

# Conception et Programmation orientées objet

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

Date de publication : 07/09/2006

Dernière mise à jour :

Critique de Conception et Programmation orientées objet de *Bertrand Meyer*

- I - Description
- II - Table des matières
- III - Critique : Complet et complexe
- IV - Liens annexes

## I - Description



Écrit dans un style d'une clarté absolue, cet ouvrage réussit le tour de force d'être à la fois un livre de réflexion approfondie sur les concepts objet et un ouvrage pratique, apportant des réponses concrètes aux questions que se posent tous les programmeurs et concepteurs de logiciels: Comment construire des applications logicielles fiables, performantes, faciles à utiliser et à faire évoluer ?

Partant d'une analyse des écueils souvent rencontrés dans les projets de développement, Bertrand Meyer définit une liste de critères de qualité logicielle et en déduit un ensemble de règles et de principes qui conduisent naturellement aux grands concepts objet.

Théorie du typage, gestion de la mémoire, techniques d'héritage, programmation concurrente, objets distribués, persistance et bases de données objet, analyse et conception, réutilisabilité... : tous les aspects du processus de développement logiciel et des technologies objet sont expliqués et discutés dans leurs moindres détails, dans un style brillant, souvent drôle et provocateur.

Une lecture dont on sort intellectuellement subjugué, et techniquement mieux armé pour répondre à ses défis professionnels.

Ce livre s'adresse à tous les architectes logiciel, ingénieurs méthode et programmeurs impliqués dans des projets de développement logiciel et souhaitant approfondir leur compréhension des technologies objet, ainsi qu'aux étudiants en informatique (niveau 2e cycle ou écoles d'ingénieur).

## II - Table des matières

- Les enjeux du génie logiciel
  - Qualité logicielle
  - Critères de l'approche objet
- Les principes sous-jacents de l'approche objet
  - Modularité
  - Réutilisabilité
  - Vers l'approche objet
  - Types de données abstraits
- Technologies objet
  - Classes
  - Objets et références
  - Gestion de la mémoire et ramasse-miettes
  - Généricité
  - Fiabilité logicielle et conception par contrats (assertions, préconditions et postconditions, invariants de classes...)
  - Gestion des exceptions
  - Héritage et polymorphisme
  - Héritage multiple
  - Techniques d'héritage avancées
  - Les problèmes de typage
  - Objets globaux et constantes
- Méthodologie objet
  - Méthodes de génie logiciel
  - Design patterns
  - Héritage : étude de cas
  - Comment déterminer les classes
  - Principes de la conception des classes
  - Bien utiliser l'héritage
  - Règles d'écriture du code
  - Méthodes d'analyse orientée objet
  - Processus de construction logicielle
  - Enseigner l'approche objet et le génie logiciel
- Techniques avancées
  - Programmation concurrente
  - Persistance et bases de données objet
  - Conception d'interfaces graphiques
- Appliquer la méthode sous différents langages et outils de développement
  - Programmation objet et Ada
  - Programmer objet à l'aide de langages non-objet (C, Pascal, Fortran)
  - Simula, Smalltalk, extensions Lisp, C++, Java...

- Un environnement de développement orienté objet : composants, le langage (Eiffel), techniques de compilation, outils et bibliothèques
- Annexes
  - Extraits des bibliothèques de base
  - Généricité versus héritage
  - Principes, règles, préceptes et définitions
  - Glossaire des technologies objet
  - Bibliographie : plus de 400 références de livres, d'articles et de sites Web.

### III - Critique : Complet et complexe

Quand on prend ce livre dans les mains, on s'attend à une lecture studieuse et attentive. Mais en lisant le début, on se rend déjà compte que *Bertrand Meyer* a pensé à nous. Une première partie est consacrée aux bases de l'orienté objet, en gros ce dont il faudra absolument retenir en reposant ce livre.

Réutilisabilité et modularité sont les maître-mots de ce livre, et toutes les techniques actuelles pour y parvenir sont analysées dans les détails. De la généricité au garbage collector, on se rend compte qu'il n'y a aucun langage qui réponde à tous les critères de ce livre. Plusieurs langages sont étudiés pour voir ce qu'il aurait fallu améliorer - le garbage collector en C++ -, c'est très intéressant.

En ce qui concerne la méthodologie objet, le livre est à nouveau complet, comment pourrait-il en être autrement ? Une partie consacrée à des problèmes spécifiques montre que l'auteur sentait sans doute venir l'évolution actuelle, la programmation concurrente étant de plus en plus utilisée depuis les annonces d'Intel et AMD. La dernière partie permettra de voir un peu mieux comment appliquer ce qu'on a appris pendant tout le livre.

Malgré les passages que l'on peut passer en première lecture, le livre est tout de même d'une difficulté certaine. La lecture de chaque chapitre entraîne une réflexion qu'on doit mener pour bien assimiler les concepts. Eu égard à la quantité et la qualité des propos dans ce livre, on peut être décourager par le temps nécessaire pour une lecture en détail complète, d'où l'étoile en moins.

## IV - Liens annexes

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

 ***Achat sur Amazon.fr***

 ***Lien vers le site de l'éditeur***



