

Génie Logiciel

Niveau : 3^{ème} année licence (MI)

Chapitre 4 : Diagramme de classes et d'objets (Vue statique)

Chargés de cours :

Dr M. MOHAMMEDI
& Dr N. YESSAD

Année universitaire 2025/2026

Plan du chapitre 4

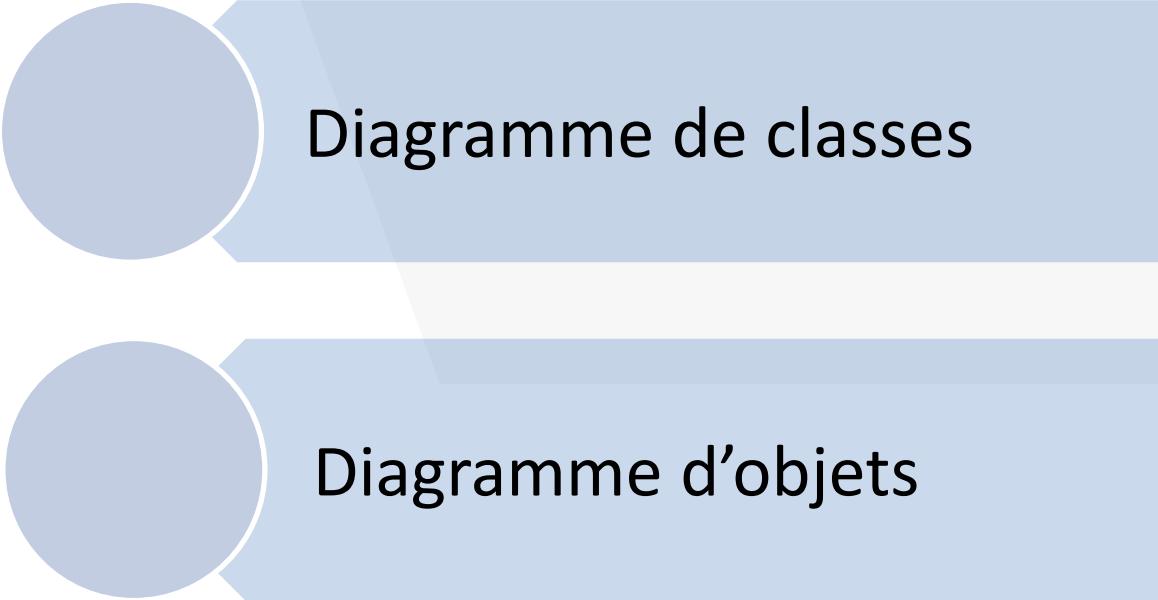


Diagramme de classes

Diagramme d'objets

Introduction

- Un diagramme de classes constitue l'un des **pivots essentiels** de la modélisation avec UML.
- Ce diagramme permet de donner la **représentation statique du système** à développer en intégrant dans chaque classe la partie dédiée aux données et celle consacrée aux traitements.
- Cette représentation est centrée sur les concepts de **classes et leurs relations**.

Diagramme de classes: Concepts de base

Classe

- Une classe est un concept **abstrait** qui décrit un **ensemble d'objets** ayant les **mêmes propriétés** (attributs) et **un même comportement** (opérations).
- Une classe peut représenter des éléments **variés** comme :
 - des éléments concrets (ex. :des avions) ;
 - des éléments abstraits (ex. : service) ;
 - des éléments d'une application (ex. : boites de dialogues) ;
 - des éléments comportementaux (ex. : des tâches), etc.

Diagramme de classes: Concepts de base

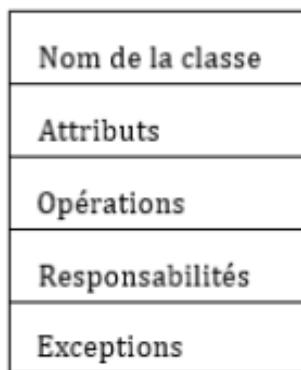
Représentation UML d'une classe

- Une classe se représente à l'aide d'un **rectangle** comportant **plusieurs compartiments**. **Les trois** compartiments de base sont :
 - la désignation de la classe
 - la description des attributs
 - la description des opérations
- **Deux** autres compartiments peuvent être aussi indiqués:
 - la description des responsabilités qui énumère l'ensemble des tâches devant être assurées par la classe.
 - la description des exceptions qui énumère les situations exceptionnelles devant être traitées et gérées par la classe.

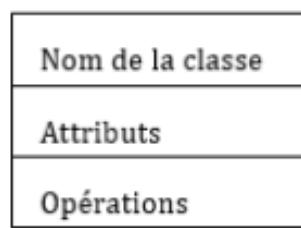
Remarque: Il est à noter qu'une description d'une classe peut être restreinte à un nombre réduit de compartiments selon les objectifs de la modélisation

Diagramme de classes: Concepts de base

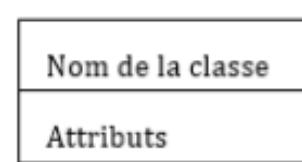
Représentation UML d'une classe



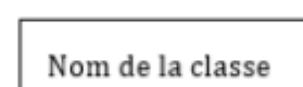
Description complète



Description à trois
compartiments
(couramment utilisée)



Description à deux
compartiments



Description réduite à
la désignation de la
classe

Diagramme de classes: Concepts de base

Attributs

- Un attribut est une **propriétés élémentaires** d'une classe.
- Représente les données encapsulées dans les objets de cette classe.
- Un attribut est définie par **un nom, un type de données, une visibilité** et peut être initialisé.
- La syntaxe complète de déclaration d'un attribut est la suivante :

Visibilité / Nom attribut : type [=valeur initiale {propriétés}]

- Le nom de l'attribut doit **être unique** dans la classe et son type peut être de **type primitif** (entier, chaîne de caractère,etc).
- La valeur initiale est une valeur facultative donnée à l'initialisation d'un attribut de la classe.
- Le champ {propriétés} représente aussi des valeurs marquées facultatives (ex. "interdit" pour mise à jour interdite).

Diagramme de classes: Concepts de base

Attributs (suite)

- Les attributs dérivés sont des attributs dont leurs valeur peut être calculée à partir d'autres attributs de la classe.
- Les attributs dérivés sont notés par "**/nom de l'attribut dérivé**"

Diagramme de classes: Concepts de base

Opérations

- Une opération est une **fonction applicable** aux objets d'une classe qui permet de décrire leurs comportements.
- Il est à noter qu'une méthode est l'implémentation d'une opération.
- Chaque opération est désigné soit par:
 - **son nom** , ou
 - **son nom, sa liste de paramètres et son type de résultat.**
- la description complète des opérations est la suivante :

Visibilité / Nom opération (paramètres) [:[type résultat] {propriétés}]
- Chaque paramètre peut être décrit en plus de son nom, par son type et sa valeur par défaut. L'absence de paramètres est indiquée par ().

Diagramme de classes: Concepts de base

Visibilité des attributs / opérations

- Chaque attribut ou opération d'une classe peut être de type :
 - Public (+)** : Attribut/ opération visible par tous ;
 - Protégé (#)** : Attribut/ opération visible seulement à l'intérieur de la classe et pour toutes les sous-classes de la classe ;
 - Privé (-)** : Attribut/ opération seulement visible à l'intérieur de la classe ;
 - Paquetage (~)** : Attribut/ opération ou classe visible seulement l'intérieur du paquetage où se trouve la classe.

Diagramme de classes: Association entre classes

- Une association est une **relation sémantique** entre classes, qui décrit les connexions structurelles entre leurs instances.
- Une association est représentée par un **lien reliant deux classes** identifiée par un nom (forme verbale) et **des multiplicités** sur les extrémités.
- Il est possible d'exprimer ainsi **des rôles** sur le lien d'association.



Diagramme de classes: Association entre classes

Multiplicité d'association

- La multiplicité spécifie **le nombre d'instances** d'une classe pouvant être liées à **une seule instance** d'une classe associée.
- Le domaine de valeurs peut être décrit sous plusieurs formes : **intervalle fermé, valeurs exactes, valeurs indéterminées**.
- Ce présent tableau illustre les principales multiplicités :

Domaine de valeurs	Interprétation
1	Une et une seule instance
0 .. 1	Zéro ou une instance
N	N instances (entier naturel)
M .. N	de M à N instances (entiers naturels)
*	De zéro à un nombre indéterminé d'instances
0 .. *	De zéro à un nombre indéterminé d'instances
1 .. *	De un à un nombre indéterminé d'instances

Diagramme de classes: Association entre classes

Multiplicité d'association (Suite)

- L'interprétation des multiplicités dans cet exemple est la suivante: à une instance de la classe A correspond 1 à nombre indéterminé d'instances de B et à une instance de la classe B correspond 2 à 10 instances de A.

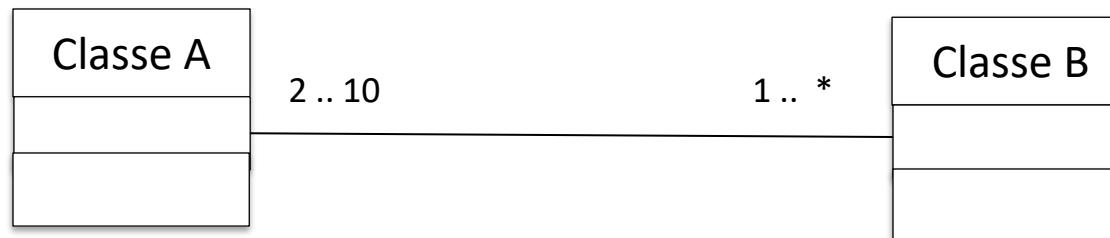
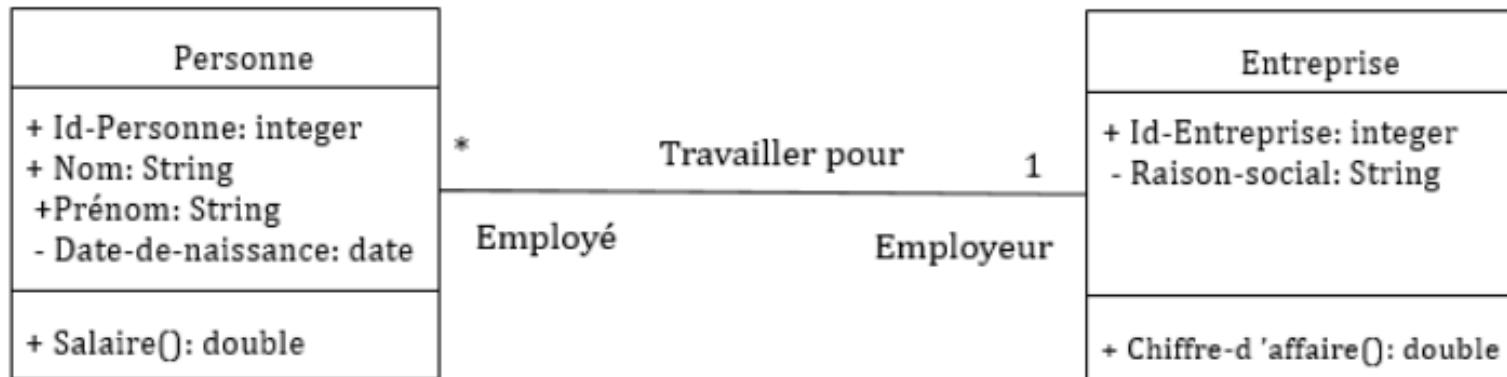


Diagramme de classes: Association entre classes

Rôle d'association

Le rôle est une forme nominale, décrit **la fonction d'une classe** pour une association donnée. il est placé à **une extrémité de l'association**.



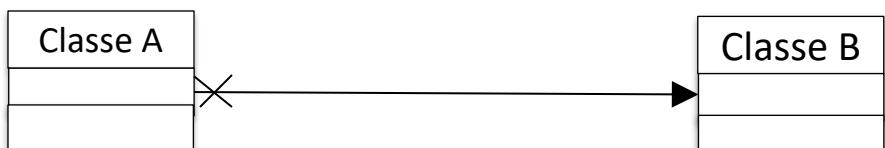
- Dans cet exemple, une personne travaille pour une et une seule entreprise.
- L'entreprise peut accueillir plusieurs personnes à la fois.
- L'entreprise est l'employeur des personnes qui travaillent pour elle et une personne a un statut d'employé dans l'entreprise.

Diagramme de classes: Association entre classes

Navigabilité

- La navigabilité indique si l'association fonctionne de manière unidirectionnelle ou bidirectionnelle,
- Elle est représentée par une ou deux extrémités fléchées.
- La non-navigabilité se représente par un " X".

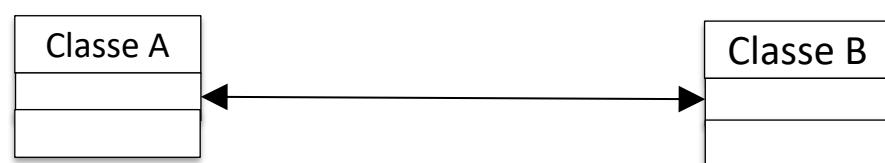
Les situations possibles de navigabilité sont représentées comme suit :



Navigabilité unidirectionnelle de A vers B
Pas de navigabilité de B vers A
non définie explicitement



Navigabilité unidirectionnelle de B vers A
Pas de navigabilité de A vers B
non définie explicitement



Navigabilité bidirectionnelle entre A et B
Habituellement représenté sans flèches sur
les extrémités

Diagramme de classes: Formes d'association

a) Association binaire

Une association binaire est matérialisée par un lien entre deux classes associées.



Remarque: Il est possible d'avoir plusieurs associations entre deux classes.

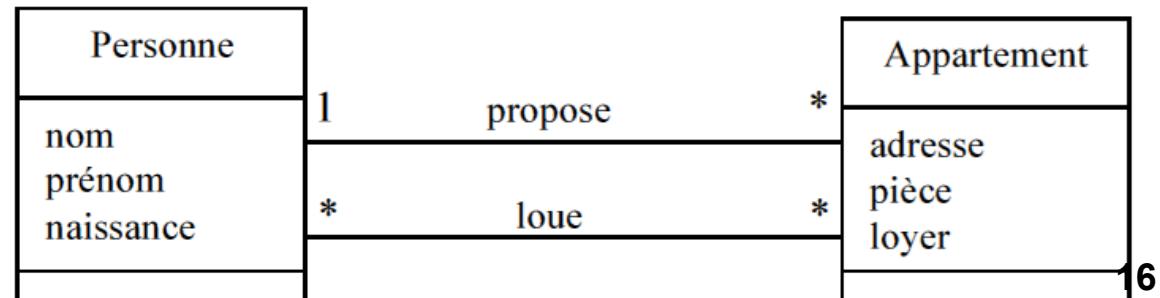


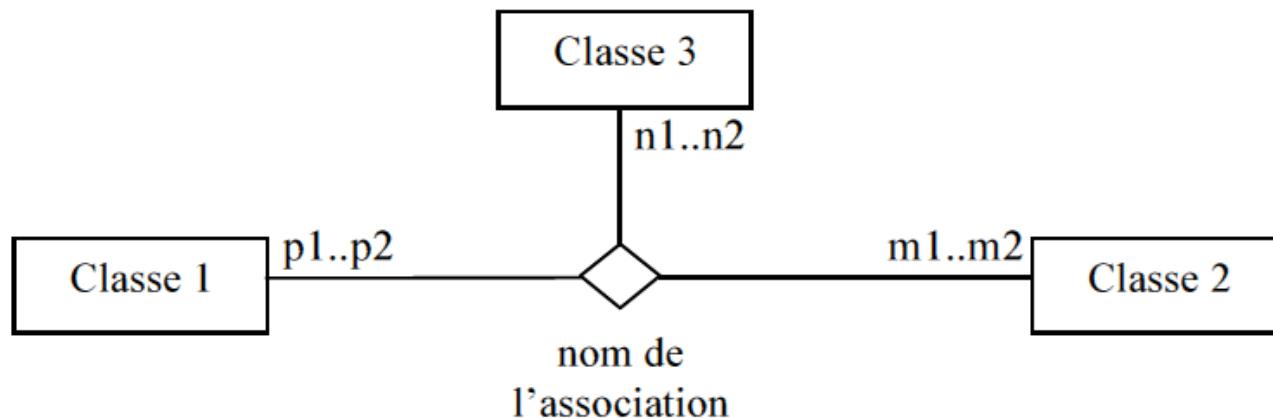
Diagramme de classes: Formes d'association

b) Association n-aire

- ❑ Une association n-aire relie **plus de deux** classes.
- ❑ Elle est représenté en utilisant **un losange** reliant **toutes les classes** en question. **Le nom de l'association**, le cas échéant, apparaît **à proximité du losange**.
- ❑ Dans une association n-aire, la multiplicité apparaissant sur le lien de chaque classe s'applique sur **une instance de chacune des classes à l'exception de la classe considérée**.

Diagramme de classes: Formes d'association

b) Association n-aire (suite)



Pour une association ternaire, les cardinalités se lisent de la façon suivante :

- Pour un couple d'instances de la classe 1 et de la classe 2, il y a au minimum $n1$ instances de la classe 3 et au maximum $n2$.

Diagramme de classes: Formes d'association

c) Association réflexive

- Une association est dite réflexive lorsque **le lien existe entre des objets de la même classe**.
- Deux personnes peuvent être amies (on considère que l'amitié est réciproque).

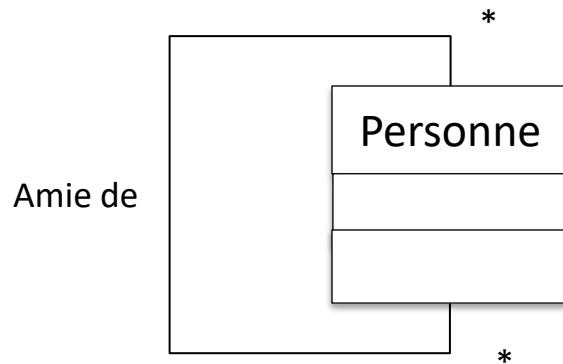


Diagramme de classes: Formes d'association

d) Classe d'association

- Une classe-association est une association qui est aussi une classe.
- Elle permet de décrire soit des attributs soit des opérations propres à l'association.
- Une classe-association est représenté par un trait en pointillé entre la classe et l'association qu'elle représente

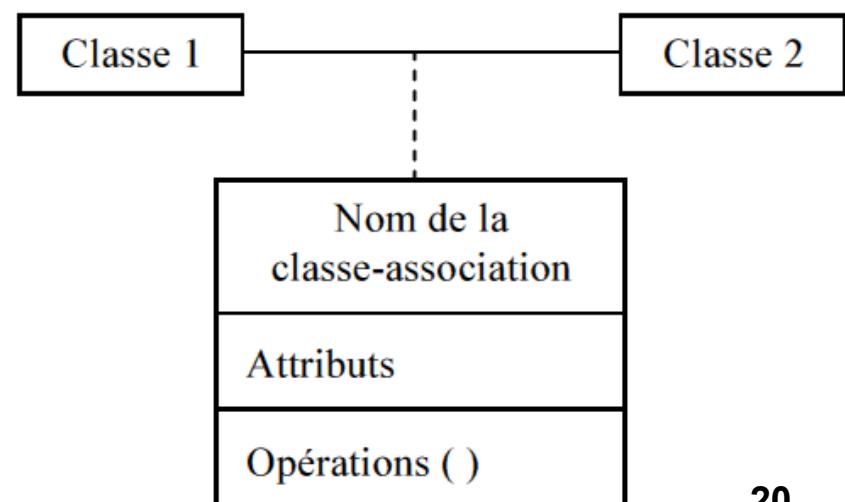


Diagramme de classes: Formes d'association

d) Classe d'association (suite)

- L'association "employer" entre une société et personne possède comme propriétés le salaire et la date d'embauche.
- En effet ces deux propriétés n'appartiennent ni à la société, qui peut employer plusieurs personnes, ni aux personnes qui peuvent avoir plusieurs emplois.

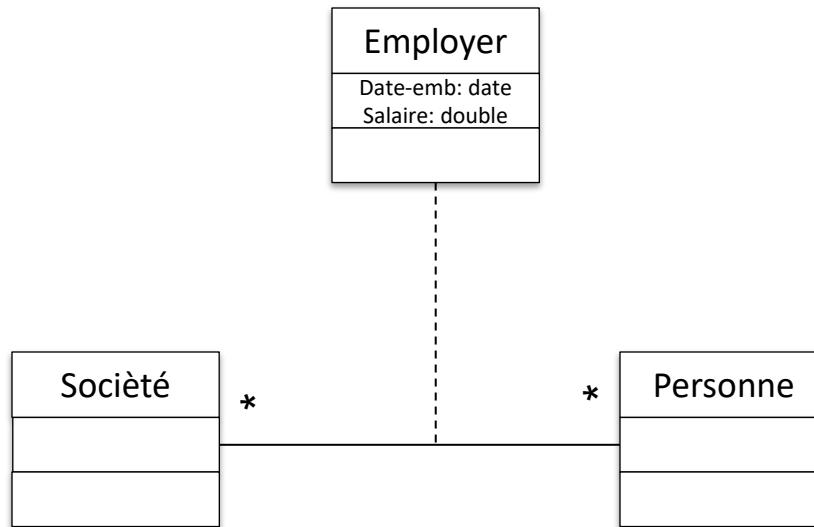


Diagramme de classes: Formes d'association

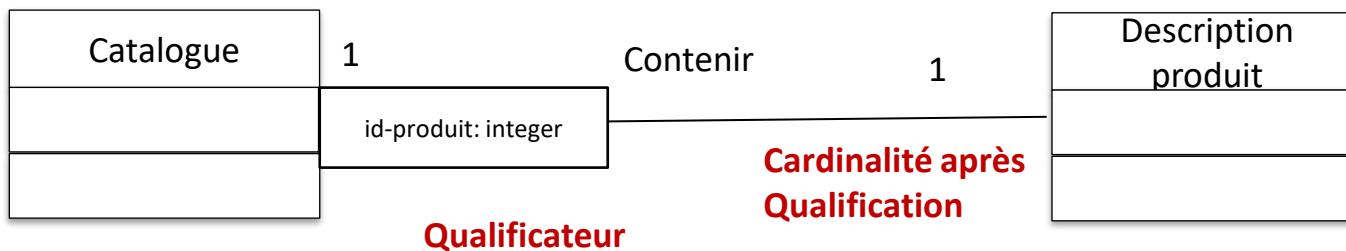
e) Association qualifiée

- Une association qualifiée permet de restreindre la cardinalité d'une association en ajoutant **un qualificateur** (aussi appelé clé ou index).
- Ce qualificateur est constitué d'un ou plusieurs attributs qui permettent de cibler un ou plusieurs objets en particulier.
- Le qualificateur est placé dans **un rectangle à l'extrémité de l'association** (extrémité opposée à la classe dont nous limitons la cardinalité).

Diagramme de classes: Formes d'association

e) Association qualifiée (suite)

- Un objet de la classe Catalogue est relié à un nombre indéterminé d'objet de la classe Description produit.
- Par contre, la classe Catalogue associée au qualificateur «Idproduit» n'est reliée qu'à un seul objet de la classe Description produit



Agrégation et composition entre classes

Agrégation

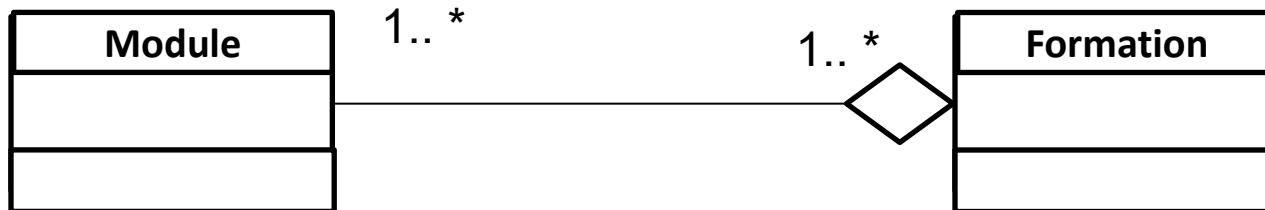
- Une agrégation est une association qui permet de représenter un lien de **type "ensemble"** comprenant **des "éléments"**.
- Il s'agit d'une relation **d'inclusion structurelle ou comportementale** d'un élément dans un ensemble.
- Une agrégation est noté graphiquement par un "**losange vide**" du côté de l'agrégat.



Agrégation et composition entre classes

Agrégation

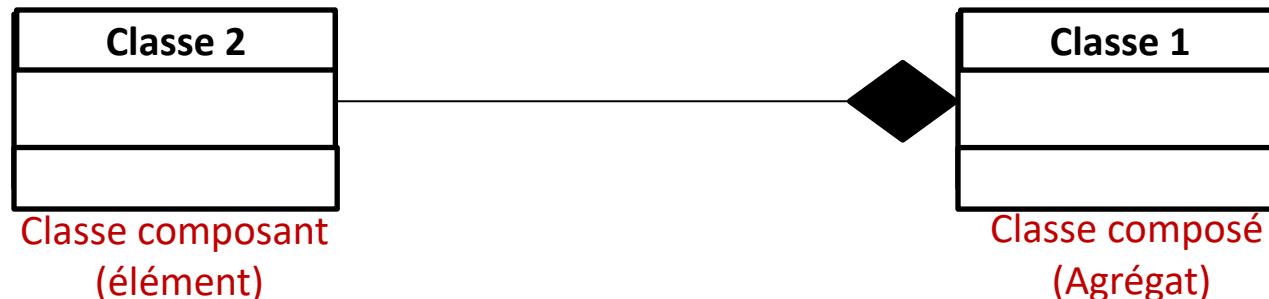
- Dans le cadre d'une formation, le cursus de cette dernière est une agrégation de modules à enseigner.
- À noter que la suppression d'une formation ne conduit pas automatiquement à la suppression des modules étant donné que ces derniers peuvent très bien être enseignés dans d'autres formations.



Agrégation et composition entre classes

Composition

- Une composition également appelée **agrégation composite** ou **agrégation forte**, décrit une relation d'agrégation dans laquelle il existe **une contrainte de durée de vie** entre la classe "composé" et la ou les classes "composant".
- C'est-à-dire la destruction de la classe "composé" implique la destruction de la ou les classes "composant".
- Une composition est noté graphiquement par un "**losange plein**" du côté de l'agrégat.



Agrégation et composition entre classes

Composition (suite)

- Soit l'exemple suivant, modélisant un appartement appartient à un immeuble.



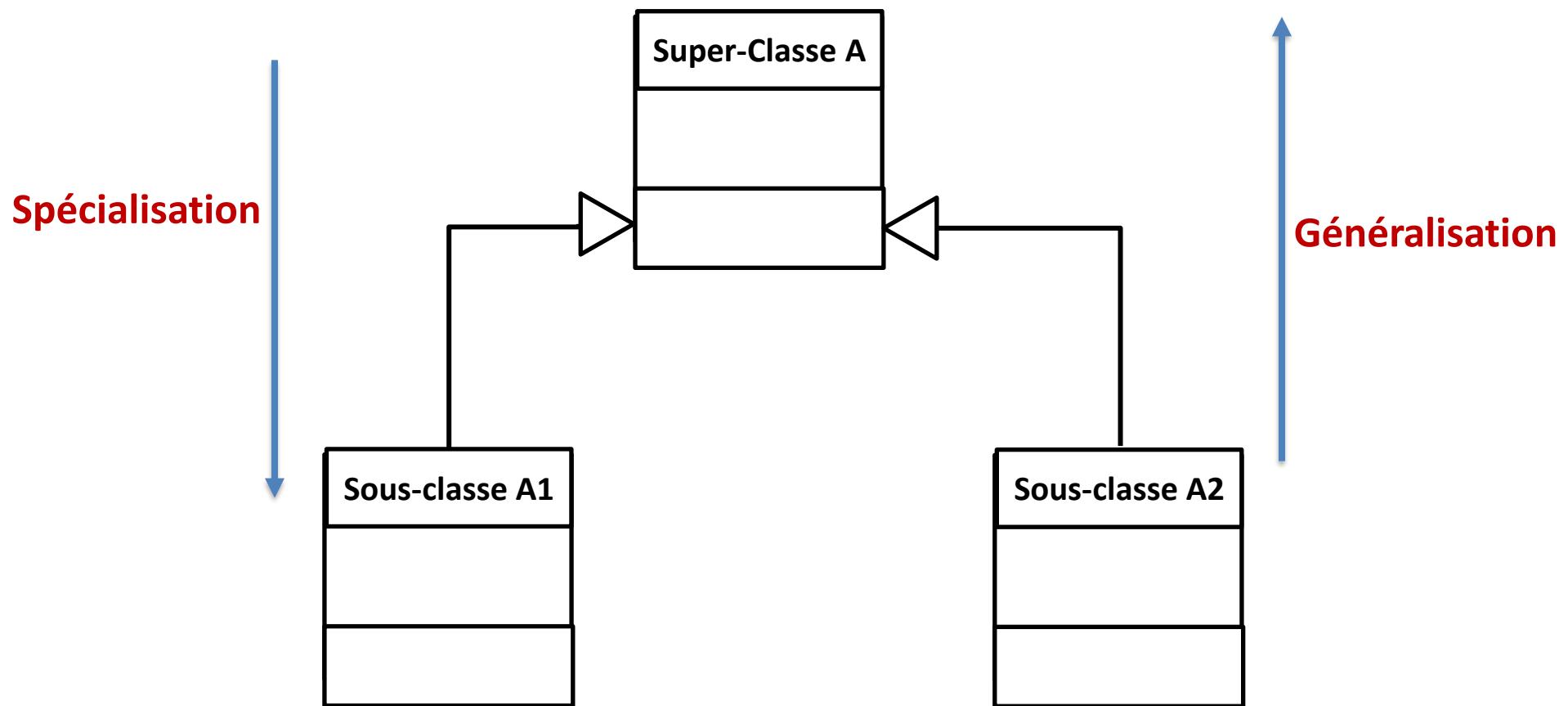
- Ici, il s'agit d'une agrégation forte (composition).
- La destruction de l'immeuble entraîne automatiquement la destruction des appartements associés.
- Le composé (composite) est la classe **Immeuble** et le composant est la classe **Appartement**.

Généralisation et spécialisation

- La généralisation décrit la **factorisation des éléments communs** (attributs et/ou opérations) d'un ensemble de classes dans une classe plus générale appelée super-classe (ou classe mère).
- La spécialisation représente le **phénomène inverse**, pouvoir **dériver à partir d'une classe** ou superclasse **des sous-classes** (ou classes filles) ayant des propriétés spécifiques les distinguant les une des autres.
- Dans le langage UML, ainsi que dans la plupart des langages objets, ces deux propriétés se traduisent par **le concept d'héritage**.
- Une relation d'héritage est noté graphiquement par **une flèche avec trait plein dont la pointe est un triangle fermé** désignant l'élément le plus général.

Généralisation et spécialisation

les sous-classes A1 et A2 héritent de la super-classe A.



Généralisation et spécialisation

Les propriétés principales de l'héritage sont :

- Une sous-classe possède **toutes les caractéristiques** de la super-classe, mais elle ne peut pas accéder aux caractéristiques privées de cette dernière ;
- Une sous-classe peut **redéfinir une ou plusieurs opérations de la super-classe** ;
- Toutes les associations de la classe mère s'appliquent aux classes dérivées ;
- Une instance d'une classe peut être utilisée partout où une instance de sa classe mère est attendue ;
- Une classe peut avoir plusieurs parents ;
- La relation d'héritage n'est pas propre aux classes. Elle s'applique à d'autres éléments du langage comme les acteurs, les cas d'utilisation ;
- La relation d'héritage est une relation non-réflexive, non-symétrique.

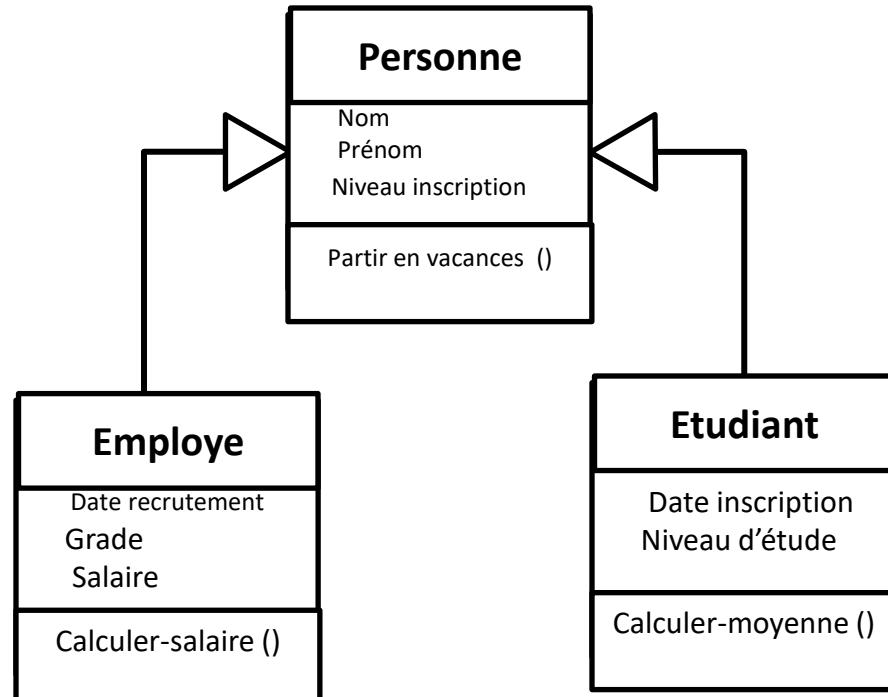
Héritage

a) Héritage simple

- L'héritage est simple lorsque **une classe hérite d'une seule super-classe**.
- Les classes « Employé » et « Étudiant » sont des dérivés de la classe « Personne » et héritent des propriétés de cette dernière.

La classe « Employé » se distingue par d'autres attributs tels que la date de recrutement, le grade et le salaire

La classe « Étudiant » possède une date d'inscription, un niveau d'étude et une moyenne annuelle.

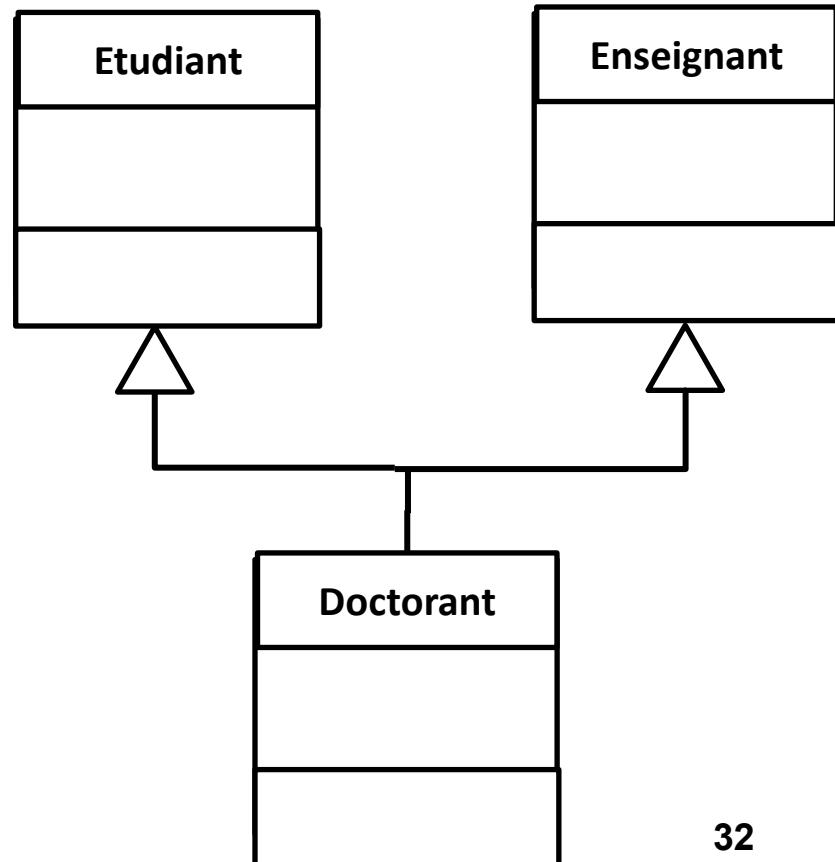


Héritage

b) Héritage multiple

- L'héritage est multiple quand une même classe hérite **de deux ou plusieurs super-classes.**

- Un doctorant est à la fois un étudiant et un enseignant. Le doctorant hérite à la fois des propriétés des enseignants et de celles des étudiants



Abstraction

- Parfois une classe ne peut exister que par l'existence de ses classes descendantes.
- Une telle classe est dite classe abstraite qui n'a pas d'instance directe, mais dont les classes descendantes ont des instances.
- Dans une relation d'héritage, la super-classe est par **définition une classe abstraite**.
- Une classe abstraite peut être représentée soit en mettant **son nom en italique, soit par rajout du stéréotype « abstract »**

Polymorphisme

- ❑ Il est possible de **redéfinir une opération** dans des classes héritant d'une classe de base.
- ❑ Il est alors possible d'appeler la méthode d'un objet sans se soucier de son type intrinsèque : il s'agit du **polymorphisme** .
- ❑ C'est un mécanisme permettant à un même nom de méthode d'avoir plusieurs réalisations possibles
- ❑ Permet de faire abstraction des détails des classes spécialisées d'une famille d'objet, en les masquant par une interface commune (qui est la classe de base).
- ❑ Nous distinguons deux denominations:
 - ❑ le polymorphisme statique (la surcharge de méthode)
 - ❑ le polymorphisme dynamique (la redéfinition de méthode ou encore la surcharge héritée)

Polymorphisme statique

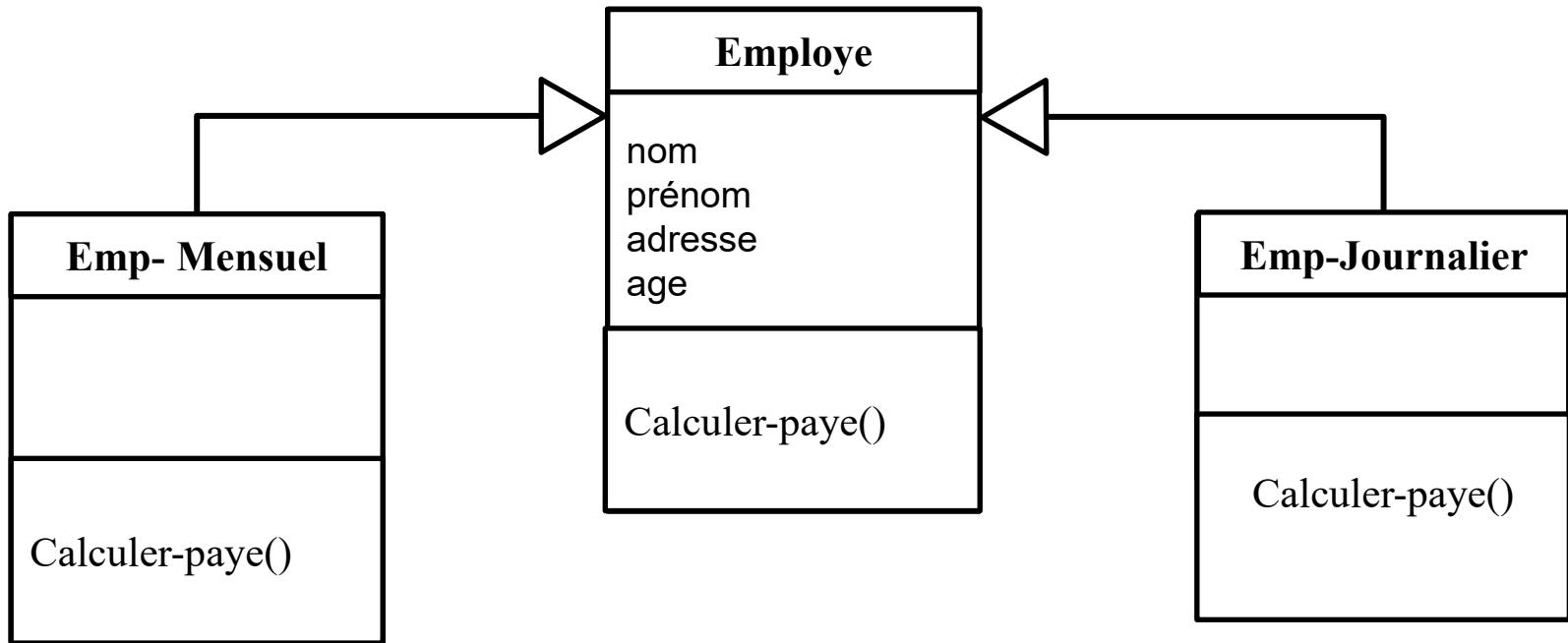
- La surcharge de méthode (*polymorphisme statique de méthode*) est une fonctionnalité classique des langages orientés objet;
- Elle consiste dans le fait qu'une classe peut disposer de **plusieurs méthodes ayant le même nom**, mais avec des paramètres formels différents ou éventuellement un type de retour différent.
- On dit alors que ces méthodes n'ont pas la même signature.

Remarque: On appelle **signature** d'une méthode l'**en-tête** de la méthode avec ses **paramètres formels** et leur **type**.

Polymorphisme dynamique

- La redéfinition de méthode (ou **polymorphisme dynamique**) est spécifique aux langages orientés objet.
- Elle est mise en œuvre lors de l'héritage d'une classe mère vers une classe fille dans le cas d'une méthode ayant la même signature dans les deux classes.
- Dans ce cas les actions dues à l'appel de la méthode, dépendent du code inhérent à chaque version de la méthode (celle de la classe mère, ou bien celle de la classe fille).

Polymorphisme dynamique



Si l'on reprend l'exemple de l'héritage simple, l'opération «`Calculer-paye()` » dont héritent les classes «**Emp- Mensuel**» et «**Emp-Journalier**» pourrait avoir des implémentations différentes pour ces deux sous-classes. D'où, il serait préférable d'implémenter cette opération par deux méthodes différentes : une pour la sous-classe «**Emp- Mensuel** » et une pour la sous-classe «**Emp-Journalier**».

Exercice d'application

Une entreprise nationale de vente d'appareil électroménager souhaite réaliser une première expérience d'analyse objet avec le langage UML sur un petit sous ensemble de son SI. Ce sous-ensemble concerne le suivi des personnels des agences locales implantées dans les régions. Chaque région est pilotée par une direction régionale qui a en charge un certain nombre d'agences locales. Une direction régionale est caractérisée par un code et un libellé. Chaque agence est caractérisée par un code, un intitulé, une date de création et une date de fermeture.

À une agence sont rattachées une à plusieurs personnes. Chaque personne est caractérisée par les données : numéro, qualité (M., Mme, Mlle), nom, prénom, date de naissance, date prévisionnelle d'arrivée, date d'arrivée et date de départ. Il est demandé d'élaborer le diagramme de classe de ce premier sous-ensemble du SI de cette entreprise.

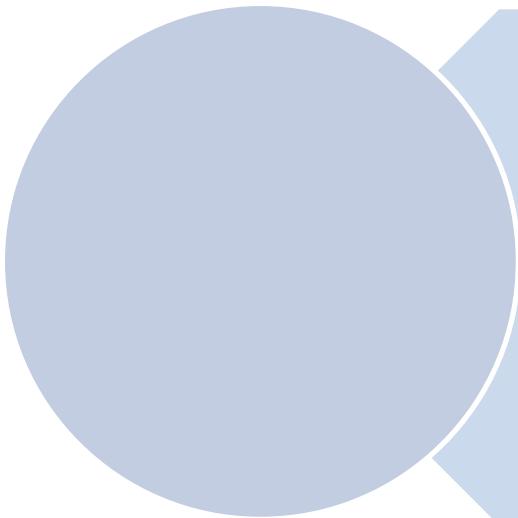


Diagramme d'objets

Diagramme d'objets

- Le diagramme d'objets fait parties des diagrammes structurels (statique).
- Il représente les objets d'un système (c.a.d. les instances des classes) et leurs liens (c.a.d. les instances des associations) à **un instant donné**.
- Il donne **une vue figée** du système à un moment précis.
- A un diagramme de classe correspond **une infinité de diagrammes d'objets**.
- Nous nous servirons du diagramme d'objet **pour donner des exemples, des cas de figure**, qui permettront d'affiner le diagramme de classe et de mieux le comprendre.

Diagramme d'objets

Représentation graphique

- Chaque objet est représenté dans un **rectangle** dans lequel **figure le nom de l'objet (souligné)** et éventuellement la valeur d'un ou **plusieurs** de ses **attributs**.
- Comme pour la représentation d'une classe, la représentation d'un objet pourra être plus ou moins détaillée.

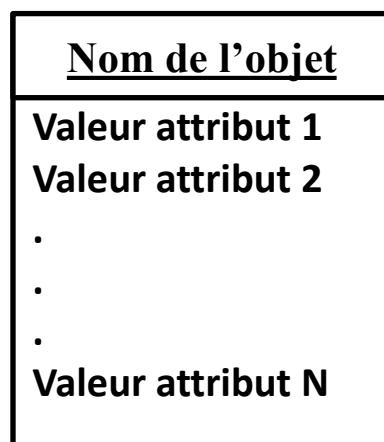


Diagramme d'objets

Représentation graphique (suite)

- Le nom d'un objet peut être désigné sous trois formes :
 1. nom de l'objet, désignation directe et explicite d'un objet ;
 2. nom de l'objet : nom de la classe, désignation incluant le nom de la classe ;
 3. nom de la classe, désignation anonyme d'un objet d'une classe donnée.
- Pour la classe Voiture, nous avons les exemples de représentation d'objets

Voiture
Type
Nom
Année



<u>mavoiture</u>
Seat
Arona
2019

Mavoiture: Voiture
Seat
Arona
2019

:Voiture
Seat
Arona
2019

Classe voiture

Représentation UML d'un objet

Diagramme d'objets

Exemple

Dans le diagramme de classes ci-dessous, nous représentons un lien entre deux classes Commande et Client.

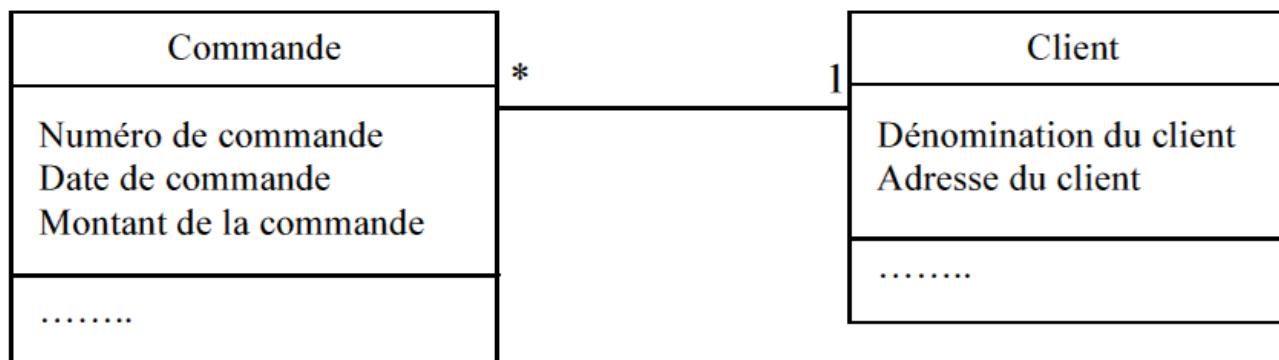
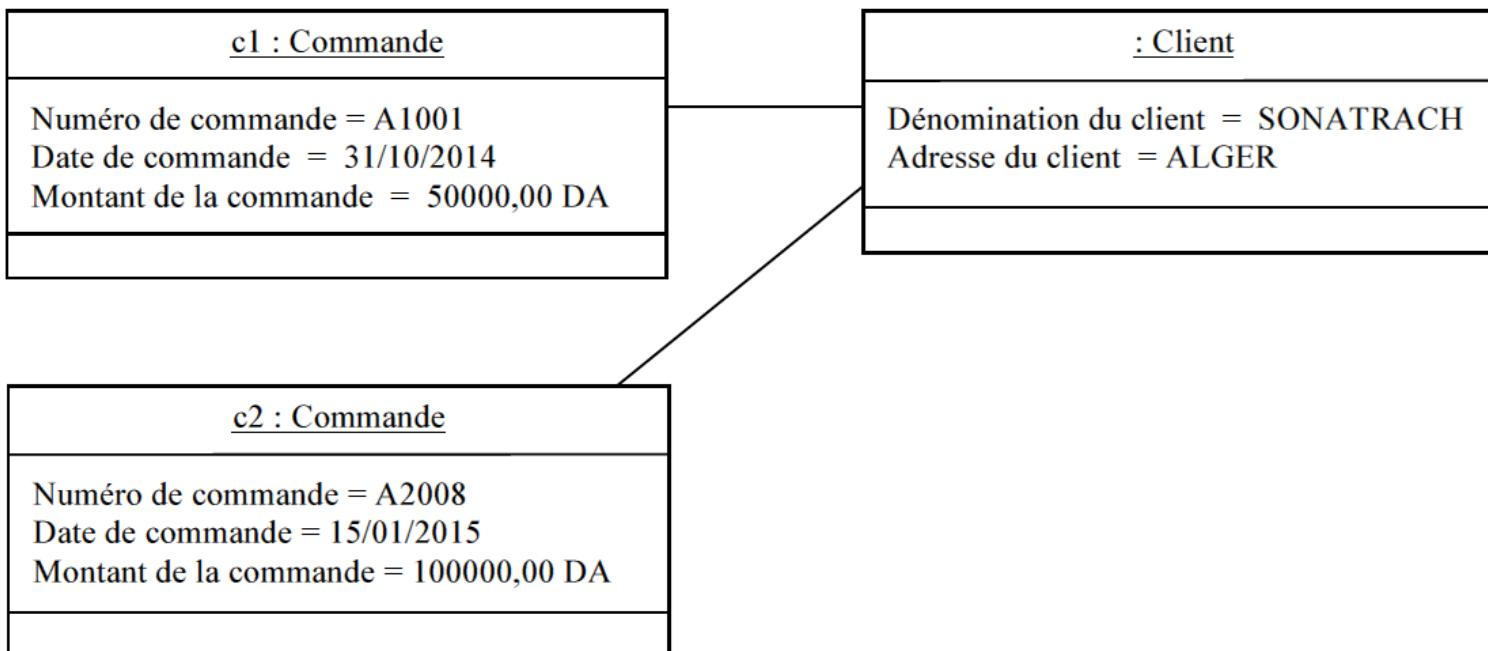


Diagramme d'objets

Exemple (suite)

Si nous nous intéressons à un client en particulier pour voir les commandes qu'il a effectué, nous représentons ceci par un diagramme d'objets donné comme suit :



Fin du Chapitre 4.

