

DESIGN PATTERNS

Introduction

✿ Pourquoi ?

✿ Objectifs de la présentation.

Définition

- ✿ En français - Modèles de conception
et aussi « Motifs de conception », « Patron
de conception ».
- ✿ Solutions à des problèmes
classiques de conception.
- ✿ « L'algorithmie de la conception »

Définition (2)

Un design pattern nomme, motive et explique un design général qui répond à un problème fréquent dans un système orienté objet.

Il décrit le problème, la solution, quand appliquer la solution et ses conséquences.

Il donne aussi des conseils d'implémentation et des exemples.

[GoF]

Définition (3)

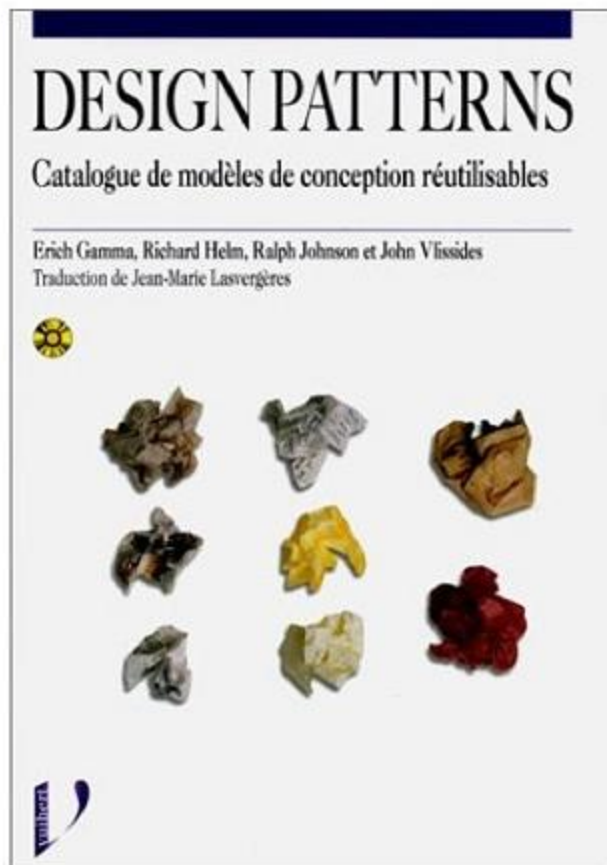
- ⚙ Solutions connues pour résoudre des problèmes connus
- ⚙ Indépendant de tout langage informatique
- ⚙ « Bibliothèque conceptuelle »

Un peu d'histoire (1)

✿ C. Alexander « A Pattern Language: Towns, Buildings, Construction » [1977]

« Les utilisateurs connaissent mieux le bâtiment dont ils ont besoin que les architectes »

Un peu d'histoire (2)



OOPSLA 1987

Kent Beck Ward Cunningham

GoF « Gang of Four »

Design patterns. Elements of reusable Object-Oriented Software [1994]

Erich Gamma, Richard Helm, Ralph Johnson et John Vlissides

Pourquoi les connaître ?

« Chaque patron décrit un problème qui se manifeste constamment dans notre environnement et décrit le cœur de la solution à ce problème, d'une façon telle que l'on puisse réutiliser cette solution des millions de fois, sans jamais le faire deux fois de la même manière »

Christopher Alexander.

Pourquoi les connaître ? (2)

- ✿ Facilite la communication entre développeurs.
- ✿ Décompose un problème complexe en plusieurs problèmes simples.
- ✿ Facilite la conception. (Améliore sa compréhension, auto-documentation des concepts)

Pourquoi les utiliser ?

- ✿ Pour la même raison que l'on utilise une fonction, une classe, un objet, une bibliothèque ou un framework.
- ✿ Ne pas réinventer la roue.
- ✿ Bénéficier du savoir faire d'experts dans des contextes éprouvés. (fiables, robustes & connus)

Fiche d'identité

- ⚙ Nom
- ⚙ Description du problème
- ⚙ Description de la solution
 - (a) **Quels sont les éléments qui participent à la solution ?**
 - (b) **Quels sont leur rôle**
 - (c) **Comment interagissent t ils entre eux ?**
- ⚙ Exemples
- ⚙ Critiques

Les 23 Patterns « GoF »

✿ Créateurs

- (a) **Factory method**
- (b) **Abstract factory**
- (c) **Builder**
- (d) **Prototype**
- (e) **Singleton**

✿ Structuraux

- (a) **Adapter**
- (b) **Bridge**
- (c) **Composite**
- (d) **Decorator**
- (e) **Façade**
- (f) **Flyweight**
- (g) **Proxy**

✿ Comportementaux

- (a) **Interpreter**
- (b) **Template method**
- (c) **Chain of responsibility**
- (d) **Command**
- (e) **Iterator**
- (f) **Mediator**
- (g) **Memento**
- (h) **Observer**
- (i) **State**
- (j) **Strategy**
- (k) **Visitor**