CHAPITRE I: INTRODUCTION À UML

Nour H. BEN SLIMEN ATTAOUI

Introduction

- **Programmation Orienté Objet**

Modéliser informatiquement des éléments d'une partie du monde réel en un ensemble d'entités informatiques (objets)

- **Intérêt d'une méthode objet**
  - Définir le problème à haut niveau sans rentrer dans les spécificités du langage
  - Définir un problème du façon graphique
  - Utiliser les services offertes par l'objet sans rentrer dans le détail de programmation (Encapsulation)

Introduction

- UML (Unified Modeling Language) est un langage unifié pour la modélisation objet:
  - UML est un langage de modélisation objet qui propose une notation et une sémantique associée à cette notation (i.e. de démarche proposant un enchaînement d'étapes et d'activités qui mènent à la résolution d'un problème posé),
  - UML a une approche entièrement **objet** (i.e. couvrant tout le cycle de développement: analyse, conception et réalisation)

Introduction

- UML est un langage qui sert à communiquer des connaissances sur un sujet.
- Les modèles (diagramme) d'UML facilitent la compréhension du système puisqu'ils contiennent des connaissances essentielles sur le sujet avec différents points de vue.
- UML permet de modéliser différents types de systèmes (système d'information, de gestion, système industriel …)
- UML permet de modéliser le quoi (les besoins) du système et le comment (l'architecture) du système, de visualiser (graphiquement et textuellement) le système avant qu'il ne soit construit.
- UML permet de documenter les modèles utilisés, chaque modèle contient une documentation pour communiquer des connaissances tout au long de la conception.
- UML propose aussi une notation, qui permet de représenter graphiquement les éléments de modélisation du métamodèle.
- Cette notation graphique est le support du langage UML.

Les points forts d'UML

- **UML est un langage formel et normalisé**
  - gain de précision
  - gage de stabilité
  - encourage l'utilisation d'outils
- **UML est un support de communication performant**
  - cadre l'analyse.
  - facilite la compréhension de représentations abstraites complexes.
  - Son caractère polyvalent et sa souplesse en font un langage universel.

Les diagrammes d'UML

- Les diagrammes sont construits à partir d'éléments de base : classe, association, paquetage, etc. parce qu'il est impossible de modéliser un système complexe à partir d'un seul diagramme.
- UML propose un ensemble de diagrammes qui permettent de se concentrer sur plusieurs aspects du système. Chaque diagramme véhicule une sémantique précise. Combinés, ces différents diagrammes offrent une vue complète des aspects statiques et dynamiques du système. Les différents diagrammes sont les suivants :

Les diagrammes d'UML

- **Vues statiques:**
  - Les diagrammes de cas d'utilisation décrivent le comportement et les fonctions d'un système du point de vue de l'utilisateur
  - Les diagrammes de classes décrivent la structure statique, les types et les relations des ensembles d'objets
  - Les diagrammes d'objets décrivent les objets d 'un système et leurs relations
  - Les diagrammes de composants décrivent les composants physiques et l'architecture interne d'un logiciel
  - Les diagrammes de déploiement décrivent la répartition des programmes exécutables sur les différents matériels

Les diagrammes d'UML

- **Vues dynamiques :**
  - Les diagrammes de collaboration décrivent les messages entre objets (liens et interactions)
  - Les diagrammes d'états-transitions décrivent les différents états d'un objet
  - Les diagrammes d'activités décrivent les comportements d'une opération (en termes d'actions)
  - Les diagrammes de séquence décrivent de manière temporelle les interactions entre objets et acteur

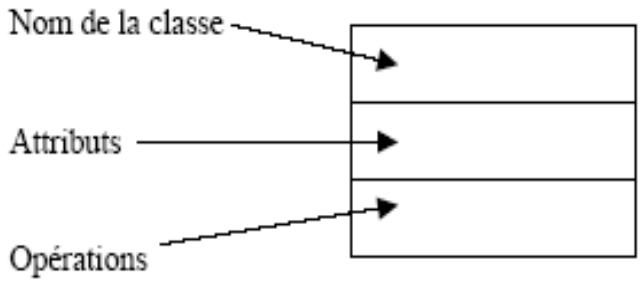
Chapitre II: Diagramme des classes (DCL)

Nour H. BEN SLIMEN ATTAWI

UML : DIAGRAMME DE CLASSES

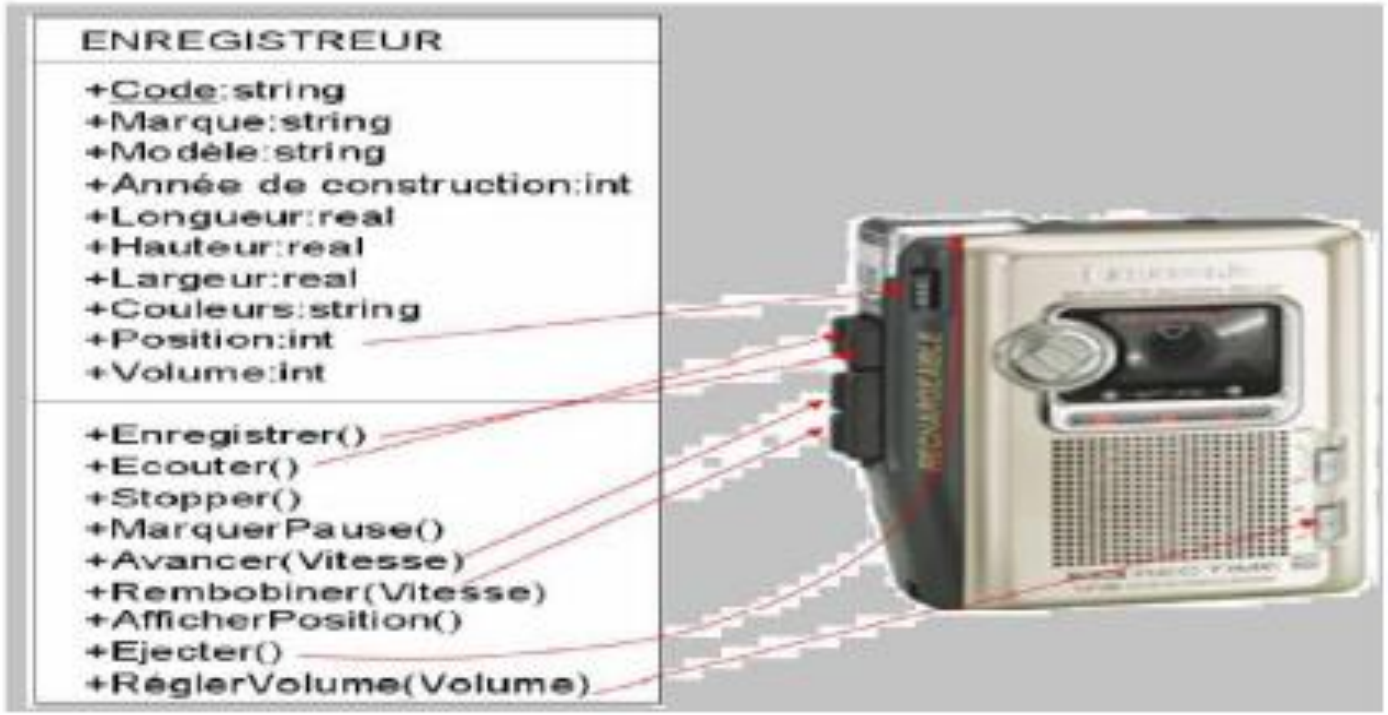
Une Classe

- La classe est un modèle décrivant les caractéristiques communes et le comportement d'un ensemble d'objets
- Elle modélise l'ensemble des objets ayant :
  - attributs similaires
  - comportement en commun (les opérations)
  - relations communes avec d'autres objets
- Une classe est formée par un ensemble d'attributs et d'opérations



UML : DIAGRAMME DE CLASSES

- Exemple:



UML : DIAGRAMME DE CLASSES

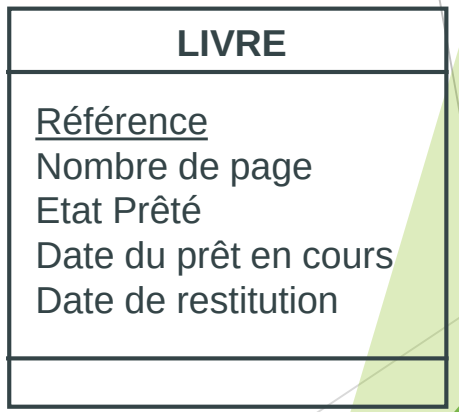
APPROCHE OBJET:

- Représentation d'un concept ou d'une chose, ayant des limites et un sens dans un contexte donné.
- Physique ou Abstraite
- Caractéristiques :
  - **Informations :** attributs décrivant l'objet
  - **Identifiant :** permet d'individualiser les objets entre eux
  - **Comportement :** implémenter sous forme de « méthodes »
  - **Messages :** moyen de communication entre objets
  - **Encapsulation :** gestion des accès

UML : DIAGRAMME DE CLASSES

ATTRIBUT :

- Donnée gérée par la classe (donc pour toutes les instances)
- Valeur unique par instance (mais la même peut être dans plusieurs occurrences)
- Identifiant est un attribut :
  - particulier, autant de valeurs que d'instances
  - valeurs déterminent les instances de la classe



UML : DIAGRAMME DE CLASSES

ATTRIBUT :

- **La forme la plus courte d'un attribut est: nom de l'attribut**
- **La forme la plus complète d'un attribut est la suivante:**

Visibilité nom [multiplicité] : type = valeur initiale

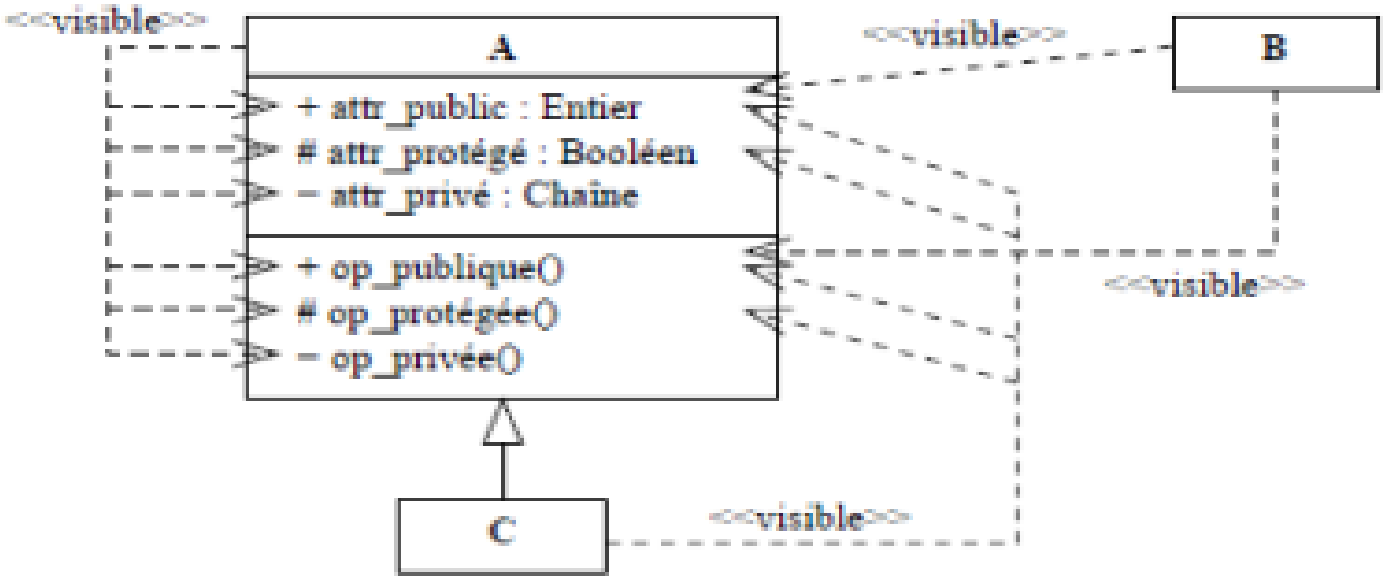
1. la visibilité :

- publique : l'attribut peut être utilisé par plusieurs classes, cette caractéristique est indiquée par le symbole : +
- protégé : l'accessibilité est limitée à la classe et aux sous classes, cette caractéristique est indiquée par le symbole : #
- privé : l'accessibilité est limitée à la classe uniquement, cette caractéristique est indiquée par le symbole : -

Si aucun symbole n'est spécifié, la visibilité est par défaut **privée**.

UML : DIAGRAMME DE CLASSES

- La visibilité concerne les attributs et les méthodes



UML : DIAGRAMME DE CLASSES

2. **Multiplicité:**

Représente le nombre d'instances qu'un attribut peut avoir, c'est une extension de la multiplicité. Exemple : adresse [1..3] (une, deux ou trois adresses : domicile fixe, bureau, vacances)

3. **Type**

Peuvent être des types simples (chaîne de caractères, entier, booléen,…) ou des types complexes (enregistrement, classe,…).

4. **Valeur initiale**

c'est une valeur prise par défaut par un attribut. Exemple : x : entier=0.

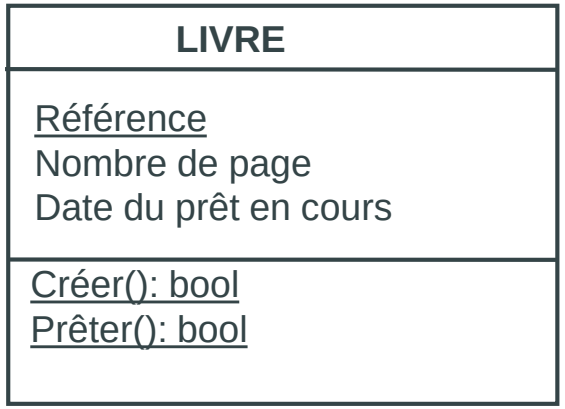
UML : DIAGRAMME DE CLASSES

Attribut dérivé :

- L'attribut est dérivé d'autres attributs de la classe, cette caractéristique est indiquée par le symbole: ***/*** .

Exemple : */ age (date système – date naissance)*.

UML : DIAGRAMME DE CLASSES



- **OPERATIONS - METHODES:**
  - **Opération :** effectuée par l'objet
  - **Méthode :** effectuée par la classe
- UML différencie l'opération de la méthode. Une méthode est l'implémentation de l'opération, c'est un algorithme exécutable, généralement désigné dans un langage de programmation ou du texte structuré.

UML : DIAGRAMME DE CLASSES

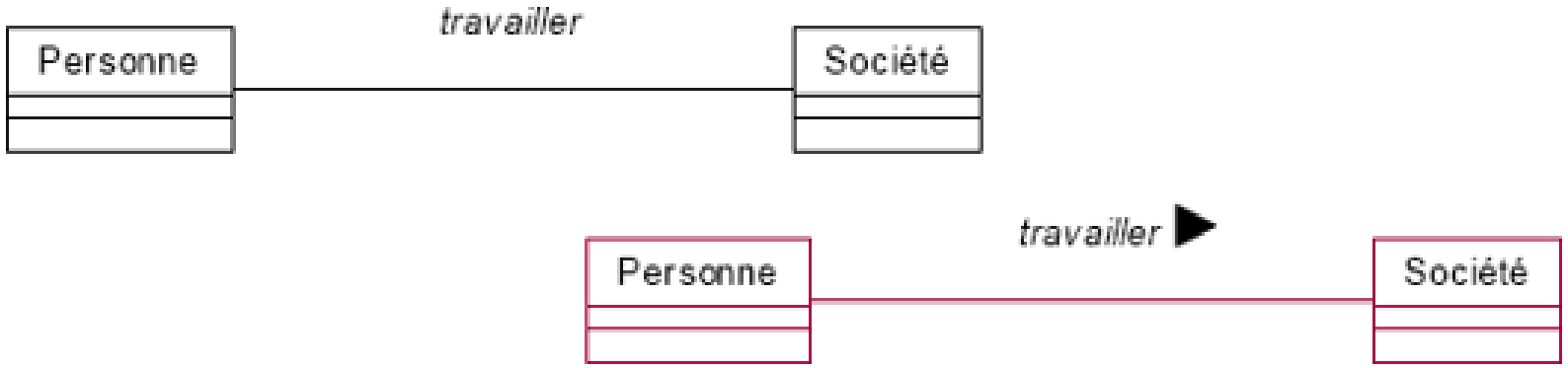
- **LIENS & ASSOCIATIONS:**
  - **Lien :**
    - connexion physique ou conceptuelle entre instances de classes
  - **Association:**
    - Instance du lien (ex : rédiger, créer)
  - **Rôle :**
    - Spécifie la contribution (ex : être rédigée par)

UML : DIAGRAMME DE CLASSES

Le nom de l'association:

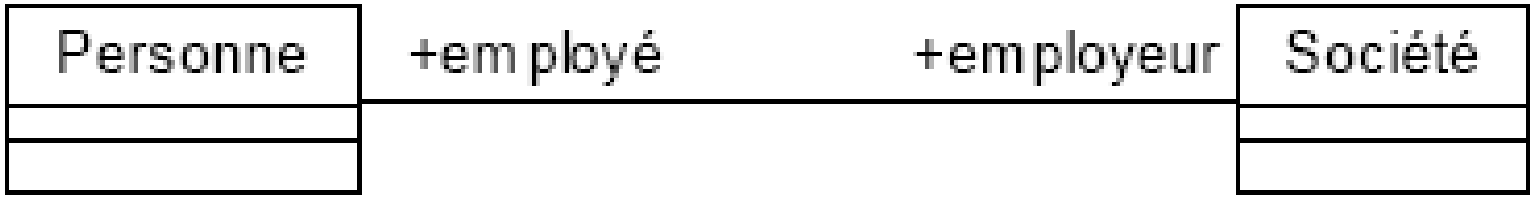
- Il sert à décrire la nature de la relation. Pour éviter les ambiguïtés, on peut donner un sens de lecture à ce nom en utilisant un triangle qui pointe dans le sens désiré.

Exemple



UML : DIAGRAMME DE CLASSES

- **Le rôle d'une association:** quand une classe participe à une association, elle y joue un rôle spécifique, il est possible de nommer explicitement ce rôle
- Exemple:



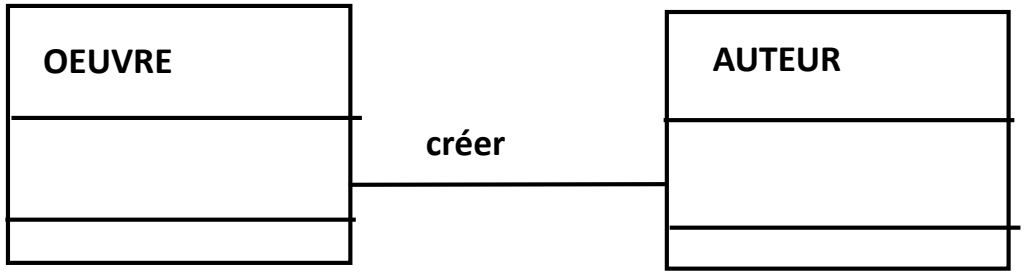
UML : DIAGRAMME DE CLASSES

- **MULTIPLICITES (cardinalités):** représente combien d'instances d'une classe participent à une seule instance d'une classe associée.

1	une instance et une seule
0..1	Zéro ou Une instance
M..N	intervalle fixe de nombre d'instances (entiers naturels)
0..*	Aucune ou plusieurs instances
*	
1..*	Au moins une instance ou plusieurs
N	Nombre fixe d'instances(entier naturel)

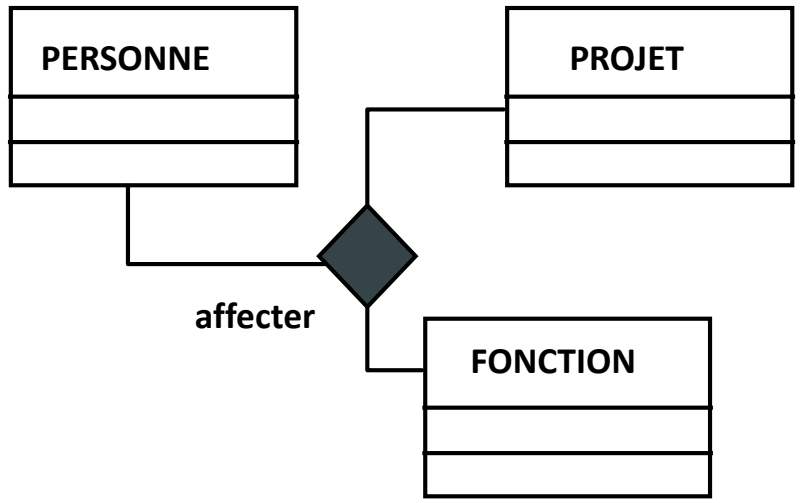
UML : DIAGRAMME DE CLASSES

- **TYPES ASSOCIATIONS:**
  - **Association binaire :** associe 2 classes



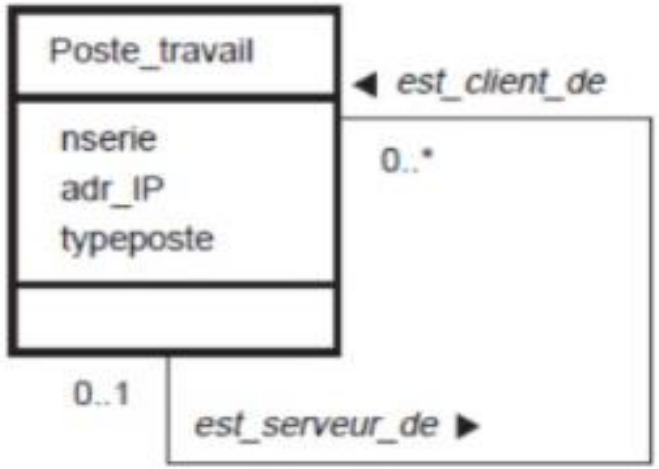
UML : DIAGRAMME DE CLASSES

- **TYPES ASSOCIATIONS:**
  - **Association n-aire :** associe plus de 2 classes



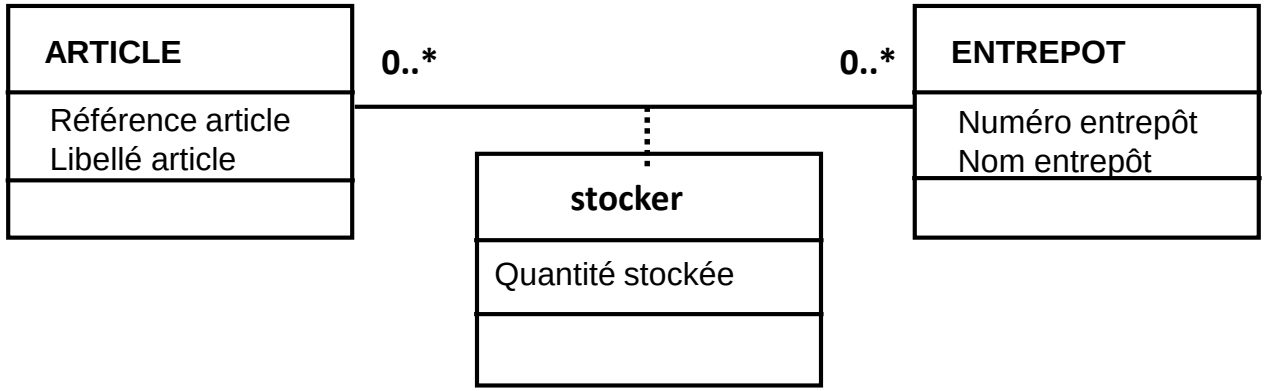
UML : DIAGRAMME DE CLASSES

- **TYPES ASSOCIATIONS:**
  - **Association réflexive :** associe une classe à elle même
    - *Dans ce cas, indispensable de nommer les associations (rôles)*



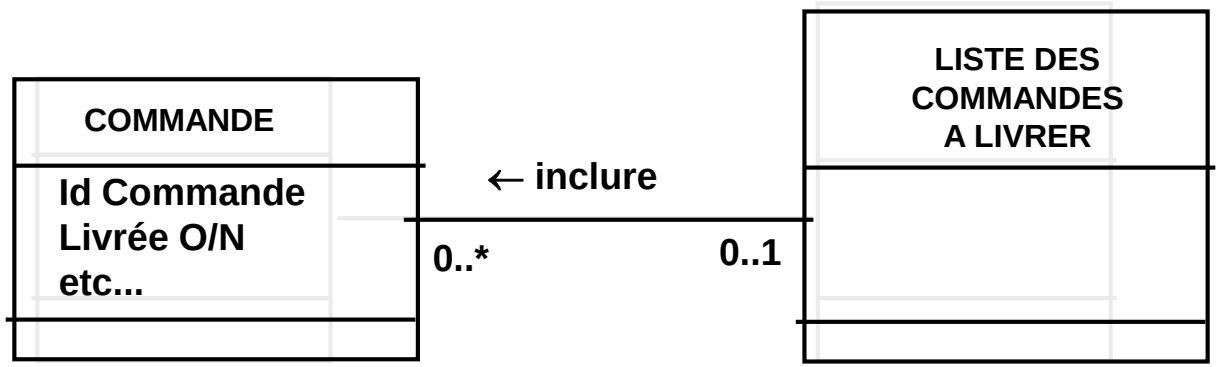
UML : DIAGRAMME DE CLASSES

- **TYPES ASSOCIATIONS:**
  - **Classe d'association:** classe liée à une association
  - Permet de paramétrer une association entre deux classe par une classe,



UML : DIAGRAMME DE CLASSES

- **CLASSES PERSISTANTES :** une classe est persistante quand elle perdure dans le temps



Classe persistante

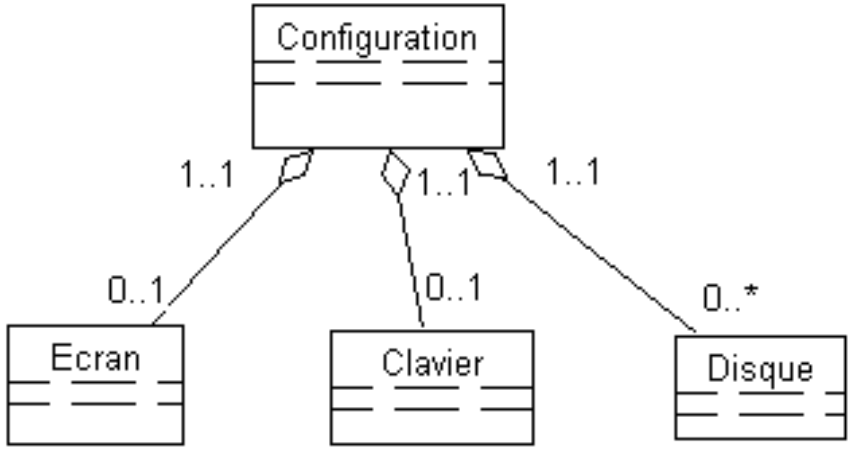
Classe non persistante

UML : DIAGRAMME DE CLASSES

- **AGREGATION**

C’est une forme spéciale d’association qui permet de montrer qu’une classe est une partie d’une autre classe, c’est une relation composé-composant,

Exemple:



UML : DIAGRAMME DE CLASSES

- **COMPOSITION**

La composition est un cas particulier d'agrégation dans laquelle

- La composition se représente par un losange noir coté composé.
- La destruction de l'objet composite implique la destruction de ses composants.
- Dans la composition, la multiplicité du côté composite ne doit pas être supérieure à 1 (i.e. 1 ou 0..1).

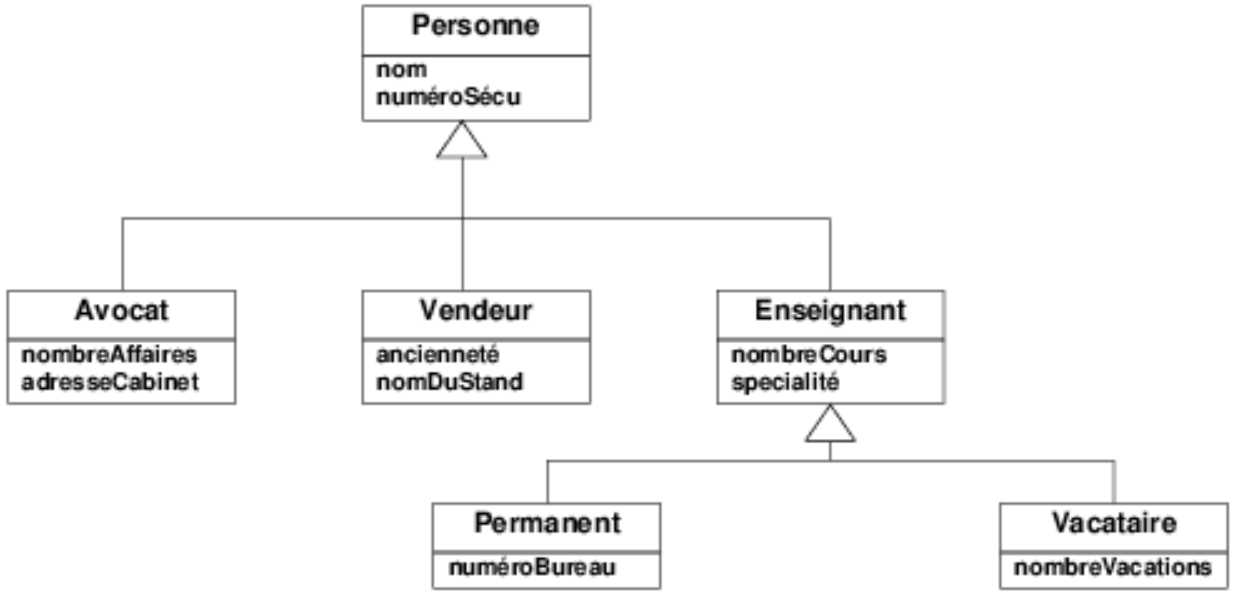
Exemple:

UML : DIAGRAMME DE CLASSES

- **GENERALISATION/SPECIALISATION**
  - Le but de la généralisation est la factorisation des caractéristiques de plusieurs classes appelées classes enfants dans une classe appelée classe parente,
  - La classe parente rassemble donc les attributs et les opérations communes des classes initiales qui deviennent des classes enfants
  - La généralisation/spécialisation s'appelle aussi un héritage et exprime intuitivement le fait que les classes enfants « héritent » des caractéristiques de leurs classes parentes.

UML : DIAGRAMME DE CLASSES

Exemple:



- Un objet Personne peut désigner une instance des classes Personne Avocat, Vendeur, Enseignant Vacataire et Permanent ;
- un objet Enseignant peut désigner une instance des classes Enseignant Vacataire et Permanent ;