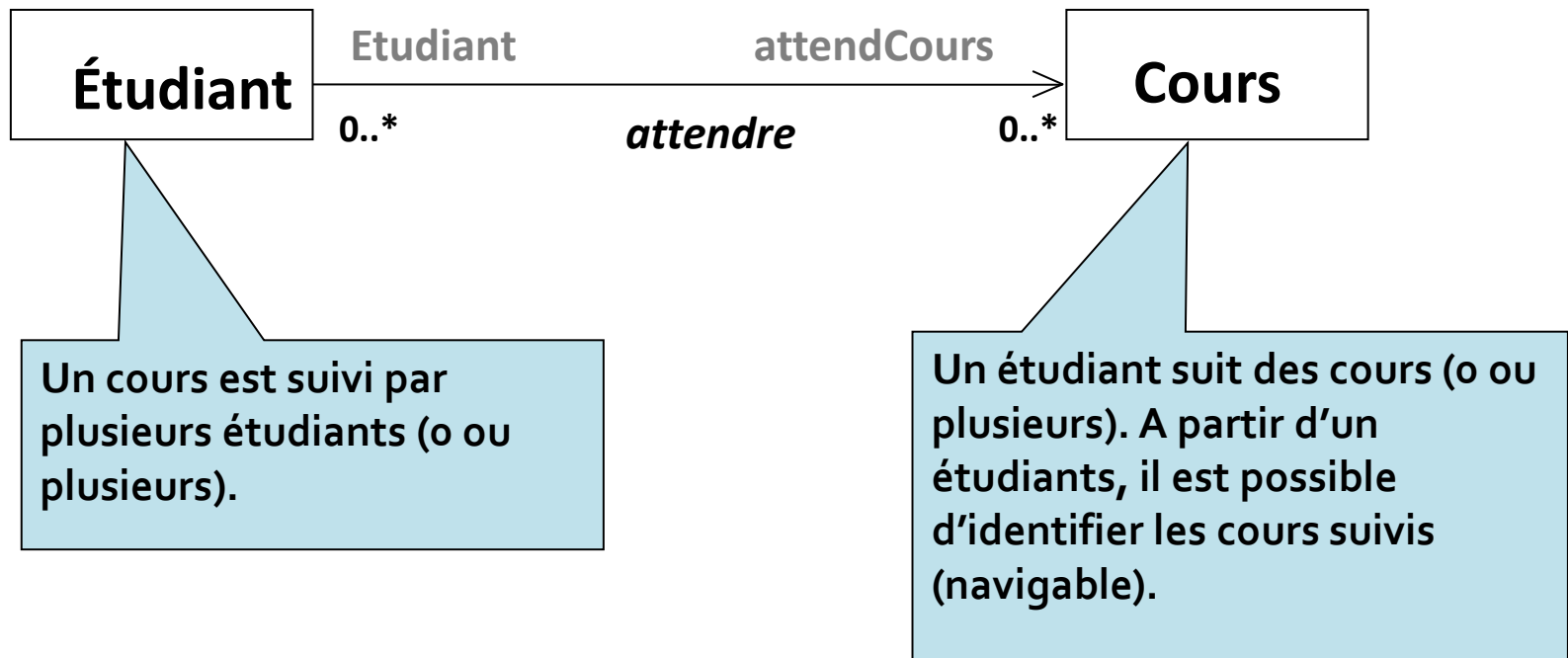
Associations

- Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation.

- *Une association* est paramétrée par :
 - Un nom (Le rôle joué par l'entité connectée)
 - Une multiplicité (0, 1, *, 1..*, *, 1, ...)
 - Un genre d'agrégation (composite, agrégation, none)

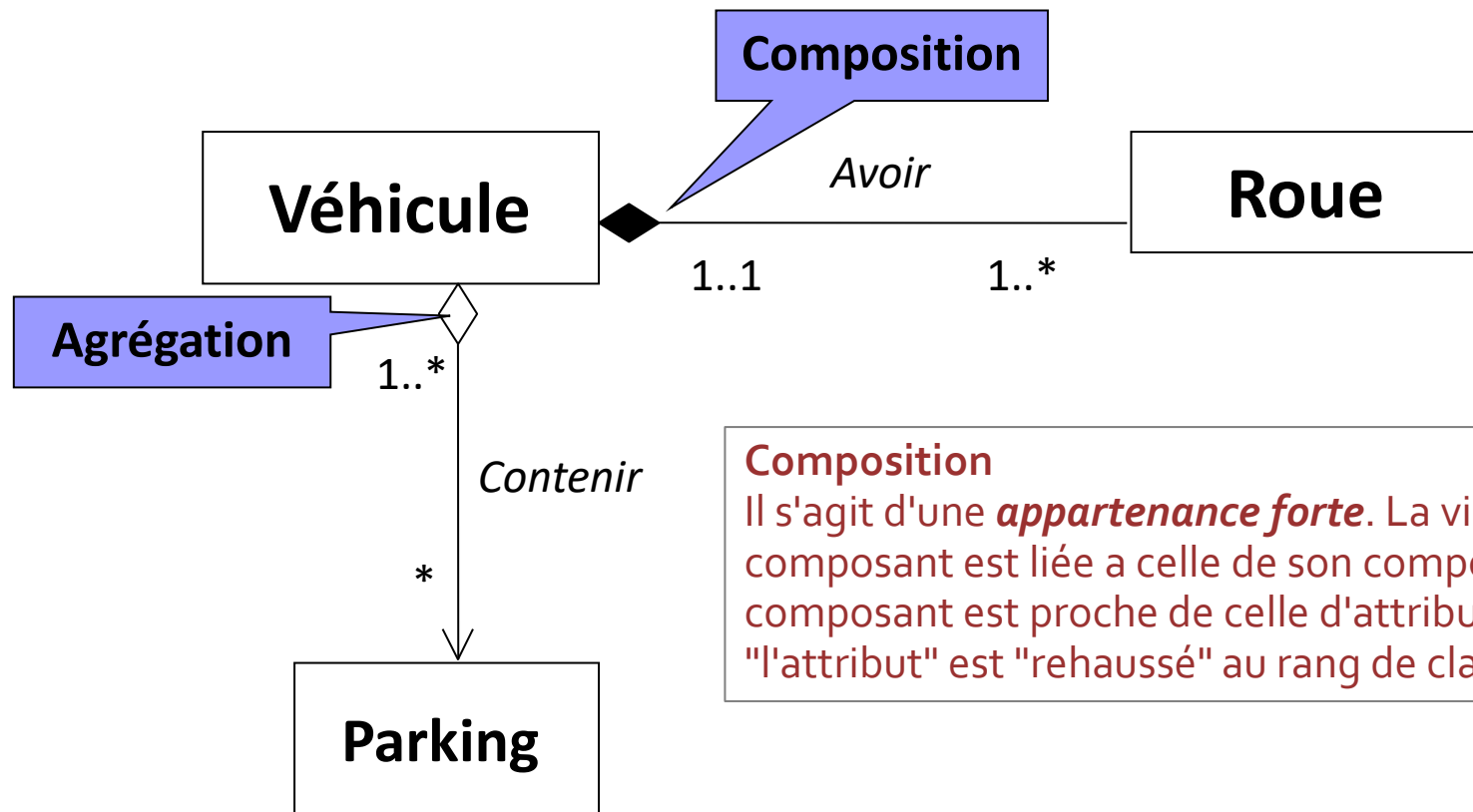


Associations





Agrégation et composition



Composition

Il s'agit d'une **appartenance forte**. La vie de l'objet composant est liée à celle de son composé. La notion de composant est proche de celle d'attribut, si ce n'est que "l'attribut" est "rehaussé" au rang de classe.

Agrégation.

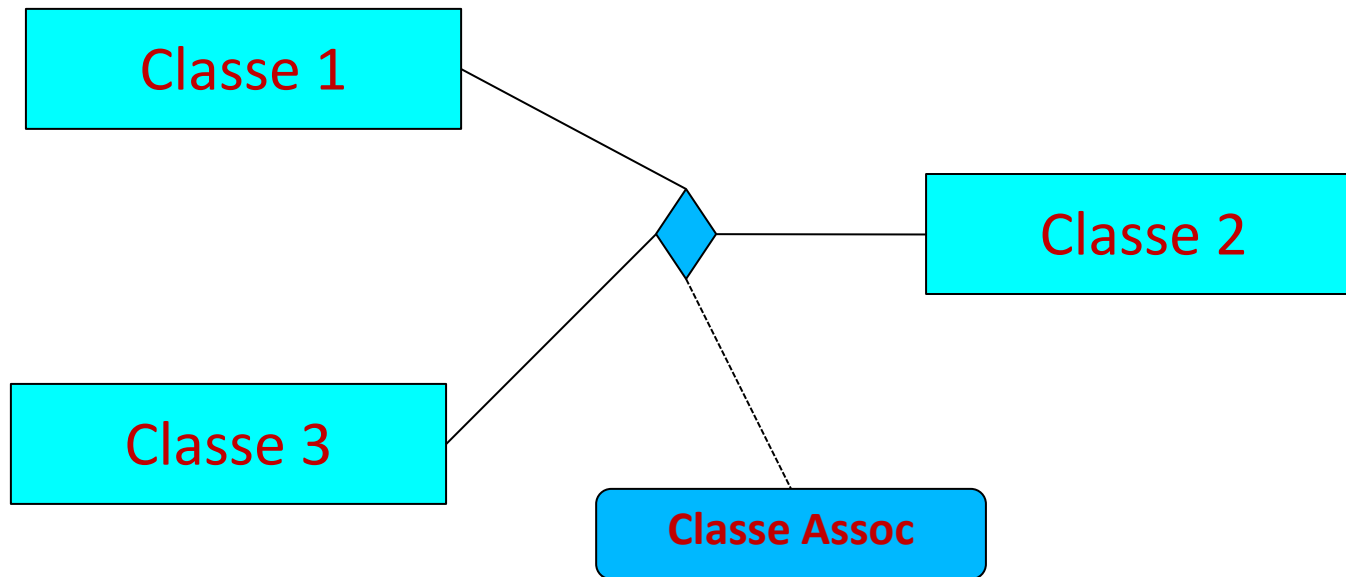
C'est une relation particulière qui attribue à l'une des classes le rôle d'**agrégat** et à l'autre classe le rôle d'**agrégé**.

L'agrégation peut être assimilée à une **appartenance faible**.



Associations

- Les associations ***N-aire*** connectent plusieurs éléments entre eux.
- Les associations ***N-aire*** sont très peu utilisées, elles sont même à éviter.





Classes-Associations

- Une classe-association est une association qui est aussi une classe.
- Les classes-associations sont utilisées lorsque les associations doivent porter des informations.
- Il est toujours possible de se passer des classes-associations.

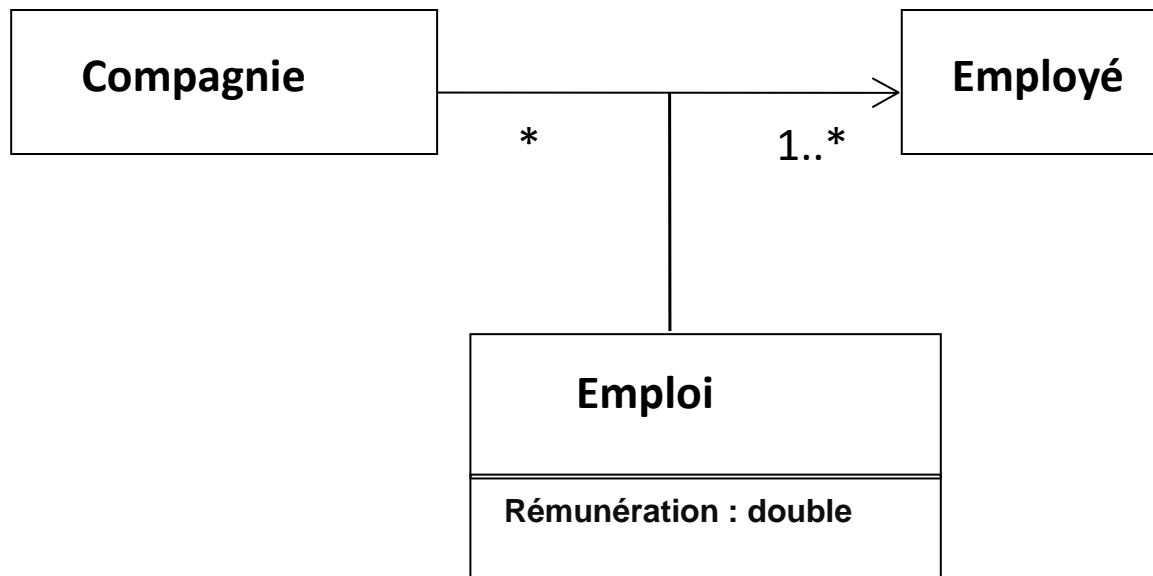




Diagramme de Classe - Fin

- Les diagrammes de classes sont les diagrammes les plus utilisés.
 - Ils permettent de décrire des programmes objet.
 - Ils permettent de décrire le schéma logique de bases de données.
 - Ils permettent de décrire des relations de concepts (modèle métier).
- Les diagrammes de classes peuvent être de différents niveaux d'abstraction.



Application – Elaboration d'un diagramme de classes

Une entreprise de fabrication et de distribution de matériels agricoles dispose d'une usine et de plusieurs dépôts de stockage/expédition.

Un produit est caractérisé par un numéro, un libellé et un prix unitaire. Chaque produit peut être stocké dans un ou plusieurs dépôts.

Un dépôt est caractérisé par son numéro et sa localisation. Dans chaque dépôt, on doit connaître la quantité de chacun des produits.

Un client est déterminé par son numéro, son nom, son adresse, son chiffre d'affaire et un taux de remise.

Chaque client est livré à partir d'un dépôt privilégié, ou à partir d'un dépôt de secours, en cas de problème.

Toute commande client peut concerner un ou plusieurs produits. Elle est caractérisée par un id commande et une date.

Tout mouvement de stock doit enregistrer la nouvelle quantité disponible.

Toutes les données manipulées dans cette application doivent être en mode privé.

Les objets créés pour chacune des classes du DC doivent initialiser tous les attributs de la classe.

On doit prévoir également un moyen pour afficher les données des différentes classes du modèle.



```
NumEmprunteur -> NomEmprunteur
NumEmprunteur -> AdrEmprunteur
NumEmprunteur -> TelEmprunteur
(NumEnprunteur, idMateriel) -> DateEmprunt
(NumEnprunteur, idMateriel) -> DateRetour
idMateriel -> idCategorie
idMateriel -> idMarque
idMateriel -> idCategoriePrix
idCategorie -> LibCategorie
idMateriel -> AnneeSortie
idCategoriePrix -> LibCategoriePrix
idMateriel -> PrixLocation
idMateriel -> idEtat
idMarque -> NomMarque
idCategoriePrix -> PrixLocation
idEtat -> LibEtat # {'EXCELLENT', 'TRES BON', 'BON', 'MOYEN', 'EN REPARATION'}
```