

Une introduction au concept de programmation et à son environnement

Site: [Plateforme Moodle Technofutur Tic](#)
Cours: Algorithmique Java 2023
Livres: Une introduction au concept de programmation et à son
environnement

Imprimé par: Jean-Christophe Miler
Date: Wednesday 9 August 2023, 18:21

Description

Temps estimé : environ 1h

Table des matières

- 1. La notion d'ordinateur et de programme informatique**
- 2. L'algorithmique : activité au coeur du métier de programmeur**
- 3. Notion de langage de programmation pour comprendre ce qu'est Java**

1. La notion d'ordinateur et de programme informatique

Bienvenue dans ce cours d'algorithmiquement à l'aide de Java !

Avant de commencer, nous allons introduire certains concepts pour pouvoir partir sur de bonnes bases. Commençons par rapidement expliquer ce qu'est un **ordinateur** ainsi qu'un **programme informatique** (portant aussi le nom de **logiciel** ou de **software**).

Quel est le lien entre ces concepts ? Hé bien ce sont les **sciences informatiques**. KESAKO ? Les sciences informatiques correspondent à l'étude du *traitement automatique* de l'*information* par des machines telles que les ordinateurs.

Un **ordinateur** est en fait une machine programmable, c-à-d. une machine dont on peut changer le fonctionnement grâce à des programmes. Changer le fonctionnement signifie que l'on peut ajouter ou retirer des fonctionnalités à la machine programmable. Pensez pas exemple à votre smartphone. On peut y ajouter ou retirer des applications mobiles. Réaliser cela signifie que l'on ajoute ou retire des fonctionnalités à votre smartphone. Si vous ajoutez une application de boussole, et bien votre smartphone pourra alors vous donner l'orientation du Nord, du Sud, etc...

Un **programme informatique** est une séquence d'instructions spécifiant les opérations à appliquer sur des données en vue d'obtenir un résultat prédéterminé. Ces programmes informatiques permettent donc de changer les instructions de la machine programmable pour ajouter/retirer des fonctionnalités à l'ordinateur.

Le travail du programmeur d'applications est de déterminer quelles seront ces instructions et de les programmer dans un langage de programmation. Dans le cadre de ce cours, nous travaillerons avec le langage Java. Mais il en existe bien évidemment beaucoup d'autres. Envie d'en avoir une liste assez longue... C'est par ici : https://fr.wikipedia.org/wiki/Liste_de_langages_de_programmation.

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

3. Notion de langage de programmation pour comprendre ce qu'est Java

Comme expliqué dans la section précédente, un **langage de programmation** est un langage formel constitué d'un vocabulaire et de règles. Le vocabulaire comprend l'ensemble des signes et des termes que l'on peut utiliser comme instructions et, dès lors, qui seront compris par l'ordinateur. La plupart du temps, ces termes sont des mots en anglais. Nous en verrons toute une série ensemble durant ce cours. Les règles sont des contraintes décrivant comment les signes et les termes peuvent être utilisés et agencés ensemble. Les règles permettent donc de structurer le langage de programmation alors que le vocabulaire permet de décrire le contenu des instructions.

Java est donc un langage de programmation. Il est très connu et très utilisé dans les entreprises (attention : un langage très utilisé représente à peine 15 à 20% des développements). Un article intéressant à ce sujet sur Developpez.com est disponible ici

: <https://www.developpez.com/actu/185087/Quels-sont-les-langages-de-programmation-les-plus-utilises-par-les-developpeurs-Une-analyse-des-evenements-publics-sur-GitHub/>.

Si vous connaissez Java, est-ce suffisant ?

Cette question est difficile en fait. Disons que toute une série d'entreprises utilisent exclusivement Java mais toute une série d'autres entreprises demandent la maîtrise de plusieurs langages de programmation. Néanmoins, une fois que vous connaissez un langage, le second est plus facile à apprendre. Une fois que vous en connaissez deux, le troisième est encore plus facile, etc...

Comment pratiquement développer ?

Pour programmer une application, et bien nous utilisons bien évidemment une autre application. Ces applications portent un nom : IDE pour **integrated development environment**. Ces IDE sont utilisés depuis le début des années 1980. Aujourd'hui, plus aucun programmeur ne travaille sans un IDE.

Les IDE comprennent un ensemble d'outils qui permettent de faciliter le travail des programmeurs (et bien évidemment, d'en augmenter la productivité). Les IDE contiennent normalement un éditeur de texte, des boutons d'action représentant des raccourcis de programmation, un outil de compilation ainsi qu'un débogueur permettant de traiter les bugs de programmation.

Les principaux IDE Java sont : NetBeans, Eclipse, DrJava, JDeveloper, IntelliJ, etc...

Dans le cadre de cette formation, nous allons utiliser un IDE online pour faciliter votre travail. Il s'agit de REPL (<https://repl.it>). Il est pédagogique et facile d'usage. Il n'est pas recommandé pour une utilisation industrielle.

Qu'est-ce qu'un compilateur ?

Un **compilateur** est en fait un programme informatique permettant de transformer le code source programmé par le développeur en un logiciel interprétable par l'ordinateur. N'oublions pas que l'ordinateur comprend des suites de 0 et de 1. Dès lors, le code source Java contenant de nombreux signes et termes proches du langage humain doit être transformé pour être compréhensible par la machine. Ce processus de transformation s'appelle la compilation.