

Génie Logiciel 1

Cours 6 et 8

Diagramme de classes

Diagramme d'objets

Modélisation structurale

- Les classes et les objets modélisent les objets matériels ou immatériels qui existent dans le système qu'on essaie de décrire
- Les relations entre les classes et les objets établissent les connexions entre les divers éléments de modélisation et montrent l'agencement architectural
- Le diagramme de classes et le diagramme d'objets sont les pièces maîtresses de la vue structurale
 - Dans UML, ils sont répertoriés comme des diagrammes montrant la structure "statique"
 - Le diagramme d'objets est un exemple du diagramme de classes
 - Le diagramme d'objets donne une photographie du système dans le temps

Diagramme de classes

- C'est un type de modèle statique
- Il décrit la vue statique du système en terme de classes et de relations entre elles
- Utilisé dans l'analyse, la conception et le développement
- Pour créer un diagramme de classe, il faut :
 - Identifier et décrire les classes
 - Identifier et décrire les relations qui existent entre les classes
- Comment trouver les classes
 - Quelles sont les informations qui doivent être stockées et analysées ?
 - Une information qui doit être stockée, transformée, analysée ou traitée est candidate d'une classe. L'information étant prise dans son sens le plus large (donnée, événement, transaction, etc.)

Diagramme de classes

- Comment trouver les classes
 - A-t-on des systèmes externes ?
 - Un système externe pourrait être vu comme une classe qui n'est pas dans le système étudié mais qui interagit avec lui
 - A-t-on des patrons, librairies de classes, composantes, etc. ?
 - Ces choses contiennent des classes candidates
 - Y a-t-il des supports que le système doit traiter ?
 - Tous les supports techniques connectés au système se transforment en classes candidates qui traitent ces supports
 - A-t-on des parties organisationnelles en contact avec le système ?
 - Ces unités sont représentées par des classes
 - Quels rôles ont les acteurs dans le système ?
 - Ces rôles peuvent être vus comme des classes
 - Exemples : Utilisateur, Opérateur système, client, etc.

Diagramme de classes

- Représentation d'une classe

- **Le nom**

- Écrit en gras et centré
 - Dérivé du domaine du problème
 - Non ambigu

Nom



Voiture

Attributs



Matricule
Marque
Vitesse
Nbrportes

Opérations



Conduire()

- **Les attributs**

- Décrivent les caractéristiques des objets
 - Captent l'information qui décrit et identifie une instance spécifique de la classe
 - Seuls les attributs intéressants du système doivent être modélisés

Diagramme de classes

- Représentation d'une classe
 - **Les attributs**
 - Un attribut est typé (type de base ou type composé) et possède une visibilité
 - La visibilité décrit si l'attribut est visible lorsqu'il est référencé à partir de classes autre que celle où il est défini
 - Exemples de visibilité
 - Privé (-) : L'attribut ne peut être accédé à partir d'autres classes
 - Public (+) : L'attribut peut être vu et utilisé par d'autres classes
 - protégé (#) : L'attribut est utilisé par les sous-classes de la classe

Diagramme de classes

- Représentation d'une classe
 - Les attributs
 - Exemple

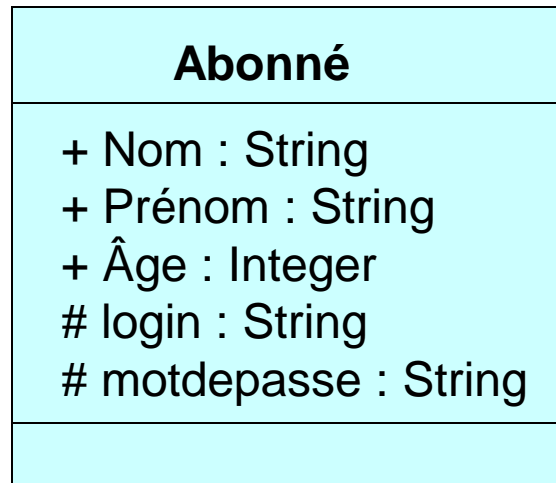
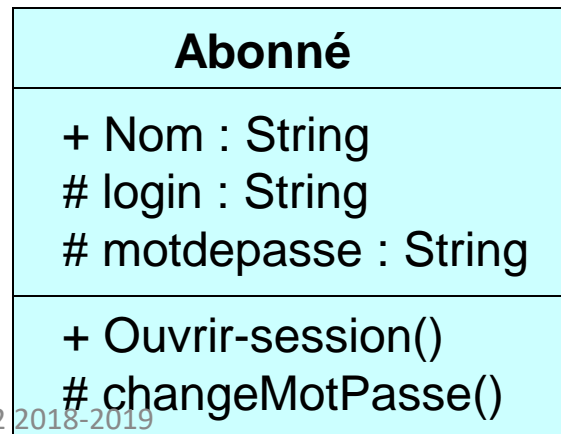


Diagramme de classes

- Représentation d'une classe

- Les opérations

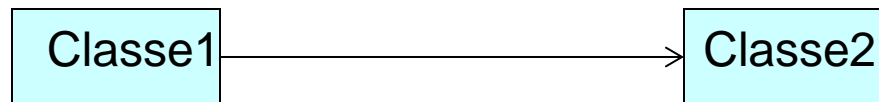
- Utilisées pour manipuler les attributs ou exécuter certaines actions
 - Signature d'une opération = Son nom, son type de retour et ses paramètres
 - Les opérations décrivent qu'est ce qu'une classe peut faire et quels services elle offre aux autres classes
 - Les mêmes visibilité s'appliquent aux attributs et aux opérations (public, protégé, privé)
 - Exemple



Relations entre les classes

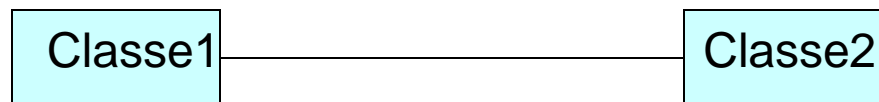
- L'association

- C'est une connexion entre les classes (elle est aussi une connexion entre les objets de ces classes)
- Association orientée



Les objets de Classe1 sont accessibles aux objets de Classe2

- Association non orientée (Bi-directionnelle)



Les objets de Classe2 sont accessibles à partir de ceux de Classe1 et vice-versa

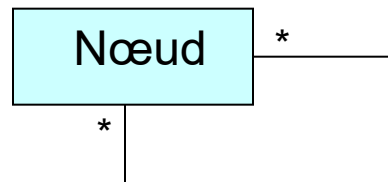
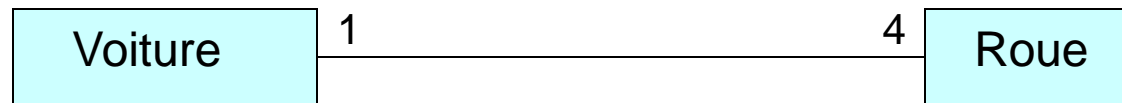
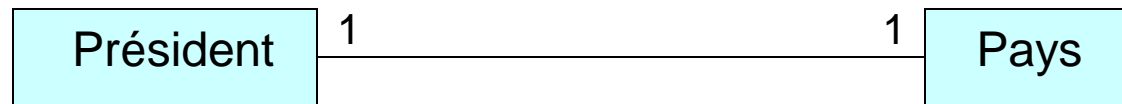
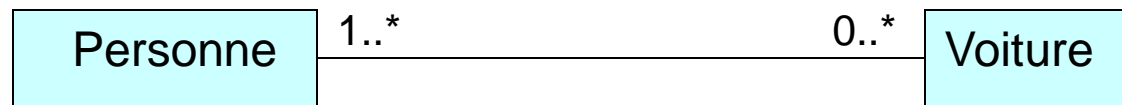
Relations entre les classes

- L'association
 - Multiplicités

1	Un et un seul
0 .. 1	Zéro ou un
n	n (entier naturel)
m .. n	De m à n (entiers naturels)
*	De zéro à plusieurs
0 .. *	De zéro à plusieurs
1 .. *	De un à plusieurs

Relations entre les classes

- L'association
 - Multiplicité
 - Exemples



Un réseau est composé de nœuds inter-connectés

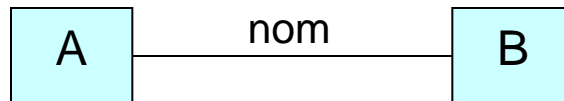
Association réflexive

Relations entre les classes

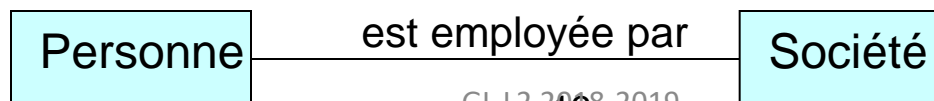
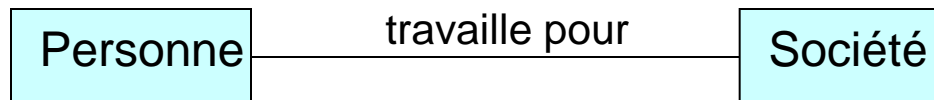
- L'association

- Nommage des associations

- Une association peut être nommée pour faciliter la compréhension des modèles. Dans ce cas le nom est indiqué au milieu du lien symbolisant l'association

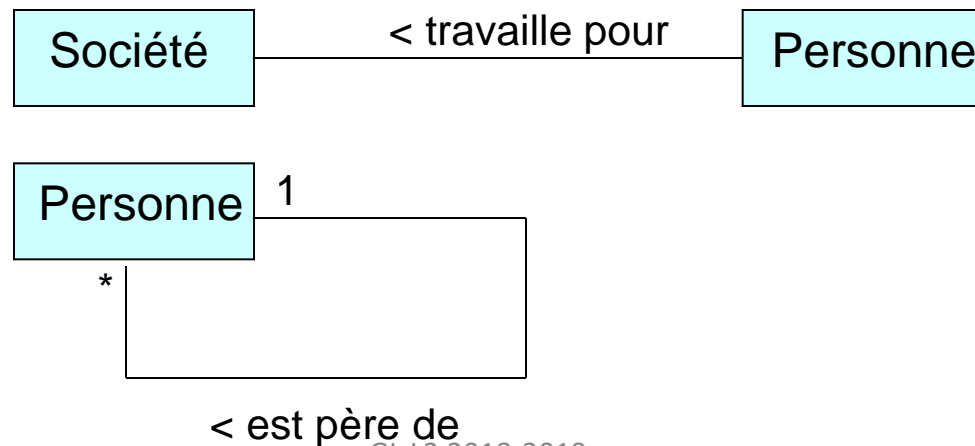


- L'usage recommande de choisir comme nom d'une association une forme verbale active (exemple : travaille pour) ou passive (exemple : est employé par)



Relations entre les classes

- L'association
 - Nommage des associations
 - Par défaut le sens du lecture du nom d'une association est de gauche à droite
 - Dans le cas où la lecture du nom est ambiguë, on peut ajouter l'un des signes < ou > pour indiquer le sens de lecture
 - Exemples

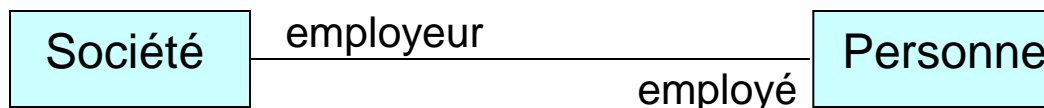
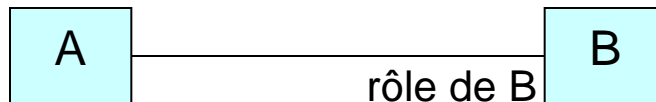


Relations entre les classes

- L'association

- Rôles des extrémités d'une association

- On peut attribuer à une extrémité d'une association un nom appelé **rôle** qui décrit comment une classe source voit une classe destination au travers de l'association
 - Le rôle est placé près de la fin de l'association et à côté de la classe à laquelle il est appliqué
 - L'utilisation des rôles est optionnelle
 - Représentation et exemple

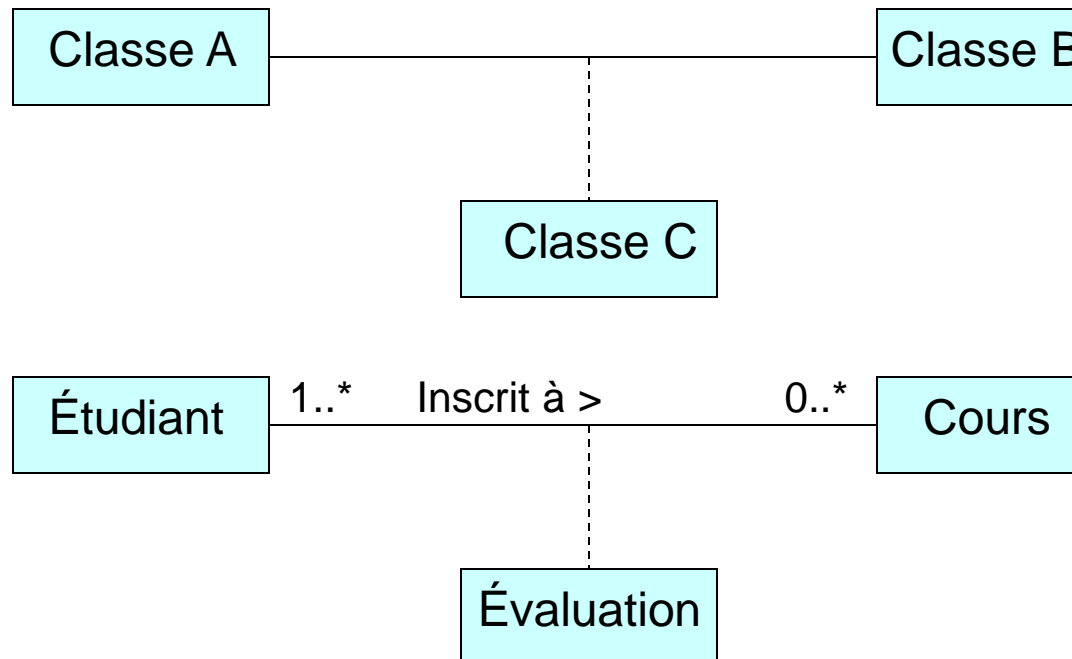


Relations entre les classes

- L'association
 - Classe d'association
 - Il est possible de représenter une association par une classe pour ajouter par exemple des attributs ou des opérations dans l'association
 - La classe attachée à l'association est appelée une **classe d'association** ou **classe associative**
 - La classe d'association possède à la fois les caractéristiques d'une association et celle d'une classe et peut à ce titre participer à d'autres relations dans le modèle
 - La classe d'association est attachée à l'association avec une ligne en pointillée

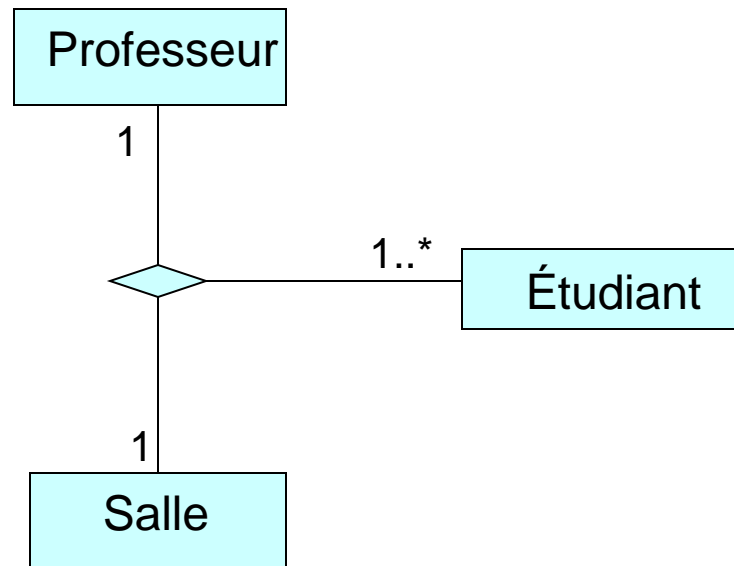
Relations entre les classes

- L'association
 - Classe d'association
 - Représentation et exemple



Relations entre les classes

- L'association
 - Association ternaire
 - Trois classes sont associées ensemble
 - Configuration déconseillée
 - Exemple



Relations entre les classes

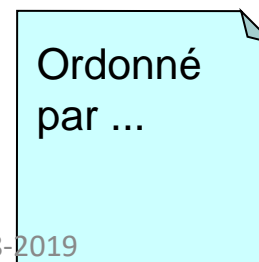
- **Contraintes sur les associations**

- Association ordonnée

- C'est une contrainte qui spécifie que les objets sont ordonnés (selon la clé, le nom, la date, etc.)
- Cette contrainte est spécifiée par le stéréotype **{Ordonné}** du côté de la classe dont les instances sont ordonnées



- Le modèle ne spécifie pas comment les objets sont ordonnés
- Pour décrire comment les objets sont ordonnés on utilise la notation graphique suivante :

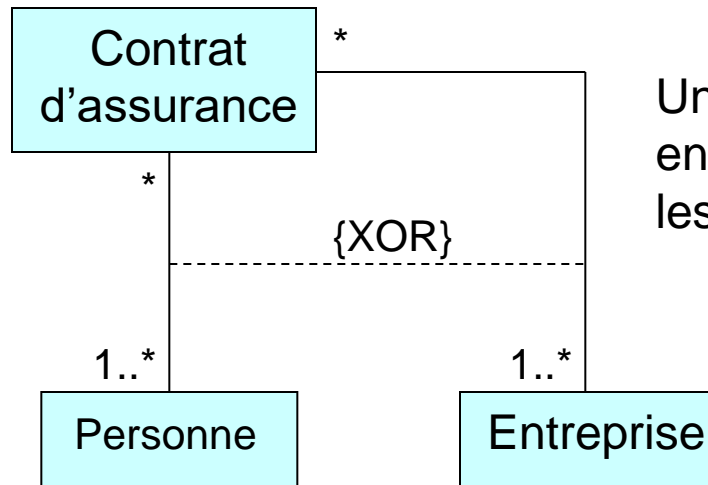


Relations entre les classes

- **Contraintes sur les associations**

- Association « ou-exclusif »

- Elle spécifie que les objets d'une classe peuvent participer au meilleurs des cas dans une association pour un certain temps
 - Exemple



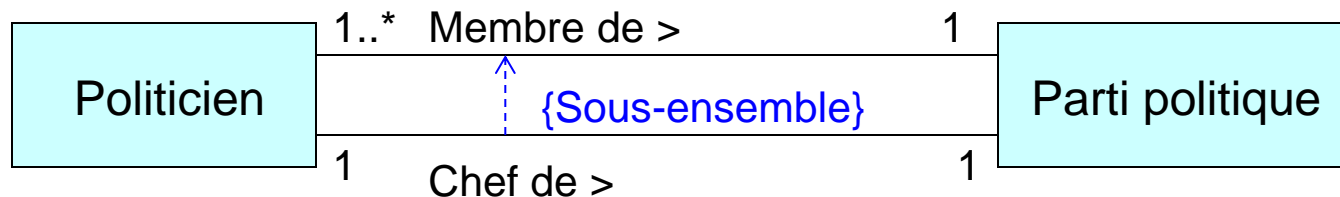
Un contrat d'assurance concerne une entreprise ou une personne mais pas les deux en même temps

Relations entre les classes

- **Contraintes sur les associations**

- Association « sous-ensemble »

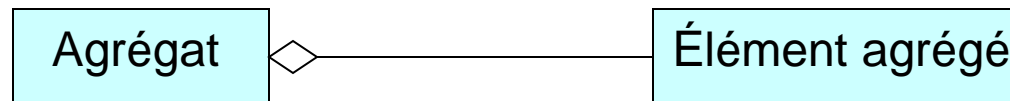
- C'est une contrainte qui indique qu'une collection est incluse dans une autre collection
 - La contrainte est placée à proximité d'une relation de dépendance entre deux associations
 - La flèche de la relation de dépendance indique le sens de la contrainte
 - Exemple



Relations entre les classes

- L'agrégation

- Une agrégation représente une association non symétrique dans laquelle une des extrémités joue un rôle prédominant par rapport à l'autre extrémité
- L'agrégation représente une **relation de type ensemble/élément**
- L'agrégation ne concerne qu'un seul rôle d'une association
- Représentation



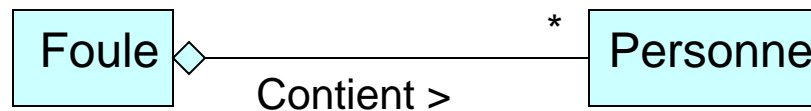
- L'agrégation permet de modéliser une contrainte d'intégrité et de désigner l'agrégat comme gérant de cette contrainte

Relations entre les classes

- L'agrégation

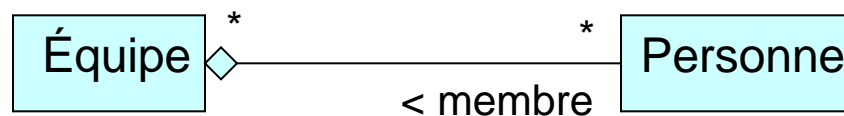
- Exemple 1

- Une personne est dans une foule
 - Une foule contient plusieurs personnes



- Exemple 2 (Agrégation partagée)

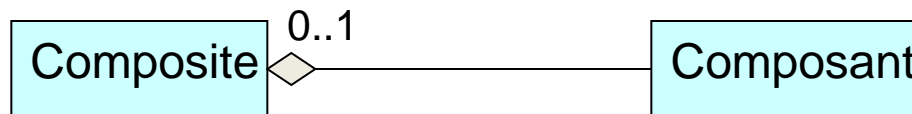
- Une personne fait partie de plusieurs équipes
 - Une équipe contient plusieurs personnes



Relations entre les classes

- La composition

- La composition est un cas particulier de l'agrégation avec un couplage plus important
- La classe qui possède le rôle prédominant dans une composition est appelée classe composite ou classe conteneur
- La composition implique une contrainte sur la valeur de la multiplicité du côté du composite qui doit être 0 ou 1. Cela veut dire qu'il n'y a pas de partage des composants
- Représentation

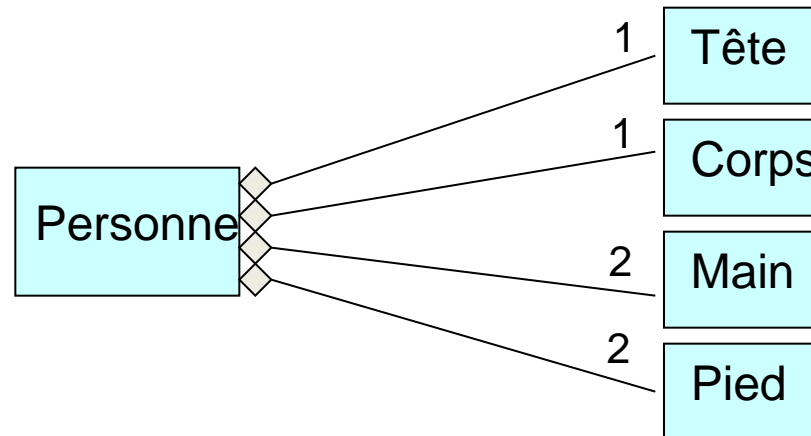


Relations entre les classes

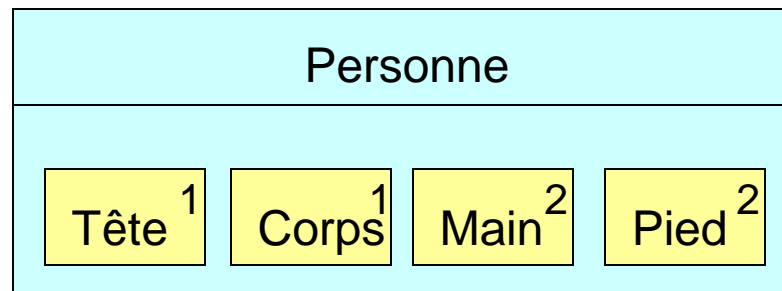
- La composition

- Exemple

Possibilité 1

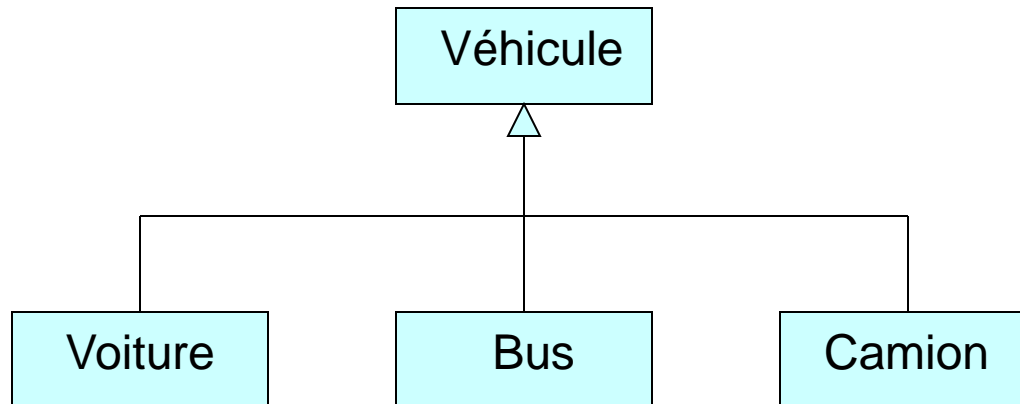


Possibilité 2



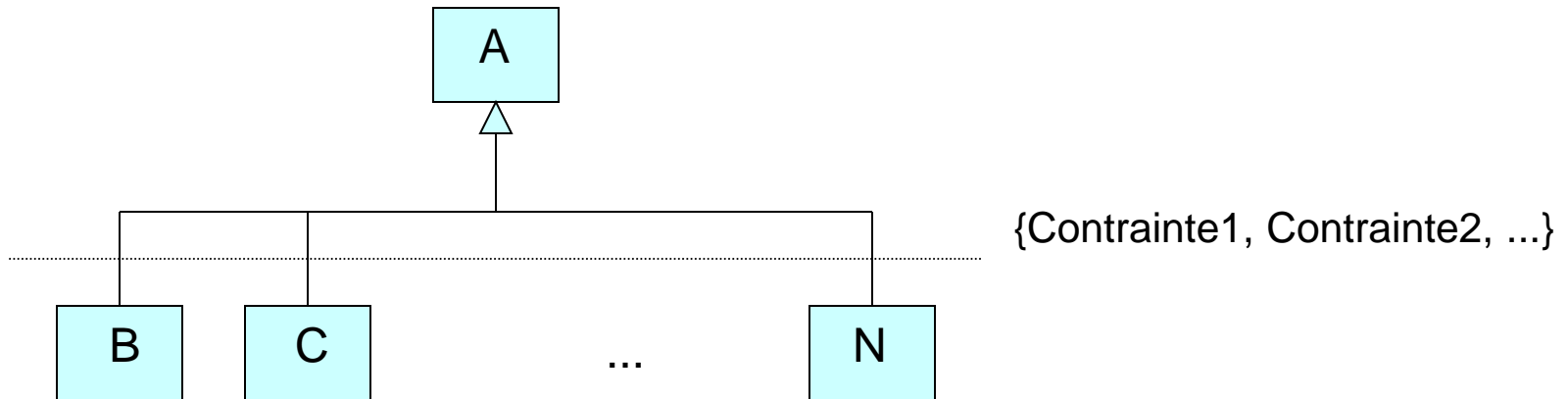
Relations entre les classes

- La généralisation
 - C'est une relation de classification entre un élément général et un élément plus spécifique
 - L'élément le plus spécifique est cohérent avec l'élément le plus général et contient plus d'informations
 - Exemple



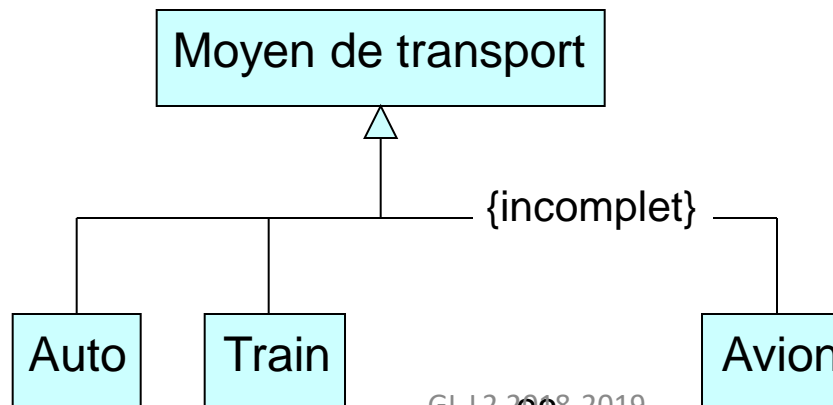
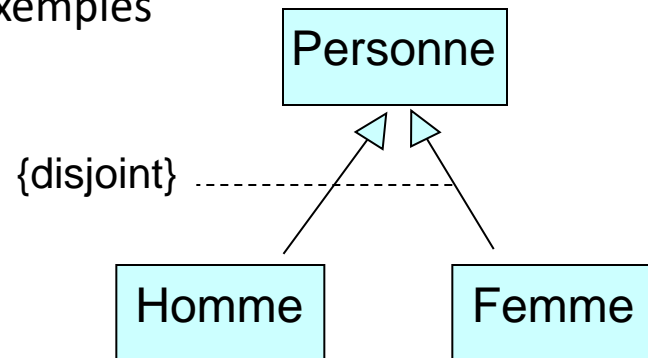
Relations entre les classes

- La généralisation
 - Généralisation avec restriction



Relations entre les classes

- La généralisation
 - Généralisation avec restriction
 - Exemples



Incomplet veut dire qu'il peut exister d'autres moyens de locomotion non énumérés

Relations entre les classes

- Raffinement
 - C'est une relation entre 2 descriptions d'une même chose mais à des différents niveau d'abstraction
 - Exemple



Relations entre les classes

■ Dérivation

- C'est une règle qui indique comment certaines choses peuvent être dérivées
- L'attribut dérivé est indiqué en faisant précéder son nom d'un «/»
- Le principe de dérivation est aussi appliqué pour des associations
- Exemple

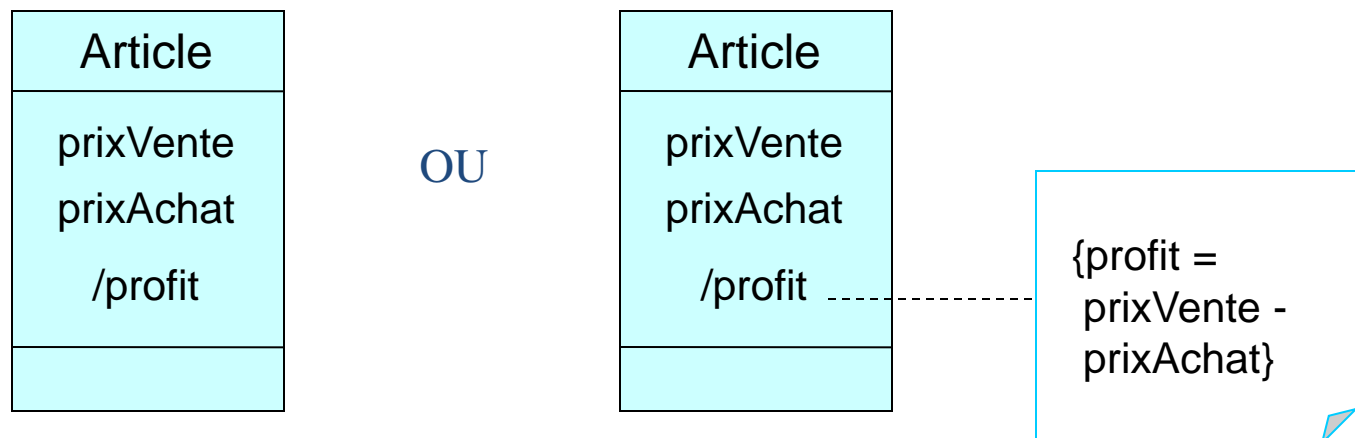


Diagramme de classes

- Classe abstraite

- C'est une classe qui ne peut pas être instanciée
- Elle factorise un comportement commun des sous-classes
- Une classe est abstraite si elle dispose d'un attribut ou d'une méthode abstraite

– Exemple

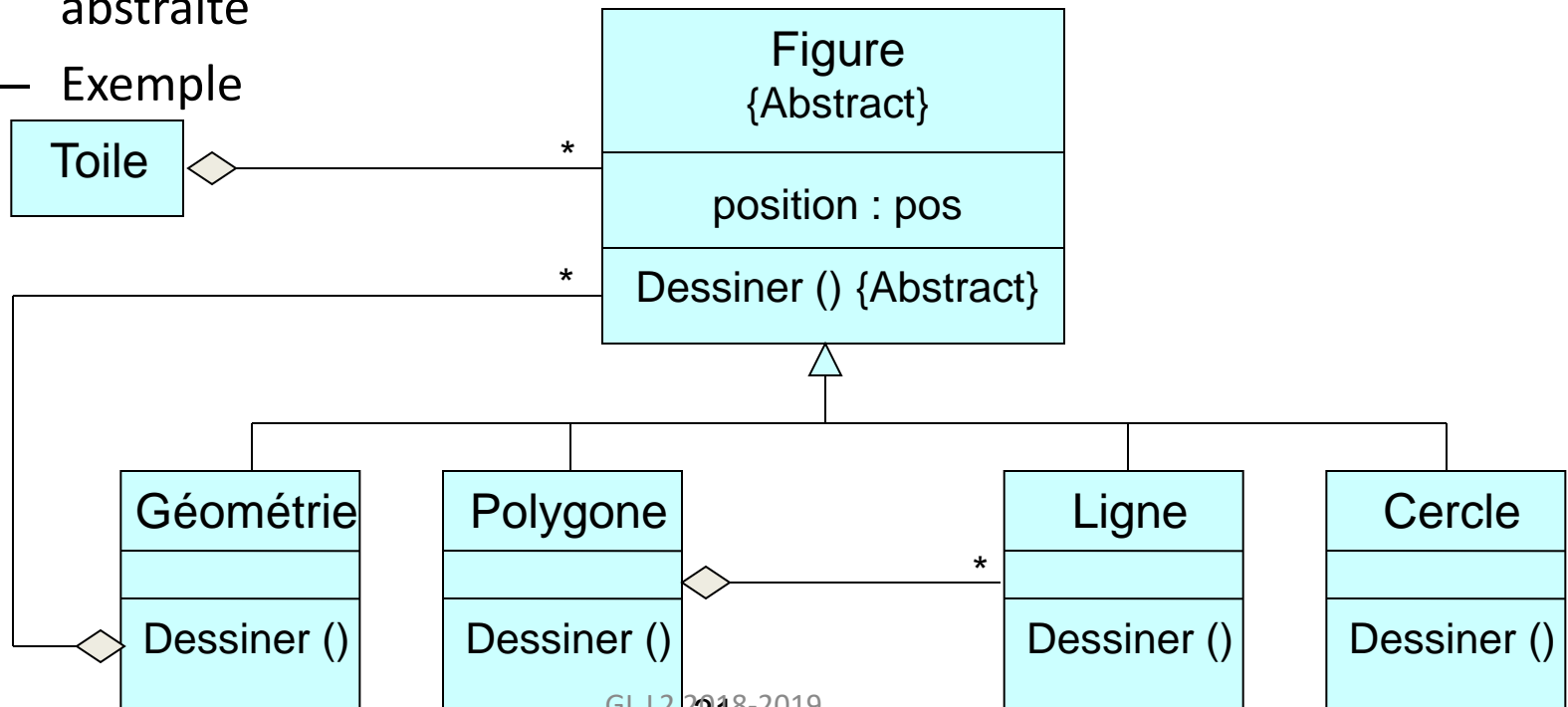


Diagramme de classes

- Exemple de diagramme de classes

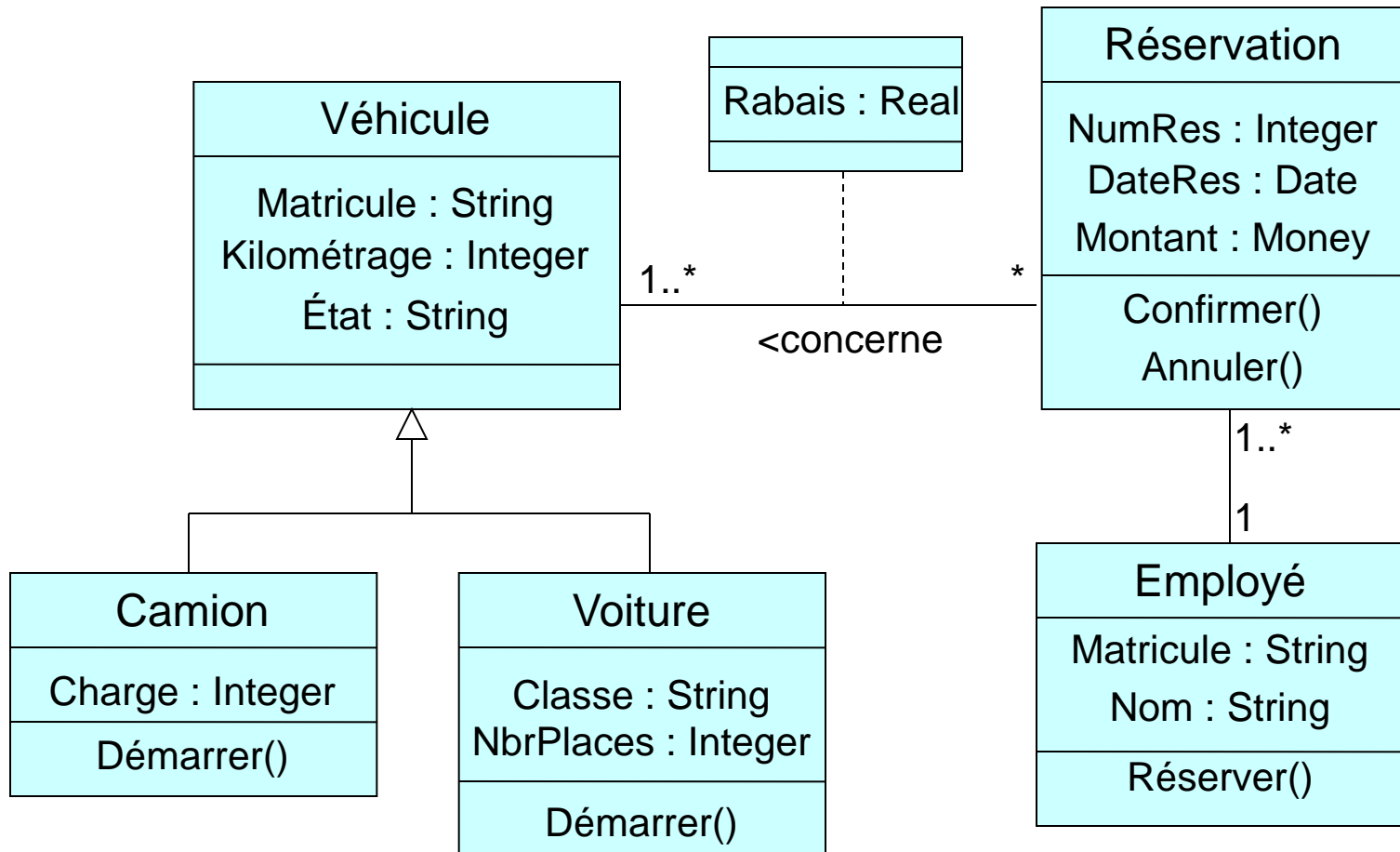


Diagramme d'objets

- Diagramme d'objets = Diagramme d'instances
- C'est un exemple de diagramme de classes avec des objets instanciés
- Montre comment le système est vu à un instant t donné dans le temps
- Notation : dérivée de celle de la classe
- Les diagrammes d'objets sont utilisés pour montrer un contexte (avant ou après une interaction entre objets par exemple)
- Plusieurs notations possibles

Nom de l'objet

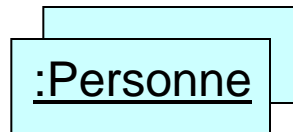
Nom de l'objet : Classe

Objet anonyme

: Classe

Diagramme d'objets

- Représentation d'un groupe d'objets instances d'une même classe
 - Un message vers le groupe d'objets atteint l'ensemble des objets du groupe



- Le nom de la classe peut contenir le chemin complet, composé à partir des noms des différents paquetages séparés par des doubles deux-points

BoutonOK : IHM::Contrôles::BoutonPoussoir

Diagramme d'objets

- Représentation des liens

- Les objets sont reliés par des liens, instances des associations entre les classes qui donnent naissance aux objets considérés
- Les liens entre objets représentent les connexions entre les instances des classes à un instant donné seulement (dans un contexte déterminé). Autrement dit, un diagramme de classe représente une situation générale alors qu'un diagramme objets représente une situation particulière
- Exemple

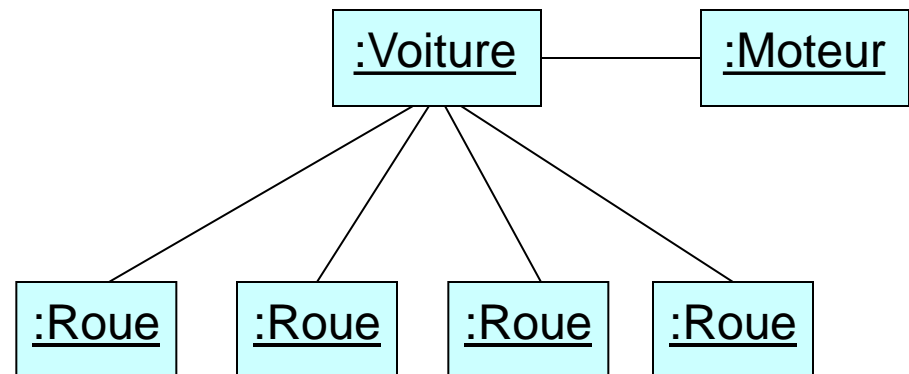
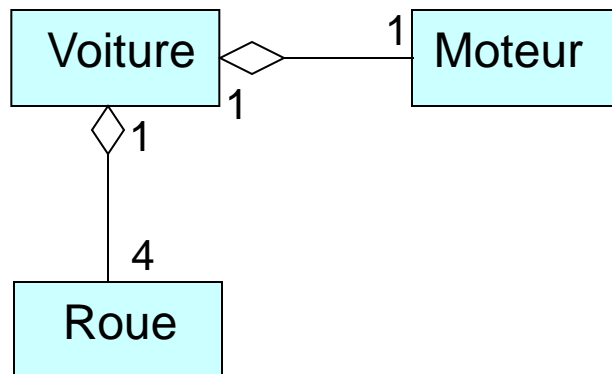
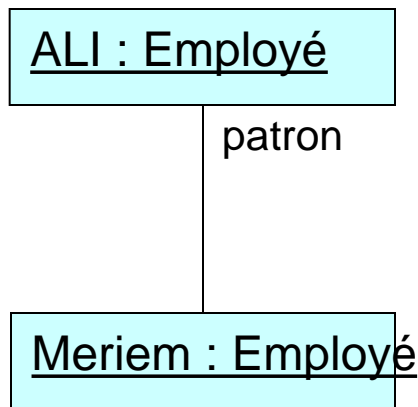


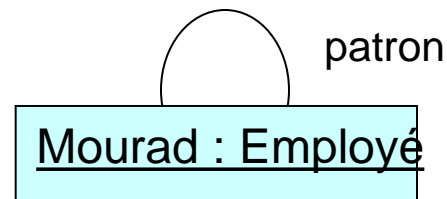
Diagramme d'objets

- Représentation des liens

- Un lien instance d'une instance réflexive peut relier un objet à lui même. Dans ce cas le lien est représenté par une boucle portée par l'objet en question
- Exemple



ALi est le patron
de Meriem



Mourad est le patron
de lui-même

Diagramme d'objets

- Les objets composites
 - Il est possible de représenter les objets composés de sous-objets au moyen d'un objet composite afin de réduire la complexité des diagrammes
 - L'objet composite se présente comme un objet habituel avec la différence que les attributs sont remplacés par des objets soit sous une forme textuelle soulignée soit sous une forme graphique
 - Représentation

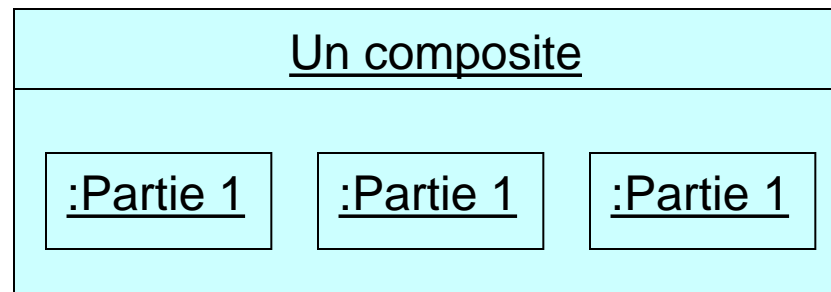


Diagramme d'objets

- Les objets composites
 - Exemple : Fenêtre à l'écran

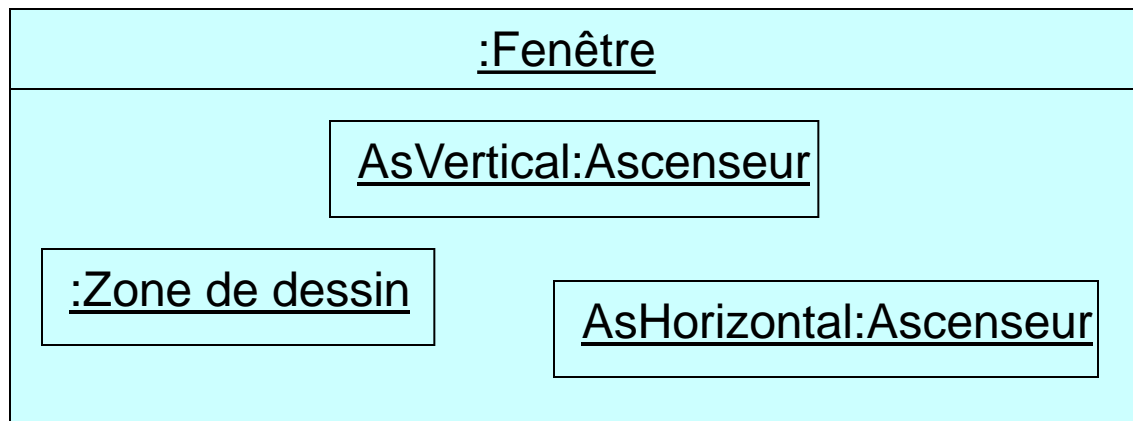
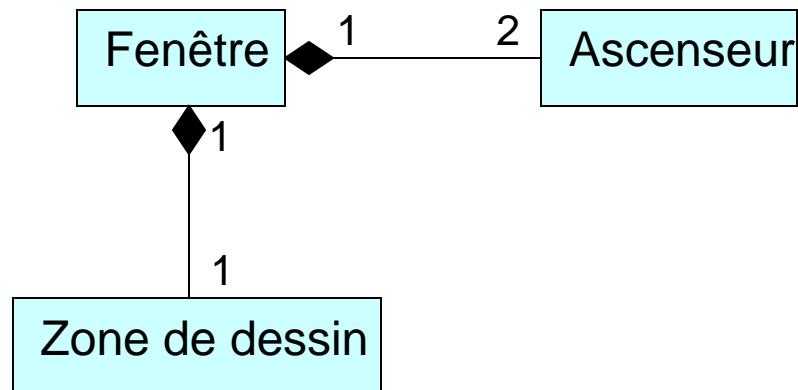


Diagramme d'objets

- Exemple de diagramme objets

Diagramme de classes

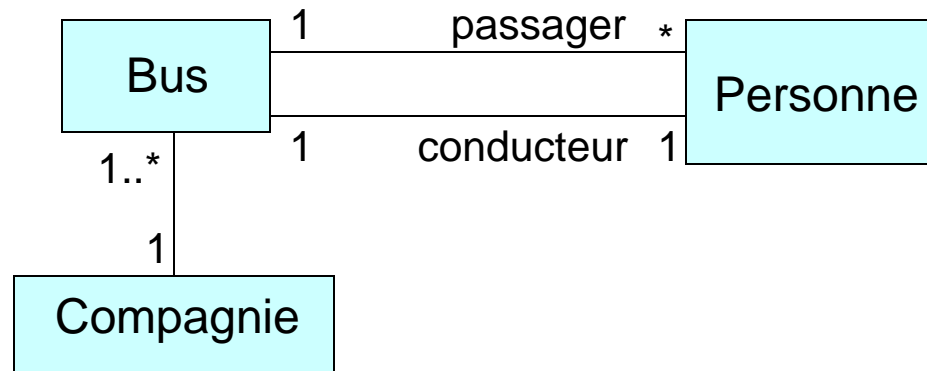
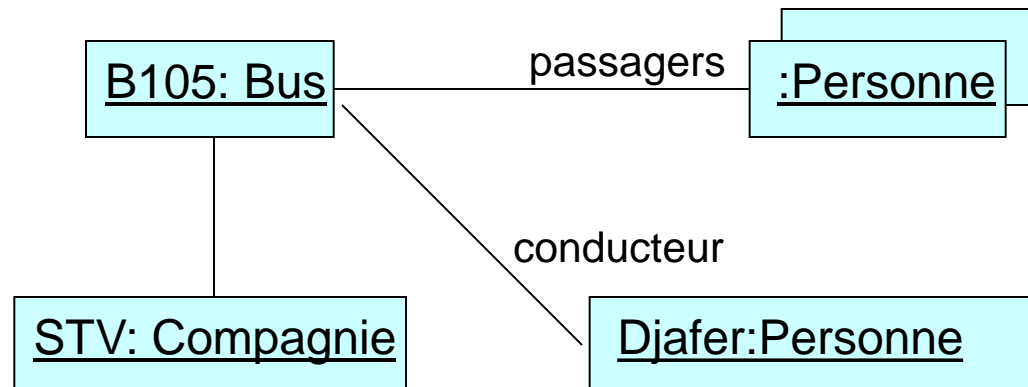


Diagramme d'objets



Références

- P-A. Muller, N. Gaertner, «Modélisation Objet avec UML»
- B. Moulin, «Object Oriented Project (Design and Management)»
- N. Tawbi et H. Yahyaoui, «Génie Logiciel IFT-16859 Partie 2, Avril 99»
- <http://uml.free.fr>