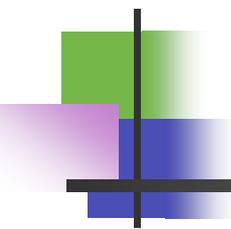
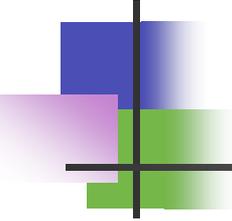


Chapitre 4 : Modélisation statique

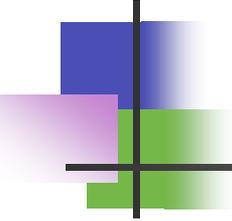


M.BOUABID, 09-2013



Plan

- Diagramme de classes
- Les extensions syntaxiques
- Diagramme objet
- Diagramme de package

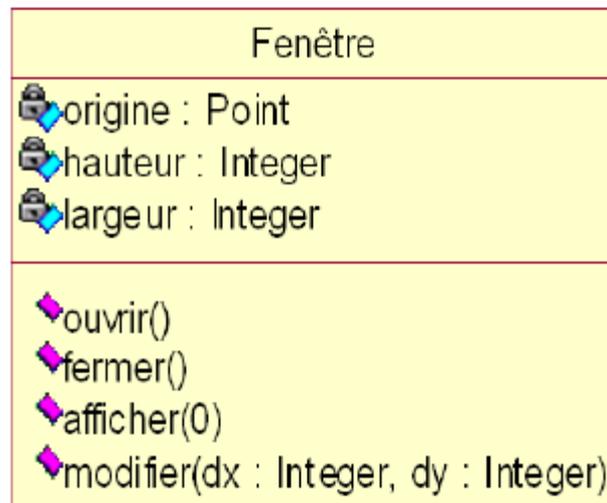


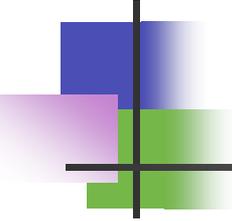
Diagrammes de classes

- Un diagramme de classe présente un ensemble de classes et d'interfaces avec leurs relations
- Il peut être utilisé en phase d'analyse et de conception
 - Conception: offrir une description détaillée L'ensemble des classes utilitaires destinées à assurer le fonctionnement du logiciel
 - Tous les attributs et méthodes doivent apparaître de façon détaillée, avec tous les types de paramètres et les types de retour

Classe

- Une classe est la description d'une collection d'objets qui partagent les mêmes attributs, les mêmes opérations, et la même sémantique





Nommer une classe

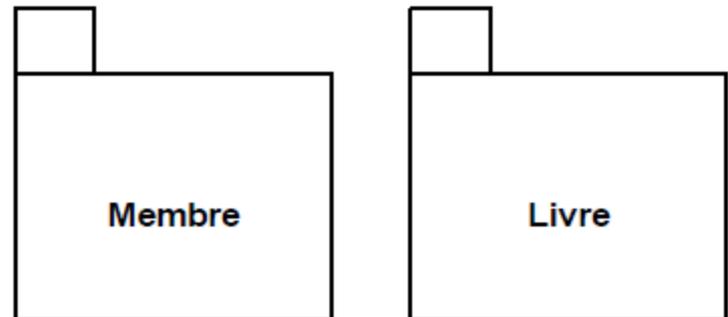
- Chaque classe doit avoir un nom qui la distingue des autres classes (unicité du nom complet)
- En pratique les noms de classes sont des noms pris dans le vocabulaire du domaine
- Il est d'usage de capitaliser la première lettre de chaque mot

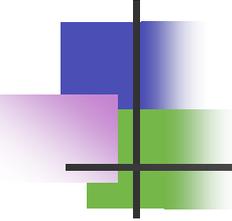
ClientBanque

java::awt::Rectangle

Packages

- Un package (paquetage) est un regroupement d'éléments (classes)
- Le regroupement peut être pour des raisons sémantiques, d'usage, etc.





Visibilité d'une classe

- Classe publique (+)
 - La classe est visible pour tous les éléments qui importent le package englobant la classe
- Classe privée (-)
 - La classe n'est visible que pour les classes qui appartiennent au même package
- Classe protégée (#)
 - La classe est visible pour les classes qui appartiennent au même package ET pour les classes contenues dans les packages enfant du package englobant

Multiplicité

- UML permet de préciser le nombre d'instances d'une classe
 - Zéro instance : classe utilitaire (attribut et opération de classe)
 - Une seule instance : classe singleton (ex: classe PDG)
 - Un nombre spécifique
 - Un nombre quelconque (défaut)

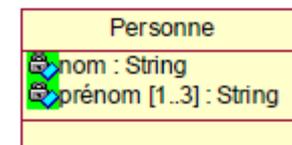


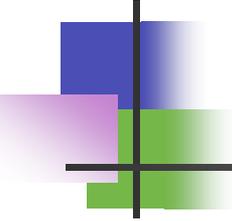
Attribut

- Les attributs forment la description de la partie structurelle ou statique



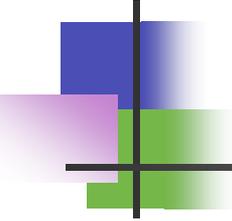
- UML permet de représenter les attributs multivalués en précisant le nombre possible de valeurs pour un attribut





Visibilité d'un attribut

- Attribut public (+)
 - L'attribut est visible pour toutes les classes du package englobant
- Attribut privé (-)
 - L'attribut n'est visible que pour la classe elle-même
- Attribut protégé (#)
 - L'attribut est visible pour toutes les sous-classes

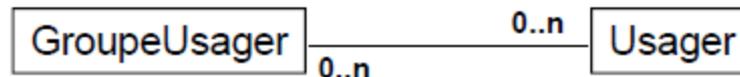


Portée d'un attribut

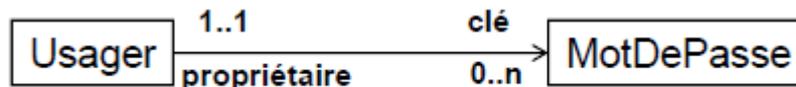
- Attribut d'instance
 - Attribut dont la valeur peut varier pour chaque instance de la classe
- Attribut de classe
 - Attribut propre à la classe dont la valeur est fixe pour toutes les instances de la classe

Associations (1/2)

- Par défaut une association est bidirectionnelle
 - Navigation dans les deux sens

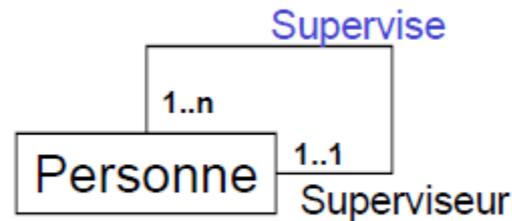


- UML permet de spécifier une limitation de la navigation
 - La flèche indique que les instances d'une classe ne "connaissent" pas les instances d'une autre



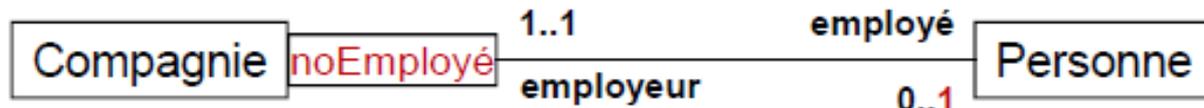
Association (2/2)

- Association peut être aussi réflexive

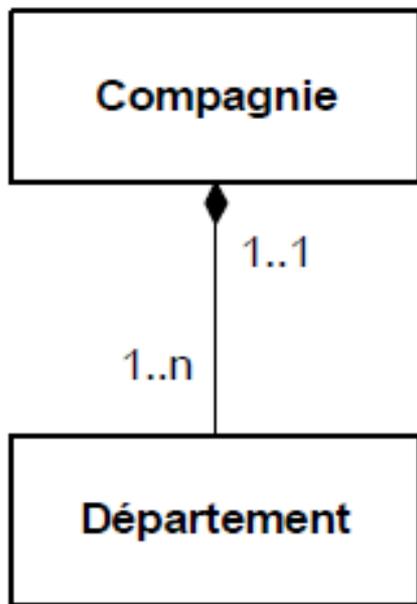


Qualification

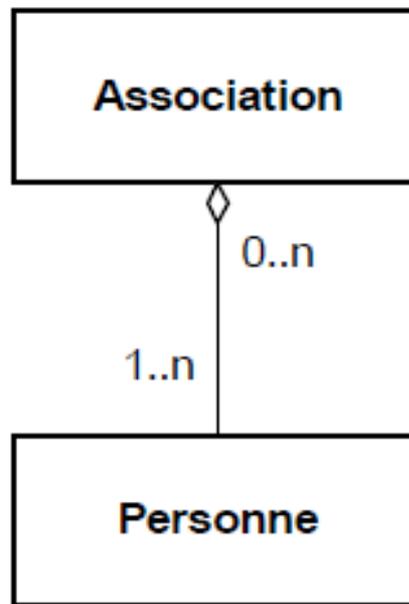
- UML permet de spécifier comment limiter le nombre d'objets en cas de navigation d'associations
- La connaissance de l'objet source et du qualificateur permet de limiter le nombre d'objets liés



Agrégation vs. Composition



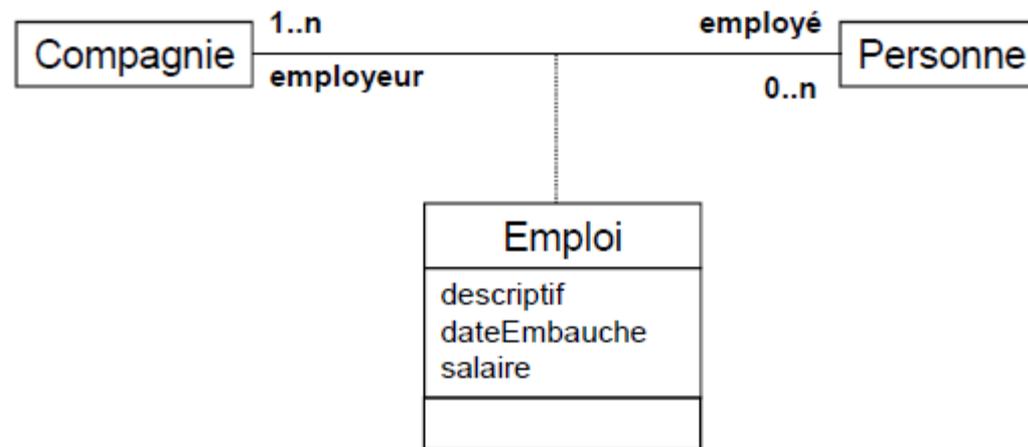
Composition



Agrégation

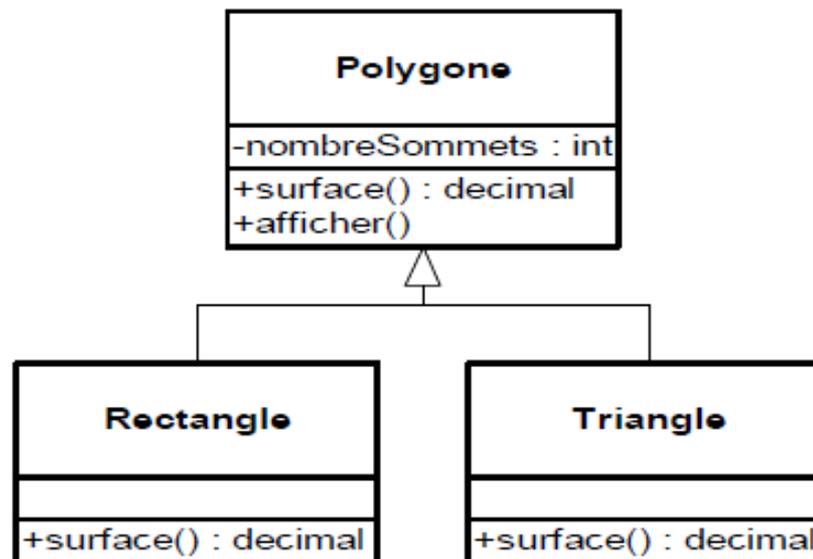
Classe d'association

- UML permet de modéliser les propriétés d'une association entre deux classes par une classe d'association



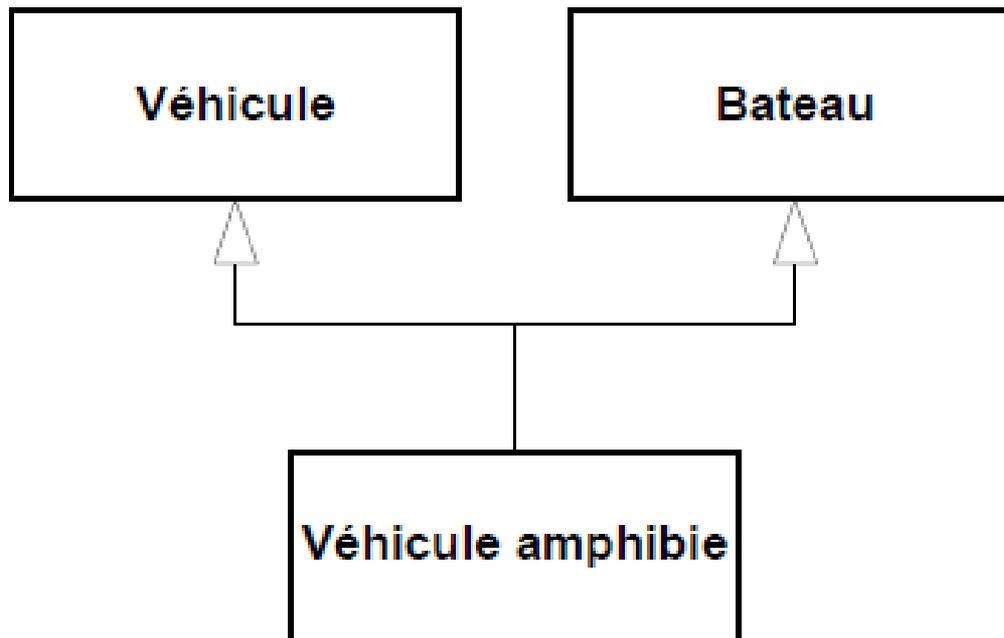
Héritage

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



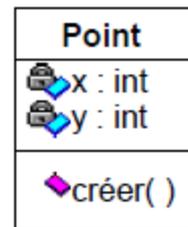
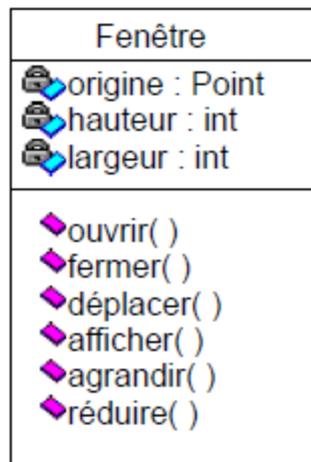
Héritage multiple

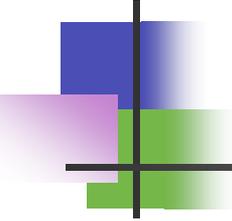
- L'héritage multiple permet à une classe d'hériter de plusieurs classes



Opération

- Les opérations décrivent la partie comportementale ou dynamique



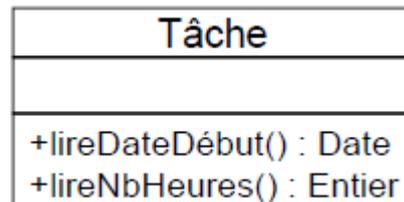


Visibilité d'une opération

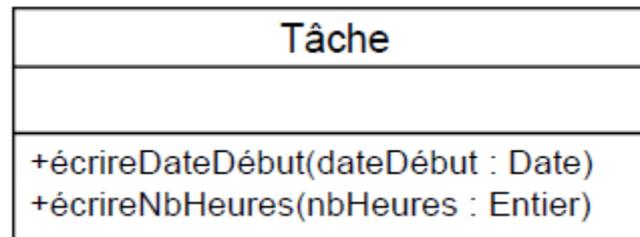
- Opération publique (+)
 - L'opération est visible pour toutes les classes du package englobant
- Opération privée (-)
 - L'opération n'est visible que pour la classe elle-même
- Opération protégée (#)
 - L'opération est visible pour toutes les sous-classes

Valeur de retour & paramètres

- Une opération peut retourner une valeur simple spécifiée par son type

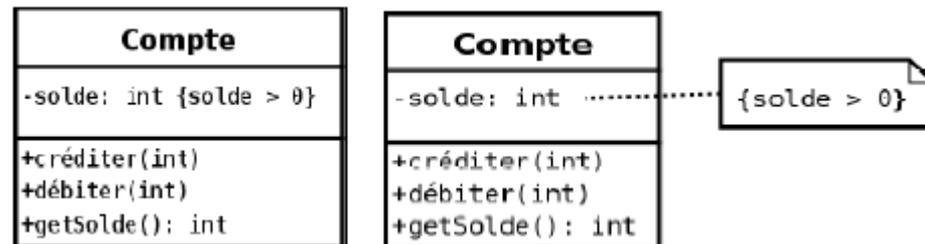


- Une opération peut avoir des paramètres d'entrée et de sortie

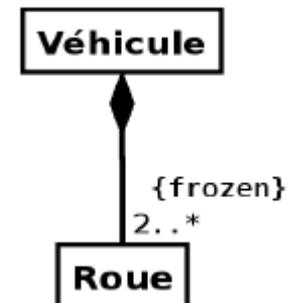


Contraintes sur les associations (1/2)

- Une contrainte est une règle de gestion exprimée entre { } ou avec OCL
 - Contrainte sur un attribut qui doit être positif

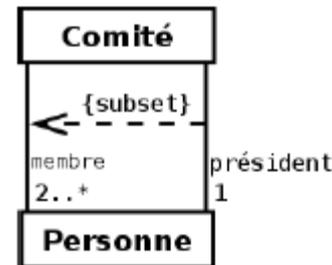


- Contrainte {frozen} précise que le nombre de roues d'un véhicule ne peut pas varier

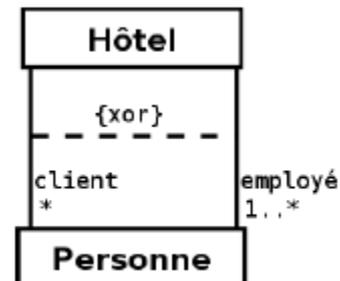


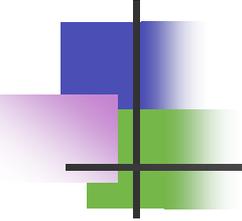
Contraintes sur les associations (2/2)

- Contrainte {subset} précise que le président est également un membre du comité



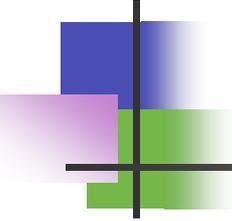
- Contrainte {xor} (ou exclusif) précise que les employés de l'hôtel n'ont pas le droit de prendre une chambre dans ce même hôtel





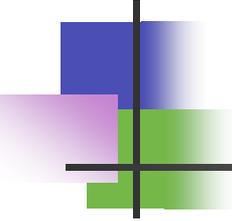
Exemple -étude de cas : un système simplifié de réservation de vols pour une agence de voyage

- 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



Objet

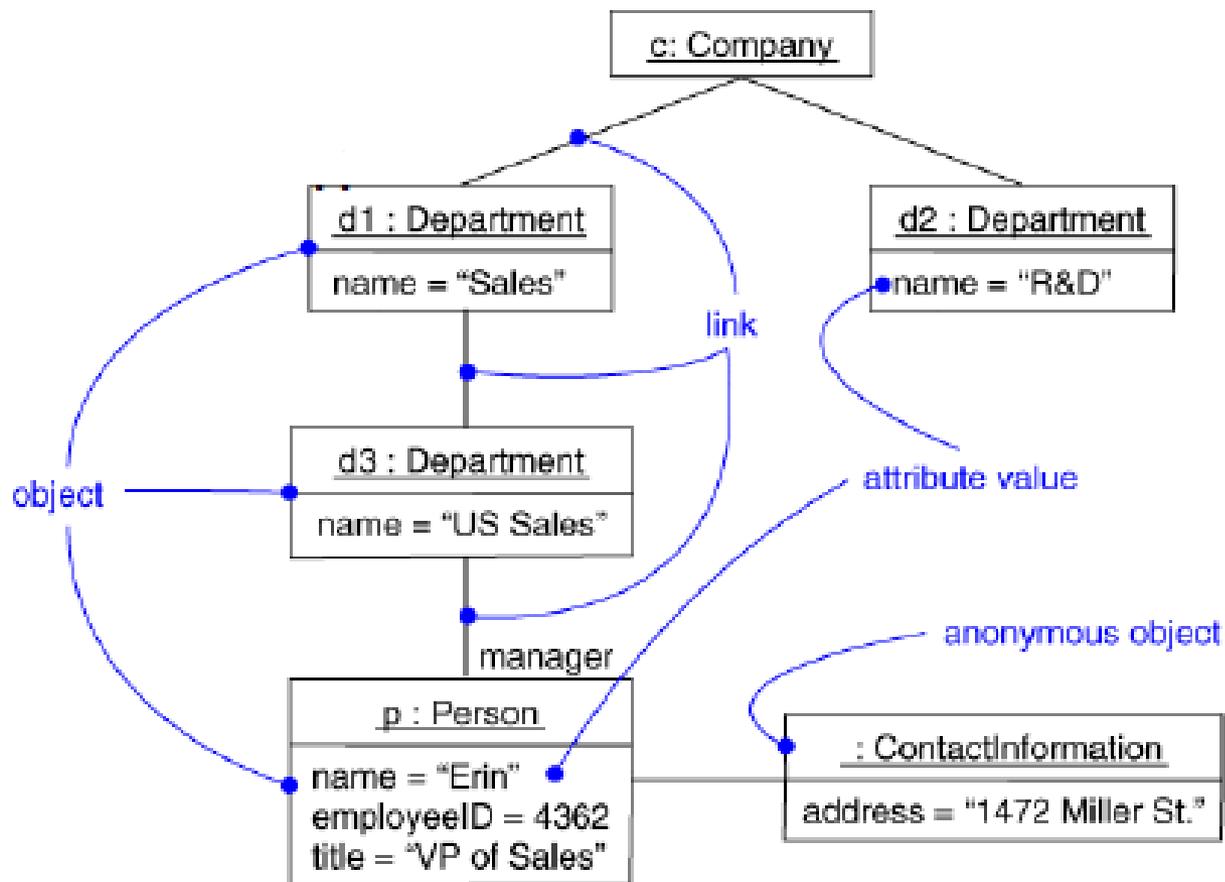
- Un diagramme d'objets présente un ensemble d'objets avec les liens qui les associent
 - Un objet est une instance d'une classe
 - Les objets représentent les concepts ou entités du monde réel



Lien

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

Représentation graphique d'un objet



Exemple : diagramme de package

