

# Introduction

## 1 De la programmation classique vers la programmation orientée objet

La programmation classique en utilisant des langages tel que C, Pascal... définit un programme comme étant un ensemble de données sur lesquelles agissent des procédures et des fonctions.

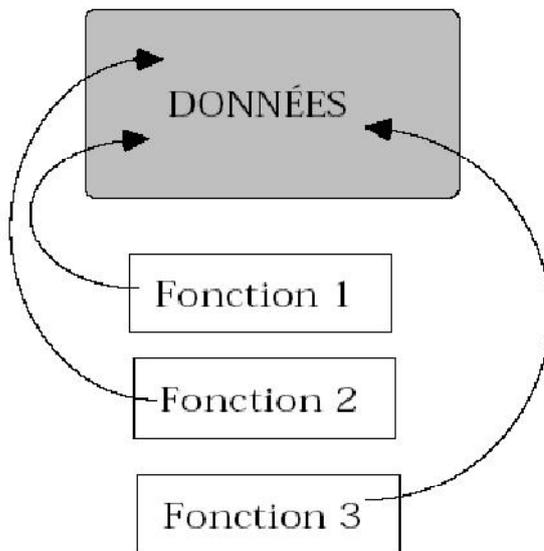
Les données constituent la partie passive du programme. Les procédures et les fonctions constituent la partie active.

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.

Cette forme de programmation, qui sépare les données des traitements, rend difficile la mise en place de contrôles, destinés à garantir la cohérence et l'intégrité des données. L'utilisateur peut en effet accéder directement aux données, sans utiliser les fonctions mises à sa disposition.

De plus, toute modification de l'implémentation des données a des conséquences sur l'ensemble des fonctions travaillant sur ces données, et donc sur l'ensemble des traitements utilisant ces fonctions.

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 et afin de remédier à ces inconvénients. Ils sont fondés sur la connaissance d'une seule catégorie d'entités informatiques qui est l'objet.

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 en interaction chacun disposant d'une partie procédures et d'une partie données. Les objets interagissent par envoi de messages.

Un objet représente un concept, une idée ou toute entité du monde physique, comme une voiture, une personne, un compte... L'OO est facilement concevable car il décrit des entités comme il en existe dans le monde réel.

L'OO supplante peu à peu le procédural dans les grands programmes car il présente d'énormes avantages : facilité d'organisation, réutilisation, méthode plus intuitive, possibilité d'héritage, facilité de correction, projets plus faciles à gérer.

## 2- Historique

Contrairement à ce que l'on pourrait croire, le concept de la programmation orientée objet n'est pas récent.

C'est dans les années 60 que deux brillants chercheurs norvégiens, Kristen Nygaard et Ole-Johan Dahl, ont développé les bases de la programmation objet en créant le langage Simula.

Smalltalk, développé à partir de 1971 par Alan Kay et publié en 1980, (au centre de recherche informatique californien de XEROX) est le premier vrai langage objet avec environnement de développement graphique intégré.

Toujours en 1980, le danois Bjarne Stroustrup développe le C++ (initialement appelé C with classes).

ADA95 pour le ADA, Object Pascal pour le Pascal, PHP5 pour le PHP, ...

Dans les années 1990, James Gosling développe le langage Oak pour Sun Microsystems. Oak sera rebaptisé Java en 1995. Ce langage a été pensé pour être indépendant du matériel qui exécute ses programmes « précompilés », pour être robuste et sûr, et surtout plus facile à programmer que le C++. Aujourd'hui, le langage Java est très largement utilisé à la fois sur le Web, les stations de travail, les téléphones et autres tablettes. Il appartient à Oracle.

1- En 2001, Microsoft présente C# (prononcé C Sharp - sharp) ; langage développé par le danois Anders Hejlsberg qui n'est autre que le créateur du Turbo Pascal et l'architecte de Delphi. C# est un langage objet proche de Java permettant de programmer des applications dans les environnements Microsoft tels que les stations de travail, les applications web, les smartphones ou les tablettes.