

Centre universitaire
Abdelhafid Boussouf
Mila

Faculté des sciences et de la
technologie

Département math et
informatique



Génie logiciel

Chapitre 4

Diagramme de classe & diagramme d'objet

Mme. S.HEDJAZ

II. Diagramme de classe/objet



01

Définitions et principes de base

02

Représentation d'une classe

03

Les types des relations entre classe

04

Diagramme d'objet



Définitions et principes de base

1- Objectif:

- Le diagramme de classes est sans doute le diagramme le plus important à représenter pour les méthodes d'analyse orientées objet. C'est le point central de tout développement orienté objet.
- Le diagramme de classes exprime la structure statique du système en termes de classes et de relations entre ces classes
- L'intérêt du diagramme de classe est de modéliser les entités du système d'information



Définitions et principes de base

2- Définition:

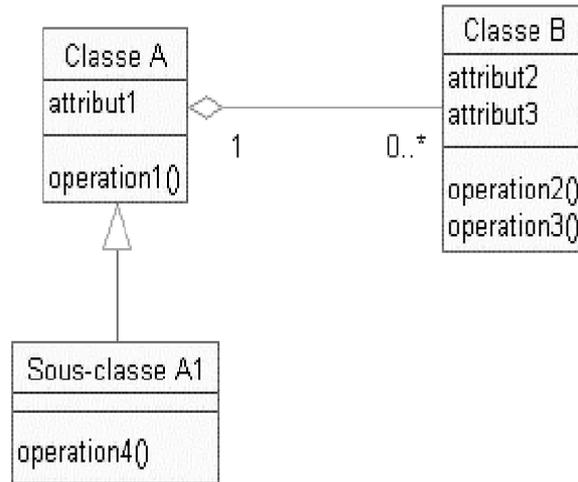
- **Un diagramme de classes** est une collection d'éléments de modélisation statique qui montre la structure d'un modèle.
- **Un diagramme de classes** fait abstraction des aspects dynamiques et temporels du système
- **Une classe** est une description abstraite d'un ensemble d'objets du domaine de l'application : elle définit leur structure, leur comportement et leurs relations
- **Une classe** représente la description d'un ensemble d'objets possédant les mêmes caractéristiques.



Définitions et principes de base

3- Représentation:

Le diagramme de classes met en œuvre **des classes**, contenant des **attributs** et des **opérations**, et reliées par **des associations** ou des **généralisations**.





Représentation d'une classe

3.1 - Classe:

Une classe est représentée par un rectangle séparé en trois parties:

- La première partie représente **le nom de la classe**
- La deuxième partie représente **les attributs de la classe**
- La troisième partie représente **les opérations de la classe**

a. Formalisme:

NOM CLASSE
Attribut_1 : int
Attribut_2 : int
Attribut_3 : int
Operation_1 () : void
Operation_2 () : void

b. Exemple:

Personne
Idpersonne:string Nom:string Prénom:string
Saisir Info() CalculerSalaire():float



Représentation d'une classe

3.2 - Attribut:

- ✓ Un attribut représente la modélisation d'une information élémentaire représentée par son nom et son format
- ✓ UML définit 3 niveaux de visibilité pour les attributs :
 - **public (+)** : l'élément est visible pour tous les clients de la classe
 - **protégé (#)** : l'élément est visible pour les sous-classes de la classe
 - **privé (-)** : l'élément n'est visible que par les objets de la classe dans laquelle il est déclaré
- ✓ L'identifiant est un attribut particulier, qui permet de repérer de façon unique chaque objet, instance de la classe.
- ✓ Par commodité de gestion, on choisit parfois de conserver dans un attribut le résultat d'un calcul effectué à partir d'autres classes : on parle alors **d'attribut dérivé**. Pour repérer un attribut dérivé : on place un « / » devant son nom.



Représentation d'une classe

3.2 - Opération:

- ✓ l'opération représente un élément de comportement des objets, défini de manière globale dans la classe.
- ✓ Une opération est une fonctionnalité assurée par une classe. La description des opérations peut préciser les paramètres d'entrée et de sortie ainsi que les actions élémentaires à exécuter.
- ✓ Comme pour les attributs, on retrouve 3 niveaux de visibilité pour les opérations

FACTURE	
+ No facture	: int
+ Date	: Date
+ Montant	: double
+ / Montant TVA	: double
+ Op publique ()	
# Op protégée ()	
- Op privée ()	



Les types des relations entre classe

✓ Il existe plusieurs types de relations entre classes :

1 - Association:

- ✓ Une association est une relation entre deux classes qui décrit les connexions structurelles entre leurs instances. Une association indique donc qu'il peut y avoir des liens entre des instances des classes associées.
- ✓ Une association est une relation statique n-aire (le plus souvent : elle est binaire) : c'est à dire qu'elle relie plusieurs classes entre elles.
- ✓ Une association n-aire possède n rôles qui sont les points terminaux de l'association. Chaque classe qui participe à l'association joue un rôle. Les rôles sont définis par 2 propriétés :
 - Nom de rôle = indication sur la participation de la classe à l'association
 - Multiplicité = définit le nombre d'instances de l'association pour une instance de la classe.

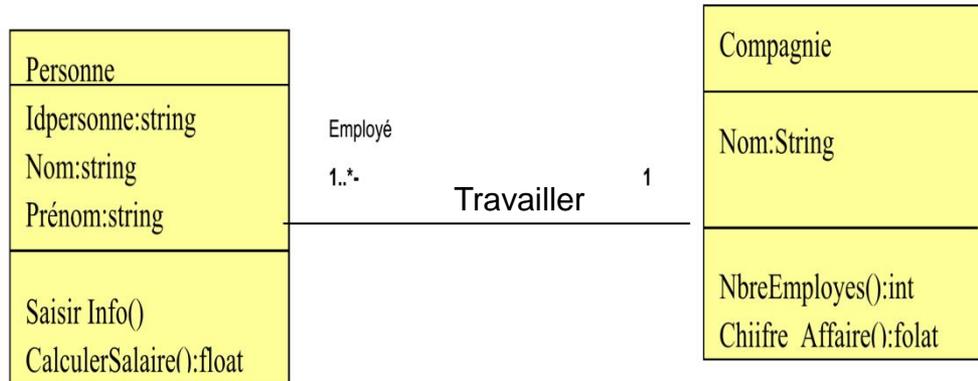
03

Les types des relations entre classe

1.1 - Exemple:

Dans ce diagramme, le nom de l'association est Travailler, le nom du rôle de la classe Personne pour l'association est: Employé. Le diagramme se lit comme suit :

- ✓ Une personne travail pour une seule entreprise.
- ✓ Dans une entreprise il y 'a une à plusieurs personnes:



1	un et un seul
0..1	Zéro ou un
m..n	De m à n (entier)
*	Plusieurs
0..*	De zéro à plusieurs
1..*	d'un à plusieurs



Les types des relations entre classe

1.2 - La navigabilité:

- Une navigabilité placée sur une terminaison cible indique si ce rôle est accessible à partir de la source.
- Par défaut les associations sont navigables dans les 2 sens. Dans certains cas, une seule direction de navigation est utile : l'extrémité d'association vers laquelle la navigation est possible porte alors une flèche.
- Dans l'exemple ci-dessous, les instances de A voient les instances de B mais les instances de B ne voient pas les instances de A.

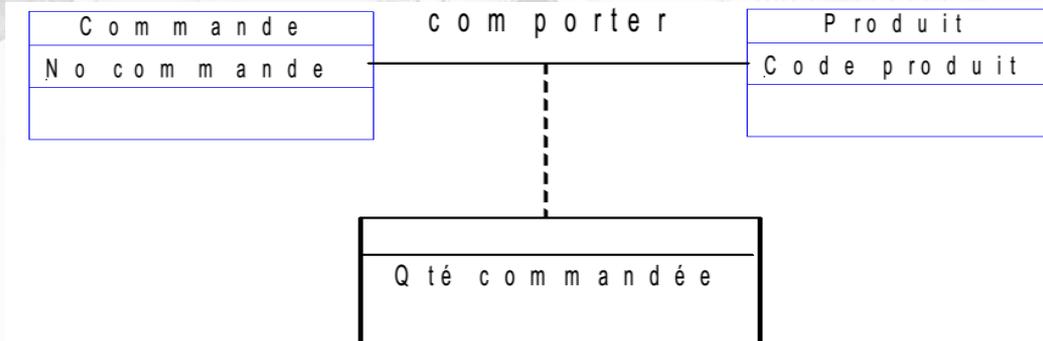




Les types des relations entre classe

1.3 – La classe association :

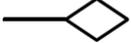
- Les attributs d'une classe dépendent fonctionnellement de l'identifiant de la classe. Parfois, un attribut dépend fonctionnellement de deux identifiants, appartenant à deux classes différentes.
- Par exemple, l'attribut « quantité commandée » dépend fonctionnellement du numéro de commande et du « code produit ».
- On va donc placer l'attribut « quantité commandée » dans l'association « comporter ».
- Dans ce cas, l'association est dite « porteuse d'attributs ».

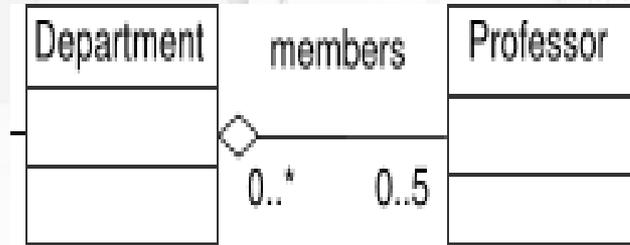


03

Les types des relations entre classe

1.4 – L'agrégation:

- L'agrégation représente une association non symétrique dans laquelle une des extrémités joue un rôle prédominant par rapport à l'autre extrémité.
- L'agrégation se représente toujours avec un petit losange du côté de l'agrégat. 
- Elle est asymétrique du type « ensemble / élément » ou « contenant / contenu »

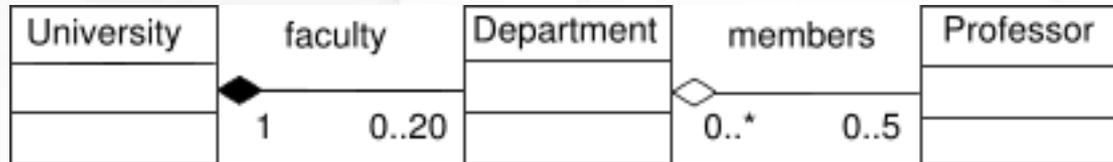


03

Les types des relations entre classe

1.5- La composition:

- La composition est un cas particulier de l'agrégation dans laquelle la vie des composants est liée à celle des agrégats. Elle fait souvent référence à une contenance physique.
- La composition implique, en plus de l'agrégation, une coïncidence des durées de vie des composants : la destruction de l'agrégat (ou conteneur) implique automatiquement la destruction de tous les composants liés





Les types des relations entre classe

2- La généralisation / spécialisation :

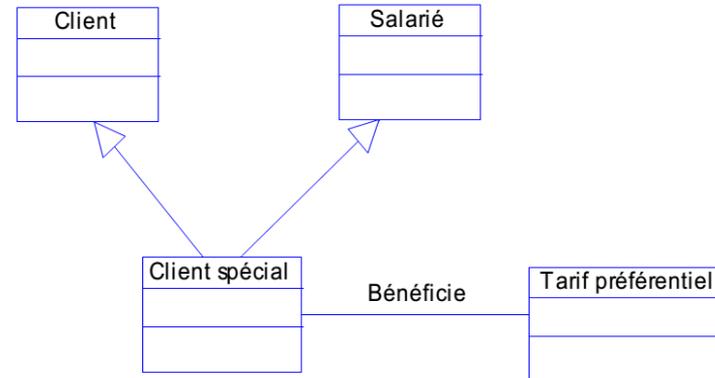
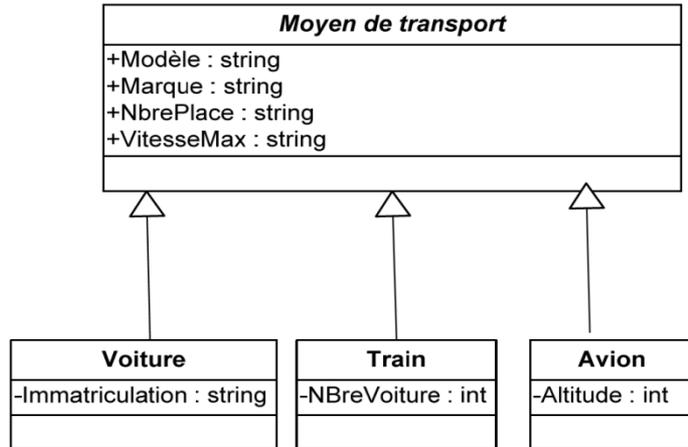
- Le principe de **généralisation / spécialisation** permet d'identifier parmi les objets d'une classe (générique) des sous-ensembles d'objets (des classes spécialisées) ayant des définitions spécifiques.
- La classe plus spécifique (appelée aussi classe fille, classe dérivée, classe spécialisée, classe descendante ...) est cohérente avec la classe plus générale (appelée aussi classe mère, classe générale ...), c'est-à-dire qu'elle contient par héritage tous les attributs, les membres, les relations de la classe générale, et peut contenir d'autres.
- Une classe peut avoir plusieurs parents, on parle alors **d'héritage multiple**
- Une **classe abstraite** est une classe qui ne s'instancie pas directement, mais qui représente une simple abstraction afin de factoriser les propriétés communes des sous-classes. **Elle se note en italique.**

03

Les types des relations entre classe

2.1 - Exemples:

- La classe « *Moyen de transport* » est une classe abstraite
- La généralisation multiple consiste à fusionner plusieurs classes en une seule classe. La classe « client spécial » est une spécialisation de client et de salarié. Ce modèle permet d'indiquer que l'on accorde des tarifs préférentiels aux salariés.





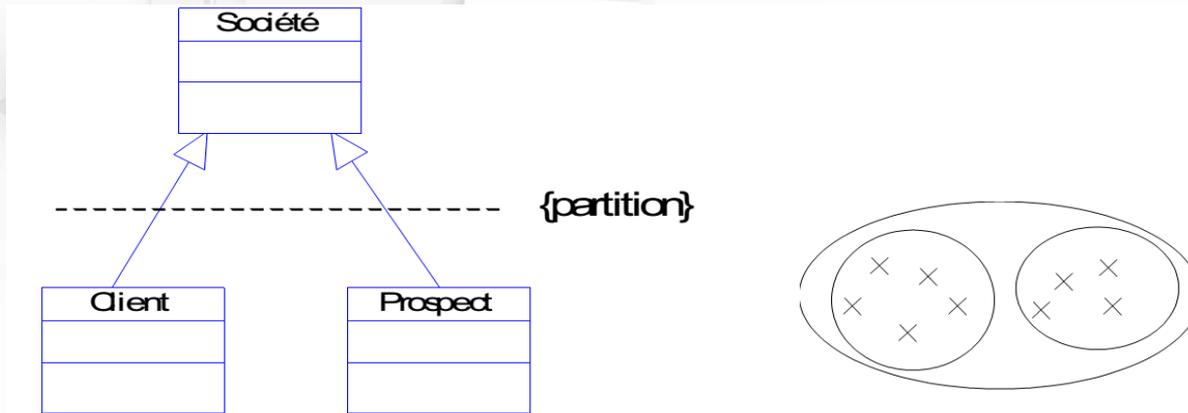
Les types des relations entre classe

2.2 – Contraintes sur les associations:

- Il existe plusieurs types de contraintes sur les associations :

a– Contrainte de partition:

- Elle indique que toutes les instances d'une classe correspondent à une et une seule instance des classes liées.
- Toutes les sociétés sont soit clientes, soit considérées comme des prospects.

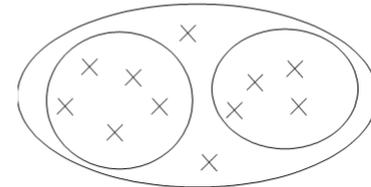
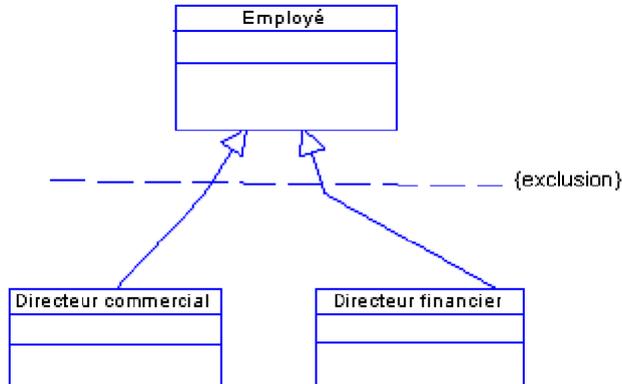


03

Les types des relations entre classe

b- Contrainte d'exclusion:

- Elle permet de préciser qu'une instance d'association exclut une autre instance.
- Par exemple, un employé ne peut être à la fois directeur financier et directeur commercial.

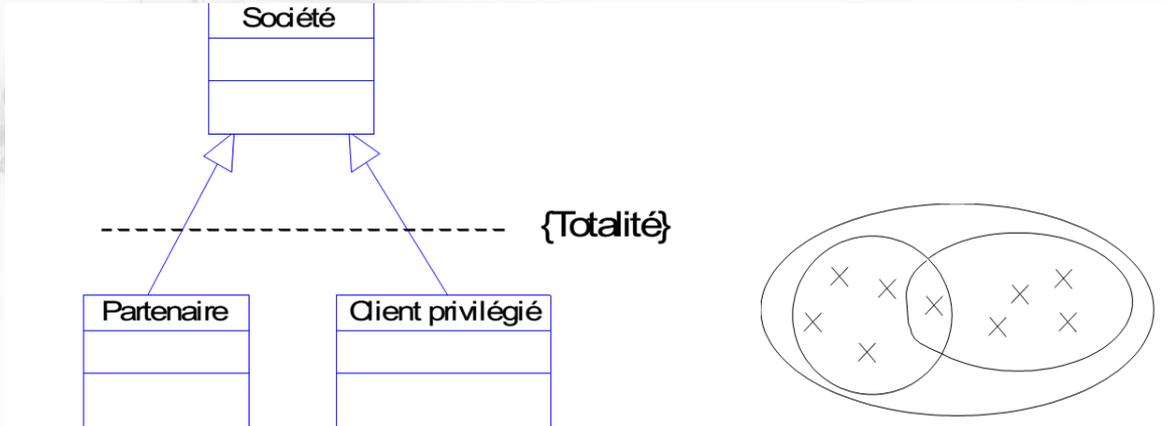


03

Les types des relations entre classe

b- Contrainte de totalité:

- Toutes les instances d'une classe correspondent au moins à une des instances des classes liées.
- Toute société est au moins partenaire ou client privilégiée. Et elle peut être les 2 à la fois.





Les types des relations entre classe

b- Contrainte d'inclusion:

- Elle permet de préciser qu'une collection est incluse dans une autre collection. (la flèche de la relation de dépendance indique le sens de la contrainte).
- Par exemple, on pourra indiquer que le contractant d'un contrat fait obligatoirement partie des individus assurés

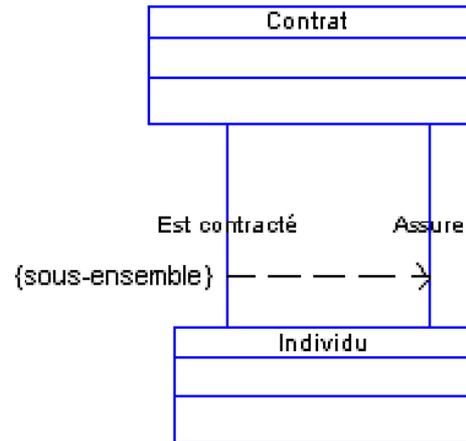




Diagramme d'objet

1- Définition:

- Le diagramme d'objets permet de mettre en évidence des liens entre les objets. Les objets, instances de classes, sont reliés par des liens, instances d'associations.
- A l'exception de la multiplicité, qui est explicitement indiquée, le diagramme d'objets utilise les mêmes concepts que le diagramme de classes. Ils sont essentiellement utilisés pour comprendre ou illustrer des parties complexes d'un diagramme de classes.

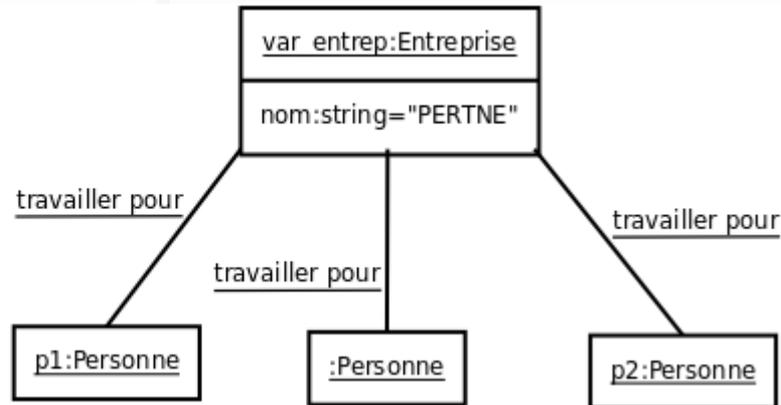
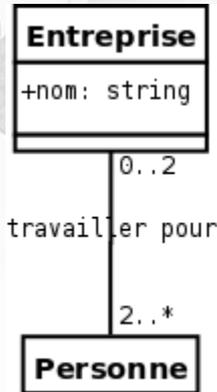




Diagramme d'objet

1- Représentation:

- Graphiquement, un objet se représente comme une classe. Cependant, le compartiment des opérations n'est pas utile. De plus, le nom de la classe dont l'objet est une instance est **précédé d'un << : >> et est souligné**. Pour différencier les objets d'une même classe, leur identifiant peut être ajouté devant le nom de la classe.
- Dans un diagramme d'objets, les relations du diagramme de **classes deviennent des liens**. La relation de généralisation ne possède pas d'instance, elle n'est donc jamais représentée dans un diagramme d'objets. Graphiquement, un lien se représente comme une relation, mais, s'il y a un nom, il est souligné.

Exercice

Considérons les phrases suivantes :

1. Un répertoire contient des fichiers
2. Une pièce contient des murs
3. Les modems et les claviers sont des périphériques d'entrées/sorties
4. Une transaction boursière est un achat ou une vente
5. Un compte bancaire peut appartenir à une personne physique ou morale
6. Deux personnes peuvent être mariées
7. Une personne utilise un langage de programmation dans un projet
8. Un payé possède une capitale
9. Mohammed programme son simulateur de vol en java
10. Java, Eiffel sont des langages orientés objet

- Déterminer la relation statique appropriée (généralisation, instanciation, agrégation, composition, lien ou association) dans chaque phrase de l'énoncé précédent et déterminer le degré de la relation dans le cas d'une association.

- Dessinez les diagrammes de classes correspondants aux phases 4, 6, 8, 9.

Bibliographies

- **Uml 2 pratique de la modélisation**, Benoît Charroux, Yann Thierry-Mieg, Aomar Osmani
Ni <https://fr.slideshare.net/nassimamine3994/uml-2-pratique-de-la-modlisation>
- **Uml 2 par la pratique**, Pascal roques
- **Les cahiers du programmeur**, Pascal roques
- **Uml en action**, Pascal roques
-