CHAPITRE 1 :

CONCEPTS DE BASE DE LA POO

Objectifs spécifiques

- Introduire les facteurs de naissance de la POO
- Introduire la définition de la POO
- Introduction au concept de l'approche OO

Eléments de contenu

I. De la programmation classique vers la POO

II. Définition

III. Concepts de base de la POO

Volume Horaire :

Cours : 1 heure 30

TD : 0 heure

1.1 De la programmation classique vers la programmation orientée objet

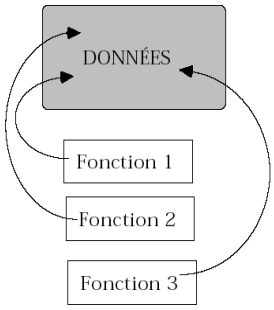
La programmation classique telle que étudiée au travers des langages C, Pascal… définie un programme comme étant un ensemble de données sur lesquelles agissent des procédures et des fonctions.

Les données constituent la partie passive du programme. Les procédures et les fonctions constituent la partie active.

Programmer dans ce cas revenait à :

- définir un certain nombre de variables (structures, tableaux…)
- écrire des procédures pour les manipuler sans associer explicitement les unes aux autres.

Exécuter un programme se réduit alors à appeler ces procédures dans un ordre décrit par le séquençage des instructions et en leur fournissant les données nécessaires à l'accomplissement de leurs tâches.



Dans cette approche données et procédure sont traitées indépendamment les unes des autres sans tenir compte des relations étroites qui les unissent.

Les questions qu'on peut poser dans ce cas :

1. Cette séparation (données, procédures) est elle utile ?
2. Pourquoi privilégier les procédures sur les données (Que veut-on faire ?) ?
3. Pourquoi ne pas considérer que les programmes sont avant tout des ensembles objets informatiques caractérisé par les opérations qu'ils connaissent ?

Les langages objets sont nés pour répondre à ces questions. Ils sont fondés sur la connaissance d'une seule catégorie d'entités informatiques : les objets.

Un objet incorpore des aspects statiques et dynamiques au sein d'une même notion.

Avec les objets ce sont les données qui deviennent prépondérantes. On répond tout d'abord à la question « De quoi parle-t-on ? »

Un programme est constitué d'un ensemble d'objets chacun disposant d'une partie procédures et d'une partie données. Les objets interagissent par envoie de messages.

1.2 Définitions

1.2.1 POO

La POO est une méthode d'implémentation dans laquelle les programmes sont organisés sous formes de collections coopératives d'objets, dont chacun représente une instance d'une classe quelconque et dont toutes les classes sont membres d'une hiérarchie de classes unis à travers des relations d'héritage.

1.2.2 Algorithmique Objet

Dans l'approche orienté objet un algorithme sera essentiellement vu comme un ensemble d'objets auxquels l'utilisateur envoie des messages et qui s'en envoient pendent le fonctionnement.

Ces objets seront toujours pour l'utilisateur des boites noires et qui contiendront des variables locales, inconnues de l'environnement, et qui ne s'y intéressera d'ailleurs pas. Le seul moyen d'accéder à ces objets sera l'envoie des messages qu'ils sont capables de comprendre.

Rq : La spécification d'un système dans l'approche OO va s'axer principalement sur la détermination des objets à manipuler. Une fois cette étape réalisé le concepteur n'aura plus qu'à réaliser les fonctions de haut niveau qui s'appuient sur les objets et les familles d'objets définis.

1.2.3 Objet et classe

➔ Un objet est une entité logicielle :

- Ayant une identité
- Capable de sauvegarder un état c'est-à-dire un ensemble d'information dans des variables internes.
- Répondant à des messages précis en déclenchant des activations internes appropriés qui changent l'état de l'objet. Ces opération sont appelées méthodes. Ce sont des fonctions liées à des objets et qui précisent le comportement de ces objets.

1.2.4 Attributs

Les attributs d'un objet sont l'ensemble des informations se présentant sous forme de variable et permettant de représenter l'état de l'objet.

1.2.5 Message

Un message est une demande d'activation d'une méthode envoyé à un objet.

1.2.6 Méthodes

Une méthode est une fonction ou procédure liée à un objet qui est déclenchée à la réception d'un message particulier : la méthode déclenchée correspond strictement au message reçu. La liste des méthodes définies au sein d'un objet constitue l'interface de l'objet pour l'utilisateur : ce sont les messages que l'objet peut comprendre si on les lui envoie et dont la réception déclenche les méthodes correspondantes.

1.2.7 Signature

La signature d'une méthode représente la précision de son nom, du type de ses arguments et du type de donnée retournée.

1.3 Concept de base de la POO

La POO se base sur les notions clés suivantes :

- Encapsulation
- Abstraction
- Classe et objets
- Héritage
- Polymorphisme

1.3.1 Encapsulation

L'encapsulation est le faite qu'un objet renferme ses propres attributs et ses méthodes. Les détails de l'implémentation d'un objet sont masqués aux autres objets du système à objets. On dit qu'il ya encapsulation de données et du comportement des objets.

On précise trois modes d'accès aux attributs d'un objet.

- Le mode ***public*** avec lequel les attributs seront accessibles directement par l'objet lui même ou par d'autres objets. Il s'agit du niveau le plus bas de protection.
- Le mode ***private*** avec lequel les attributs de l'objet seront inaccessibles à partir d'autres objets : seules les méthodes de l'objet pourront y accéder. Il s'agit du niveau le plus fort de protection.
- Le mode ***protected :*** cette technique de protection est étroitement associée à la notion d'héritage (suite du cours).

1.3.2 Abstraction

C'est le faite de se concentrer sur les caractéristiques importantes d'un objet selon le point de vue de l'observateur.

Exemple : Voiture

L'abstraction est un principe qui consiste à ignorer certains aspects d'un sujets qui ne sont pas importants pour le problème dans le but de se concentrer sur ceux qui le sont.

1.3.3 Classes, objets, instances

→ Une classe est un ensemble d'objets qui ont en commun :

- les mêmes méthodes
- les mêmes types d'attributs

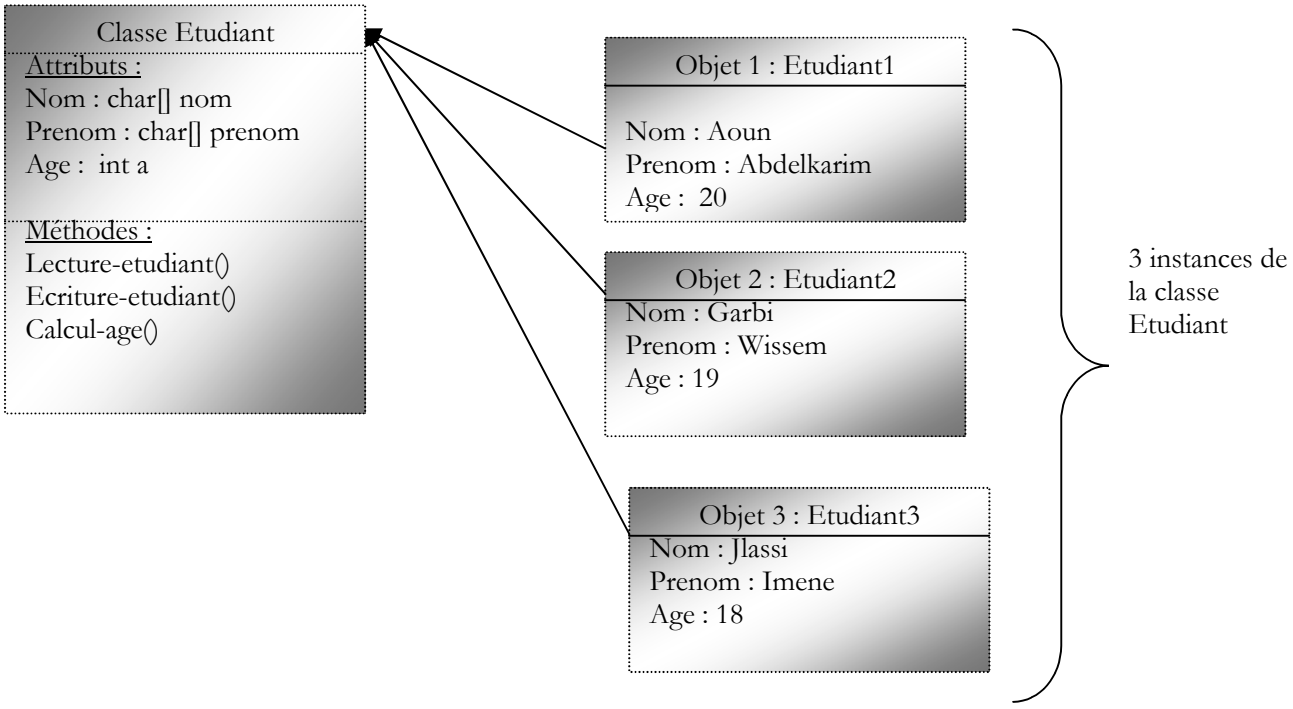
Classe = attributs + méthodes + instanciations

→ Une instance d'une classe est un objet particulier d'une classe qui peut activer les méthodes de la classe et qui a des valeurs particulières de ses attributs.

→ On définit l'objet comme l'instance d'une classe. La classe représente pour l'objet ce que représente le type pour la variable. Un objet est caractérisé par :

- Son identité (OID) : valeur unique et invariante qui caractérise l'objet
- Son comportement : qui est défini à travers les méthodes applicables à cet objet et qui caractérisent sa façon d'agir et de réagir dans son environnement.
- Son état : qui constitue l'ensemble des valeurs des attributs de cet objet.

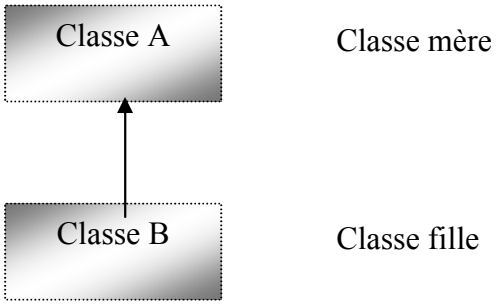
→ Une classe est un type abstrait qui encapsulent données et traitement. C'est une sorte de moule qui permet ensuite de créer autant d'instances qu'on veut. Ces instances seront des objets de la classe auxquelles on pourra effectivement envoyer des messages qui activeront les méthodes correspondantes.



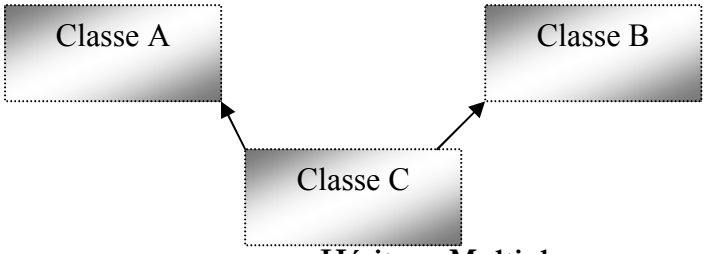
1.3.4 Héritage

La notion d'héritage est une relation entre différentes classes permettant de définir une nouvelle classe en se basant sur les classes existantes. On parle d'héritage simple lorsqu'uen classe fille ne possède qu'une classe mère.

On parle d'héritage multiple lorsqu'une classe fille possède plusieurs classes filles.



Héritage simple



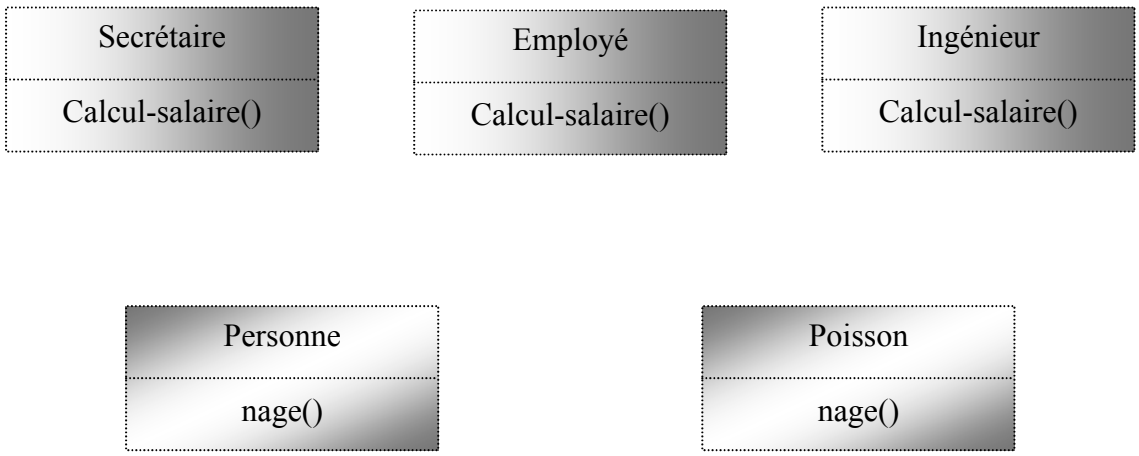
Héritage Multiple

1.3.5 Polymorphisme

Le terme polymorphisme issu du grec signifie la faculté de prendre plusieurs formes.

Une entité est polymorphe si à l'exécution elle peut se référer à des instances de classe différentes.

Exemples :



Le polymorphisme est un mécanisme qui permet à une sous classe de redéfinir une méthode dont elle a hérité tout en gardant la même signature de la méthode.