

[Critique]Les Design Patterns en Java : Les 23 modèles de conception fondamentaux

par [Eric Reboisson](#)

Date de publication : 04/03/2007

Dernière mise à jour : 04/03/2007

Présentation du livre "Les Design Patterns en Java : Les 23 modèles de conception fondamentaux", et mon avis le concernant.

- I - Présentation du livre
- II - Table des matières
- III - Résumé
- IV - Présentation des auteurs
- V - Critique
- VI - Commandez

I - Présentation du livre

II - Table des matières

- 1 Introduction
 - 2 **Introduction aux interfaces**
 - 3 ADAPTER
 - 4 FACADE
 - 5 COMPOSITE
 - 6 BRIDGE
 - 7 **Introduction à la responsabilité**
 - 8 SINGLETON
 - 9 OBSERVER
 - 10 MEDIATOR
 - 11 PROXY
 - 12 CHAIN OF RESPONSABILITY
 - 13 FLYWEIGHT
 - 14 **Introduction à la construction**
 - 15 BUILDER
 - 16 FACTORY METHOD
 - 17 ABSTRACT FACTORY
 - 18 PROTOTYPE
 - 19 MEMENTO
 - 20 **Introduction aux opérations**
 - 21 TEMPLATE METHOD
 - 22 STATE
 - 23 STRATEGY
 - 24 COMMAND
 - 25 INTERPRETER
 - 26 **Introduction aux extensions**
 - 27 DECORATOR
 - 28 ITERATOR
 - 29 VISITOR
-
- a Recommandations
 - b Solution
 - c Code source d'Oozinoz
 - d Introduction à UML

III - Résumé

Par son approche pratique fondée sur de nombreux exemples d'application, *Design Patterns en Java* vous amène à une compréhension approfondie des patterns de conception, condition sine qua non pour tirer le meilleur parti de leur puissance dans le développement d'applications en [Java](#). Ce document révolutionnaire introduit les fonctionnalités les plus récentes de Java et les meilleures pratiques aux 23 [patterns](#).

Forts de leur grande expérience en tant qu'instructeurs et programmeurs [Java](#), Steve Metsker et William Wake vous éclaireront sur chaque [pattern](#), au moyen de programmes [Java](#) réels, de diagrammes [UML](#) et d'exercices clairs et pertinents. Vous passerez rapidement de la théorie à l'application en apprenant comment écrire un meilleur code ou restructurer du code existant pour le rationaliser, le rendre plus performant et plus facile à maintenir.

Si vous êtes un programmeur [Java](#) désireux de gagner du temps grâce à l'écriture d'un code plus efficace, les explications lumineuses de cet ouvrage, étayées de nombreux exemples, conseils et techniques vous aideront à y parvenir. Vous maîtriserez la puissance des patterns et améliorerez vos programmes, tant en matière de conception que d'écriture et de maintenance du code.

IV - Présentation des auteurs

Steven John Metsker est directeur consultant chez Dominion Digital, une société de services informatiques et de réingénierie de processus métier. Il est spécialiste des techniques orientées objet sous-jacentes à la création de logiciels épurés et performants. Il est l'auteur des ouvrages *Building Parsers with Java#*, *Design Patterns Java# Workbook* et *Design Patterns in C#* (parus chez Addison-Wesley).

William C. Wake (www.xpl23.com) est consultant logiciel, coach et instructeur indépendant, avec plus de vingt années d'expérience en programmation. Il a travaillé auparavant pour Capital One Financial, DMR Trecom et VTLIS Inc. Il est l'auteur de *Refactoring Workbook* et *Extreme Programming Explored* (parus chez Addison-Wesley).

V - Critique

Cet ouvrage est le second que j'aborde concernant les [design patterns](#) en [Java](#), un domaine qu'il est aujourd'hui nécessaire de connaître lorsque l'on souhaite aspirer à devenir un développeur [Java](#) avancé.

Ce livre est une bonne introduction (et même plus) aux [design patterns](#) puisque qu'il présente les 23 principaux patterns et ceci au travers d'explications claires et illustrées. J'ai apprécié l'exemple de l'entreprise de fabrication de feux d'artifices qui est utilisé tout au long du livre afin de présenter les applications concrètes des [design patterns](#) dans le développement d'une application d'entreprise. L'approche de chaque [design pattern](#) se fait au travers d'une introduction accompagnée d'une définition, d'une mise en situation appliquée à une problématique de l'entreprise, d'exercices ponctuels en interaction avec le lecteur (le mieux pour progresser étant de réfléchir réellement aux questions posées et ensuite de lire les solutions en annexe). J'ai apprécié également les schémas [UML](#) des différents [design patterns](#) (et pour ceux qui ne sont pas familiarisés avec cette notation, vous trouverez dans le livre une annexe de présentation d'[UML](#)). Vous trouverez également dans le livre quelques trucs et astuces pour aller plus loin, notamment ce que sont "Le principe de substitution de Liskov", "La loi de Demeter", des liens vers des ressources sur [Internet](#), et les sources [Java](#) du projet présenté tout au long du livre.

Je reprocherai cependant une écriture parfois trop scolaire et peu attrayante, le sujet traité assez complexe nécessitant de capter constamment toute l'attention du lecteur (j'ai du parfois relire trois fois le même passage pour bien comprendre).

Je conseillerais donc "Les Design Patterns en Java : Les 23 modèles de conception fondamentaux" à tous les développeurs [Java](#) qui souhaitent aller plus loin et approfondir leurs connaissances avec la mise en pratique des [designs patterns](#).

VI - Commandez

Commandez sur www.amazon.fr : 39,90 EUR TTC seulement - **Livraison Gratuite !!!**