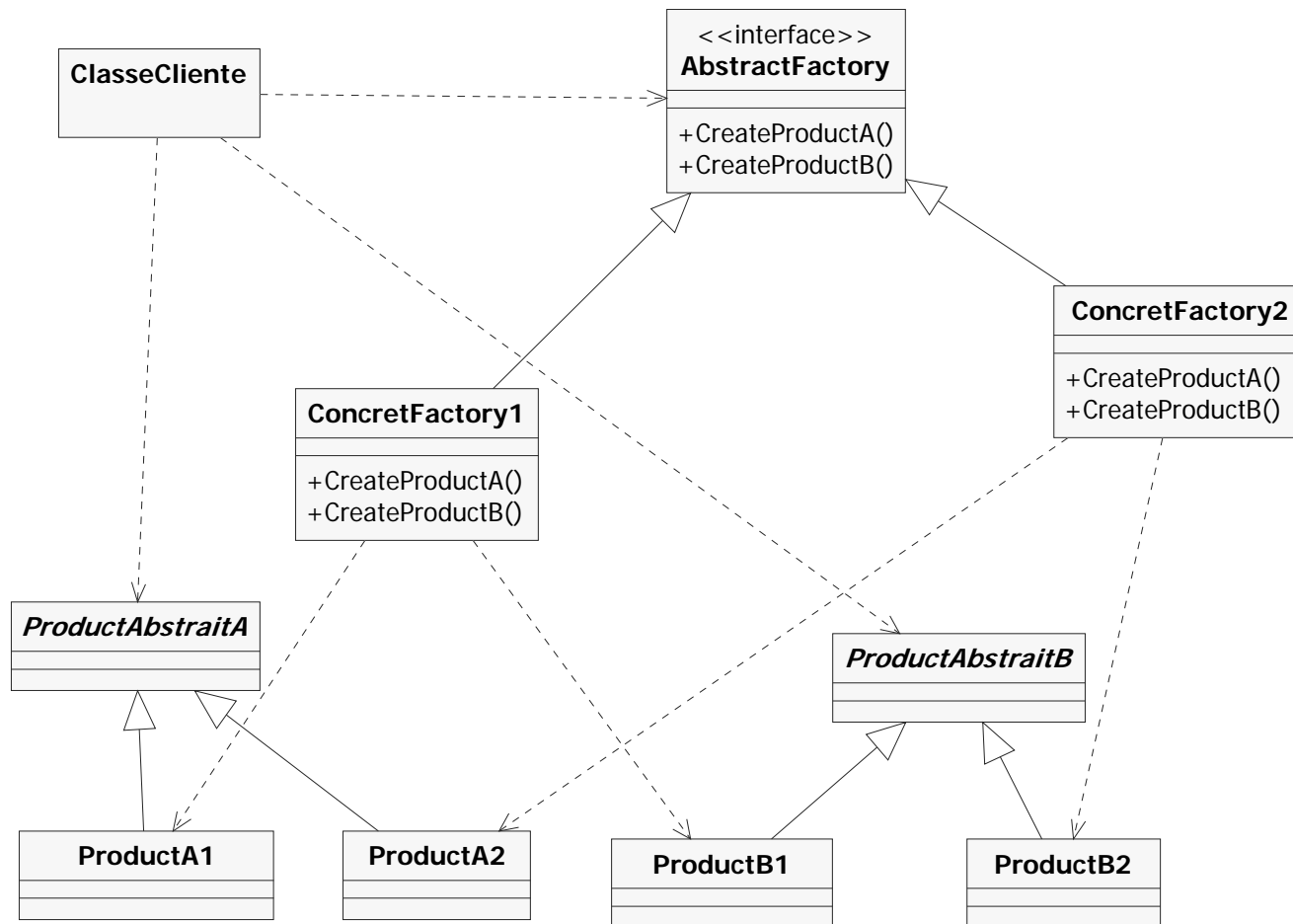
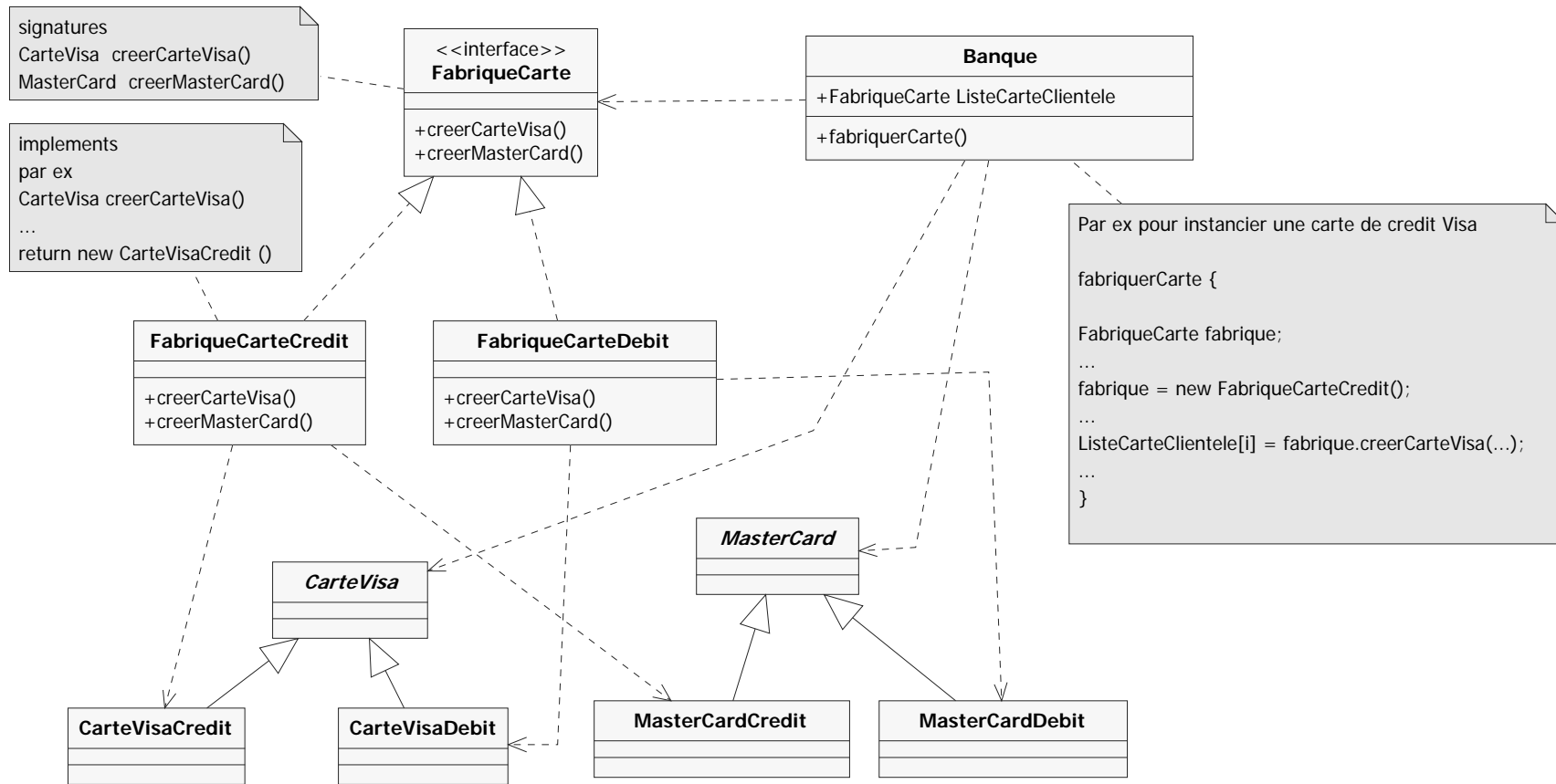


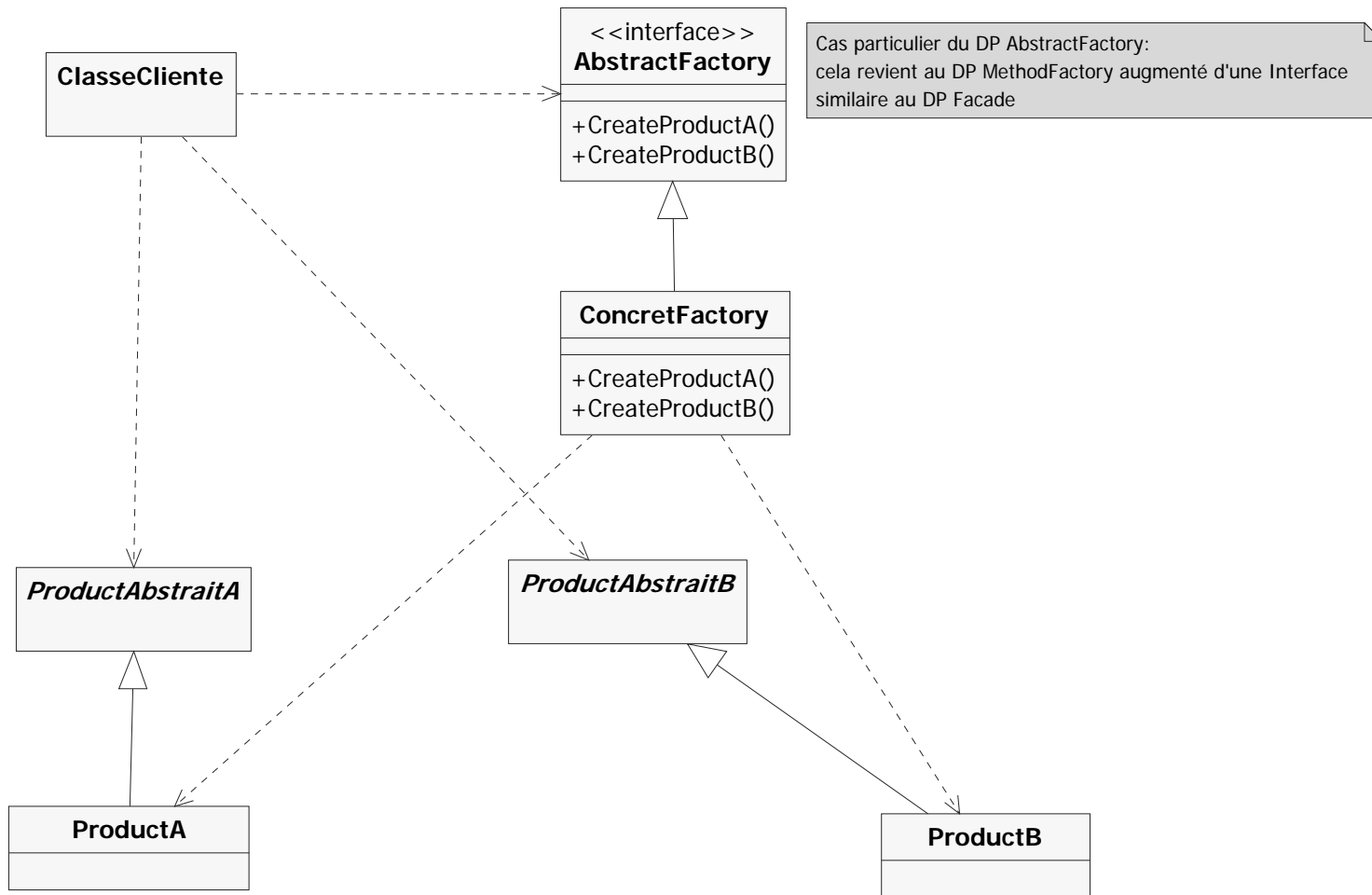
Les DP sont classés selon le GOF en 3 catégories :

- 1) Patterns de construction
- 2) Patterns de structuration
- 3) Patterns de comportement

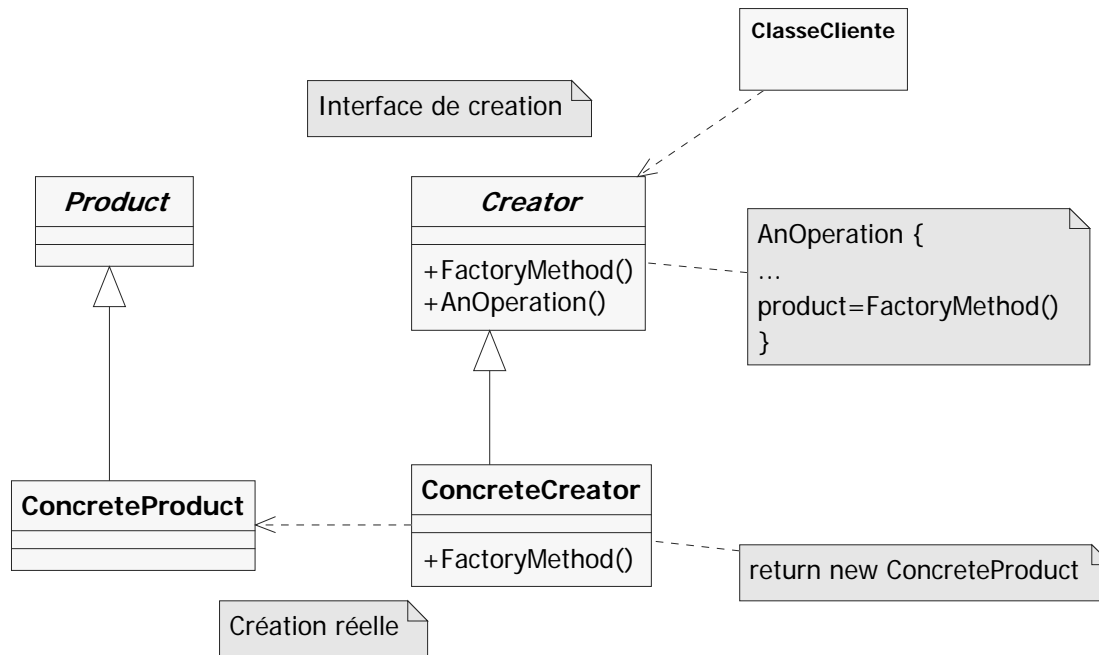
But : créer des objets de 2 hiérarchies sans connaître l'implémentation des classes

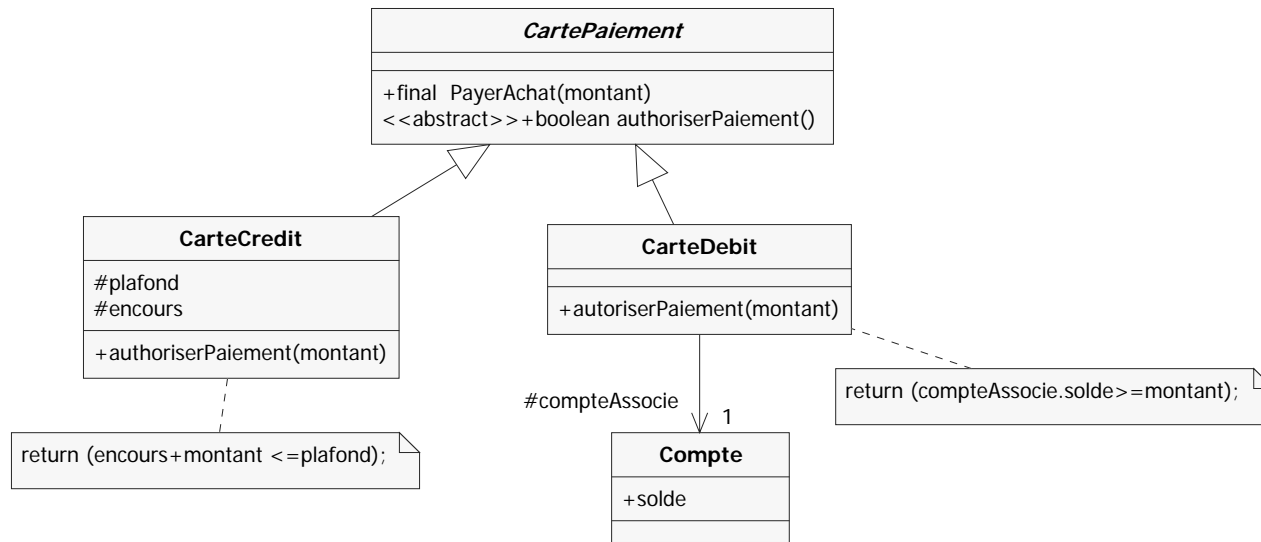


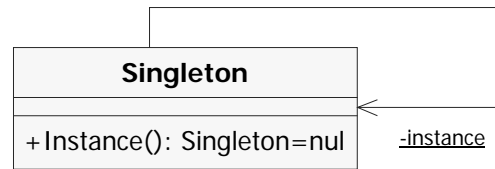


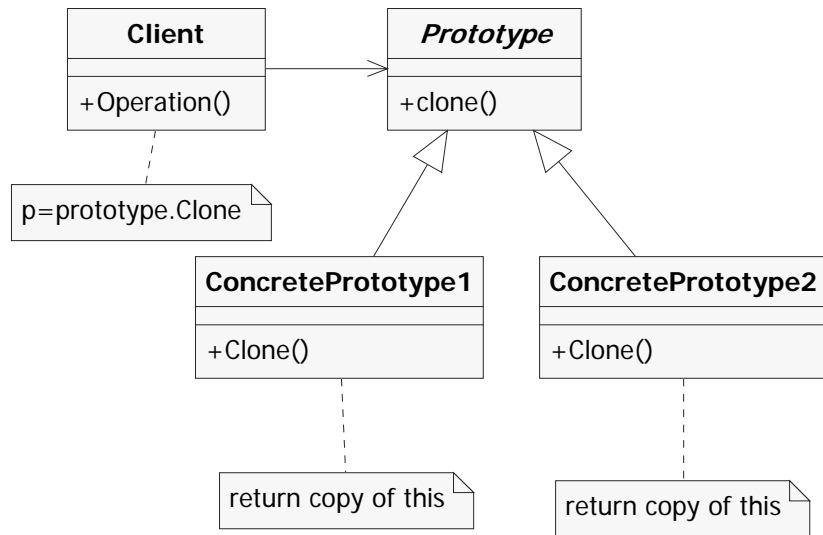


But : introduire une méthode abstraite de création d'un objet en reportant aux sous-classes concrètes la création effective





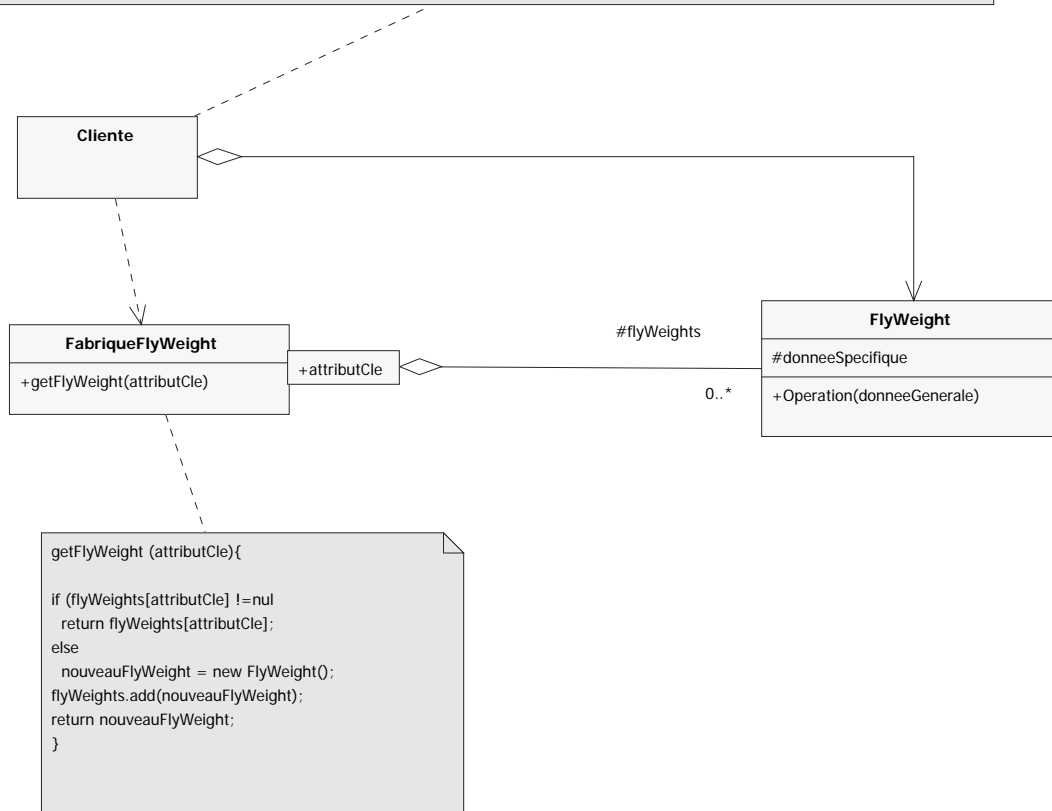


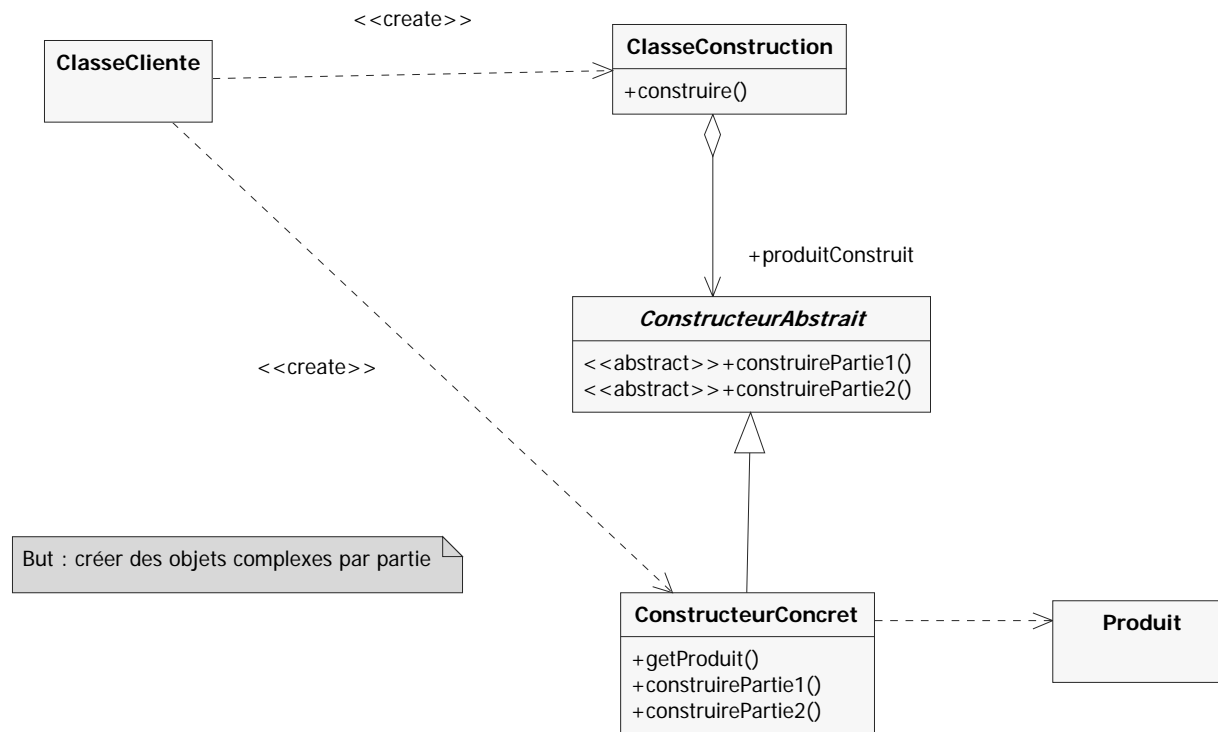


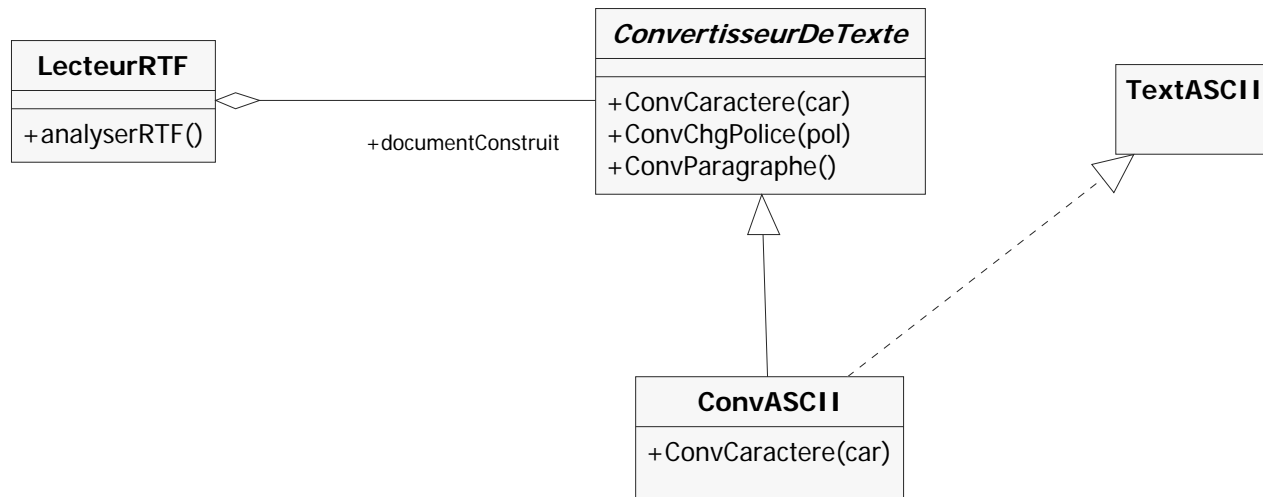
But: partager de façon efficace de nombreux objets de petite taille

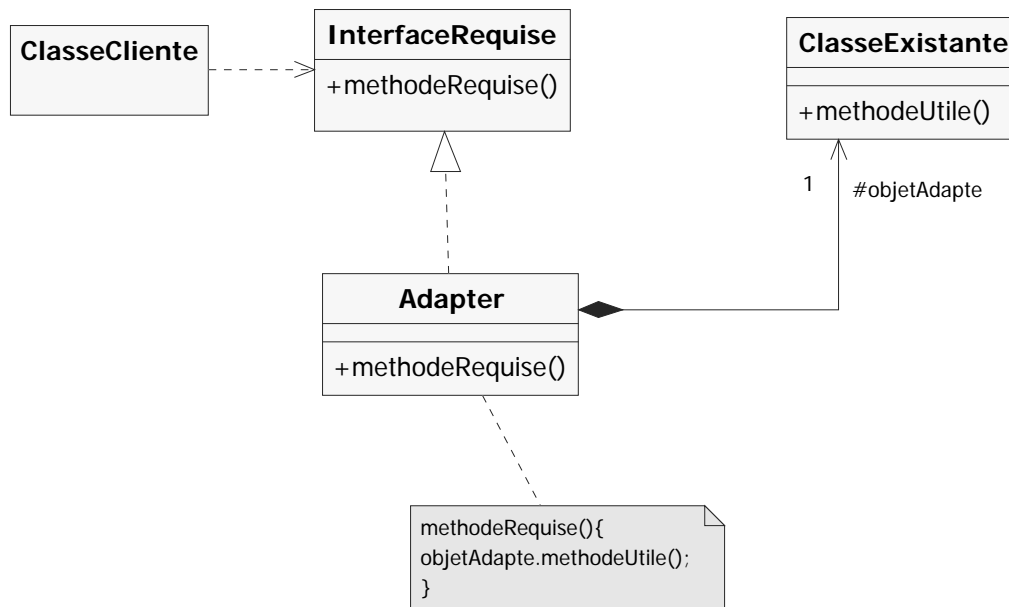
Ce DP propose de partager les données communes entre objets et donne un mécanisme pour créer les données spécifiques

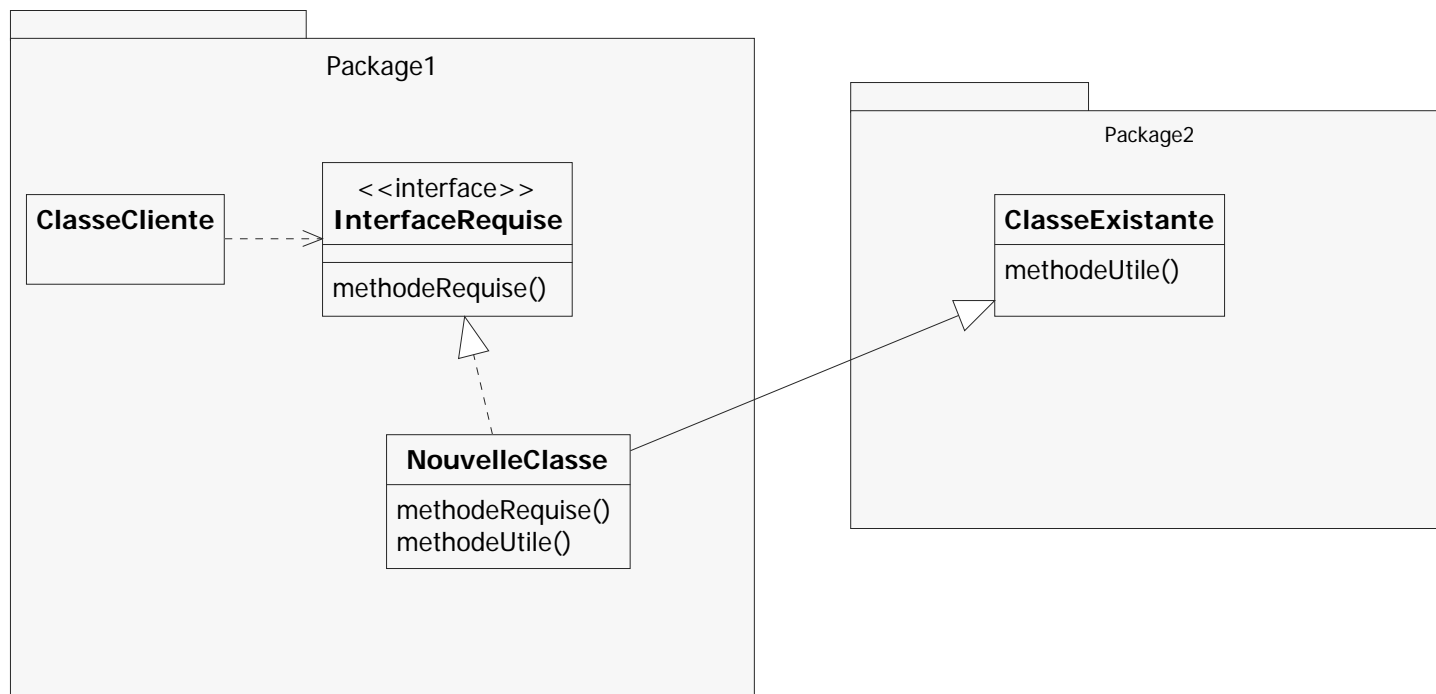
La classe Cliente ne doit pas creer elle-meme ses données spécifiques mais passer par getFlyWeight;





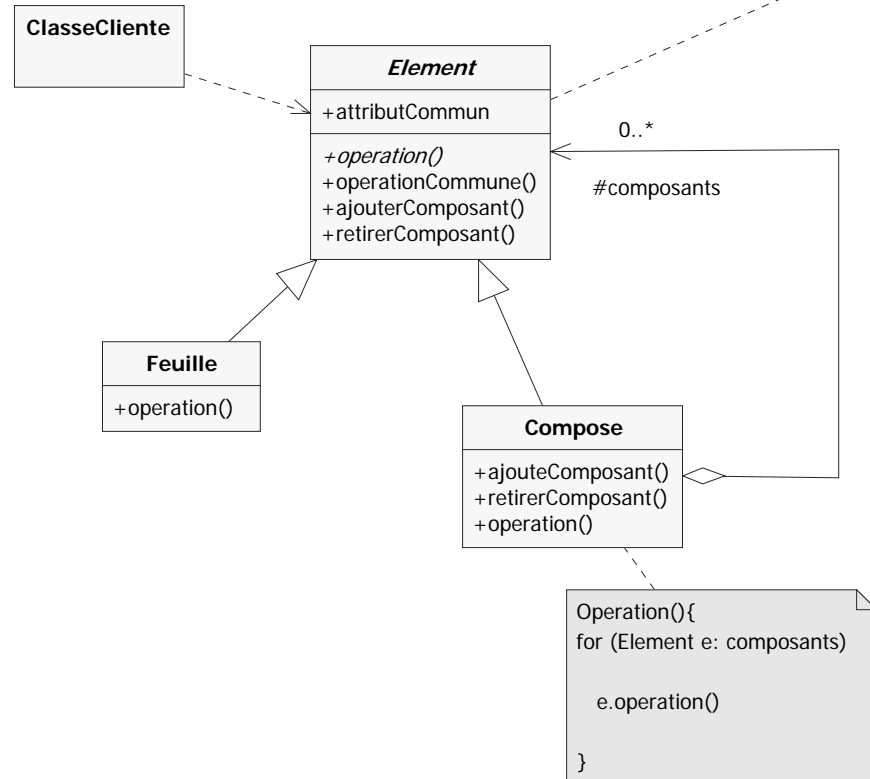


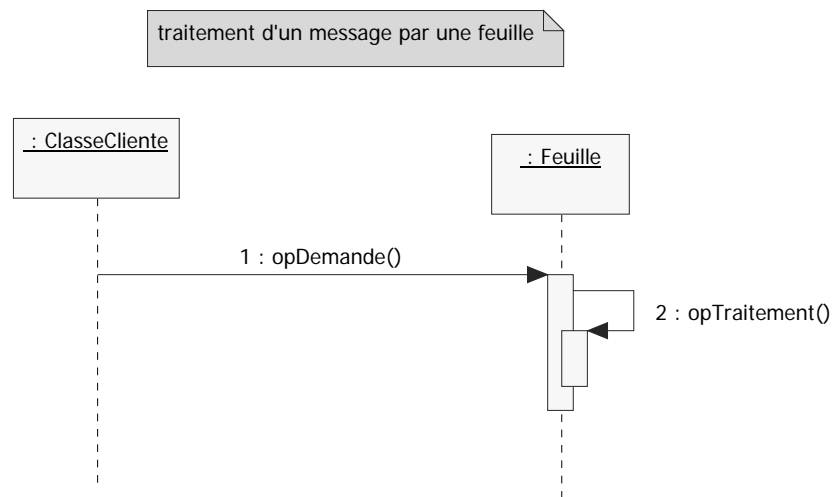


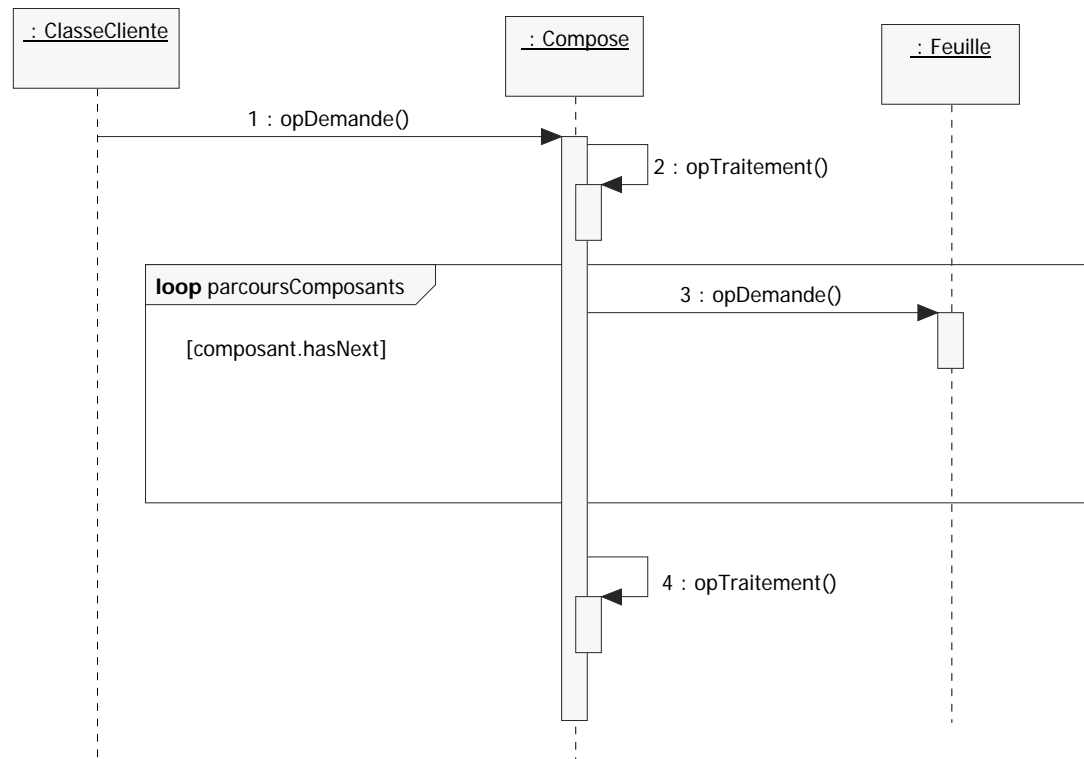


But : représenter une hiérarchie.
les clients ignorent s'ils communiquent
avec des objets composés ou non

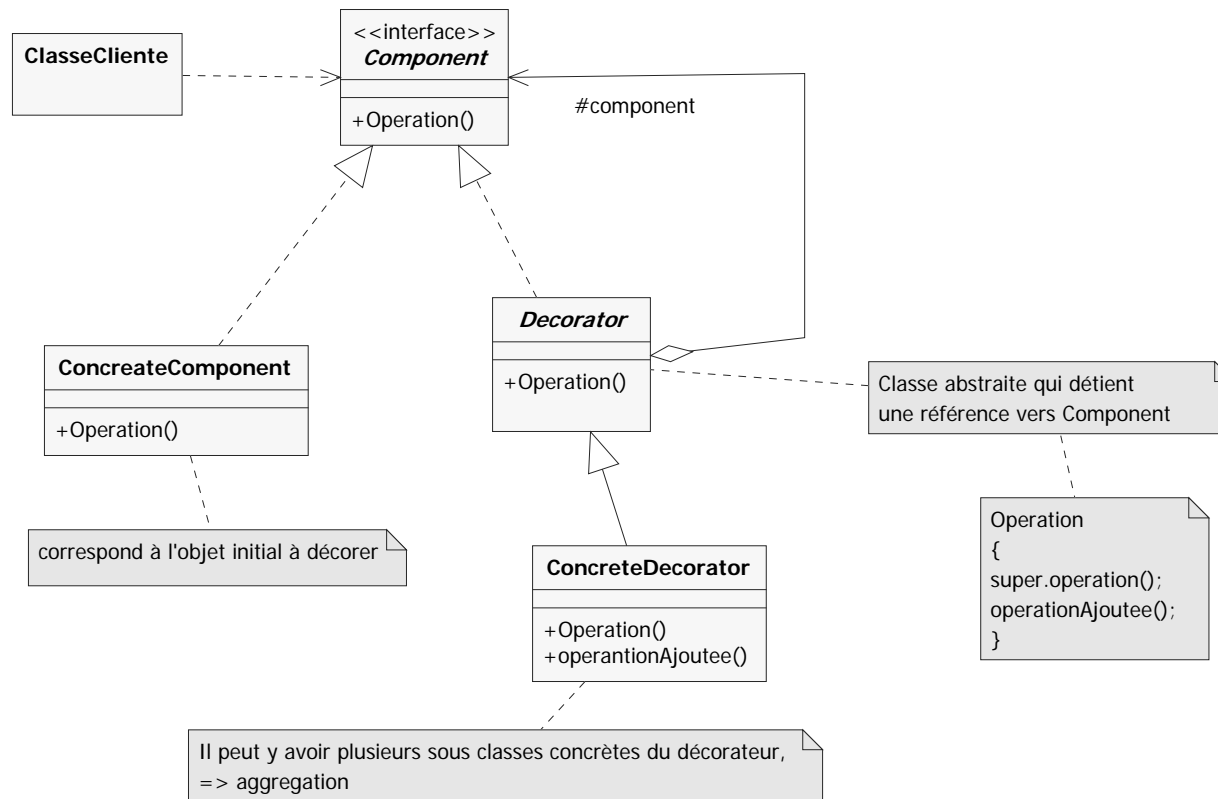
Classe Abstraite :
-éventuellement attributs communs et méthodes commune
-interface des objets de la composition
-décrit la signature des méthodes d'ajout ou suppression des composants



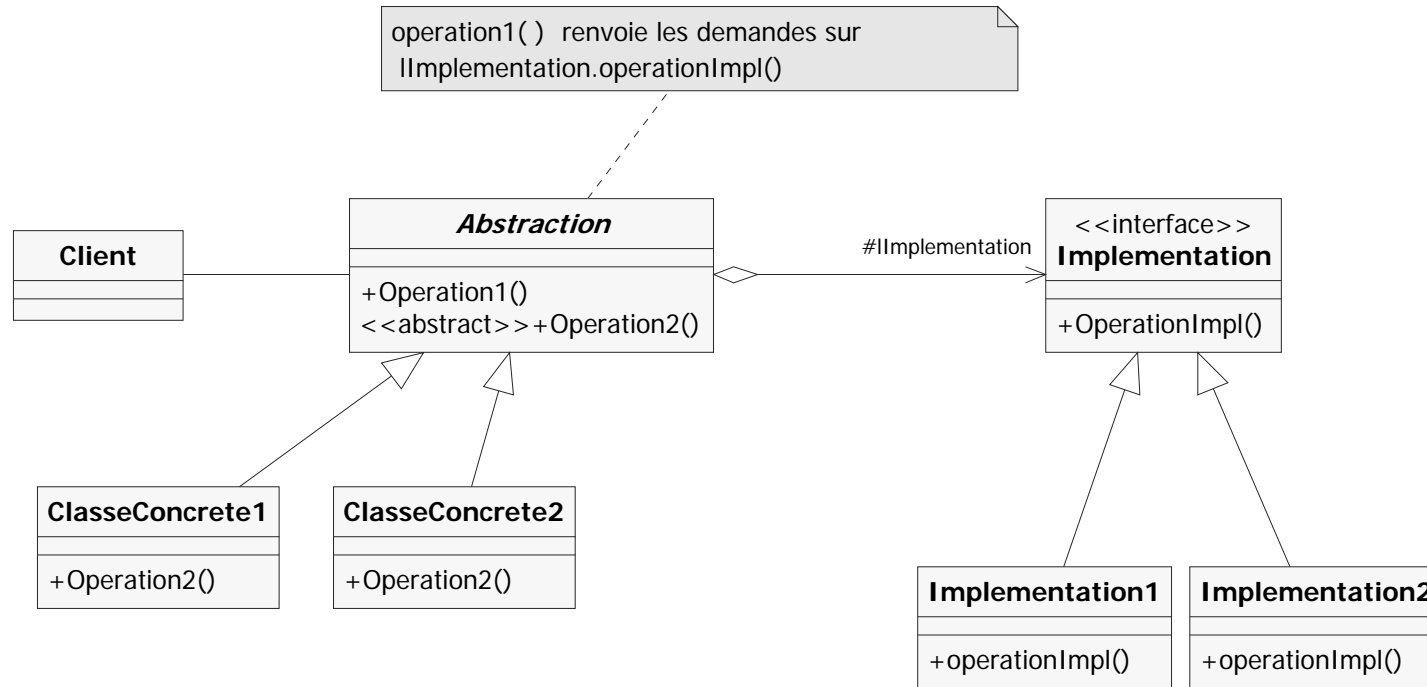


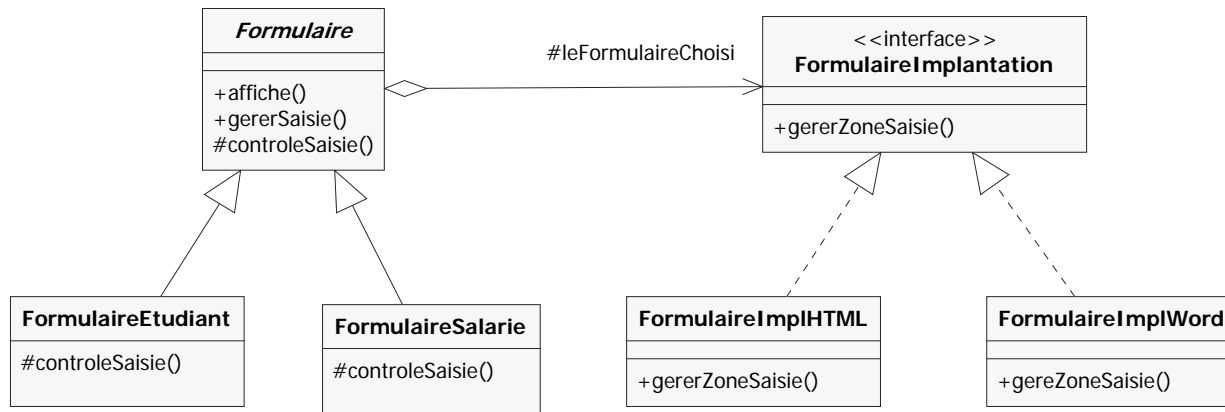


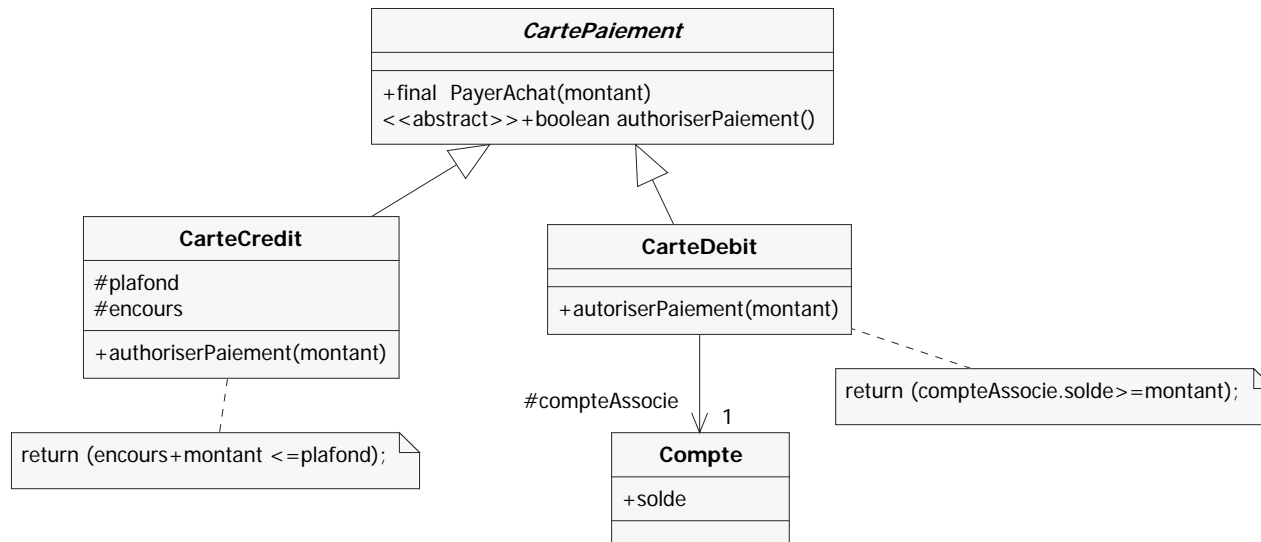
But : ajouter dynamiquement des fonctionnalités supplémentaires à un objet sans modifier l'interface de l'objet



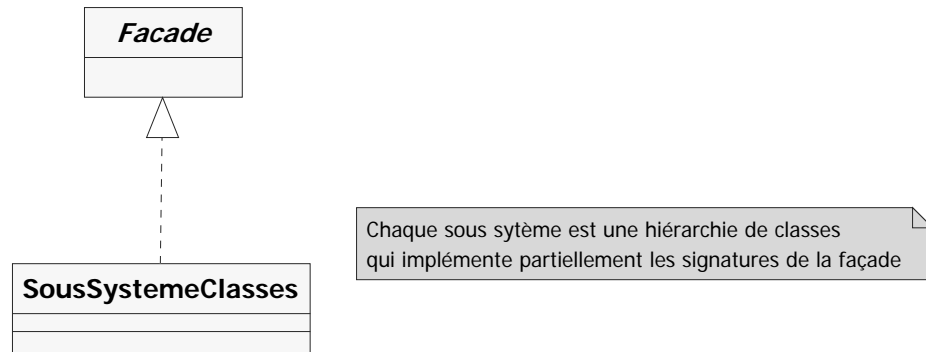
But : séparer l'aspect implémentation de sa modélisation (methodes d'interface)
ainsi l'implémentation et sa représentation par l'interface peuvent évoluer de façon indépendante



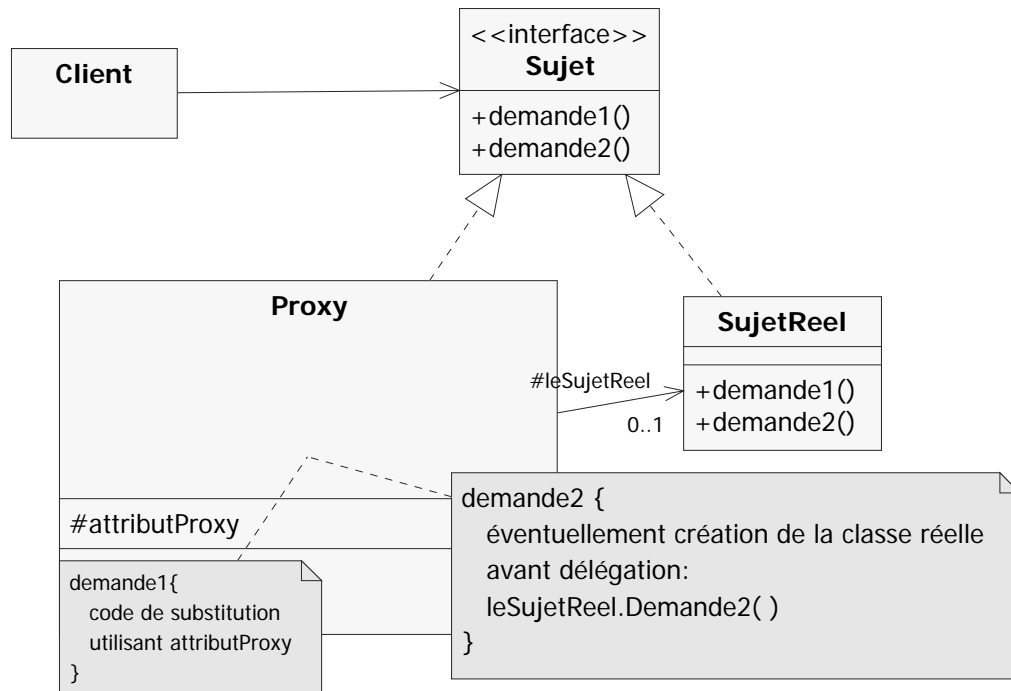




But: regrouper dans une interface commune les interfaces de classes pour rendre plus facile l'utilisation à une classe cliente.
Ce DP contribue à la modélisation de composants logiciels.

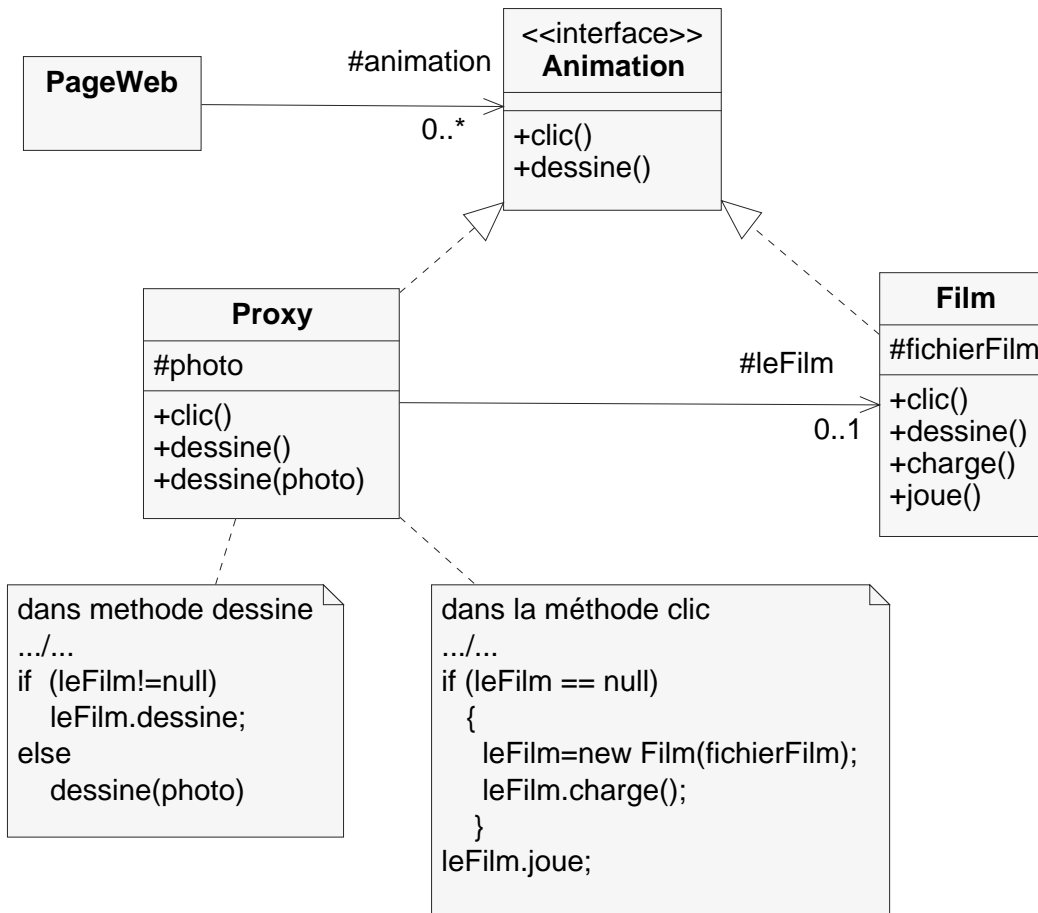


But : Le Proxy se substitue à la classe réelle de façon transparente pour la classe cliente .
Ce mécanisme permet de retarder la création de la classe réelle quand on en a réellement besoin.



En fait 2 comportements sont possibles:
soit le Proxy effectue lui-même la demande par un code de substitution,
soit il la délègue à la classe réelle.

Sur votre page Web, vous affichez des liens vers vos vidéos YouTube favorites.
Lorsque la page s'affiche, une photo est dessinée pour chaque lien,
en attendant que l'utilisateur clique pour jouer une vidéo qui est alors chargée.



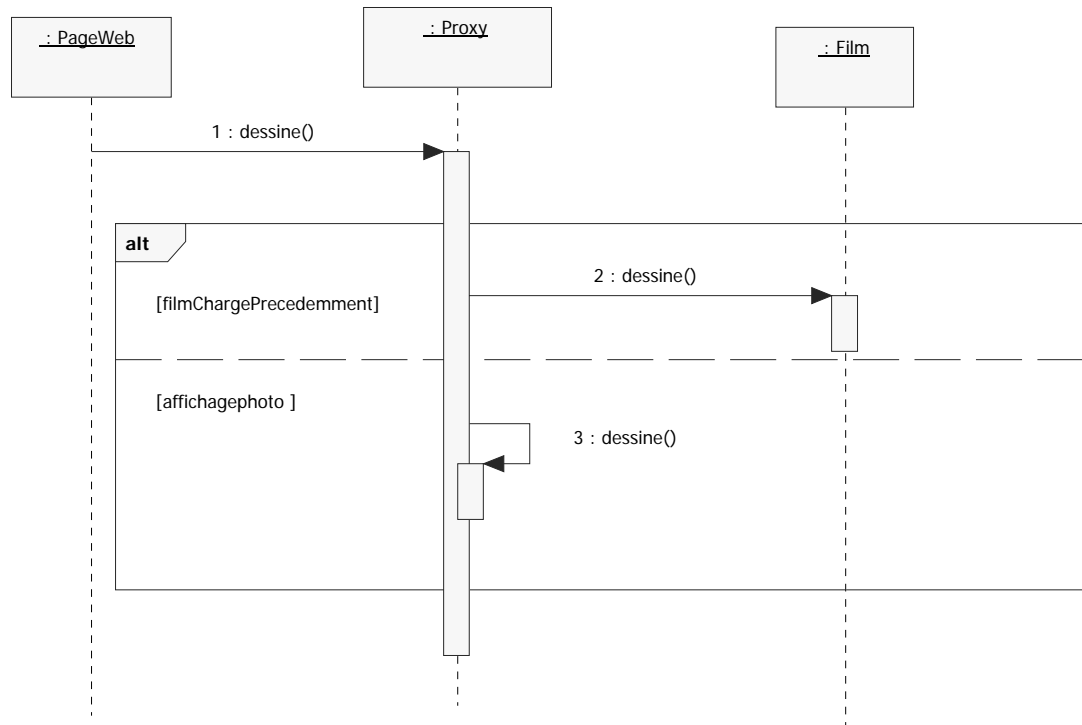
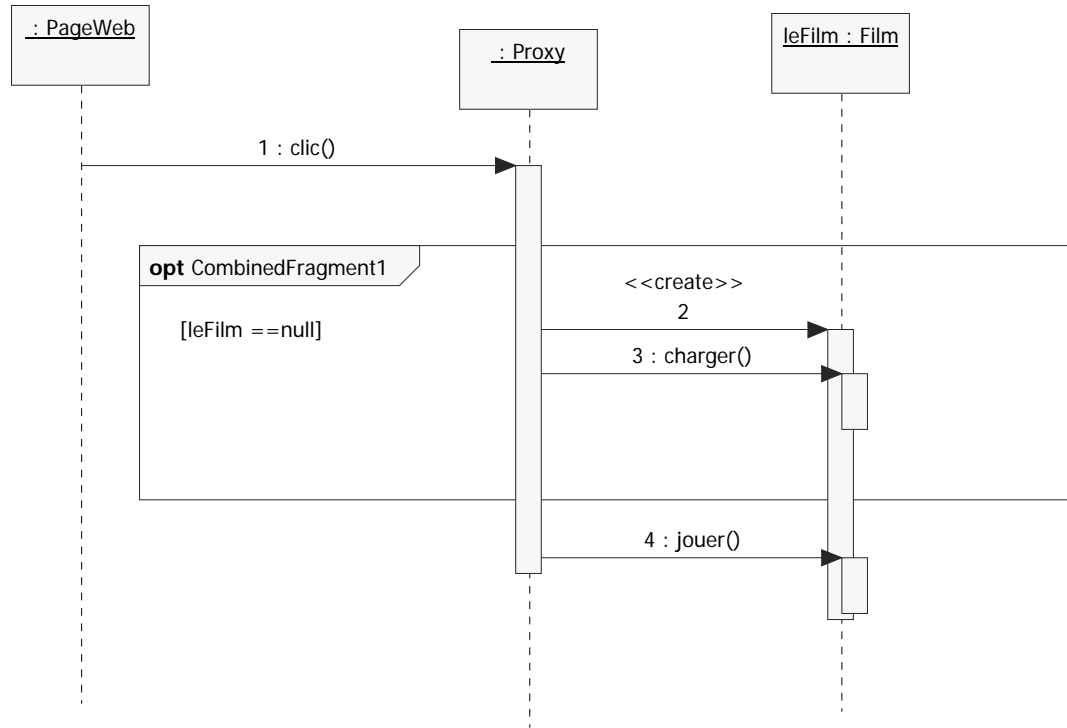
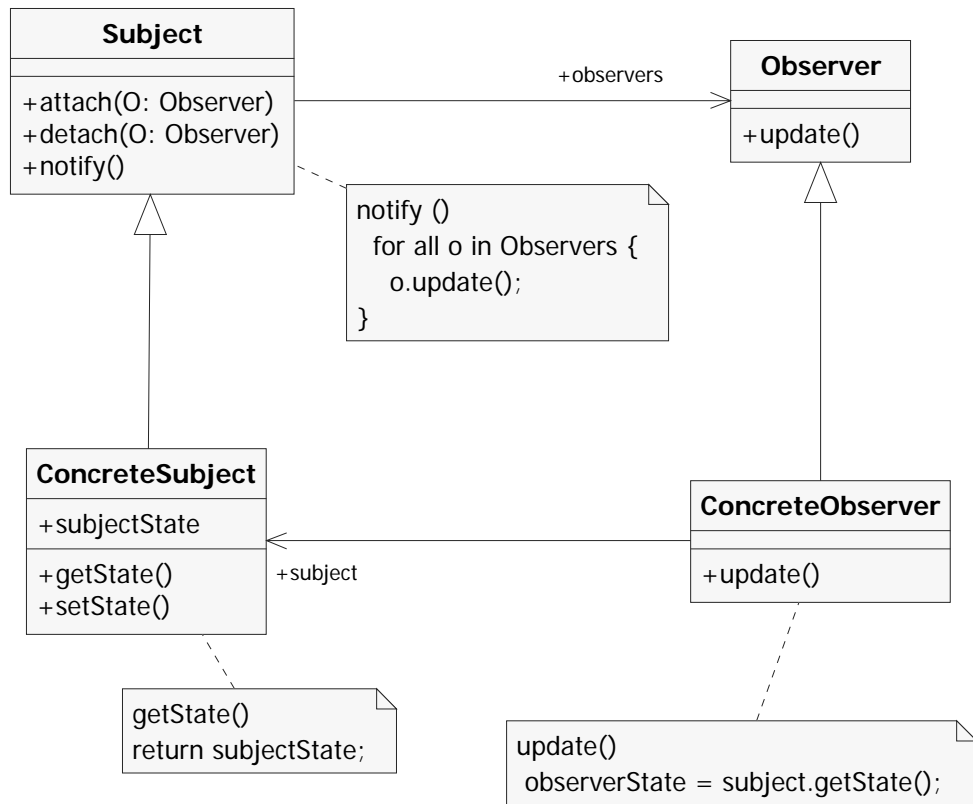
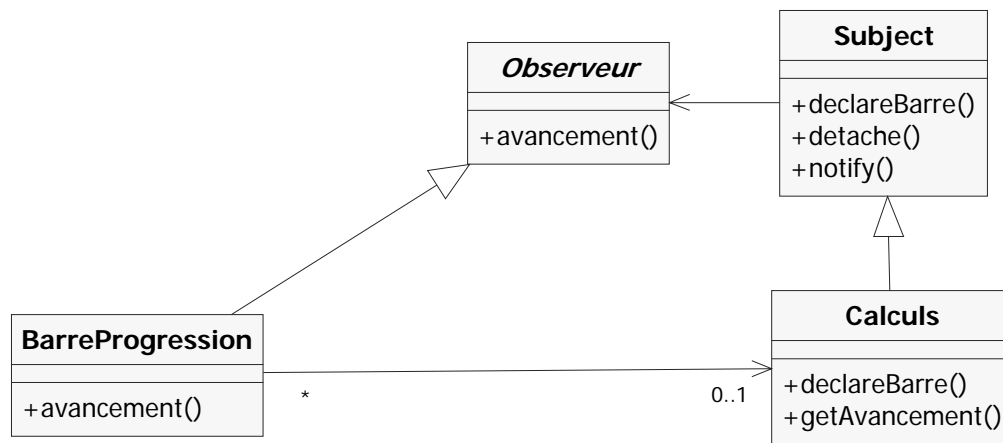
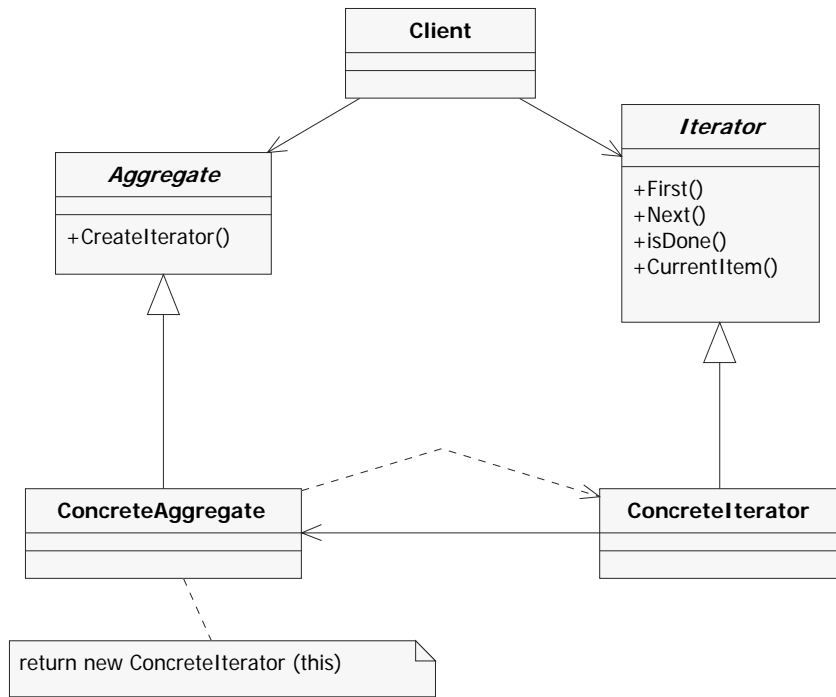


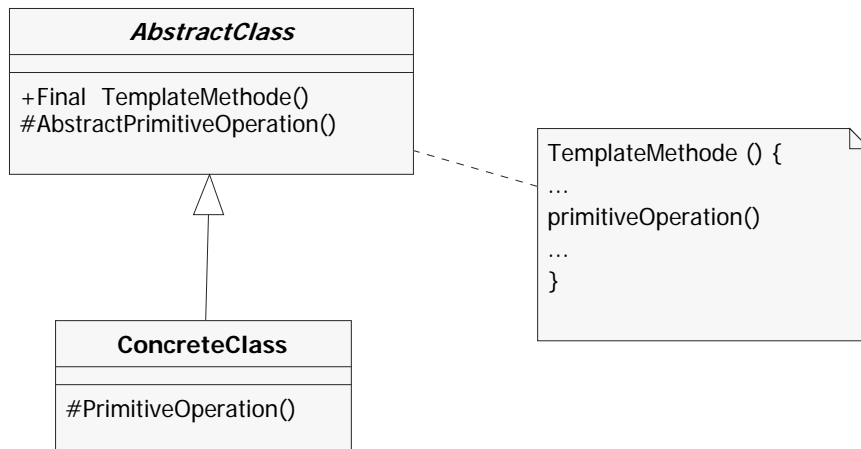
Diagramme de séquence du message Clic



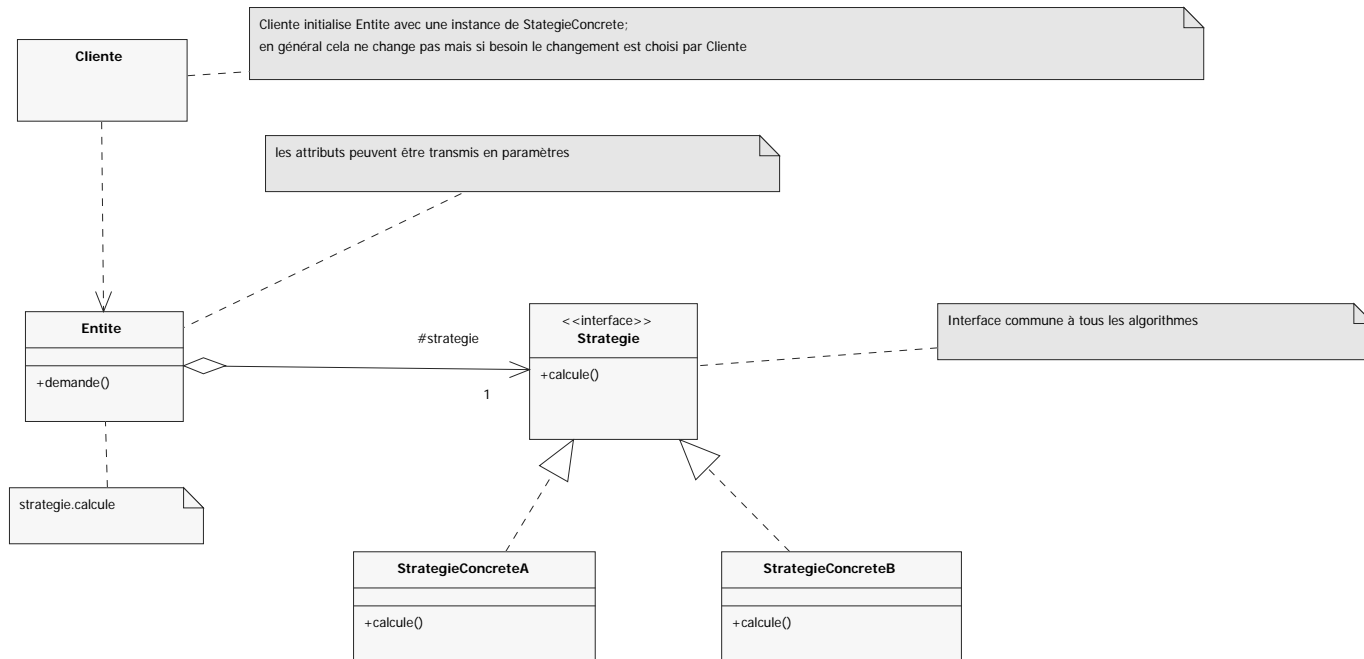








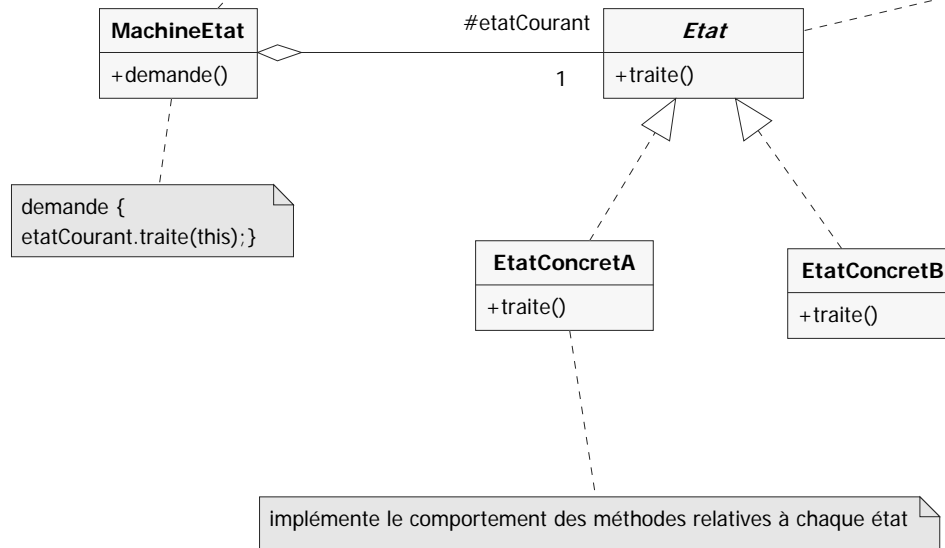
But : adapter le comportement et les algorithmes d'un objet sans changer les interactions de cet objet avec les clients



But : permettre à un objet d'adapter son comportement en fonction de son état interne
en évitant l'implantation de cette dépendance par des

classe concrète d'objets considérés comme machine à états.
On peut en faire le diagramme d'états
maintient une référence vers une instance d'une sous-classe d'Etat; c'est l'état courant

donne les signatures des méthodes liées à l'état



But : construire une opération à réaliser sur les éléments d'un ensemble d'objets, sans modifier ces classes

