

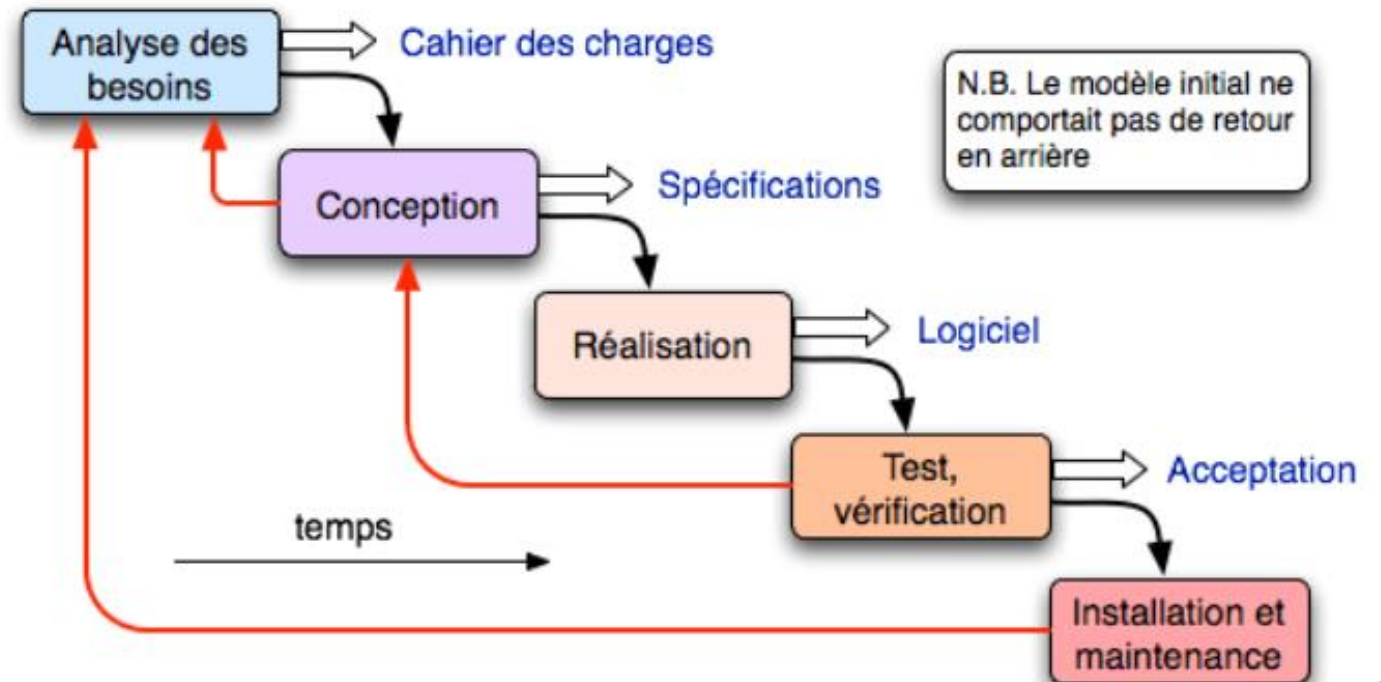
Génie Logiciel



- Dispensé par Dr. Msc. Ir. **MWAMBA KASONGO Dahouda**
Docteur en génie logiciel et systèmes d'information
Machine and Deep Learning Engineer

- Assisté Master Student, Ir. Jason MUSA

Heure : 08H00 – 12H00



Chapitre 4

Diagramme de classe

Diagramme d'objet

TD numero

- 1.Qu'est ce qu'un cas d'utilisation ?
- 2.A quoi sert un diagramme de cas d'utilisation?
- 3.Dessinez et expliquer la relation entre acteurs
- 4.Dessinez et Donnez la difference entre relation d'inclusion et une relation d'extension entre cas d'utilisation
- 5.A quoi sert la description textuelle d'un cas d'utilisation

CHAPITRE 4. DIAGRAMME DE CLASSE ET D'OBJET : Vue statique



4.1. Definition et principe de base

4.2. Representation d'une classe

4.3. Les types de relations entre classe

4.4. Diagramme d'objet



CHAPITRE 4. DIAGRAMME DE CLASSE ET D'OBJET : Vue statique



4.1. Diagramme de classe

4.1.1 Objectif:

- ✓ Le diagramme de classes est sans doute le diagramme le plus important à représenter pour les méthodes d'analyse orientées objet.
- ✓ C'est le point central de tout développement orienté objet.
- ✓ Le diagramme de classes exprime la structure statique du système en termes de classes et de relations entre ces classes.
- ✓ L'intérêt du diagramme de classe est de **modéliser les entités** du système d'information

4.1.2 Definition

- Un diagramme de classes est une collection d'éléments de modélisation statique qui montre la structure d'un modèle.
- Un diagramme de classes fait abstraction des aspects dynamiques et temporels du système
 - Une **classe** est une description abstraite d'un ensemble d'objets du domaine de l'application : elle définit leur **structure**, leur **comportement** et leurs **relations**.
 - Une classe représente la description d'un ensemble d'objets possédant les mêmes **caractéristiques**.



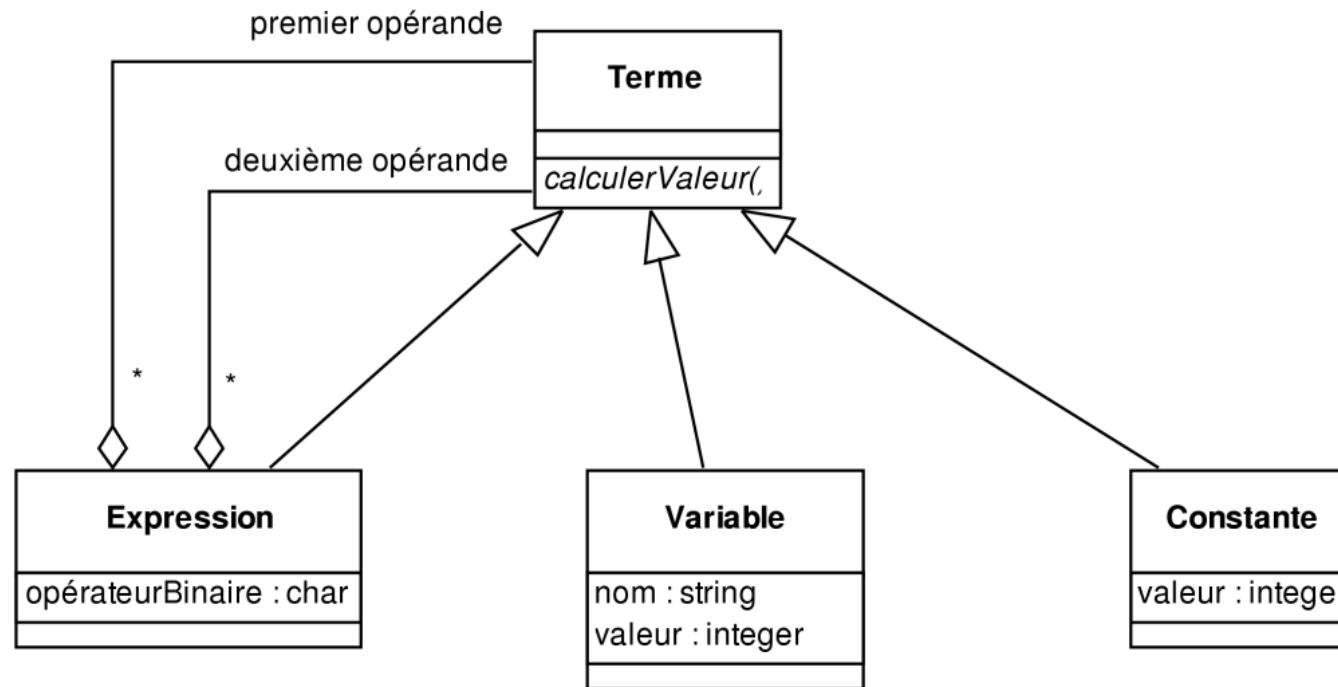
CHAPITRE 4. DIAGRAMME DE CLASSE ET D'OBJET : Vue statique



4.1. Diagramme de classe

4.1.3 Representation:

Le diagramme de classes met en œuvre **des classes**, contenant des **attributs** et des **opérations**, et reliées par des **associations** ou des **généralisations**.



CHAPITRE 4. DIAGRAMME DE CLASSE ET D'OBJET : Vue statique



4.1. Diagramme de Classe

4.1.4 Classe

Une classe est représentée par un rectangle séparé en trois parties:

- ✓ La première partie représente **le nom** de la classe,
- ✓ La deuxième partie représente **les attributs** de la classe,
- ✓ La troisième partie représente **les opérations** de la classe.

Formalisme

NOM CLASSE
Attribut_1 : int
Attribut_2 : int
Attribut_3 : int
Operation_1 () : void
Operation_2 () : void

4.1.5 Attribut

Un attribut représente la modélisation d'une information élémentaire représentée par son nom et son format (type).
UML définit 3 niveaux de visibilité pour les attributs :

- ✓ **public (+)** : l'élément est visible pour tous les éléments de la classe,
- ✓ **protégé (#)** : l'élément est visible pour les sous-classes de la classe,
- ✓ **privé (-)** : l'élément n'est visible que par les objets de la classe dans laquelle il est déclaré.



CHAPITRE 4. DIAGRAMME DE CLASSE ET D'OBJET : Vue statique



4.1. Diagramme de classe

4.1.6 Operation

Une opération représente un élément de comportement des objets, défini de manière globale dans la classe.

- ✓ Une opération est une fonctionnalité assurée par une classe.
 - ✓ La description des opérations peut préciser les paramètres d'entrée et de sortie ainsi que les actions élémentaires à exécuter.
- Comme pour les attributs, on retrouve 3 niveaux de visibilité pour les opérations:

FACTURE	
+ No facture	: int
+ Date	: Date
+ Montant	: double
+ / Montant TVA	: double
+ Op publique ()	
# Op protégée ()	
- Op privée ()	

- **Visibilité publique** : Les opérations marquées comme publiques sont accessibles à tout le système, sans restrictions. Elles peuvent être appelées depuis n'importe quel module, classe, ou fonction, et sont donc visibles dans l'ensemble du programme.
- **Visibilité protégée** : Ce niveau limite l'accès aux opérations à l'intérieur de la classe où elles sont définies et aux classes qui en héritent. Les opérations protégées sont visibles uniquement pour la classe parente et ses sous-classes, ce qui garantit une certaine encapsulation tout en permettant une extension par héritage.
- **Visibilité privée** : Les opérations privées sont accessibles uniquement au sein de la classe où elles sont définies. Cela signifie qu'elles ne peuvent pas être appelées ni modifiées directement depuis l'extérieur de cette classe ou par des sous-classes. Ce niveau est souvent utilisé pour sécuriser les données sensibles ou les processus internes.



CHAPITRE 4. DIAGRAMME DE CLASSE ET D'OBJET : Vue statique



4.1. Diagramme de classe

4.1.7 Type des relations entre classe

il existe plusieurs types de relations entre classes :

1 - Association:

- Une association est une relation entre deux classes qui décrit les connexions structurelles entre leurs instances. Une association indique donc qu'il peut y avoir des liens entre des instances des classes associées.
- Une association est une relation statique n-aire (le plus souvent : elle est binaire) : c'est à dire qu'elle relie plusieurs classes entre elles.
- Une association n-aire possède n rôles qui sont les points terminaux de l'association.

Chaque classe qui participe à l'association joue un rôle. Les rôles sont définis par 2 propriétés :

- ✓ **Nom de rôle** = indication sur la participation de la classe à l'association
- ✓ **Multiplicité** = définit le nombre d'instances de l'association pour une instance de la classe.



CHAPITRE 4. DIAGRAMME DE CLASSE ET D'OBJET : Vue statique



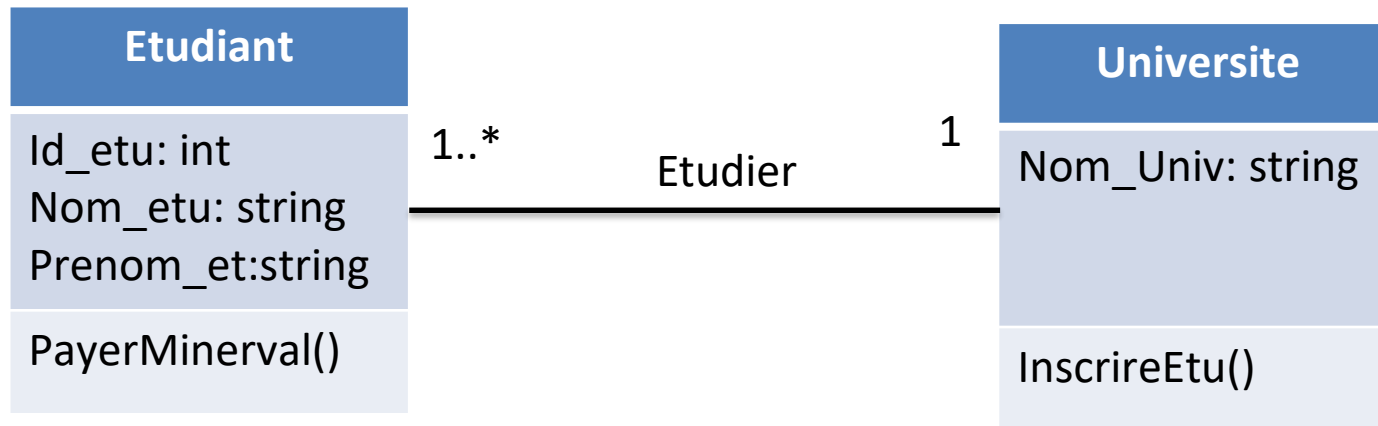
4.1. Diagramme de classe

4.1.7 Type des relations entre classe

1 – Example : Association:

Dans ce diagramme, le nom de l'association est **Etudier**, le nom du rôle de la classe **Etudiant** pour l'association est: étudié.
Le diagramme se lit comme suit :

- ✓ Un étudiant étudie dans une seule Université.
- ✓ Dans une université il y a une à plusieurs étudiants:



1	Un et un seul
0..1	Zéro ou un
m..n	De m a n (entier)
*	Plusieurs
0..*	De zero a plusieurs
1..*	D'un a plusieurs



CHAPITRE 4. DIAGRAMME DE CLASSE ET D'OBJET : Vue statique



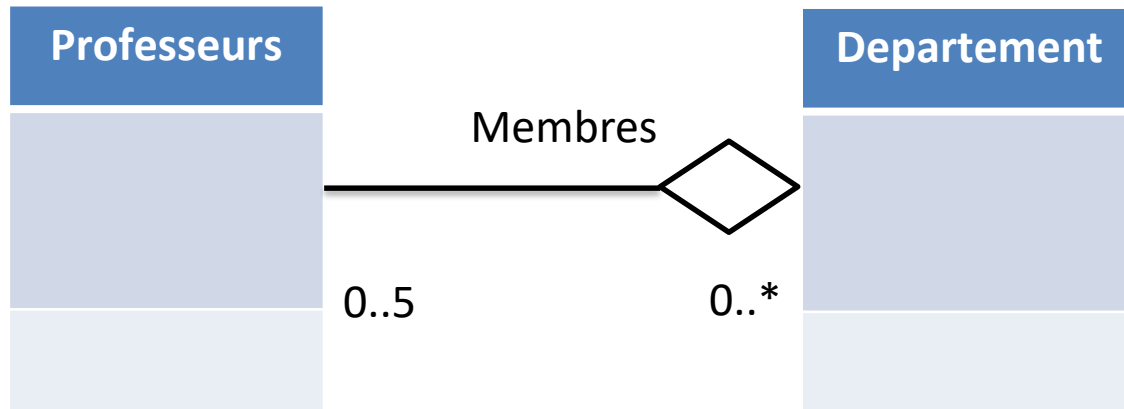
4.1. Diagramme de classe

4.1.7 Type des relations entre classe

1 – L'agrégation

L'agrégation représente une association non symétrique dans laquelle une des extrémités joue un rôle prédominant par rapport à l'autre extrémité.

- ✓ L'agrégation se représente toujours avec un petit losange du côté de l'agregat.
- ✓ Elle est asymétrique du type « ensemble / élément » ou « contenant / contenu »



CHAPITRE 4. DIAGRAMME DE CLASSE ET D'OBJET : Vue statique



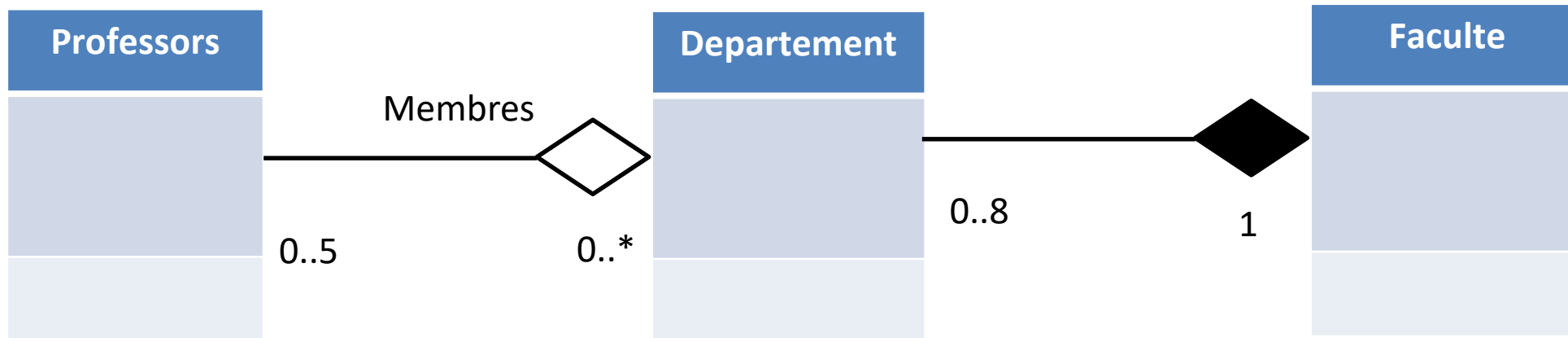
4.1. Diagramme de classe

4.1.7 Type des relations entre classe

1 – La composition

La composition est un cas particulier de l'agrégation dans laquelle la vie des composants est liée à celle des agrégats. Elle fait souvent référence à une contenance physique.

La composition implique, en plus de l'agrégation, une coïncidence des durées de vie des composants : **la destruction de l'agrégat (ou conteneur) implique automatiquement la destruction de tous les composants liés.**



CHAPITRE 4. DIAGRAMME DE CLASSE ET D'OBJET : Vue statique



4.1. Diagramme de classe

4.1.7 Type des relations entre classe

1 – La Généralisation/ spécialisation

Le principe de **généralisation** / **spécialisation** permet d'identifier parmi les objets d'une classe (générique) des sous-ensembles d'objets (des classes spécialisées) ayant des définitions spécifiques.

✓ La classe plus spécifique (appelée aussi classe fille, classe dérivée, classe spécialisée, classe descendante ...) est cohérente avec la classe plus générale (appelée aussi classe mère, classe générale).

c'est-à-dire qu'elle contient par héritage tous les attributs, les membres, les relations de la classe générale, et peut contenir d'autres.

Une classe peut avoir plusieurs parents, on parle alors d'**héritage multiple**.

Une **classe abstraite** est une classe qui ne s'instancie pas directement, mais qui représente une simple abstraction afin de factoriser les propriétés communes des sous-classes. Elle se note en italique.



CHAPITRE 4. DIAGRAMME DE CLASSE ET D'OBJET : Vue statique



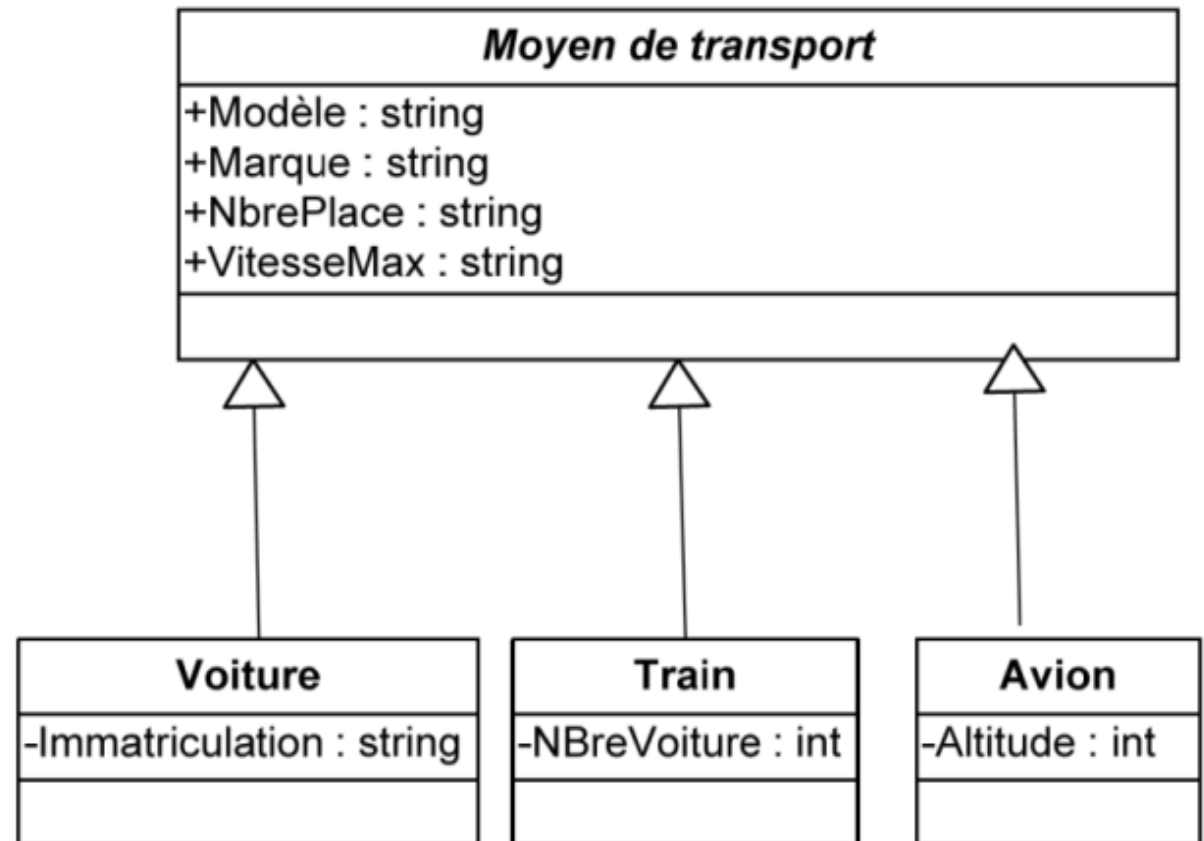
4.1. Diagramme de classe

4.1.7 Type des relations entre classe

1 – La Généralisation/ spécialisation

Illustration:

La classe « *Moyen de transport* » est une classe abstraite.
La généralisation multiple consiste à fusionner plusieurs classes en une seule classe.



CHAPITRE 4. DIAGRAMME DE CLASSE ET D'OBJET : Vue statique



4.2. Diagramme d'objet

4.2.1 Definition

- ✓ Le diagramme d'objets permet de mettre en évidence des liens entre les objets.
- ✓
- ✓ Les objets, instances de classes, sont reliés par des liens, instances d'associations.

A l'exception de la multiplicité, qui est explicitement indiquée, le diagramme d'objets utilise les mêmes concepts que le diagramme de classes.

Ils sont essentiellement utilisés pour comprendre ou illustrer des parties complexes d'un diagramme de classes.

Un diagramme d'objets est une instance d'un diagramme de classes représentant des objets et les liens qui les unissent.

Un diagramme d'objets ne montre pas les interactions entre les objets.



CHAPITRE 4. DIAGRAMME DE CLASSE ET D'OBJET : Vue statique



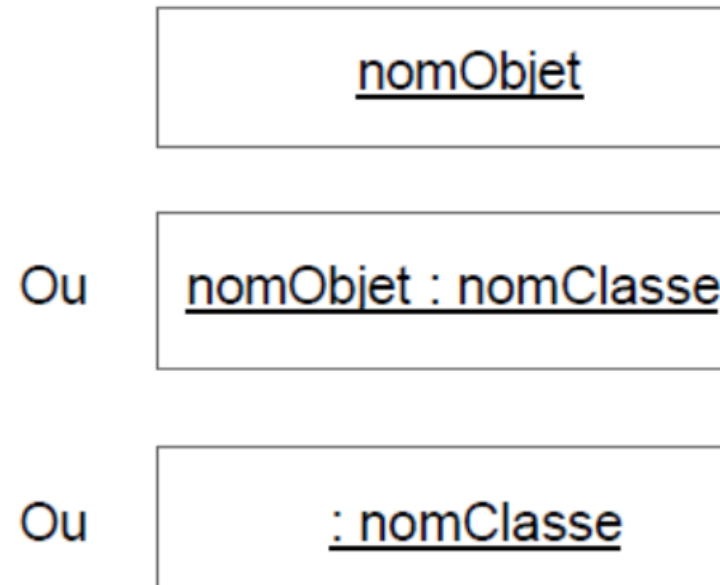
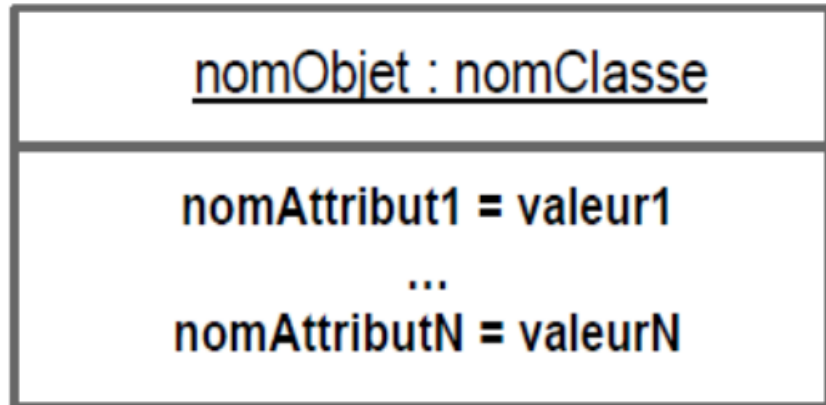
4.2. Diagramme d'objet

4.2.2 Representation

Un diagramme d'objet est représenté par un rectangle découpé en deux compartiments :

- ✓ Le nom de l'objet,
- ✓ La liste des attributs valorisés

Le nom de l'objet peut être présenté selon 3 formats plus ou moins précis:



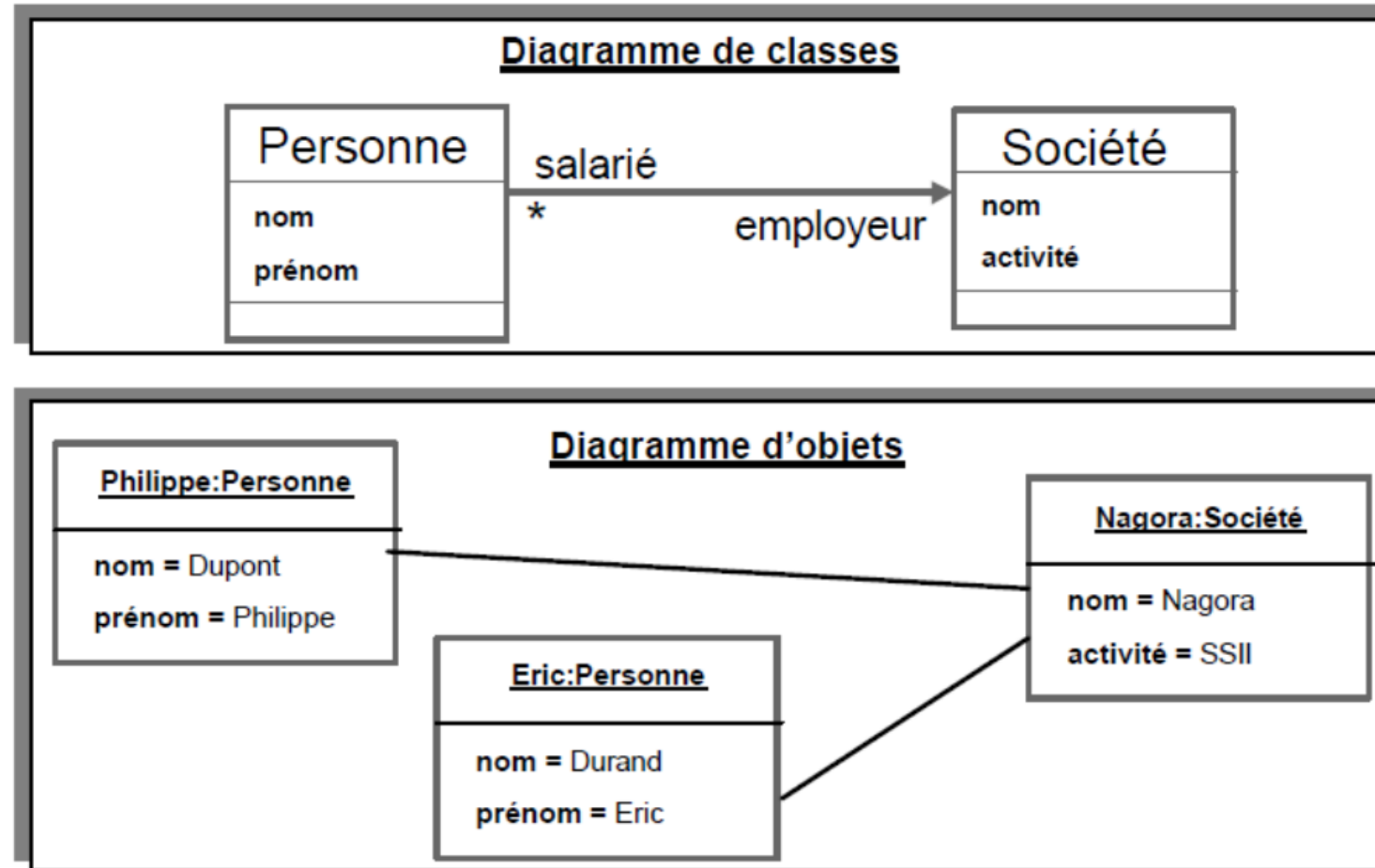
CHAPITRE 4. DIAGRAMME DE CLASSE ET D'OBJET : Vue statique



4.2. Diagramme d'objet

4.2.2 Representation

Exemple :



CHAPITRE 4. DIAGRAMME DE CLASSE ET D'OBJET : Vue statique

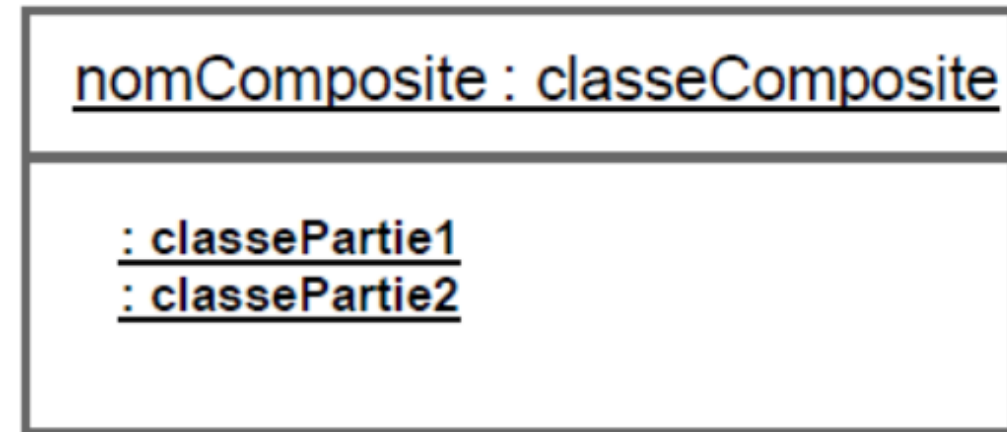
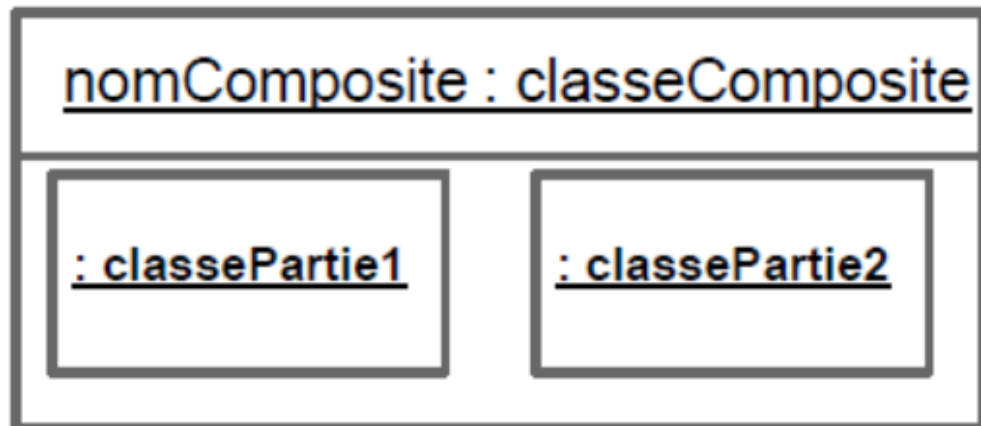


4.2. Diagramme d'objet

4.2.3 Composite

Un objet compose de sous-objets est appelé **composite**.

- ✓ Les objets composites sont issus de classes liées par une composition.
- ✓ Les attributs d'un objet composite sont eux-mêmes des objets.
- ✓ Les sous-objets peuvent être représenté de manière graphique ou textuelle.



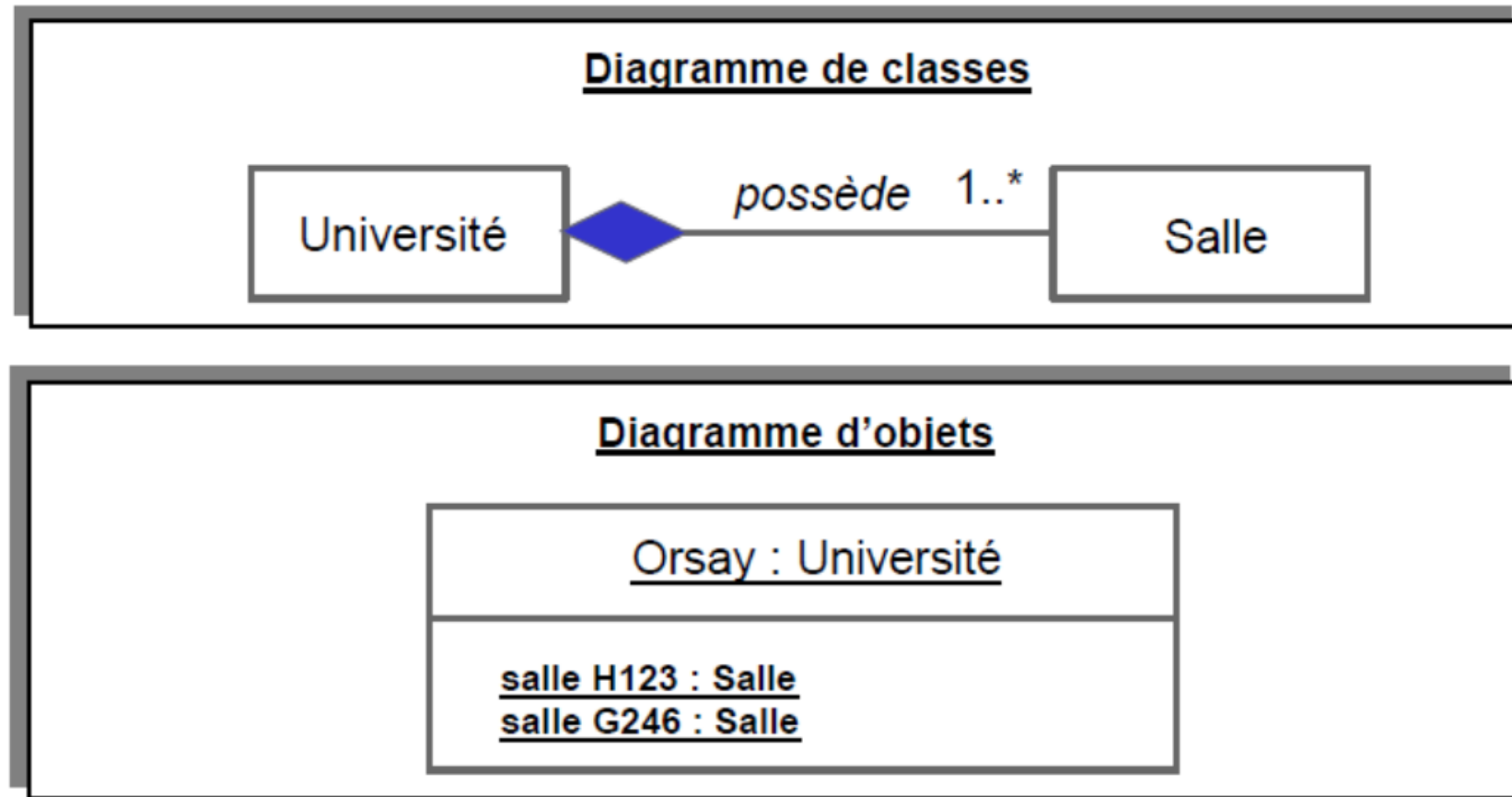
CHAPITRE 4. DIAGRAMME DE CLASSE ET D'OBJET : Vue statique



4.2. Diagramme d'objet

4.2.3 Composite

Exemple :



CHAPITRE 4. DIAGRAMME DE CLASSE ET D'OBJET : Vue statique



4.2. Diagramme d'objet

4.2.4 A retenir

Le diagramme d'objets ne doit être utilisé que pour clarifier certaines structures complexes apparaissant sur un diagramme de classes.

Tous les objets du diagramme de classes ne doivent pas obligatoirement figurer sur le diagramme d'objets.

Le diagramme d'objets peut servir de base à un diagramme d'interaction.



CHAPITRE 4. DIAGRAMME DE CLASSE ET D'OBJET : Vue statique



4.3. Autres Diagramme UML (Diagramme d'interaction)

4.3. Objectif

➤ **Représenter les communications avec le logiciel et au sein du logiciel.**

- **Diagramme de communication**

- ✓ Représentation **spatiale** des objets et des acteurs et de leurs interactions.

- **Diagramme de séquence**

- ✓ Représentation **temporelle** des interactions entre les objets et entre les objets et les acteurs.

- ✓ **Chronologie** des messages échangés entre les objets et avec les acteurs.



CHAPITRE 4. DIAGRAMME DE CLASSE ET D'OBJET : Vue statique



4.3. Autres Diagramme UML (Diagramme d'interaction)

4.4. Exemple : A partir d'un diagramme de classe et d'un cas d'utilisadation

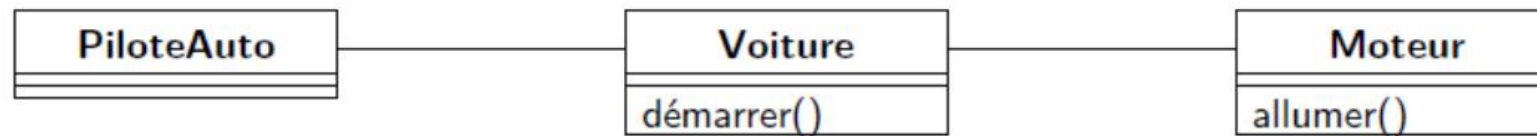


Diagramme de communication



Diagramme de séquence

