

L'HERITAGE

Table des matières

L'HERITAGE	1
Introduction	2
Qu'est-ce que l'héritage ?.....	3
Comment définir une relation d'héritage ?	5

Introduction

Comme vous le savez, les langages de programmation orientés objets sont apparus avant le langage UML. Les langages de programmation orientés objets utilisent des concepts de base comme : Encapsulation, agrégation/composition, héritage, polymorphisme etc. Ces concepts sont importants et primordiales dans la programmation orientée objet. Ces concepts fondamentaux en programmation orientée objet existaient avant l'avènement du UML. Lors de l'élaboration d'un digramme de classes, le concepteur doit définir ces concepts (Héritage, agrégation/composition) et il doit faire l'effort de les identifier. C'est pour cela, que nous trouvons les notions d'agrégation/composition, héritage comme type d'association dans le diagramme de classes.

Qu'est-ce qu'une association ?

Dans un diagramme de classes, une relation, ou une association, est une connexion entre deux classes. Le rôle de cette association est d'ajouter une sémantique au diagramme de classes.

Voici les relations que nous trouvons dans un diagramme de classe :

- ✓ Relation d'association.
- ✓ Relation d'agrégation/Composition.
- ✓ Relation d'héritage.
- ✓ Relation de dépendance.
- ✓ Relation orientée ou relation navigable.
- ✓ Classe association
- ✓ Association qualifiée
- ✓ Interface.

Est-ce que nous pouvons nous contenter d'utiliser que des relations de type association, c'est-à-dire celles qui sont modélisées en UML par un simple trait ?

D'une part et au niveau de la sémantique, si nous utilisons que des relations d'association, notre diagramme de classes reste toujours incomplet. Ce type d'association, relation d'association, ne nous fournis pas toutes les informations et tout le détail sur le type de relation entre les classes. Lors de l'élaboration du diagramme de classe, l'identification et la définition des associations particulières comme agrégation/composition, héritage, relation orientée etc. est obligatoire.

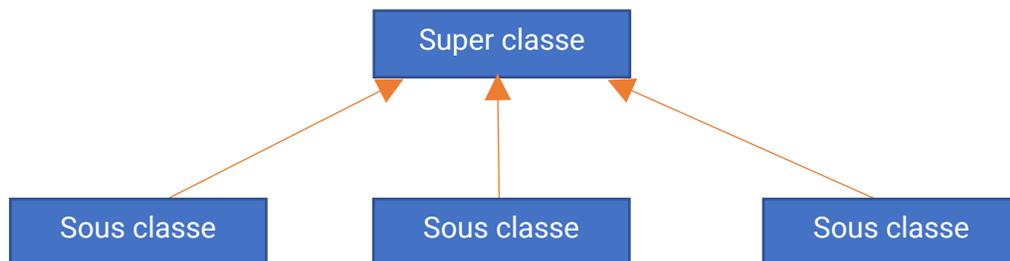
D'autre part et au niveau de l'implémentation, c'est-à-dire lors de la programmation, la définition de ces types d'association est importante, nous allons bénéficier de la puissance de la programmation orientée objet : Héritage, agrégation, interface etc.

L'objectif de ce cours est d'identifier et de comprendre ce que c'est l'héritage.

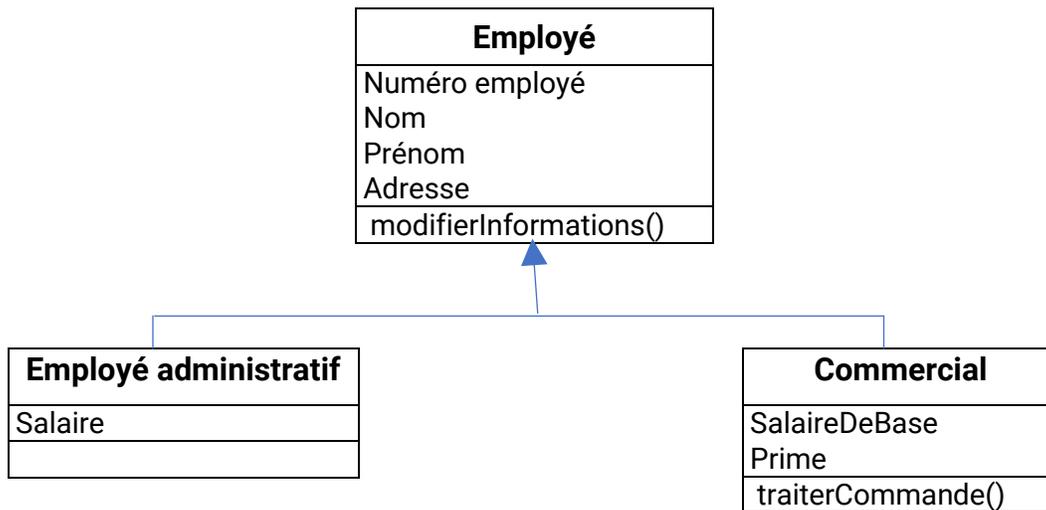
Qu'est-ce que l'héritage ?

L'héritage est un mécanisme qui provient de la programmation objet. L'héritage permet de regrouper des classes ayant des caractéristiques communes, attributs et opérations, en une classe générique appelée super classe. Cette super classe a comme attributs et opérations ces caractéristiques communes, et les autres classes seront appelées classes filles et chacune possède ses propres attributs et opérations. Nous obtiendrons ainsi, une hiérarchie entre les classes.

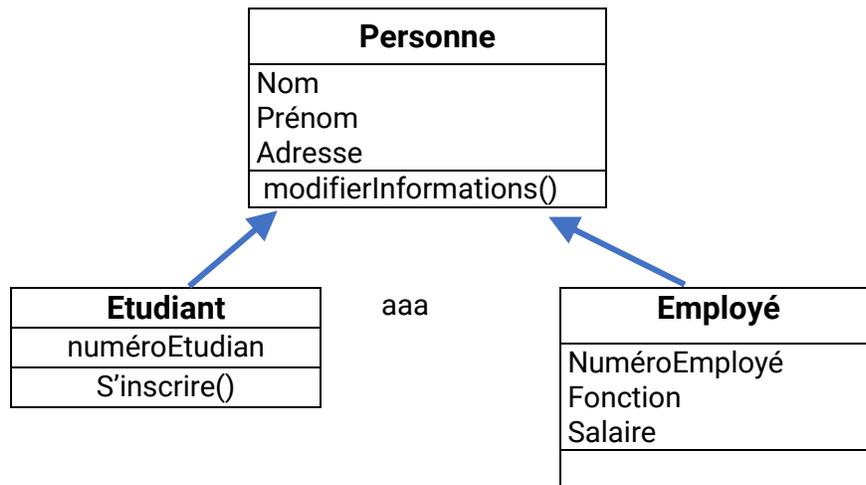
Cette hiérarchisation des classes entre une super classe et des sous classes est exprimée par la relation « est un » et est traduite graphiquement par une flèche qui pointe vers la classe la plus générale.



Dans l'exemple suivant, les classes Commercial et Employé administratif partagent les mêmes attributs : numéro employé, le nom, le prénom et l'adresse et la même opération : `modifierInformations()`. Au lieu de les répéter pour chacune, les attributs et l'opération communs sont regroupés dans une superclasse `Employé`. Dans les classes `Employé administratif` et `Commercial`, nous plaçons dans chaque classe, les attributs et les opérations spécifiques à chacune des deux classes. Dans la classe `Employé administratif`, on trouve l'attribut `Salaire`, alors que dans la classe `Commercial`, il y a les attributs `SalaireDeBase` et `prime`



D'une autre façon, l'héritage est la propriété qui permet à une sous-classe d'hériter la structure, c'est-à-dire les attributs, et les opérations de sa superclasse, auxquelles viendront s'ajouter ses propres attributs et opérations. Un objet A créé d'une sous-classe, peut hériter les attributs et les opérations de sa superclasse auquel s'ajoutent ses propres attributs et ses propres méthodes. Ainsi, en plus des attributs et des méthodes de l'objet A, ce dernier dispose des attributs des opérations de superclasse. Ces derniers sont considérés comme les siens.



Dans cet exemple, les classes Etudiant et Employé sont des sous-classes de la classe Personne, la flèche exprime la hiérarchie entre la sous-classe et la superclasse. Chacune des sous-classes, hérite des attributs de la superclasse Personne : nom, prénom adresse. En outre, les classes Etudiant et Employé ont des attributs spécifiques : numéroEtudian, numéroEmployé, fonction et salaire.

Comment définir une relation d'héritage ?

Vous devez définir un héritage entre les classes C1 et C2, si les objets de C1 constituent un ensemble et les objets de C2 constituent un sous ensemble de l'ensemble des objets de C1.

Dans l'exemple précédent, les objets de la classe Personne constitue un ensemble et les objets de la classe Etudiant constituent un sous ensemble de l'ensemble Personne. La même chose pour la classe Employé, les objets de cette classe constituent un sous ensemble de l'ensemble Personne.

