

Objets, Classes, Association et Multiplicité

Sommaire

Qu'est-ce qu'un objet ?.....	2
Qu'est-ce qu'une classe ?	3
Associations entre classes	4
La multiplicité ou les cardinalités	6
Rôle de l'association	8

Dans ce cours, je vous présente les concepts objet, classe, association et multiplicité ainsi que les notations UML associées.

Qu'est-ce qu'un objet ?

Les concepts objet et classes sont utilisés dans la programmation orientée objet en abrégation POO. En programmation orientée objet, un objet est considéré comme une variable et une classe est considérée comme un type dans un langage de programmation classique (langage C par exemple).

En programmation classique, nous créons des variables et nous les gérons par des sous programmes. En programmation orientée objet, un objet comporte en même temps des variables et des fonctions qui manipulent ces variables. Ainsi, la classe est un modèle ou un Template qui va servir à créer un objet dans la RAM. Un objet est appelé aussi une instance.

Lorsque nous sommes amenés à créer une application, la première difficulté est d'identifier les classes. Alors la question qui se pose est comment identifier les classes ?

Les classes sont identifiées lors de la phase de la modélisation du système à étudier. Ce dernier est constitué d'objets. C'est-à-dire des choses qui existent qui font partie du système et ayant un rôle. Ces objets peuvent avoir une existence physique ou virtuelle.

Par exemple :

L'étudiant Ali ayant le numéro 125 est inscrit à la filière Master Droit international

Dans cette phrase, nous détectons deux objets :

- 1) Etudiant Ali numéro 125, un objet physique
- 2) Filière Droit international : un objet virtuel

Etudiant Ali
Numéro 125 Nom Ali

Filière Master Droit international
Nom filière : Droit international

Chaque objet est caractérisé par des propriétés et des attributs. Par exemple, les attributs d'un étudiant sont : Numéro étudiant et nom étudiant.

En comparaisant avec la méthode Merise, nous pouvons dire qu'un objet est équivalent à une occurrence.

Qu'est-ce qu'une classe ?

En Merise ou en base de données, nous disons une entité en UML, on dit une classe. Entité et Classe désignent la même chose.

Une classe est une description ou une abstraction d'un ensemble d'objets homogène.

Dans l'exemple suivant, les étudiant E1, E2 et E3 représentent trois objets. Ces trois objets partagent les mêmes attributs : Numéro étudiant et nom étudiant et nous pouvons les décrire par une entité qui s'appelle Etudiant. On parle de la classe Etudiant.

Etudiant E1	Etudiant E2	Etudiant E3
Numéro 122 Nom : E1	Numéro 123 Nom : E2	Numéro 123 Nom : E3

La classe Etudiant peut être représentée graphiquement par :

Etudiant
Numéro étudiant Nom étudiant

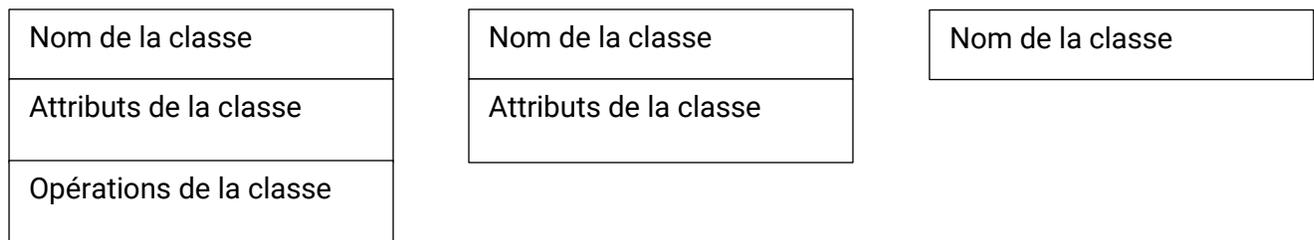
En plus des attributs, une classe comporte des opérations ou ce qu'on appelle aussi des méthodes.

Dans cet exemple, la classe Etudiant, en plus des attributs, comporte deux opérations : `getNom()` et `modifierInformation()`.

Etudiant
Numéro étudiant Nom étudiant
<code>getNom()</code> <code>ModifierInformation()</code>

En Merise, une entité comporte que des attributs ou des propriétés. En UML et en programmation orientée objet, la classe comporte des attributs et des opérations.

En modélisation, il y a trois façons de représenter graphiquement une classe :



Associations entre classes

Dans n'importe quel système, tout objet faisant partie du système est relié à d'autres objets. Il n'y a pas d'objets indépendants. Les objets dans un système se communiquent entre eux. Dans la modélisation, nous devons représenter cette association entre objets.

Les associations représentent les liens ou les relations entre deux ou plusieurs classes. Il existe plusieurs types d'association entre les classes. Voici les plus importantes :

- Association
- Association réflexive
- Agrégation
- Composition
- Héritage

- Classe d'association

Soit l'exemple suivant représentant une association entre les deux classes Etudiant et Filière.
L'association **Est inscrit** exprime qu'un étudiant est inscrit dans une filière.



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 **Est inscrit** à pointe vers Filière pour traduire le fait que l'étudiant est inscrit dans une filière.

[Comment identifier les associations ?](#)

Les associations sont souvent déduites des verbes du discours.

Par exemple dans la phrase suivante : Un client possède un compte bancaire

Ici nous avons deux classes : Client et Compte et elles sont liées par l'association Possède.



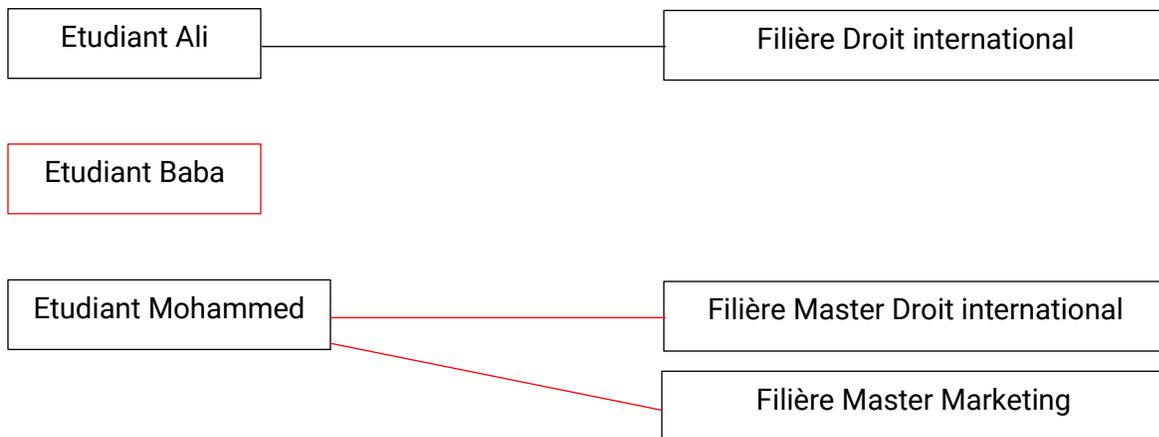
La multiplicité ou les cardinalités

En Merise, nous parlons de cardinalités et en UML, nous parlons de multiplicité.

Qu'est-ce que la multiplicité ?

Pour identifier la multiplicité, il faut poser la question pour un objet, combien de fois il participe à l'association.

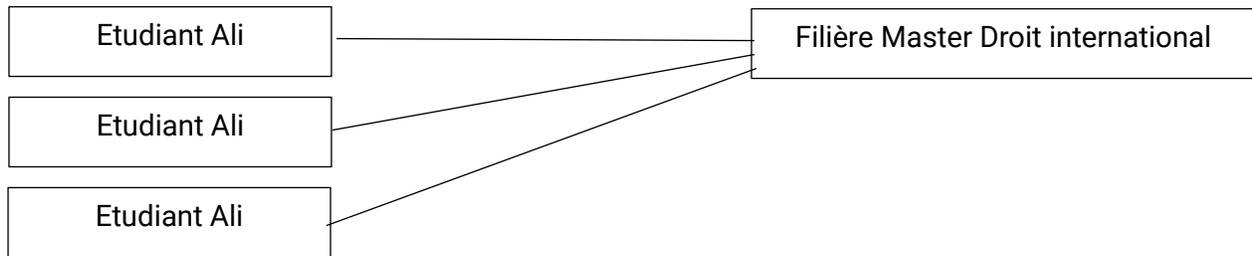
L'exemple suivant montre que l'objet Etudiant Ali participe à l'association **Est inscrit** une et une seule fois. Nous ne pouvons pas trouver un étudiant qui n'est inscrit à aucune filière. Aussi, nous ne trouvons pas un étudiant inscrit à plusieurs filières.



L'exemple suivant présente l'expression de la multiplicité pour la classe Etudiant. Vous constatez que le chiffre 1 est mentionné à côté de la classe Filière. Le chiffre 1 exprime qu'un étudiant est inscrit à une et une seule filière.



Passons maintenant à l'identification de la multiplicité de la classe Filière. Prenons un objet de la classe filière, par exemple, Master droit international.



L'objet Filière Master Droit International est lié à l'objet Etudiant Ali, à l'objet Etudiant Baba et lié à l'objet Etudiant Mohammed. Nous comprenons qu'un objet de la classe Filière peut être lié à plusieurs objets de la classe Etudiant. Nous exprimons cette multiplicité de la classe Filière par :



Vous constatez la présence de 1 ..* à côté de la classe Etudiant pour exprimer, qu'une filière a un ou plusieurs étudiant.

Le tableau suivant résumé les différentes multiplicités possibles

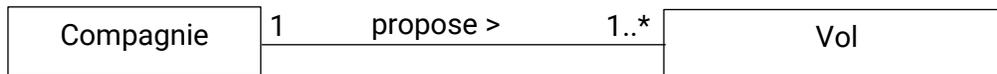
Multiplicité	Signification
1 ou 1..1	Un et un seul
0..1	Zéro à un
*	De zéro à plusieurs
0..*	De zéro à plusieurs
1..*	D'un à plusieurs
M..N	De M à N

Rôle de l'association

Soit la phrase suivante :

Des compagnies aériennes proposent différents vols

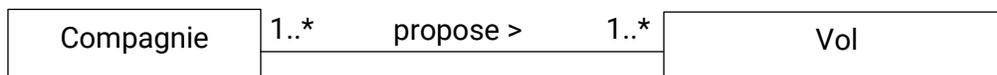
La modélisation en UML de cette phrase est la suivante :



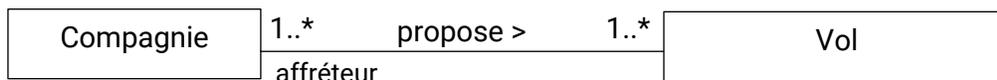
La multiplicité à côté de la classe VOL, 1..*, se traduit par Une compagnie propose un ou plusieurs vols.

La multiplicité à côté de la classe Compagnie, 1, se traduit par un vol est proposé par une et une seule compagnie.

Il y a des cas où un vol peut être partagé entre plusieurs affréteurs (contrat de location avec d'autres compagnies), dans ce cas, la multiplicité à côté de la classe Compagnie change et elle devient 1..*. Voilà le diagramme de classe modifié :



Alors pour préciser sur notre diagramme de classes, le pourquoi de la multiplicité 1..* à côté de la classe Compagnie, nous pouvons ajouter l'information affréteur comme ci-dessous :



L'information *affréteur* ajoutée au diagramme de classe s'appelle **Rôle**.

Qu'est-ce qu'un rôle ?

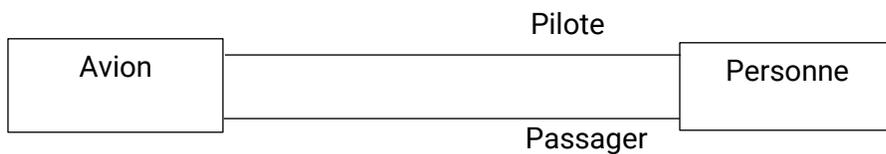
L'extrémité d'une association est appelée **rôle**. Chaque association binaire (entre deux classes) possède deux rôles. Le rôle décrit comment une classe voit une autre classe au travers d'une association. Le nom du rôle se distingue du nom d'une association par son placement de l'extrémité de l'association.

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*.

