

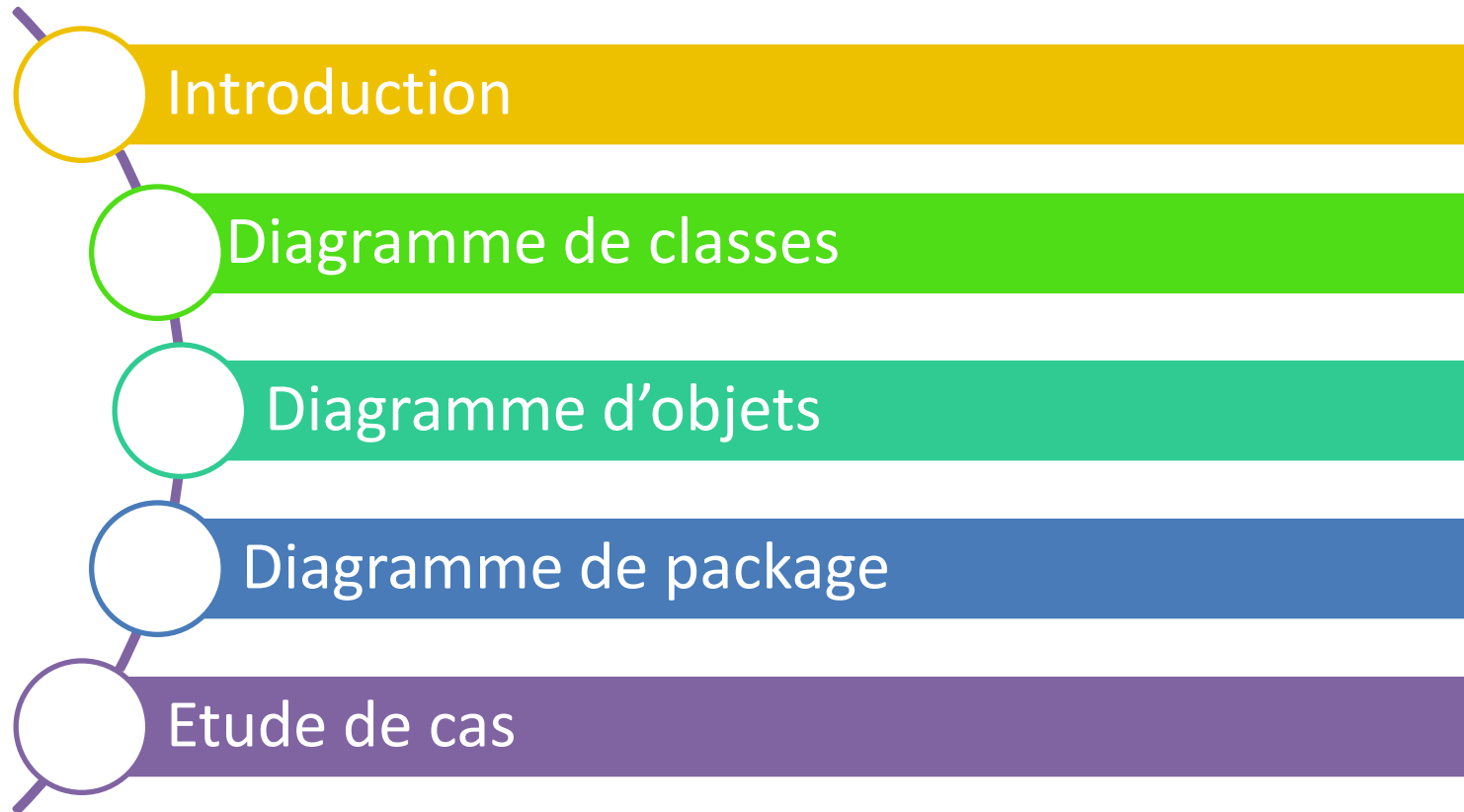


Chapitre 3:Analyse statique

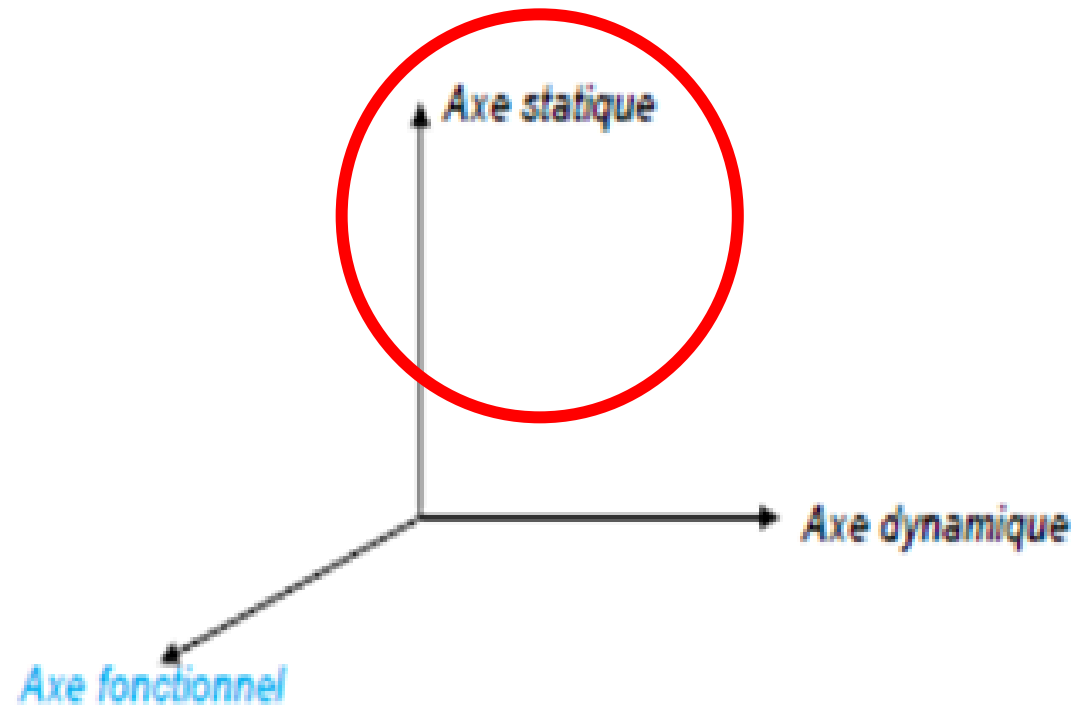
Module Langage de modélisation UML

**Année Universitaire
2016-2017**

PLAN



Objectif



SECTION 1 : DIAGRAMME DE CLASSES

Présentation du Diagramme de Classes

- ❖ Il peut être utilisé en phases d'analyse et de conception :

Analyse : description préliminaire de la structure des entités et du nom de quelques attributs et méthodes.

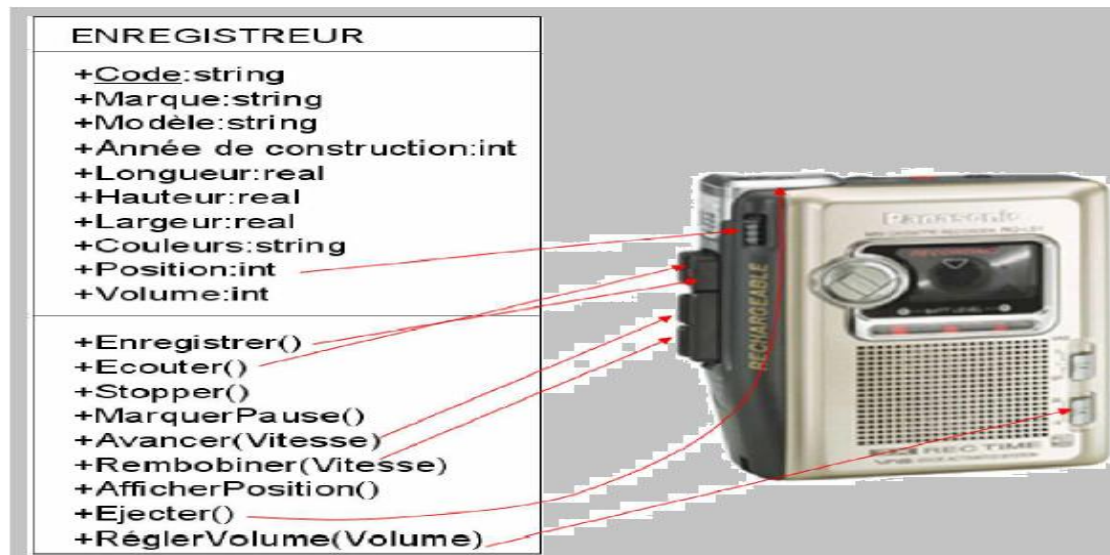
Conception : description détaillée de l'ensemble des classes. Les attributs et méthodes doivent apparaître de façon détaillée avec tous les types de paramètres et les types de retour.

- ❖ Représentation de la structure statique du système.
- ❖ Ensemble de classes et de relations entre elles (associations, généralisations, etc.).

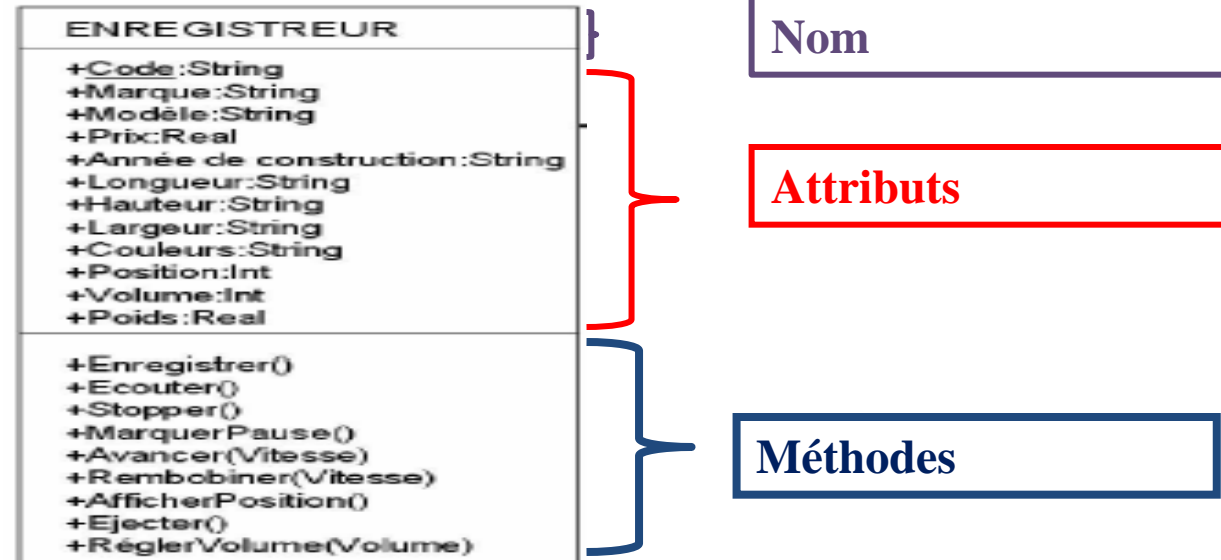
Classe: Définition

La classe est un modèle décrivant les caractéristiques communes et le comportement d'un ensemble d'objets

Une classe est formée par un ensemble d'attributs et d'opérations



Classe: Représentation Graphique



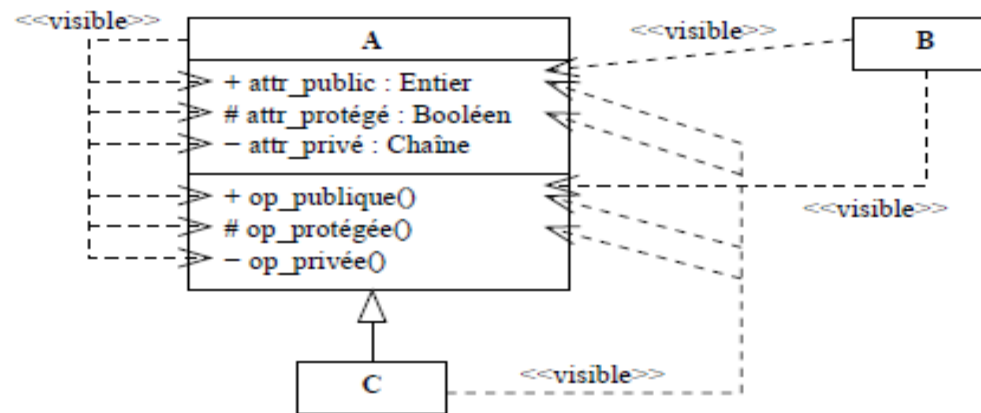
Classe: Déclarations

- Le nom de l'attribut doit être unique dans la classe. La syntaxe de la déclaration d'un attribut est la suivante: **<visibilité> [/]**
<nom_attribut> : <type> ['['<multiplicité>']
[{<contrainte> }] [= <valeur_par_défaut>]
- La déclaration d'une opération contient les types des paramètres et le type de la valeur de retour, sa syntaxe est la suivante : **<visibilité><nom_méthode>**
([<paramètre_1>, ... , <paramètre_N>]) :
[<type_renvoyé>] [{<propriétés> }]

Classe: VISIBILITE

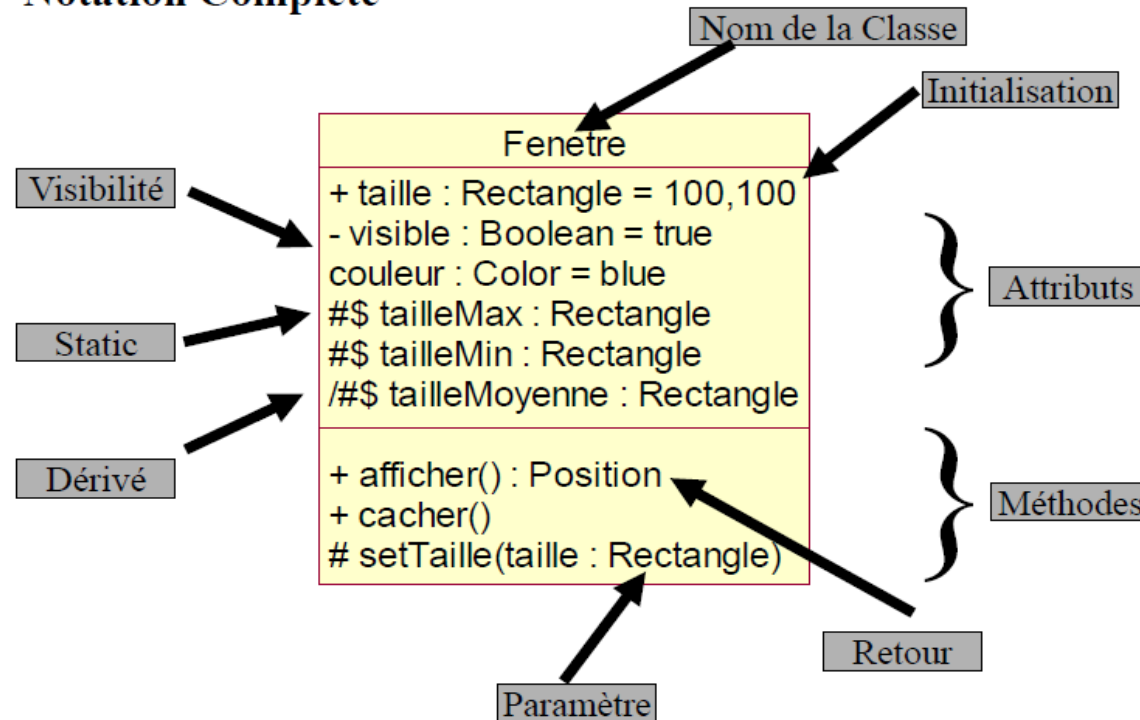
La visibilité concerne les attributs et les méthodes

- Public (+) : visible pour toutes les classes.
- Privé (-) : n'est visible que pour la classe elle-même.
- Protégé (#) : visible pour la classe et toutes les sous-classes.



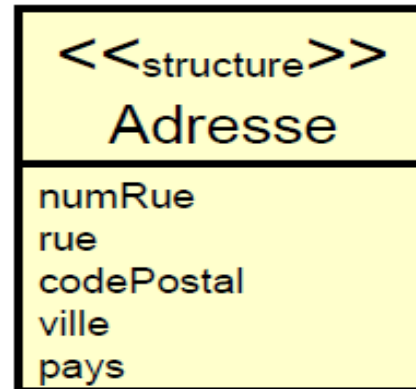
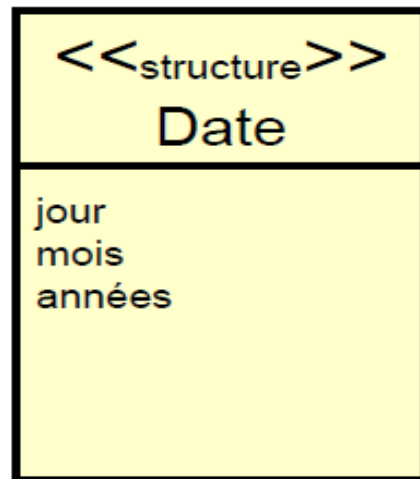
Classe: Notation complète

Notation Complète



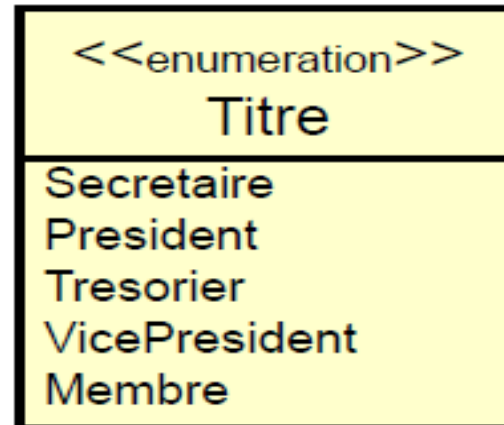
Classe: Structure

- **Définition** : Ensemble d'attributs pouvant être regroupés.



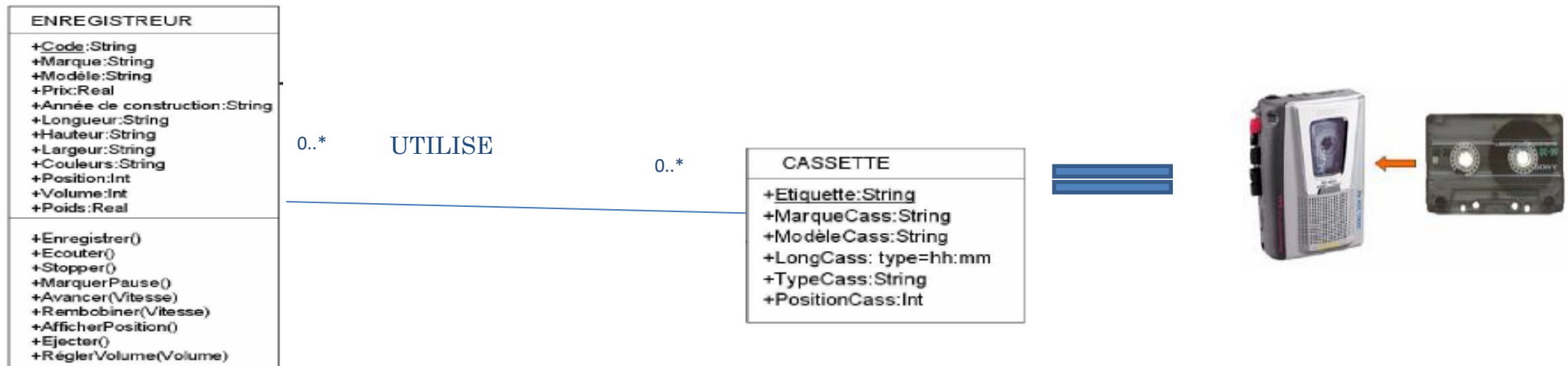
Classe: Énumération

- **Définition** : Ensemble de littéraux d'énumération, ordonné.



Association: Définition

- Une association est une relation entre deux classes qui décrit les connexions structurelles entre leurs instances (objets).



Une association peut être nommée(facultatif)

Association: Multiplicité

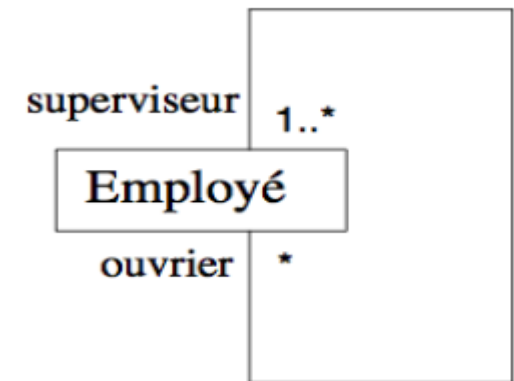
Multiplicité:

- Aux deux extrémités d'une association, on doit faire figurer une indication de multiplicité: (q1..q2)
- Elle spécifie sous la forme d'un intervalle d'entiers positifs ou nuls le nombre d'objets qui peuvent participer à une relation avec un objet de l'autre classe, dans le cadre d'une association.

Multiplicity	Option
0..0	0
0..1	
1..1	1
0..*	*
1..*	
5..5	5
m..n	

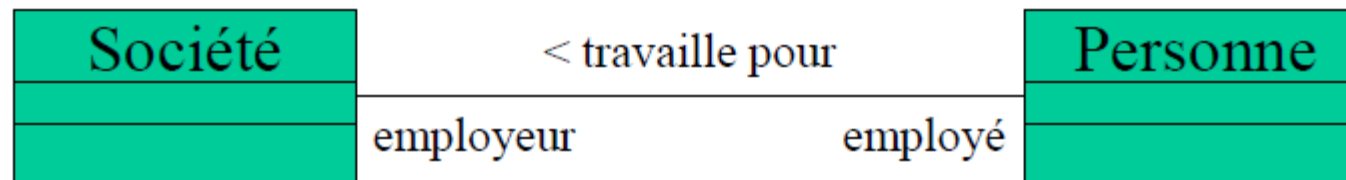
Association: Association réflexive

- Une association est dite réflexive si elle s'applique a des objets d'une même classe.



Association: Rôle

- **Rôles** : spécifie la fonction d'une classe pour
- une association donnée (indispensable pour les associations réflexives)
- Le rôle définit la manière dont une classe intervient dans une relation
- L'intérêt des rôles dans le cas où plusieurs associations lient deux classes : distinction des concepts attachés aux associations

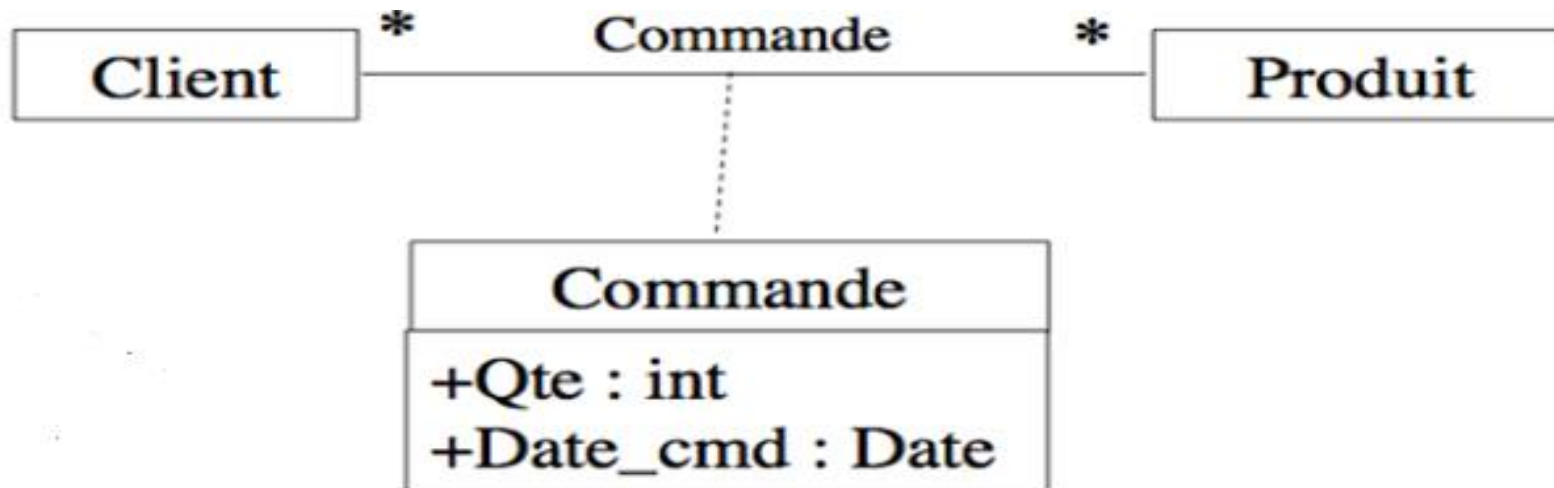


Association: Classe d'association

Une association qui contient des attributs et qui ne participe pas à des relations avec d'autres classe est appelée **classe d'association**

Il s'agit d'une classe qui réalise la navigation entre les instances d'autres classes.

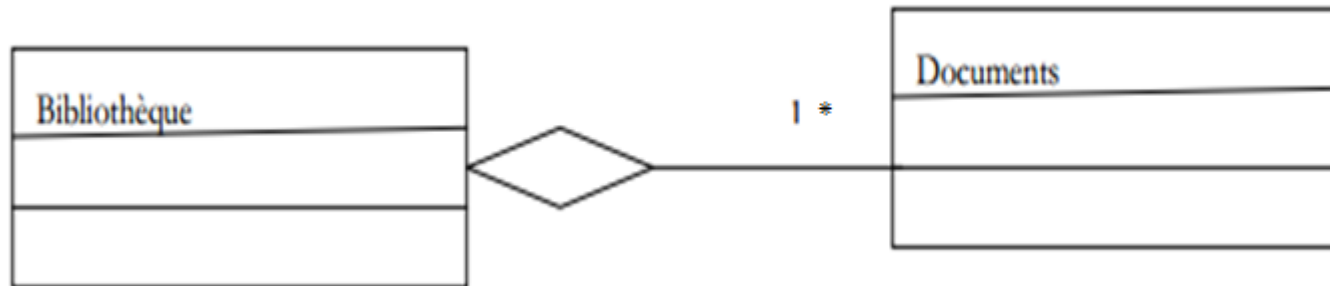
Elle sert connecter les classes entre elles



Association : Agrégation

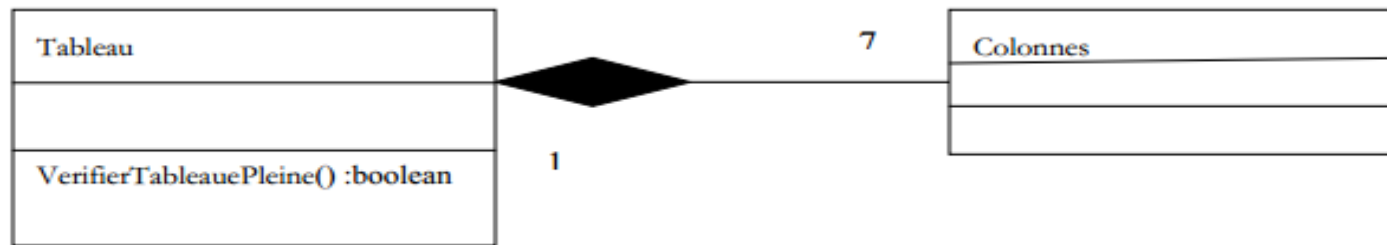
C'est un type particulier d'association exprimant une relation de contenance.

Elle définit la relation «partie de »



Association : Composition

Une composition est une agrégation forte, elle est représentée par le symbole



Différences entre une agrégation et une composition :

Pour la relation de composition :

- La destruction de l'objet composite implique la destruction de ses composants.
- La multiplicité du côté composite ne doit pas être supérieure à 1 (i.e. 1 ou 0..1).
- La composition est une agrégation non partagée.

HERITAGE

L'héritage : mécanisme basé sur la généralisation et qui permet aux sous-classes d'hériter, c'est à dire d'avoir les mêmes attributs, opérations et associations que la super-classe.

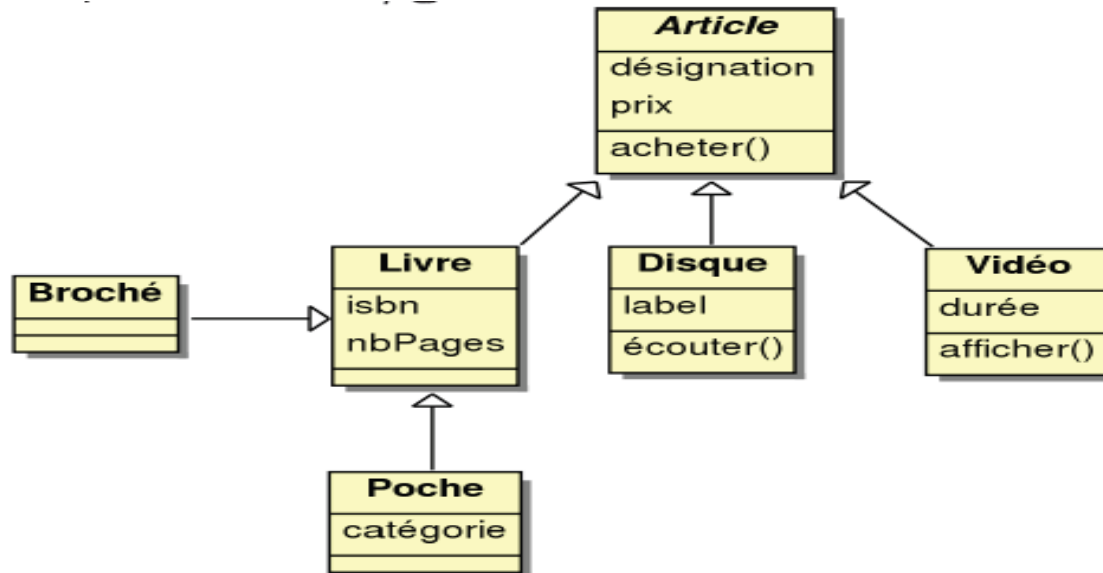


DIAGRAMME DE PACQUAGE

Pacquette : Définition

- Un package (ou paquetage) est un regroupement d'éléments.
- Le regroupement peut être pour des raisons sémantiques, d'usage, de lieux, etc.
- Un paquetage est susceptible de contenir n'importe quel élément de modélisation UML



SECTION 3: DIAGRAMME D'OBJETS

Diagramme d'objets: Présentation

Un diagramme de d'objets :

- Représente les liens structurels entre instances de classes
- Facilite la compréhension de structures complexes
- Trois représentations possibles des instances

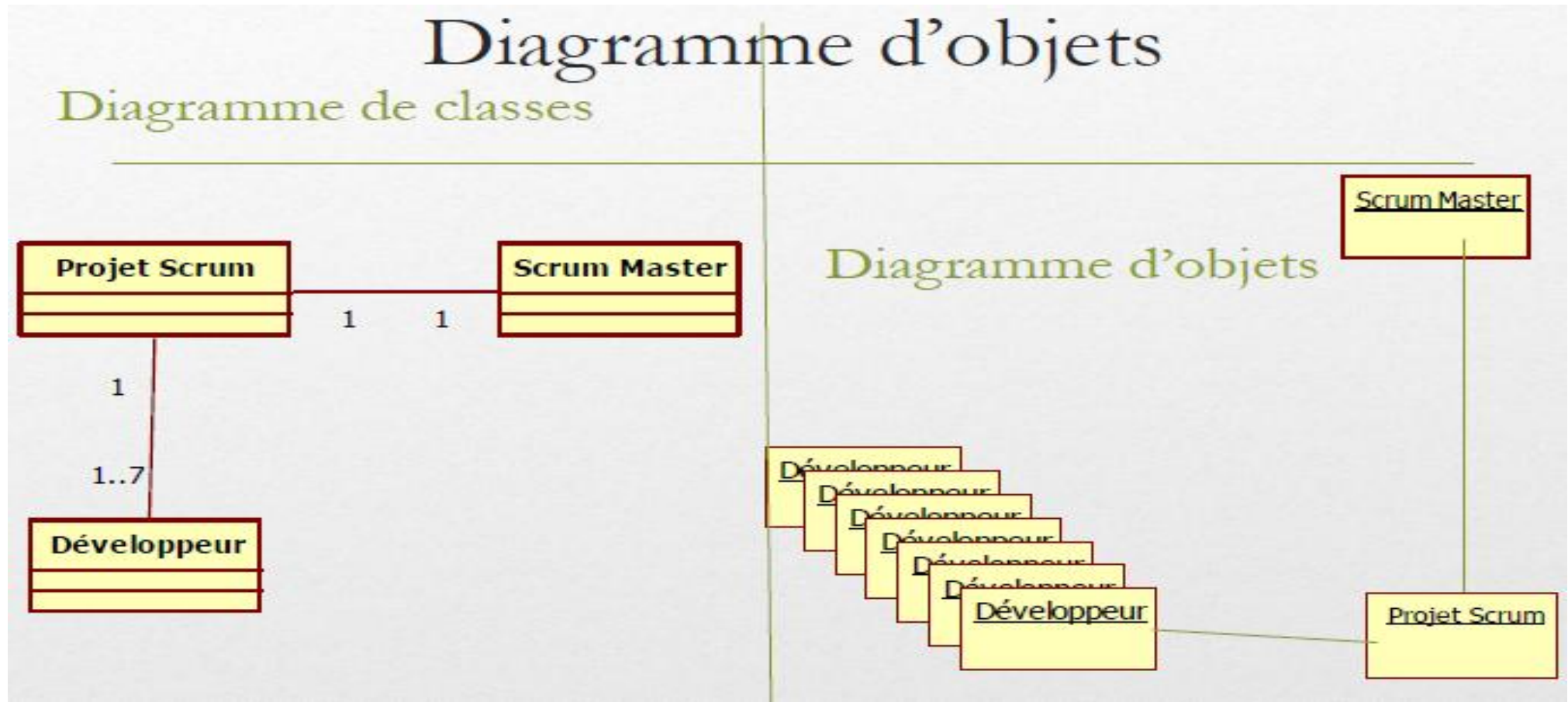
Diagramme d'objets

- Un *diagramme d'objets* présente un ensemble d'objets avec les liens qui les associent
 - Un *objet* est une instance d'une classe
 - Les objets représentent les concepts ou entités du monde réel
- Exemple:



- Un *lien* est une instance d'une association.
- Les liens représentent les relations entre concepts ou entités du monde réel.

Diagramme d'objets



D. ETUDE DE CAS (1/17)

- Cette étude de cas concerne un système simplifié de réservation de vols pour une agence de voyage
 1. Des compagnies aériennes proposent différents vols
 2. Un vol est ouvert à la réservation et refermé sur ordre de la compagnie
 3. Un client peut réserver un ou plusieurs vols, pour des passagers différents
 4. Une réservation concerne un seul vol et un seul passager
 5. Une réservation peut être annulée ou confirmée
 6. Un vol a un aéroport de départ et un aéroport d'arrivée
 7. Un vol a un jour et une heure de départ et un jour et une heure d'arrivée
 8. Un vol peut comporter des escales dans des aéroports
 9. Une escale a une heure d'arrivée et une heure de départ
 10. Chaque aéroport dessert une ou plusieurs villes

D. ETUDE DE CAS (2/17)

- On procède phrase par phrase
 - Des compagnies aériennes proposent différents vols



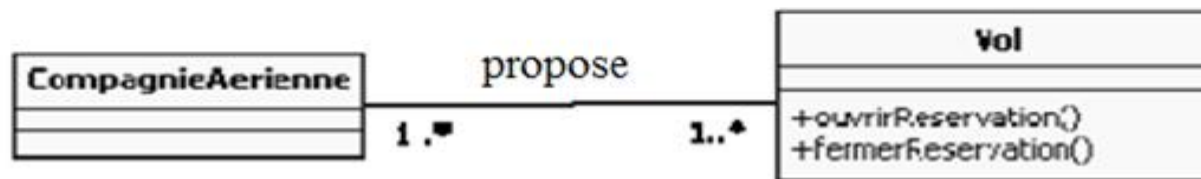
- La multiplicité **1..*** du cote de la classe **Vol** a été préféré à une multiplicité **0..*** car l'agence s'associe aux compagnies aériennes qui proposent au moins un vol
- Pas d'indication sur la multiplicité du coté de la classe **Compagnie Aérienne**
 - Question à poser à l'expert



D. ETUDE DE CAS (3/17)

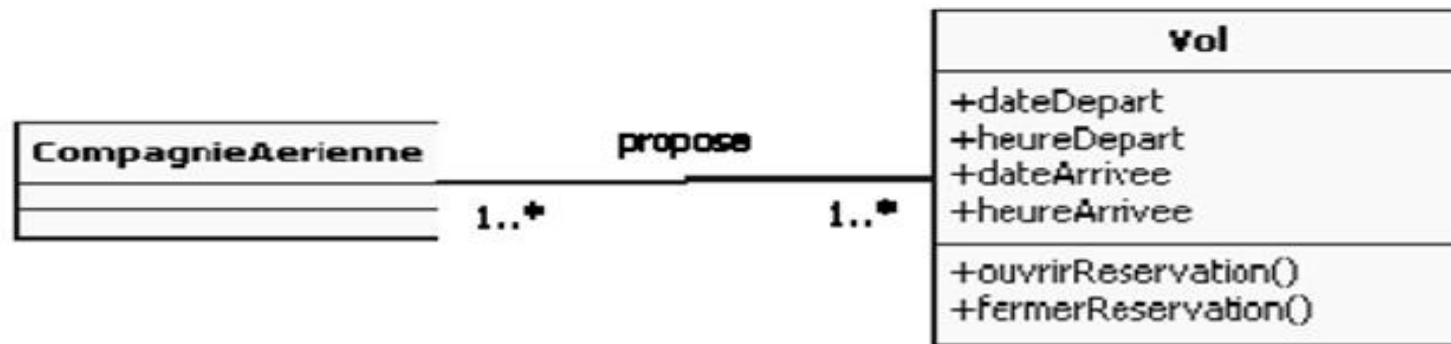
2. Un vol est ouvert à la réservation et refermé sur ordre de la compagnie

- Les notions d'ouverture et de fermeture de la réservation représentent des concepts dynamiques
- Il s'agit de changement d'état d'un objet Vol sur ordre d'un objet CompagnieAerienne



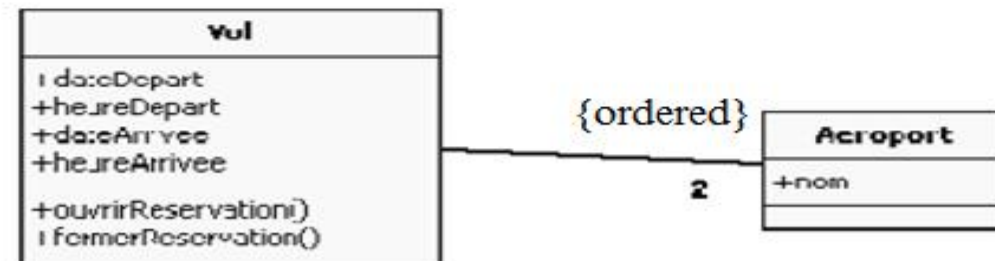
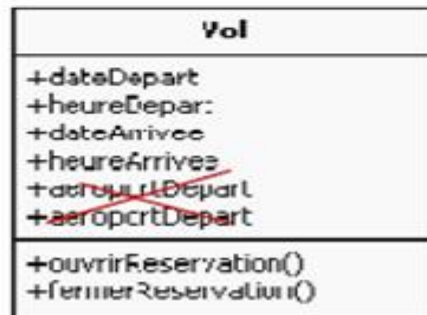
D. ETUDE DE CAS (4/17)

3. Un vol a un jour et une heure de départ et un jour et une heure d'arrivée
- Toutes ces notions de dates et d'heures représentent simplement des valeurs pures. Nous les modélisons donc comme des attributs non pas comme des objets



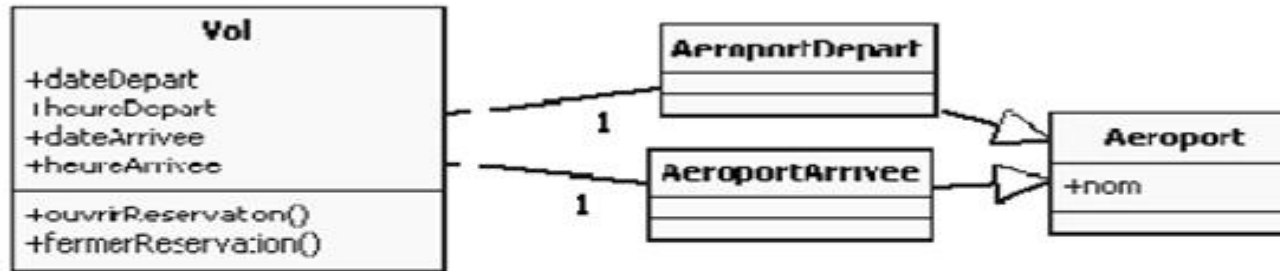
D. ETUDE DE CAS (5/17)

4. Un vol a un aéroport de départ et un aéroport d'arrivée
- Sol1: Contrairement aux notions d'heure et de date qui sont des types simples, la notion d'aéroport est complexe
 - Un aéroport a un nom, dessert une ville, a une capacité, etc.
 - C'est la raison pour laquelle, il faudrait créer une classe Aéroport plutôt que de simples attributs aéroportDépart et aéroportArrivée
 - Cette sol1 consiste à créer une association avec une multiplicité 2 du côté de classe Aéroport en perdant les notions de départ et d'arrivée
 - Ajoutez la contrainte {ordered} pour indiquer que les 2 aéroports liés au vol sont ordonnées (l'arrivée est toujours après le départ)



D. ETUDE DE CAS (6/17)

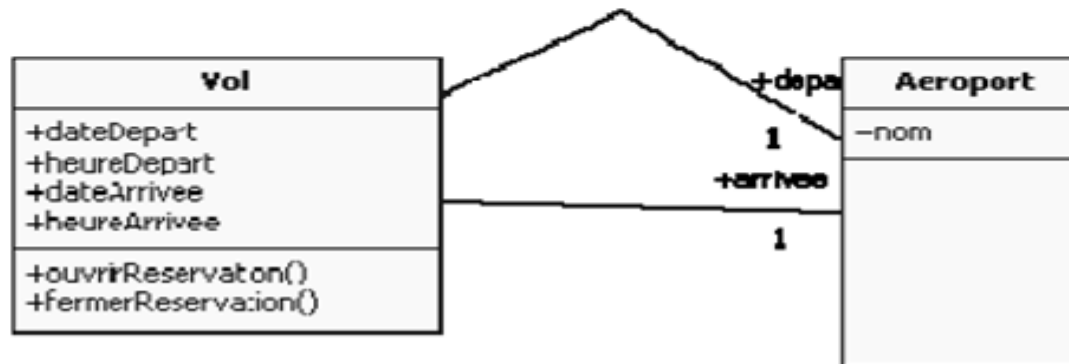
5. Un vol a un aéroport de départ et un aéroport d'arrivée
- Sol2: Création de deux sous classes de la classe Aéroport



- Cette solution est incorrecte
 - Tout aéroport est successivement aéroport de départ pour certains vols et aéroport d'arrivée pour d'autres
 - Les classes **AeroportArrivee** et **AeroportDepart** ont donc exactement les mêmes instances redondantes, ce que devrait décourager d'en faire deux classes distinctes

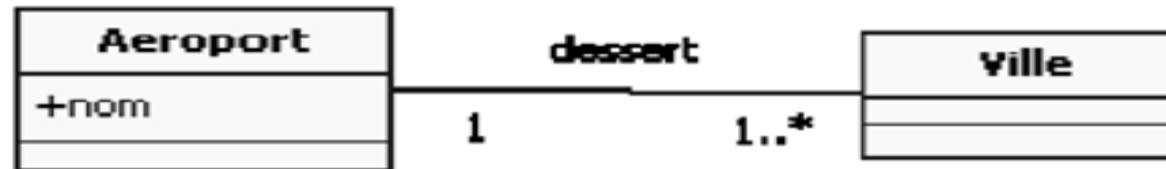
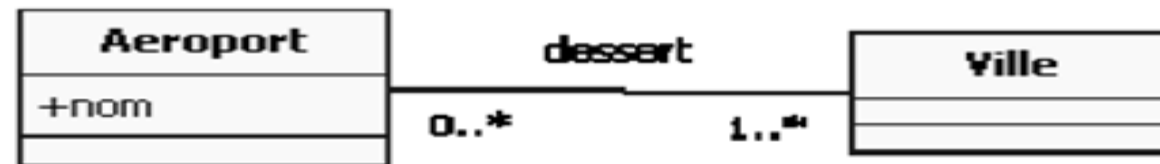
D. ETUDE DE CAS (7/17)

6. Un vol a un aéroport de départ et un aéroport d'arrivée
- Sol3: Création de deux associations entre les classes Vol et aéroport, chacune affectée d'un rôle différent avec une multiplicité égale à exactement 1



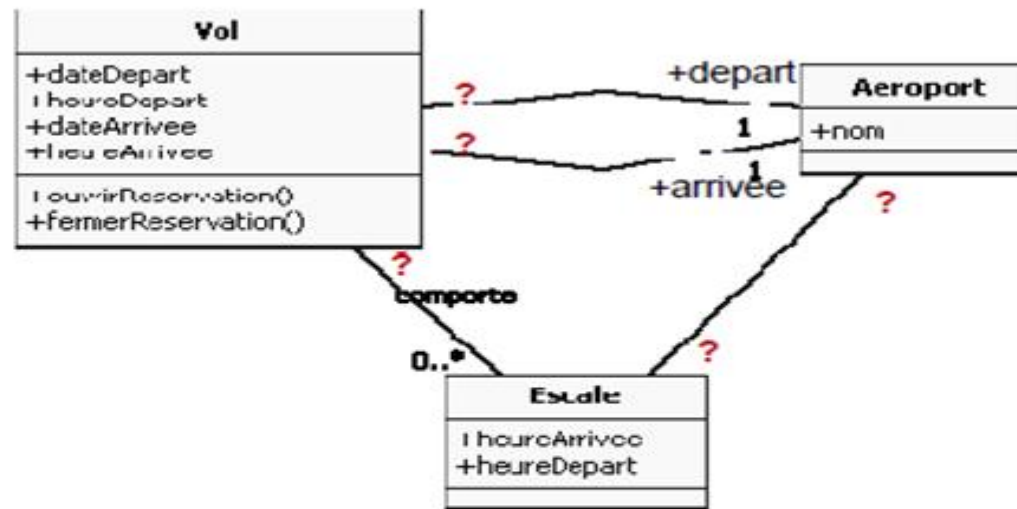
D. ETUDE DE CAS (8/17)

7. Chaque aéroport dessert une ou plusieurs villes



D. ETUDE DE CAS (9/17)

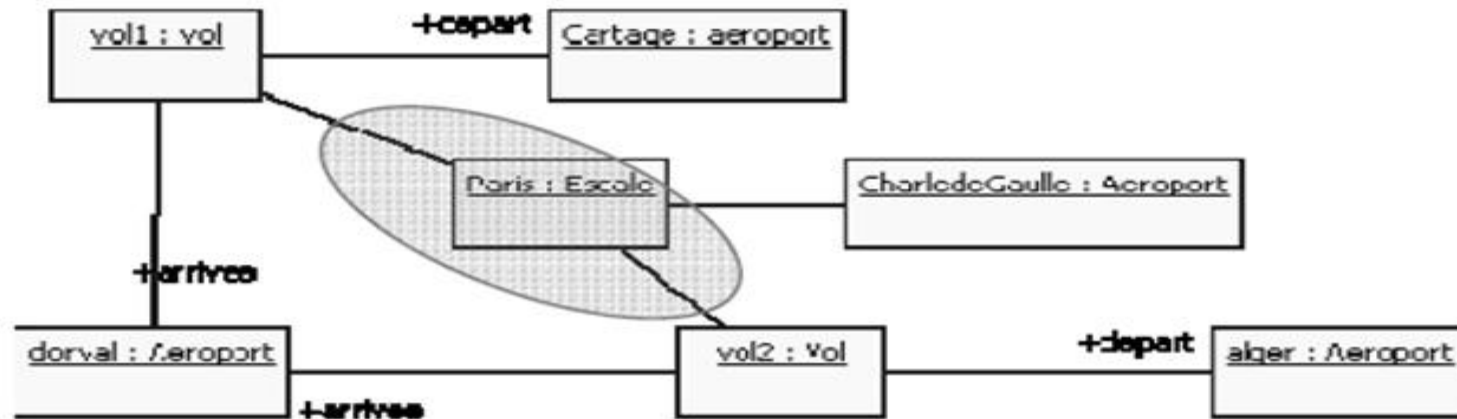
8. Un vol peut comporter des escales dans des aéroports
9. Une escale a une heure d'arrivée et une heure de départ



- La phrase 8 est ambiguë et ne peut nous donner de façon précise une information sur les multiplicités: Nécessité d'expert métier
 - Une escale a un lieu dans un et un seul aéroport
 - Un aéroport peut servir à plusieurs escales
 - Un aéroport peut servir de départ ou d'arrivée à plusieurs vols

D. ETUDE DE CAS (10/17)

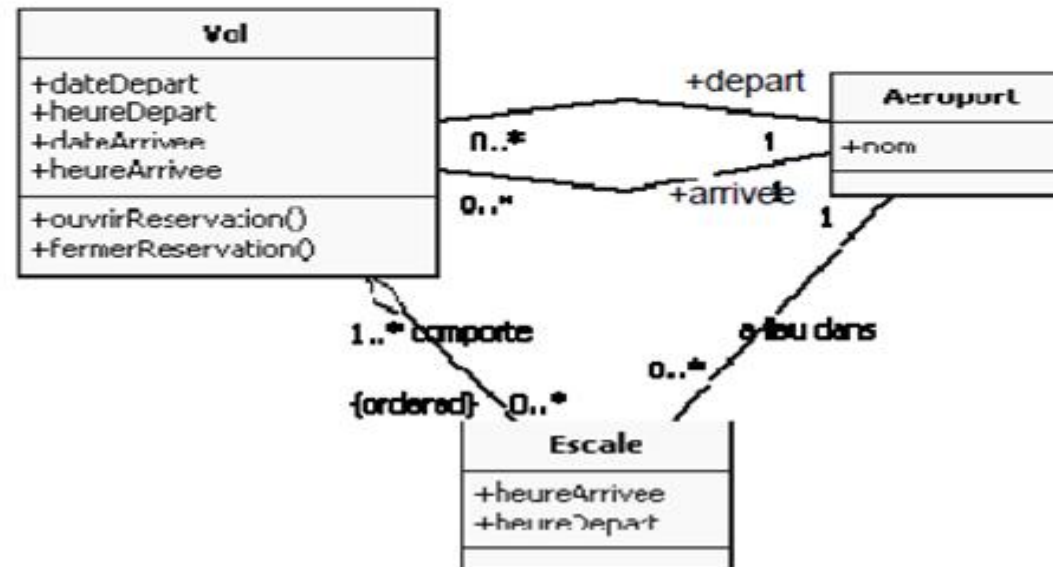
- On pourrait penser qu'une escale n'appartient qu'à un vol et un seul, mais est-ce bien certain?
- Contre-exemple, sous forme de diagramme d'objet



- Une escale peut donc appartenir à deux vols différents

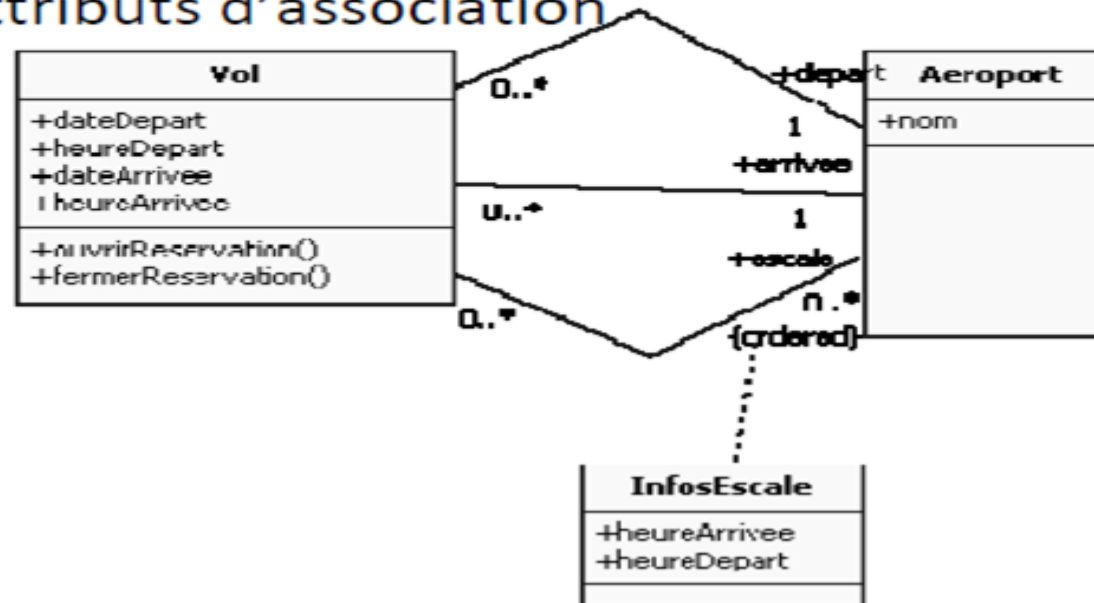
D. ETUDE DE CAS (11/17)

- L'association entre Vol et Escale est une agrégation mais pas une composition
- Les escales sont ordonnées par rapport au vol



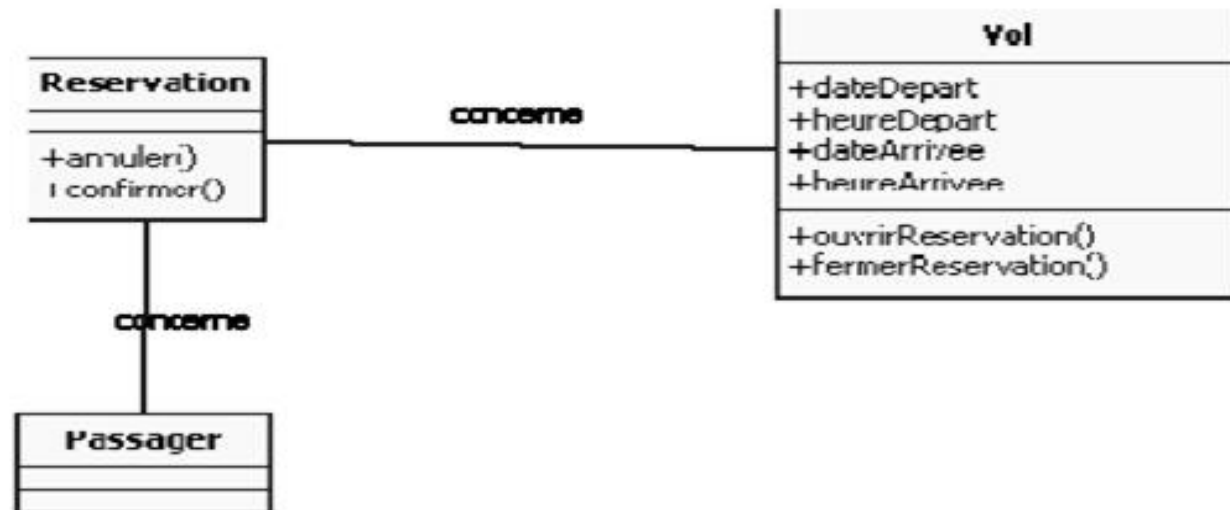
D. ETUDE DE CAS (12/17)

- Sol2 pour les phrases 8 et 9
 - Escale comme un troisième rôle joué par un aéroport par rapport à un vol
 - Les attributs heureArrivee et heureDepart deviennent alors des attributs d'association



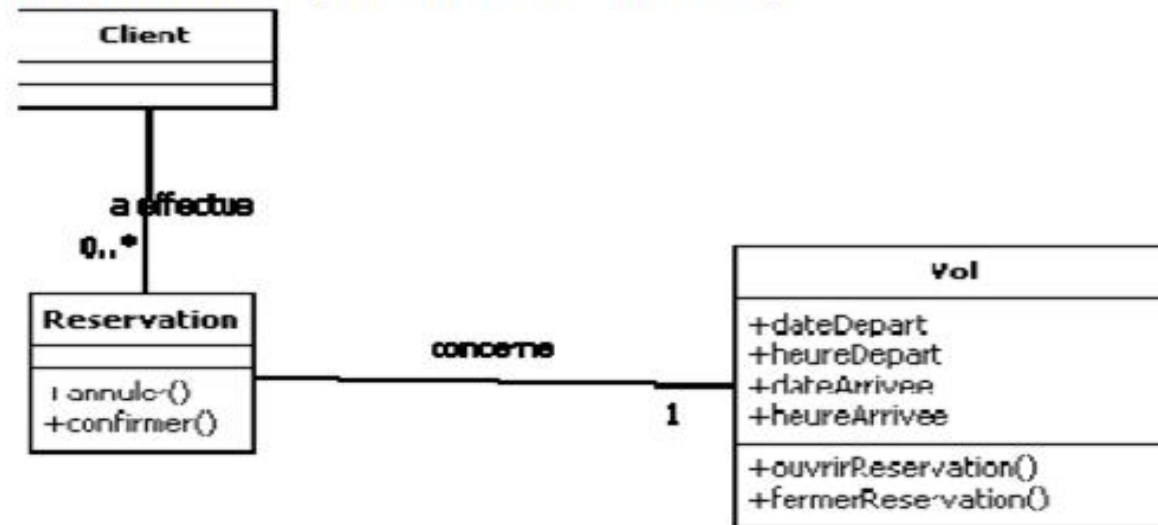
D. ETUDE DE CAS (13/17)

4. Une réservation concerne un seul vol et un seul passager
5. Une réservation peut être annulée ou confirmée



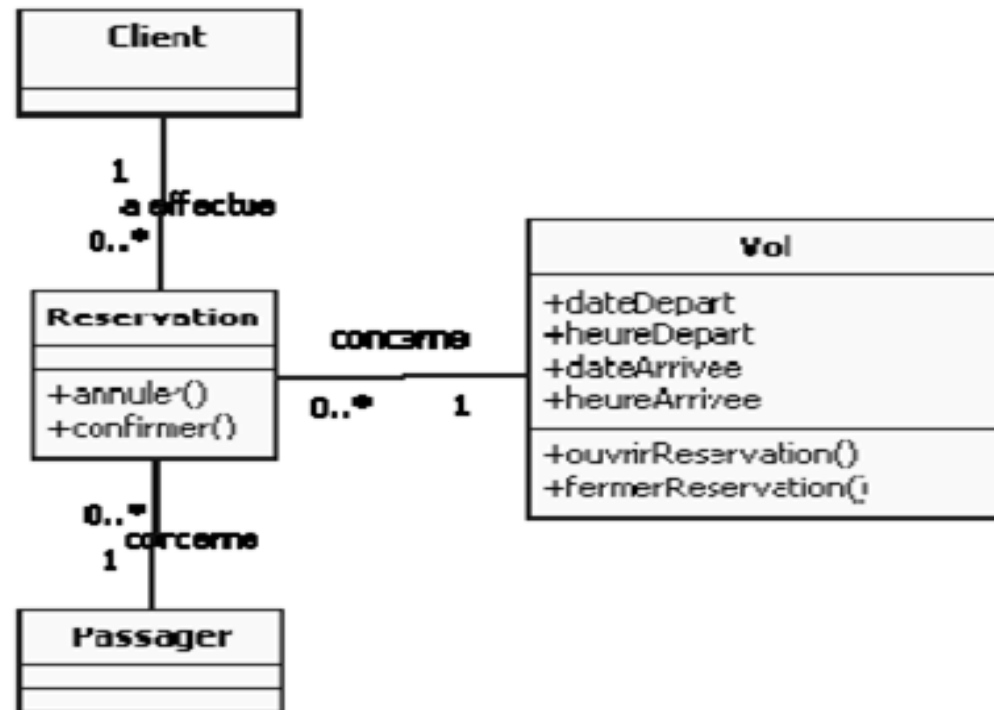
D. ETUDE DE CAS (14/17)

3. Un client peut réserver un ou plusieurs vols, pour des passages différents
- La phrase peut prêter à confusion en raison de la relation directe qu'il semble impliquer entre client et vol
 - Le verbe "réserver" masque le concept déjà identifié de réservation
 - Reformulation de la phrase: un client peut effectuer plusieurs réservations (chacune portant sur un vol)



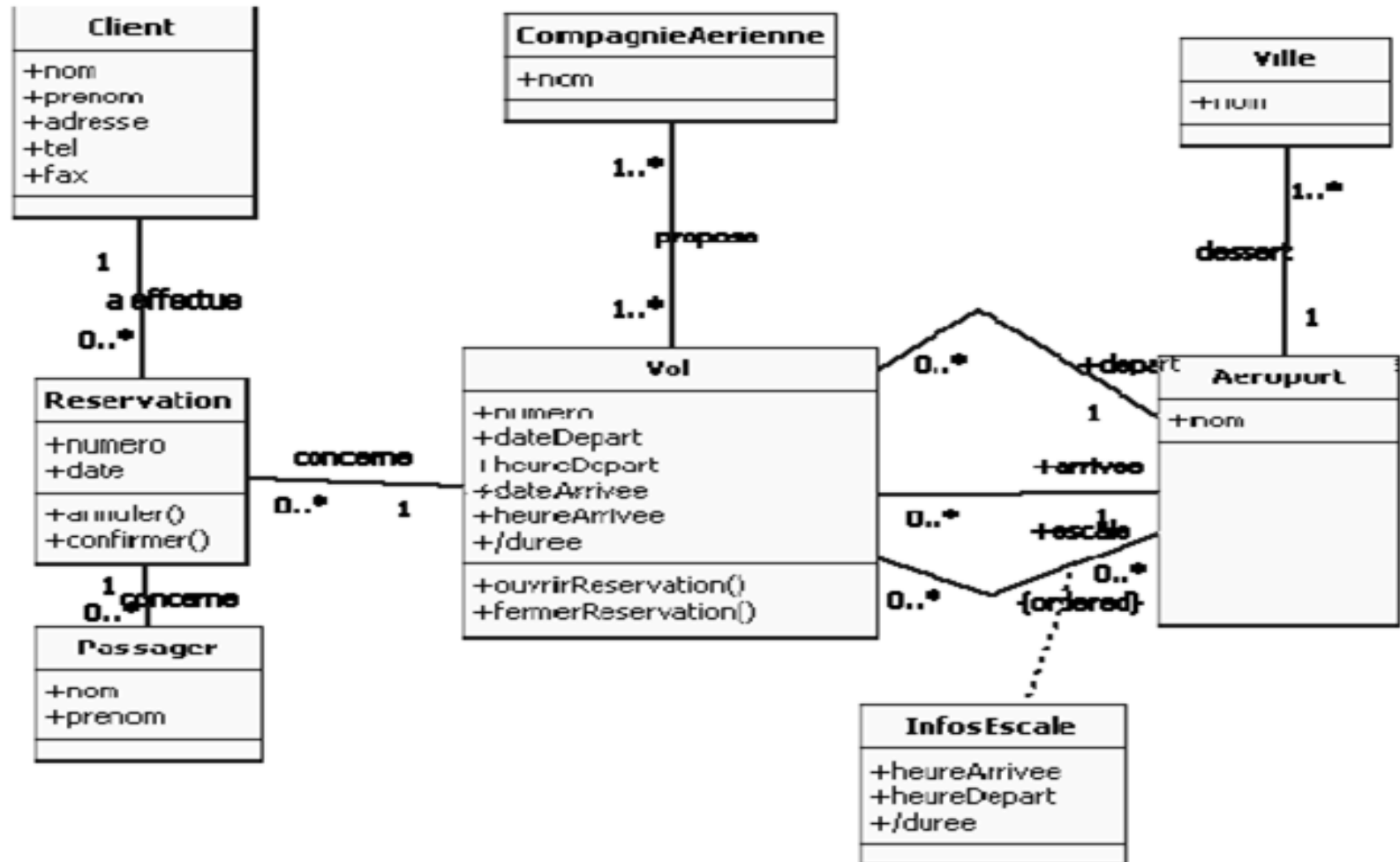
D. ETUDE DE CAS (15/17)

3. Un client peut réserver un ou plusieurs vols, pour des passages différents
 - Il est clair qu'une réservation a été effectuée par un et un seul client
 - Un même vol peut être concerné par zéro ou plusieurs réservations
4. Une réservation concerne un seul vol et un seul passager



D. ETUDE DE CAS (16/17)

- Diagramme de classe d'analyse est le suivant



D. ETUDE DE CAS (17/17)

- Diagramme de package

