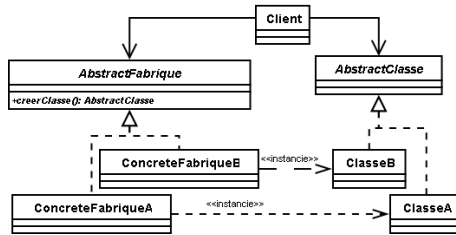


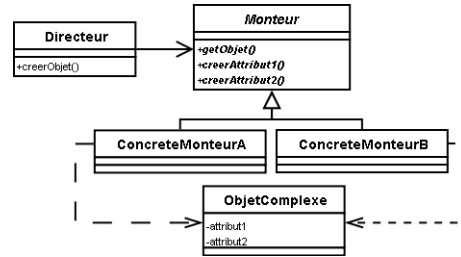
DESIGN PATTERNS DU « GANG OF FOUR »

Fabrique abstraite



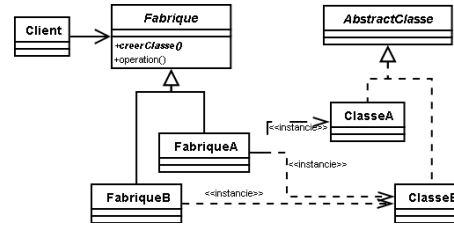
Fournir une interface pour créer des objets d'une même famille sans préciser leurs classes concrètes.

Monteur



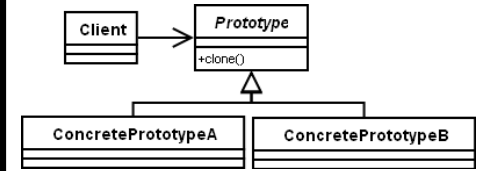
Séparer la construction d'un objet complexe de sa représentation. Permettre d'obtenir des représentations différentes avec le même procédé de construction.

Fabrique



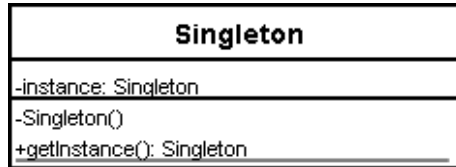
Définir une interface pour la création d'un objet, mais laisser les sous-classes décider quelle classe instancier. Déléguer l'instanciation aux sous-classes.

Prototype



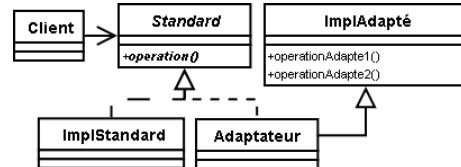
Spécifier les genres d'objets à créer en utilisant une instance comme prototype. Créer un nouvel objet en copiant ce prototype.

Singleton



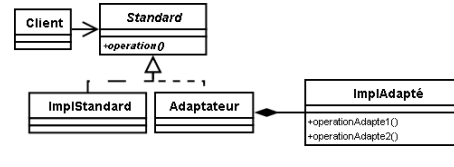
Restreindre le nombre d'instance d'une classe à une et une seule. Fournir une méthode pour accéder à cette instance unique.

Adaptateur (héritage)



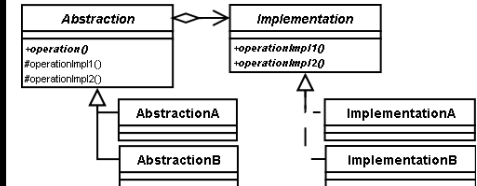
Convertir l'interface d'une classe dans une autre interface comprise par la partie cliente. Permettre à des classes de fonctionner ensemble, ce qui n'aurait pas été possible sinon.

Adaptateur (composition)



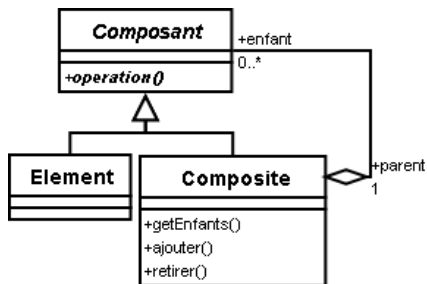
Convertir l'interface d'une classe dans une autre interface comprise par la partie cliente. Permettre à des classes de fonctionner ensemble, ce qui n'aurait pas été possible sinon.

Pont



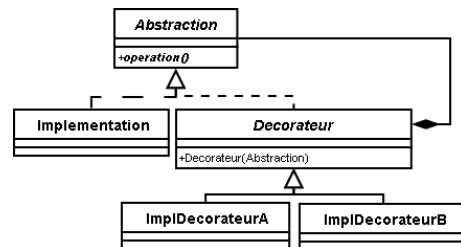
Découpler l'abstraction d'un concept de son implémentation. Permettre à l'abstraction et l'implémentation de varier indépendamment.

Composite



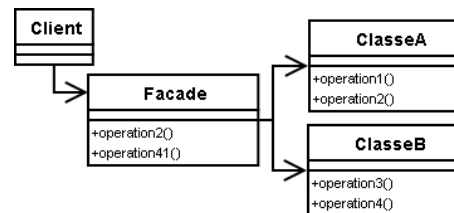
Organiser les objets en structure arborescente. Permettre à la partie cliente de manipuler un objet unique et un objet composé de la même manière.

Décorateur



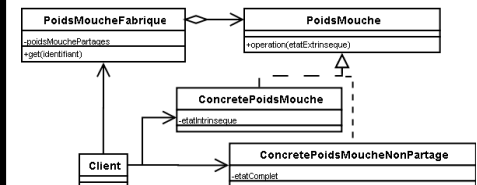
Ajouter dynamiquement des responsabilités (non obligatoires) à un objet. Éviter de sous-classer la classe pour rajouter ces responsabilités.

Façade



Fournir une interface unique en remplacement d'un ensemble d'interface d'un sous-système. Définir une interface de haut-niveau pour rendre le sous-système plus simple d'utilisation.

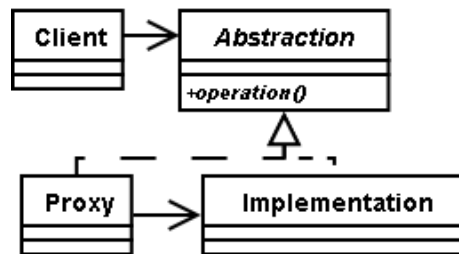
Adaptateur



Utiliser le partage pour gérer efficacement un grand nombre d'objets de faible granularité.

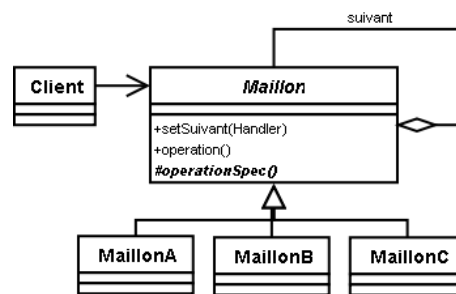
DESIGN PATTERNS DU « GANG OF FOUR »

Proxy



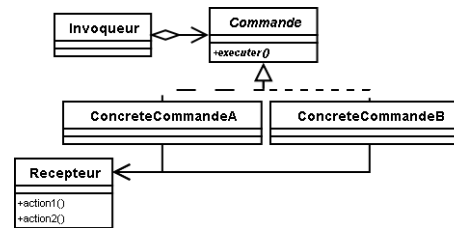
Fournir un intermédiaire entre la partie cliente et un objet pour contrôler les accès à ce dernier.

Chaîne de responsabilité



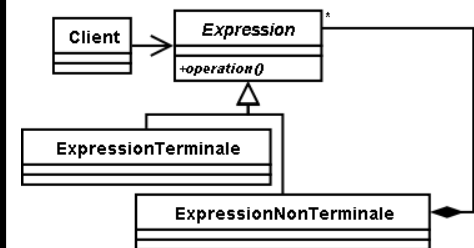
Eviter le couplage entre l'émetteur d'une requête et son receveur.
Chainer les objets receveurs et passer la requête tout le long de la chaîne.

Commande



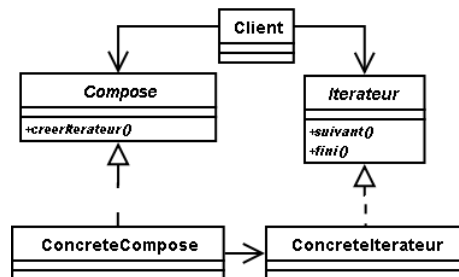
Encapsuler une requête sous la forme d'objet.
Paramétrer facilement des requêtes diverses.
Permettre des opérations reversibles.

Interpréteur



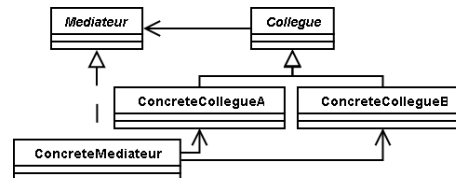
Définir une représentation de la grammaire d'un langage.
Utiliser cette représentation pour interpréter les éléments de ce langage.

Itérateur



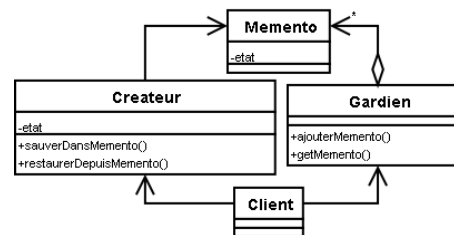
Fournir un moyen de parcourir séquentiellement les éléments d'un objet composé.

Médiateur



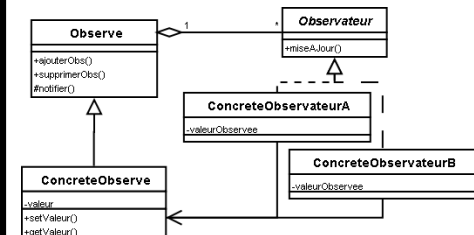
Gérer la transmission d'infos entre des objets.
Avoir un couplage faible entre les objets.
Pouvoir varier leur interaction indépendamment.

Memento



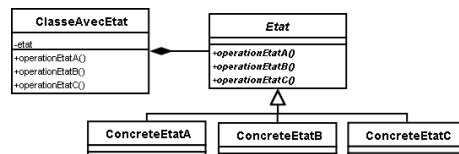
Sauvegarder l'état interne d'un objet en respectant l'encapsulation, afin de le restaurer plus tard.

Observateur



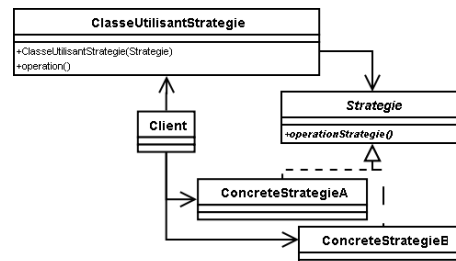
Prévenir des objets observateurs, enregistrés auprès d'un objet observé, d'un événement.

Etat



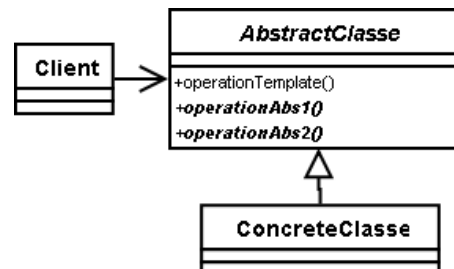
Changer le comportement d'un objet selon son état interne

Stratégie



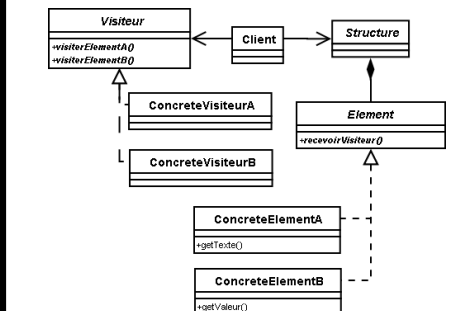
Définir une famille d'algorithmes interchangeables.
Permettre de les changer indépendamment de la partie cliente.

Patron de méthode



Définir le squelette d'un algorithme, en déléguant certaines étapes à des sous-classes.

Visiteur



Séparer un algorithme d'une structure de données.