

LE DIAGRAMME DE CLASSES

Partie 1

Introduction	2
Objet	3
Classe	3
Attribut et opération	5
Visibilité des attributs et opérations	6
Les associations entre classes	7
Les classes-associations	11
Qualification d'association ou association qualifiée	12
Association de dimension supérieure à 2 ou association n-aire	12

Introduction

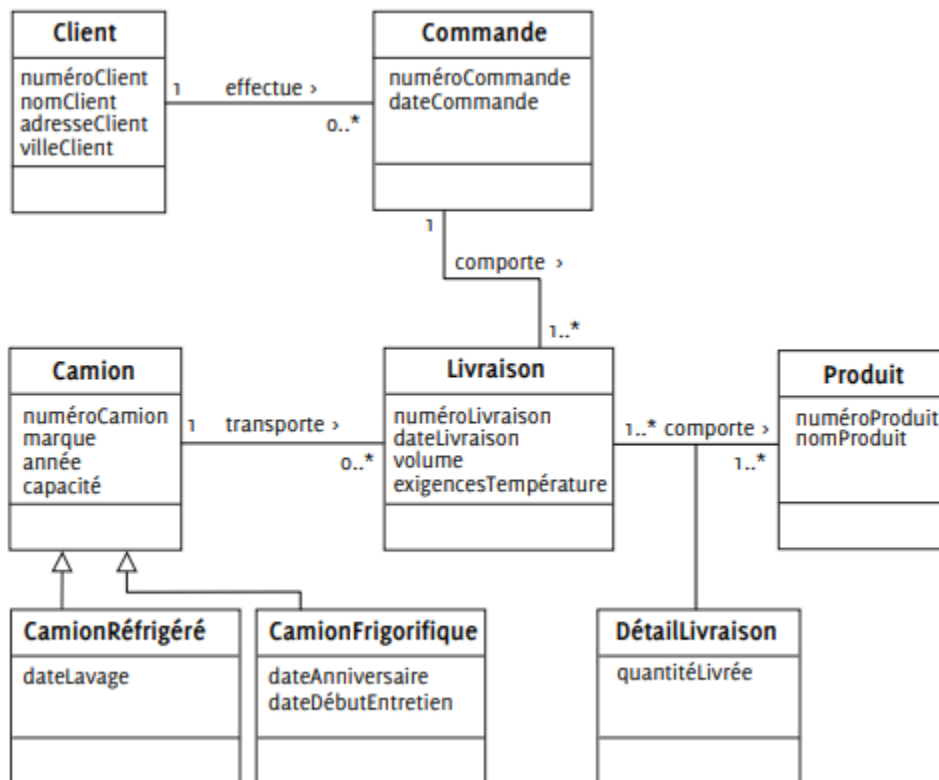
Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet. Le diagramme de classes montre la structure interne du système à développer.

Le diagramme de classes est un modèle qui représente les classes et leurs associations.

Dans sa version complète, il comporte les classes, leurs attributs, leurs opérations et leurs associations. Une version plus restreinte omettra les opérations. En version de travail, l'analyste pourra même omettre d'indiquer les attributs et ne représenter que les classes et les associations.

L'exemple suivant montre un diagramme de classes modélisant un processus de livraison dans une entreprise de livraison à l'aide de 8 classes : Client, Commande, Camion, Livraison, Produit, CamionRéfrigéré, CamionFrigorifique, DétailLivraison.

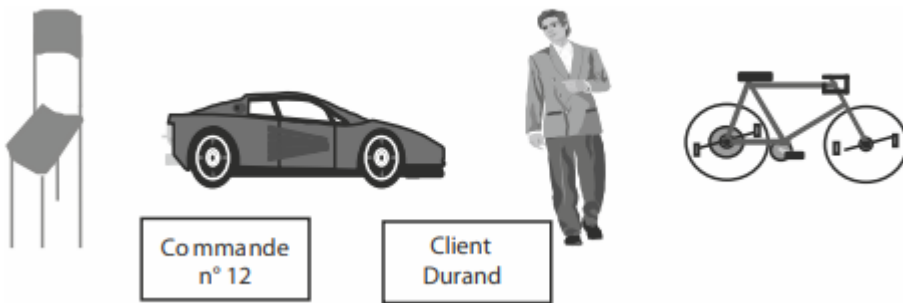
Les classes possèdent des attributs (la classe Client est caractérisé par un numéro, un nom client, une adresse client, et une ville). Les classes sont reliées entre elles par des associations symbolisées par des traits et des flèches.



Objet

Un objet est un concept, une abstraction ou une chose qui a un sens dans le contexte du système à modéliser. Chaque objet a une identité et peut être distingué des autres sans considérer a priori les valeurs de ses propriétés.

L'exemple suivant montre des objets physiques (Chaise, voiture, Personne, Vélo) et des objets de gestion (Commande, Client).



Un objet est caractérisé par les valeurs de ses propriétés qui lui confèrent des états significatifs suivant les instants considérés.

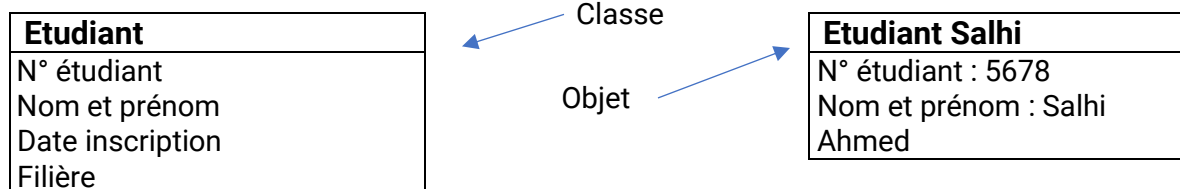
Classe

L'élaboration d'un diagramme de classes est une activité itérative qui requiert une bonne connaissance du processus et de l'organisation dans lesquels le système s'inscrit. Première chose à faire est d'identifier les classes.

Qu'est-ce qu'une classe ?

Une classe décrit un groupe d'objets ayant les mêmes propriétés (attributs), un même comportement (opérations).

Un objet est une instance d'une classe. Un objet d'une classe doit avoir **des valeurs uniques** pour tous les attributs définis dans la classe et doit pouvoir exécuter toutes les méthodes de la classe.



Groupe
sinscrire() Modifier_informations() Changer_filiere() Changer_groupe()

Date inscription : 03/09/2021 Filière : Mathématique Groupe : A

En UML, la représentation d'une classe est constituée de trois parties. La première partie contient le nom de la classe. La deuxième partie contient les attributs. La troisième partie contient les opérations.

Nom de la classe
Attributs
Opérations

Dans un diagramme de classes, une classe peut être représentée par ses attributs et ses opérations, et c'est possible d'omettre les opérations et d'une façon réduite, une classe, peut être représentée que par son nom.

Voici les trois façons de représentation d'une classe :

Nom de la classe
Attributs
Opérations

Nom de la classe
Attributs

Nom de la classe

Soit l'exemple suivant, dans lequel, nous allons identifier les différentes classes :

La société Primeurs TED possède une flotte d'une centaine de camions et se spécialise dans le transport de fruits et légumes. Un nouveau processus de gestion des commandes de livraison sera bientôt mis en place. À l'avenir, un client pourra passer une commande de deux façons : directement, via le site Internet transactionnel de Primeurs TED ou par téléphone. Une commande pourra comporter une ou plusieurs livraisons. Lorsqu'un client passera une commande il indiquera, pour chaque livraison, le lieu d'origine de la livraison, la destination de la marchandise à livrer ainsi que la date et le volume. Par exemple, un

maraîcher (agriculteur de légumes) contactera Primeurs TED en début d'été et « commandera » une livraison de laitues chaque mardi et chaque jeudi entre le 20 juin et le 30 septembre. Les livraisons du mardi sont destinées au Marché de l'Est et celles du jeudi au Marché du Centre. Chaque livraison comportera 300 cageots de laitues. Certaines commandes seront plus simples, une commande pouvant ne comporter qu'une livraison, par exemple, 10 palettes de sacs de pommes de terre et 25 palettes de sacs de carottes à livrer d'un point A à un point B, à une date donnée. Pour Primeurs TED, une commande est composée d'une ou plusieurs livraisons à effectuer pour un client. Une livraison a une destination, une date de livraison, un volume donné et certaines exigences de température. Bien que l'entreprise ne vende aucun produit, on doit garder la trace des produits transportés pour chaque livraison. Une livraison pourra comporter un ou plusieurs produits et un produit donné pourra faire partie de plusieurs livraisons (par exemple, le produit sac de pommes de terre de 10 kilos pourra faire partie de centaines de livraisons). Une livraison peut toujours être effectuée par un seul camion. Un répartiteur déterminera quel camion effectuera chaque livraison. Le résultat de cette répartition sera accessible par le contremaître des conducteurs de camion.

Lors de l'identification des classes, On s'interrogera au sujet **des entités** à savoir les personnes, les lieux, les événements et les choses, qui font partie du processus ou qui interagissent avec le processus et au sujet desquelles on doit saisir et/ou conserver des données.

Dans notre cas, nous avons les classes suivantes :

Client, Commande, Livraison, Produit, Camion, Répartiteur, Contremaître et Conducteur de camion.

Attribut et opération

Chaque classe possède des attributs qui lui sont propres et dont on souhaite garder la trace. Ce seront les données qui seront entreposées dans les bases de données du système d'information.

Une opération est une fonction applicable aux objets d'une classe. Une opération permet de décrire le comportement d'un objet.

Voici un exemple de la classe Etudiant avec ses attributs et ses opérations

Etudiant
N° étudiant Nom et prénom Date inscription Filière Groupe
sinscrire() Modifier_informations() Changer_filiere() Changer_groupe()

Cette forme de représentation de classe est la plus simple car elle ne fait apparaître les caractéristiques des attributs : Type de l'attribut (entier, réel, etc), sa visibilité (privé ou public), sa valeur initiale etc. et les caractéristiques des opérations : visibilité de l'opération (privé ou public) les paramètres de l'opération) etc. Cette forme de représentation de classe est utilisée lors de la modélisation.

Visibilité des attributs et opérations

La visibilité des attributs et des opérations exprime l'encapsulation ou non de ces derniers. UML définit trois niveaux de visibilité pour les attributs et les opérations. Chaque attribut ou opération d'une classe peut être de type **public**, **privé** ou **protégé**.

- 1) **Public** : l'attribut ou la méthode est accessible ou visible en dehors de la classe.
- 2) **Privé** : l'attribut ou la méthode est accessible ou visible qu'à l'intérieur de la classe.
- 3) **Protégé** : l'attribut ou la méthode est accessible ou visible à l'intérieur de la classe et dans toutes les sous classes de la classe.

Les symboles + (pour désigner public), # (pour désigner protégé) et - (pour désigner privé) sont indiqués devant chaque attribut ou opération pour signifier le type de visibilité autorisé pour les autres classes

L'exemple suivant montre que les attributs ont la visibilité privée (signe - moins) c'est-à-dire qu'ils sont encapsulés, alors que les méthodes ont la visibilité public (le signe +).

Etudiant

<ul style="list-style-type: none"> - N° étudiant - Nom et prénom - Date inscription - Filière - Groupe
<ul style="list-style-type: none"> + sinscrire() + Modifier_informations() + Changer_filiere() + Changer_groupe()

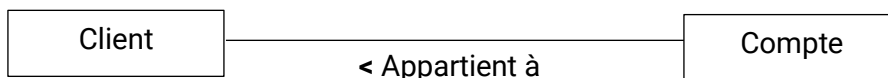
Les associations entre classes

L'un des principaux rôles du diagramme de classes est d'indiquer les associations entre les classes. Ces associations et leur multiplicité dépendent du contexte organisationnel, c'est-à-dire du processus et de l'environnement.

Après avoir trouvé les classes, il convient de les relier entre elles. Les relations entre classes expriment les liens sémantiques ou structurels.

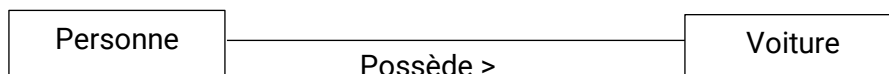
Les relations les plus utilisées sont **l'association, l'agrégation, la composition, la dépendance et l'héritage**.

Les associations sont représentées par une ligne. Le nom de l'association est écrit au-dessus ou au-dessous de la ligne. Cependant, quand l'association est évidente son nom peut être omis.



Le sens de l'association peut être précisé par les signes inférieur < ou supérieur > si cela est nécessaire. Dans l'exemple ci-dessus l'association Appartient à pointe vers Client pour traduire le fait que le compte appartient au client.

Dans l'exemple suivant, l'association Possède pointe vers Voiture pour traduire le fait qu'une personne possède une voiture.



Rôle de l'association

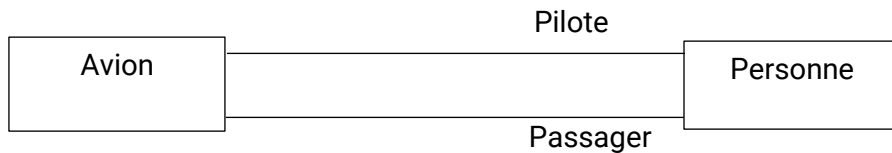
L'extrémité d'une association est appelée **rôle**. Chaque association binaire (entre deux classes) possède deux rôles. Le rôle décrit comment une classe voit une autre classe au travers d'une association. Le nom du rôle se distingue du nom d'une association par son placement de l'extrémité de l'association.

Ainsi, dans l'exemple suivant, le rôle du client est d'être détenteur du compte et celui du compte est d'être possédé par le client.

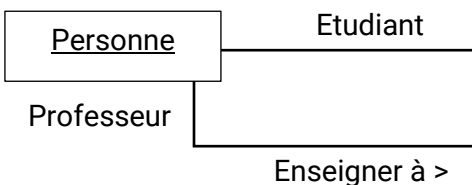


Lorsque deux classes sont reliées par une seule association, le nom des classes suffit souvent à caractériser le rôle. Le nommage des rôles prend tout son intérêt lorsque plusieurs associations relient deux classes.

Dans l'exemple suivant, les noms des rôles permettent de distinguer deux objets : pilote et passager appartenant à la classe Personne.



Aussi, dans l'exemple suivant, les noms des rôles permettent de distinguer deux objets : professeur et étudiant appartenant à la classe Personne et reliés par l'association *Enseigner à*.



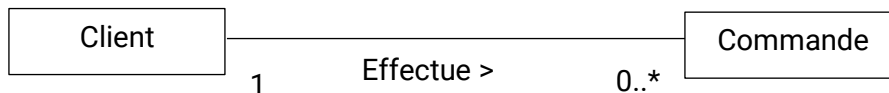
Multiplicité ou cardinalité des associations

La multiplicité ou la cardinalité d'une association est située à chaque extrémité de celle-ci. La multiplicité définit un domaine de valeurs pour préciser le nombre d'objets ou d'instances de la

classe apparaissant à cette extrémité pouvant s'associer à un seul et unique objet de la classe apparaissant à l'autre extrémité.

Dans l'exemple suivant nous avons :

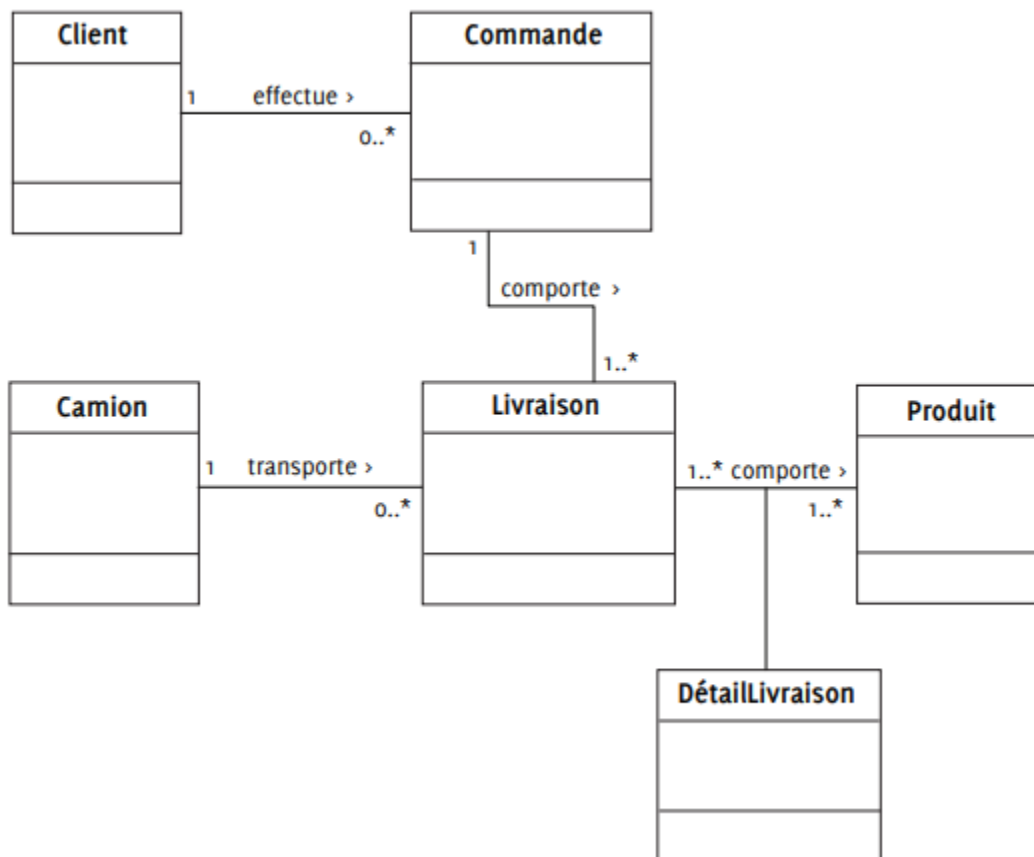
- Un Client effectue 0..* Commande. La multiplicité 0..* de l'association effectuée entre les classes Client et Commande signifie qu'un client peut n'effectuer aucune commande ou en effectuer plusieurs. Comment est-il possible qu'un client n'effectue aucune commande ? On peut imaginer une situation où des représentants de l'entreprise font du démarchage auprès de clients afin d'augmenter la part de marché. Un client peut être recruté – donc existe – avant même d'avoir effectué une commande. Par ailleurs, le même client pourra effectuer plusieurs commandes pendant toute la période où il fera affaire avec l'entreprise.
- Une Commande est effectuée par 1 Client. Le 1 situé près de Client sur le lien effectuée signifie qu'à une commande est associé un seul client.



Le tableau suivant résume les différentes multiplicités possibles

Multiplicité	Signification
1	Un et un seul
0..1	Zéro à un
*	De zéro à plusieurs
0..*	De zéro à plusieurs
1..*	D'un à plusieurs
M..N	De M à N

Dans l'exemple du processus de livraison de Primeurs TED, les classes et leurs associations sont représentées dans le diagramme suivant :



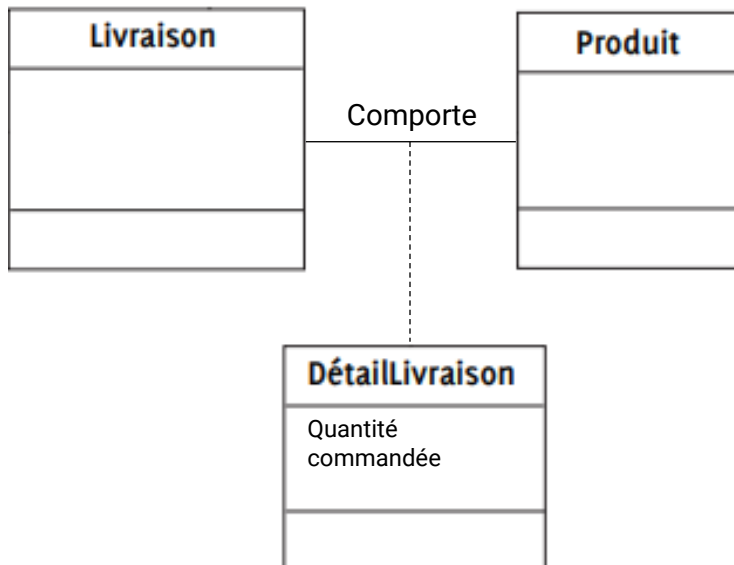
- Un Client effectue 0..* Commande. La multiplicité 0..* de l'association effectue entre les classes Client et Commande signifie qu'un client peut n'effectuer aucune commande ou en effectuer plusieurs.
- Une Commande est effectuée par 1 Client. Le 1 situé près de Client sur le lien effectue signifie qu'à une commande est associé un seul client.
- Une Commande comporte 1..* Livraison. La description du processus précisait qu'une commande pouvait comporter plusieurs livraisons. La mention 1..* placée près de Livraison sur le lien est précisée dans le texte. C'est le cas de la commande passée par le maraîcher qui comporte une livraison chaque mardi et jeudi entre le 20 juin et le 30 septembre.
- Une Livraison correspond à 1 Commande. Le chiffre 1 placé près de Commande sur le même lien signifie qu'une livraison ne correspond qu'à une commande.

- Un Camion transporte 0..* Livraison. Le même camion pourra être affecté à plusieurs livraisons.
- Une Livraison est transportée par 1 Camion. Puisque chez Primeurs TED une livraison est entièrement transportée par un camion, on a placé un 1 près de camion.
- Une livraison comporte 1..* Produit. Une livraison peut comporter un seul produit, par exemple que la laitue ou plusieurs produits sacs de pommes de terre et sacs de carottes.
- Un produit est comporté par 1..* Livraison. Un produit est comporté par une ou plusieurs livraisons. Un produit peut être livré qu'une seule fois et peut être livré plusieurs fois ainsi plusieurs livraisons.

Les classes-associations

Une association peut être représentée par une classe dans le cas où l'association comporte des attributs et/ou des méthodes. Une classe de ce type est appelée **classe-association** ou classe associative. Une classe-association peut participer à d'autres relations. Une association-classe utilise comme notation **une ligne pointillée** pour attacher une classe à une association.

Dans l'exemple suivant, l'association entre les classes **Livraison** et **Produit** est représentée par la classe **DétailLivraison** ayant l'attribut **Quantité commandée**.

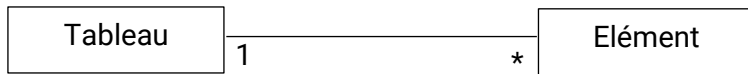


Qualification d'association ou association qualifiée

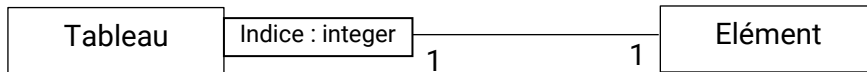
La qualification d'une association permet, dans certains cas, de transformer une multiplicité indéterminée ou infinie, d'une extrémité de l'association, en une multiplicité finie.

Dans l'exemple suivant nous modélisons un tableau qui permet de référencer un ensemble d'éléments sans et avec qualification :

Modélisation sans qualification



Modélisation avec qualification



Association de dimension supérieure à 2 ou association n-aire

Une association n-aire relie plus de 2 classes ou plus. La gestion de ce type d'association est très délicate, notamment quand on ajoute la multiplicité.

Les traits pleins qui viennent de toutes les classes associées convergent vers un losange central pouvant éventuellement accueillir une classe-association. La multiplicité apparaissant sur le lien de chaque classe s'applique sur une instance du losange.

L'exemple suivant une association

