Algorithmique et programmation en Java

Cours et exercices corrigés

4e édition

RESSOURCES NUMÉRIQUES

DUNOD

Vincent Granet

Algorithmique et programmation en Java

Cours et exercices corrigés

4e édition

DUNOD

Illustration de couverture :

Abstract background - Spheres © Dreaming Andy – Fotolia.com

5 rue Laromiguière, 75005 Paris
www.dunod.com

ISBN 978-2-10-071287-8

à Maud

Avant-propos

Longtemps attendue, la version 8 de JAVA est sortie en mars 2014, avec de nombreuses nouveautés. Sans doute, la plus importante est l'ajout des *fonctions anonymes* (lambda expressions) et son incidence tant sur le langage que sur son API.

Pour cette quatrième édition d'*Algorithmique et Programmation en Java*, l'ensemble de l'ouvrage a été révisé pour tenir compte des *fonctions anonymes*. En particulier, il inclut un nouveau chapitre qui présente le paradigme fonctionnel et la façon dont il est mis en œuvre avec JAVA 8.

L'informatique est une science mais aussi une technologie et un ensemble d'outils. Ces trois composantes ne doivent pas être confondues, et l'enseignement de l'informatique ne doit pas être réduit au seul apprentissage des logiciels. Ainsi, l'activité de programmation ne doit pas se confondre avec l'étude d'un langage de programmation particulier. Même si l'importance de ce dernier ne doit pas être sous-estimée, il demeure un simple outil de mise en œuvre de concepts algorithmiques et de programmation généraux et fondamentaux. L'objectif de cet ouvrage est d'enseigner au lecteur des méthodes et des outils de construction de programmes informatiques valides et fiables.

L'étude de l'algorithmique et de la programmation est un des piliers fondamentaux sur lesquels repose l'enseignement de l'informatique. Ce livre s'adresse principalement aux étudiants des cycles informatiques et élèves ingénieurs informaticiens, mais aussi à tous ceux qui ne se destinent pas à la carrière informatique mais qui seront certainement confrontés au développement de programmes informatiques au cours de leur scolarité ou dans leur vie professionnelle.

Les seize premiers chapitres présentent les concepts de base de la programmation *impérative* en s'appuyant sur une *méthodologie objet*. Le chapitre 13 est également dédié à la *programmation fonctionnelle*. Ils mettent en particulier l'accent sur la notion de preuve des programmes grâce à la notion d'*affirmations* (antécédent, conséquent, invariant) dont la vérification formelle garantit la validité de programmes. Ils introduisent aussi la notion de *complexité* des algorithmes pour évaluer leur performance.

Les onze derniers chapitres étudient en détail les structures de données abstraites classiques (liste, graphe, arbre...) et de nombreux d'algorithmes fondamentaux (recherche, tri, jeux et stratégie...) que tout étudiant en informatique doit connaître et maîtriser. D'autre part, un chapitre est consacré aux interfaces graphiques et à leur programmation avec SWING.

La présentation des concepts de programmation cherche à être indépendante, autant que faire se peut, d'un langage de programmation particulier. Les algorithmes seront décrits dans une notation algorithmique épurée. Pour des raisons pédagogiques, il a toutefois bien fallu faire le choix d'un langage pour programmer les structures de données et les algorithmes présentés dans cet ouvrage. Ce choix s'est porté sur le langage à objets JAVA [GJS96], non pas par effet de mode, mais plutôt pour les qualités de ce langage, malgré quelques défauts. Ses qualités sont en particulier sa relative simplicité pour la mise en œuvre des algorithmes, un large champ d'application et sa grande disponibilité sur des environnements variés. Ce dernier point est en effet important ; le lecteur doit pouvoir disposer facilement d'un compilateur et d'un interprète afin de résoudre les exercices proposés à la fin des chapitres. Enfin, JAVA est de plus en plus utilisé comme langage d'apprentissage de la programmation dans les universités. Pour les défauts, on peut par exemple regretter l'absence de l'héritage multiple, et la présence de constructions archaïques héritées du langage C. Ce livre n'est toutefois pas un ouvrage d'apprentissage du langage JAVA. Même si les éléments du langage nécessaires à la mise en œuvre des notions d'algorithmique et de programmation ont été introduits, ce livre n'enseignera pas au lecteur les finesses et les arcanes de JAVA, pas plus qu'il ne décrira les nombreuses classes de l'API. Le lecteur intéressé pourra se reporter aux très nombreux ouvrages qui décrivent le langage en détail, comme par exemple [GR11, Bro99, Ska00, Eck00].

Les corrigés de la plupart des exercices, ainsi que des applets qui proposent une vision graphique de certains programmes présentés dans l'ouvrage sont accessibles sur le site web de l'auteur à l'adresse :

`www.polytech.unice.fr/~vg/fr/livres/algojava`

Cet ouvrage doit beaucoup à de nombreuses personnes. Tout d'abord, aux auteurs des algorithmes et des techniques de programmation qu'il présente. Il n'est pas possible de les citer tous ici, mais les références à leurs principaux textes sont dans la bibliographie. À Olivier Lecarme et Jean-Claude Boussard, mes professeurs à l'université de Nice qui m'ont enseigné cette discipline au début des années 1980. Je tiens tout particulièrement à remercier ce dernier qui fut le tout premier lecteur attentif de cet ouvrage alors qu'il n'était encore qu'une ébauche, et qui m'a encouragé à poursuivre sa rédaction. À Carine Fédèle qui a bien voulu lire et corriger ce texte à de nombreuses reprises, qu'elle en soit spécialement remercier. Enfin, à mes collègues et mes étudiants qui m'ont aidé et soutenu dans cette tâche ardue qu'est la rédaction d'un livre.

Enfin, je remercie toute l'équipe Dunod, Carole Trochu, Jean-Luc Blanc, et Romain Hennion, pour leur aide précieuse et leurs conseils avisés qu'ils m'ont apportés pour la publication des quatre éditions de cet ouvrage.

Sophia Antipolis, avril 2014.

Table des matières

Chapitre 1

Introduction

Les informaticiens, ou les simples usagers de l'outil informatique, utilisent des systèmes informatiques pour concevoir ou exécuter des programmes d'application. Nous considérerons qu'un environnement informatique est formé d'une part d'un ordinateur et de ses équipements externes, que nous appellerons *environnement matériel*, et d'autre part d'un système d'exploitation avec ses programmes d'application, que nous appellerons *environnement logiciel*. Les programmes qui forment le logiciel réclament des méthodes pour les construire, des langages pour les rédiger et des outils pour les exécuter sur un ordinateur.

Dans ce chapitre, nous introduirons la terminologie et les notions de base des ordinateurs et de la programmation. Nous présenterons les notions d'environnement de développement et d'exécution d'un programme, nous expliquerons ce qu'est un langage de programmation et nous introduirons les méthodes de construction des programmes.

1.1 ENVIRONNEMENT MATÉRIEL

Un *automate* est un ensemble fini de composants physiques pouvant prendre des états identifiables et reproductibles en nombre fini, auquel est associé un ensemble de changements d'états non instantanés qu'il est possible de commander et d'enchaîner sans intervention humaine.

Un *ordinateur* est un automate *déterministe* à composants électroniques. Tous les ordinateurs, tout au moins les ordinateurs monoprocesseurs, sont construits, peu ou prou, sur le modèle proposé en 1944 par le mathématicien américain d'origine hongroise VON NEUMANN. Un ordinateur est muni :

- D'une *mémoire*, dite *centrale* ou *principale*, qui contient deux sortes d'informations : d'une part l'information traitante, les *instructions*, et d'autre part l'information trai-

tée, les *données*. Cette mémoire est formée d'un ensemble de cellules, ou *mots*, ayant chacune une *adresse unique*, et contenant des instructions ou des données. La représentation de l'information est faite grâce à une codification *binaire*, 0 ou 1. On appelle *longueur de mot*, caractéristique d'un ordinateur, le nombre d'éléments binaires, appelés *bits*, groupés dans une simple cellule. Les longueurs de mots usuelles des ordinateurs actuels (ou passés) sont, par exemple, 8, 16, 24, 32, 48 ou 64 bits. Cette mémoire possède une capacité *finie*, exprimée en gigaoctets (Go) ; un *octet* est un ensemble de 8 bits, un kilo-octet (Ko) est égal à 1 024 octets, un mégaoctet est égal à 1 024 Ko, un gigaoctet (Go) est égal à 1 024 Mo, et enfin un téraoctet (To) est égal à 1 024 Go. Actuellement, les tailles courantes des mémoires centrales des ordinateurs individuels varient entre 4 Go à 8 Go [1].

- D'une *unité centrale de traitement*, formée d'une *unité de commande* (UC) et d'une *unité arithmétique et logique* (UAL). L'unité de commande extrait de la mémoire centrale les instructions et les données sur lesquelles portent les instructions ; elle déclenche le traitement de ces données dans l'unité arithmétique et logique, et éventuellement range le résultat en mémoire centrale. L'unité arithmétique et logique effectue sur les données qu'elle reçoit les traitements commandés par l'unité de commande.
- De *registres*. Les registres sont des unités de mémorisation, en petit nombre (certains ordinateurs n'en ont pas), qui permettent à l'unité centrale de traitement de ranger de façon temporaire des données pendant les calculs. L'accès à ces registres est très rapide, beaucoup plus rapide que l'accès à une cellule de la mémoire centrale. Le rapport entre les temps d'accès à un registre et à la mémoire centrale est de l'ordre de 100.
- D'*unités d'échanges* reliées à des périphériques pour échanger de l'information avec le monde extérieur. L'unité de commande dirige les unités d'échange lorsqu'elle rencontre des instructions d'entrée-sortie.

Jusqu'au milieu des années 2000, les constructeurs étaient engagés dans une course à la vitesse avec des microprocesseurs toujours plus rapides. Toutefois, les limites de la physique actuelle ont été atteintes et depuis 2006 la tendance nouvelle est de placer plusieurs (le plus possible) microprocesseurs sur une même puce (le circuit-intégré). Ce sont par exemple les processeurs 64 bits Core i7 d'INTEL ou AMD FX d'AMD, qui possèdent de 4 à 8 processeurs.

Les ordinateurs actuels possèdent aussi plusieurs niveaux de mémoire. Ils introduisent, entre le processeur et la mémoire centrale, des mémoires dites *caches* qui accélèrent l'accès aux données. Les mémoires caches peuvent être *primaires*, c'est-à-dire situées directement sur le processeur, ou *secondaires*, c'est-à-dire situées sur la carte mère. Certains ordinateurs introduisent même un troisième niveau de cache. En général, le rapport entre le temps d'accès entre les deux premiers niveaux de mémoire cache est d'environ 10 (le cache de niveau 1 est le plus rapide et le plus petit). Le temps d'accès entre la mémoire cache secondaire et la mémoire centrale est lui aussi d'un rapport d'environ 10.

Les *équipements externes*, ou *périphériques*, sont un ensemble de composants permettant de relier l'ordinateur au monde extérieur, et notamment à ses utilisateurs humains. On peut distinguer :

1. Notez qu'avec 32 bits, l'espace adressage est de 4 Go mais qu'en général les systèmes d'exploitation ne permettent d'utiliser qu'un espace mémoire de taille inférieure. Les machines 64 bits actuelles, avec un système d'exploitation adapté, permettent des tailles de mémoire centrale supérieures à 4 Go.