Conception Orientée Objet

AMOSSE EDOUARD

INSPIRÉ DU COURS DE FREDERIC MALLET

Diagramme de Classes

- Diagramme représentant une vue statique du système
- Décrit les responsabilités du système
- La base des diagrammes de composantes et de déploiement
- Utilisé pour représenter les classes et leurs relations
- Décrit les attributs et opérations des classes

Les classes

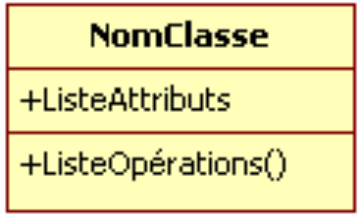
Nom de la classe

Attributs

Opérations

Classe

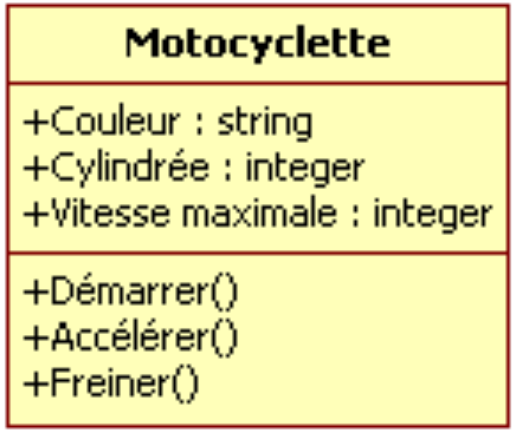
- Décrit le domaine de définition d'un ensemble d'objets.
- Description abstraite de cet ensemble.
- Vue comme la factorisation des éléments communs à un ensemble d'objets.
- Modélisée à l'aide d'un rectangle, comportant 3 zones :
  - La désignation de la **classe**
  - La description des **attributs**
  - La description des **opérations**



- La première zone est obligtoire (le nom), les autres sont optionnelles

Nom d’une classe

- Le nom de la classe peut être qualifié par un «stéréotype» ou d’autres précisions (auteur de la classe par exemple).
- Un nom de classe est toujours au singulier : pas de "s" à la fin, même si conceptuellement une classe est un ensemble d'instances.

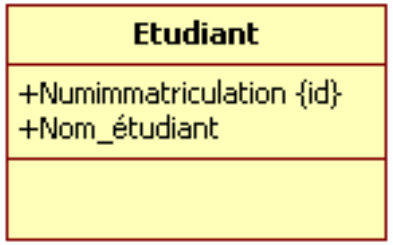


Attributs

- La description complète des attributs d'une classe comporte les caractéristiques suivantes :
  - Nom d'attribut : nom unique dans sa classe.
  - Multiplicité: la multiplicité indique le nombre minimum et le nombre maximum de valeurs possibles de l'attribut pour une instance de la classe. Par défaut la multiplicité est [1,1]. Exemple prénom [1,5] si une personne peut avoir au maximum cinq prénoms (facultative).
  - Type : type de l'attribut (integer, string,.....)
  - Valeur initiale : la valeur par défaut à la création de l'attribut (facultative). Exemple date = date du jour.

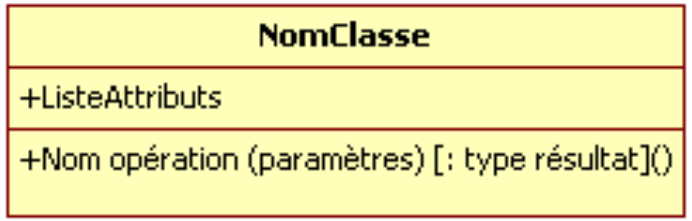
Attributs (suite)

- Un attribut dont la valeur peut être calculée à partir d'autres attributs de la classe est un attribut dérivé qui se note « /nom de l'attribut dérivé ».
- Un attribut est appelé identifiant (marqué avec {id}) de la classe s'il joue un rôle particulier en permettant de repérer de façon unique chaque instance de la classe.
- Exemple



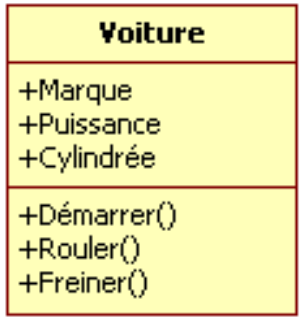
Opérations

- La description complète d'une opération d'une classe peut comporter un certain nombre de caractéristiques :
  - Nom opération
  - Paramètres : 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 ( ).
  - Type résultat : type de valeur retournée. Par défaut, une opération ne retourne pas de valeur.



Opérations (Suite)

- Une méthode est l'implémentation d'une opération sur une classe. Ainsi, en spécifications et conception orientée objet (OMT, UML,...), on parle d'opérations et en programmation orientée objet (JAVA, C++) on parle de méthodes.
- Exemple : La classe « Voiture » possède quatre opérations : « démarrer() », « rouler() », « freiner() » et « arrêter() » sans paramètre ni valeur de retour.



Visibilité des attributs et des opérations

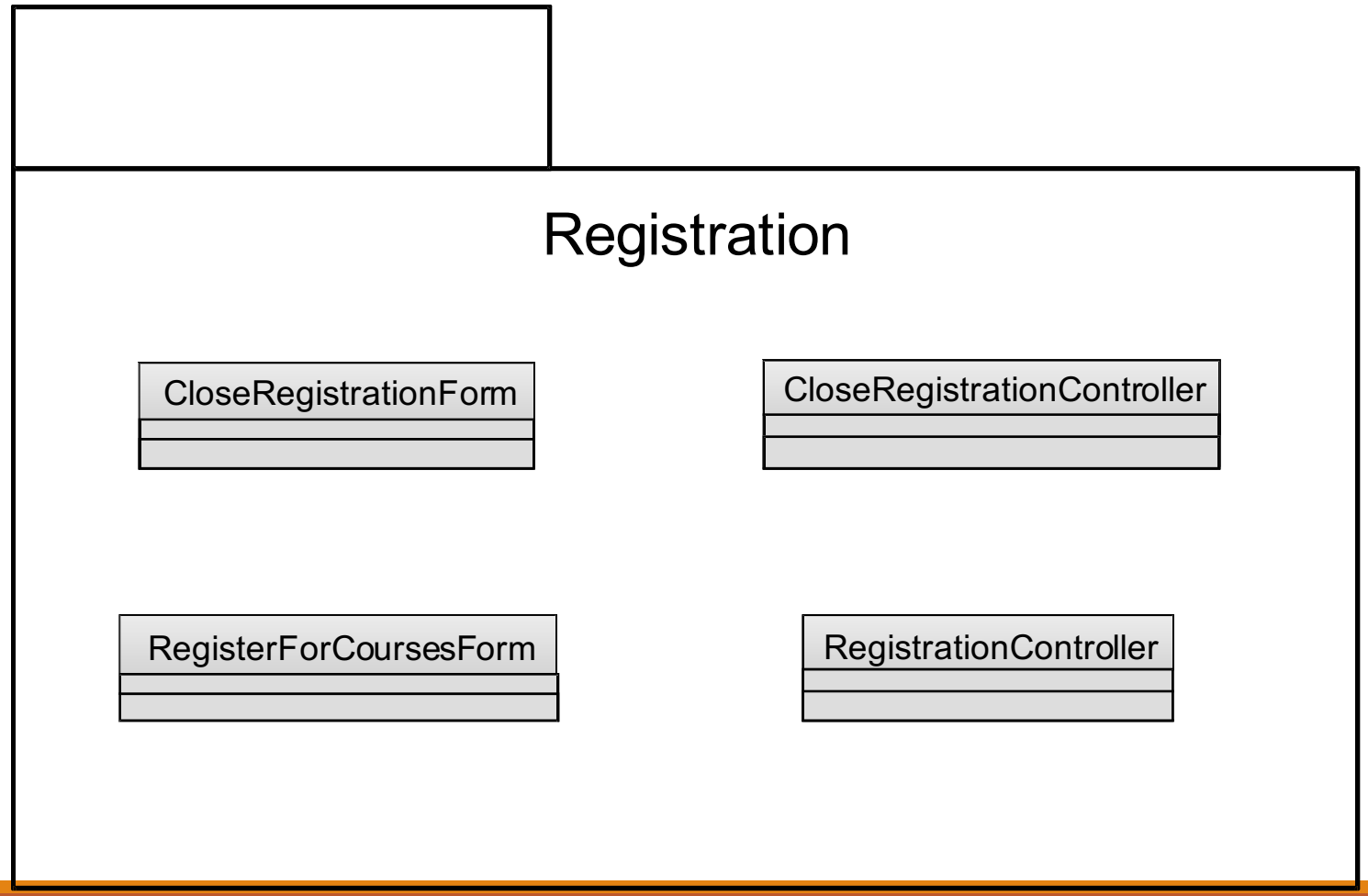
- Trois niveaux de visibilité pour les propriétés :
  - Public (+) : C'est le niveau le plus faible; cela revient à se passer de la notion d'encapsulation et à rendre visibles les attributs et les opérations pour toutes les classes.
  - Protégé (#) : C'est relâcher légèrement le niveau d'encapsulation; l'attribut ou l'opération est visible seulement à l'intérieur de la classe et pour toutes ses sous classes.
  - Privé (-) : C'est le niveau le plus fort; l'attribut ou l'opération est visible seulement à l'intérieur de la classe.

Exemple : Visibilité des attributs et des opérations

- La classe Voiture possède cinq attributs.
- Trois attributs sont des attributs publics : Marque, Puissance et Cylindrée.
- L'attribut Année est un attribut protégé.
- L'attribut ChiffreAffaire est un attribut privé, qui ne sera visible qu'à l'intérieur de la classe.

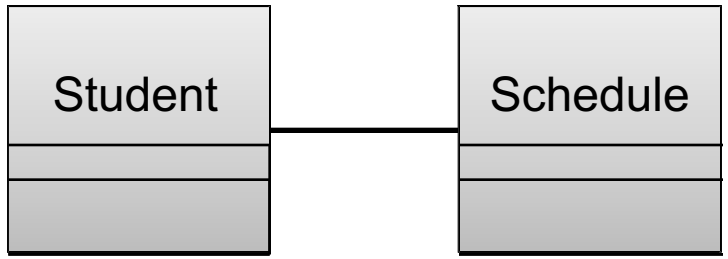
Diagramme de Classe - Organisation

- Les classes sont organisées en pacage



Associations

- Traduisent les relations sémantiques entre deux ou plusieurs classes et spécifiant les connections entre leurs instances.
- Relation structurelle spécifiant que les objets d'un type sont connectés à des objets d'une autre type.



Association entre deux classes distincts

Association entre des instances d'une même classe.

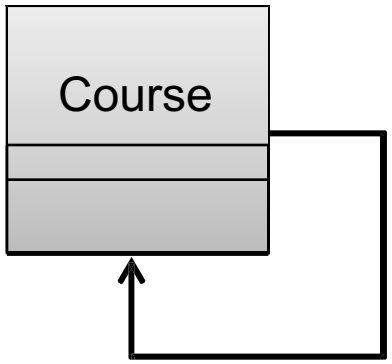
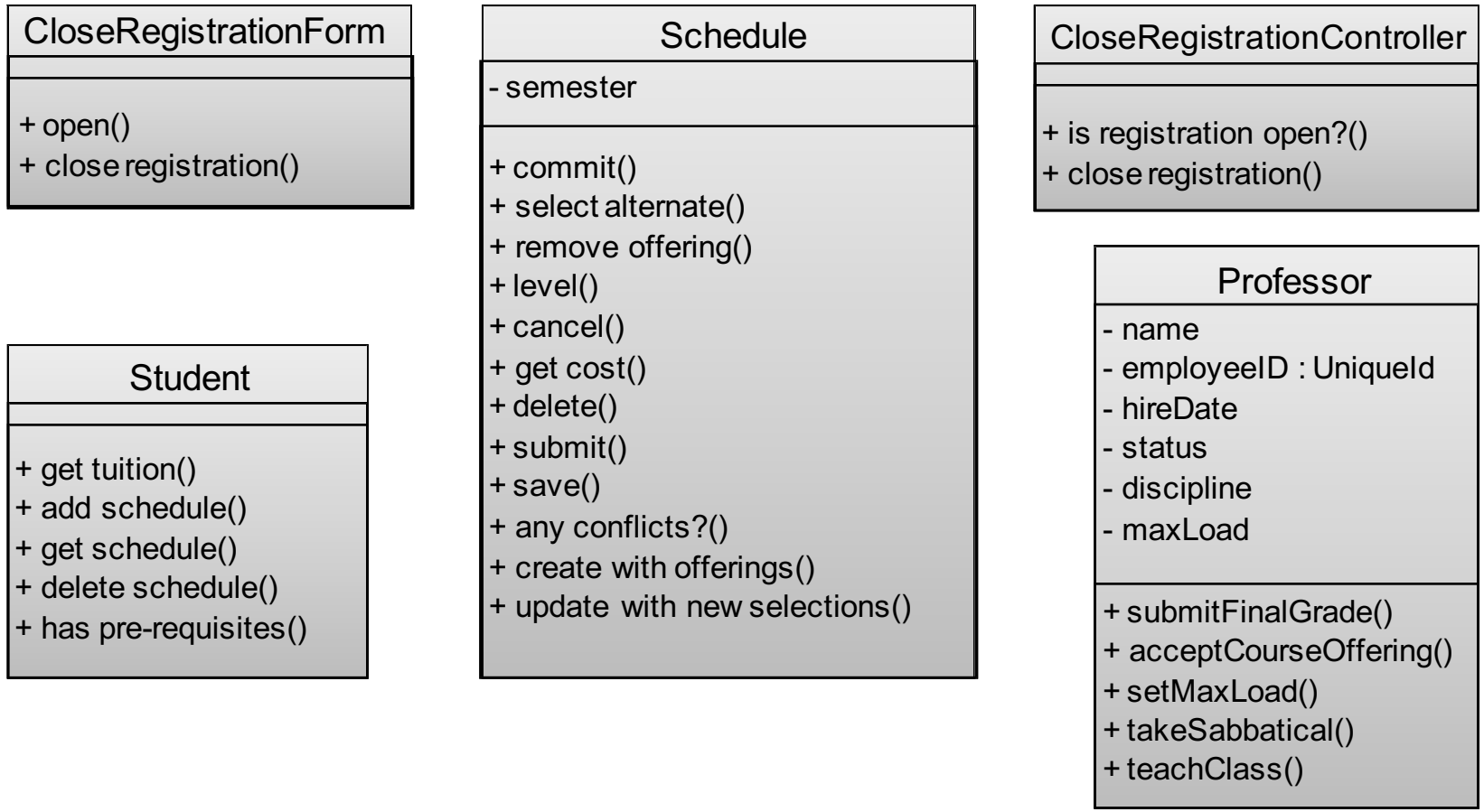


Diagramme structurel

- Vue statique du système

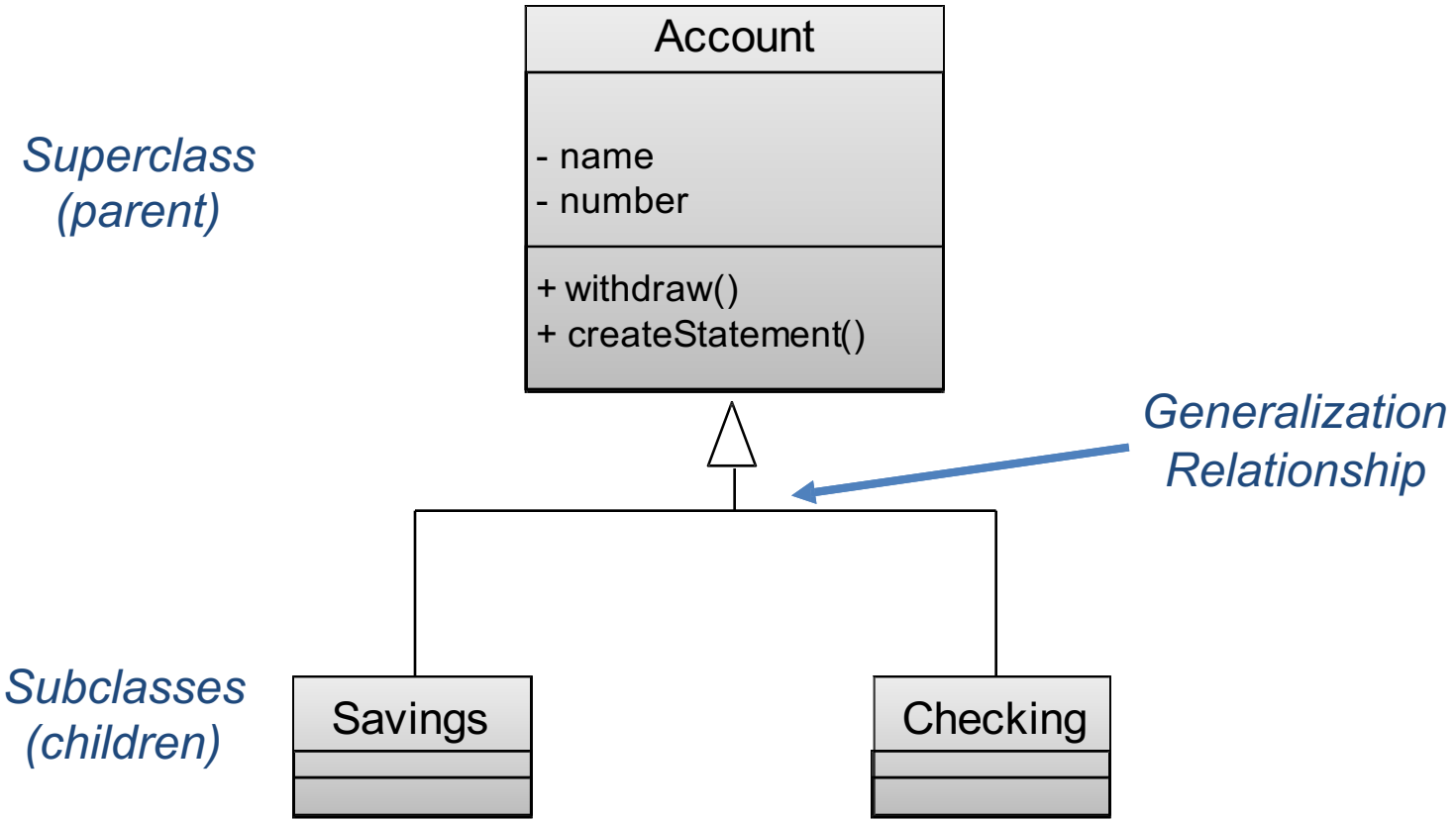


Généralisation - Révision

- Relation entre des classes où l'une des classes partage la structure ou le comportement d'une autre classe
- Défini une hiérarchie d'abstractions où des sous-classes hérite d'une ou plusieurs super classes.
  - Héritage simple
  - Héritage multiple
- Traduit la relation « est un », « est une sorte de »
- La généralisation est une relation non réflexive: une classe ne peut pas dériver d'elle-même
- La généralisation est une relation non symétrique: si une classe B dérive d'une classe A, alors la classe A ne peut pas dériver de la classe B.
- La généralisation est par contre une relation transitive: si C dérive d'une classe B qui dérive elle-même d'une classe A, alors C dérive également de A.

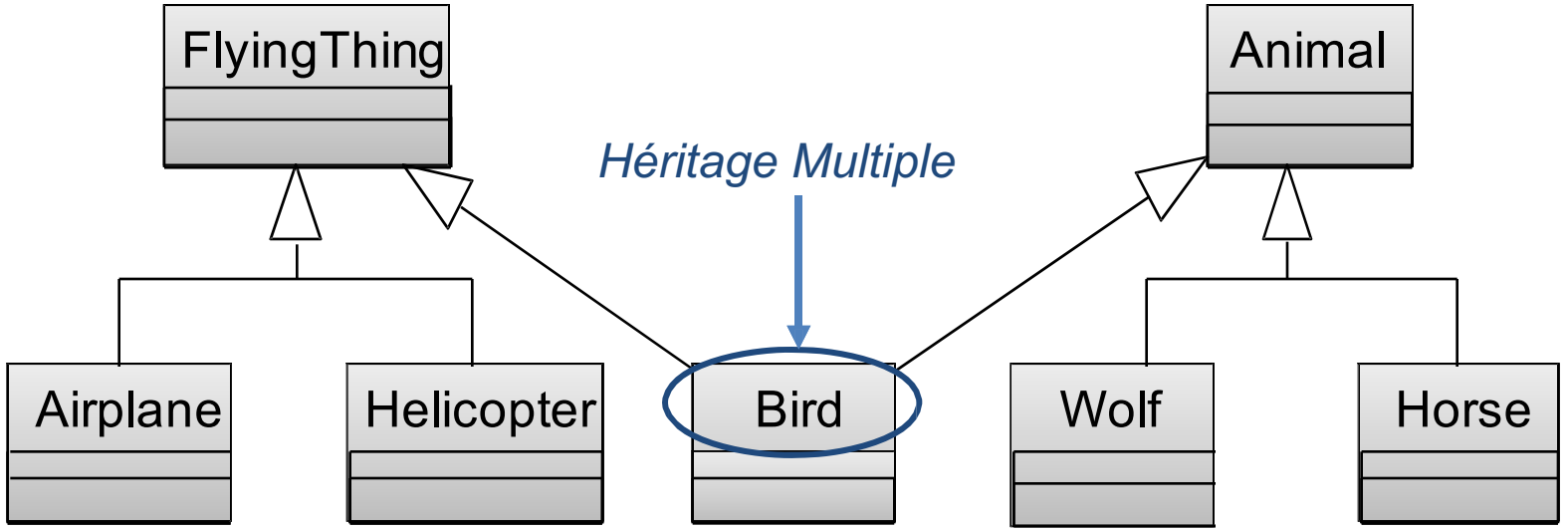
Héritage Simple

- Une classe hérite d'une autre



Héritage Multiple

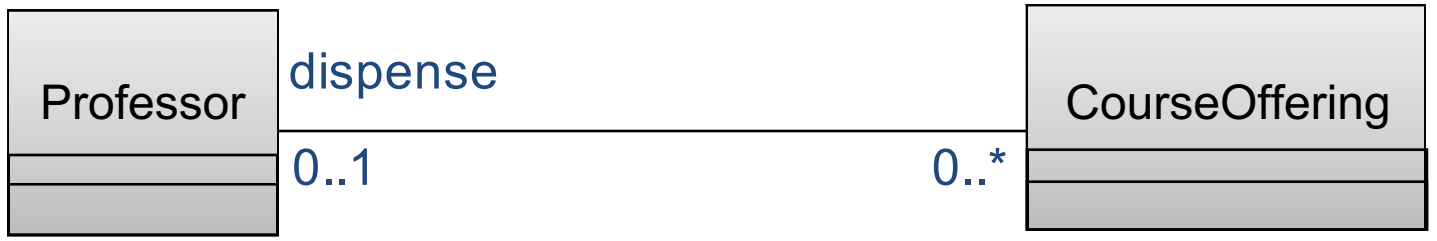
- Une classe hérite d'une ou plusieurs autres classes



- L'héritage multiple est à utiliser avec modération, que si c'est vraiment indispensable.
- N'est pas supporté par la plupart des langages OO (ex. Java)

Associations / Multiplicités

- Nombre d'instances d'une classe pouvant être mis en relation avec les instances d'une autre classe.
- Pour chaque relation, deux multiplicités sont à définir, l'une de chaque côté de l'association
  - Un professeur peut enseigner plusieurs cours
  - Un cours peut être dispensé par 0 ou un professeur.

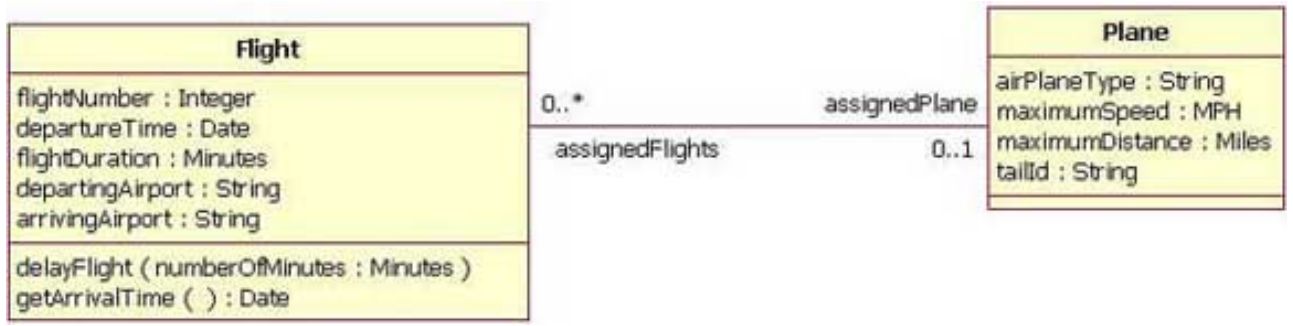


Les différents types de multiplicités

Indicateur	Signification
0…1	0 ou 1
n	Uniquement n (ex 1➔ uniquement 1)
0..*	0 ou plusieurs
*	0 ou plusieurs
1...*	1 ou plusieurs
0..n	0 à n (ex. 0..5 ➔ 0 à 5)

❖Associations bidirectionnelles

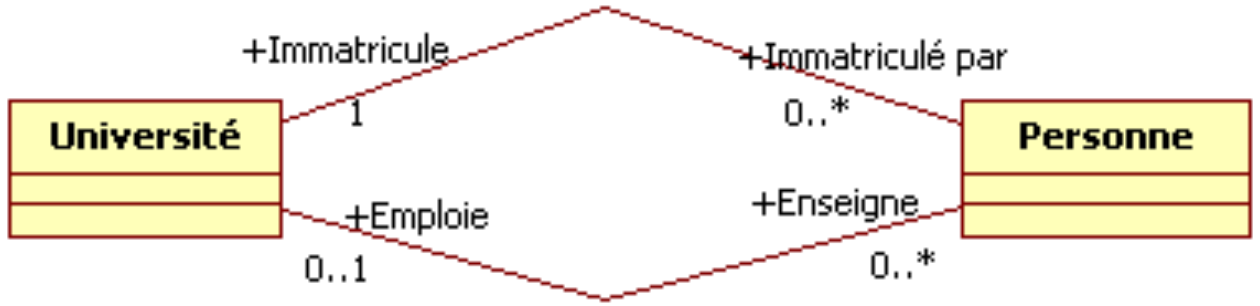
- ❖Cardinalité renseignée de chaque côté de l'association
- ❖Comportement par défaut
- ❖Chaque classe participant à l'association connaît le nombre d'instances que peut contenir l'autre classe



- ❖Un vol peut être assigné à 0 ou plusieurs avions
- ❖A tout instant, un avion n'est assigné qu'à 0 ou un vol

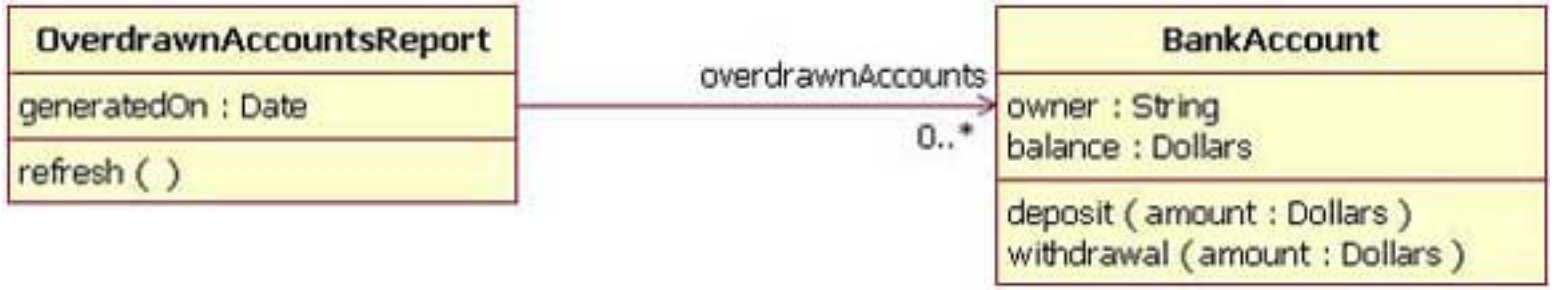
Associations bidirectionnelles (Suite)

- Sur cet exemple on peut lire les multiplicités suivantes :
  - Une Université *Immatricule* zéro ou plusieurs (0..*) Personnes.
  - Une Personne est *ImmatriculéPar* une (1) Université.
  - Une Université *Emploie* zéro ou plusieurs (0.. *) Personnes.
  - Une Personne *Enseigne* dans au plus une (0..1) Université.



Associations unidirectionnelles

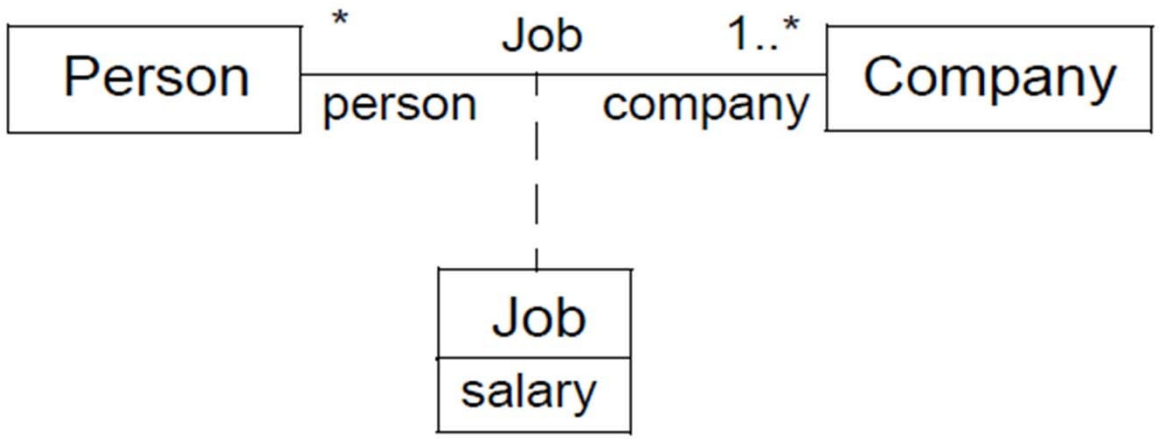
- Cardinalité renseignée que d'un côté de l'association
- Seules les instances du côté de la cardinalité sont au courant du nombre d'instances de l'autre classe



- Un compte bancaire est à découvert 0 ou plusieurs fois
- Mais pour un découvert, on ne connaît pas combien de compte bancaires lui sont associés

Classe d’association

- Une personne est employé dans 1 ou plusieurs compagnies
- Une compagnie emploie 1 ou plusieurs personne
- ➔ Traduit une classe Job permettant de représenter les informations spécifiques liées au travail d’un employé dans une entreprise (ex. salaire, date de prise de fonction, …).



Agrégation

- Type spécifique d'association permettant de traduire la relation « fait partie de »
- Il existe deux types d'agrégation :
  - Agrégation basique
  - Agrégation de composition
  - Traduit qu'une classe fait partie d'une autre mais peut vivre sans l'existance de la classe dont elle fait partie.

  - Dans cet exemple, les pneus de la voiture peuvent être créés avant la voiture (dans une autre compagnie).
  - L'agrégation basique se traduit par un losange non rempli du côté de la classe parent

Agrégation Basique (2)

- Traduit qu'une classe fait partie d'une autre mais peut vivre sans l'existance de la classe dont elle fait partie.

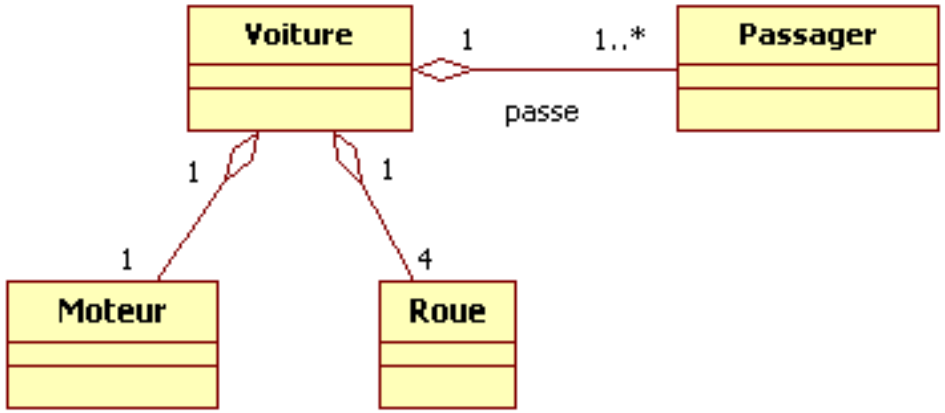
- Dans cet exemple, les pneus de la voiture peuvent être créés avant la voiture (dans une autre compagnie).
- L'agrégation basique se traduit par un losange non rempli du côté de la classe parent

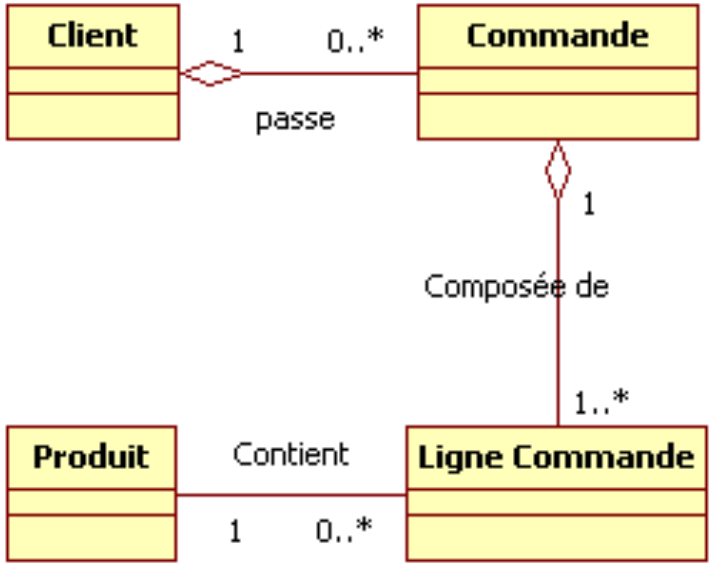
Agrégation De Composition(2)

- Autre forme d'agrégation qui se traduit par le fait que la classe « enfant » ne peut exister indépendamment de son parent.

- Dans cet exemple, un département ne peut exister sans une compagnie.
- Modélisée par un losange rempli du côté de la classe parent

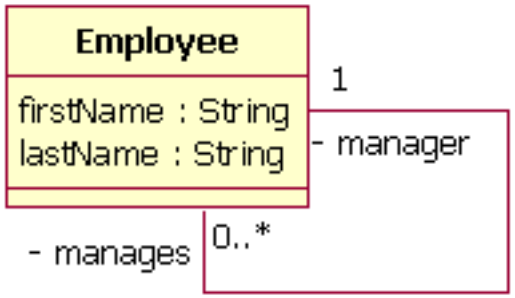
Agrégations - Exercices





Associations réflexives

- Une classe peut s'associer à elle-même
- Cela ne signifie pas que les instances d'une classe sont associées à elles-mêmes;
- Les instances de la classe peuvent s'associer à d'autres instances de la même classe.



- Dans cet exemple, un employé peut être le manager de 0 ou ^plusieurs autres employés.
- Tout employé est géré par exactement 1 manager (qui est aussi un employé)

Associations et Rôles

- Les rôles permettent de nommer les associations entre les classes
- Ils sont utiles pour la lisibilité et la compréhension du diagramme.
- Généralement défini par des verbes.