Design Patterns

Design Patterns
Patrons de conception
Expérience en boîte

Architecture logicielle

- L'Art (8e) de l'architecture logicielle est un art difficile car il ne peut se baser sur des loi physiques (quasi)immuables
- Le concept Design Pattern:
  - Des techniques éprouvées pour résoudre des problèmes récurrents
  - Il ne reste *presque* qu'à brancher...

« Dans un contexte, solution éprouvée à un problème récurrent »

Référence ...

- GoF *Gang of Four*
  - Gamma Helm Johnson Vlisside
  - Design patterns : elements of reusable object-oriented software
  - Design patterns : catalogue de modèles de conception réutilisables
  - 005.12 DES à la bibliothèque Copernic (10ex)

1995 !

Design Patterns

- La conception orienté objet est une activité difficile il faut trouver les bons objets les factoriser en classes de la bonne granularité, définir des interfaces et des hiérarchies de classes et surtout établir des relations entre tout cela.
- Écrire des éléments modifiables est difficile, des éléments réutilisables est encore plus difficile.

Design Patterns

- Une conception flexible et réutilisable est rarement obtenue du premier jet.
- Avec l'expérience on tend a produire de la bonne conception, comment capitaliser cette expérience, c'est l'objectif des patrons de conception (Design Patterns).

Qu'est-ce qu'un <patron>

- Patron/Template/Pattern
- Titre : produit un tas de questions intéressantes
- OK ! « l'omelette aux <girolles>, c'est une omelette dans laquelle tu jettes à mi-cuisson des <girolles> sautées»
- NOK ! « j'ai le même <jardin> que <ma belle-mère>, mais <sans> les <nains de jardins> »

Patrons
capitalisation de l'expérience

- Un patron décrit une situation fréquente et une réponse éprouvée à cette situation
- Le patron peut s'utiliser *tel quel* ou être adapté ou combiné avec d'autre pour répondre aux besoins
- Les patrons décrits ici sont ceux fournis par le *Gof* qui sont devenus une référence pour l'ensemble de la communauté des développeurs objet

Qu'est-ce qu'un <patron>

- Template/Pattern
- Titre : produit un tas de questions intéres
- « l'omelette aux <girolles>, c'est
  quelle tu jettes à mi-cuiss
- « j'ai le même <i
  <sans> les

Rappel
on ne fait pas de copier-coller de code

Patrons
capitalisation de l'expérience

- La démarche d'apprentissage des patrons de conception :
  - comprendre leur utilité
  - accepter leur utilité
  - lire et relire les patrons
  - les intérioriser pour les utiliser
  - LES UTILISER ! *PRESQUE* PARTOUT !
    - Attention risque « Overdesign » !

DP : description

- Nom !
- Les problèmes : QUAND l'appliquer
- Solution : description (UML), responsabilités, collaborations entre classes/objets - COMMENT
- Conséquences
  - Résultats
  - Impacts
  - Compromis

3 Types de Design Pattern

- Patrons de création/construction
  - liés au problème de choisir la classe responsable des création, de choisir le type créé
  - permet l'encapsulation des classes effectives
- Patrons structurels
  - liés au problèmes d'organisation des objets dans un logiciel
    - Composition des classes et des objets
- Patrons comportementaux (behavioral)
  - liés au problèmes de communication entre les objets
    - Distribution des responsabilités

Classe / Objet

		Purpose		
		Creational	**Structural**	**Behavioral**
Scope	**Class**	Factory Method (107)	Adapter (139)	Interpreter (243) Template Method (325)
	Object	Abstract Factory (87) Builder (97) Prototype (117) Singleton (127)	Adapter (139) Bridge (151) Composite (163) Decorator (175) Facade (185) Proxy (207)	Chain of Responsibility (223) Command (233) Iterator (257) Mediator (273) Memento (283) Flyweight (195) Observer (293) Stat Strategy (315) Visitor (331)

Portée (Scope) : à quel niveau s'applique un DP ? Au niveau des classes ou des objets ?

Portée

- Les patrons au niveau des classes s'occupent des relations entre les classes et les sous-classes. Ces relations sont fixées *statiquement* à la compilation.
  - Exemple: Les patrons de création au niveau des classes délèguent une partie de la création des objets aux sous-classes.
- Les patrons au niveau des objets s'occupent des relations entre les objets, lesquels sont plus dynamiques et peuvent être modifiées à l'exécution
  - Les patrons de création au niveau des objets délèguent la création à d'autres objets.