

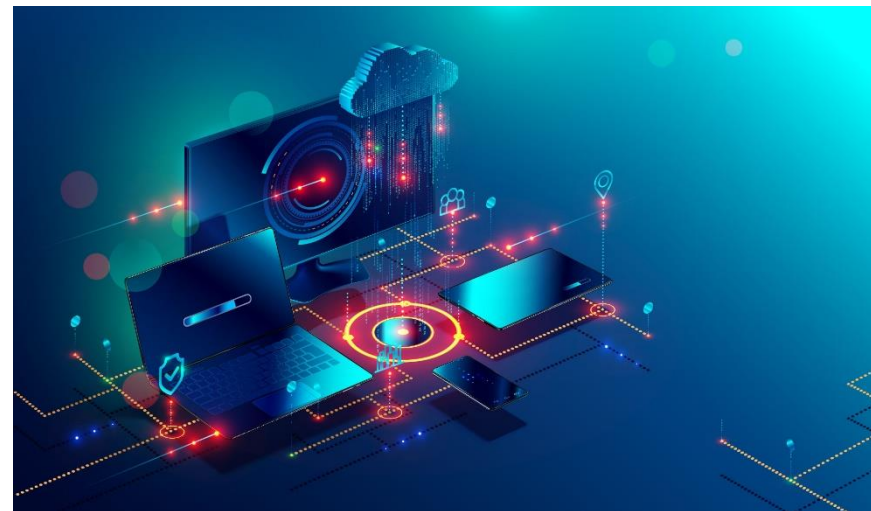
Analyse Orientée Objet

I. Introduction

II. Approche objet et système d'information.

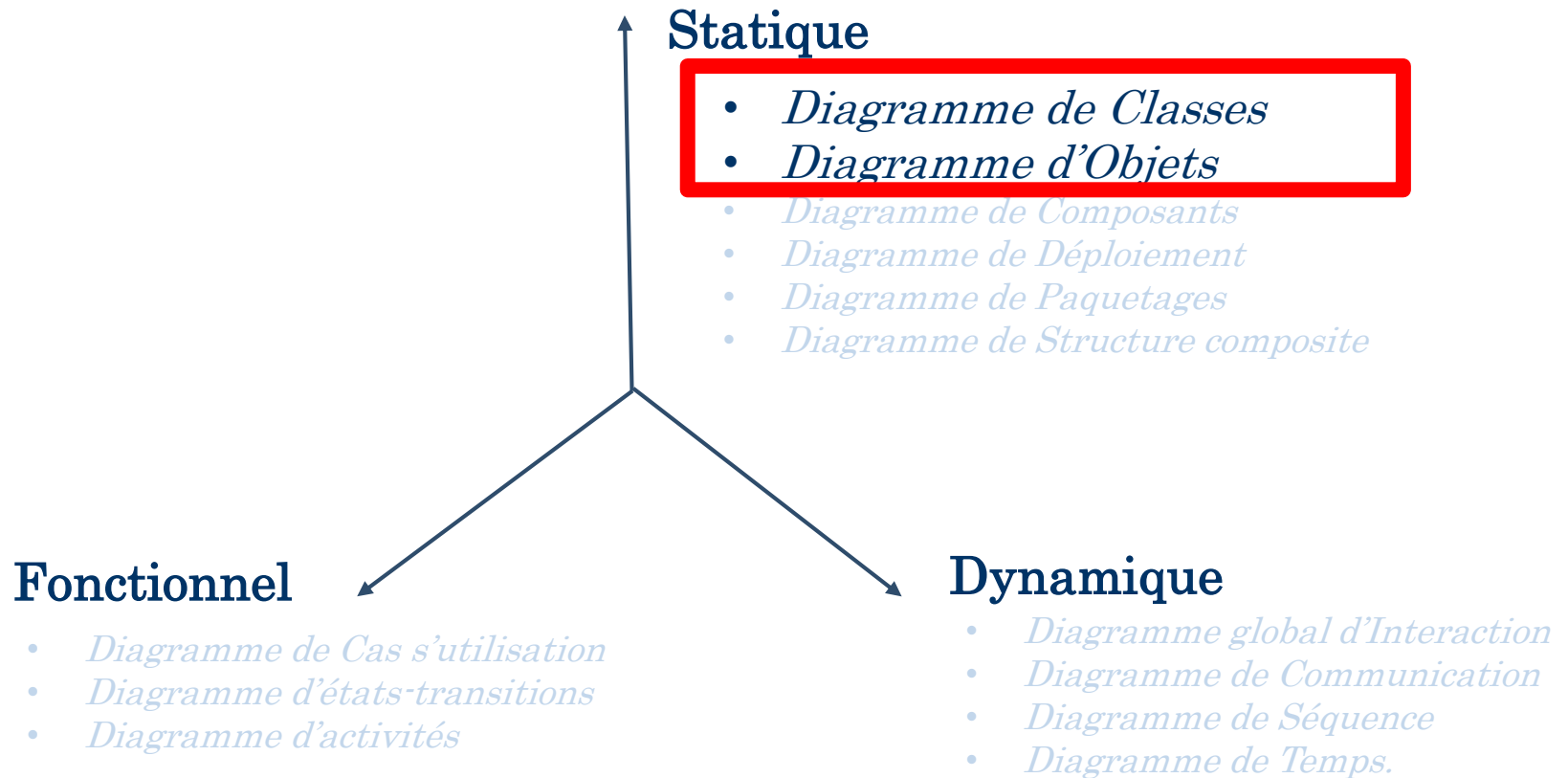
III. Principes Objet

IV. UML



IV. UML

3. Les diagrammes UML.



IV. UML

Diagramme de Classes/objets

1. Objectif.
2. Classes et objets.
3. Relations entre classes.
4. Diagramme d'objets.
5. Construction d'un diagramme de classes.
6. Package.

IV. UML

Diagramme de Classes/objets

1. **Objectif.**
2. Classes et objets.
3. Relations entre classes.
4. Diagramme d'objets.
5. Construction d'un diagramme de classes.
6. Package.

IV. UML

Diagramme de Classes/objets

1. Objectif:

Un diagramme des classes:

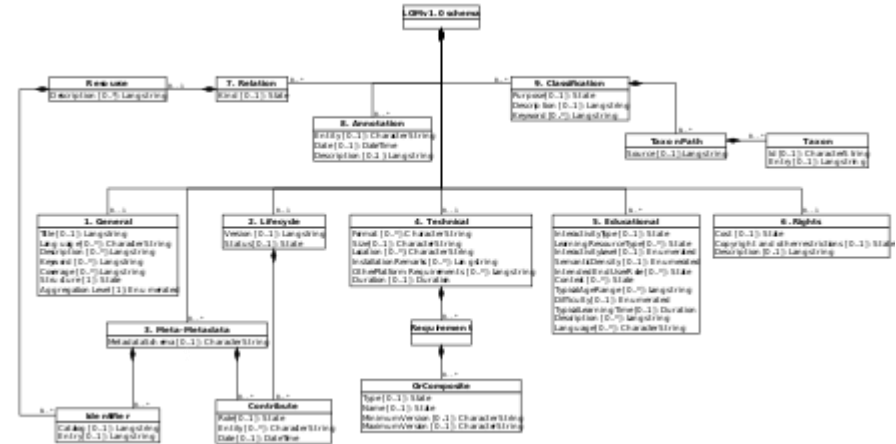
- Représente la structure **statique** et **interne** d'un système.
- Permet donc de représenter l'ensemble des **informations** gérées par le domaine.
- Les informations sont **structurées** et **groupées** dans des **classes**.

Le diagramme des classes est considéré comme le plus important de la modélisation orientée objet.

IV. UML

Diagramme de Classes/objets

1. Objectif:



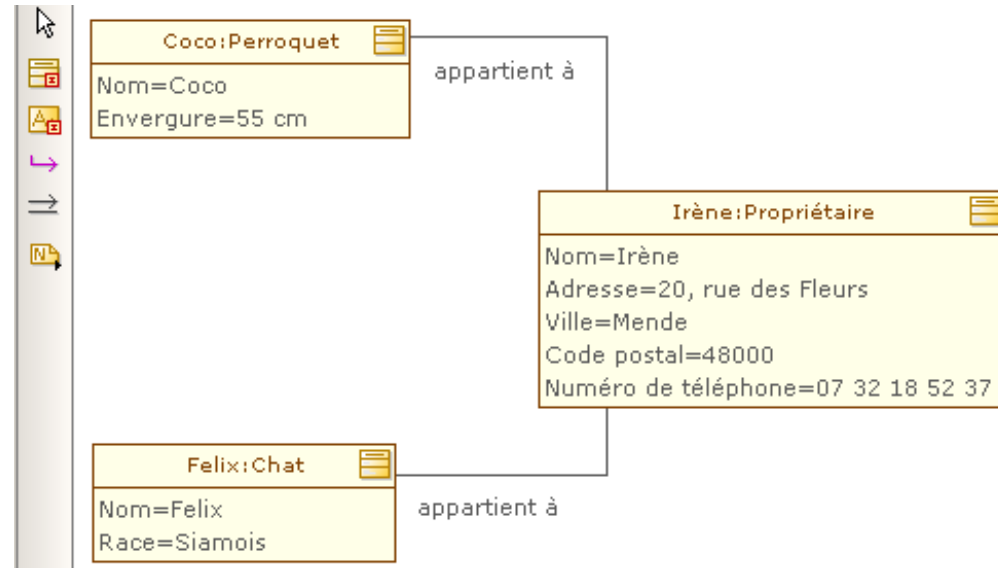
Toutes les informations mémorisées, manipulées, transformées, analysées et partagées pour accomplir les finalités du domaine

➡ *doivent figurer* dans le diagramme des classes.

IV. UML

Diagramme de Classes/objets

1. Objectif:



Un diagramme d'objets:

- Représente les **objets**(instances de classes) et **leurs liens** pour donner une **vue figée** du système à **un instant** donné.
 - Permet d'illustrer le diagramme des classes en se basant sur un exemple qui explique le modèle.
- => diagramme des occurrences en analyse relationnelle.

IV. UML

Diagramme de Classes/objets

1. Objectif.
- 2. Classes et objets.**
3. Relations entre classes.
4. Diagramme d'objets.
5. Construction d'un diagramme de classes.

IV. UML

Diagramme de Classes/objets

2. Classes et objets:

- Définitions.
- Représentation.
- Encapsulation et visibilité.
- Attributs.
- Méthodes.

IV. UML

Diagramme de Classes/objets

2. Classes et objets:

- Définitions.
- Représentation.
- Encapsulation et visibilité.
- Attributs.
- Méthodes.

IV. UML

Diagramme de Classes/objets

2. Classes et objets:

➤ Définitions:

Une **classe** est la description d'un ensemble d'objets ayant une **sémantique**, des **attributs** et des **méthodes** communes.

La **classe** spécifie l'ensemble des caractéristiques qui composent des objets de même type.

IV. UML

Diagramme de Classes/objets

2. Classes et objets:

Une classe représente des éléments:

- **Concrets** (Personnes, Voitures, Avions, ...).
- **Abstrait**s (Commandes, Contrats, ...).
- **Comportementaux** (Tâches, Traitements, ...)

IV. UML

Diagramme de Classes/objets

2. Classes et objets:

➤ Définitions:

Un **objet** est une instance d'une classe.

Un objet est la concrétisation d'une classe.

C'est une entité qui a une **identité**, un **état** et un **comportement**.

Objet = Etat + Comportement + Identité.

IV. UML

Diagramme de Classes/objets

2. Classes et objets:

- Définitions.
- Représentation.
- Encapsulation et visibilité.
- Attributs.
- Méthodes.

IV. UML

Diagramme de Classes/objets

2. Classes et objets:

➤ Représentation:

Une **classe** est composée de compartiments:

Compartiment 1 : Nom de la classe.

Compartiment 2 : Attributs.

Compartiment 3 : Opérations.

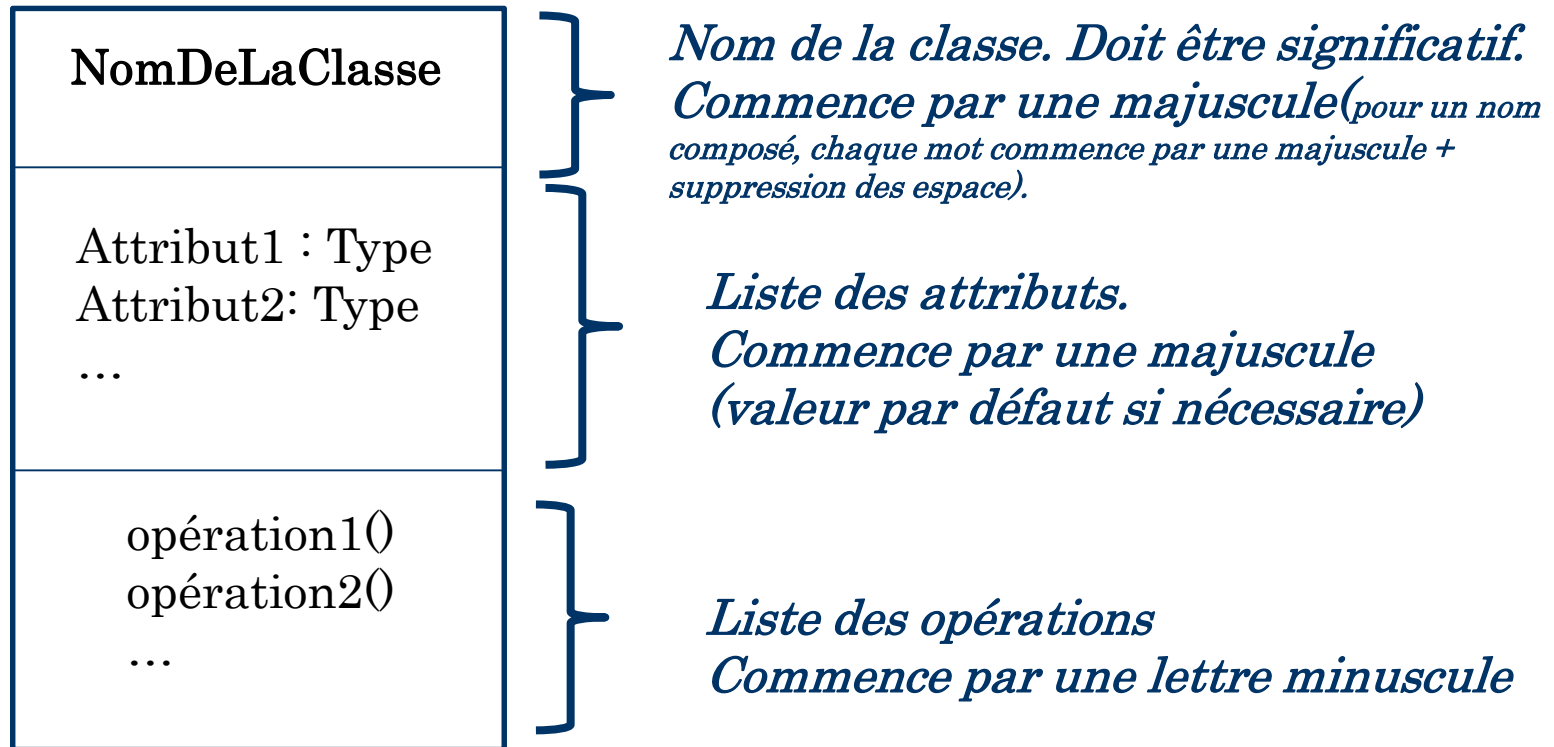
Possibilité d'omettre des attributs et/ou opérations.

IV. UML

Diagramme de Classes/objets

2. Classes et objets:

➤ Représentation: Classe



D'autres compartiments peuvent être ajoutés: responsabilités, exceptions, ...

IV. UML

Diagramme de Classes/objets

2. Classes et objets:

➤ Exemple: Classe Etudiant

Etudiant

Nom : String
Prénom: String
DateNaissance: Date
Sexe: String {M, F}

calculAge() : Integer



IV. UML

Diagramme de Classes/objets

2. Classes et objets:

➤ Exemple: Classe Voiture

Voiture
Marque : String Puissance : String Cylindr� : Integer
rouler() d�marrer() arr�ter()

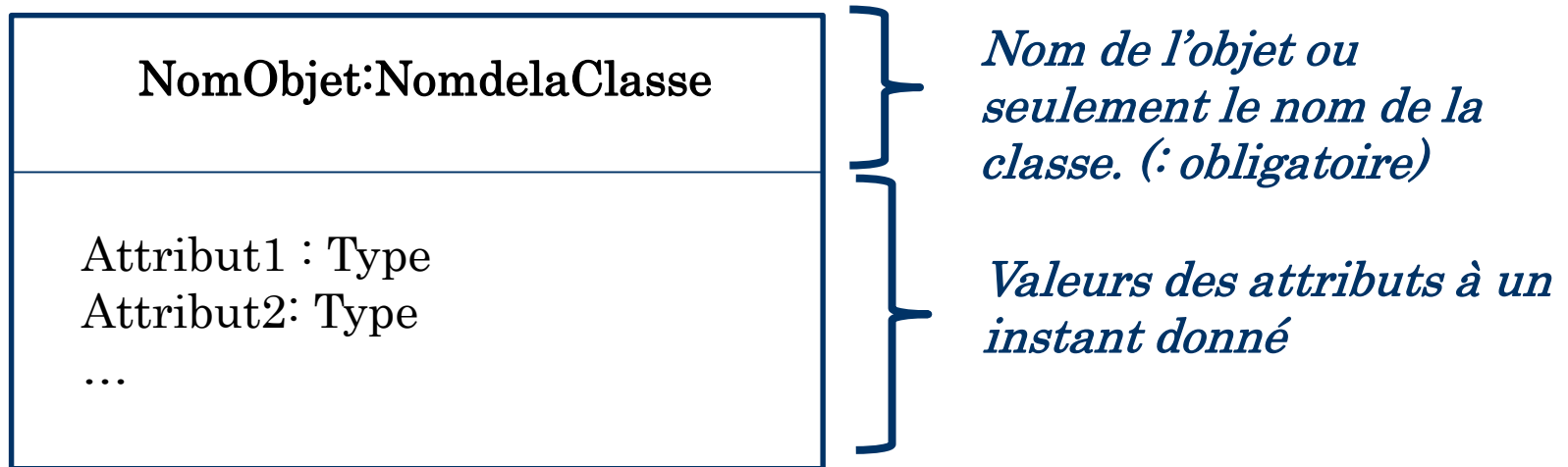


IV. UML

Diagramme de Classes/objets

2. Classes et objets:

➤ Représentation: Objet

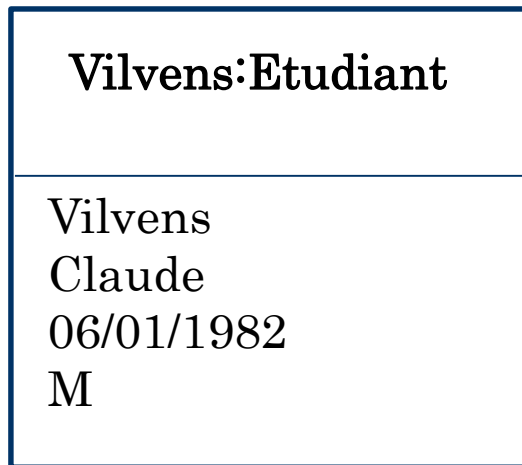


IV. UML

Diagramme de Classes/objets

2. Classes et objets:

➤ Exemple: Objet



IV. UML

Diagramme de Classes/objets

2. Classes et objets:

- Définitions.
- Représentation.
- Encapsulation et visibilité.
- Attributs.
- Méthodes.

IV. UML

Diagramme de Classes/objets

2. Classes et objets:

➤ Encapsulation et visibilité

Un **objet** est défini à la fois par des informations et par des comportements.

L'**encapsulation** consiste à rassembler les données et les méthodes au sein d'une structure en cachant l'implémentation de l'objet.

IV. UML

Diagramme de Classes/objets

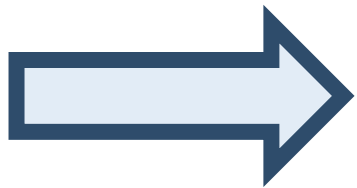
2. Classes et objets:

➤ Encapsulation et visibilité

L'**encapsulation** empêche l'accès aux données par un autre moyen que les services proposés.

Permet donc de garantir l'intégrité des données contenues dans l'objet.

(Seul les détenteurs d'une clé peuvent ouvrir le coffre fort).



Définir des niveaux de visibilités des éléments.

IV. UML

Diagramme de Classes/objets

2. Classes et objets:

➤ Encapsulation et visibilité

La **visibilité** déclare la *possibilité* ou l'*impossibilité* par un élément d'être accessible depuis un autre espace.

IV. UML

Diagramme de Classes/objets

2. Classes et objets:

➤ Encapsulation et visibilité

Il existe quatre visibilités prédéfinies:

Public (+)	Accès autorisé pour tout élément.
Private (-)	Accès autorisé seulement par un élément de la classe.
Protected (#)	Accès autorisé par un élément de la classe ou de ses descendants.
Package (~) ou rien Tildé	Accès autorisé par les seuls éléments du même package.

IV. UML

Diagramme de Classes/objets

2. Classes et objets:

- Définitions.
- Représentation.
- Encapsulation et visibilité.
- Attributs.
- Méthodes.

IV. UML

Diagramme de Classes/objets

2. Classes et objets:

➤ Attributs

Une **classe** correspond à un concept global d'information et se **compose d'un ensemble** d'informations élémentaires: **Attributs**

Syntaxe de déclaration d'un attribut

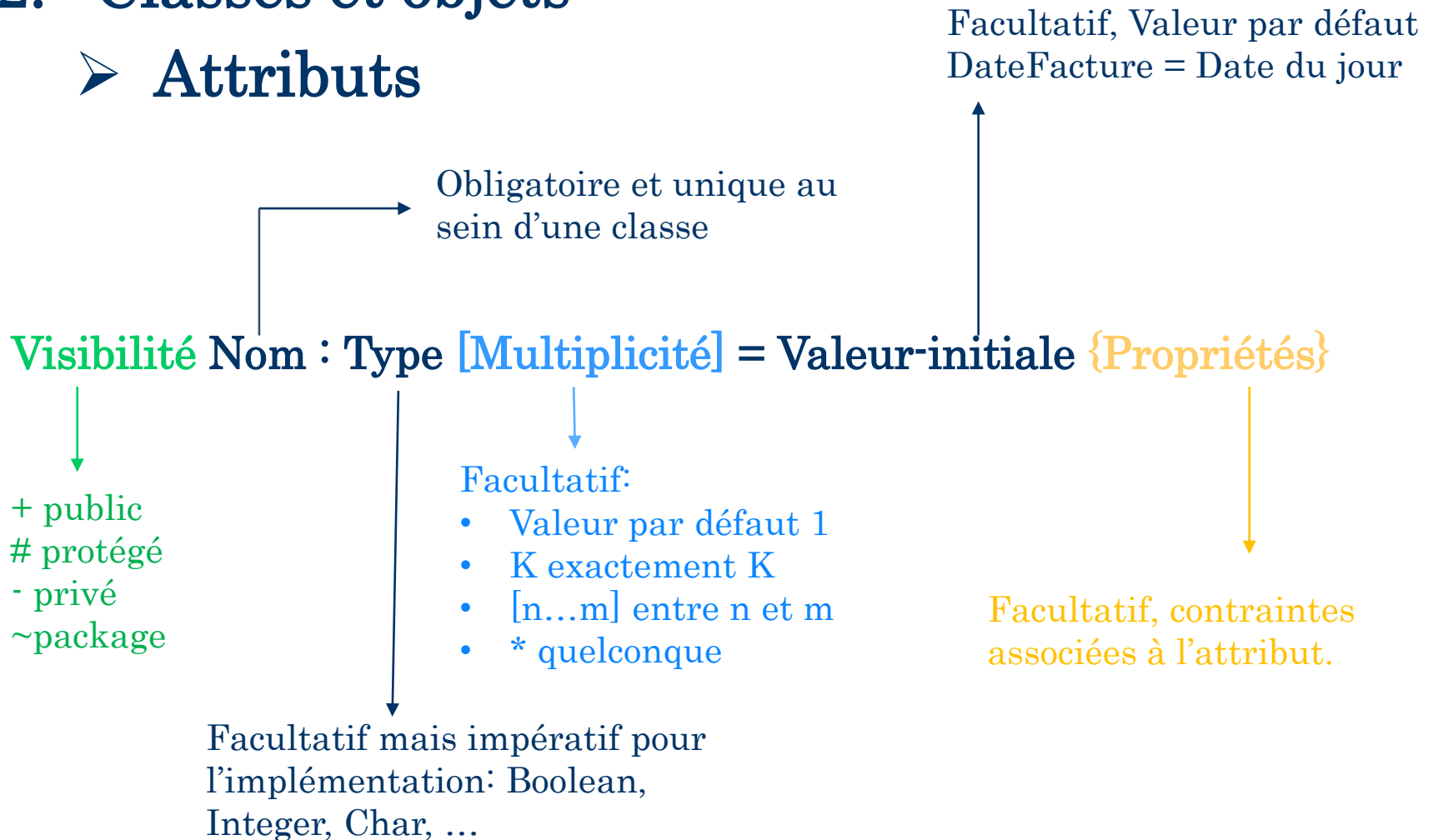
Visibilité Nom : Type [Multiplicité] = Valeur-initiale {Propriétés}

IV. UML

Diagramme de Classes/objets

2. Classes et objets:

➤ Attributs



IV. UML

Diagramme de Classes/objets

2. Classes et objets:

➤ Attributs identifiant d'une classe

Un attribut identifiant joue un rôle particulier, car il permet de repérer de façon **unique** chaque objet, instance de la classe.

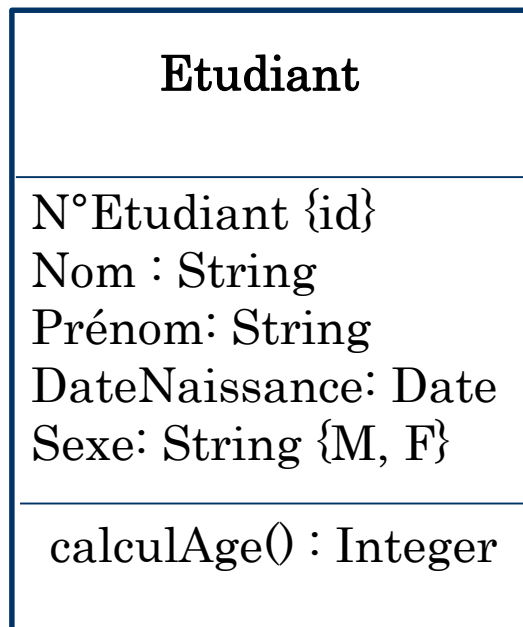
Entre l'identifiant et les autres attributs de la classe, il existe une DF.

IV. UML

Diagramme de Classes/objets

2. Classes et objets:

- **Exemple: Attributs identifiant d'une classe**



*Connaissant le N°Etudiant,
on doit pouvoir déduire son
nom, prénom, ...*

IV. UML

Diagramme de Classes/objets

2. Classes et objets:

➤ Attributs à valeur unique

On peut définir un attribut qui garde une valeur **unique** et partagée par toutes les instances de la classe.

Exemple :

*Valeur de la variable $PI = 3,14$ définit dans la classe *Math* du langage *Java*.*

La valeur PI reste la même quelque soit l'objet de cette classe.

IV. UML

Diagramme de Classes/objets

2. Classes et objets:

➤ Attribut dérivé

Un attribut « *dérivé* » est un attribut qui peut être **calculé** à partir d'autres attributs.

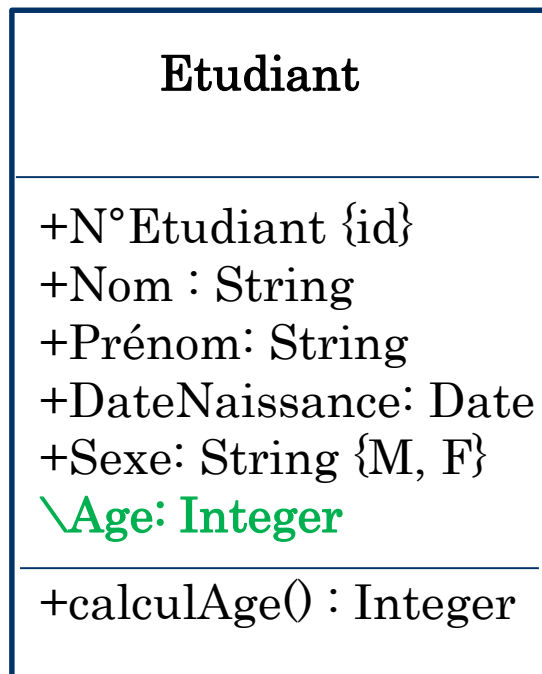
Syntaxe : l'ajout de \ devant son nom

IV. UML

Diagramme de Classes/objets

2. Classes et objets:

➤ Attribut dérivé



Age est utilisé comme un vrai attribut mais calculé par une méthode.

IV. UML

Diagramme de Classes/objets

2. Classes et objets:

- Définitions.
- Représentation.
- Encapsulation et visibilité.
- Attributs.
- Méthodes.

IV. UML

Diagramme de Classes/objets

2. Classes et objets:

- Méthode ou opération
- Le comportement de l'objet est modélisé par un ensemble de méthodes.
- Chaque méthode correspond à une implémentation bien précise d'un comportement.

IV. UML

Diagramme de Classes/objets

2. Classes et objets:

➤ Méthode ou opération

Une **méthode** (opération) permet de définir ce que tous les objets de la classe sont **capables** de réaliser.

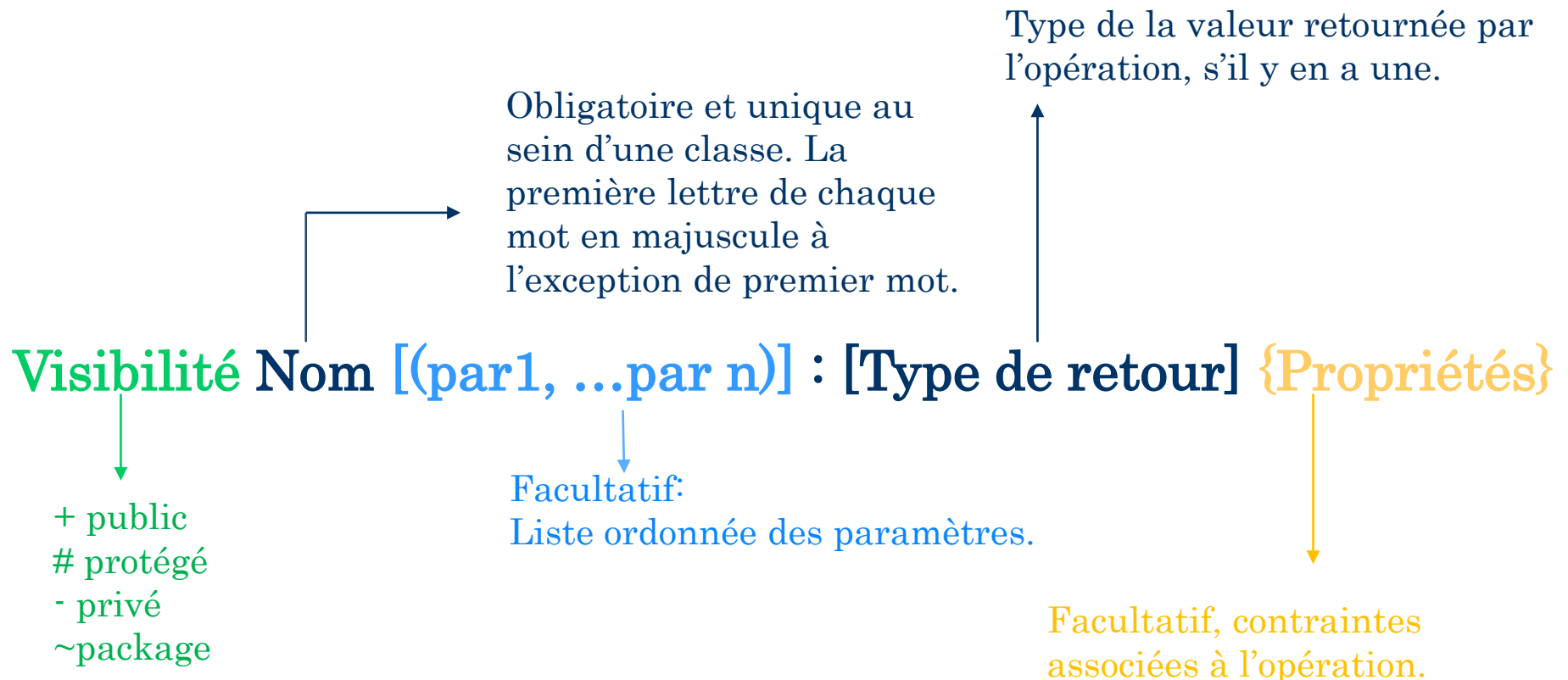
- Dans une classe, une opération doit-être **unique** (même nom et mêmes types de paramètres).
- Une opération est dite **surchargée**, si elle apparaît plusieurs fois avec le même nom mais des paramètres différents.
- Il est impossible que deux opérations ne se distinguent que par la valeur retournée.

IV. UML

Diagramme de Classes/objets

2. Classes et objets:

➤ Méthode

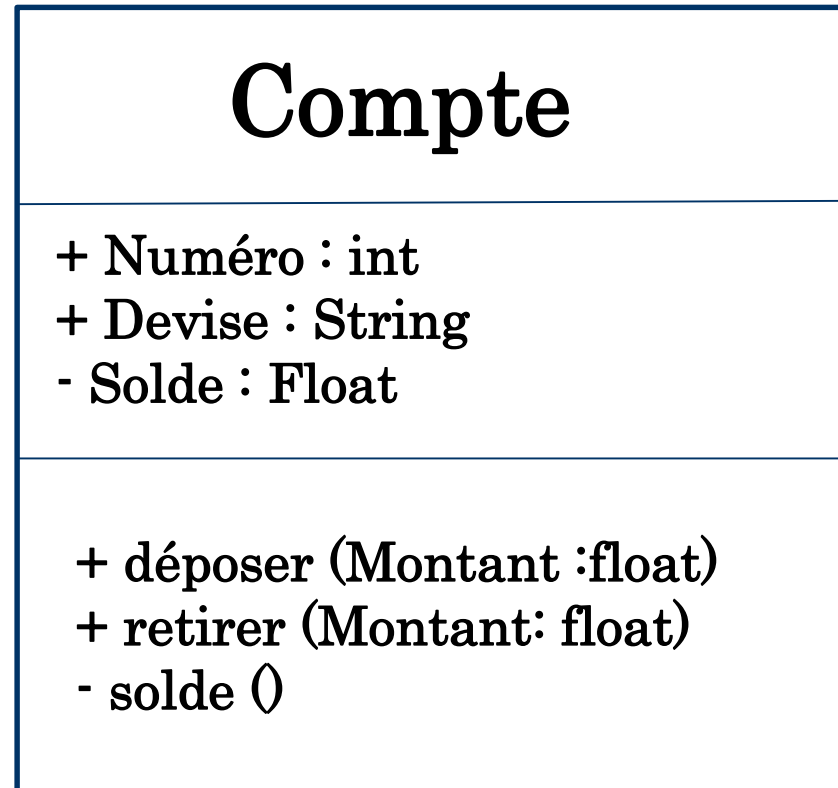


IV. UML

Diagramme de Classes/objets

2. Classes et objets:

➤ Exemple:

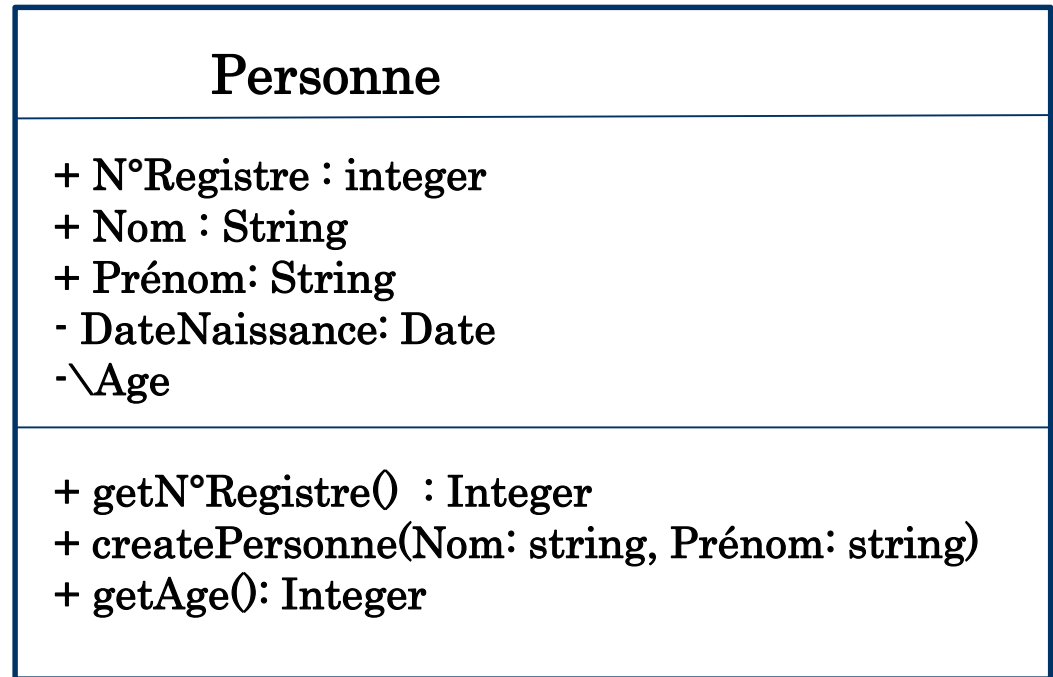


IV. UML

Diagramme de Classes/objets

2. Classes et objets:

➤ Exemple:



IV. UML

Diagramme de Classes/objets

Exercices: Proposez une modélisation, étape par étape, en faisant apparaître, en fonction des connaissances disponibles, les changements du modèle.

1. Une personne est caractérisée par son nom, son prénom, son sexe et son âge.
Les responsabilités de la classe sont entre autres le calcul de l'âge et le calcul des charges. Les attributs de la classe sont privés.
2. Deux types de revenus sont envisagés, le salaire et toutes les sources de revenus autres que le salaire, qui sont tous deux représentés par des entiers. On calcule les charges en appliquant un coefficient fixe de 15% sur les salaires et un coefficient de 20% sur les revenus.

IV. UML

Diagramme de Classes/objets

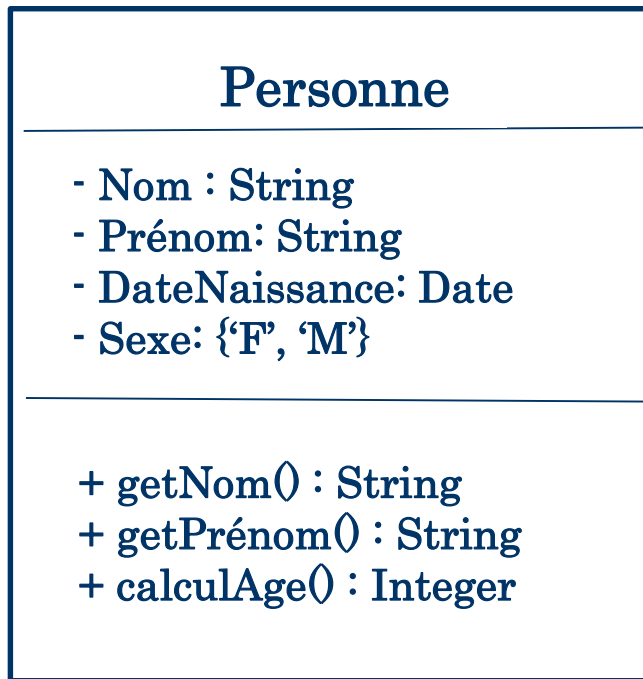
Exercices: Proposez une modélisation, étape par étape, en faisant apparaître, en fonction des connaissances disponibles, les changements du modèle.

1. Une personne est caractérisée par son nom, son prénom, son sexe et son âge.
Les responsabilités de la classe sont entre autres le calcul de l'âge et le calcul des charges. Les attributs de la classe sont privés.
2. Deux types de revenus sont envisagés, le salaire et toutes les sources de revenus autres que le salaire, qui sont tous deux représentés par des entiers. On calcule les charges en appliquant un coefficient fixe de 15% sur les salaires et un coefficient de 20% sur les revenus.

IV. UML

Diagramme de Classes/objets

Exercices:



IV. UML

Diagramme de Classes/objets

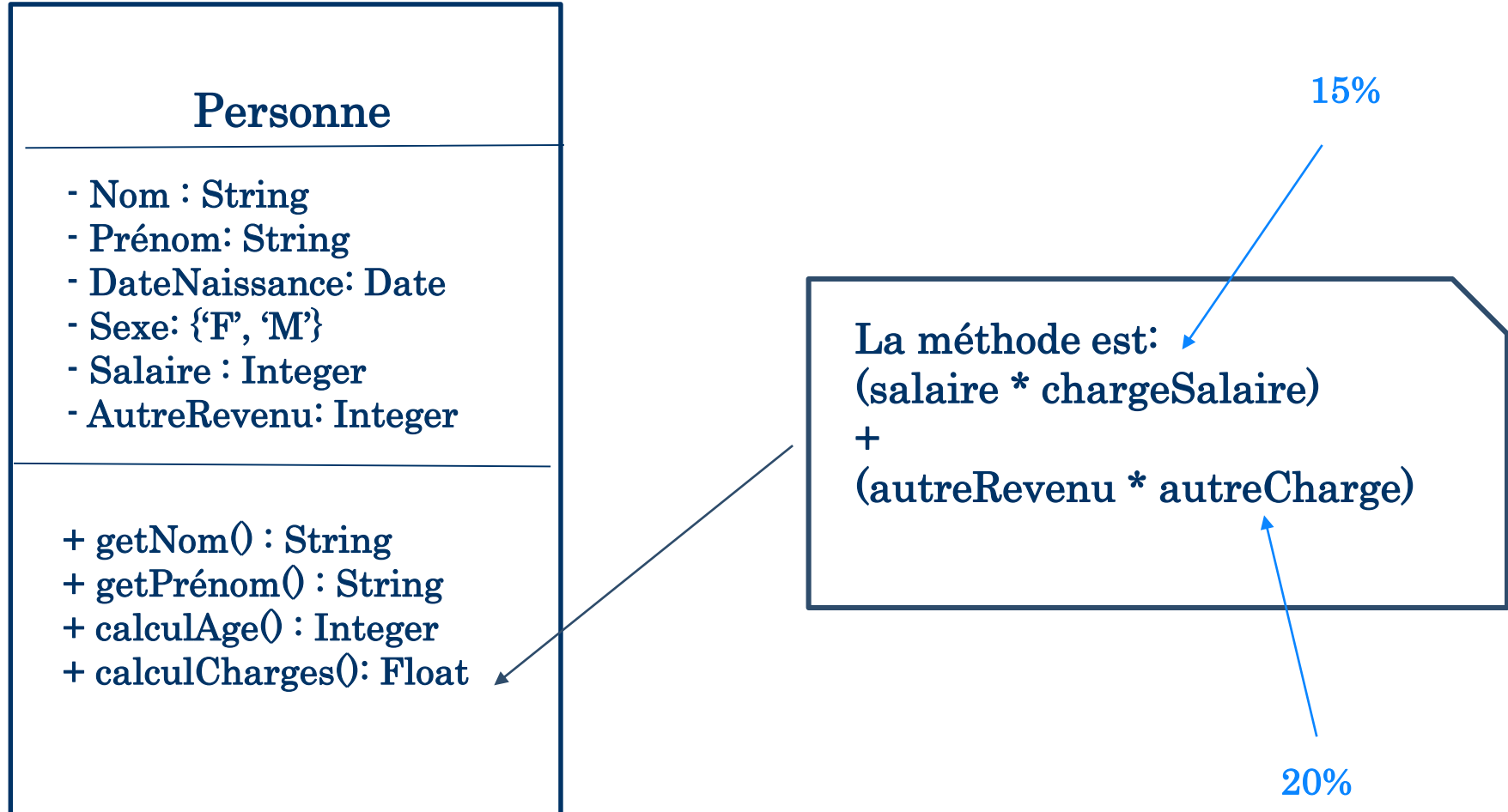
Exercices: Proposez une modélisation, étape par étape, en faisant apparaître, en fonction des connaissances disponibles, les changements du modèle.

1. Une personne est caractérisée par son nom, son prénom, son sexe et son âge.
Les responsabilités de la classe sont entre autres le calcul de l'âge et le calcul des charges. Les attributs de la classe sont privés.
2. Deux types de revenus sont envisagés, le salaire et toutes les sources de revenus autres que le salaire, qui sont tous deux représentés par des entiers. On calcule les charges en appliquant un coefficient fixe de 15% sur les salaires et un coefficient de 20% sur les revenus.

IV. UML

Diagramme de Classes/objets

Exercices:



IV. UML

Diagramme de Classes/objets

1. Objectif.
2. Classes et objets.
3. **Relations entre classes.**
4. Diagramme d'objets.
5. Construction d'un diagramme de classes.

IV. UML

Diagramme de Classes/objets

3. Relations entre classes.

1. Association.
2. Multiplicité.
3. Classe d'association.
4. Association qualifiée.
5. Relation d'agrégation/Composition.
6. Relation de dépendance.
7. Relation d'héritage.
8. Autres relations et contraintes.

IV. UML

Diagramme de Classes/objets

3. Relations entre classes.

1. Association.
2. Multiplicité.
3. Classe d'association.
4. Association qualifiée.
5. Relation d'agrégation/Composition.
6. Relation de dépendance.
7. Relation d'héritage.
8. Autres relations et contraintes.

IV. UML

Diagramme de Classes/objets

1. Association.

- Après la modélisation des classes, il convient alors de les **relier** entre elles.
- Une **association** est une relation structurelle, qui indique que les instances d'une classe sont **reliées** aux instances d'une autre classe.

IV. UML

Diagramme de Classes/objets

1. Association.

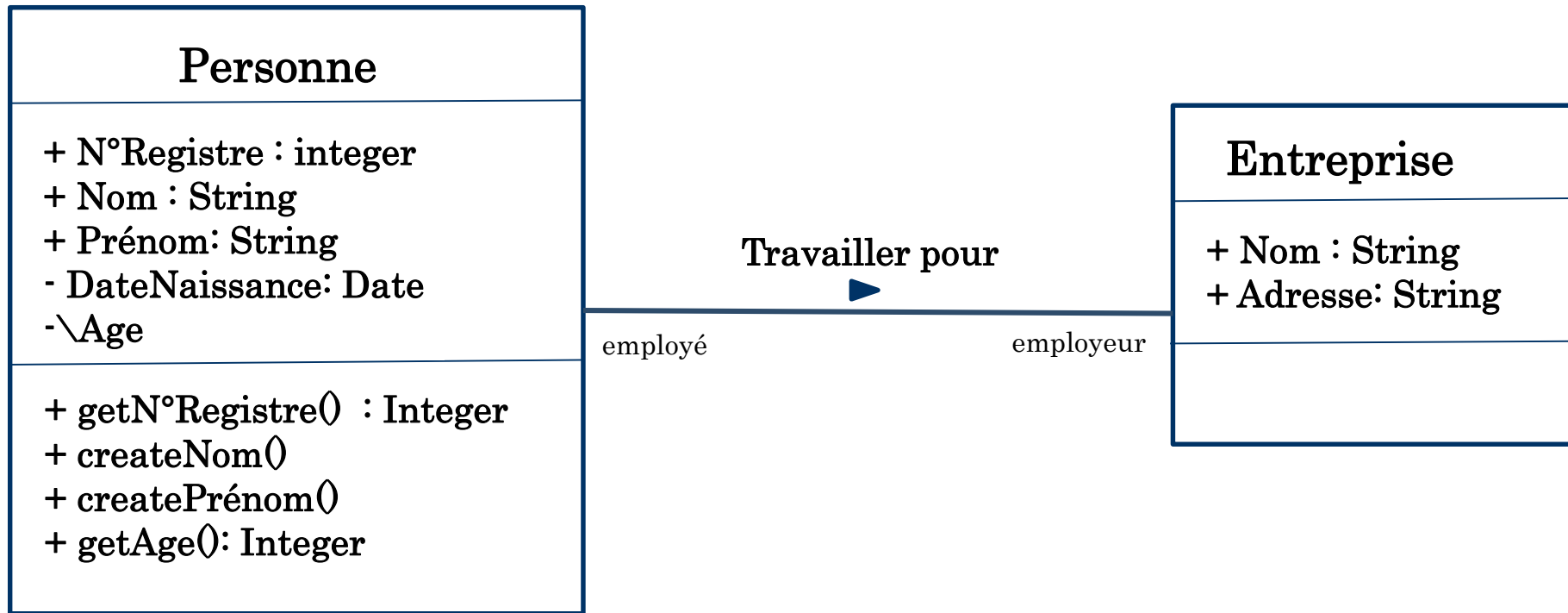
- Une association qui relie **deux** classes est dite **binaire**.
- Une association qui relie **plus de deux** classes est dite **n-aire**.

IV. UML

Diagramme de Classes/objets

1. Association.

➤ Exemple:

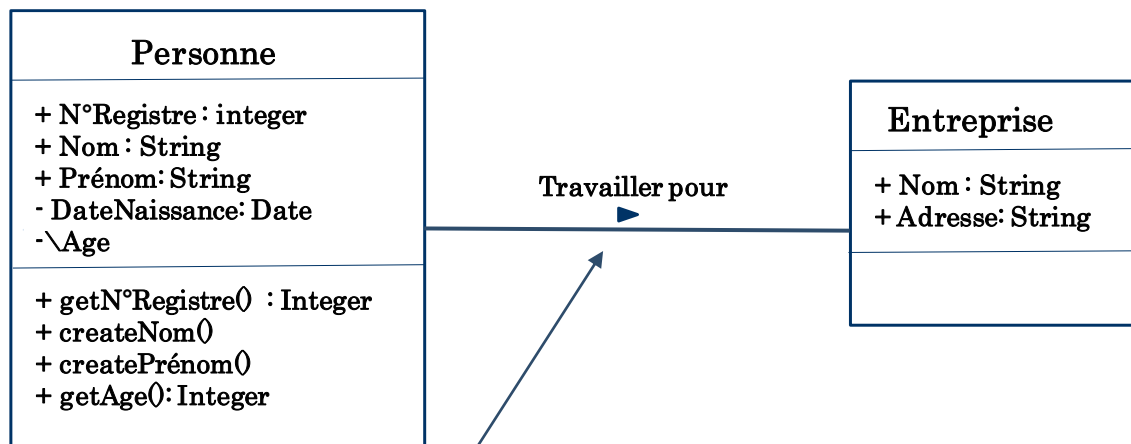


IV. UML

Diagramme de Classes/objets

1. Association.

➤ Exemple:



Une association binaire est:

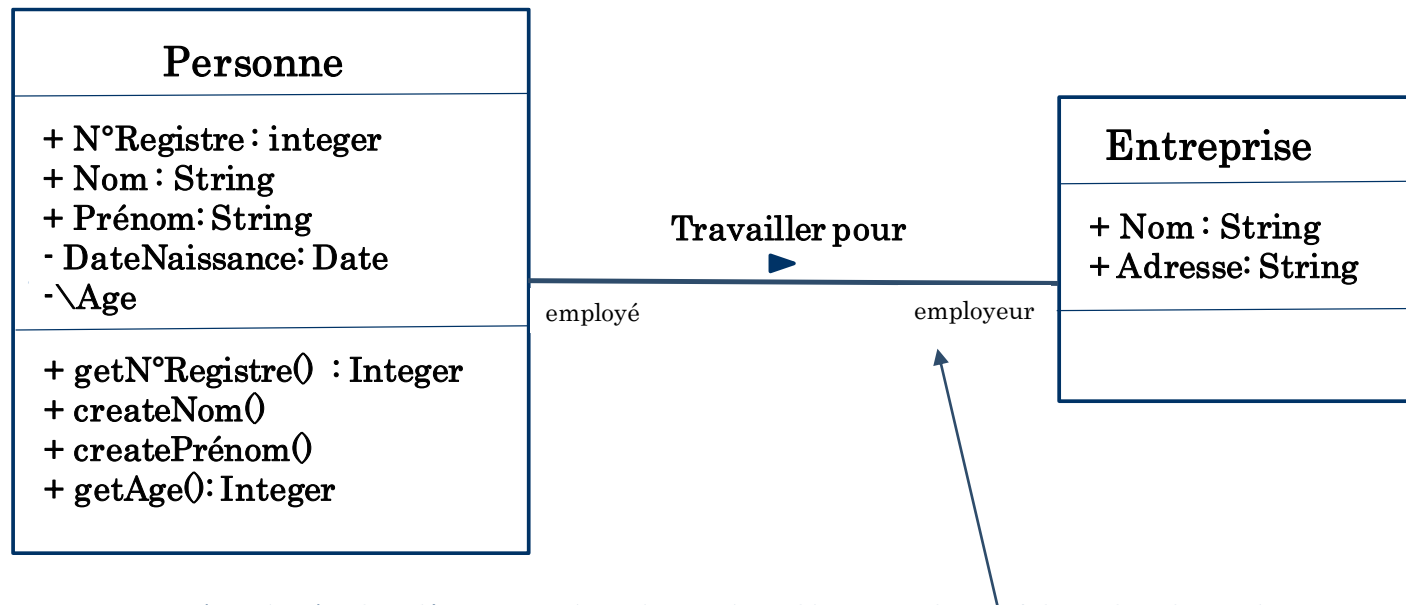
- Modélisée par un trait plein
- Peut avoir un nom (à l'actif ou passif selon le sens de lecture).
- Et un sens de lecture pour améliorer la lisibilité. (► ou ◄)

IV. UML

Diagramme de Classes/objets

1. Association.

➤ Exemple:



Chaque extrémité de l'association indique le rôle de la classe dans la relation.

IV. UML

Diagramme de Classes/objets

3. Relations entre classes.

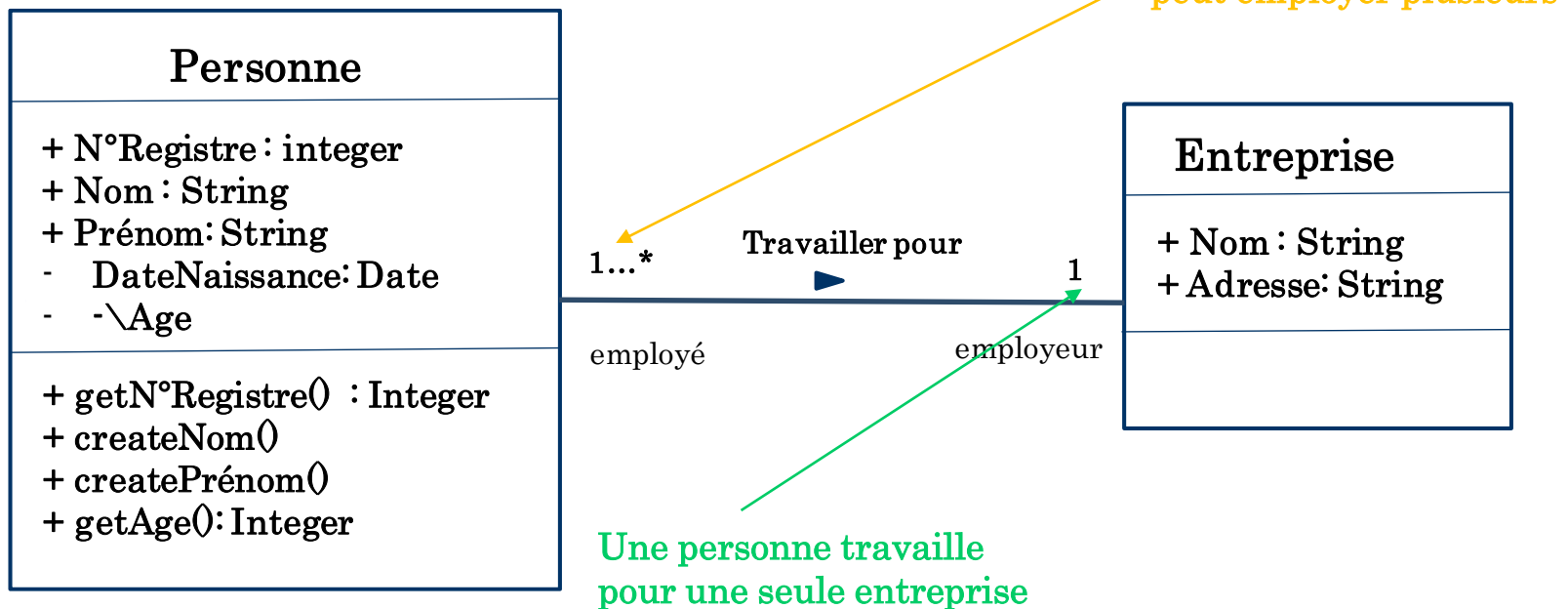
1. Association.
2. Multiplicité.
3. Classe d'association.
4. Association qualifiée.
5. Relation d'agrégation/Composition.
6. Relation de dépendance.
7. Relation d'héritage.
8. Autres relations et contraintes.

IV. UML

Diagramme de Classes/objets

2. Multiplicité.

Chaque extrémité de l'association précise le **nombre d'objets** de la classe qui interviennent dans la relation.



IV. UML

Diagramme de Classes/objets

2. Multiplicité.

Chaque **extrémité** d'une association A-B, a une **contrainte de multiplicité** qui exprime le nombre **minimum** et **maximum** d'objets d'une classe A pouvant être reliés à un objet de classe B.

Multiplicité	Signification
1...1 ou 1	Un et un seul
0...1	De 0 à 1
0...* ou *	De 0 à n
1...*	De 1 à n
n...m	De n à m (entiers naturels)

IV. UML

Diagramme de Classes/objets

3. Relations entre classes.

1. Association.
2. Multiplicité.
3. Classe d'association.
4. Association qualifiée.
5. Relation d'agrégation/Composition.
6. Relation de dépendance.
7. Relation d'héritage.
8. Autres relations et contraintes.

IV. UML

Diagramme de Classes/objets

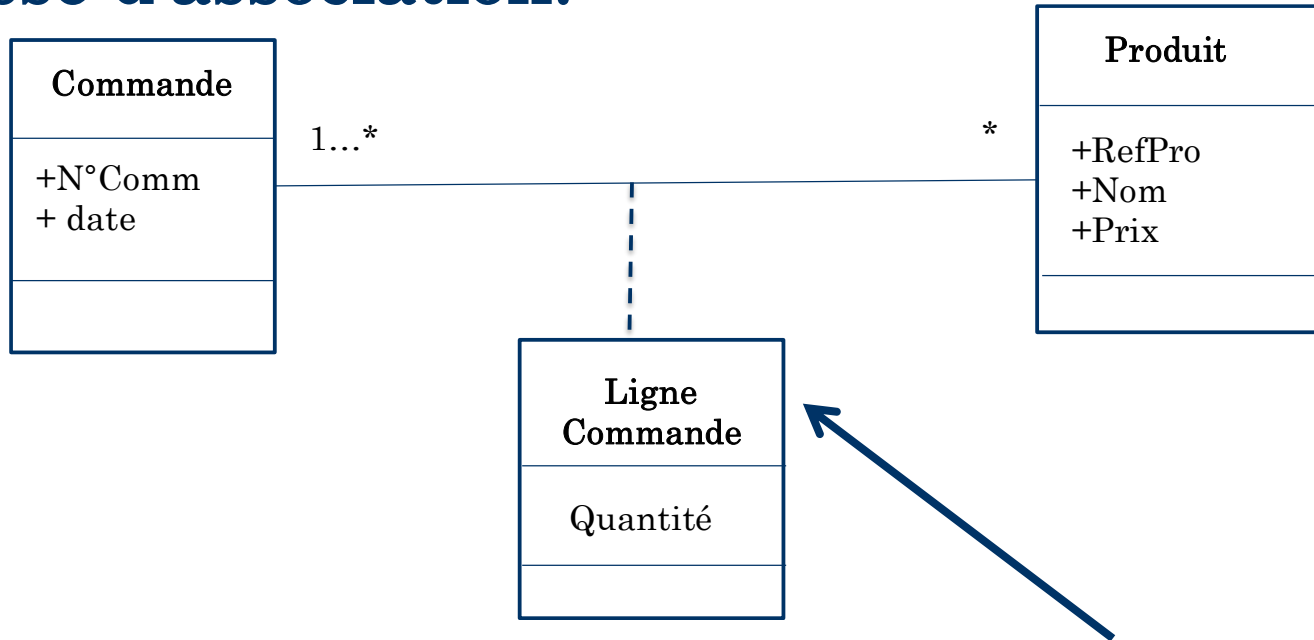
3. Classe d'association.

- Une association peut avoir **ses propres propriétés**, qui ne sont disponibles dans aucune des classes qu'elle lie.
- Dans le modèle objet, **seules les classes** peuvent avoir des **propriétés**. Cette association devient alors une classe appelée « **Classe-association** ».
- Une **classe-association** sera alors définie et considérée comme toutes les autres classes du modèle.

IV. UML

Diagramme de Classes/objets

3. Classe d'association.



Classe d'association

Représentée par un trait discontinu entre la classe et l'association.

IV. UML

Diagramme de Classes/objets

3. Relations entre classes.

1. Association.
2. Multiplicité.
3. Classe d'association.
4. Association qualifiée.
5. Relation d'agrégation/Composition.
6. Relation de dépendance.
7. Relation d'héritage.
8. Autres relations et contraintes.

IV. UML

Diagramme de Classes/objets

4. Classe qualifiée.

- Parfois l'association entre deux classes donne peu d'indications sur l'implication des classes dans la relation.
=> **modélisation imprécise** et **multiplicité indéterminée**.
- Il est intéressant de **restreindre** la portée de l'association à quelques **instances ciblées** de la classe cible. Ces attributs permettant de cibler un sous-ensemble d'instances sont appelés **qualificatifs**.
- Une association **qualifiée** va permettre dans certains cas, de **transformer** une multiplicité **indéterminée** en une multiplicité **finie**.

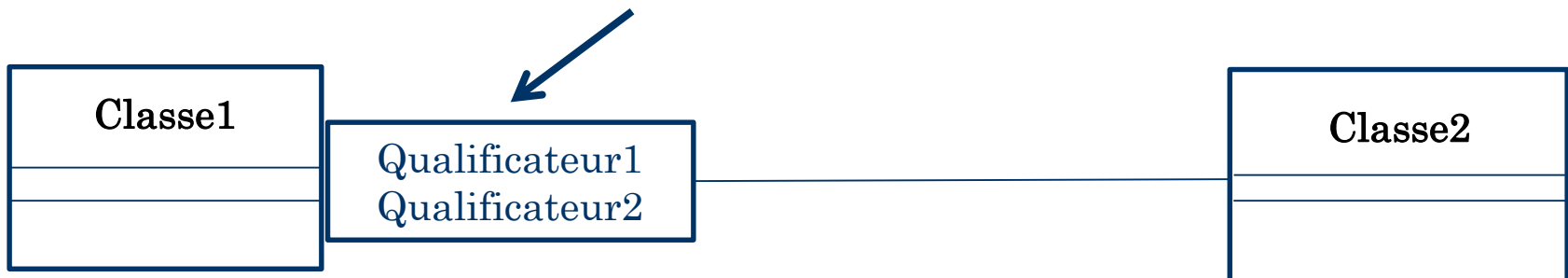
IV. UML

Diagramme de Classes/objets

4. Classe qualifiée.

➤ Représentation:

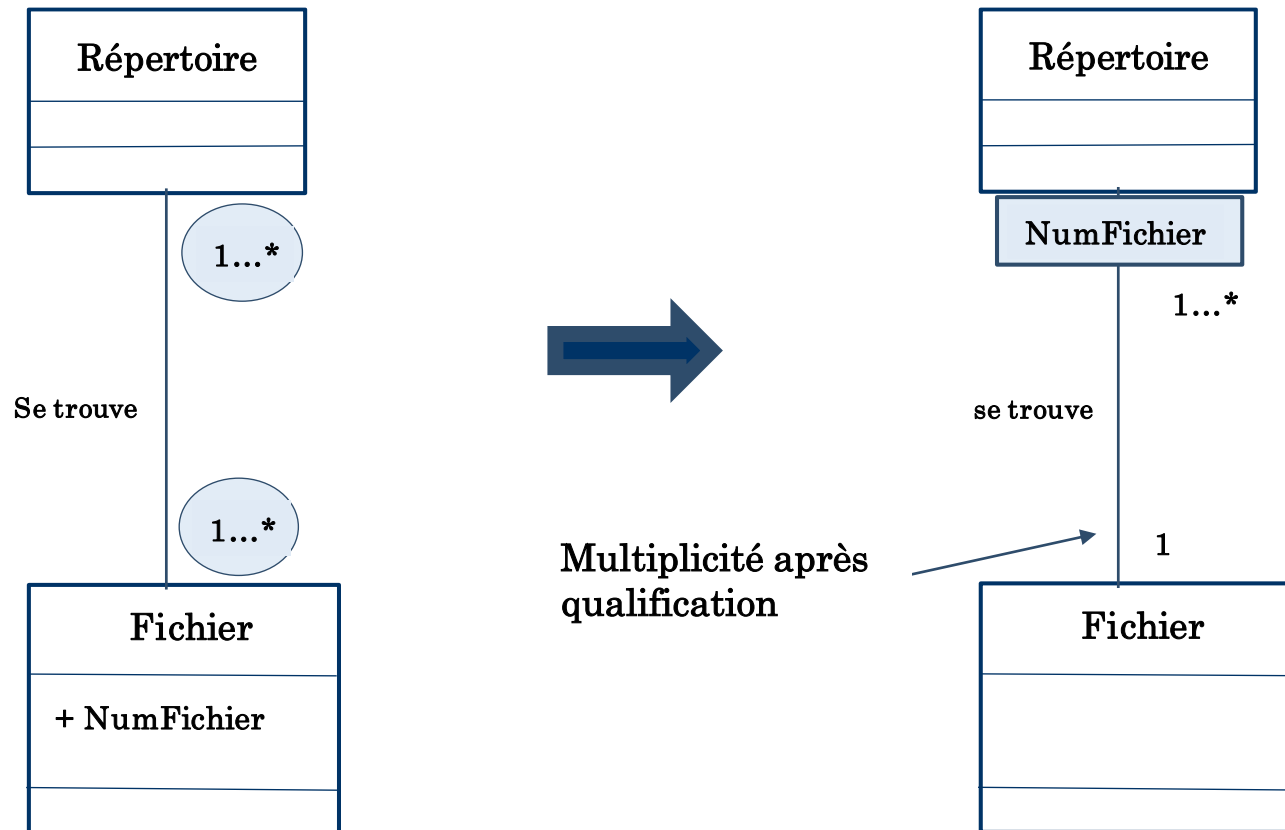
Représentée par un rectangle
contenant le qualificatif



IV. UML

Diagramme de Classes/objets

Exemple:

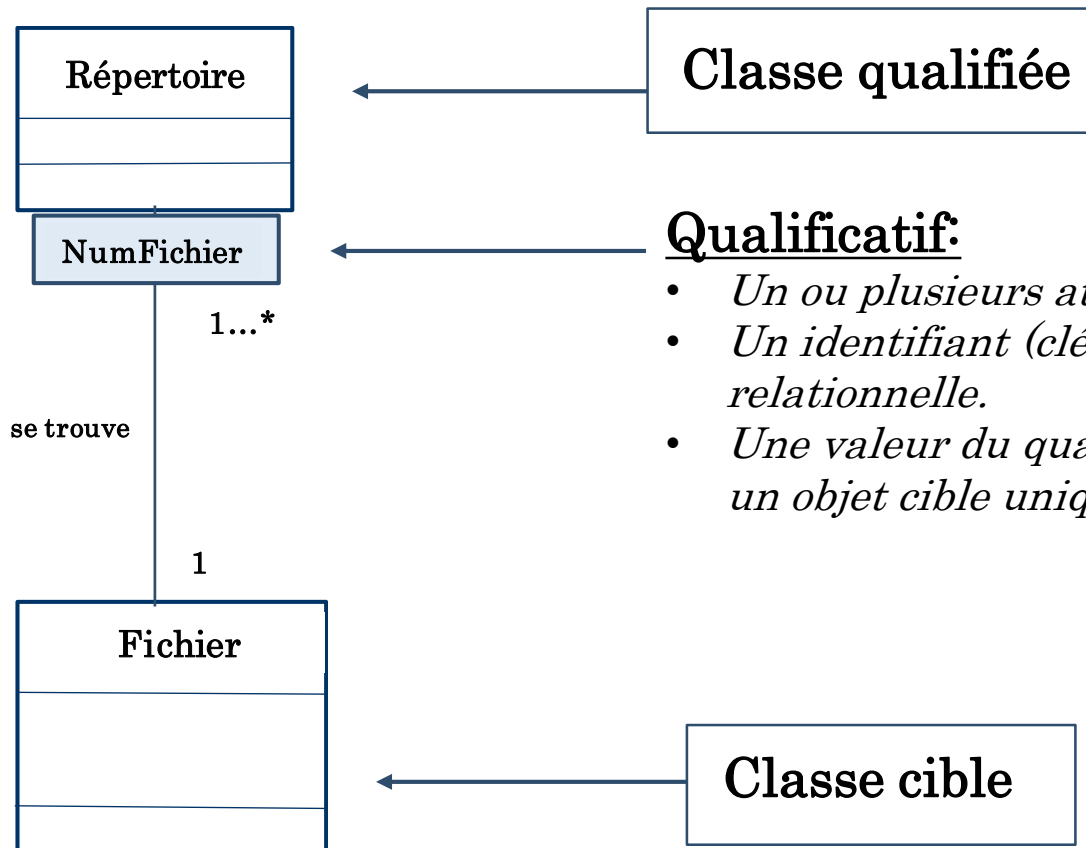


But: Restreindre la portée de l'association à quelques instances ciblées en fonction d'un ou plusieurs attributs.

IV. UML

Diagramme de Classes/objets

Exemple:



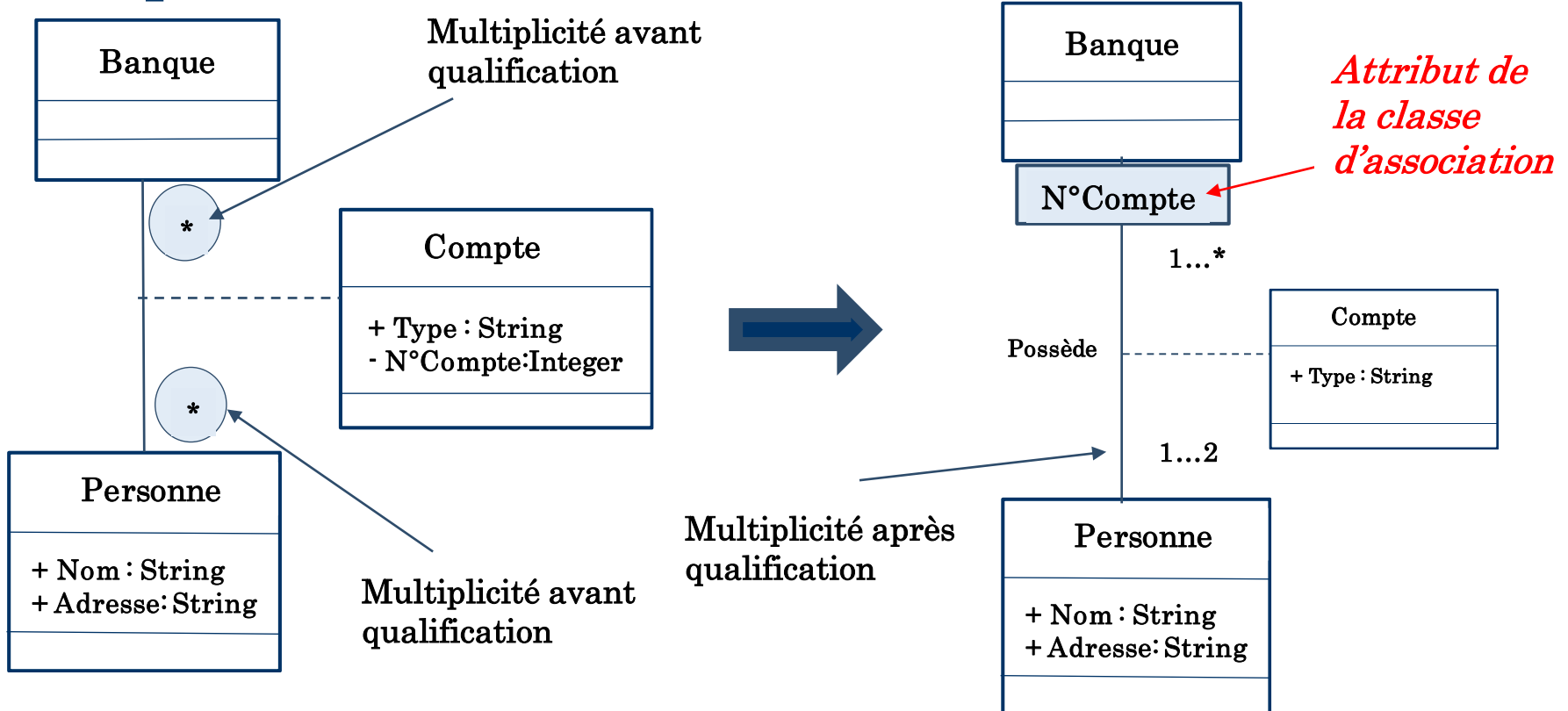
Qualificatif:

- *Un ou plusieurs attributs de la classe cible.*
- *Un identifiant (clé primaire) dans une analyse relationnelle.*
- *Une valeur du qualificatif permet de désigner un objet cible unique*

IV. UML

Diagramme de Classes/objets

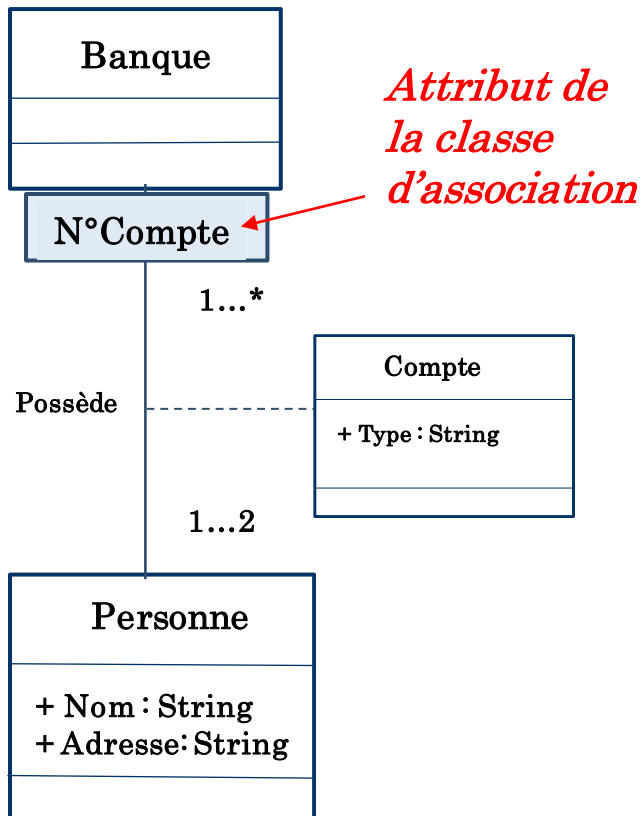
Exemple:



IV. UML

Diagramme de Classes/objets

Exemple:



Lecture du diagramme:

- *Un compte dans une banque appartient à au plus deux personnes. Une instance du couple {Banque, numéro compte} est en association avec une à deux instances de la classe personne.*
- *Mais une personne peut posséder plusieurs comptes dans plusieurs banques. Une instance de la classe personne peut être associé à plusieurs instances du couple {Banque, numéro de compte}*

IV. UML

Diagramme de Classes/objets

Remarques:

- *Il ne faut pas utiliser de qualificatif dans une association **n-aire** ($n > 2$).*
- *Un qualificatif agit toujours sur une association dont la **multiplicité est plusieurs** du côté cible et devient faible une fois l'association qualifiée.*
- *En considérant un objet qualifié, une valeur du qualificatif doit permettre de désigner un objet **cible unique** ou un très petit ensemble d'objets cibles.*

IV. UML

Diagramme de Classes/objets

3. Relations entre classes.

1. Association.
2. Multiplicité.
3. Classe d'association.
4. Association qualifiée.
5. **Relation d'agrégation/Composition.**
6. Relation de dépendance.
7. Relation d'héritage.
8. Autres relations et contraintes.

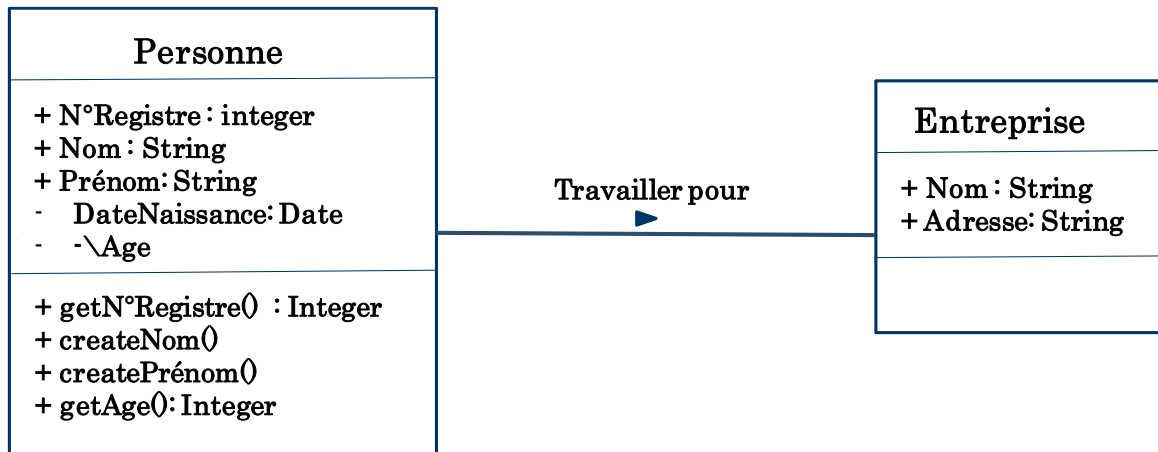
IV. UML

Diagramme de Classes/objets

5. Relation d'agrégation/Composition.

Une association simple entre classes représente une relation structurelle entre deux classes de **même niveau** conceptuel.

Aucune des deux n'est plus importante que l'autre.



IV. UML

Diagramme de Classes/objets

5. Relation d'agrégation.

Pour modéliser une relation *tout/partie* où une classe constitue un élément plus grand (*le tout*) composé d'éléments plus petits (*les parties*).



Utiliser une Agrégation

IV. UML

Diagramme de Classes/objets

5. Relation d'agrégation.

- L'**agrégation** est une forme particulière de l'association, elle représente la **relation d'inclusion** d'un élément dans un ensemble.
- On représente une **agrégation** par l'ajout d'un losange vide (◊) du côté de l'agregat.

IV. UML

Diagramme de Classes/objets

Agrégation: Exemple



- Cette association particulière (**agrégation**) permet de marquer une **étroite dépendance** entre les instances.
- Une instance (l'**agrégat**) est **composée** de plusieurs objets (**agrégés**).

IV. UML

Diagramme de Classes/objets

Caractéristiques d'une agrégation:

- **Antisymétrique:**

Si un objet A fait partie d'un objet B, alors l'objet B ne fait pas partie de l'objet A.

- **Transitive:**

Si un objet A fait partie d'un objet B, et que B fait partie d'un objet C, alors A doit faire partie de l'objet C.

IV. UML

Diagramme de Classes/objets

5. Relation d'agrégation/Composition.

Une **composition** est une relation d'agrégation particulière.

Une **composition** = Agrégation plus forte



Utiliser une Composition

IV. UML

Diagramme de Classes/objets

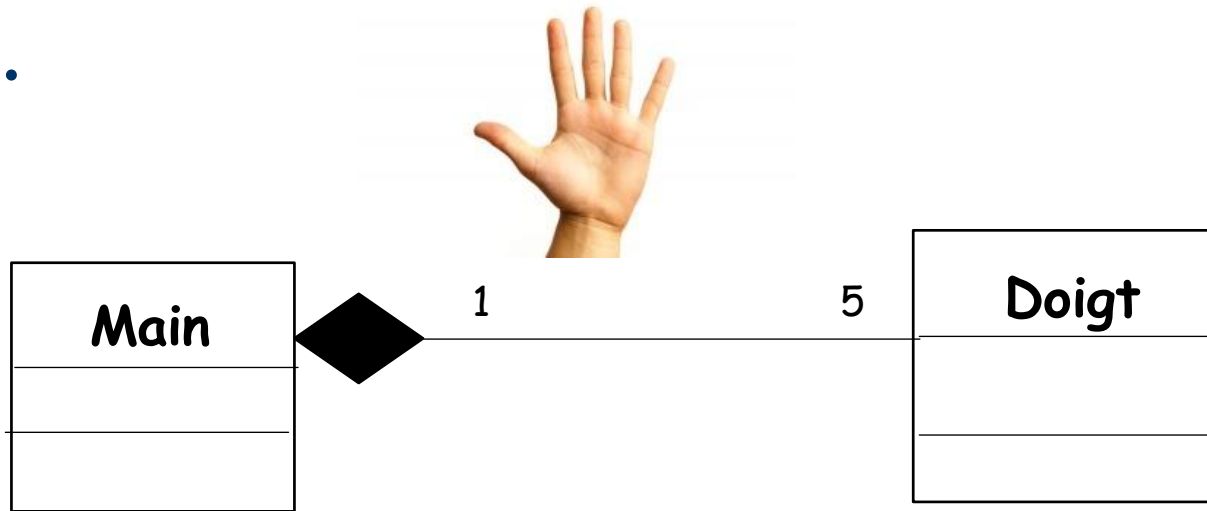
5. Relation de Composition.

- La **composition** est une forme particulière d'agrégation, dans laquelle la **vie des composants** est liée à celle de l'agrégat.
- On représente une **composition** par l'ajout d'un losange plein (◆) du côté de l'agrégat.

IV. UML

Diagramme de Classes/objets

Exemple.



- Dans une **composition** l'agrégat ne peut être multiple.
Une **instance** de la *partie* appartient toujours à au plus une instance de *l'élément composite*.
- La **destruction** de l'objet **composite** indique la **destruction** de ses composants.

IV. UML

Diagramme de Classes/objets

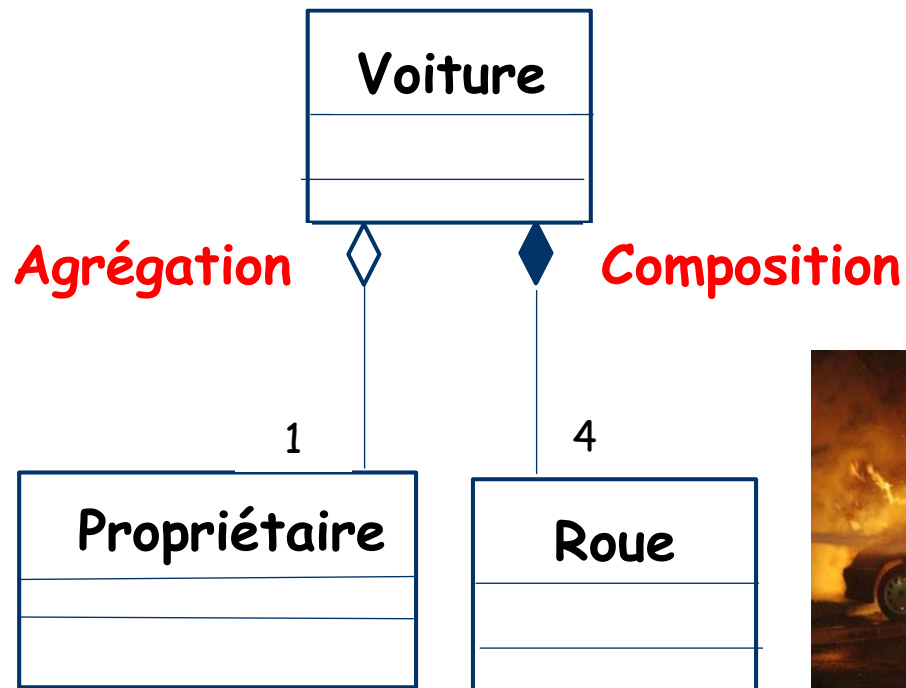
Les principales différences entre agrégation et composition:

Base de comparaison	Agrégation	Composition
De base	En agrégation, il existe une relation dans laquelle un enfant peut exister indépendamment du parent.	En composition, un enfant ne peut exister indépendamment du parent.
Type de relation	« a un »	« partie de »
Type d'association	Faible	Forte
Une fonction	La suppression de l'assemblage n'affecte pas ses pièces.	Si l'objet de la classe propriétaire est supprimé, cela affecte l'objet de la classe contenant.

IV. UML

Diagramme de Classes/objets

Agrégation ou composition:



IV. UML

Diagramme de Classes/objets

3. Relations entre classes.

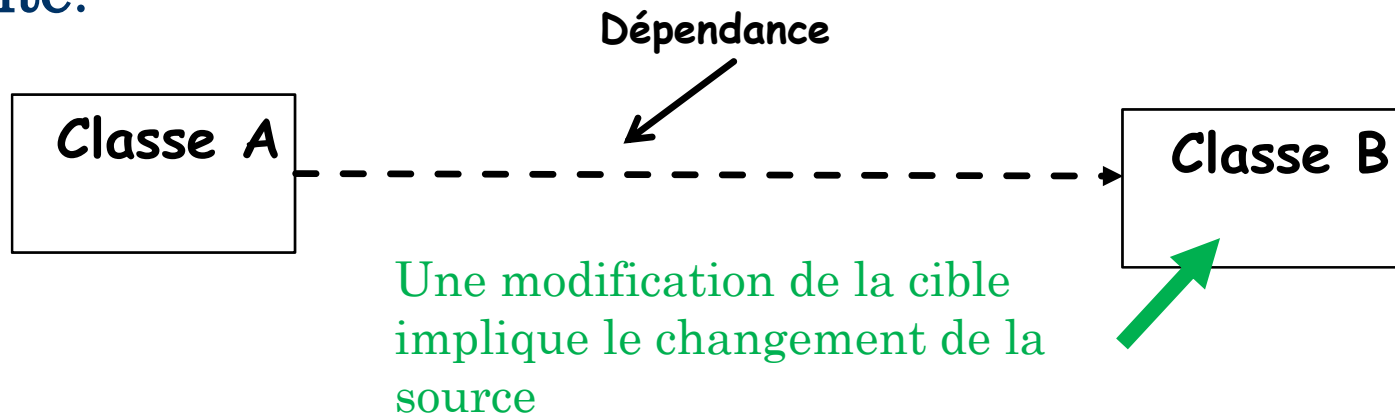
1. Association.
2. Multiplicité.
3. Classe d'association.
4. Association qualifiée.
5. Relation d'agrégation/Composition.
6. Relation de dépendance.
7. Relation d'héritage.
8. Autres relations et contraintes.

IV. UML

Diagramme de Classes/objets

6. Relation de dépendance.

- Une dépendance est une relation **unidirectionnelle** exprimant une dépendance sémantique entre les éléments du modèle.
- C'est la forme la **plus faible** de relation entre classes.
- On représente une dépendance par un **trait discontinu orienté**.



IV. UML

Diagramme de Classes/objets

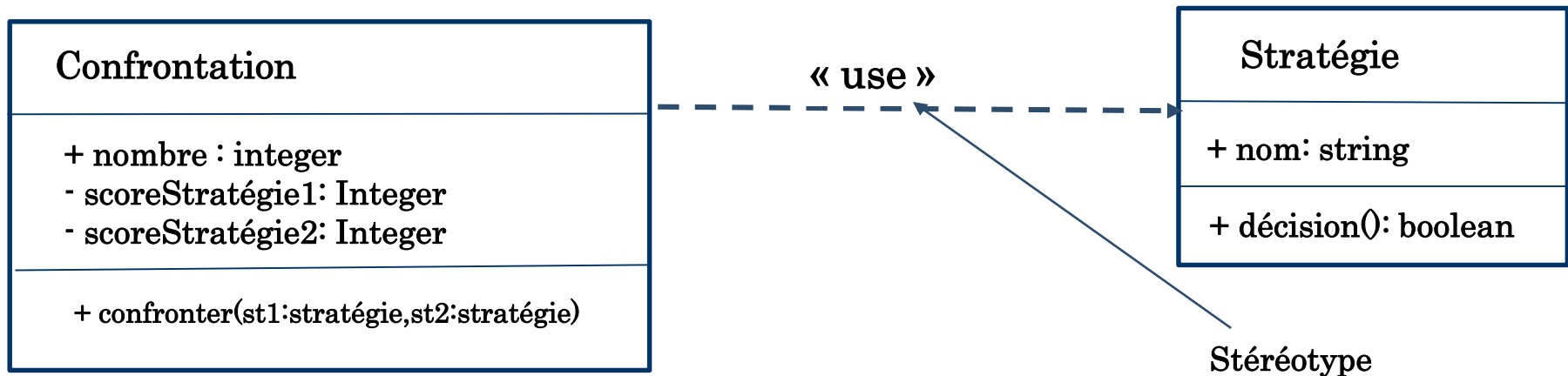
6. Relation de dépendance.

- Une **dépendance** s'interprète comme une relation de type « use ».
- Elle est utilisée :
Quand une **classe utilise un objet** d'une autre classe comme argument dans la signature d'une méthode.
Ou
Lorsque l'objet de l'autre classe est **créé à l'intérieur** de la méthode.

IV. UML

Diagramme de Classes/objets

Exemple:



La classe *Confrontation* utilise la classe *Stratégie*.

La classe *confrontation* possède une opération « *confronter* » dont deux paramètres sont du type *stratégie*.

IV. UML

Diagramme de Classes/objets

3. Relations entre classes.

1. Association.
2. Multiplicité.
3. Classe d'association.
4. Association qualifiée.
5. Relation d'agrégation/Composition.
6. Relation de dépendance.
7. Relation d'héritage.
8. Autres relations et contraintes.

IV. UML

Diagramme de Classes/objets

7. Relation d'héritage:

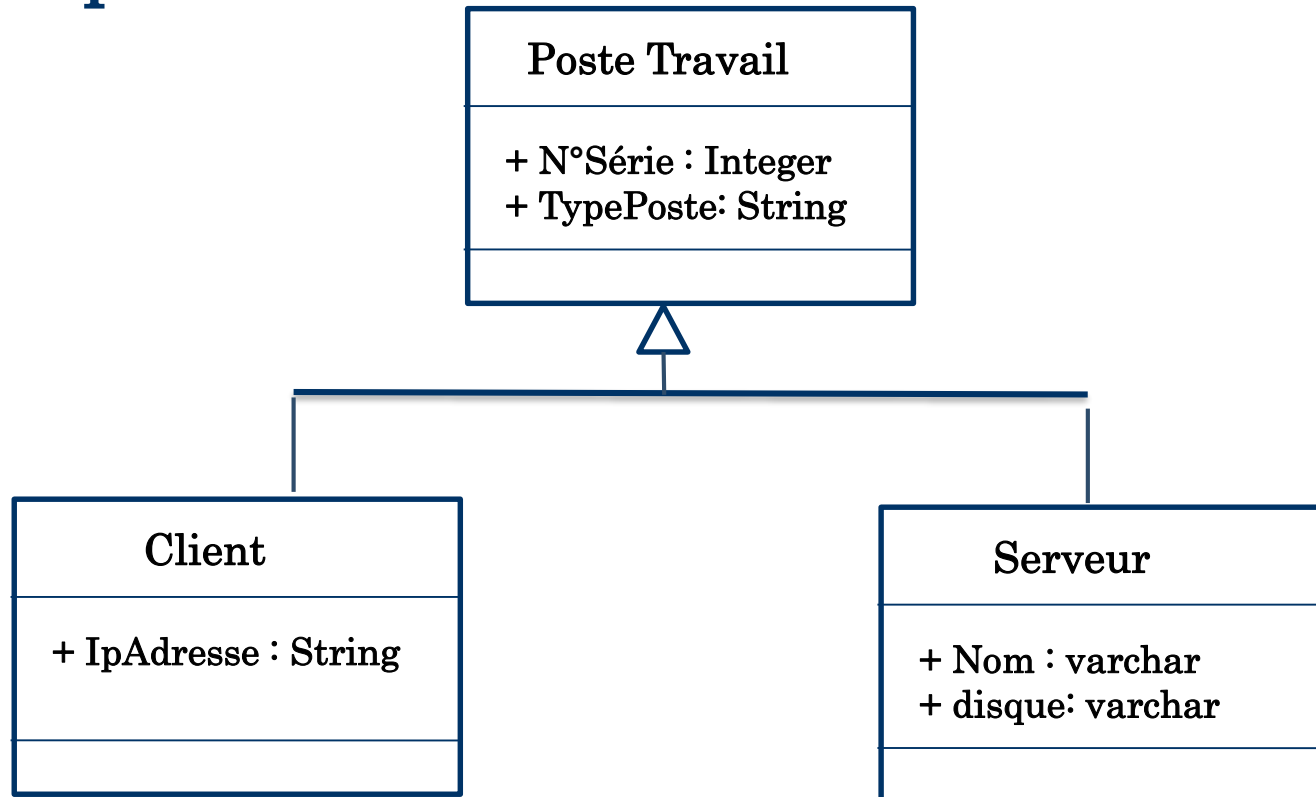
- L'héritage est une relation de **généralisation/spécialisation**.
- Les éléments spécialisés **héritent** de la **structure** et du **comportement** des éléments plus généraux.

Voir chapitre III: 3. Héritage.

IV. UML

Diagramme de Classes/objets

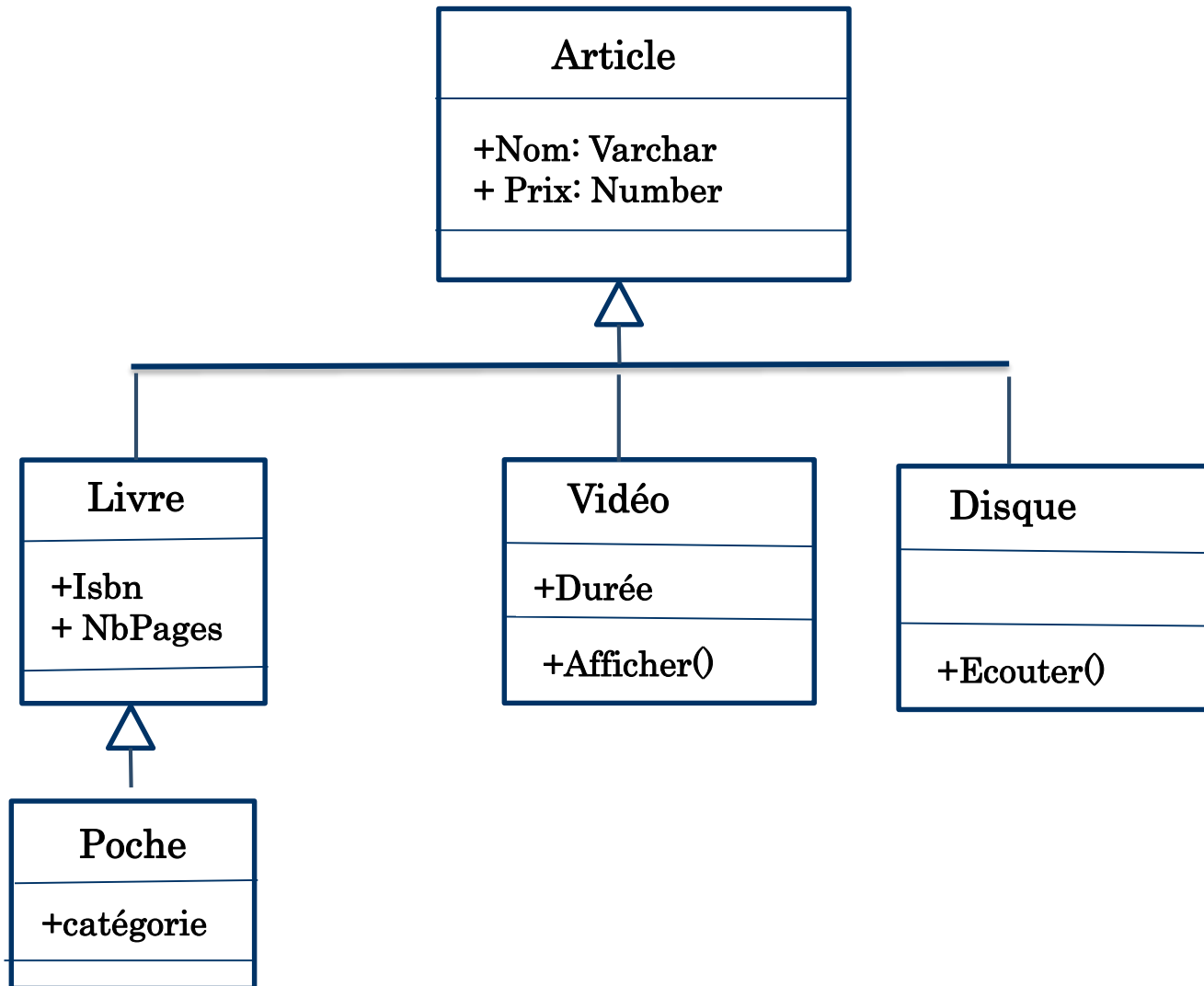
Exemples:



IV. UML

Diagramme de Classes/objets

Exemples:



IV. UML

Diagramme de Classes/objets

3. Relations entre classes.

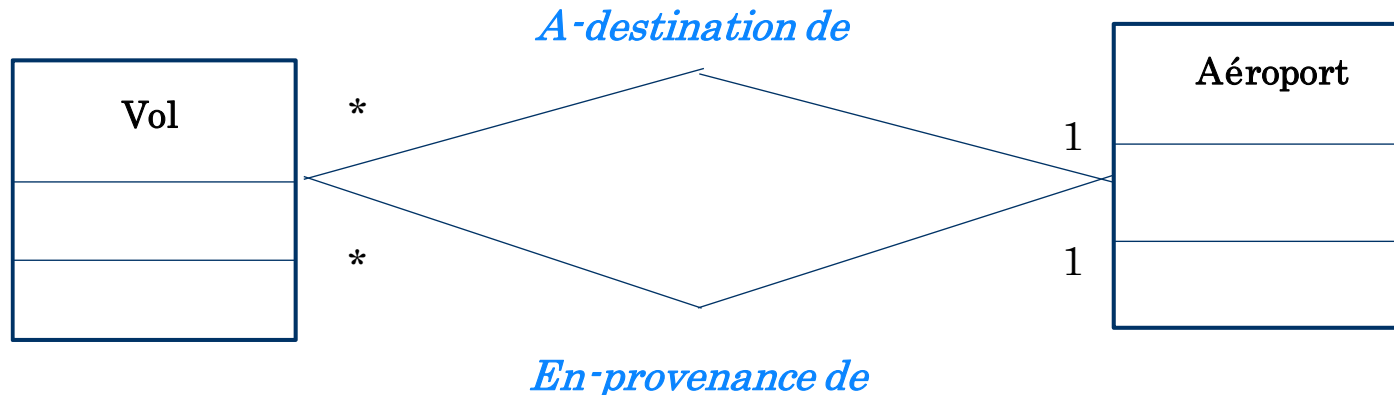
1. Association.
2. Multiplicité.
3. Classe d'association.
4. Association qualifiée.
5. Relation d'agrégation/Composition.
6. Relation de dépendance.
7. Relation d'héritage.
8. Autres relations et contraintes.

IV. UML

Diagramme de Classes/objets

8. Autres associations et contraintes:

Plusieurs associations entre classes:



Chacune des associations a une sémantique propre.

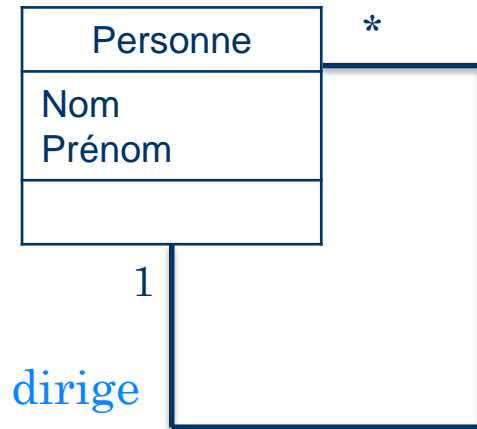
- **Aéroport** (provenance ou destination) concerne **plusieurs vols**.
- Un **vol** concerne **un seul** aéroport de **destination** et un seul aéroport de **provenance**.

IV. UML

Diagramme de Classes/objets

8. Autres associations et contraintes:

Association récursive:



Les rôles sont
obligatoires.

est dirigé

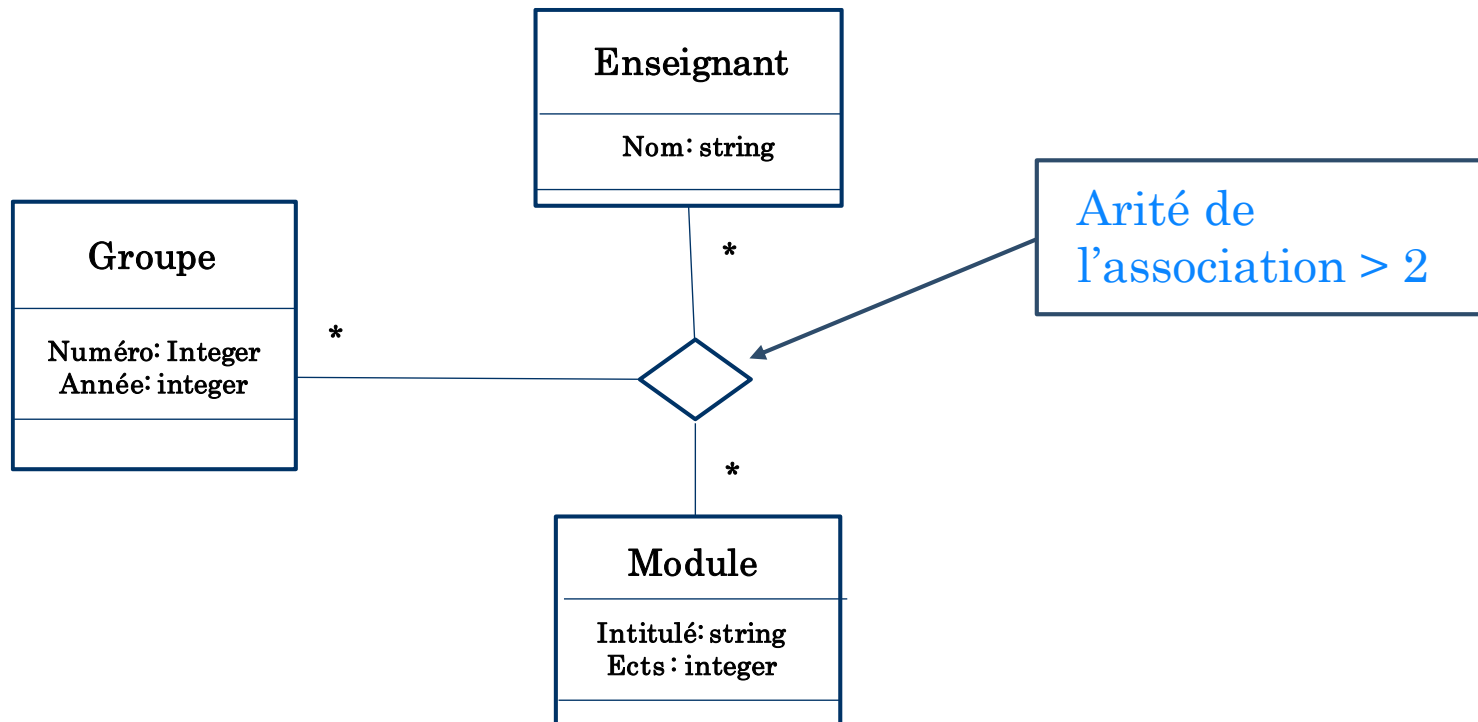
- Quand les **extrémités** de l'association pointent vers la **même classe**.
- Une association **réflexive (récursive)** indique qu'une **instance** de la classe impliquée peut être **reliée à elle-même** ou à d'autres **instances** de la même classe.

IV. UML

Diagramme de Classes/objets

8. Autres associations et contraintes:

Association n-aire ($n > 2$):



IV. UML

Diagramme de Classes/objets

8. Autres associations et contraintes:

Contraintes:

Les diagrammes UML seuls sont souvent insuffisants.
Il est souvent nécessaire d'ajouter **des contraintes** pour compléter les diagrammes.

Les contraintes se divisent en deux parties:

- Sous forme de **texte** entre accolades {...}
- En **OCL** (Object Constraint Language).
Langage informel avec sémantique précise.
Il est la contribution d'IBM à la norme UML.

IV. UML

Diagramme de Classes/objets

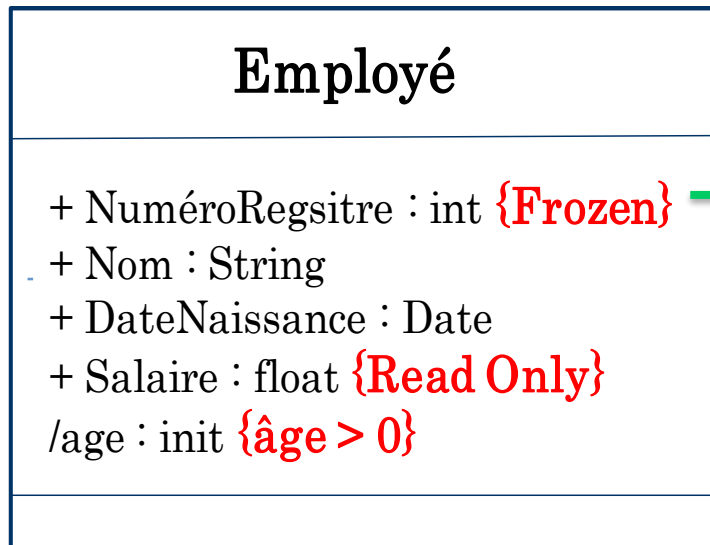
Contraintes:

1. Contraintes sur les attributs.
2. Contraintes sur les relations.

IV. UML

Diagramme de Classes/objets

1. Contraintes sur les attributs:



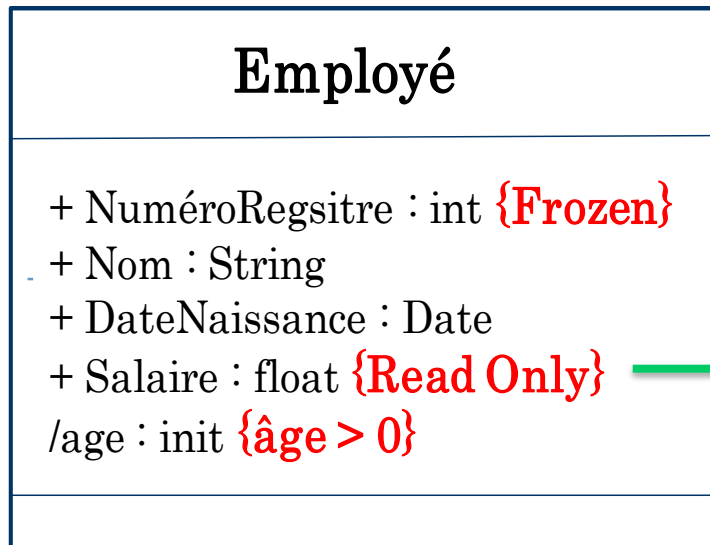
La contrainte **{frozen}** signifie que l'attribut une fois créé **ne changera jamais**.

« NuméroRegistre » de l'employé ne peut plus changer une fois créé.

IV. UML

Diagramme de Classes/objets

1. Contraintes sur les attributs:



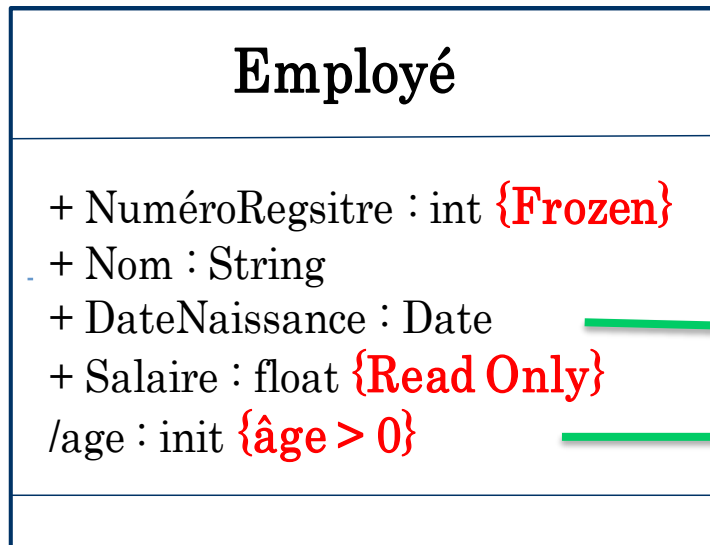
La contrainte **{Read Only}**, signifie que la valeur peut **changer** à l'intérieur de la classe mais pas à l'extérieur.

« Salaire » le salaire ne peut pas être modifié à l'extérieur de la classe
« Employé »

IV. UML

Diagramme de Classes/objets

1. Contraintes sur les attributs:



→ {Âge = diff (DateNaissance, Today)}

→ /Âge est calculé = attribut dérivé.

{Âge > 0} : âge doit être positif

IV. UML

Diagramme de Classes/objets

2. Contraintes sur les relations:

Les **contraintes** sur les relations sont des expressions qui précisent **le rôle** ou la portée d'un élément de modélisation.

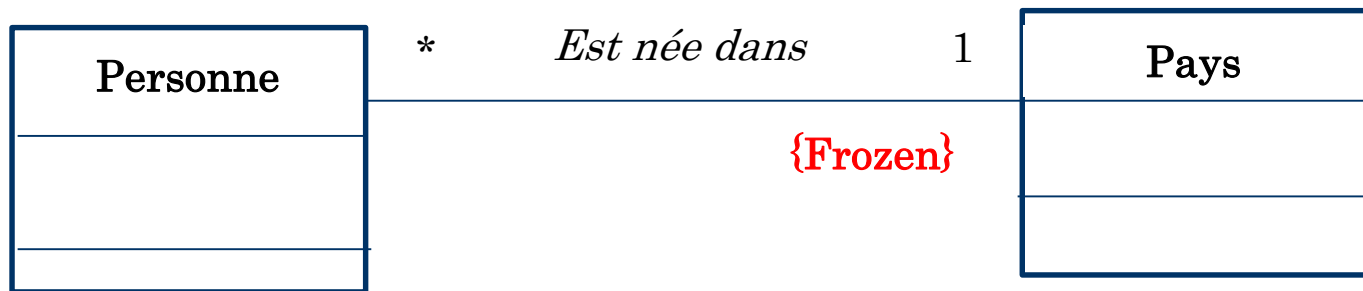
Elles permettent d'étendre ou **de préciser la sémantique** de l'association (restreindre le nombre d'instances visées).

IV. UML

Diagramme de Classes/objets

2. Contraintes sur les relations:

{Frozen}:



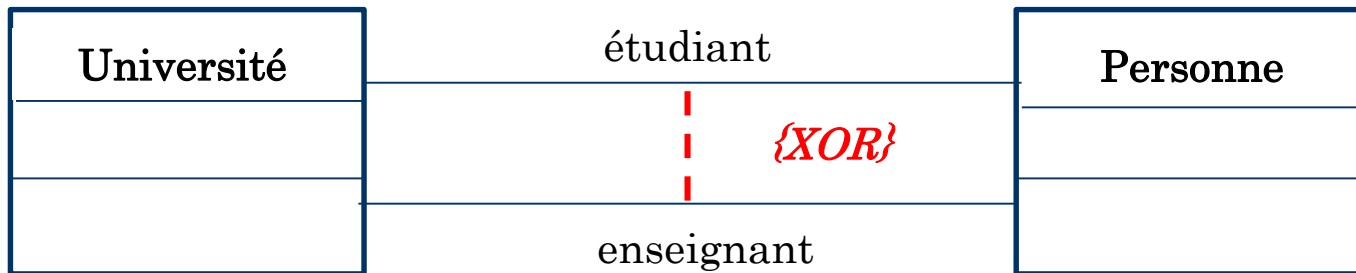
Une personne est née dans un pays, cette association ne peut être modifiée.

IV. UML

Diagramme de Classes/objets

2. Contraintes sur les relations:

{Xor} ou {Ou exclusif}:



- Chaque objet de la classe, participe à l'une ou l'autre association mais pas aux deux à la fois.
- Une personne joue le rôle d'enseignant ou d'étudiant dans une université mais pas les deux à la fois.

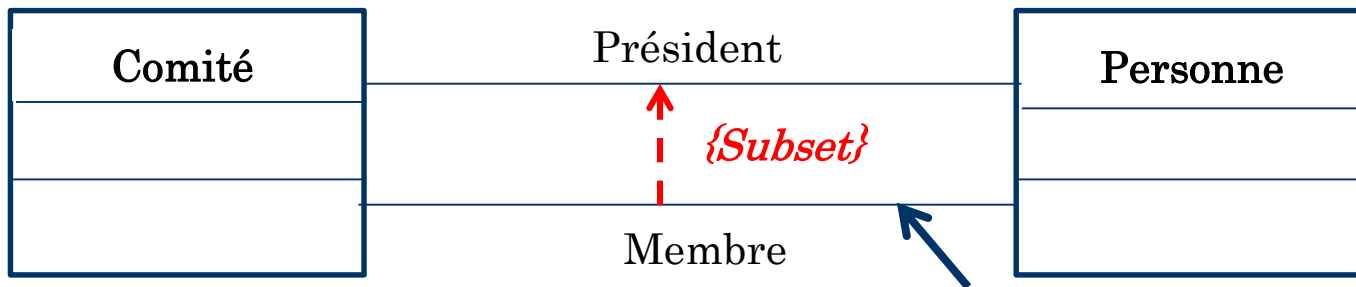
On peut utiliser ({contraintes}) toutes les autres contraintes (OU, Totalité,) connues.

IV. UML

Diagramme de Classes/objets

2. Contraintes sur les relations:

{subset}:



Indique que le président est également un membre du comité

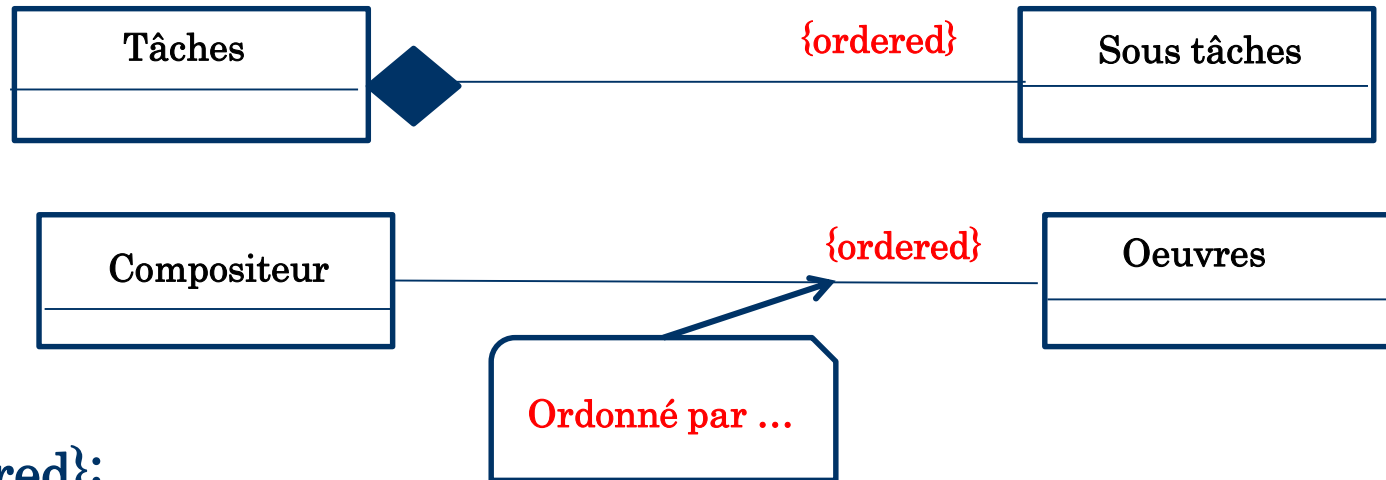
- Cette contrainte indique qu'une **collection est incluse** dans une **autre collection**.
- La flèche de la relation de dépendance indique le sens de la contrainte.

IV. UML

Diagramme de Classes/objets

2. Contraintes sur les relations:

{ordered}:



{ordered}:

Cette contrainte signifie que les objets sont ordonnés.
Le modèle ne signifie pas comment les objets sont ordonnés.

On peut préciser l'**ordre** par un commentaire sous forme de note

IV. UML

Diagramme de Classes/objets

1. Objectif.
2. Classes et objets.
3. Relations entre classes.
4. **Diagramme d'objets.**
5. Construction d'un diagramme de classes.

IV. UML

Diagramme de Classes/objets

4. Diagramme d'objets:

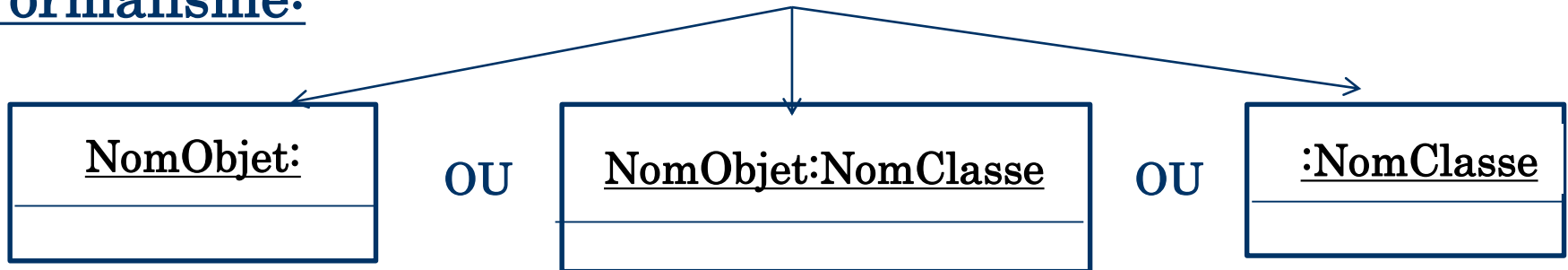
- Représente les **objets** (instances d'une classe) et leurs liens.
- Prend une image (snapshot) d'un système à un instant donné.



Permet d'éclaircir un diagramme des classes.

Toujours souligné les : sont obligatoires

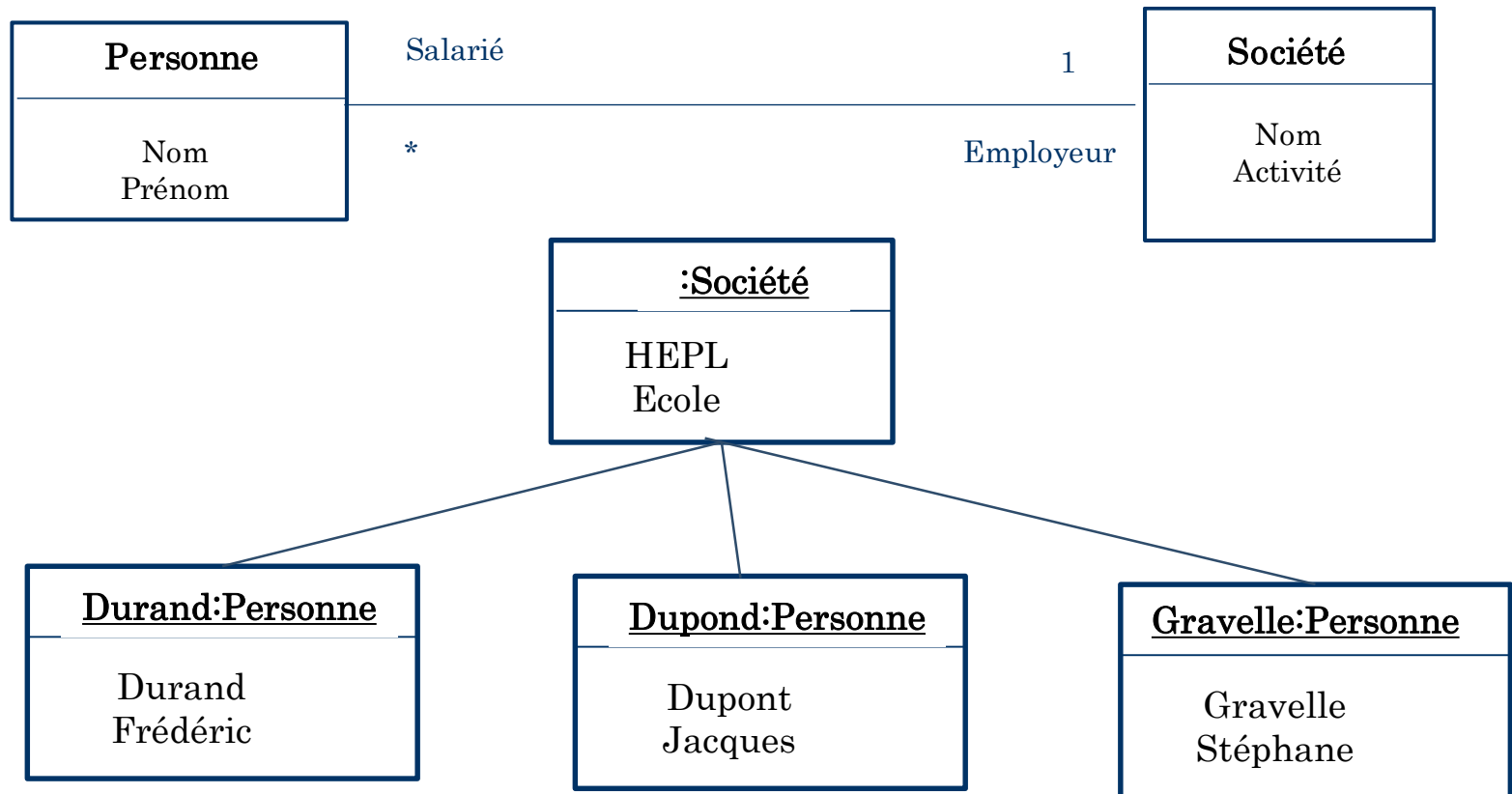
Formalisme:



IV. UML

Diagramme de Classes/objets

4. Diagramme d'objets: Exemples



IV. UML

Diagramme de Classes/objets

1. Objectif.
2. Classes et objets.
3. Relations entre classes.
4. Diagramme d'objets.
5. **Construction d'un diagramme de classes.**

IV. UML

Diagramme de Classes/objets

5. Construction d'un diagramme des classes:

Synthèse:

Pour élaborer un diagramme de classes il faut:

- *Identifier les concepts du domaine et les modéliser en **classes**;*
- *Identifier les **associations pertinentes** entre les concepts;*
- *Réfléchir aux **multiplicités** à chaque extrémité d'association;*
- *Ajouter des **attributs** aux classes du domaine;*
- *Utiliser les **diagrammes d'objets** pour illustrer les diagrammes de classes;*
- *Ajouter les **opérations** aux classes.*

IV. UML

Diagramme de Classes/objets

1. Objectif.
2. Classes et objets.
3. Relations entre classes.
4. Diagramme d'objets.
5. Construction d'un diagramme de classes.
6. **Package**

IV. UML

Diagramme de Classes/objets

Package:

Définition

Un **package** en UML (paquetage en français) est un regroupement d'éléments de modèle et de diagramme. Il permet de **modéliser** les éléments de modélisation en **groupes**.

Il peut contenir tout **types d'éléments** du modèle: des **classes**, des **cas d'utilisation**, des **interfaces**, etc. et même des **paquetages imbriqués** (décomposition hiérarchique).

IV. UML

Diagramme de Classes/objets

Package:

Pourquoi les packages?

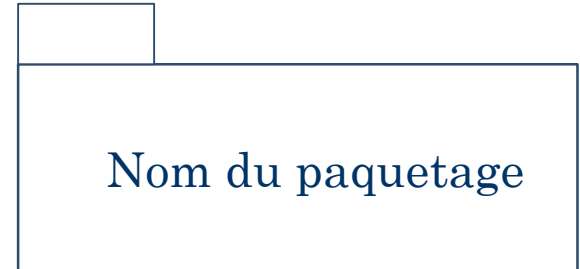
Les packages permettent de **structurer un modèle en unités cohérentes**, et donnent une **vision globale** plus claire.

IV. UML

Diagramme de cas d'utilisation

Package:

Un paquetage est représenté comme un **dossier** avec son nom inscrit dedans.



Il est possible de représenter explicitement le contenu d'un paquetage.
Dans ce cas, le nom est placé dans l'onglet.



IV. UML

Diagramme de cas d'utilisation

Package:

Remarques:

- Les **éléments** contenus dans un paquetage doivent représenter un **ensemble** fortement **cohérent** et sont généralement de **même nature** et de **même niveau sémantique**.
- Tout élément n'appartient qu'à **un seul** paquetage.
- Les paquetages constituent un mécanisme de gestion important des problèmes de **grande taille**.
- Il existe un **paquetage racine unique**, éventuellement anonyme, qui contient la totalité des modèles d'un système.

IV. UML

Diagramme de Classes/objets

Exercices: Donnez le diagramme de classes correspondant aux situations suivantes:

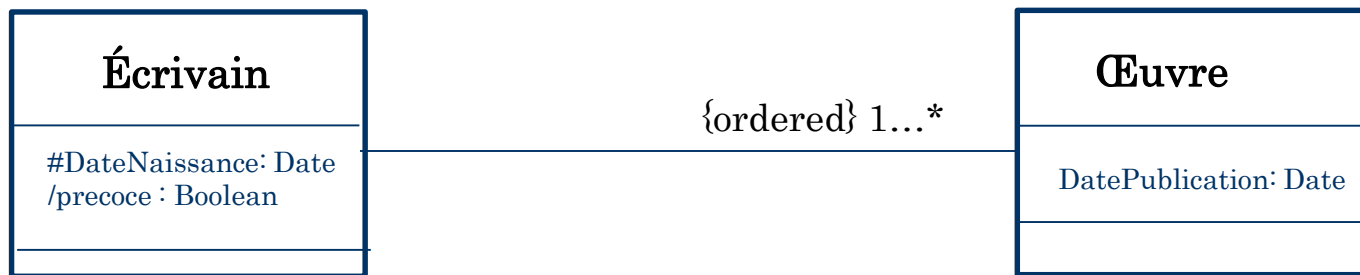
1. Un écrivain possède au moins une œuvre. Ses œuvres sont ordonnées selon l'année de publication. Si la première publication est faite avant l'âge de dix ans l'écrivain est dit « précoce ».

IV. UML

Diagramme de Classes/objets

Exercices: Donnez le diagramme de classes correspondant aux situations suivantes:

1. Un écrivain possède au moins une œuvre. Ses œuvres sont ordonnées selon l'année de publication. Si la première publication est faite avant l'âge de dix ans l'écrivain est dit « précoce ».



Les deux classes principales sont : **Écrivain** et **Œuvre**.

Les œuvres sont ordonnées selon la **date de publication**.

L'attribut dérivé « précoce » est à « 0 » ou « 1 » en fonction du résultat du calcul suivant:

$$DatePublication - DateNaissance < 10$$

IV. UML

Diagramme de Classes/objets

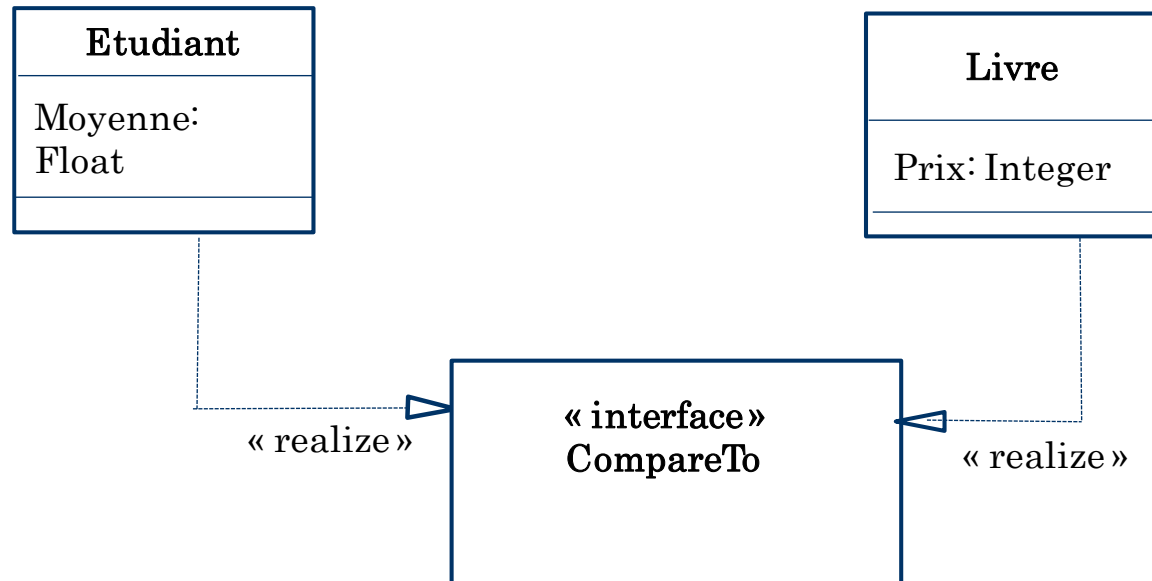
Exercices: Donnez le diagramme de classes correspondant aux situations suivantes:

2. Les étudiants de l'HEPL peuvent être comparés par rapport à l'attribut moyenne générale, par contre les livres de la bibliothèque sont comparables par rapport à leurs prix de vente.
Pour des raisons d'homogénéité des interfaces présentées par les classes, tous les objets comparables utilisent la même opération *CompareTo(Instance)*.

IV. UML

Diagramme de Classes/objets

Solution:



L'opération *CompareTo(instance)* compare tout type d'objet.

Elle est donc abstraite.

Si elle est commune à toutes les classes, alors on doit la définir dans une interface.

Cette interface sera réalisée (« realize ») par les classes qui l'utilisent.

IV. UML

Diagramme de Classes/objets

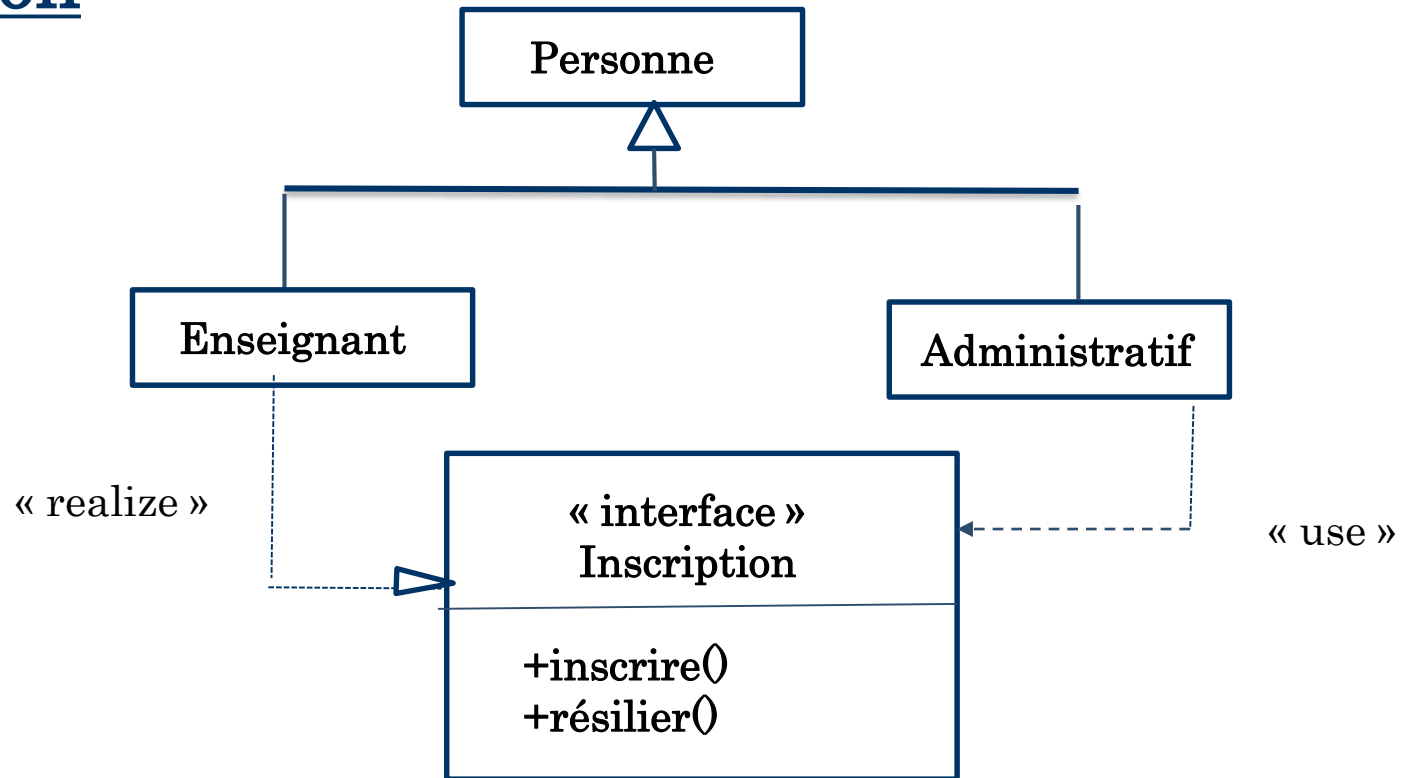
Exercices: Donnez le diagramme de classes correspondant aux situations suivantes:

3. L'université propose des cours de langue accessibles aux agents administratifs et aux enseignants. La procédure d'inscription est la même pour les deux catégories de personnes. Une personne inscrite peut également résilier son inscription. Simplifiez la modélisation en faisant apparaître uniquement les classes « *Enseignant* » et « *Administratif* ».

IV. UML

Diagramme de Classes/objets

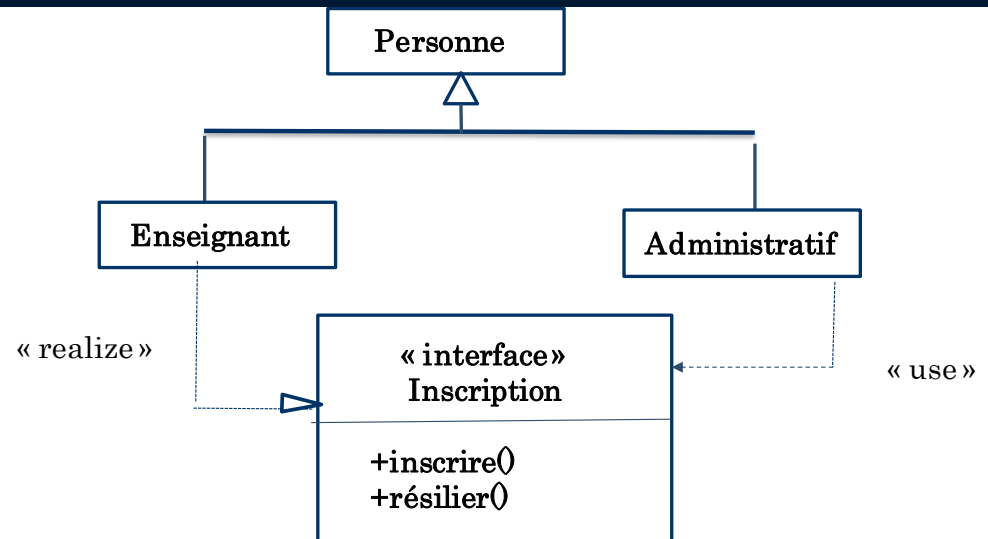
Solution:



IV. UML

Diagramme de Classes/objets

Solution:



Les *administratifs* et les *enseignants* sont des personnes particulières. Ils partagent les opérations *inscrire()* et *résilier()*, qui font partie de la même interface.

La **réalisation** de l'interface est fait de la même manière par les deux classes dérivées de *Personne*.

Mais il suffit qu'une seule des deux classes le **réalise**(pas de redondance).

La classe *enseignant* **réalise** l'interface inscription.

La classe *administratif* **utilise** le résultat.

IV. UML

Diagramme de Classes/objets

Exercice:

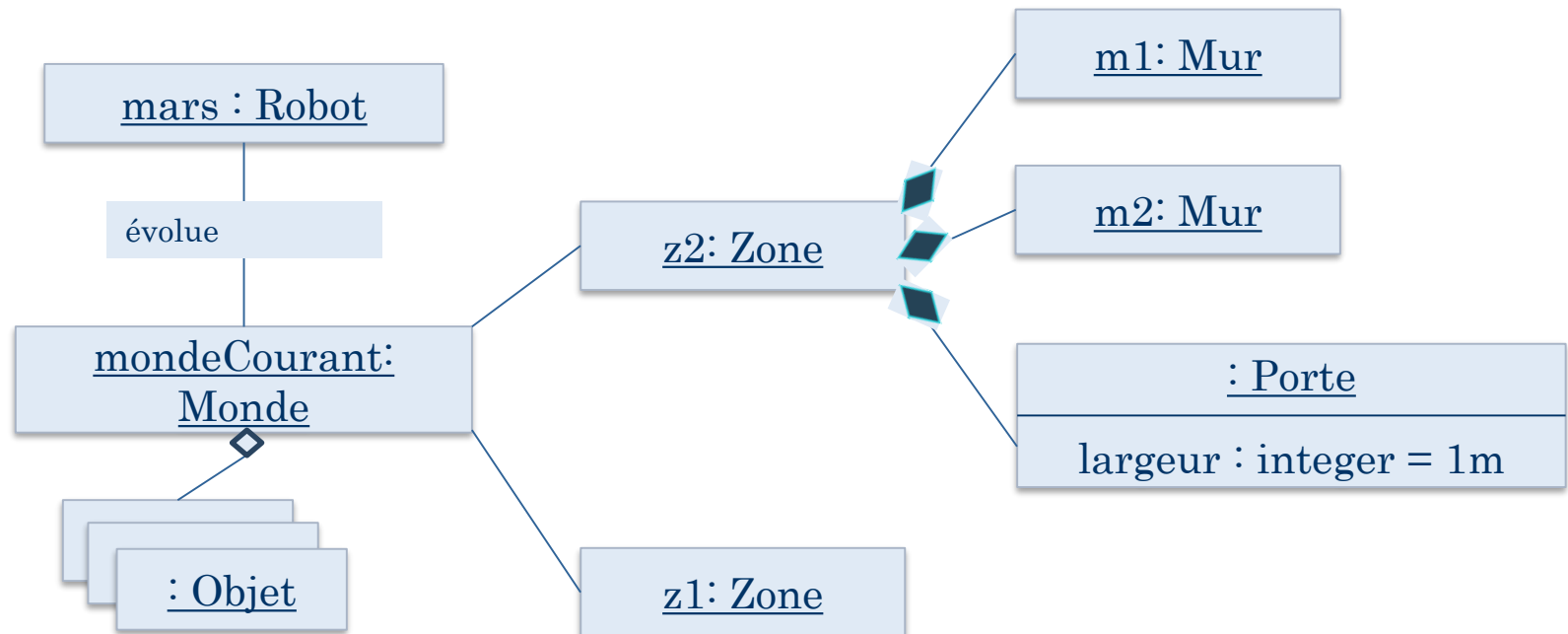
Un robot se déplace dans un environnement composé de zones, de murs et de portes, proposez un diagramme d'objets décrivant la situation suivante :

Le robot Mars est en mouvement. Il est lié à une instance mondeCourant de la classe Monde décrivant les mondes possibles où peut évoluer le robot. Le robot peut manipuler des objets se trouvant dans le monde dans lequel il évolue. A l'instant qui nous intéresse, le robot Mars est en mouvement et mondeCourant est lié aux zones z1 et z2. La zone z2 est composée de deux murs (m1 et m2) et d'une porte. La largeur de la porte est de 1 mètre.

IV. UML

Diagramme de Classes/objets

Solution:



IV. UML

Diagramme de Classes/objets

Exercice:

Représentez le diagramme de classes correspondant à l'énoncé suivant:

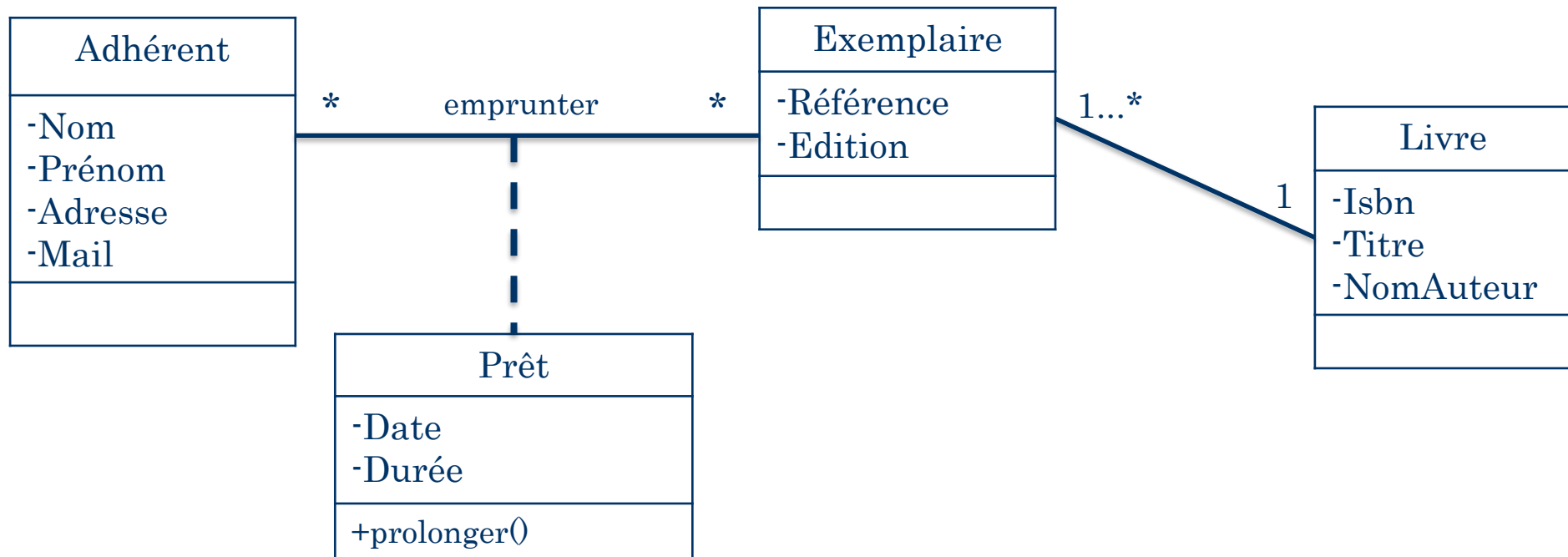
Une bibliothèque souhaite gérer les prêts des livres auprès de ses adhérents :

- Pour chaque adhérent, le système enregistre son nom, son prénom, son adresse, ainsi que son adresse mail
- Les livres sont caractérisés par leur titre, leur numéro d'ISBN et le nom de l'auteur
- Un même livre peut bien sûr se trouver en plusieurs exemplaires, chacun d'entre eux portant une référence et une année d'édition
- Pour tout prêt, le système enregistre la date et la durée prévue.
Une prolongation du prêt est envisageable

IV. UML

Diagramme de Classes/objets

Solution:



IV. UML

Diagramme de Classes/objets

Exercices:

Modélisez le diagramme de classes correspondant au SI suivant:

On souhaite gérer les cours dispensés dans plusieurs écoles.

- ✓ Chaque école est structurée en départements, qui regroupent chacun des enseignants spécifiques. Parmi ces enseignants, l'un est responsable du département.

- ✓ Chaque enseignant ne dispense qu'une seule matière.

- ✓ Une matière peut être enseignée par plusieurs enseignants, mais a toujours lieu dans la même salle de cours (chacune avec un nombre de places connues).

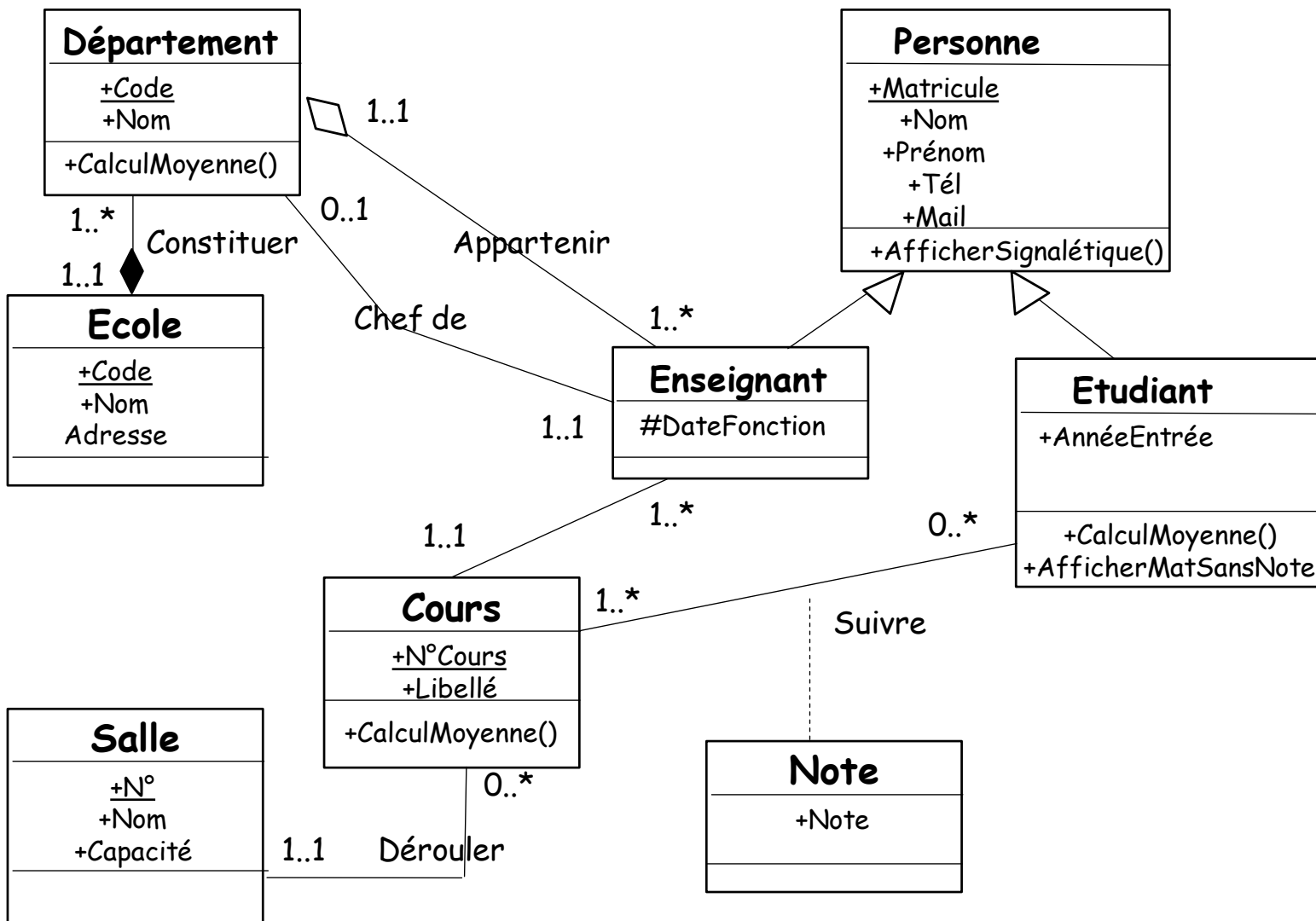
On désire pouvoir calculer la moyenne par matière, par département, ainsi que la moyenne générale d'un élève et pouvoir afficher les matières dans lesquelles il n'a pas été noté.

Bien entendu on veut pouvoir imprimer les signalétiques d'un élève et/ou d'un enseignant.

IV. UML

Diagramme de Classes/objets

Solution:



IV. UML

Diagramme de Classes/objets

Exercices: Donnez le diagramme de classes correspondant aux situations suivantes:

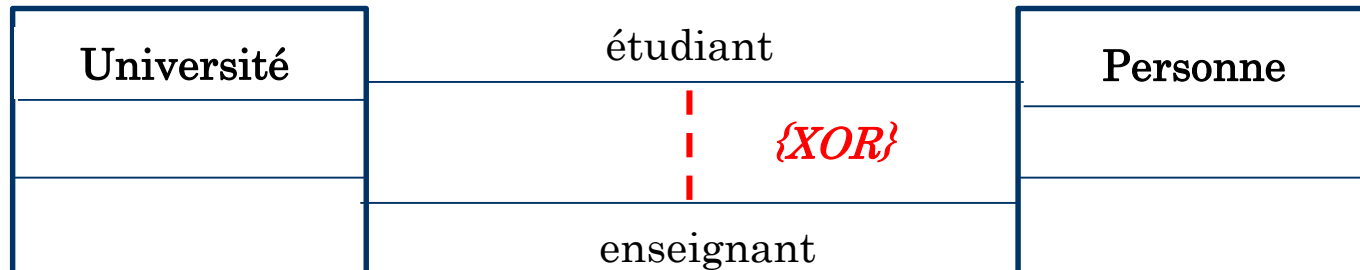
1. Les personnes qui sont associées à l'université sont des étudiants ou des professeurs.

IV. UML

Diagramme de Classes/objets

Exercices: Donnez le diagramme de classes correspondant aux situations suivantes:

1. Les personnes qui sont associées à l'université sont des étudiants ou des professeurs.



IV. UML

Diagramme de Classes/objets

Exercices: Donnez le diagramme de classes correspondant aux situations suivantes:

2. Tous les jours, le facteur distribue le courrier aux habitants de sa zone d'affectation. Quand il s'agit de lettres, il les dispose dans les boîtes aux lettres. Quand il s'agit d'un colis, le destinataire du courrier doit signer un reçu.

IV. UML

Diagramme de Classes/objets

Exercices: Donnez le diagramme de classes correspondant aux situations suivantes:

3. Tous les jours, le facteur distribue le courrier aux habitants de sa zone d'affectation. Quand il s'agit de lettres, il les dispose dans les boîtes aux lettres. Quand il s'agit d'un colis, le destinataire du courrier doit signer un reçu.

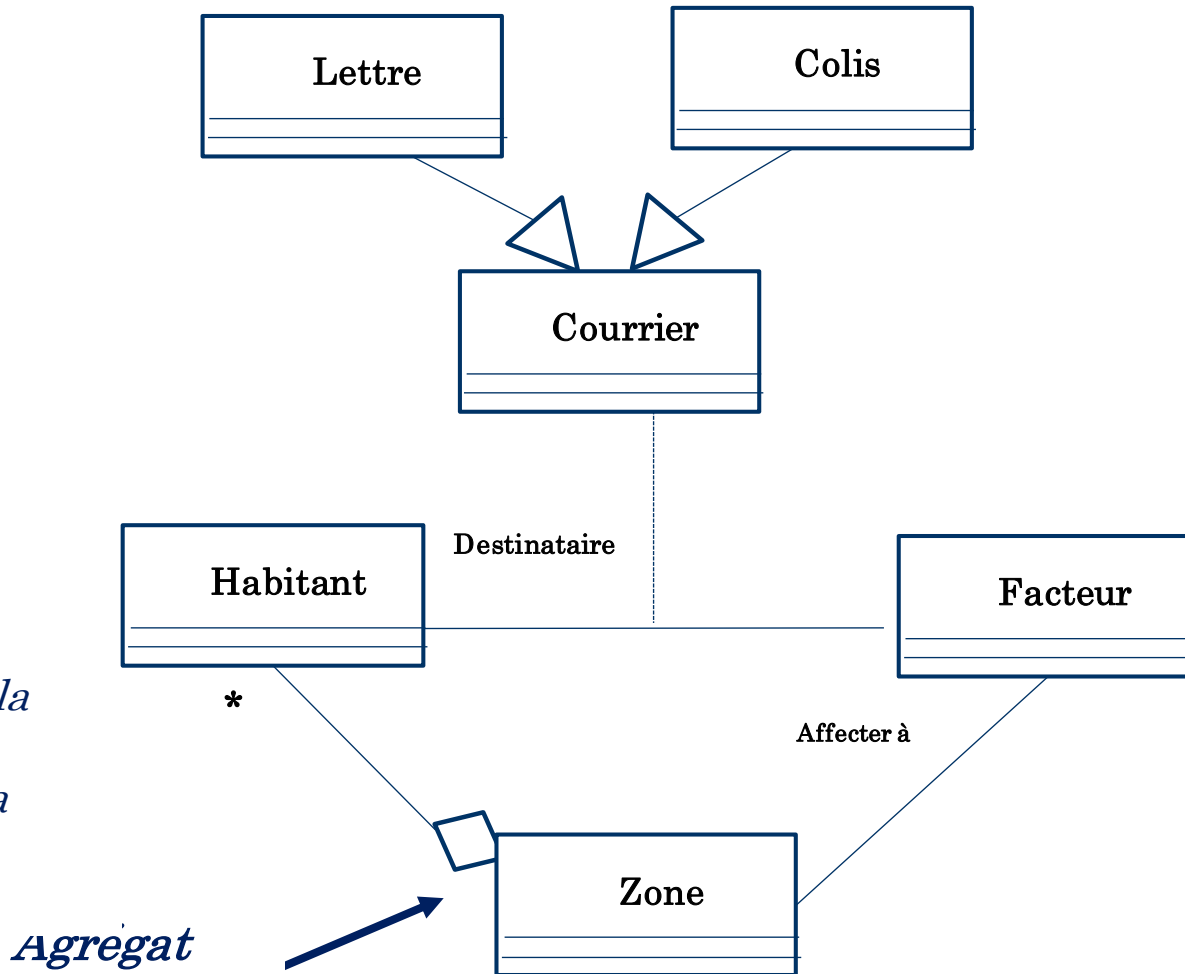
Les classes candidates sont : Facteur, Courrier, ZoneDAffectation, Lettre, BoîteAuxLettres, Colis, Destinataire.

- *La classe BoîteAuxLettres n'est pas pertinente.*
- *Un colis et une lettre sont des courriers particuliers.*
- *Le destinataire est le rôle d'un habitant quand il reçoit un courrier. Donc il ne sera pas représenté par une classe.*
- *Un facteur dessert une zone d'affectation qui abrite plusieurs habitants.*
- *Le seul lien entre le facteur et l'habitant est la distribution du courrier (Classe d'association).*

IV. UML

Diagramme de Classes/objets

Solution:



La destruction de la zone d'affectation n'entraînera pas la destruction de ses habitants.

Bibliographie

1. Benoît CHARROUX, Aomar OSMANI, Yann THIERRY-MIEG. UML2 Pratique de la modélisation. 3^{ème} édition. PEARSON.
2. Laurent AUDIBERT. UML2 de l'apprentissage à la pratique. 2^{ème} édition. ELLIPSES.
3. Christian SOUTOU. Modélisation des bases de données (UML et les modèles entité-association). 3^{ème} édition. EYROLLES.
4. Chantal MORLEY, Jean HUGUES, Bernard LEBLANC. 4^{ème} édition. UML 2 pour l'analyse d'un système d'information. DUNOD.
5. Hugues BERSINI. L'orienté objet. 3^{ème} édition. EYROLLES.
6. Laurent DEBRAUWER, Fien VAN DER HEYDE. UML 2.5. 4^{ème} édition. ENI Editions.
7. Jean-Luc HAINAUT. Bases de données concepts, utilisation et développement. DUNOD.
8. Gilles ROY. Conception de bases de données avec UML. Presses de l'université du Québec.
9. Craig LARMAN. UML2 et les design patterns. 3^{ème} édition. PEARSON Education.
10. Frank BARBIER. UML 2 et MDE. DUNOD.
11. Laurent DEBRAUWER, Naouel KARAM. UML 2 entraînez-vous à la modélisation. Seconde édition. ENI Editions.
12. Corine COSTA. Cours Projets et bureau d'études.