

**Jérôme Molière**

Préfaces de Cosme Charlier (Cdiscount), Michel Maudet (Linagora) et Alexandre Faria (Decathlon)

# Java

pour les professionnels

Outils et méthodes



SUR LE MÊME THÈME

E. SARRION. - **JavaScript - Vue.js côté client et Node.js côté serveur.**  
N°0100836, 2022, 240 pages.

C. DELANNOY. - **Programmer en Java - Couvre Java 10 à Java 14.**  
N°0100018, 11<sup>e</sup> édition, 2022, 962 pages.

A. TASSO. - **Le livre de Java premier langage - Avec 109 exercices corrigés.**  
N°0100703, 14<sup>e</sup> édition, 2022, 600 pages.

C. DELANNOY. - **Exercices en Java - 175 exercices corrigés (couvre Java 8).**  
N°67385, 4<sup>e</sup> édition, 2017, 346 pages.

Retrouvez nos bundles (livres papier + e-book) et livres numériques sur  
<http://izibook.eyrolles.com>

**Jérôme Molière**

Préfaces de Cosme Charlier (Cdiscount), Michel Maudet (Linagora) et Alexandre Faria (Decathlon)

# Java pour les professionnels

Outils et méthodes

● Éditions  
**EYROLLES**

ÉDITIONS EYROLLES  
61, bd Saint-Germain  
75240 Paris Cedex 05  
www.editions-eyrolles.com

Mise en pages : Gaël Thomas

*Depuis 1925, les éditions Eyrolles s'engagent en proposant des livres pour comprendre le monde, transmettre les savoirs et cultiver ses passions !*

*Pour continuer à accompagner toutes les générations à venir, nous travaillons de manière responsable, dans le respect de l'environnement. Nos imprimeurs sont ainsi choisis avec la plus grande attention, afin que nos ouvrages soient imprimés sur du papier issu de forêts gérées durablement. Nous veillons également à limiter le transport en privilégiant des imprimeurs locaux. Ainsi, 89 % de nos impressions se font en Europe, dont plus de la moitié en France.*

En application de la loi du 11 mars 1957, il est interdit de reproduire intégralement ou partiellement le présent ouvrage, sur quelque support que ce soit, sans l'autorisation de l'Éditeur ou du Centre français d'exploitation du droit de copie, 20, rue des Grands Augustins, 75006 Paris.

© Éditions Eyrolles, 2023, ISBN : 978-2-416-01173-3

# Table des matières

---

|                                                        |          |
|--------------------------------------------------------|----------|
| <b>Avant-propos</b> .....                              | <b>1</b> |
| Origines de ce livre .....                             | 1        |
| Objectifs .....                                        | 1        |
| Sources d'inspiration .....                            | 2        |
| Remerciements .....                                    | 2        |
| Dédicace .....                                         | 2        |
| <br>                                                   |          |
| <b>CHAPITRE 1</b>                                      |          |
| <b>Concepts fondamentaux</b> .....                     | <b>3</b> |
| Pourquoi parler de programmation fonctionnelle ? ..... | 3        |
| Là où les soucis commencent .....                      | 6        |
| Les piliers de la programmation fonctionnelle .....    | 10       |
| Effets de bord .....                                   | 11       |
| Fonctions pures .....                                  | 11       |
| Fonction d'ordre supérieur .....                       | 12       |
| Non-mutabilité .....                                   | 14       |
| VAVR et la programmation fonctionnelle .....           | 14       |
| Survol de VAVR .....                                   | 15       |
| Fonctions en tant qu'objets standards .....            | 15       |
| Fonctions partielles .....                             | 17       |
| Évaluation tardive de fonctions avec Lazy .....        | 18       |
| Memoization .....                                      | 19       |
| Currying .....                                         | 26       |
| Structures de données fonctionnelles .....             | 26       |
| <i>Option</i> .....                                    | 26       |
| <i>Either</i> .....                                    | 27       |
| <i>Tuple</i> .....                                     | 29       |
| <i>Try</i> .....                                       | 30       |
| <i>Pattern matching</i> .....                          | 30       |
| En pratique .....                                      | 40       |
| Détection d'une année bissextile .....                 | 41       |
| Koans fonctionnels .....                               | 42       |
| Conclusion .....                                       | 42       |

## CHAPITRE 2

**Kata poker..... 43**

|                                         |    |
|-----------------------------------------|----|
| Le projet .....                         | 43 |
| Domaine fonctionnel .....               | 43 |
| Rappels sur le TDD .....                | 44 |
| Mise en œuvre du TDD .....              | 45 |
| Détecter une paire .....                | 45 |
| <i>Tests unitaires</i> .....            | 45 |
| <i>Code de la fonction</i> .....        | 46 |
| <i>Classe modélisant une main</i> ..... | 47 |
| Détecter un brelan .....                | 47 |
| <i>Tests unitaires</i> .....            | 47 |
| <i>Code de la fonction</i> .....        | 48 |
| <i>Refactoring</i> .....                | 48 |
| Détecter un carré .....                 | 49 |
| <i>Tests unitaires</i> .....            | 49 |
| <i>Code de la fonction</i> .....        | 49 |
| Détecter une double paire .....         | 50 |
| <i>Tests unitaires</i> .....            | 50 |
| <i>Code de la fonction</i> .....        | 50 |
| Détecter un full house .....            | 51 |
| <i>Tests unitaires</i> .....            | 51 |
| <i>Code de la fonction</i> .....        | 52 |
| Détecter une couleur .....              | 52 |
| <i>Tests unitaires</i> .....            | 52 |
| <i>Code de la fonction</i> .....        | 53 |
| <i>Refactoring</i> .....                | 53 |
| Détecter une quinte .....               | 55 |
| <i>Tests unitaires</i> .....            | 55 |
| <i>Code de la fonction</i> .....        | 56 |
| Détecter une quinte flush .....         | 57 |
| <i>Tests unitaires</i> .....            | 57 |
| <i>Code de la fonction</i> .....        | 57 |
| Détecter une quinte flush royale .....  | 58 |
| Comparer deux mains .....               | 58 |
| Conclusion .....                        | 59 |

## CHAPITRE 3

**Conception fonctionnelle ..... 61**

|                                               |    |
|-----------------------------------------------|----|
| Limites de la conception traditionnelle ..... | 61 |
| Rendre visible notre logique métier .....     | 63 |
| Conclusion .....                              | 64 |

## CHAPITRE 4

|                                                                     |           |
|---------------------------------------------------------------------|-----------|
| <b>Tests basés sur les propriétés (PBT)</b> .....                   | <b>65</b> |
| Pourquoi opter pour le PBT .....                                    | 65        |
| Limites des tests par l'exemple .....                               | 65        |
| PBT .....                                                           | 66        |
| Exemple simpliste avec VAVR Testing .....                           | 67        |
| Autour du PBT .....                                                 | 69        |
| PBT et monkey testing .....                                         | 69        |
| Autres frameworks de PBT .....                                      | 69        |
| PBT, fuzzing et Mutation Based Testing .....                        | 71        |
| Détourner le PBT de son usage premier .....                         | 72        |
| Problématique fréquente pendant les tests .....                     | 72        |
| Vers une gestion des données de test avec le TDM .....              | 73        |
| Prendre le contrôle de l'aléatoire avec VAVR-Property Testing ..... | 73        |
| Utilisation concrète du générateur JQwik avec un exemple .....      | 75        |
| Conclusion .....                                                    | 79        |

## CHAPITRE 5

|                                                      |           |
|------------------------------------------------------|-----------|
| <b>Loom</b> .....                                    | <b>81</b> |
| Rappels sur le modèle du temps partagé .....         | 81        |
| Gestion des processus légers .....                   | 81        |
| Partage du temps processeur .....                    | 82        |
| Un contre-exemple en Java .....                      | 83        |
| Loom et les threads virtuels .....                   | 84        |
| Des threads invisibles au niveau du système .....    | 84        |
| Les threads virtuels de Loom .....                   | 86        |
| <i>Création de threads virtuels.</i> .....           | 86        |
| <i>Threads virtuels et cycle de vie.</i> .....       | 87        |
| <i>Threads virtuels et threads plate-forme</i> ..... | 88        |
| <i>Intérêt des threads virtuels.</i> .....           | 89        |
| Pratiquer ces concepts .....                         | 89        |
| Conclusion .....                                     | 89        |

## CHAPITRE 6

|                                                         |           |
|---------------------------------------------------------|-----------|
| <b>Microbenchmarking avec JMH</b> .....                 | <b>91</b> |
| Rappels et éléments structurants .....                  | 91        |
| Sans objectif, pas de performance .....                 | 91        |
| L'intuition est la plus mauvaise des conseillères ..... | 92        |
| Savoir comparer .....                                   | 93        |
| Microbenchmarking en Java .....                         | 94        |
| Référentiel guidant le choix .....                      | 94        |
| Un premier microbenchmark .....                         | 94        |
| JMH .....                                               | 95        |

|                                  |            |
|----------------------------------|------------|
| Cas concret .....                | 95         |
| Projet .....                     | 96         |
| Premier benchmark avec JMH ..... | 96         |
| Profiling de code avec JMH ..... | 99         |
| <b>Conclusion</b> .....          | <b>100</b> |
| <b>Défi</b> .....                | <b>101</b> |

## CHAPITRE 7

### **Robustesse applicative ..... 103**

|                                                  |            |
|--------------------------------------------------|------------|
| <b>Resilience4j</b> .....                        | <b>103</b> |
| Utilisation de la bibliothèque .....             | 104        |
| Canevas d'utilisation de la bibliothèque .....   | 105        |
| Mise en place du CircuitBreaker .....            | 106        |
| Mise en place du Retry .....                     | 107        |
| <b>Utiliser l'aléatoire dans vos tests</b> ..... | <b>109</b> |
| Property Based Testing .....                     | 109        |
| Fuzz testing .....                               | 110        |
| <i>Historique</i> .....                          | 110        |
| <i>Principe</i> .....                            | 111        |
| <i>Domaine d'utilisation</i> .....               | 111        |
| <b>Conclusion</b> .....                          | <b>112</b> |

## CHAPITRE 8

### **Observabilité..... 113**

|                                                     |            |
|-----------------------------------------------------|------------|
| <b>Outils de suivi</b> .....                        | <b>113</b> |
| Logs applicatifs .....                              | 113        |
| Métriques .....                                     | 114        |
| Traces .....                                        | 114        |
| <b>Gestion des logs applicatifs</b> .....           | <b>114</b> |
| Logback – Concepts essentiels .....                 | 115        |
| Maîtriser ses logs avec Logback .....               | 119        |
| <i>Utiliser le bon niveau de logs</i> .....         | 119        |
| <i>Utiliser un Appender asynchrone</i> .....        | 119        |
| <i>Limiter le nombre de messages envoyés</i> .....  | 120        |
| <i>Limiter la taille des messages envoyés</i> ..... | 121        |
| <i>Gérer les données sensibles</i> .....            | 121        |
| <b>Gestion des métriques</b> .....                  | <b>122</b> |
| Mise en œuvre élémentaire en Java .....             | 125        |
| <b>Traces</b> .....                                 | <b>133</b> |
| <b>Conclusion</b> .....                             | <b>134</b> |

### **Conclusion..... 135**

### **Index..... 137**

# Avant-propos

---

## Origines de ce livre

Pourquoi écrire un énième livre sur Java ? Pourquoi se lancer dans un tel chantier alors que la littérature abonde de titres sur ce langage ? Il me semble que beaucoup de livres expliquent comment démarrer avec ce langage ou comment utiliser tel ou tel *framework*, mais que peu (voire aucun) sont dédiés à des utilisateurs chevronnés du langage, leur offrant un moyen de passer un cap dans leur carrière. C'est ce défi que cet ouvrage va essayer de relever. Il ne s'agit donc pas d'un livre pour un débutant en langage Java.

## Objectifs

Les objectifs sont multiples, mais la cible est unique : fournir un concentré d'informations permettant de revendiquer un bon bagage technique et d'évoluer avec de meilleurs outils dans des projets futurs. Ce livre sera l'occasion pour les lecteurs de se plonger dans les thèmes suivants :

- la **programmation fonctionnelle**, ou comment dépasser les limites inhérentes à Java malgré le virage fonctionnel pris depuis Java 8 ;
- la **gestion des logs** et des traces applicatives, ainsi que la **remontée de métriques**, pour collecter l'information basique permettant de gérer efficacement son application en production ;
- la **performance** des applications Java, à travers des technologies comme le projet Loom (et ses *threads* virtuels) ou GraalVM (la VM Java conçue pour le *cloud*) ou bien encore JMBH (l'outil de *microbenchmarking*).

L'exploration de ces termes permettra de découvrir diverses bibliothèques de grande qualité et comme VAVR ou Resilience4J, permettant de passer un cap en matière d'intégration de la robustesse dans vos applications.

## Sources d'inspiration

L'idée de ce livre germe depuis quelques mois en raison de mon travail auprès des deux géants français Cdiscount et Decathlon, qui m'a permis de réunir les premières briques servant de matériau à cet ouvrage et, surtout, ma convaincu du bien-fondé de cette idée.

## Remerciements

Merci à mon entourage et à mon comité de relecteurs, ainsi qu'à Alexandre Habian (Éditions Eyrolles) pour son soutien.

## Dédicace

À ma sœur, qui aurait été si fière de corriger mes maladresses et mes fautes...