

Claude Delannoy

Programmer en Java

9^e édition

© Groupe Eyrolles, 2000-2014, ISBN : 978-2-212-14007-1

EYROLLES



Avant-propos

À qui s'adresse ce livre

Cet ouvrage est destiné à tous ceux qui souhaitent maîtriser la programmation en Java. Il s'adresse à la fois aux étudiants, aux développeurs et aux enseignants en informatique.

Il suppose que le lecteur possède déjà une expérience de la programmation dans un autre langage (C, C++, Visual PHP, Python...). En revanche, la connaissance de la programmation orientée objet n'est nullement nécessaire, pas plus que celle de la programmation d'interfaces graphiques ou d'applications Web.

Contenu de l'ouvrage

Les fondements de Java

Les chapitres 1 à 11 sont consacrés aux fondements du langage : types primitifs, opérateurs et expressions, instructions, classes, héritage, tableaux et chaînes de caractères. Les aspects les plus fondamentaux de la programmation orientée objet que sont le polymorphisme, la surdéfinition et la redéfinition des méthodes y sont également étudiés de façon approfondie, aussi bien dans leur puissance que dans leurs limitations.

Tous les aspects du langage sont couverts, y compris ceux qui sont spécifiques à Java, comme les interfaces, les classes internes, les classes anonymes, les exceptions ou les threads. Les moins usités font généralement l'objet d'un paragraphe intitulé *Informations complémentaires* dont la connaissance n'est pas indispensable à l'étude de la suite de l'ouvrage.

Par ailleurs, le chapitre 21 présente les possibilités de programmation générique introduites par Java 5. Sa place tardive dans l'ouvrage est surtout justifiée par son lien étroit avec les collections (génériques depuis Java 5) présentées au chapitre 22

Le chapitre 23 a été ajouté à cette nouvelle édition pour présenter les expressions lambda, introduites par Java 8.

Puis, le chapitre 24 présente les annotations (bien intégrées dans le langage depuis Java 6) et décrit en même temps les possibilités d'introspection qui permettent d'en tirer véritablement profit.

Enfin, bien qu'il ne s'agisse plus de fondements du langage, mais plutôt de techniques de développement, nous avons jugé bon d'introduire les populaires *Design Patterns*. Le chapitre 28 propose l'implémentation en Java des modèles les plus répandus : *Factory Method*, *Abstract Factory*, *Composite*, *Adapter*, *Decorator*, *Strategy*, *Template Method* et *Observer*.

Les principales API

Le JDK (*Java Development Kit*) de Java est livré, en standard, avec différentes bibliothèques, paquetages ou API (*Application Programming Interface*) fournissant de nombreuses classes utilitaires. Les chapitres 12 à 20, 22, 23 et 25 à 27 examinent les API qui correspondent aux besoins les plus universels et qui, à ce titre, peuvent être considérés comme partie intégrante du langage.

Les chapitres 12 à 19 sont consacrés à la programmation d'interfaces graphiques en Java à l'aide de l'API nommée *Swing* : événements et écouteurs ; boutons, cases à cocher et boutons radio ; boîtes de dialogue ; menus ; barres d'outils ; actions abstraites ; événements générés par le clavier, la souris, les fenêtres et la focalisation ; questionnaires de mise en forme ; affichage de textes et de dessins ; applets. Dans cette partie, l'accent est mis sur les mécanismes fondamentaux qui interviennent en programmation graphique et événementielle.

Le chapitre 20 traite de l'API relative aux entrées-sorties, unifiées à travers la notion de flux. Il intègre les nouveautés les plus importantes introduites par Java 7 (nommées NIO.2).

Le chapitre 22 décrit les principales structures de données qu'on regroupe souvent sous le terme de collection : listes, ensembles, vecteurs dynamiques, queues et tables associatives.

Le chapitre 23 a été ajouté à cette nouvelle édition pour étudier les streams introduits par Java 8. Notez que ceux-ci ont été placés dans le même chapitre que les expressions lambda, compte tenu de leur fréquente utilisation conjointe.

Le chapitre 25 a, lui aussi, été ajouté à cette nouvelle édition pour décrire la nouvelle API de gestion des temps et des dates, introduite par Java 8.

Le chapitre 26 se veut une introduction aux possibilités de programmation côté serveur, offertes par les servlets, les JSP et les JavaBeans. En toute rigueur, il s'agit là, non plus d'API standards de Java, mais de spécifications de Java EE (*Java Enterprise Edition*).

Enfin, le chapitre 27 présente l'API standard JDBC permettant d'exploiter des bases de données locales ou distantes.

Pour aller plus loin

Après l'étude de cet ouvrage consacré à ce que l'on pourrait appeler les « bases élargies du langage », le lecteur pourra appréhender aisément l'importante documentation des classes standards Java et de leurs méthodes¹. Il sera alors parfaitement armé pour développer ses propres applications, aussi complexes et spécialisées soient-elles, notamment les applications côté serveur à base d'EJB ou les applications distribuées, sujets non traités dans cet ouvrage.

Forme de l'ouvrage

L'ouvrage est conçu sous la forme d'un cours. Il expose progressivement les différentes notions fondamentales, en les illustrant systématiquement de programmes complets accompagnés d'un exemple d'exécution.

Pour en faciliter l'assimilation, les fondements du langage sont présentés de façon indépendante de la programmation d'interfaces graphiques, en s'appuyant sur les possibilités qu'offre Java d'écrire des applications à interface *console*.

Dans la partie consacrée à la programmation graphique, les composants sont introduits suffisamment progressivement pour permettre au lecteur de les découvrir en tant qu'utilisateur de logiciel. L'expérience montre en effet, que, pour réaliser une bonne interface graphique, un développeur doit non seulement savoir programmer correctement les composants concernés, mais également bien connaître leur ergonomie.

Outre son caractère didactique, nous avons conçu l'ouvrage d'une manière très structurée pour qu'il puisse être facilement consulté au-delà de la phase d'apprentissage du langage. Dans cet esprit, il est doté d'une table des matières détaillée et d'un index fourni dans lequel les noms de méthodes sont toujours accompagnés du nom de la classe correspondante (il peut y avoir plusieurs classes). Les exemples complets peuvent servir à une remémoration rapide du concept qu'ils illustrent. Des encadrés permettent de retrouver rapidement la syntaxe d'une instruction, ainsi que les règles les plus importantes. Enfin, des annexes fournissent des aide-mémoire faciles à consulter :

- liste des fonctions mathématiques (classe *Math*) ;
- liste des exceptions standards ;
- liste des composants et des en-têtes de leurs méthodes ;
- liste des événements, écouteurs et méthodes correspondantes ;

1. Par exemple, en consultant le site officiel de Java : www.oracle.com/technetwork/java/.

- liste des classes et interfaces liées aux collections et méthodes correspondantes ;
- outils de professionnalisation des applications (pour la plupart introduits par Java 6).

L'ouvrage, les versions de Java et C++

Si les instructions de base de Java n'ont pratiquement pas évolué depuis sa naissance jusqu'à sa version 5, il n'en va pas de même de ses bibliothèques standards. En particulier, le modèle de gestion des événements a été fortement modifié par la version 1.1. Une nouvelle bibliothèque de composants graphiques, Swing, est apparue dans la version 1.2 de l'édition Standard de Java, renommée à cette occasion J2SE (*Java 2 Standard Edition*). Après deux nouvelles versions nommées respectivement J2SE 1.3 et J2SE 1.4, Sun a modifié son système de numérotation en introduisant J2SE 5 en 2004, puis Java SE 6 en 2006, Java SE 7 en 2011, enfin Java SE 8 en 2014. Nous parlerons plus simplement de Java 5, Java 6, Java 7 et Java 8 pour nous référer à ces dernières.

Depuis J2SE 1.2, chaque édition Standard de Java complétée par un ensemble de spécifications, nommé J2EE (*Java 2 Enterprise Edition*) jusqu'à la version 4 et Java EE (*Java Enterprise Edition*) depuis la version 5 ; ces spécifications sont dédiées notamment au développement côté serveur et aux applications réparties¹.

D'une manière générale, l'ouvrage tient compte de l'historique du langage en mentionnant au fil du texte les apports des versions successives depuis Java 5. Citons :

- les nouveautés fondamentales introduites par Java 5 : types énumérés, types enveloppes, boxing/unboxing automatiques, arguments variables en nombre, boucle *for... each*, programmation générique (chapitre 21) ; notez que le chapitre relatif aux collections est prévu pour tenir compte de leur aspect générique, mais aussi pour permettre l'utilisation d'anciens codes² ;
- les apports de Java 6 : nouveau gestionnaire de mise en forme *GroupLayout*, fonctionnalités permettant de professionnaliser une application (classe *Desktop*, classe *Console*, action sur la barre des tâches du système), nouvelles interfaces et classes de collections (*Deque*, *AraryDeque*, *NavigableSet*, *NavigableMap*) ;
- les apports de Java 7 : emploi de chaînes dans l'instruction *switch*, gestion des *catch* multiples, gestion automatique des ressources dans un bloc *try*, nouvelles possibilités de gestion de flux dites NIO.2 ;
- les importantes nouveautés introduites par Java 8 : les expressions lambda et les streams qui font l'objet du chapitre 23 ; la nouvelle bibliothèque de gestion des temps et des dates qui fait l'objet du chapitre 25.

1. Il existe une troisième édition de Java, Java ME (*Java Micro Edition*), destinée aux développements d'applications embarquées pour les téléphones mobiles et divers appareils électroniques grand public.

2. Avec le temps, les remarques correspondantes finissent par avoir un caractère plus historique qu'opérationnel.

Compte tenu de la popularité du langage C++, nous avons introduit de nombreuses remarques titrées *En C++*. Elles mettent l'accent sur les liens étroits qui existent entre Java et C++, ainsi que sur leurs différences. Elles offriront des passerelles utiles non seulement au programmeur C++ qui apprend ici Java, mais également au lecteur qui, après la maîtrise de Java, souhaitera aborder l'étude de C++¹.

1. L'ouvrage *Programmer en langage C++*, du même auteur, chez le même éditeur, s'adresse à un public ayant déjà la maîtrise d'un langage tel que Java.