

2. L'algorithmique : activité au coeur du métier de programmeur

L'algorithmique est au cœur du métier de programmeur. Un **algorithme** est une description complète et non ambiguë d'un procédé utilisé pour résoudre un type de problème. L'exécution d'un algorithme résulte en un état particulier des données. Un algorithme comprend des instructions de base ainsi que des instructions de contrôle. Nous en reparlerons plus tard 😊

Sachez qu'un algorithme est indépendant de tout langage de programmation tel que Java. Un **langage de programmation** est un langage formel constitué d'un vocabulaire et de règles. Il sert à exprimer les algorithmes développés par le programmeur dans une forme exploitable et interprétable par un ordinateur. Nous étudierons ce concept au sein de la section suivante. On se focalise ici sur l'algorithmique.

Quelle différence entre un algorithme et un programme informatique ?

Le programme informatique est en fait la traduction d'un algorithme dans un langage de programmation donné. Les algorithmes peuvent être traduits dans plusieurs langages. Le programme informatique peut être interprété par l'ordinateur alors que l'algorithme pas. Il est une description abstraite de la résolution d'un type de problème alors que le programme informatique est concret.

Les qualités d'un "bon" algorithme :

1. Toutes les actions à réaliser sont définies.
2. Les actions définies ne sont pas ambiguës.
3. L'ensemble des actions permettent la prise en considération de l'ensemble des cas du problème traité.
4. L'algorithme doit être correct, c'est-à-dire que l'algorithme résout le problème posé (pas un autre problème, pas un problème (légèrement) différent, pas une partie seulement du même problème).
5. L'algorithme doit être performant.

Remarque : un programme informatique tout comme un algorithme ne peut comprendre les intentions du développeur... Il est donc important que ce développeur définisse clairement et exhaustivement les instructions devant être réalisées.

Quelques exemples d'algorithmes et de leur utilité :

- Routing Internet : rechercher les routes optimales pour l'acheminement des paquets de données échangés sur le réseau Internet.
- La cryptographie : sécurisation des données sensibles notamment échangées sur le Web (concepts de clés de chiffrement et de signature numérique).
- Calculs statistiques : calcul de moyennes, variances, maximum, minimum, nombre d'occurrences, etc...
- Allocation optimale de ressources : dans la production, la finance, le stockage, les achats, etc...
- Etc...

Mais pourquoi passer par de l'algorithmique (et donc l'étudier) ?

Le travail d'un développeur est bien sûr de programmer. Mais... ! Qu'est-ce que ce développeur va effectivement programmer ? En fait, le travail du développeur est de réfléchir et de penser correctement son code source (qui est, pour rappel, à la base du programme). Ce processus de réflexion est donc appelé *l'algorithmique*. L'objectif principal de ce cours, bien qu'illustré par un langage de développement (Java), est donc de vous aider à acquérir ce processus de réflexion indispensable à votre futur métier. Dès lors, préparez vos méninges et laissez-vous guider par les différentes leçons proposées dans cette formation. Dans quelques années, devant votre environnement informatique de programmation, vous repenserez à ce cours et à tout ce que vous aurez appris. Il vous aidera en effet dans votre travail quotidien 😊 .

Les structures algorithmiques

On peut identifier trois structures algorithmiques de base que nous allons voir tout au long de cette formation :

1. La **séquence** : réaliser une suite d'instructions les unes après les autres.
2. La **structure alternative** : sur base d'un choix, réaliser telle ou telle suite d'instructions.
3. La **structure itérative** : répéter plusieurs fois une même suite d'instructions