

Design Patterns Head First

par Matthieu Brucher (<http://matthieu-brucher.developpez.com/>) (Blog)

Date de publication : 12/05/2006

Dernière mise à jour : 28/08/2006

Critique du livre Design Patterns Head First d'*Eric Freeman et Elisabeth Freeman*

- I - Description de l'éditeur
- II - Table des matières
- III - Critique : Une introduction aux Design Patterns agréable
- IV - Liens annexes

I - Description de l'éditeur



Si vous souhaitez vous distinguer des développeurs ordinaires et devenir une force de proposition pour les architectures d'applications, vous devez maîtriser les design patterns. Pour cela, rien ne vaut l'expérience de ceux qui ont déjà été confrontés aux mêmes problèmes. Vous profiterez à la fois de leur savoir-faire et de ce qui se fait de mieux en matière de conception. Qui plus est, vous gagnerez un temps précieux que vous occuperez bien mieux à autre chose de plus complexe et de plus amusant, ce qui fera le bonheur de vos chefs. Le plus important demeure que vous ne voulez pas vous endormir en apprenant les design patterns.

Si vous connaissez déjà la collection Tête la première, vous savez déjà à quoi vous en tenir. Voici les principaux bénéfices que vous tirerez de cet ouvrage :

- Distinguer les principaux patterns (Fabrique, Singleton, Adaptateur, Itérateur, Composite, etc.);
- Apprendre à les utiliser : quand et comment les appliquer dans vos conceptions, sans pour autant vous laisser gagner par la patternite
- Contrôler l'accès aux objets avec des proxies
- Maîtriser les mécanismes MVC et de son acolyte Web : MVC 2
- Ne plus se sentir désœuvré face à des problèmes de conception en apprenant à réaliser des conceptions souples qui s'adaptent à vos besoins
- Arriver à reconnaître les design patterns qui vous entourent au quotidien dans les API Java, les frameworks ou autres applications, et comprendre enfin comment ils fonctionnent
- Réaliser des conceptions bien pensées et faciles à maintenir.

Sous des dehors amusants et ludiques, cet ouvrage n'en demeure pas moins un guide d'apprentissage pas à pas. L'approche des auteurs est axée sur la pédagogie. Les concepts s'appuient sur des images et des exercices qui stimulent le cerveau. Jeux, images, interviews, mots-croisés, sont autant de moyens de vous amener à exploiter intelligemment les design patterns et de vous transmettre une vision claire et pragmatique des problèmes de conception. L'heure est venue de vous plonger dans les design patterns, tête la première !

II - Table des matières

- Bienvenue aux Design Patterns
- Tenez vos objets au courant : le pattern Observateur
- Décorez les objets : le pattern Décorateur
- Un peu de cuisine orientée objet : les pattern fabriques
- Des objets uniques en leur genre : le pattern Singleton
- Encapsuler l'invocation : le pattern Commande
- Savoir s'adapter : les patterns Adaptateurs et Façade
- Encapsuler les algorithmes : le pattern Patron de méthode
- Des collections bien gérées : les patterns Itérateurs et Composite
- L'état des choses : le pattern Etat
- Contrôler l'accès aux objets : le pattern Proxy
- Patterns de Patterns : patterns composés
- Les patterns dans le monde réel : Mieux vivre avec les patterns

III - Critique : Une introduction aux Design Patterns agréable

Quand on passe d'un livre-catalogue tel *Design Patterns* à celui-ci, on sent un grand soulagement. C'est agréable à lire, c'est clair, structuré. Chaque chapitre présente un *pattern* ou une famille de *patterns*, par exemple les *Factories*. A chaque fois, on a une petite histoire, un mini projet à suivre et qu'il faut construire. L'avantage est clairement qu'on sait où on pourrait appliquer ces méthodes, ça permet de gagner du temps.

Il y a pleins de petits exercices à faire, simples mais efficaces. Les illustrations sont des diagrammes - il le faut bien - ou des petites photos ou caricatures humoristiques. L'humour a une place primordiale, les auteurs ont bien compris que pour intéresser, il faut accrocher, et aussi accrocher par l'humour. On est vraiment plus dans un catalogue.

Par rapport à la référence en la matière, *Design Patterns*, ce livre ne présente que la moitié de ces mêmes *patterns* - on ne réinvente pas la roue -, la "bonne" moitié peut-être. En tout cas, si tous ces *patterns* avaient été abordés, on se serait retrouvé avec un pavé de 1200 pages, pas facile à lire et à assimiler, ce qui n'est pas le but non plus. Celui-ci est clairement d'aider les gens à commencer à les utiliser. Par la suite, on pourra toujours feuilleter un catalogue.

Le seul vrai inconvénient à mon avis est d'avoir traduit le nom des *patterns*. Ca fait français, OK, mais l'avantage de l'anglais est qu'on peut en discuter avec n'importe qui dans le monde. Comment va-t-on traduire patron de méthode ? Pense-t-on directement à Template Method ? Pas forcément, je pense. Le langage utilisé est le Java, mais ce n'est pas problématique. Si on sait que les pointeurs n'existent pas, que *abstract*, c'est *virtual* et que *implements* est un héritage, on comprend le livre.

IV - Liens annexes

 ***Critique sur la page de livres Conception***

 ***Achat sur Amazon.fr***

 ***Lien vers le site de l'éditeur***

