

Analyse et modélisation des processus métiers / Conception UML

Nassiri Naoual

GI S3

2023-2024



Chapitre 1

Introduction UML

Systeme d'information

Un **système d'Information** représente l'ensemble des éléments participant :

- à la collecte
- à la gestion,
- au traitement,
- au transport et,
- à la diffusion,

de **l'information** au sein de l'organisation.

Donnée et Information

- Les données sont des faits bruts sur l'organisation et ses transactions commerciales. La plupart des données ont peu de signification et d'utilités eux-mêmes.
- Une information est un ensemble de données organisées

Problématique



Comment le client
a exprimé son besoin



Comment le chef de
projet l'a compris



Comment l'ingénieur
l'a conçu



Comment le
programmeur l'a écrit



Comment le responsable
des ventes l'a décrit



Comment le projet
a été documenté



Ce qui a finalement
été installé



Comment le client
a été facturé



Comment la hotline
répond aux demandes



Ce dont le client avait
réellement besoin

Objectif de la modélisation

- Les systèmes deviennent de plus en plus complexes et dépassent la compréhension et la maîtrise par un seul individu. Le recours à un modèle conceptuel s'avère indispensable
- Un modèle est un langage commun entre les différents profils (métier, technique, etc.)
- Un modèle est une représentation abstraite simplifiée de la réalité.
- La modélisation Permet de :
 - Réduire la complexité en éliminant les détails.
 - Décrire le fonctionnement global du système
 - Permet de répartir les tâches et les automatiser (génération de code)
- Il est utilisé et progressivement enrichi dans toutes les étapes d'un projet : spécification, analyse, conception, test, intégration et rétro-ingénierie

Méthode de conception et d'analyse

Il existe de nombreuses méthodes :

- Méthodes fonctionnelles de décomposition hiérarchique, la modélisation est réalisée à partir de fonctions que doit réaliser le système ;
- Méthodes objets comme **UML**, on identifie les objets manipulés par le système, avec leurs états et leurs comportements.

Approche Objet

- La Conception Orientée Objet (COO) est la méthode qui conduit à des architectures logicielles fondées sur les objets du système, plutôt que sur une décomposition fonctionnelle.
- C'est la structure du système qui lui donne sa forme.
- On peut partir des objets du domaine (briques de base) et remonter vers le système global : approche ascendante.
 - Attention, l'approche objet n'est pas seulement ascendante

Objet

- Un objet représente une entité du monde réel (ou du monde virtuel pour les objets immatériels) qui se caractérise par une identité, des états significatifs et un comportement.

$$\textbf{Objet} = \textbf{Etat} + \textbf{Comportement} + \textbf{Identité}$$

- Un objet sans état ou sans comportement peut exister marginalement, mais dans tous les cas, un objet possède une identité.
- **Exemple** : une personne, une voiture, une maison, ...

Objet (État)

- L'état regroupe les **valeurs** instantanées de tous **les attributs** d'un objet sachant qu'un attribut est une information qui qualifie l'objet qui le contient.
- Chaque attribut peut prendre une valeur dans un domaine de définition donné.
- L'état d'un objet, à un instant donné, correspond à une sélection de valeurs, parmi toutes les valeurs possibles des différents attributs.

Objet (Comportement)

- Le comportement regroupe toutes les compétences d'un objet et décrit les actions et les réactions de cet objet.
- Les opérations (méthodes).

Objet (Identité)

- En plus de son état, un objet possède une identité qui caractérise son existence propre. L'identité permet de distinguer tout objet de façon non ambiguë, et cela indépendamment de son état.
- Cela permet, entre autres, de distinguer deux objets dont toutes les valeurs d'attributs sont identiques.
- L'identité est un concept, elle ne se représente pas de manière spécifique en modélisation. Chaque objet possède une identité de manière implicite

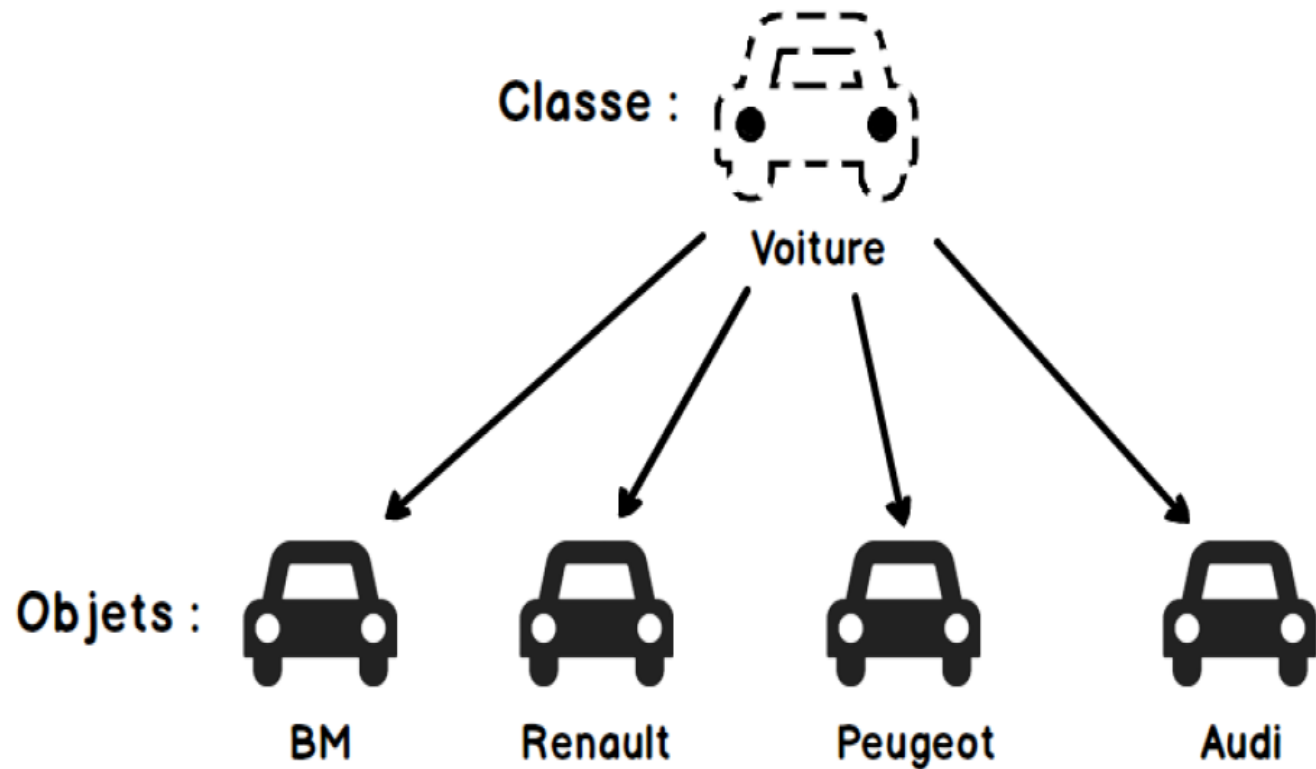
Classe (Définition)

- Une classe est l'abstraction d'un ensemble d'objets qui possèdent une structure identique (liste des attributs) et un même comportement (liste des opérations).

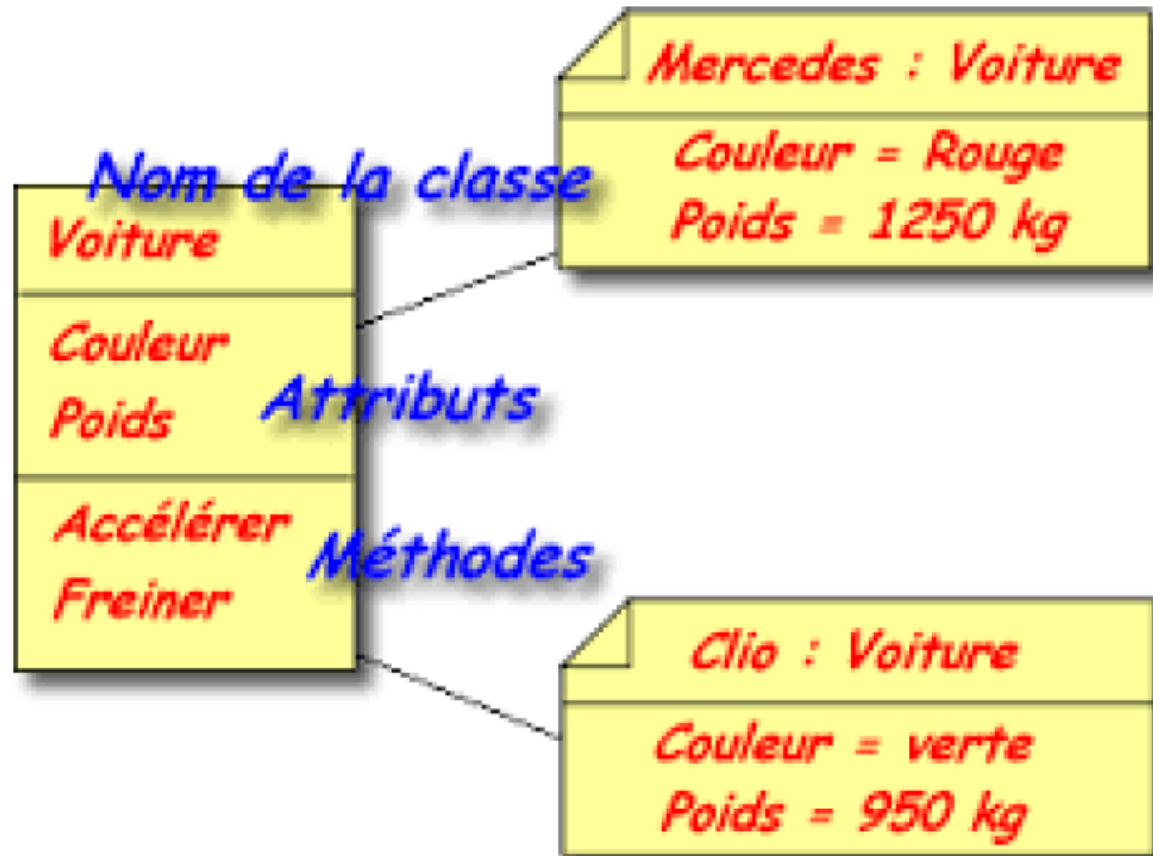
| |
|---------------|
| Nom de classe |
| Attributs |
| Opérations() |

- **Exemple** : employé , entreprise, personne...
- Un objet est une instance d'une et une seule classe.

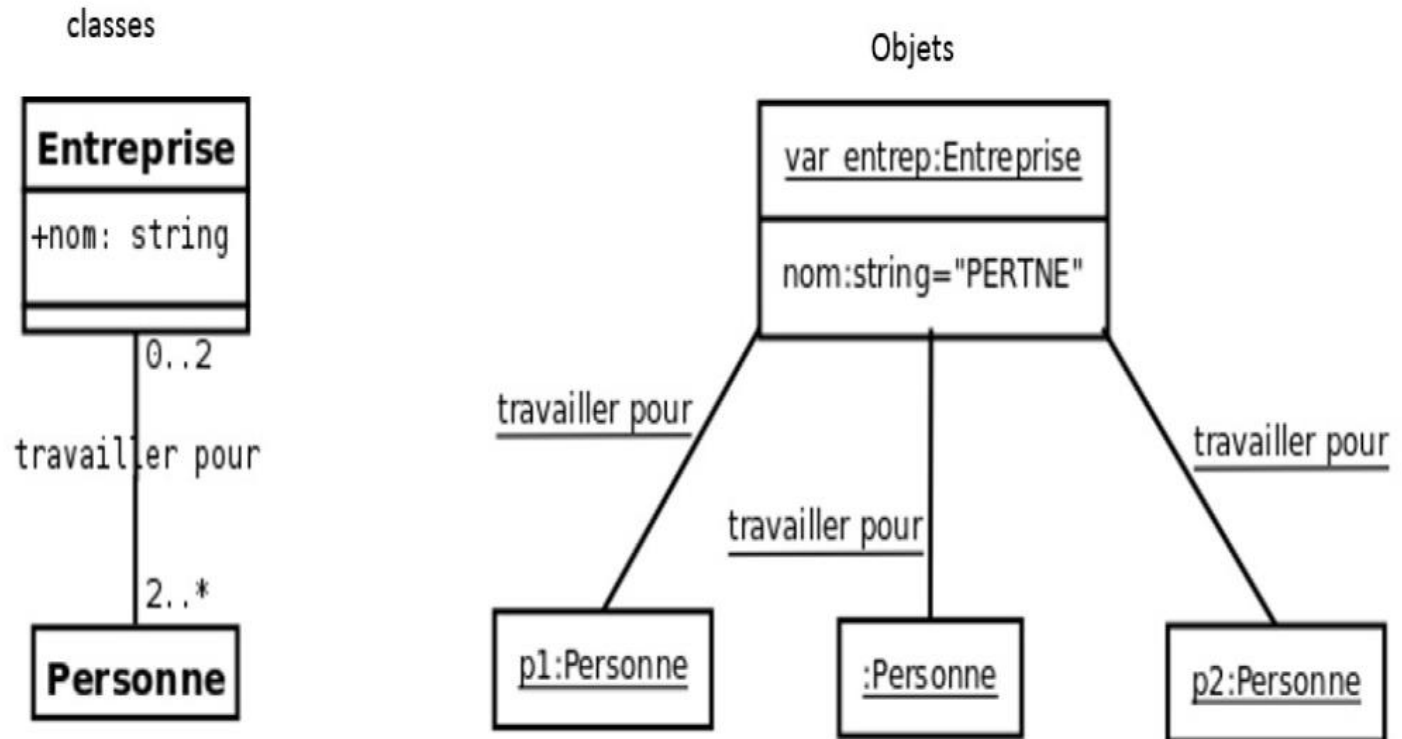
Classe (Exemple)



Classe (Exemple)



Classe (Exemple)



Analyse et conception orientées objet

- **Analyse orientée objet** : Méthode d'analyse qui examine les besoins en termes de classes et d'objets trouvés dans l'espace du problème.
- **Conception orientée objet** : Méthode de conception qui mène à une décomposition orientée objet et utilise une notation pour représenter les différents aspects du système en cours de conception

Analyse et conception orientées objet

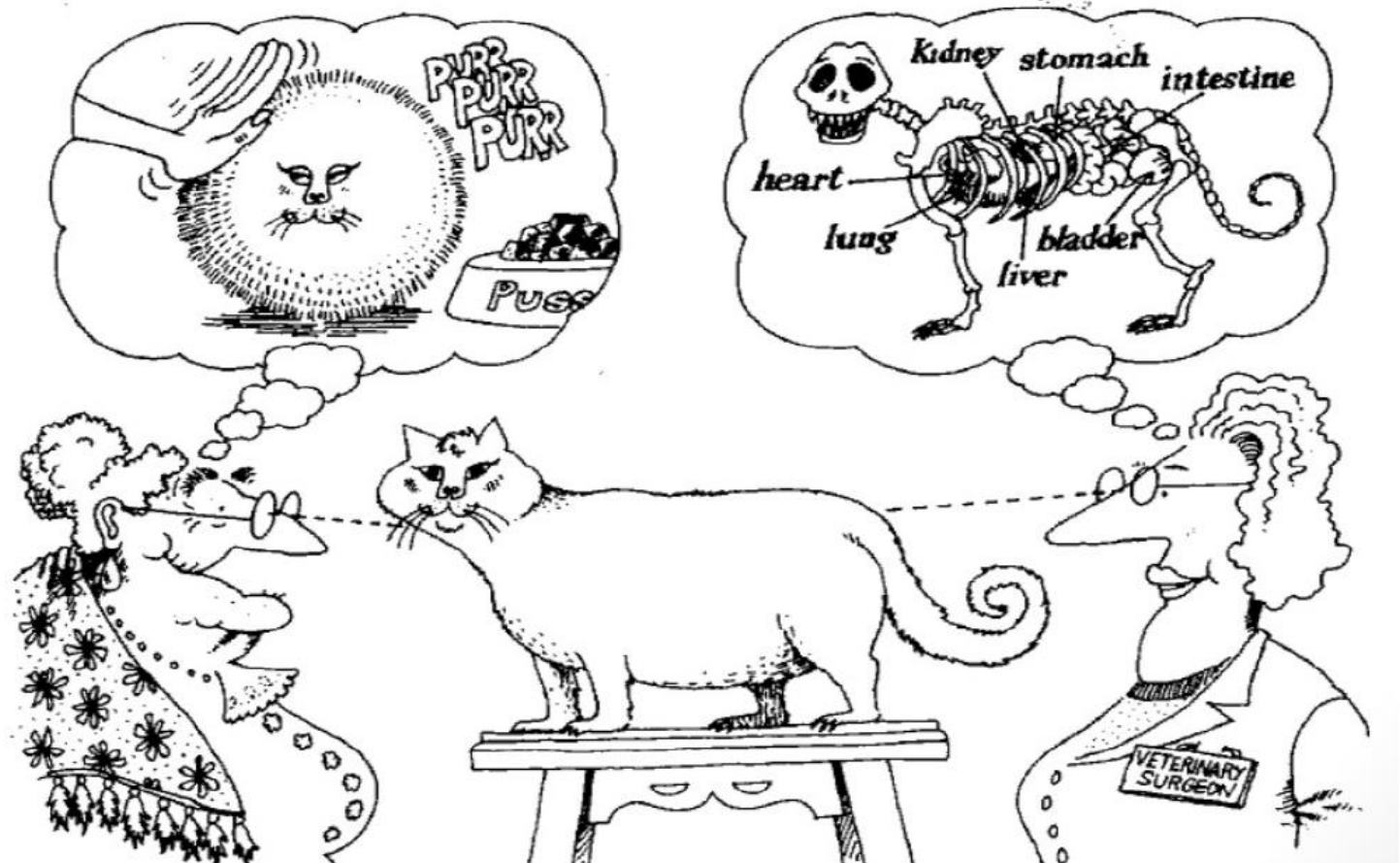
La modélisation orientée objets comporte quatre éléments principaux :

- **l'abstraction** : représentation des caractéristiques essentielles d'un objet qui permettent de le distinguer de tous les autres ;
- **l'encapsulation** : processus de compartimentation des éléments d'une abstraction qui constituent sa structure et son comportement ;
- **la modularité** : capacité qu'a un système d'être décomposé en un ensemble de modules cohérents et faiblement couplés ;
- **l'héritage** : ordonnancement d'abstractions ;

Concepts de l'approche objet (Abstraction)

- Le pilier de l'approche orienté objet :
 - Processus consistant à identifier une entité en mettant en évidence ses caractéristiques pertinentes du point de vue de son utilisation.
 - Caractéristiques essentielles d'une entité qui la distingue de tous les autres types d'entités ; une abstraction définit la frontière relative à la perspective de l'observateur.
- La voiture pour un mécanicien est considérée du point de vue de sa mécanique alors que le concessionnaire va appréhender son équipement et son prix : (mécanique / produit)

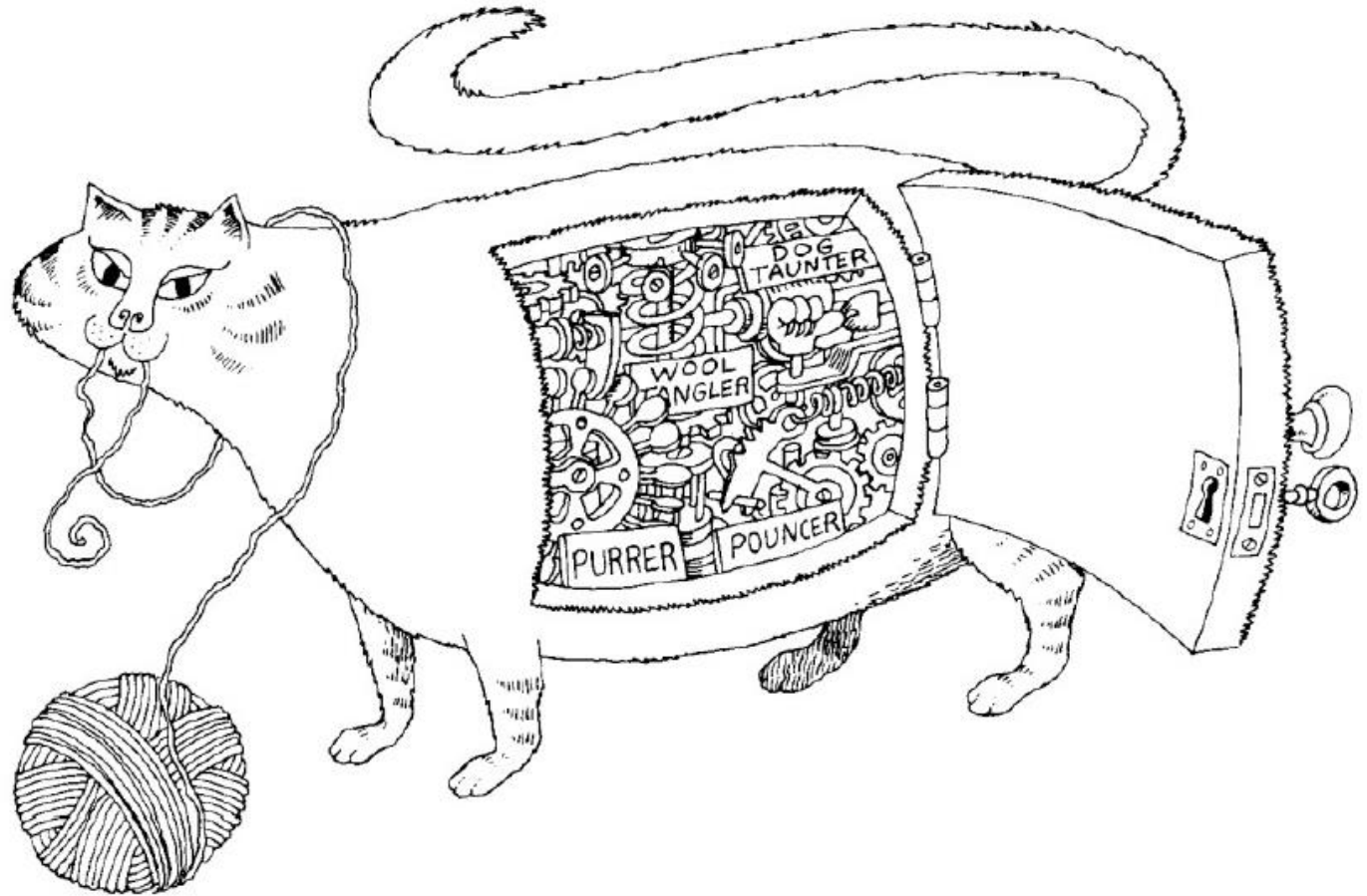
Concepts de l'approche objet (Abstraction)



Concepts de l'approche objet (Encapsulation)

- L'encapsulation est un mécanisme consistant à rassembler les données et les méthodes au sein d'une structure en cachant l'implémentation de l'objet, c'est-à-dire en empêchant l'accès aux données par un autre moyen que les services proposés.
- L'encapsulation permet donc de garantir l'intégrité des données contenues dans l'objet.

Concepts de l'approche objet (Encapsulation)

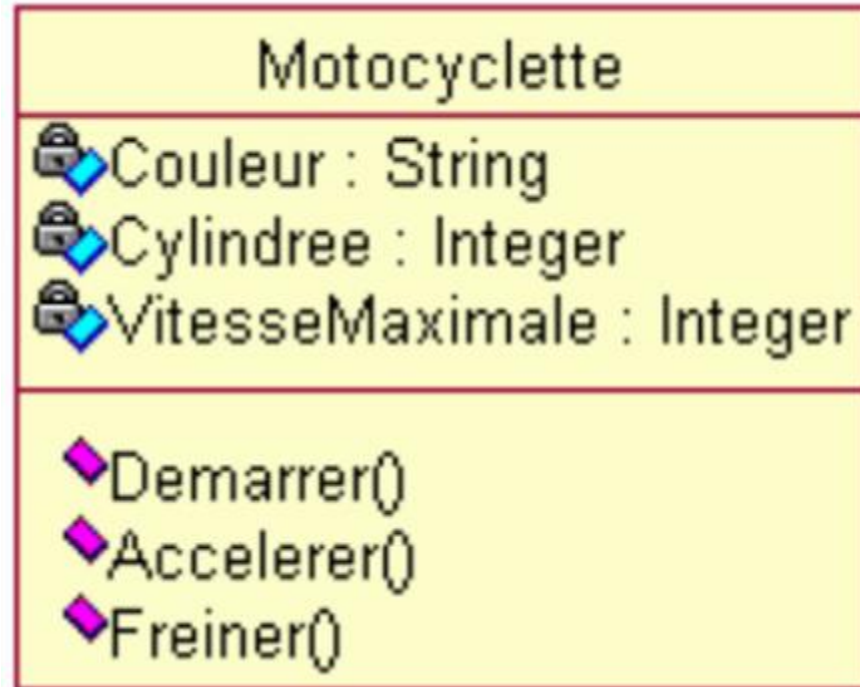


Concepts de l'approche objet (Encapsulation)

Il existe trois niveaux distincts d'encapsulation :

- Niveau privé : Le niveau le plus fort ; la partie privée de la classe est totalement opaque
- Niveau protégé : Le niveau intermédiaire ; les attributs placés dans la partie protégée sont visibles par les classes dérivées de la classe fournisseur. Pour toutes les autres classes, les attributs restent invisibles
- Niveau publique : Les attributs sont visibles pour toutes les classes

Concepts de l'approche objet (Exemple)



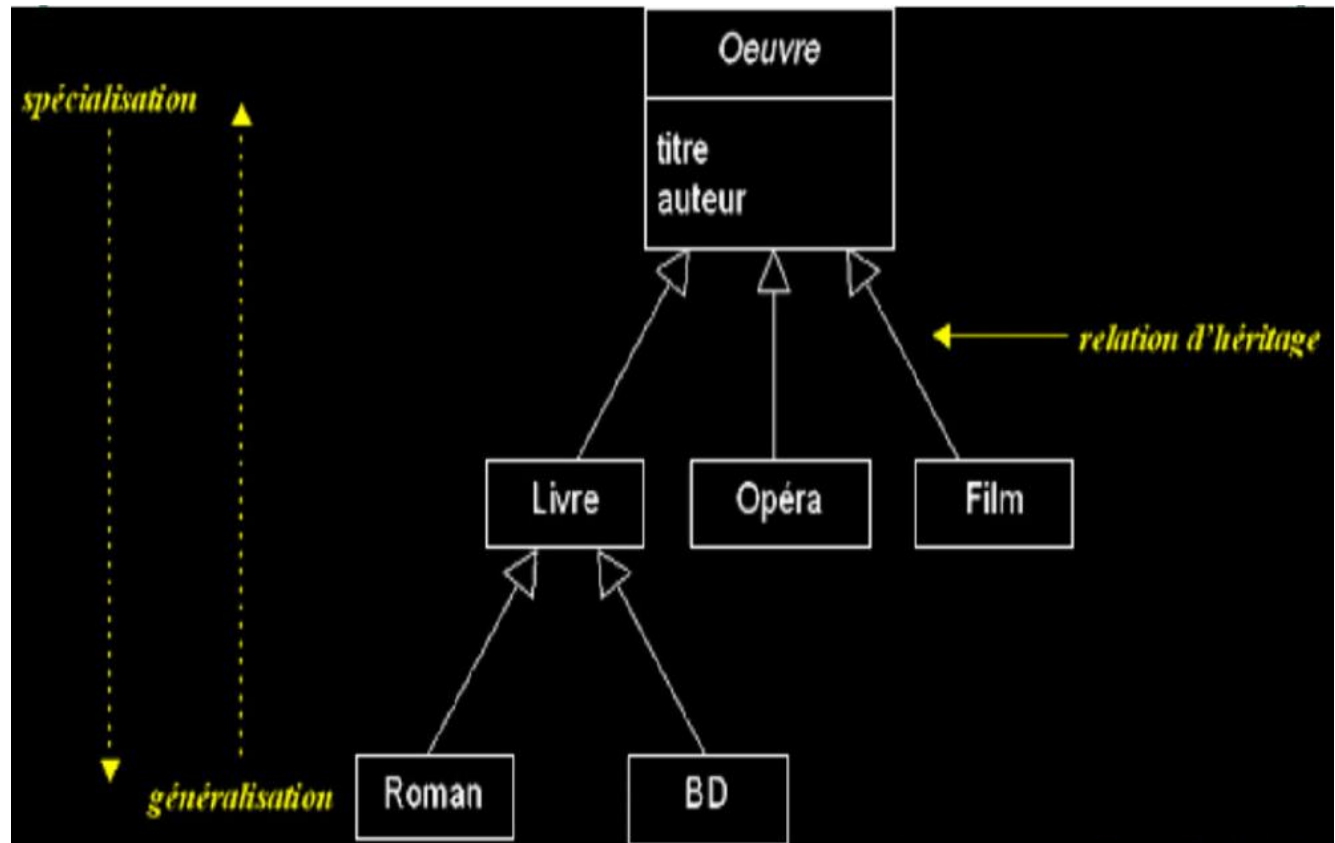
Concepts de l'approche objet (Modularité)

- **Un module** est une construction séparée. L'utilisation de modules permet de contrôler la complexité de grosses applications.
- **La modularité** est la propriété d'un système qui a été décomposé en ensemble de modules regroupant des abstractions logiquement reliées et faiblement couplés

Concepts de l'approche objet (Héritage)

- L'héritage est un mécanisme de transmission des propriétés d'une classe (ses attributs et méthodes) vers une sous-classe.
- Une classe peut être spécialisée en d'autres classes, afin d'y ajouter des caractéristiques spécifiques ou d'en adapter certaines.
- Plusieurs classes peuvent être généralisées en une classe qui les factorise, afin de regrouper les caractéristiques communes d'un ensemble de classes.
- La spécialisation et la généralisation permettent de construire des hiérarchies de classes.
- L'héritage peut être simple ou multiple.
- L'héritage évite la duplication et encourage la réutilisation.

Concepts de l'approche objet (Héritage)



Concepts de l'approche objet (Polymorphisme)

- Le polymorphisme signifie qu'une même opération peut se traduire différemment selon l'objet sur lequel elle s'applique
- Le polymorphisme augmente la généricité du code.

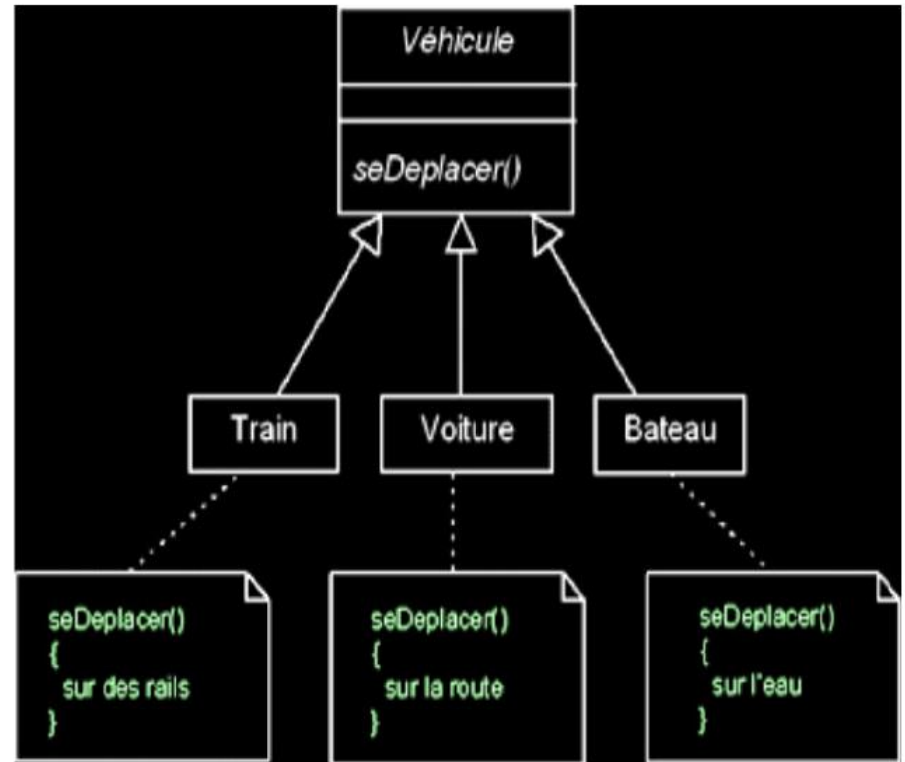
Concepts de l'approche objet (Exemple)

- L'opération déplacer dans un jeu d'échec : le déplacement dépend du type de pièce



Concepts de l'approche objet (Exemple)

```
Vehicule convoi[3] = {  
    Train("TGV"),  
    Voiture("twingo"),  
    Bateau("Titanic")  
};  
  
for (int i = 0; i < 3; i++)  
{  
    convoi[i].seDeplacer();  
}
```



Exercice

- Modéliser la classe Véhicule
 - Attributs
 - Méthodes
- Modéliser la classe Voiture qui hérite de Véhicule
 - Nouveaux attributs et méthodes

Corrigé

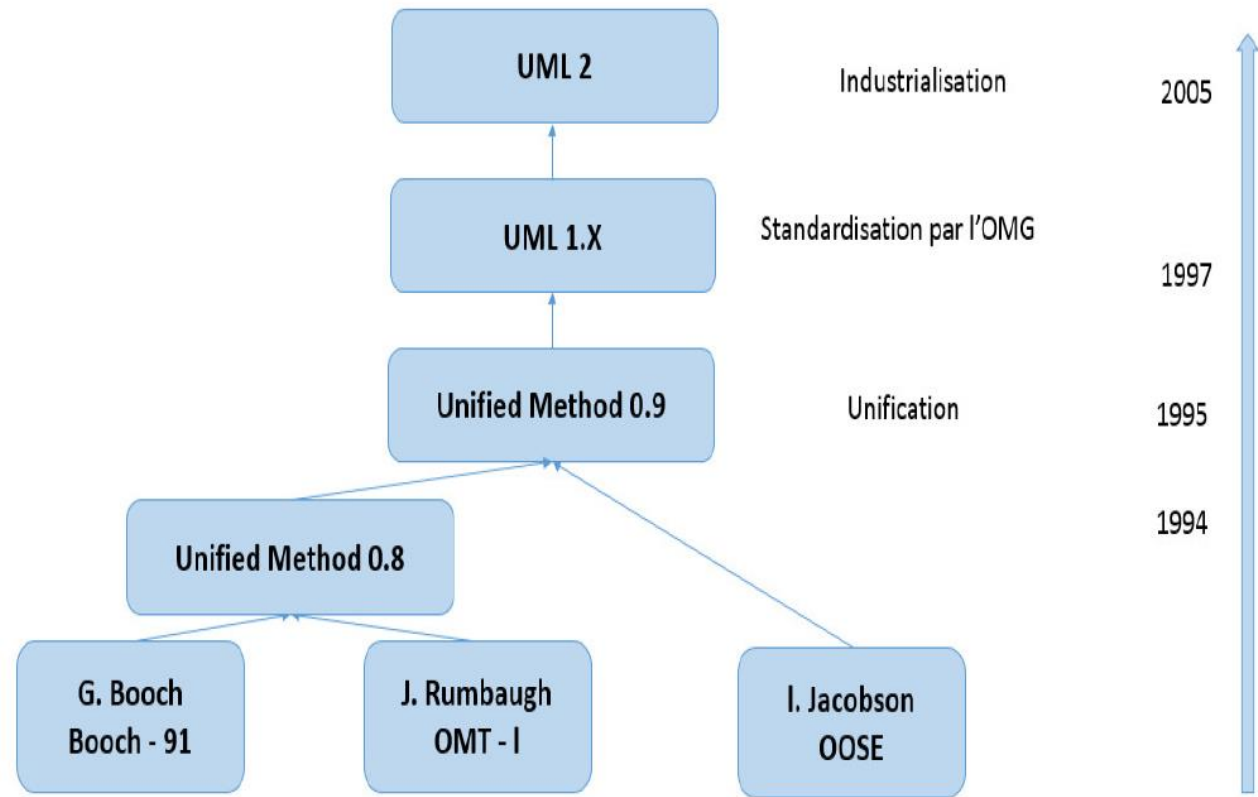
| |
|---|
| Véhicule |
| Marque Année |
| Démarrer Accélérer Avancer Reculer |

| |
|---|
| Voiture |
| + Nombre de portes Puissance Fiscale Couleur |
| + Freiner Vidange |

Unified Modeling Language (Historique)

- Au milieu des années 90, les créateurs d'UML ont décidé de créer un langage de modélisation unifié avec pour objectifs :
 - Modéliser un système des concepts à l'exécutable, en utilisant les techniques orientée objet ;
 - Réduire la complexité de la modélisation ;
 - Utilisable par l'homme comme la machine :
 - Représentations graphiques mais disposant de qualités formelles suffisantes pour être traduites automatiquement en code source ;
 - Officiellement UML est né en 1994.
- UML v2.0 date de 2005. Il s'agit d'une version majeure apportant des innovations radicales et étendant largement le champ d'application d'UML.

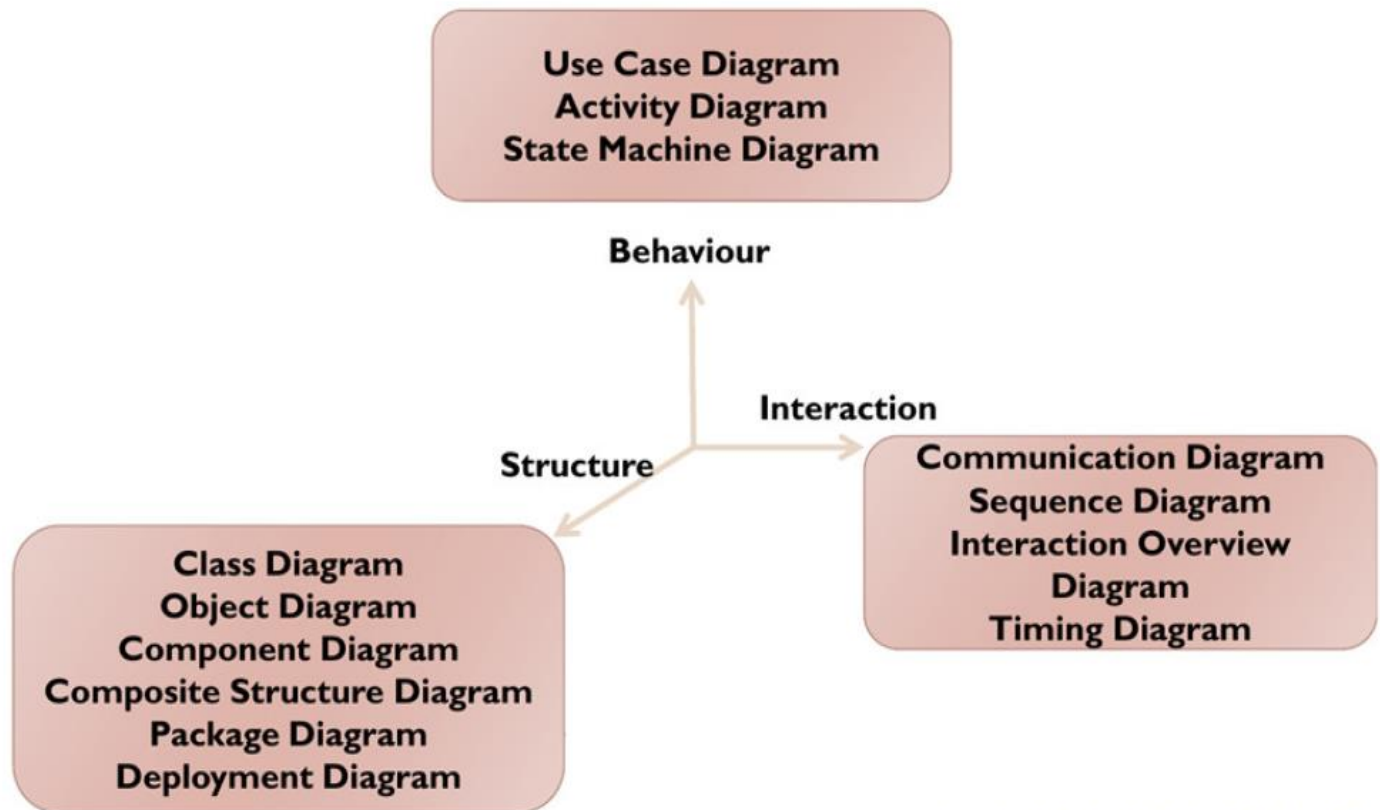
Unified Modeling Language (Historique)



Unified Modeling Language (Définition)

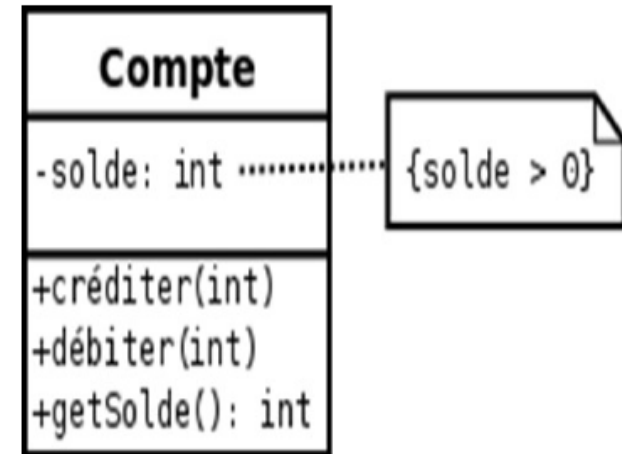
- UML n'est pas une méthode dans la mesure où elle ne présente aucune démarche. A ce titre UML est un formalisme de modélisation objet
- UML n'est donc pas une méthode ou un processus ; en raison de la diversité des cas particuliers
- UML propose un ensemble de notations pour que chacun ait à sa disposition les éléments nécessaires à la conception d'une application
- UML est donc un métalangage : il fournit les éléments permettant de construire le modèle ;
 - les éléments de modélisation (les concepts manipulés par le langage),
 - la sémantique de ces éléments (leur définition et le sens de leur utilisation).

Unified Modeling Language (Diagrammes)



Extensions (Note)

- Une note contient une information textuelle comme un commentaire, un corps de méthode ou une contrainte.
- Graphiquement, elle est représentée par un rectangle dont l'angle supérieur droit est plié.
- On peut relier une note à l'élément qu'elle décrit grâce à une ligne en pointillés. Si elle décrit plusieurs éléments, on dessine une ligne vers chacun d'entre eux.



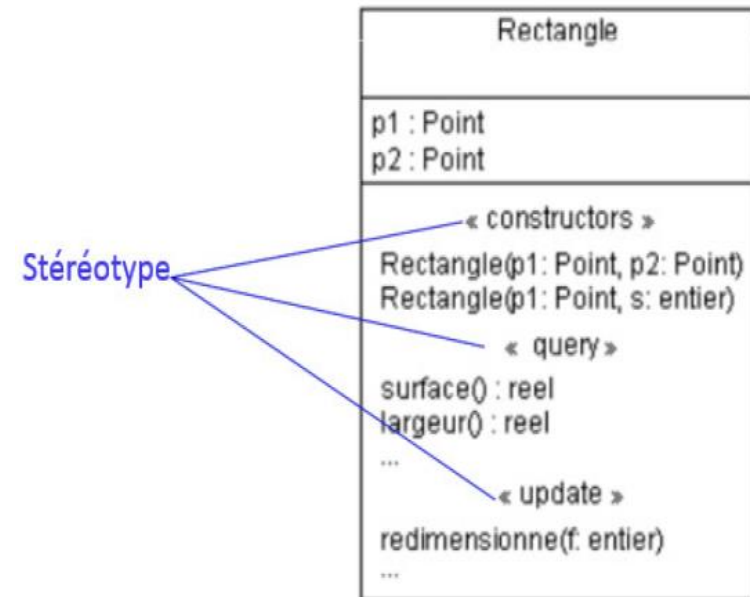
Extensions (stéréotype)

- un stéréotype est rendu comme un nom encadré par des guillemets et placé au-dessus du nom d'un autre élément. Par exemple, dans un diagramme de classe les stéréotypes peuvent être utilisés pour classer les méthodes par comportement telles que « constructor » et « getter ».
- Autrement dit :
 - Annotation s'appliquant sur un élément de modèle.
 - Mieux caractériser des variétés d'un même concept.
 - Sous classer un élément pour le spécialiser en plus fin.
 - Ne change pas les caractéristiques fondamentales des éléments auxquels ils sont appliqués.
 - Représenté par une chaîne de caractères entre : « guillemets »

Extensions (stéréotype)

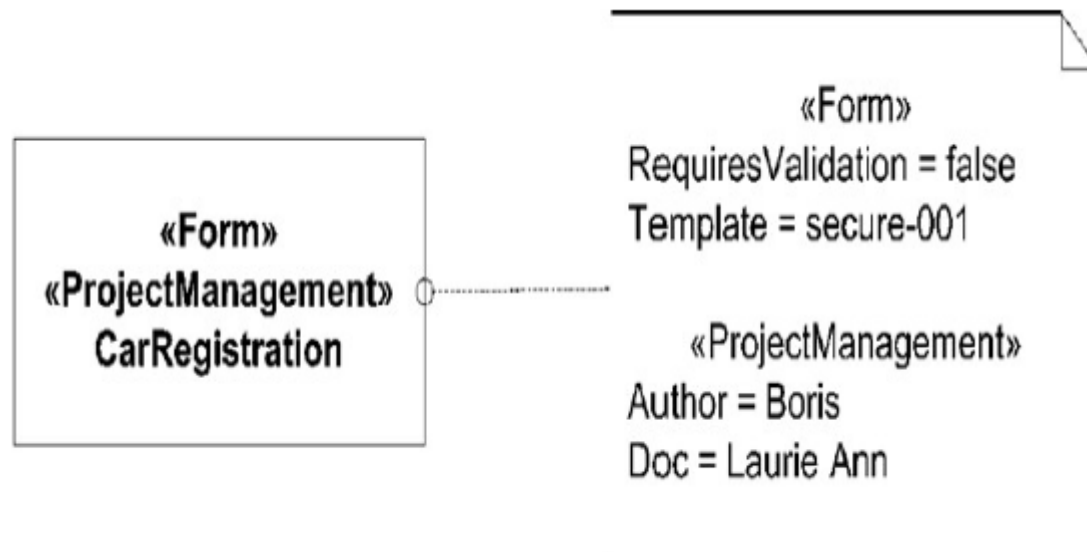
UML propose des stéréotype prédéfinis :

- «interface» : interface
- «énumération» : instances appartiennent à un ensemble fini de littéraux
- «type primitif» : instances sont des valeurs d'un type primitif
- «utilitaire» : variables et procédures globales



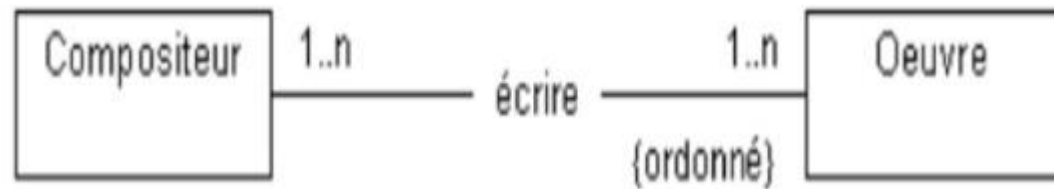
Extensions (Etiquettes)

- Les 'tagged values' permettent la création de nouvelles informations dans la spécification d'un élément.
- Représentation : paire nom, valeur.
- Elles sont souvent utilisés pour indiquer des propriétés relatives à la génération de code.



Extensions (Contrainte)

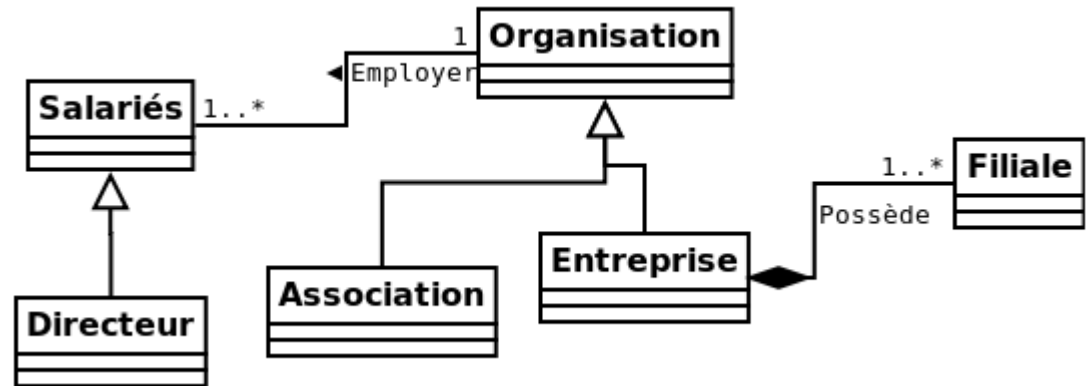
- Chaîne de caractères associée à un élément UML permet d'ajouter de nouvelles règles ou de modifier des règles existantes
- Elles peuvent être exprimées en texte libre ou en utilisant Object Constraint Language (OCL) d'UML.



Mettre l'accent sur l'importance de l'ordre de création des œuvres d'un compositeur

Exercice I

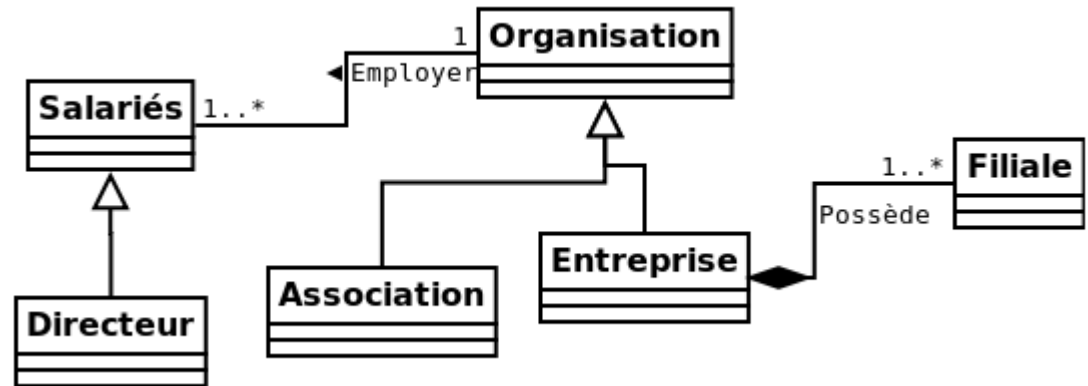
- En analysant le schéma UML ci-après, sélectionner toutes les assertions vraies.



- Une association peut employer un directeur.
- Une association peut employer plusieurs directeurs.
- Une association peut ne pas employer de directeur.
- Il existe des organisations qui ne sont ni des entreprises ni des associations.

Exercice I-Correction

- En analysant le schéma UML ci-après, sélectionner toutes les assertions vraies.



- a) Une association peut employer un directeur.
- b) Une association peut employer plusieurs directeurs.
- c) Une association peut ne pas employer de directeur.
- d) Il existe des organisations qui ne sont ni des entreprises ni des associations.