



Cours de Génie Logiciel

Sciences-U Lyon

Design Pattern
Trois exemples

<http://www.rzo.free.fr>



Sommaire

- Catalogue de Design Pattern
 - **Création**
 - Structure
 - Comportement



DP de Création

- Design Pattern de création
 - Singleton
 - Usine Abstraite
 - Méthode Usine
 - Constructeur
 - Prototype



DP de Création

- Design Pattern de création
 - Abstraction du processus d'instanciation
 - Indépendance de
 - Création
 - Composition
 - Représentation
 - Des objets
 - Importance de la composition
 - Plus que de l'héritage



DP de Création

- Design Pattern de création
 - Flexibilité
 - Ce qui est créé
 - Qui le crée
 - Comment c'est créé
 - Quand c'est créé



DP de Création

- Constructeur - builder
 - Objectif
 - Séparer dans un objet complexe la construction et les représentations
 - Motivations
 - Exemple : lecture d'un fichier RTF (Rich Text Format)
 - Conversion dans d'autres langages (ASCII, Tex)
 - On se sait pas à l'avance quels sont les formats cibles
 - Besoin d'ajouter un format sans modifier le programme de conversion



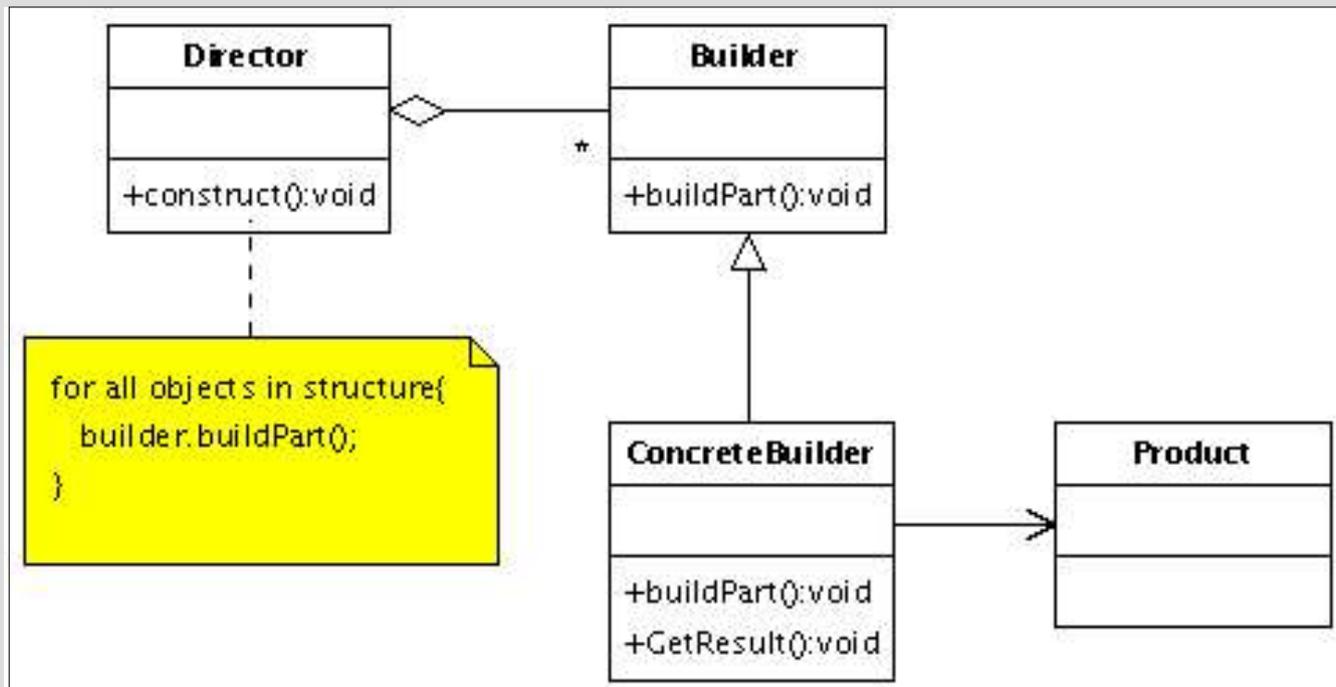
DP de Création

- Constructeur
 - Applicabilité
 - Quand l'algorithme de construction d'un objet complexe est indépendant des parties de l'objets et de la manière dont elles sont assemblées
 - Le processus permet différentes représentations de l'objet construit



DP de Création

- Constructeur
 - Structure





DP de Création

- Constructeur
 - Participants
 - Constructeur (Builder)
 - Interface abstraite pour créer les parties de l'objet créé
 - ConstructeurConcret (ConcreteBuilder)
 - Implémente l'interface Constructeur
 - Construit et assemble les parties de l'objet
 - Définit et enregistre les objets créés
 - Interface de récupération du produit (ex.: GetObject())



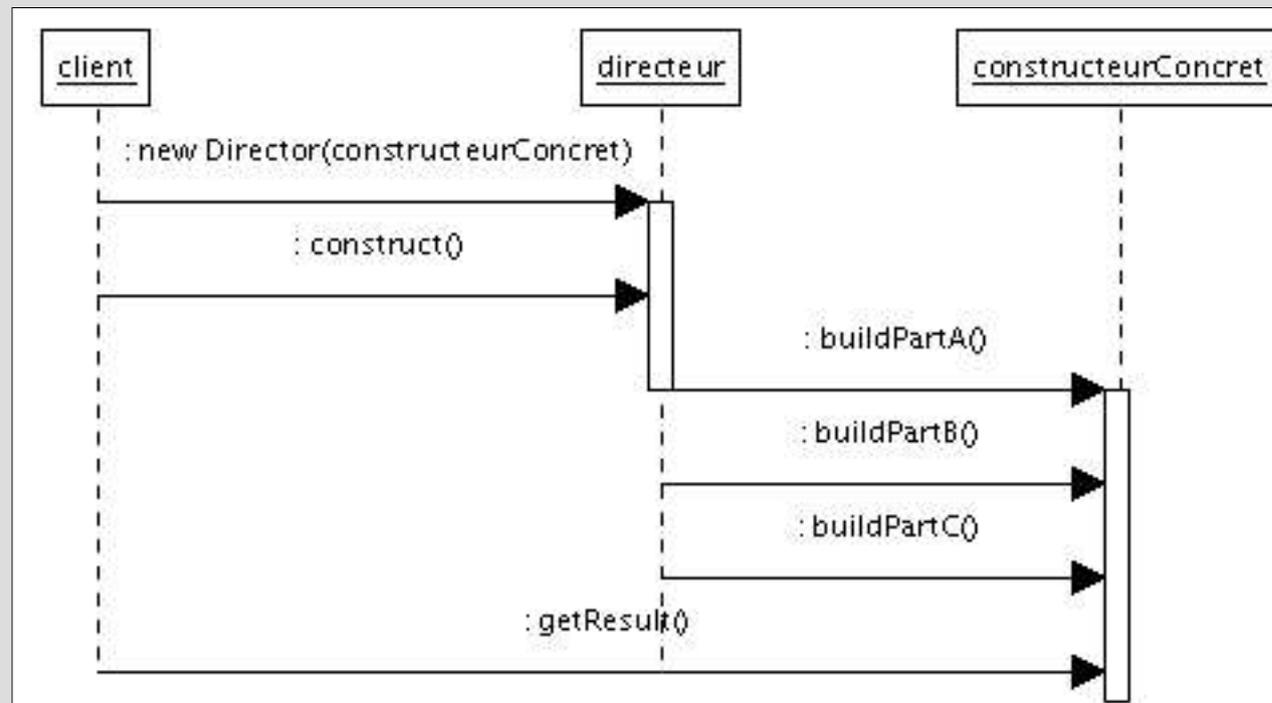
DP de Création

- Constructeur
 - Participants
 - Director
 - Contruit un objet par l'interface Constructeur
 - Product
 - Objet en construction
 - Contient des classes définissant les parties du produits
 - Interface d'assemblage



DP de Création

- Constructeur
 - Collaboration





DP de Création

- Constructeur
 - Conséquences
 - Variations possible de la représentation interne du Produit
 - Isolation du code de construction et de présentation
 - Contrôle fin du processus de construction
 - Code
 - Non représenté ici



DP de Création

- Constructeur
 - Utilisations connues
 - Parsers XML
 - Patterns liés
 - Usine abstraite
 - Produits simples ou complexes
 - Familles de produits
 - Composite
 - Format des objets créés par le Constructeur



DP de Création

- Bilan
 - Construction de classes
 - Héritage
 - Composition



Sommaire

- Catalogue de Design Pattern
 - Création
 - **Structure**
 - Comportement



DP de structure

- Design Pattern de Structure
 - Adaptateur
 - Facade
 - Proxy
 - Pont
 - Composite
 - Décorateur
 - Poids Mouches



DP de structure

- Design Pattern de Structure
 - Expression de la composition
 - Des classes
 - Des objets
 - En structure complexe
 - Patterns de Classes
 - Statiques, définis à la conception
 - Patterns d'objets
 - Dynamiques
 - Nouvelles fonctionnalités à l'exécution



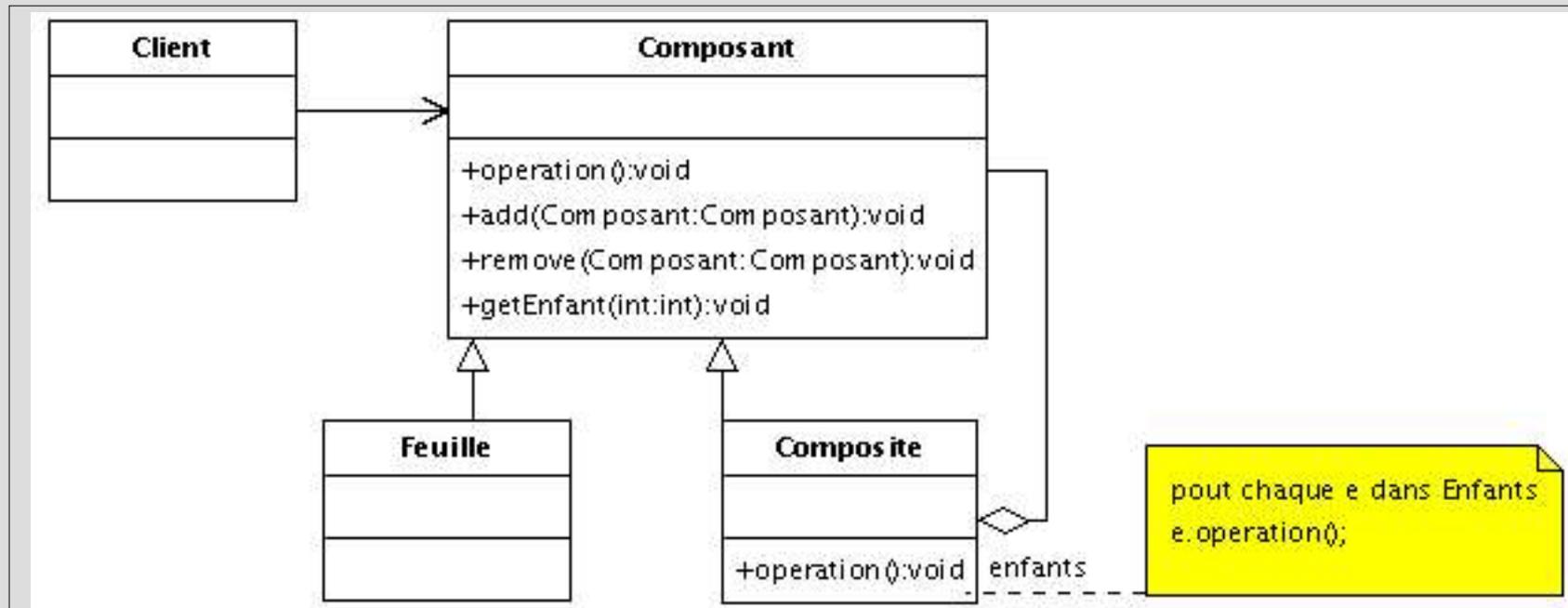
DP de structure

- Composite : Pattern d'objet
 - Objectif
 - Composition d'objets en structure d'arbre pour représenter une hiérarchie Tout-Partie
 - Traitements indifférents des objets individuels et composites par les clients
 - Motivation
 - Interfaces graphiques : structure complexe
 - Fenêtres, comprenant des sous-fenêtres, comprenant des boutons ...
 - Besoin
 - d'une structure récursive
 - Représentation identique des conteneurs et des primitives



DP de structure

- Composite : Pattern d'objet
 - Structure





DP de structure

- Composite : Pattern d'objet
 - Applicabilité
 - Représentation d'une hiérarchie d'objets Tout-Partie
 - Client ignore différences entre objets composés et objets individuels



DP de structure

- Composite : Pattern d'objet
 - Participants
 - Composant
 - Interface pour les objets de la composition
 - Interface d'accès et de gestion des composants enfants
 - Implémentation d'un comportement par défaut pour cette interface
 - (optionel) définit une interface pour accéder au composant parent
 - Feuille
 - Objets feuilles de la composition – une feuille n'a pas d'enfant
 - Définit le comportement pour les primitives (ex : bouton)



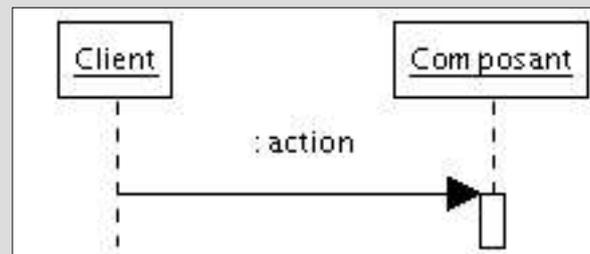
DP de structure

- Composite : Pattern d'objet
 - Participants
 - Composite
 - Définit le comportement des composants avec enfants
 - Stocke les composants enfants
 - Implémente les opérations concernant les enfants
 - Client
 - Manipule les objet par l'interface Composant



DP de structure

- Composite
 - Collaboration
 - Composant manipulé est une feuille
 - l'opération est exécutée directement
 - Composant manipulé est un composite
 - Opération généralement transmise aux enfants
 - Avec ou sans pré-post traitement





DP de structure

- Composite
 - Conséquences
 - Composants définis récursivement
 - Le client ne sais pas quel est la profondeur de la récursion
 - Simplicité du client
 - Pas de connaissance de l'implémentation
 - Ajout facile de nouveaux type de composants
 - Pas de modification du code du client
 - Conception générale
 - Pas de contrôle sur les types de composants
 - Si besoin, le contrôle doit être codé en dur



DP de structure

- Composite
 - Usages connus
 - Langages d'interfaces graphiques (ex : Swing)
 - Parser (ex : DOM pour manipulation de documents XML)
 - Portefeuilles financiers, comprenant différents types d'actions
 - Composition de Commandes (voir le Design Pattern correspondant)



DP de structure

- Composite
 - Pattern liés
 - Chaine de responsabilité
 - Lien composant-parent
 - Decorateur
 - Utilisation conjointe (Classe Composant commune)
 - Poids Mouche
 - Partage de composants
 - Iterator
 - Pour traverser les composites
 - Visiteur
 - Localise les comportements qui seraient sinon distribués dans les composants



DP de structure

- Bilan
 - Combinaison possible de différents Patterns



Sommaire

- Catalogue de Design Pattern
 - Création
 - Structure
 - **Comportement**



DP de comportement

- Design Pattern de Comportement
 - Observateur
 - Itérateur
 - Stratégie
 - Méthode Patron
 - Memento
 - Chaine de responsabilité



DP de comportement

- Design Pattern de Comportement
 - (non détaillés ici)
 - Commande
 - Interpréteur
 - Médiateur
 - Etat
 - Visiteur



DP de comportement

- Design Pattern de Comportement
 - Objectifs
 - Algorithmes
 - Distribution de responsabilité entre objets
 - Patterns de communications
 - Entre classes, entre objets
 - Masquent les flux de contrôle complexe
 - Concentration sur l'interconnection des objets



DP de comportement

- Design Pattern de Comportement
 - Rôle de l'héritage
 - Distribution de comportement entre les classes
 - Méthode Patron, Interpreteur
 - Composition
 - Couplage faible
 - Médiateur, Chaîne de responsabilité
 - Contrôle de la dépendance entre objets
 - Observateur



DP de comportement

- Design Pattern de Comportement
 - Encapsulation du comportement
 - Délégation des opérations à d'autres objets
 - Stratégie, Commande, Etat, Visiteur, Itérateur



DP de comportement

- Itérateur - Curseur
 - Objectif
 - Accès séquentiel aux éléments d'un objet agrégé
 - Sans exposer la structure de cet objet
 - Motivations
 - Parcours de liste, sans prendre en compte son implémentation
 - Parcours multiples simultanés d'une liste



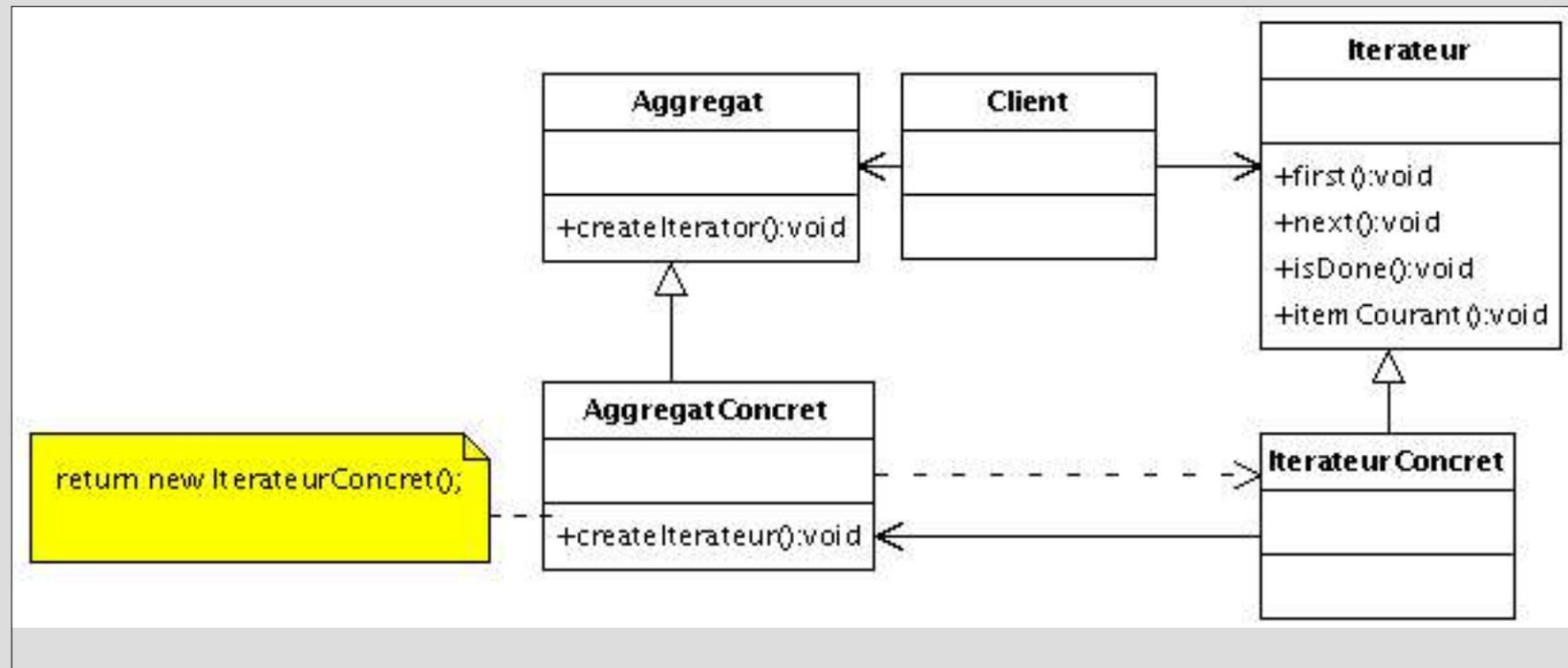
DP de comportement

- Itérateur
 - Applicabilité
 - Accès au contenu d'un objet sans exposer sa représentation interne
 - Support de traversées multiples d'objets agrégés
 - Interface uniforme indépendamment des types de structures agrégées
 - = itération polymorphique
 - Hashtable, List, Set



DP de comportement

- Itérateur
 - Structure





DP de comportement

- Itérateur
 - Participants
 - Itérateur
 - Interface pour accéder et traverser les éléments
 - ItérateurConcret
 - Implémentation de l'interface Itérateur
 - Conserve la position courante lors de la traversée de l'agrégat
 - Agrégat
 - Interface pour créer un objet Itérateur
 - AgrégatConcret
 - Implémentation de l'interface Agrégat pour retourner une instance de ItérateurConcret



DP de comportement

- Itérateur
 - Collaboration
 - ItérateurConcret conserve la position courante dans l'agrégat
 - Il peut retourner l'objet suivant de cet agrégat
 - Conséquences
 - Support de variations de la traversée de l'agrégat
 - Simplification de l'interface Agrégat
 - Plusieurs traversées possibles sur un même agrégat



DP de comportement

- Itérateur
 - Usages connus
 - Très connu
 - Classe Iterator (Java)
 - Pour différents types d'ensembles : List, Set, Hastable, etc.
 - Patterns Liés
 - Composite
 - Itérateur parcourt le composite
 - Méthode Usine
 - Mise en oeuvre pour les itérateurs polymorphiques
 - Memento



DP de comportement

- Bilan
 - Encapsulation des variations
 - Partie changeant fréquemment = 1 objet
 - Accès par le client : invariante
 - Stratégie, état, médiateur, itérateur
 - Objets comme arguments
 - Fonctionnalités
 - Visiteur
 - Données
 - Commande – requête
 - Memento – état de l'objet



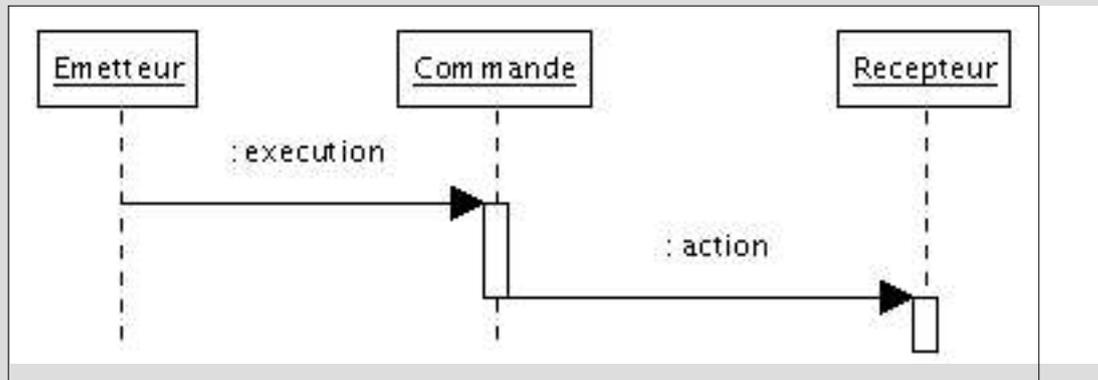
DP de comportement

- Bilan
 - Encapsulation ou distribution communication
 - Encapsulation
 - Médiateur
 - Centralisation des communications
 - Distribution
 - Observateur
 - Plusieurs observateurs (=plusieurs contraintes) par objet



DP de comportement

- Bilan
 - Découplage émission-réception
 - Ex : Commande
 - Autres : Observateur, Médiateur, Chaîne de responsabilité





DP de comportement

- Bilan
 - Patterns complémentaires entre eux
 - Ex : Chaîne de responsabilité utilise Commande
 - pour représenter les requêtes
 - Patterns complémentaires des autres Patterns
 - Ex : Visiteur parcourt un Composite



Design Pattern

- Qu'attendre des Design Patterns ?
 - Vocabulaire commun de conception
 - Et moyen d'utiliser ce vocabulaire
 - Documentation et aide à l'apprentissage
 - Pas besoin de décrire l'implémentation : le nom du Pattern suffit
 - Utile avec ou sans outils associés
 - Ajout aux méthodes existantes
 - Passage de modèle d'analyse à modèle d'implémentation
 - 'Refactoring'
 - Reconception



Design Pattern

Trois exemples

- Bilan
 - Création
 - Structure
 - Comportement

