

1 – Les principaux paradigmes de programmation

La connaissance de différents paradigmes de programmation est importante pour approfondir la compréhension de l'informatique moderne et disposer d'un éventail plus riche de stratégies de résolution de problèmes.

1 – Les paradigmes de programmation

1- One to rule them all

Le mot paradigme vient du grec et signifie "modèle". En informatique, ce mot a été adopté pour distinguer les différentes façons d'envisager l'écriture d'un programme.

Cependant, quel que soit le programme et son mode d'écriture, il est traduit sous la forme de code machine impératif : donner à la machine des instructions élémentaires pour lui dire comment effectuer ses allers-retours entre le processeur et la mémoire.

Ce style (paradigme) de programmation impératif étant au coeur de la machine, la majorité des langages l'a adopté.

2 – Pourquoi d'autres paradigme ?

En 1954, John von Neumann, lorsqu'on lui a présenté Fortran, le premier véritable langage de programmation, aurait déclaré : "Pourquoi voudrait-on autre chose que le langage machine ?".

A cette question, on peut répondre qu'il est préférable de rapprocher le langage de la pensée du programme plutôt que de lui demander de penser en fonction de la machine. Or l'être humain a différentes façons de penser : différents types de langage ont donc été proposés.

La tendance actuelle est de ne pas attacher un langage à un paradigme mais de permettre l'adoption de plusieurs paradigmes au sein d'un même langage afin de pouvoir s'adapter à différentes situations.

2 – Exemples de paradigmes de programmation

1 – La variété des paradigmes

Des dizaines de paradigmes de programmation ont été décrits, outre le paradigme impératif. Son direct "opposé" est le paradigme déclaratif où l'on décrit ce que l'on veut faire sans demander au programmeur de donner les instructions précises à la machine. On trouve notamment dans cette catégorie la programmation logique, fonctionnelle, le SQL.

A la fin des années 1950, les concepts de la programmation orientée objet apparaissent. Ils auront un impact très important sur l'informatique d'aujourd'hui et sont présentés plus en détail dans la suite.

La programmation orientée acteurs et plus généralement la programmation permettant de gérer la concurrence a pris énormément d'importance depuis une dizaine d'années, pour tenir compte du changement drastique dans l'architecture des machines qui disposent de multiples

coeur ou processeurs. Des langages comme Erlang, Elixir, Scala, Go en sont l'illustration.

La liste ne saurait être exhaustive mais il faut évoquer la programmation qui révolutionne le monde depuis 2014 : l'apprentissage automatique. On programme des machines pour qu'elles pensent et programment par elles-mêmes...

2 – La programmation modulaire

Dans le cadre de la programmation classique de type impératif, en langage Python, C, Fortran, une notion fondamentale s'est dégagée : celle de modularité à travers l'usage de fonctions. Un programme est alors constitué de 2 types d'entités distinctes : les données et les fonctions.

Les fonctions peuvent être utilisées hors de leur contexte initial mais il faut alors redéfinir les données à manipuler (à moins d'utiliser des variables global, ce qui est connu comme étant une source de beaucoup de problèmes).

Cette conception donne la primauté aux traitements. Les briques logicielles sont essentiellement des morceaux de code réutilisable n'intégrant pas de données. Cette dissociation des données et des traitements restreint l'autonomie de ces briques logicielles et par la même occasion, leur réutilisabilité, ce à quoi la notion d'objet va essayer de remédier.

3 – Le paradigme objet

Le paradigme objet a été élaboré par les Norvégiens Ole-Johan Dahl et Kristen Nygaard au début des années 60 à travers le langage Simula, et repris et approfondi par l'Américain Alan Kay dans les années 70 dans le développement du langage SmallTalk qui est devenu une référence dans le domaine. La programmation orientée objet se retrouve dans nombre de langages aujourd'hui, comme le C++, C#, Java, Eiffel, Ruby ou Ocaml qui sont des langages à classes.

Dans la programmation orientée objet, pour remédier aux difficultés posées par la programmation modulaire, on considère un nouveau type d'entité : l'objet, qui intègre les données et les traitements. Les données trouvent ainsi leur place au sein de ces briques logicielles. L'objet prend une certaine "autonomie" et est en mesure de dialoguer avec les autres objets de son univers.