Cours de Génie Logiciel

Sciences-U Lyon

Design Pattern

http://www.rzo.free.fr

Sommaire

- Les Design Pattern
  - **Introduction**
  - Concepts Objets
  - Réutilisation
  - Utilisation des Design Patterns
  - Catalogue de Design Pattern

Introduction

- Les Design Pattern
  - Le Gang of Four 'GoF'
    - Erich Gamma
    - Richard Helm
    - Ralph Johnson
    - John Vlissides
  - 'Element of reusable Object-Oriented Software'
  - D'autres nomenclatures existent
    - DP pour des domaines précis (ex : interaction)

Introduction

- Les Design Pattern

'Chaque Pattern décrit un problème qui apparait et réaparait dans notre environnement, et ensuite décrit le coeur de la solution à ce problème, de telle manière que l'on peut utiliser cette solution plus d'un million de fois, sans jamais le faire de la même manière'

Christopher Alexander.

Introduction

- Les Design Pattern
  - Trois types Principaux
    - Issus de la nomenclature 'GOF' (Gang of Four)
    - Design Patterns de Création
    - Design Patterns de Structure
    - Design Patterns de Comportement

Introduction

- Trois Familles principales
  - Design Patterns de création
    - *Comment créer de nouveaux objets*
    - 'Usine Abstraite' (Abstract Factory), Gof 87
    - 'Constructeur' (Builder), Gof 97
    - 'Méthode d'usine' (Factory Method), Gof 107
    - Prototype, Gof 117
    - Singleton, Gof 127

Introduction

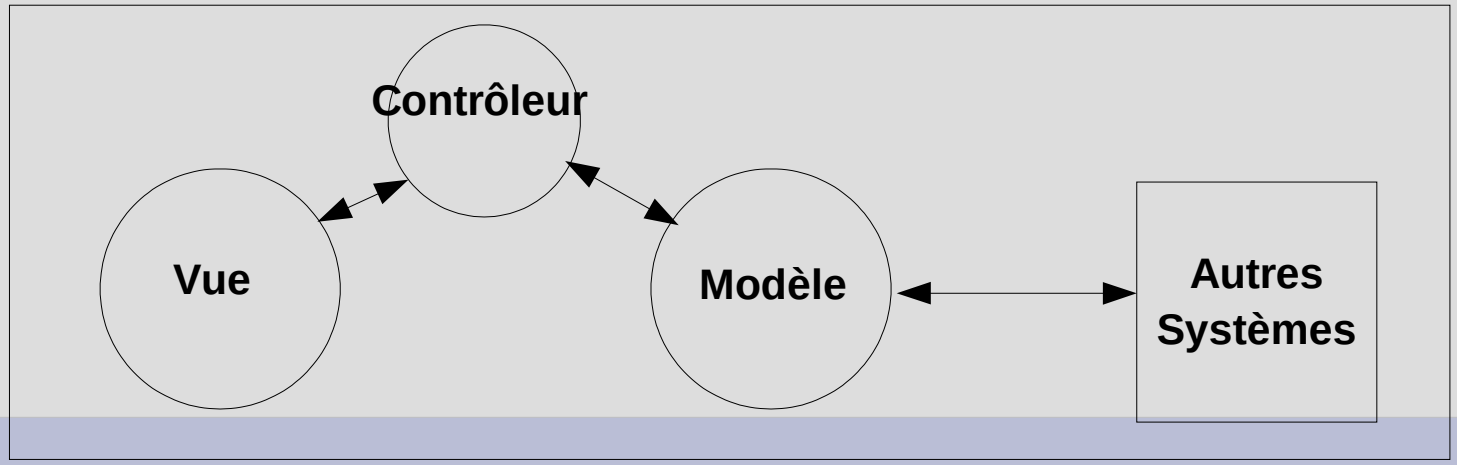
- Trois Familles principales
  - Design Patterns de Structure
    - *Comment organiser les objets*
    - 'Adapteur' (Adapter), Gof 139
    - 'Pont' (Bridge), Gof 151
    - 'Composite', Gof 163
    - 'Décorateur' (Decorator), Gof 175
    - 'Facade', Gof 185
    - 'Poids mouche' (Flyweight), Gof 195
    - 'Proxy', Gof 207

Introduction

- Trois Familles principales
  - Design Patterns de Comportement
    - 'Chaine de responsabilité' (Chain of responsability), Gof 223
    - 'Commande', Gof 233
    - 'Interpreteur', Gof 243
    - 'Iterateur', Gof 257
    - 'Médiateur', Gof 273
    - 'Memento', Gof 283
    - 'Observateur', Gof 293
    - 'Etat' (State), Gof 305
    - 'Stratégie' (Strategy), Gof 315
    - 'Méthode Patron' (Template Method), Gof 325
    - 'Visiteur' (Visitor), Gof 331

Introduction

- Un autre Design Pattern
  - MVC – Modèle/Vue/Contrôleur
    - Par exemple avec Java-Swing
    - Vue = Interface graphique
    - Modèle = traitement (réaction aux évènements)
    - Généralisé par l''Observateur'



Introduction

- Autres Familles de Design Patterns
  - Interactions
  - Concurence de données
  - Programmation distribuée
  - Programmation temps-réel
  - Interfaces utilisateurs
  - Pilotes matériels
  - Accès à une base de données orientée objets

Sommaire

- Les Design Pattern
  - Introduction
  - **Concepts Objets**
  - Réutilisation
  - Utilisation des Design Patterns
  - Catalogue de Design Pattern

Concepts Objets

- Rappel : le paradigme Objet
  - Objet
    - Données
    - Méthodes, opérations
      - Procédure de traitement de ces données
    - Requêtes, messages
      - Pour accéder aux méthodes
  - Encapsulation
    - Accès aux données par le biais des méthodes seulement
      - Etat interne non accessible de l'extérieur

Concepts Objets

- Granularité des objets
  - Taille des objets variable
    - Tableau (Array)
    - Contrôleur d'une Machine à voter
    - Application entière
  - Nombre des objets variable
    - Singleton
    - Livres d'une bibliothèque

Concepts Objets

- Spécification de l'interface d'un objet
  - Ensemble des signatures de méthodes
    - Nom de la méthode
    - Paramètres
    - Type de retour
  - Définit un type
    - Ex : Objets de type Window, qui acceptent toutes les requêtes définies par l'interface 'Window'
    - Un objet peut être de plusieurs types
      - S'il implémente plusieurs interfaces
    - Des Objets différents peuvent être de même type
    - Ex : Base de Donnée et Serveur Web (interface Serveur)

Concepts Objets

- Implémentation d'Objets
  - Création
    - Déclaration
      - Boulangerie monFournil;
    - Instanciation
      - new Boulagerie()
    - Initialisation
      - monFournil = new Boulagerie()

Concepts Objets

- Rôle des interfaces
  - Ensemble de méthodes
    - signatures
  - Définition d'un comportement
    - Spécialisation de ce comportement par héritage d'interfaces
  - Transparence de l'implémentation
    - L'utilisateur manipule les interfaces, pas les classes elles-même

Programmez en fonction des interfaces,
pas des implémentations

Concepts Objets

- Difficulté de la programmation objet
  - Flexibilité
  - Réutilisabilité
  - Des logiciels

Concepts Objets

- Héritage et composition
  - Deux techniques de réutilisation
  - Héritage
    - Accès, modification des fonctionnalités existantes
    - Réutilisation en 'Boite Blanche'
  - Composition
    - Assemblage d'objets existants
    - Réutilisation en 'Boite Noire'

Concepts Objets

- Héritage
  - Avantages
    - Définition statique à la compilation
    - Usage simple, intégré au langage
    - Modifications aisées
  - Inconvénients
    - Pas de modification à l'éxécution
    - Viol de l'encapsulation
    - Pas de modification du parent sans influer sur l'enfant
    - → Peu flexible, donc peu réutilisable

Concepts Objets

- Composition
  - Avantages
    - Définition à l'execution
      - Par acquisition de références sur d'autres objets
    - Respect de l'encapsulation
      - Manipulation des interfaces, et non des objets

Favorisez la composition d'objets
sur l'héritage de classes

Concepts Objets

- Aggrégation et association
  - Aggrégation
    - Des objets *font parties* d'un autre
    - La durée de vie des objets aggrégés est égale (au plus) à celle de l'objet aggrégat
  - Association
    - Connaissance d'un objet par un autre
      - Accès possible
    - Indépendance des durées de vies

Sommaire

- Les Design Pattern
  - Introduction
  - Concepts Objets
  - **Réutilisation**
  - Utilisation des Design Patterns
  - Catalogue de Design Pattern

Réutilisation

- Conception pour la réutilisation
  - Anticiper
    - Les besoins futurs
    - L'évolution des besoins actuels
  - Le système doit pouvoir évoluer
    - Sinon, reconception globale nécessaire
  - Design Patterns
    - Evolution indépendante des différents acteurs
    - Permet l'évolution de l'ensemble du système

Réutilisation

- Causes de difficultés d'évolutions
  - Création d'objets en spécifiant le nom de leur classe
    - Solution : créer les objets indirectement, et manipuler leur interface
    - Usine abstraite, méthode d'usine, prototype
  - Dépendance à certaines opérations
    - Par appel explicite de méthode
    - Solutions : Chaine de responsabilité, Commande
  - Dépendance hardware/software
    - Dépendance à la plate-forme
    - Solutions : Usine abstraite, Pont

Réutilisation

- Causes de difficultés d'évolutions
  - Dépendance à la représentation des objets
    - Stockage, implémentation, location connue du client
    - Solution : cacher ces informations au client
    - Usine Abstraite, Pont, Memento, Proxy
  - Dépendance algorithmique
    - Besoin d'extension, optimisation, remplacement des algorrithmes
    - Solution : Constructeur, Iterateur, Stratégie, Méthode Patron, Visiteur

Réutilisation

- Causes de difficultés d'évolutions
  - Couplage étroit entre classes
    - Solution : Couplage souple, augmente la probabilité de réutiliser les classes
    - Usine abstraite, Pont, Chaine de responsabilité, Commande, Facade, Mediateur, Observateur
  - Extension des fonctionnalités par sous-classes
    - Solution : usage d'une sous-classe, extension par la composition
    - Pont, Chaine de responsabilité, Composite, Décorateur, Observateur, Stratégie
  - Difficulté de modification de classes existantes
    - Solution : Adaptateur, Décorateur, Visiteur

Réutilisation

- Environnements de réutilisation
  - Applications
    - Réutilisation interne
  - Boites à outils
    - = Toolkit
    - Ex : librairie C
    - Pour des applications qu'on ne connait pas
    - Indépendance maximale au contexte
  - Framework
    - Réutilisation de conception
    - Documenté par les Design Pattern utilisés

Sommaire

- Les Design Pattern
  - Introduction
  - Concepts Objets
  - Réutilisation
  - **Utilisation des Design Patterns**
  - Catalogue de Design Pattern

Utilisation

- Sélection d'un Design Pattern
  - Types de DP
    - Création, structure, comportement
  - Objectifs des DP
  - Correspondance entre DP et Problème
  - Relations entre les DP
  - Cause de re-conception
  - Besoin de variabilité

Utilisation

- Usage d'un Design Pattern
  - Vision globale du pattern
    - Applicabilité et conséquences sont à prendre en compte
  - Etude détaillée
    - Structure, Participants, Collaboration
  - Analyse de l'exemple
    - Code de réalisation

Utilisation

- Usage d'un Design Pattern
  - Choix des noms des Participants
    - Pertinents dans le contexte de votre application
  - Définition des noms d'opérations
    - Spécifiques à l'application
  - Implémentation des opérations

Utilisation

- A ne pas faire
  - Usage non justifié
    - Flexibilité, variabilité
    - Mais aussi complication, coût en performance
  - Seulement quand on a besoin de la flexibilité apportée

Sommaire

- Les Design Pattern
  - Introduction
  - Concepts Objets
  - Utilisation des Design Patterns
  - **Catalogue de Design Pattern**
    - **Création**
    - Structure
    - Comportement

Catalogue
DP de Création

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

Catalogue
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

Catalogue
DP de Création

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

Catalogue
DP de Création

- Singleton
  - Nom
  - Structure
  - Applicabilité
  - Conséquences
  - Participants
  - Collaboration
  - Code

Catalogue
DP de Création

- Usine Abstraite
  - Nom
  - Structure
  - Applicabilité
  - Conséquences
  - Participants
  - Collaboration
  - Code

Catalogue
DP de Création

- Méthode Usine
  - Nom
  - Structure
  - Applicabilité
  - Conséquences
  - Participants
  - Collaboration
  - Code

Catalogue
DP de Création

- Constructeur
  - Nom
  - Structure
  - Applicabilité
  - Conséquences
  - Participants
  - Collaboration
  - Code

Catalogue
DP de Création

- Prototype
  - Nom
  - Structure
  - Applicabilité
  - Conséquences
  - Participants
  - Collaboration
  - Code

Sommaire

- Les Design Pattern
  - Introduction
  - Concepts Objets
  - Utilisation des Design Patterns
  - **Catalogue de Design Pattern**
    - Création
    - **Structure**
    - Comportement

Catalogue
DP de structure

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

Catalogue
DP de structure

- Adaptateur
  - Nom
  - Structure
  - Applicabilité
  - Conséquences
  - Participants
  - Collaboration
  - Code

Catalogue
DP de structure

- Facade
  - Nom
  - Structure
  - Applicabilité
  - Conséquences
  - Participants
  - Collaboration
  - Code

Catalogue
DP de structure

- Proxy
  - Nom
  - Structure
  - Applicabilité
  - Conséquences
  - Participants
  - Collaboration
  - Code

Catalogue
DP de structure

- Pont
  - Nom
  - Structure
  - Applicabilité
  - Conséquences
  - Participants
  - Collaboration
  - Code

Catalogue
DP de structure

- Composite
  - Nom
  - Structure
  - Applicabilité
  - Conséquences
  - Participants
  - Collaboration
  - Code

Catalogue
DP de structure

- Décorateur
  - Nom
  - Structure
  - Applicabilité
  - Conséquences
  - Participants
  - Collaboration
  - Code

Catalogue
DP de structure

- Poids Mouche
  - Nom
  - Structure
  - Applicabilité
  - Conséquences
  - Participants
  - Collaboration
  - Code

Sommaire

- Les Design Pattern
  - Introduction
  - Concepts Objets
  - Utilisation des Design Patterns
  - **Catalogue de Design Pattern**
    - Création
    - Structure
    - **Comportement**

Catalogue
DP de comportement

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

Catalogue
DP de comportement

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

Catalogue
DP de comportement

- Observateur
  - Nom
  - Structure
  - Applicabilité
  - Conséquences
  - Participants
  - Collaboration
  - Code

Catalogue
DP de comportement

- Itérateur
  - Nom
  - Structure
  - Applicabilité
  - Conséquences
  - Participants
  - Collaboration
  - Code

Catalogue
DP de comportement

- Stratégie
  - Nom
  - Structure
  - Applicabilité
  - Conséquences
  - Participants
  - Collaboration
  - Code

Catalogue
DP de comportement

- Méthode Patron
  - Nom
  - Structure
  - Applicabilité
  - Conséquences
  - Participants
  - Collaboration
  - Code

Catalogue
DP de comportement

- Memento
  - Nom
  - Structure
  - Applicabilité
  - Conséquences
  - Participants
  - Collaboration
  - Code

Catalogue
DP de comportement

- Chaine de responsabilité
  - Nom
  - Structure
  - Applicabilité
  - Conséquences
  - Participants
  - Collaboration
  - Code

Design Pattern Introduction

- Bilan