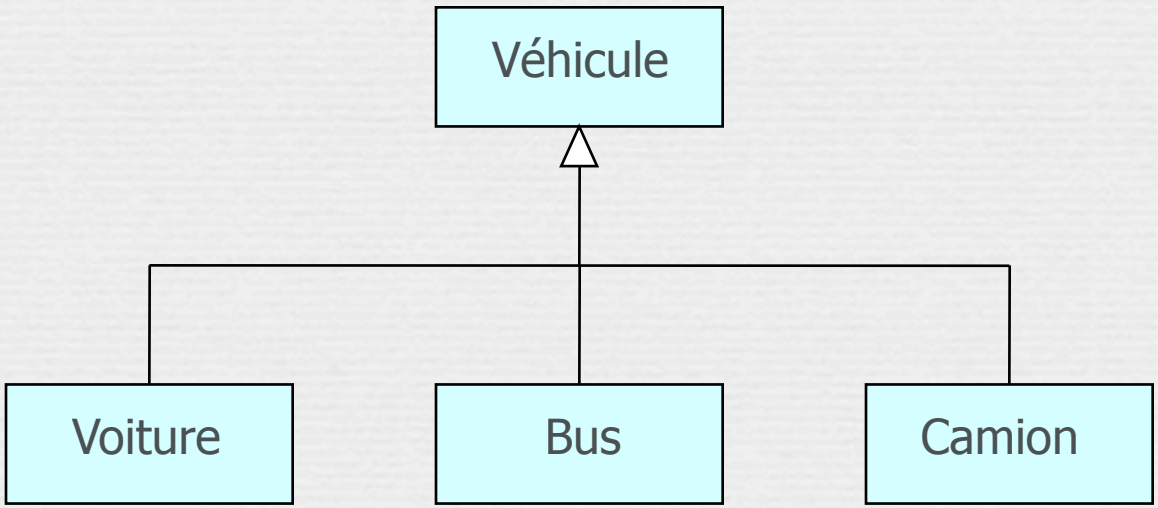
Héritage

- L'héritage est une relation entre une super-classe (classe de base) et ses sous-classes (classes dérivées)
- Deux manières d'identifier une relation d'héritage :
  - Généralisation
  - Spécialisation
- Les éléments communs (attributs, comportements, relations) sont reportés au niveau le plus haut de la hiérarchie

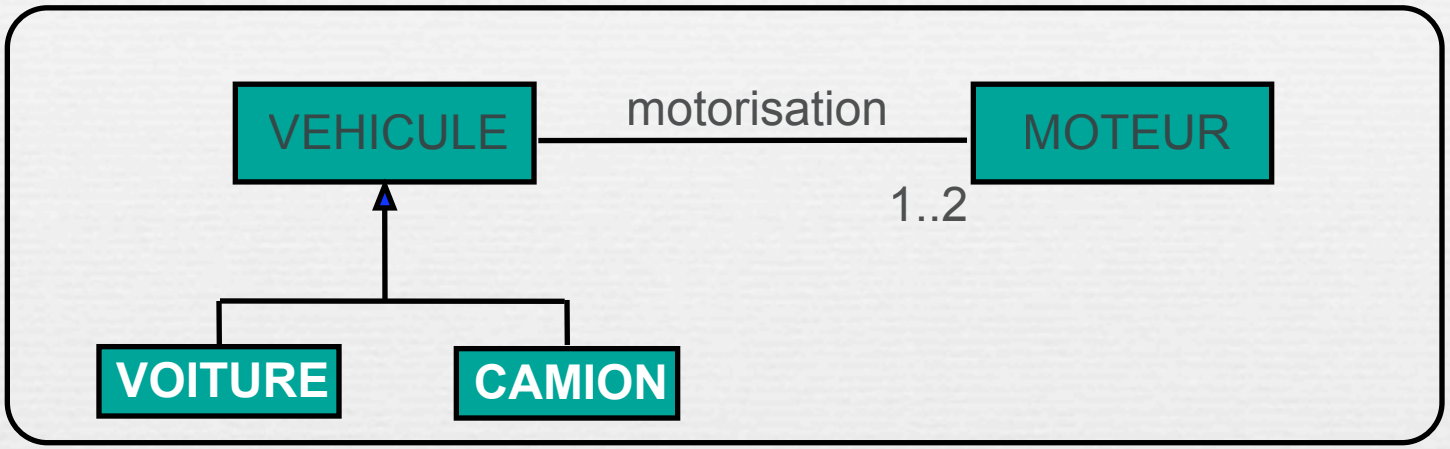
Généralisation/Spécialisation

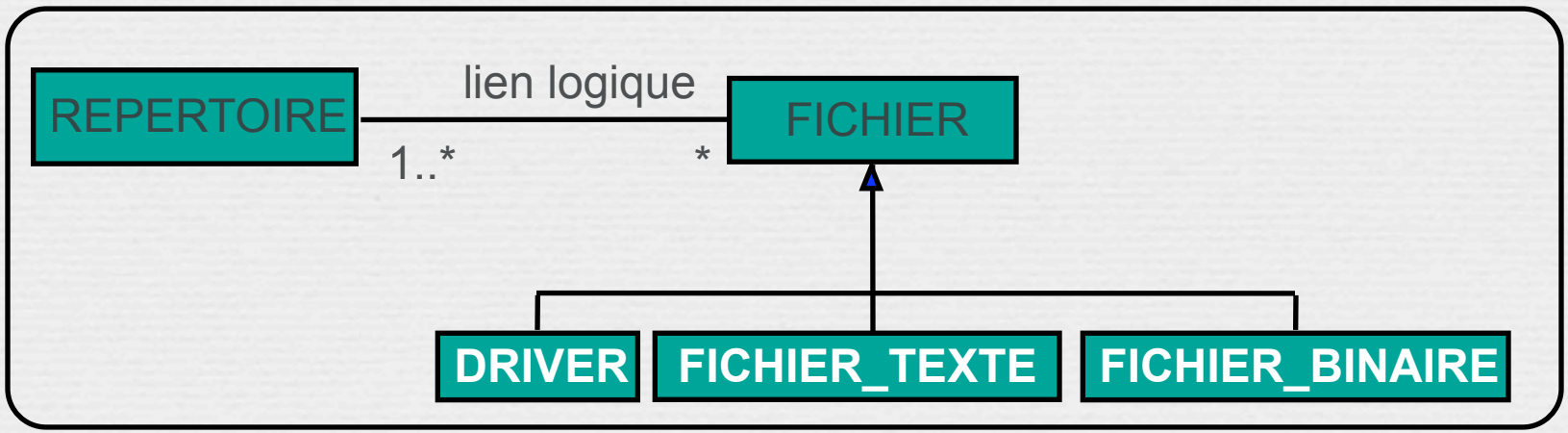
- C'est une relation de classification entre un élément général et un élément plus spécifique
- L'élément le plus spécifique est cohérent avec l'élément le plus général et contient plus d'informations
- Exemple



Héritage des relations

- Les relations sont héritées par les sous classes :

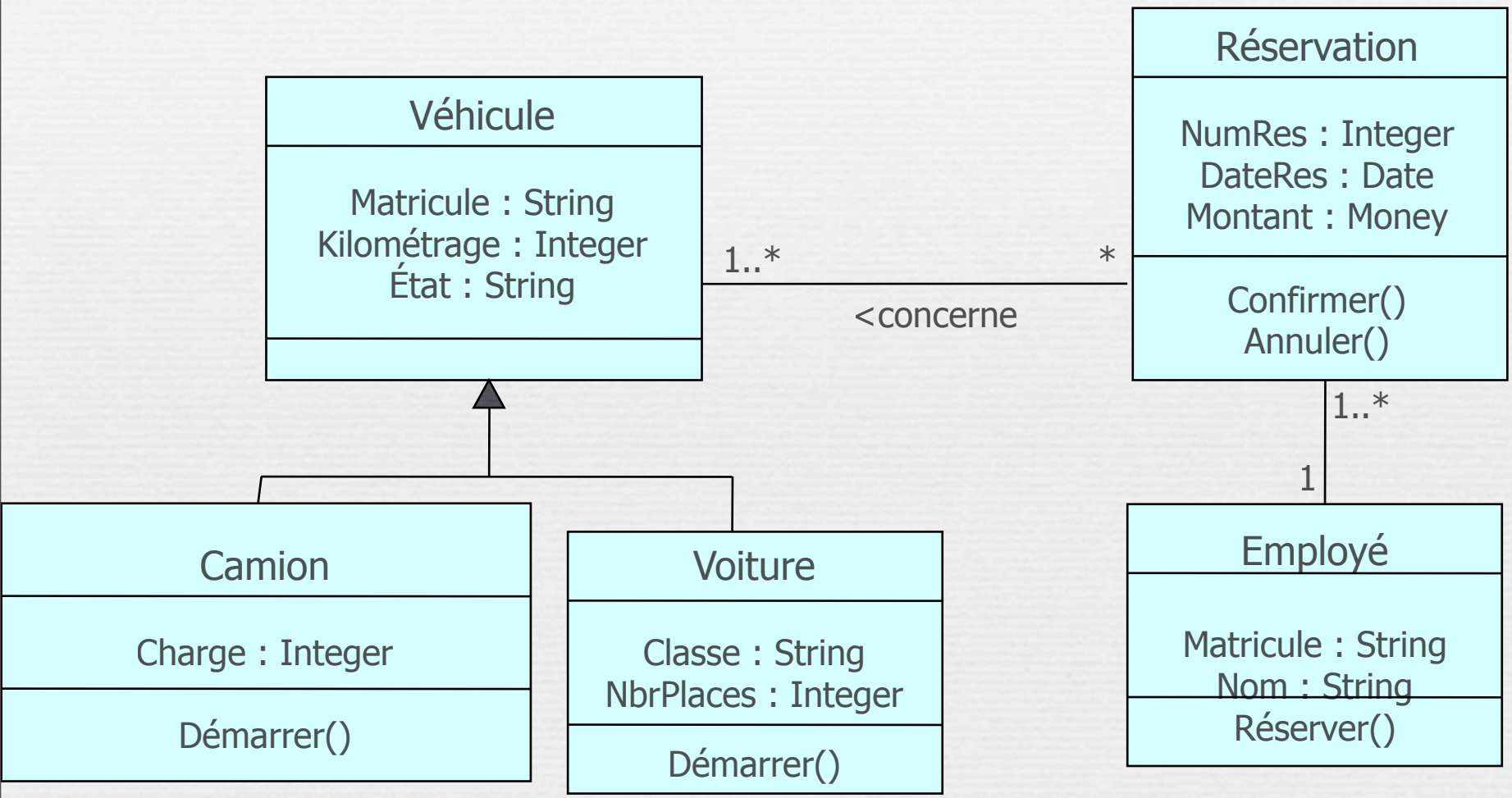




Généralisation signifie :

- Héritage
  - L'enfant acquiert les propriétés du ou des parents (s) : les attributs, les relations et les opérations
- Substituabilité
  - Il est possible de substituer une instance d'une sous-classe à une instance de la classe.

Diagramme de classes…

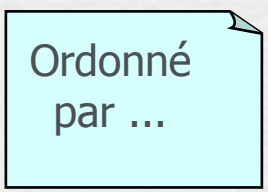



Q6 & Q7

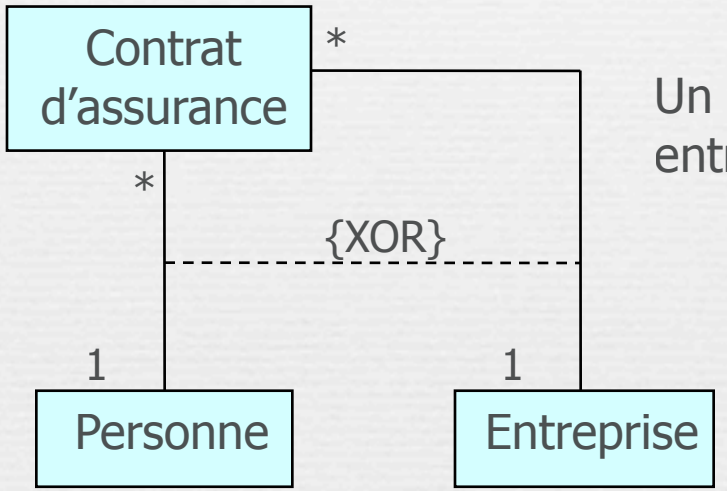
Association ordonnée

- Contraintes sur les associations pour exprimer que les objets sont ordonnés (selon la clé, le nom, la date, etc.)
- Cette contrainte est spécifiée par le stéréotype {Ordonné} du côté de la classe dont les instances sont ordonnés

- Le modèle ne spécifie pas comment les objets sont ordonnés
- Pour décrire comment les objets sont ordonnés on utilise un commentaire en employant la notation graphique suivante :

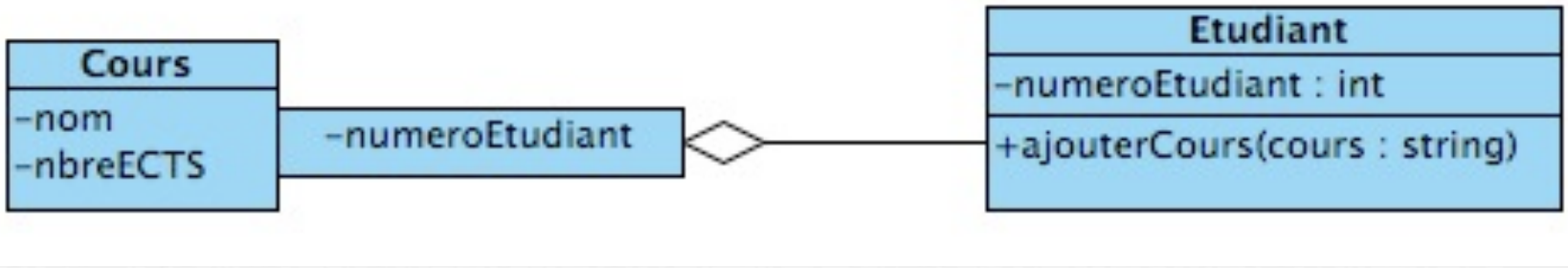


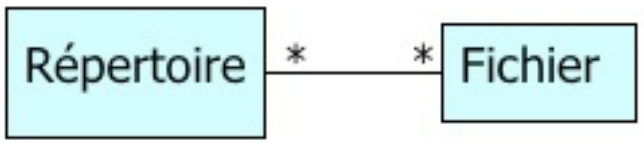
Association « ou-exclusif »



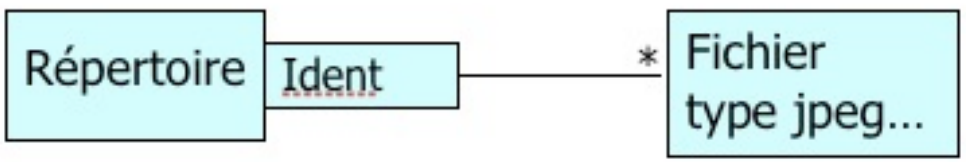
Un contrat d'assurance concerne une entreprise ou une personne mais pas les deux en même temps

Association qualifiée





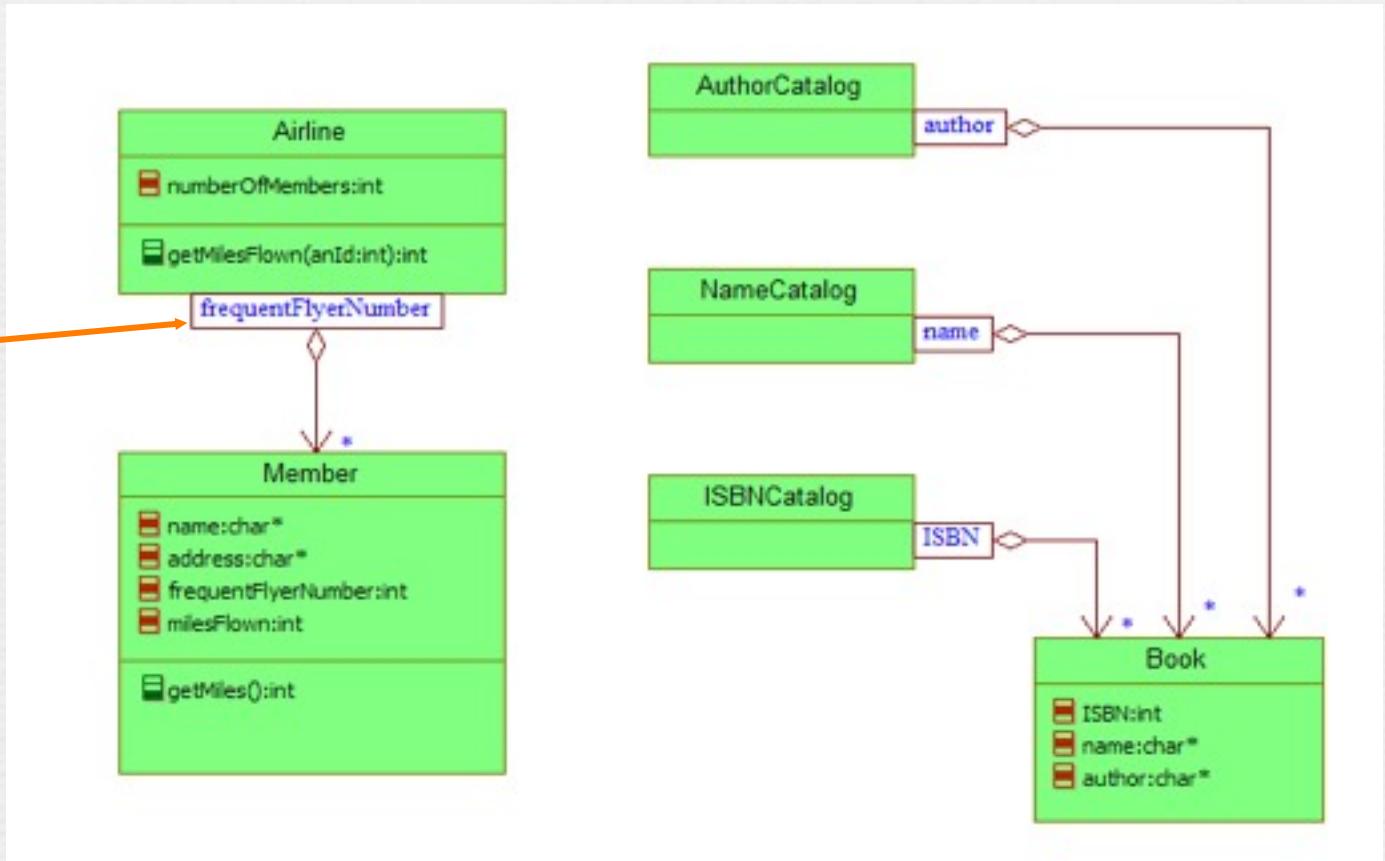
Un répertoire contient 0 ou plusieurs fichiers. Un fichier peut exister dans 0 ou plusieurs répertoires



Chaque fichier est identifié
par un identificateur dans
le répertoire

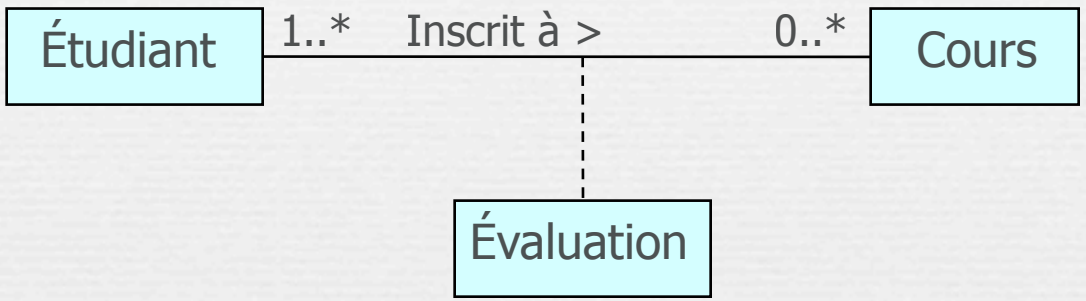
Association qualifiée

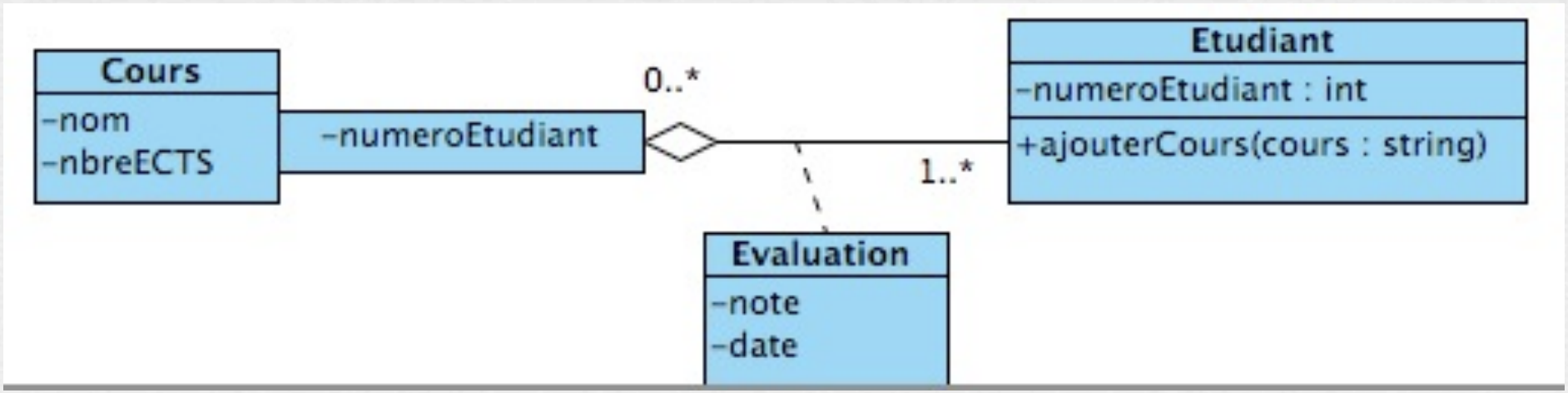
- Utilisée avec une relation de multiplicité *.
- Permet de trier la relation en fonction des valeurs d'un attribut.



Classe d'association : exemple

- Une classe d'association est utilisée quand une information semble appartenir aux deux objets ou à aucun objet





- UML Par la pratique (surtout dans sa dernière édition)
- Méthodologie en Ingénierie du logiciel, Modélisation Orientée objet, M.Grimaldi – janvier 2010

Application

Système de réservation de vol

Approfondissement Par l'exemple

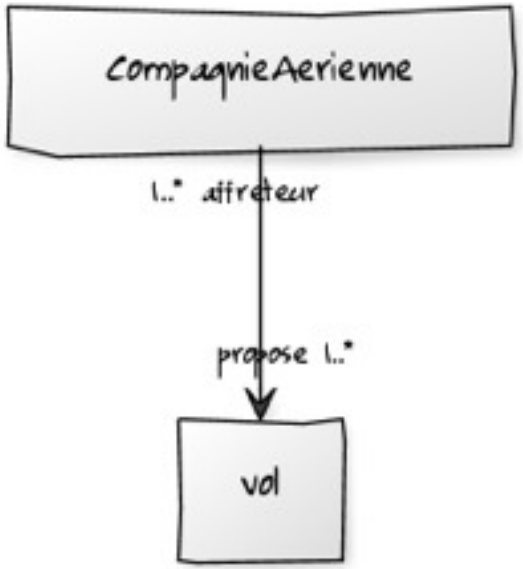
Interview des experts métier

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.

Interview des experts métier

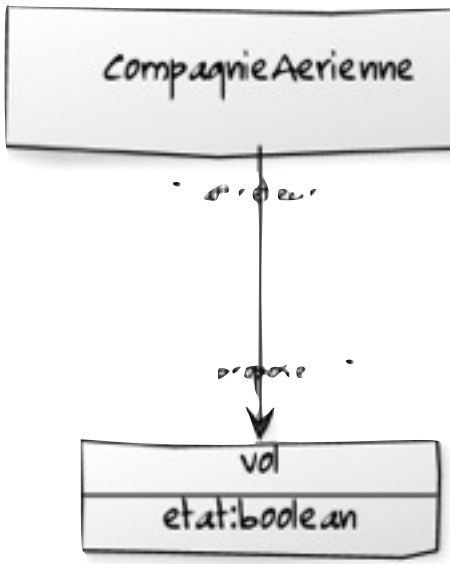
1. Des compagnies aériennes proposent différents vols.
2. Un vol est ouvert à la réservation et refermé sur ordre de la compagnie.
3. Un client peut réserver un ou plusieurs vols, pour des passagers différents.
4. Une réservation concerne un seul vol et un seul passager.
5. Une réservation peut être 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.

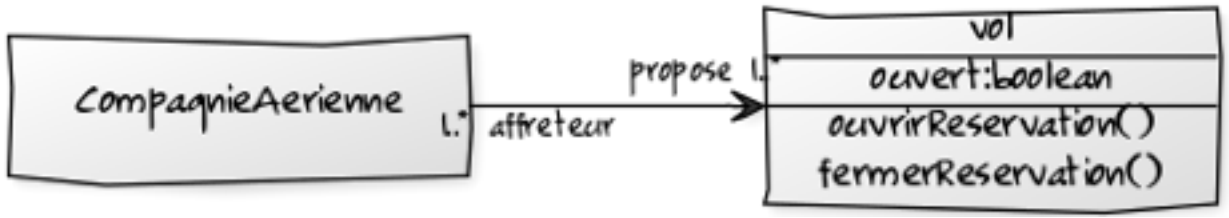
Nous partirons du principe qu'un vol est proposé le plus souvent par une seule compagnie aérienne, mais qu'il peut également être partagé entre plusieurs affréteurs.



http://yuml.me

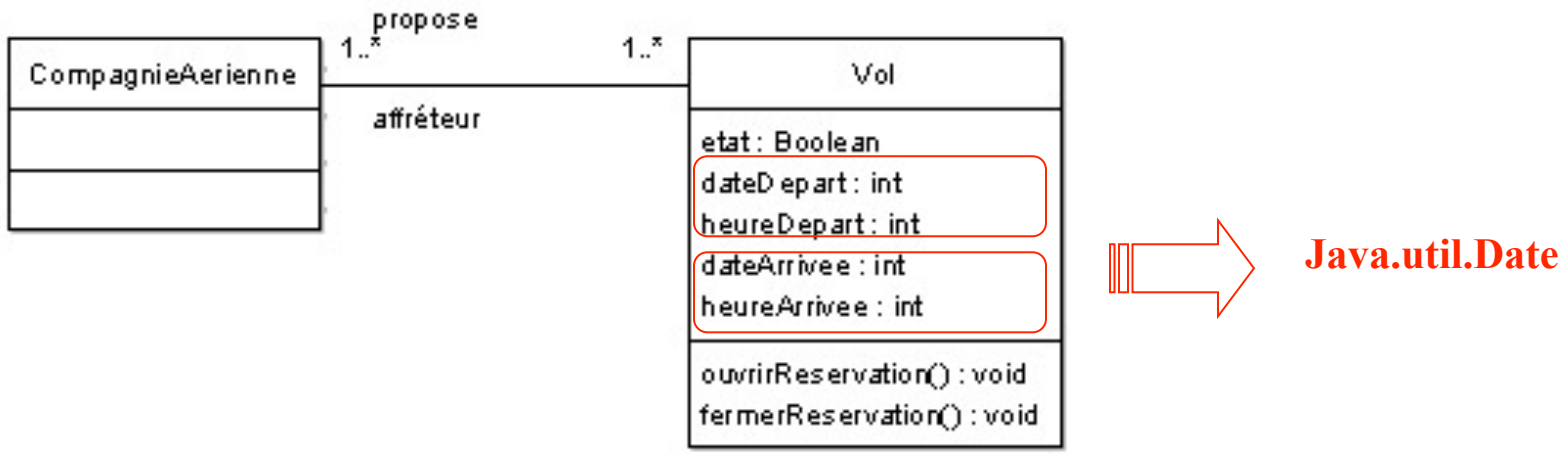
2. Un vol est ouvert à la réservation et refermé sur ordre de la compagnie.





Étape 2 - Modélisation des phrase 6, 7 et 8

7. Un vol a un jour et une heure de départ, et un jour et une heure d'arrivée.



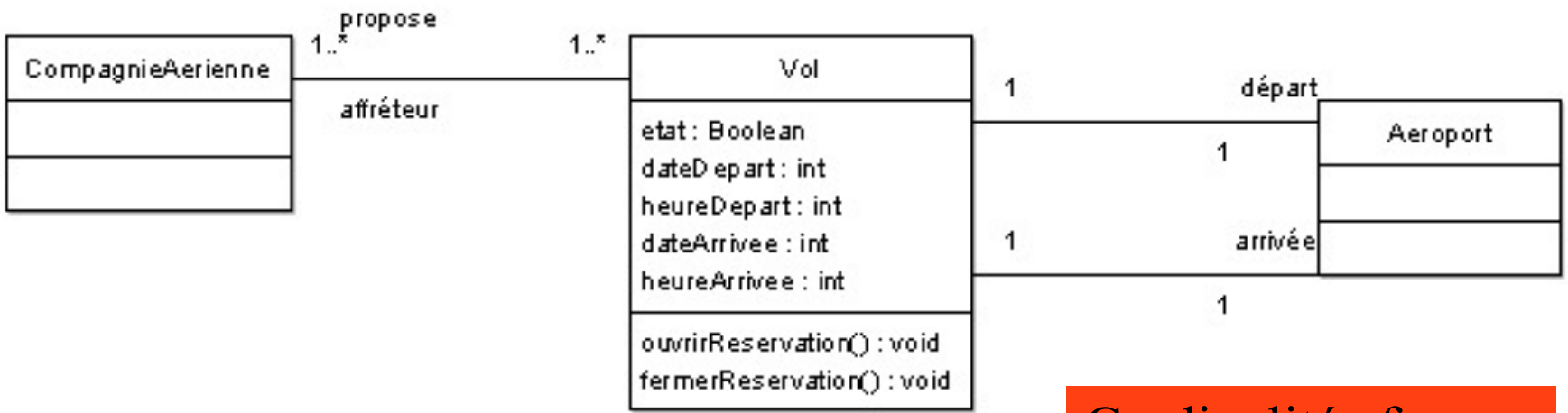
Objet ou attribut?

Un objet est un élément plus « important » qu'un attribut.

- si l'on ne peut demander à un élément que sa valeur - **attribut**
- si plusieurs questions s'y appliquent - **objet** (qui possède lui-même plusieurs attributs, ainsi que des liens avec d'autres objets.)

6. Un vol a un aéroport de départ et un aéroport d'arrivée.

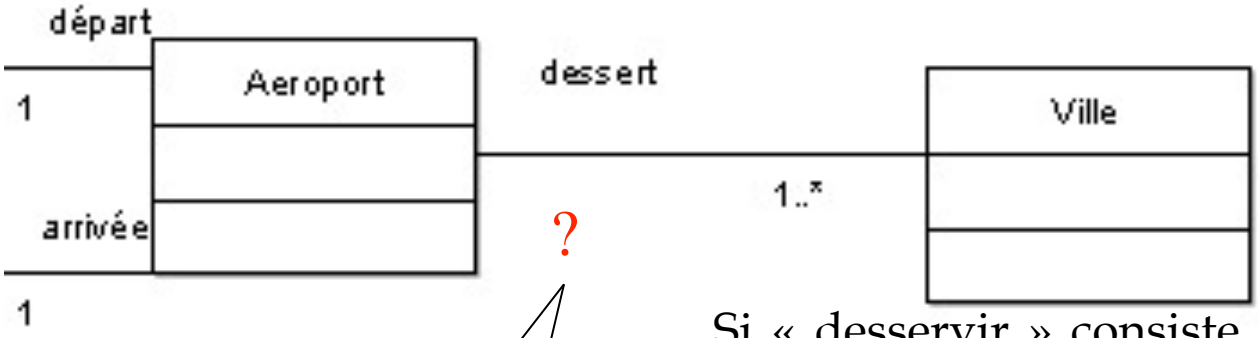
Contrairement aux notions d'heure et de date qui sont des types «*simples* », la notion d'aéroport est complexe; elle fait partie du «métier». Un aéroport ne possède pas seulement un nom, il a aussi une capacité, dessert des villes, …etc... A ce stade nous connaissons mal l'usage qui en sera fait. C'est pour ces raisons que nous préférons créer une classe *Aéroport* plutôt que de simples attributs *aeroportDepart* et *aeroportArrivee* dans la classe *Vol*.



Cardinalités fausses

Modélisation de la phrase 10.

10. Chaque aéroport dessert une ou plusieurs villes.



?

1

0..*

Si « desservir » consiste simplement à désigner le moyen de transport par les airs le plus proche, toute ville est toujours desservie par un et un seul aéroport.

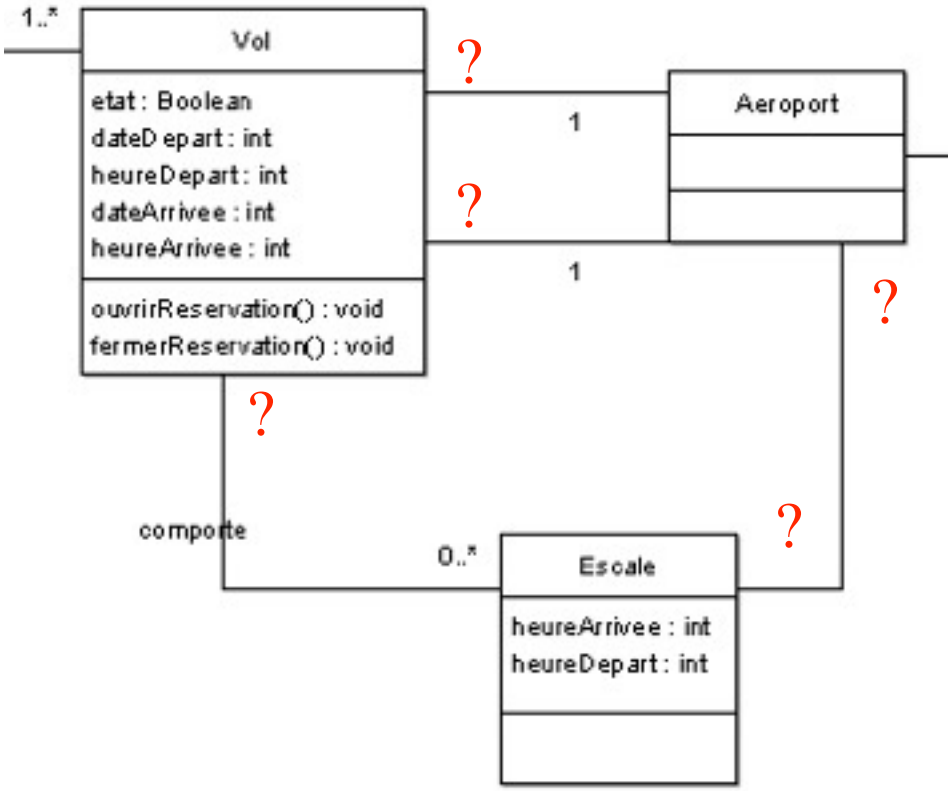
Si « desservir » vaut par exemple pour tout moyen de transport aérien se trouvant à moins de trente kilomètres, alors une ville peut être desservie par 0 ou plusieurs aéroports.

Étape 3 - Modélisation des phrases 8 et 9

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

- Chaque escale a deux propriétés : heure d'arrivée et heure de départ.
- Elle est en relation avec des vols et des aéroports, qui sont eux-mêmes des objets (phrase 8).

Il est donc naturel d'en faire une classe à son tour.

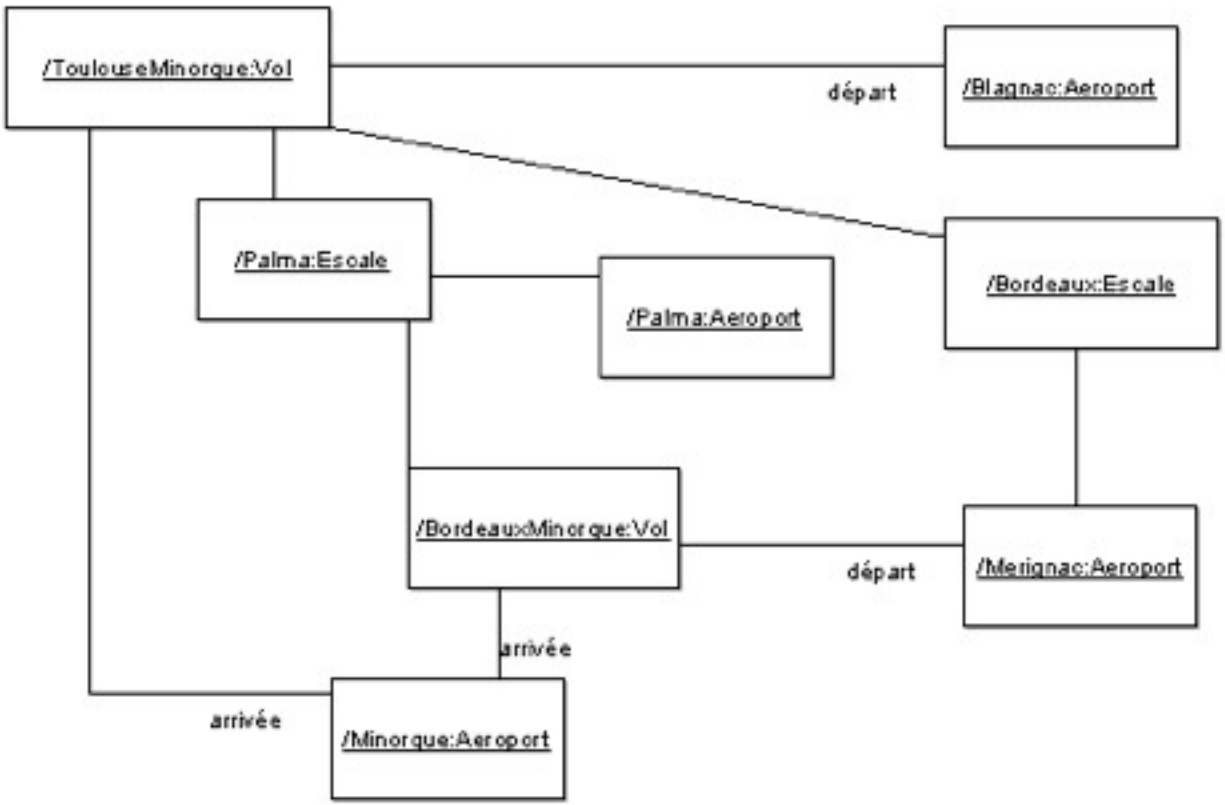


La phrase 8 est imprécise : une escale peut-elle appartenir à plusieurs vols, et quelles sont les multiplicités entre ***Escale*** et ***Aéroport*** ?

De plus, le schéma n'indique toujours pas les multiplicités du côté ***Vol*** avec ***Aéroport***

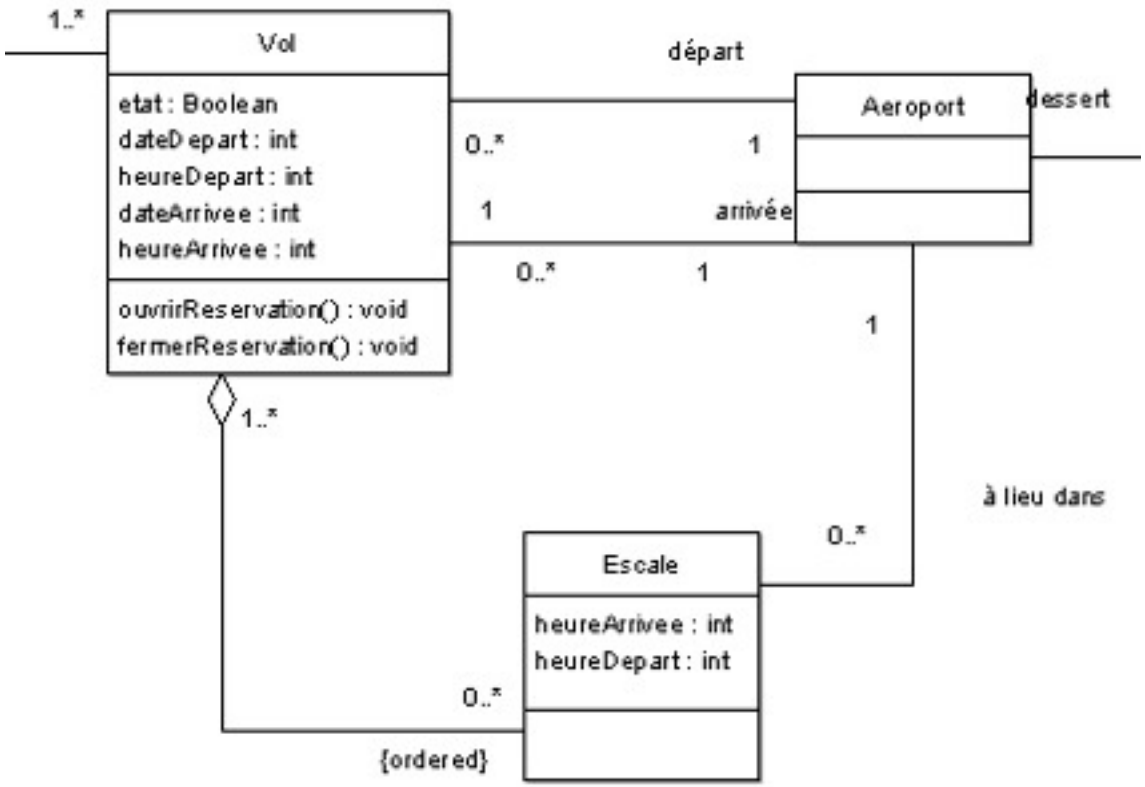
On peut ajouter les multiplicités entre *Escale* et *Aéroport:* une escale a lieu dans un et un seul aéroport, et un aéroport peut servir à plusieurs escales. De même, un aéroport peut servir de départ ou d'arrivée à plusieurs vols.

Après consultation de l'expert métier, un contre-exemple nous est donné, sous forme du diagramme d'objets (collaboration)



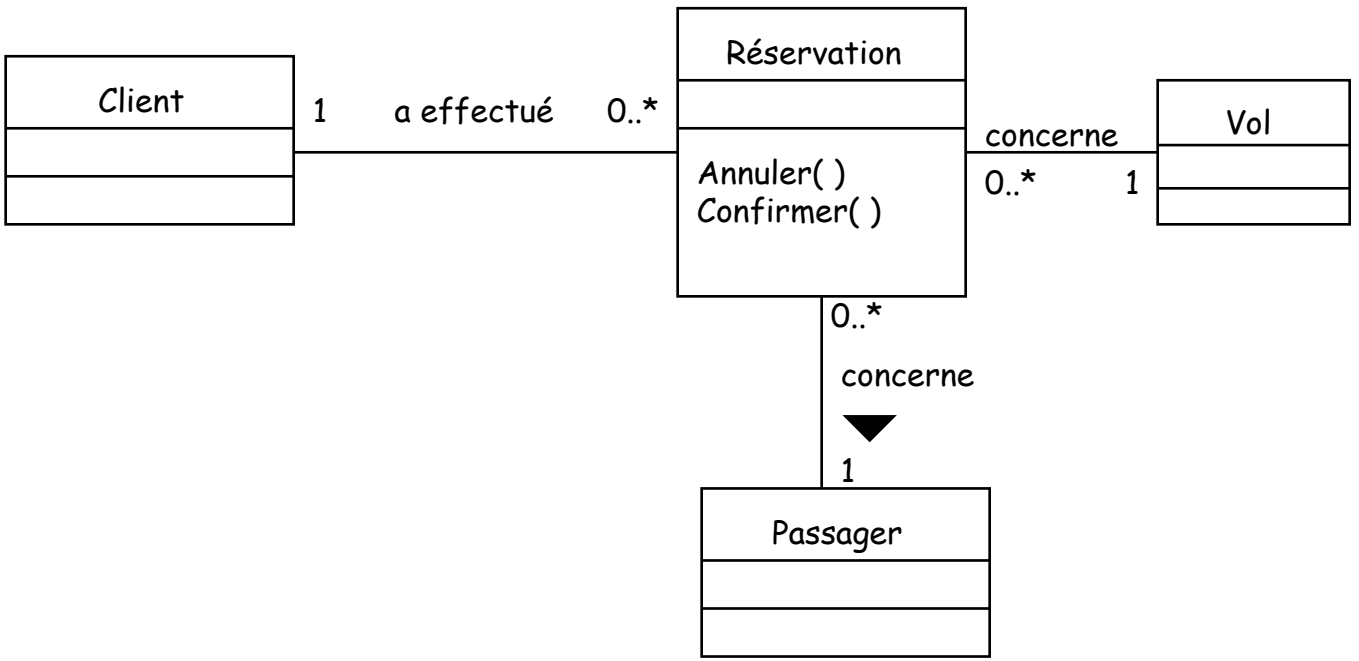
Pour finaliser le diagramme des phrases 8 et 9, il nous suffit d'ajouter :

les escales sont ordonnées par rapport au vol.



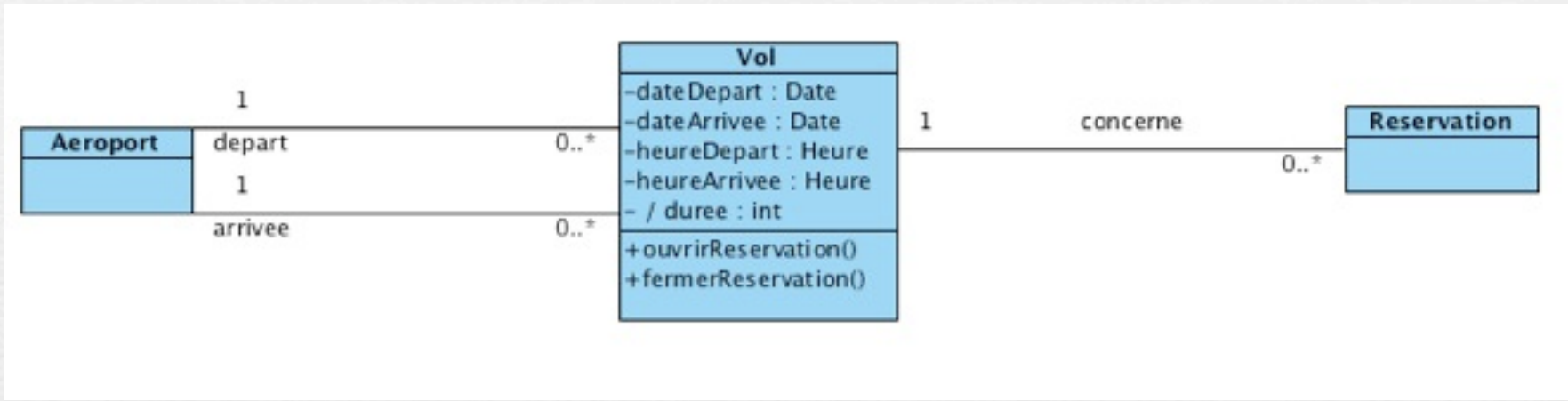
Étape 4 - Modélisation des phrases 3, 4 et 5

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.



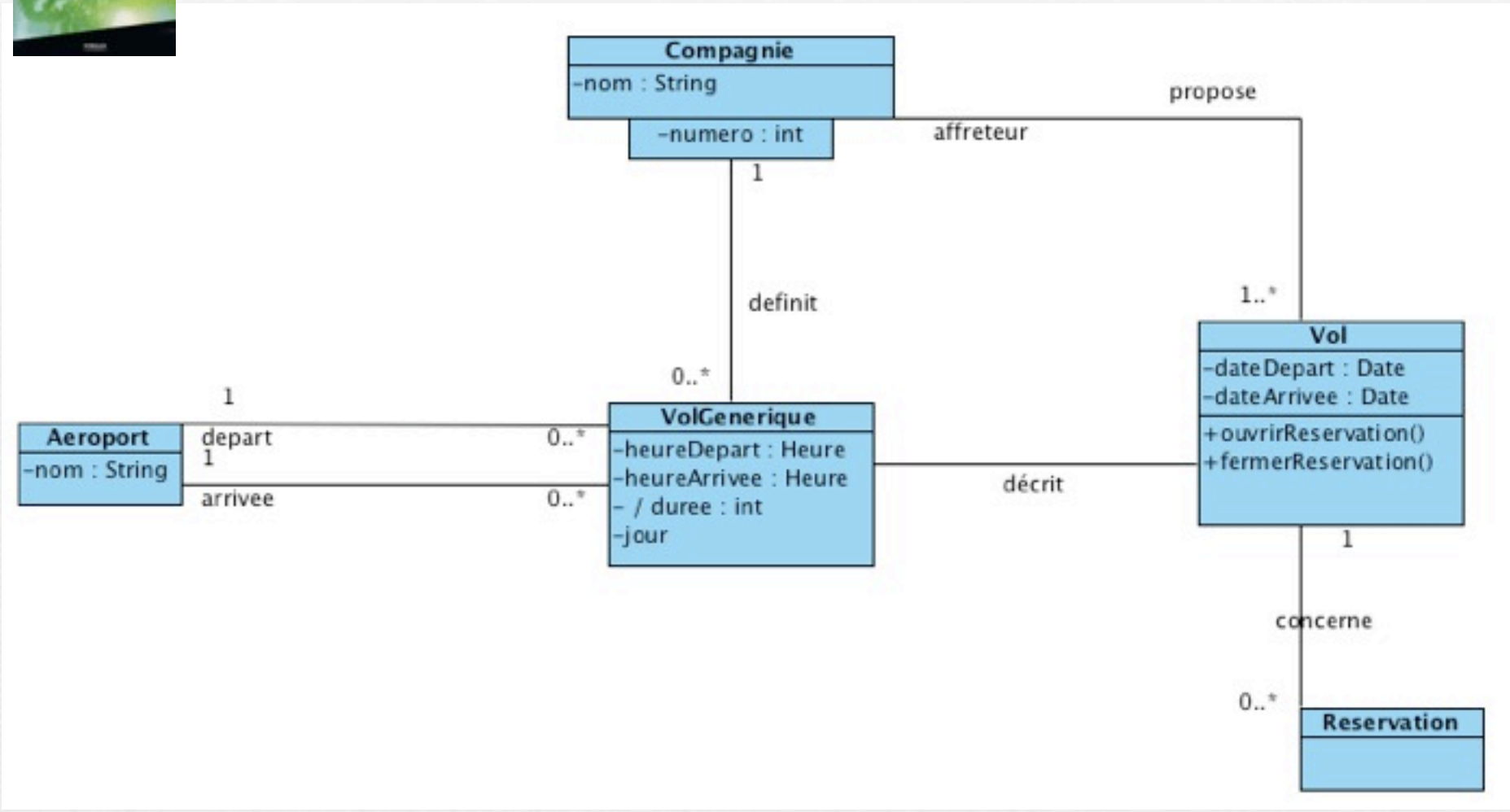
http://eric.univ-lyon2.fr/~kaouiche/uml/vol.ppt

Questions?

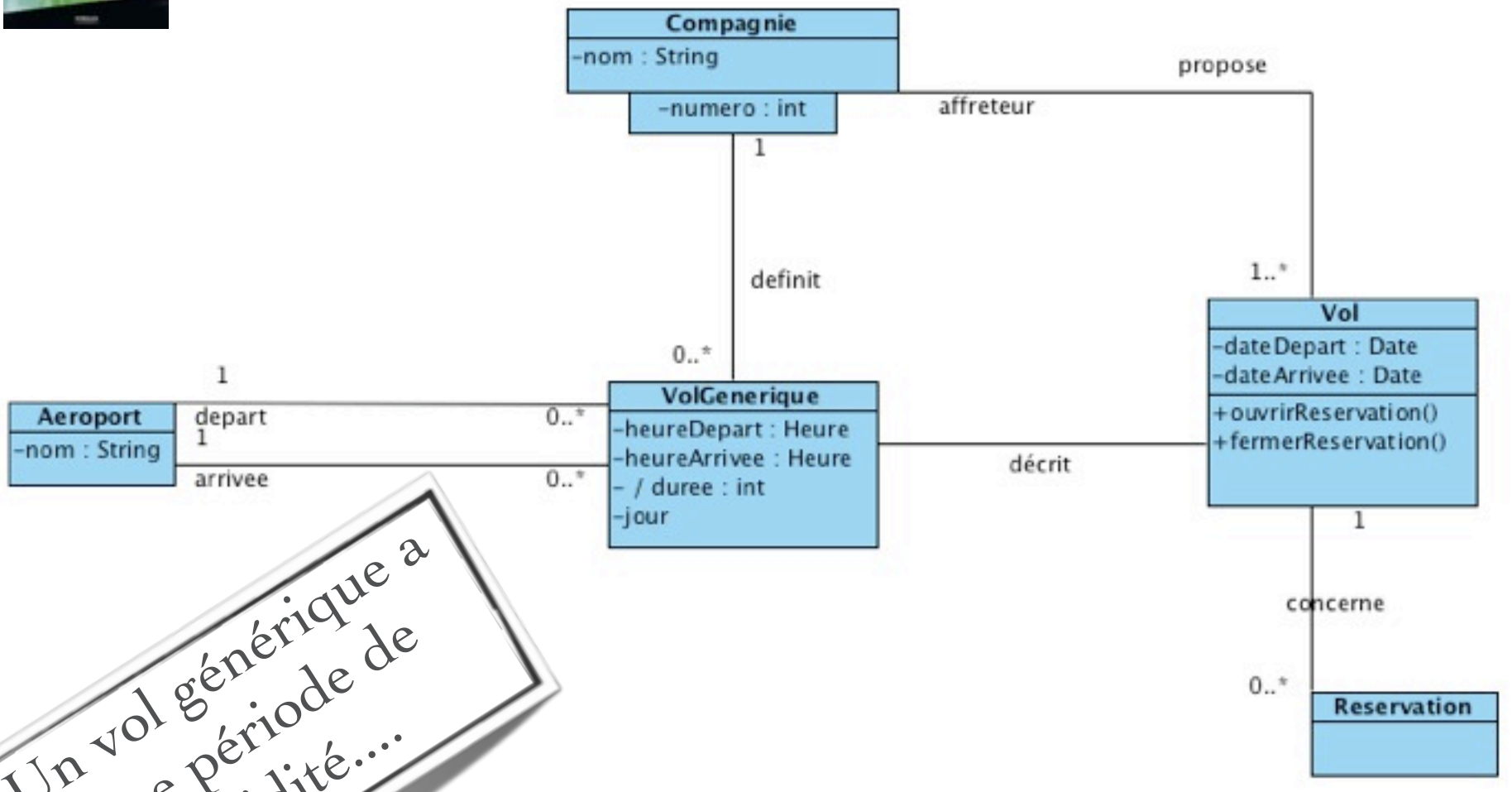


- Existe-t-il des vols Nice-Paris le lundi matin?
- Réserver le vol Nice-Paris du lundi 28/2/11 à 8h
- Les vols Nice-Tunisie sont annulés pour les 15 prochains jours... Puis rétablis.... Comment les recréer ?

Des rôles mieux définis

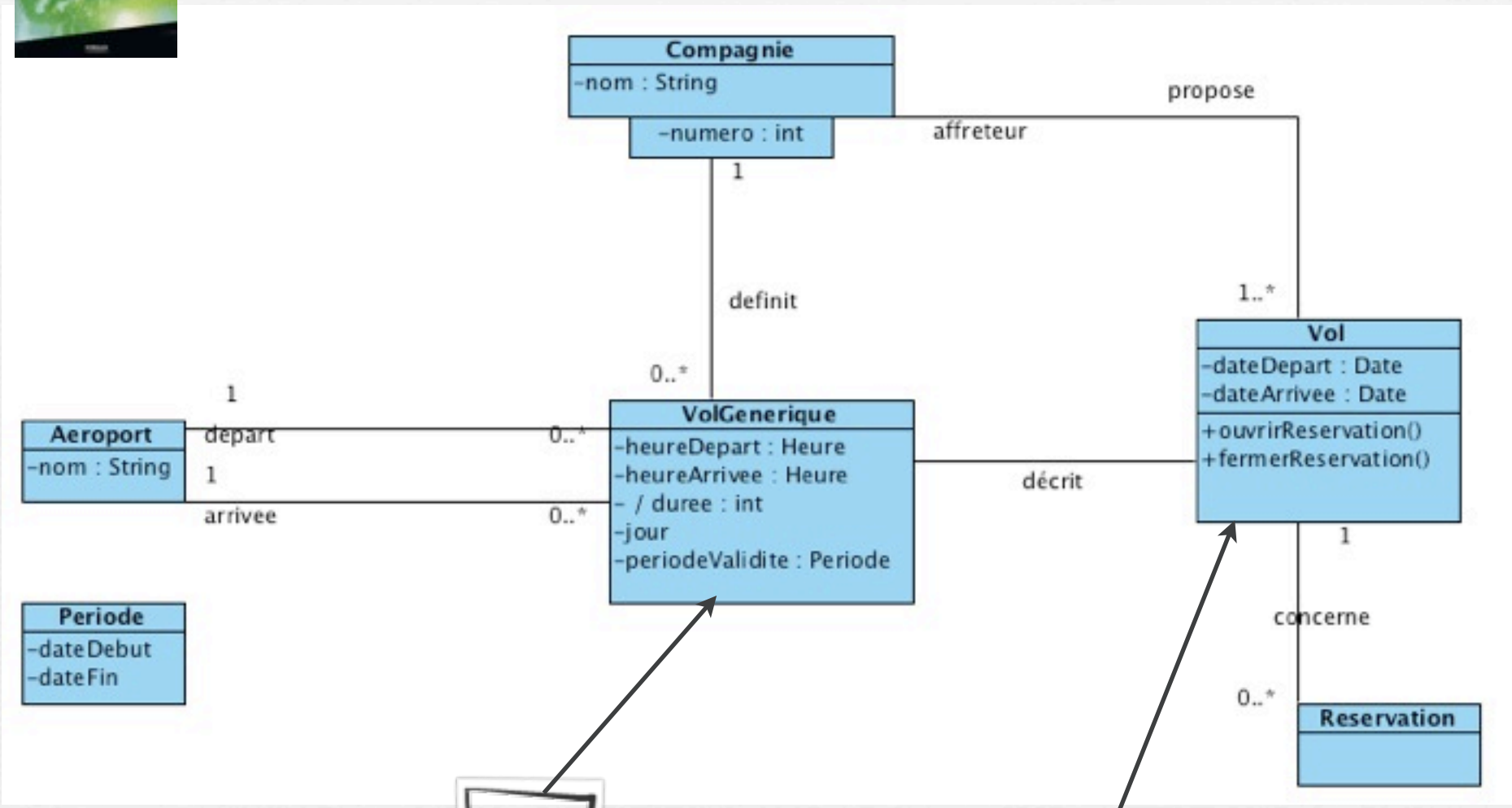


Des rôles mieux définis



Un vol générique a une période de validité....

Des rôles mieux définis



Conseils pratiques

- Réfléchir au problème avant de commencer
  - Soigner le nommage, insister sur le nommage des relations et des rôles
    - Les noms des attributs débutent par une minuscule
    - Les noms des classes débutent par une majuscule et peuvent contenir plusieurs mots concaténés commençant par une majuscule

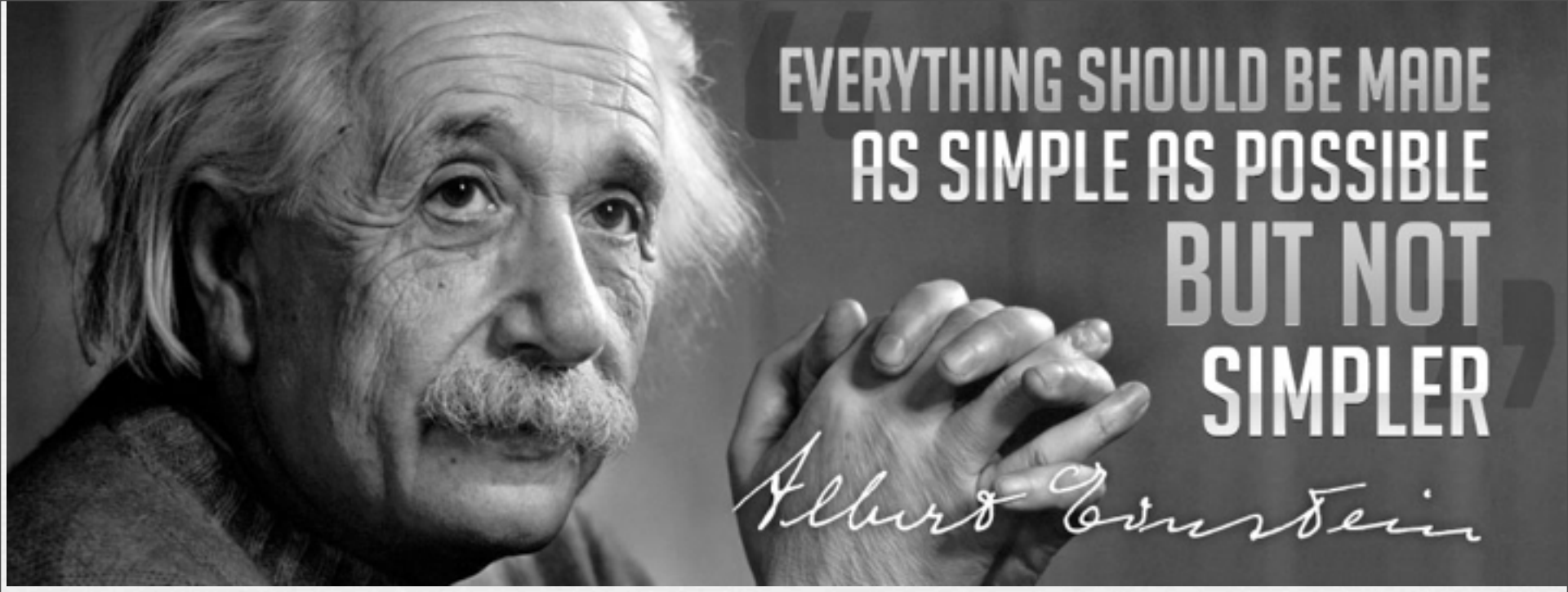
Conseils pratiques

- Réfléchir au problème avant de commencer
  - Soigner le nommage, insister sur le nommage des relations et des rôles
- Faire simple!
  - «*Things must be as simple as possible, but no simpler*». A. Einstein
  - éviter toute complication nuisible
    - se dégager de l'implémentation : raisonner objets, classes, messages, relations, attributs, opérations
  - ne pas s'inquiéter si les possibilités de la notation ne sont pas toutes exploitées

Conseils pratiques (suite)

- Approche incrémentale
  - Itérer
  - Savoir s'arrêter avant d'atteindre la perfection...
    - prise en compte qualité (niveau de précision), coûts, délais...
    - asservissement au processus de développement
- Faire simple (encore)
  - *Un bon modèle n'est pas un modèle où l'on ne peut plus rien ajouter, mais un modèle où on ne peut plus rien enlever.*

(d'après A. de St-Exupéry)

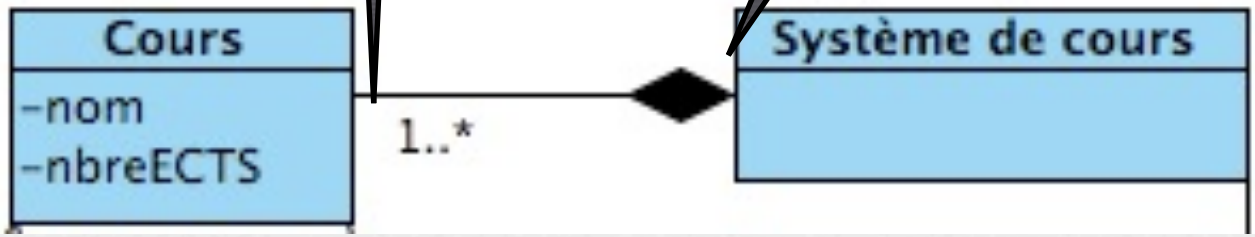

EVERYTHING SHOULD BE MADE
AS SIMPLE AS POSSIBLE
BUT NOT
SIMPLER
Albert Einstein


“Il semble que la perfection soit atteinte
non quand il n'y a plus rien à ajouter,
mais quand il n'y a plus rien à retrancher.”
ANTOINE DE SAINT-EXUPERY

Composition

Cycle de vie dépendant : la destruction du système de cours => la destruction des cours

1 seul!
Les composants ne peuvent pas être partagés



Attention
un Point de vue