

Chapitre 1

Paradigmes de programmation

1.1 Langages de programmation

On a trois catégories de langages suivant le niveau d'abstraction :

- le langage machine, lequel est le seul compréhensible par la machine, il est constitué de données et d'instructions sous forme binaire ;
- l'assembleur, lequel représente le langage machine sous forme compréhensible par un humain ;
- les langages de haut niveau, offrant un haut niveau d'abstraction et plus facilement utilisable par un humain ; ils doivent être compilés en langage machine pour que celle-ci les traite.

Les différents langages de haut niveau peuvent s'organiser des types de logiques que l'on nomme **paradigmes**.

1.2 Paradigmes de programmation

Un paradigme est un modèle de pensée, de conceptualisation. Un paradigme de programmation traduira donc la façon dont un programme a été conçu. Les différents langages de programmation sont le reflet de la diversité des paradigmes de programmation ; toutefois, un paradigme n'est pas égal à un langage, certains langages peuvent convenir à plusieurs paradigmes (comme Python). Les paradigmes sont importants donnent les règles et structures de programmation. Voici les deux principaux et quelques-uns de leurs sous-paradigmes (à noter qu'il en existe bien d'autres) :

Programmation impérative : les opérations sont décrites en séquences exécutées les unes après les autres et ont pour but de modifier l'état de la machine i.e. de la mémoire. Il s'agit du paradigme le plus répandu et cela correspond au fonctionnement de la quasi totalité des ordinateurs : leurs processeurs sont programmés pour exécuter des suites d'instructions élémentaires. On pourrait résumer ce paradigme par le fait qu'il répond à la question « **comment ?** » du problème que l'on aborde. On retrouve les bases de la programmation vue en première avec :

- la séquence d'instructions ;
- l'affectation ;

- l'instruction conditionnelle ;
- la boucle ;
- les branchements – possibilité d'être transféré à un autre endroit du programme (par exemple `try except` en Python).

Il a pour principaux sous-paradigmes la programmation procédurale et la programmation structurée avec notamment la programmation orientée objet.

Programmation déclarative : cela consiste à programmer à partir d'autres éléments logiciels indépendants du contexte et ne comportant aucun état interne. Elle produira donc le même résultat quelque soit le moment et le contexte de l'appel du programme. On pourrait résumer ce paradigme par le fait qu'il répond à la question « **quoi ?** » du problème que l'on aborde. Il a pour principaux sous-paradigmes ;

- la programmation descriptive qui permet de décrire des structures de données avec des langages comme HTML et LaTeX ;
- la programmation fonctionnelle qui considère les applications comme un ensemble fonctions mathématiques ;
- la programmation logique qui considère les applications comme relations logiques.

