Introduction à la Programmation Orientée Objet

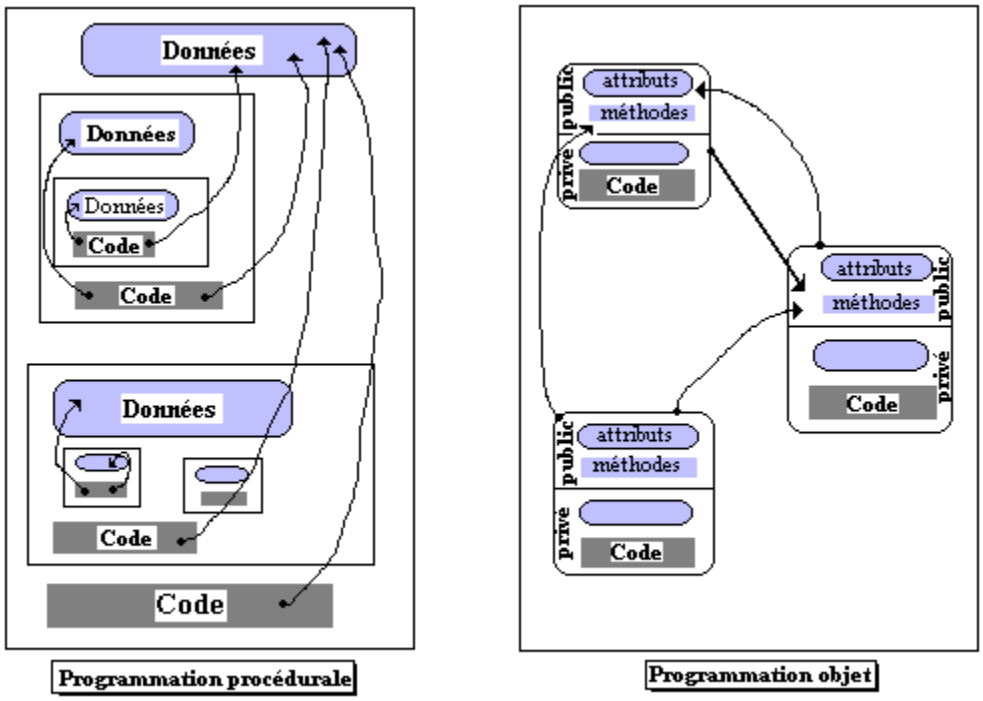
La programmation classique ou procédurale telle que le débutant peut la connaître à travers des langages de programmation comme Pascal, C etc... traite les programmes comme un ensemble de données sur lesquelles agissent des procédures. Les procédures sont les éléments actifs et importants, les données devenant des éléments passifs qui traversent l'arborescence de programmation procédurale en tant que flot d'information.

Cette manière de concevoir les programmes reste proche des machines de Von Neuman et consiste en dernier ressort à traiter indépendamment les données et les algorithmes (traduits par des procédures) sans tenir compte des relations qui les lient.

En introduisant la notion de modularité dans la programmation structurée descendante, l'approche diffère légèrement de l'approche habituelle de la programmation algorithmique classique. Nous avons défini des machines abstraites qui ont une autonomie relative et qui possèdent leurs propres structures de données ; la conception d'un programme relevait dès lors essentiellement de la description des interactions que ces machines ont entre elles.

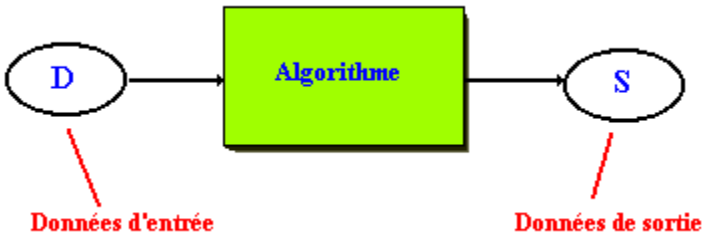
La programmation orientée objet relève d'une conception ascendante définie comme des "messages" échangés par des entité de base appelées objets.

Comparaison des deux topologies de programmation



Les langages objets sont fondés sur la connaissance d'une seule catégorie d'entité informatique : l'objet. Dans un objet, traditionnellement ce sont les données qui deviennent prépondérantes. On se pose d'abord la question : "de quoi parle-t-on ?" et non pas la question "que veut-on faire ?", comme en programmation algorithmique. C'est en ce sens que les machines abstraites de la programmation structurée modulaire peuvent être considérées comme des pré-objets. En fait la notion de TAD est utilisée dans cet ouvrage comme spécification d'un objet, en ce sens nous nous préoccupons essentiellement des services offerts par un objet indépendamment de sa structure interne.

Programmation structurée :



1. Concepts fondamentaux de La P.O.O

Nous écrirons P.O.O pour : programmation orientée objet.

Voici trois concepts qui donnent toute sa puissance à la P.O.O.

- Concept de modélisation à travers la notion de classe et d'instanciation de ces classes.
- Concept d'action à travers la notion d'envoi de messages et de méthodes à l'intérieur des objets.
- Concept de construction par réutilisation et amélioration par l'utilisation de la notion d'héritage.

1.1 les classes

Postulons une analogie entre les objets matériels de la vie courante et les objets informatiques. Un objet de tous les jours est souvent obtenu à partir d'un moule industriel servant de modèle pour en fabriquer des milliers. Il en est de même pour les objets informatiques.

Définition

Une classe est une sorte de moule ou de matrice à partir duquel sont engendrés les objets réels qui s'appellent des instances de la classe considérée.

Remarque

En POO, programmer revient donc à décrire des classes d'objets, à caractériser leur structure et leur comportement, puis à instancier ces classes pour créer des objets réels. Un objet réel est matérialisé dans l'ordinateur par une zone de mémoire que les données et son code occupent.

Une classe est composée :

- **D'attributs (ou champs, ou variables d'instances).**

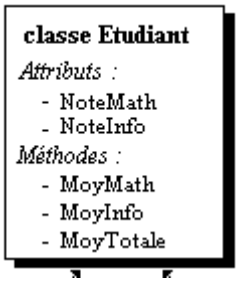
  Les attributs de la classe décrivent la structure de ses instances (les objets).

- **De méthodes (ou opérations de la classe).**

Les méthodes décrivent les opérations qui sont applicables aux instances de la classe.

Un exemple : les étudiants

Supposons que chaque étudiant soit caractérisé par sa note en mathématiques (NoteMath) et sa note en informatique (NoteInfo). Un étudiant doit pouvoir effectuer éventuellement des opérations de calcul de ses moyennes dans ces deux matières (MoyMath, MoyInfo)et connaître sa moyenne générale calculée à partir de ces deux notes (MoyTotale).



La classe Etudiant a été créée. Elle ne possède que les attributs NoteMath et NoteInfo. Les méthodes de cette classe sont par exemple MoyMath, MoyInfo, MoyTotale.

1.2 les objets

Définition

Un module représente un objet ou une classe d'objet de l'espace du problème et non une étape principale du processus total, comme en programmation descendante.

Recenser les objets du monde réel

Lors de l'analyse du problème, faire l'état de l'existant en recensant les objets du monde réel. On établit des classes d'objets et pour chaque objet on inventorie les connaissances que l'on a sur lui :

- Les connaissances déclaratives,
- Les connaissances fonctionnelles,
- L'objet réel et les connaissances que l'on a sur lui sont regroupés dans une même entité.

On décrit les systèmes en classes d'objets plutôt qu'en terme de fonction.

Exemple :

Une application de gestion bancaire est organisée sur les objets comptes, écritures, états.

Les objets rassemblent une partie de la connaissance totale portant sur le problème. Cette connaissance est répartie sur tous les objets sous forme déclarative ou procédurale.

Les objets sont décrits selon le modèle des structures abstraites de données (TAD) : ils constituent des boîtes noires dissimulant leur implantation avec une interface publique pour les autres objets. Les interactions s'établissant à travers cette interface.

Un objet :