

# **Introduction**

## Introduction

[http://fr.wikibooks.org/wiki/Patrons\\_de\\_conception/Introduction](http://fr.wikibooks.org/wiki/Patrons_de_conception/Introduction)

This Book Is Generated By [Wb2PDF](#)

using

[RenderX XEP](#), XML to PDF XSL-FO Formatter

---

## Table of Contents

1. Introduction.....	4
Sections.....	?
Comment lire ce livre ?[modifier].....	4
Pertinence d'utilisation et implémentation[modifier].....	5
Classification[modifier].....	6
Ensemble de patrons de conception[modifier].....	6
Catégorie fonctionnelle[modifier].....	6

# Introduction

### Patrons du « Gang of Four » ▼

Un patron de conception (plus connu sous le terme anglais « *Design pattern* ») est une solution générique permettant de résoudre un problème spécifique.

En général, un patron de conception décrit une structure de classes utilisant des interfaces, et s'applique donc à des développements logiciels utilisant la programmation orientée objet.

Cette notion peut paraître nouvelle, mais il s'agit en fait plutôt d'un nouveau terme pour désigner les algorithmes, et les structures de données permettant de résoudre différents problèmes. Un exemple : une liste chaînée permet d'avoir un groupe d'éléments dont le nombre n'est pas fixe, contrairement aux tableaux.

Cette notion ne s'applique donc pas seulement à la programmation orientée objet. Un autre exemple est l'architecture **MVC** (Modèle-Vue-Contrôleur) définissant une architecture où les fonctions (ou les classes en **POO**) ont un rôle bien défini.

## Comment lire ce livre ?[\[modifier\]](#)

Cette page d'introduction présente des notions générales à tous les patrons de conception. Les pages suivantes répertorient les différents patrons de conception existants. Il est donc recommandé de lire cette page, puis de lire/rechercher les pages selon le but recherché par le lecteur :

- Pour apprendre à programmer, il peut être intéressant de voir les solutions proposées par les patrons de conception, en commençant par les plus simples, ou les plus connus.
- Pour le développeur recherchant une solution à un problème, regarder les différents patrons de la catégorie du problème concerné.
- Pour le développeur cherchant à apprendre un nouveau patron de conception, aller directement à la page concernée.

Chaque page est rédigée de façon à pouvoir être lue de façon indépendante, sans avoir à lire d'autres pages pour comprendre le contenu.

Pour plus d'informations, consulter les ouvrages et les sites web cités dans le chapitre [Bibliographie et liens](#).

Pour chercher un patron de conception à partir de son nom en anglais ou en français, utiliser le moteur de recherche suivant :

## Pertinence d'utilisation et implémentation[[modifier](#)]

Utiliser des patrons de conception pour le développement de logiciels peut paraître compliqué ou superflu. Dans les applications les plus simples, l'utilisation de patrons de conception peut générer une complexité dans le code source.

Cependant, les patrons de conception sont généralement utiles pour les applications ayant une taille importante et/ou dans les projets où plusieurs applications différentes interagissent entre elles (via un moyen de communication).

Il faut également bien comprendre le rôle d'un patron de conception afin de vérifier qu'il s'applique au cas rencontré, et ne pas ajouter inutilement une complexité si les avantages liés à l'utilisation d'un patron de conception ne sont pas requis.

Apprendre par cœur chaque patron de conception est inutile car ils sont issus de la logique de conception. Une solution bien conçue peut convenir aussi bien qu'un patron de conception, et dans beaucoup de cas elle est mieux adaptée à certaines situations.

Les patrons de conception, lorsqu'ils sont détaillés, ne doivent pas forcément être appliqués à la lettre, car ils apportent des solutions génériques. La solution étant générique, lors de l'implémentation d'un patron de conception dans un langage donné (souvent imposé lors de la reprise d'un projet existant), il sera certainement nécessaire de l'adapter à la situation :

- Si une notion utilisée dans le patron de conception n'existe pas (exemple, la notion de classe en langage C), il faudra trouver une notion proche à adapter (exemple, en C, utiliser une structure) ;
- Le contexte d'utilisation de la solution peut poser un nouveau problème à résoudre dans l'implémentation du patron de conception. Par exemple, utilisé dans un contexte multi-threads, le patron de conception "Singleton" devra être implémenté de façon synchronisée.

# Classification[[modifier](#)]

Il y a différentes façons de classer les différents patrons de conception. Ce livre présente les plus connus, classés en fonction de leurs auteurs, puis, si possible, classés par catégorie fonctionnelle.

## Ensemble de patrons de conception[[modifier](#)]

Il y a différents ensembles de patrons de conception, créés par différents auteurs.

- Les plus connus sont ceux du « [Gang of Four](#) » (ou GoF : Erich Gamma, Richard Helm, Ralph Johnson et John Vlissides) décrits dans leur livre « *Design Patterns -- Elements of Reusable Object-Oriented Software* » (voir [bibliographie](#)) en 1995. Les patrons de conception tirent leur origine des travaux de l'architecte Christopher Alexander dans les années 70.
- Les patrons [GRASP](#) sont des patrons créés par Craig Larman qui décrivent des règles pour affecter les responsabilités aux classes d'un programme orienté objets pendant la conception, en liaison avec la méthode de conception BCE (pour « Boundary Control Entity » - en français MVC « Modèle Vue Contrôleur »).
- Les patrons [d'entreprise](#) (*Enterprise Design Pattern*) créés par Martin Fowler, décrivent des solutions à des problèmes courants dans les applications professionnelles. Par exemple, des patrons de couplage entre un modèle objet et une base de donnée relationnelle.
- D'autres patrons créés par divers auteurs existent et décrivent des solutions à des problèmes différents de ceux vus précédemment.

## Catégorie fonctionnelle[[modifier](#)]

Les patrons de conception peuvent être classés en fonction du type de problème qu'ils permettent de résoudre. Par exemple, les patrons de conception de création résolvent les problèmes liés à la création d'objets.