**Builder:**

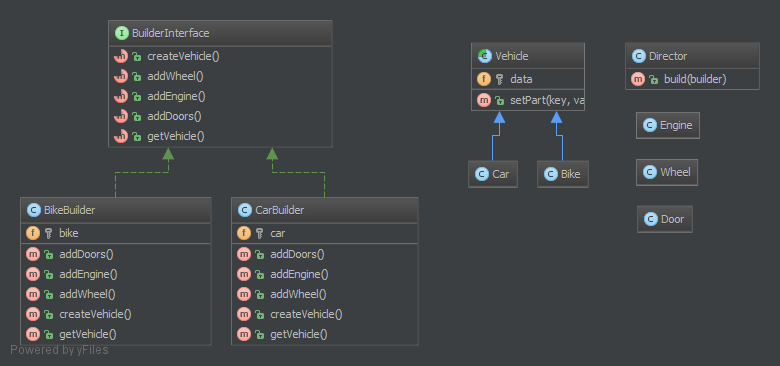
Le Patern Builder est un design patern de construction.

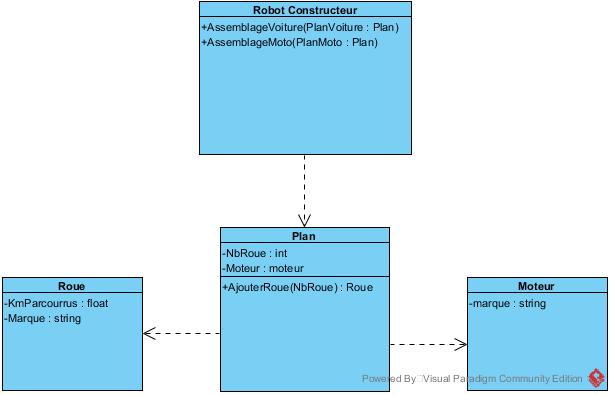
Il est utilisé pour créer des objets complexes à partir de multiples objets simples .

Exemple : un véhicule. L'objet véhicule se compose d'autres objets (roues, portières, moteur, etc...).

Concrètement on va avoir une classe “objet complexe” qui aura une liste de méthodes qui lui permettent de créer des instances des classes “objet simple”.

Le but de ce pattern est de séparer la construction d’un objet complexe de sa représentation, de cette façon, une même processus de construction peut créer plusieurs représentation différentes.

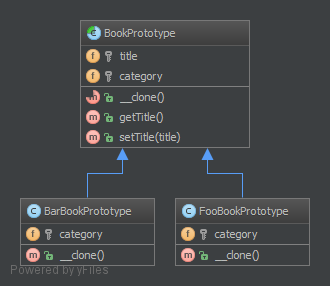


****

**Prototype :**

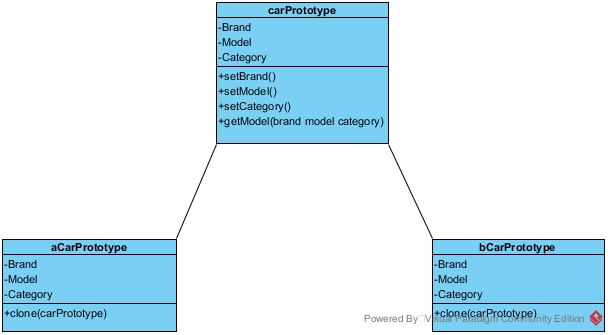
Le pattern Prototype permet de créer de nouveaux objets à partir d'objets existants. Ces objets sont appelés prototypes et possèdent une capacité de duplication. Dans le cadre de notre super méga géant constructeur d'ordinateurs et de pièces détachées, ce design pattern permettrait de créer toute la panoplie de papiers (bon de commande, facture, notice d'utilisation, etc.).

**Exemple :**



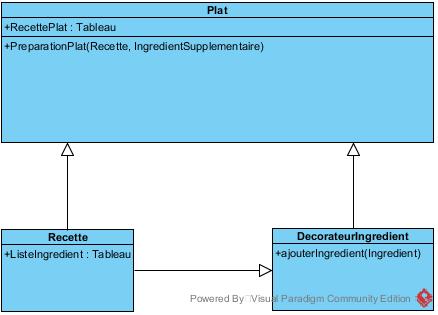
La classe Prototype est le squelette principal qui permet de créer les nouvelles copies. Les classes BarBookPrototype et FooBookPrototype sont les sous-classes spécifiques ayant leurs propres attributs. La méthode \_\_clone() retourne une copie de l'objet.

Diagramme de classe UML :



**Décoration :**

Le pattern Décorateur (Decorator en Anglais) permet d’ajouter (ou de modifier) des fonctionnalités facilement à un objet. D’une manière générale on constate que l’ajout de fonctionnalités dans un programme s’avère parfois délicat et complexe. Ce problème peut être résolu si le développeur a identifié, dès la conception, qu’une partie de l’application serait sujette à de fortes évolutions. Il peut alors faciliter ces modifications en utilisant le pattern Décorateur.



Note sur le Diagramme UML :

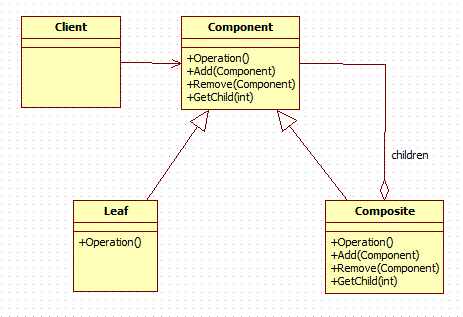
Le diagramme UML ci-dessus, donne un exemple du pattern décorateur. Dans ce pattern, la fonction « PreparationPlat » permet de prendre les ingrédients du tableau de la ListIngredient qui se trouve dans l’objet recette, mais aussi de rajouter les ingrédients de la méthode « ajouterIngrédient » qui se trouve dans l’objet « DecorateurIngredient ». Ce pattern permet de rejouter des ingrédients dans l’objet plat sans obliger de rajouter un objet ingrédient.

Il est utilisé pour :

* Ajouter de la fonctionnalité optionnelle à une classe
* Permet de ne pas se répéter (quand toute les classes ont la même fonctionnalité on peut utiliser le décorateur pour éviter cela)
* Séparer des responsabilités
* Permet de réduire la complexité du code dans une classe
* Permet de ne pas être dépendant de la hiérarchie

**Composite :** composer des objets ensembles (structure d’arbre)

Composez les objets dans les structures arborescentes pour représenter les hiérarchies partielles-entières. Composite permet aux clients de traiter des objets individuels et des compositions d'objets uniformément.



* **Composant**
  + est l'abstraction pour tous les composants, y compris ceux qui sont composés
  + déclare l'interface pour le comportement par défaut
* **Feuille**
  + représente un composant n'ayant pas de sous-éléments
  + implémente le comportement par défaut
* **Composite**
  + représente un composant pouvant avoir des sous-éléments
  + stocke des composants enfants et permet d'y accéder
  + implémente un comportement en utilisant les enfants
* **Client**
  + manipule les objets de la composition à travers l'interface de la classe Composant

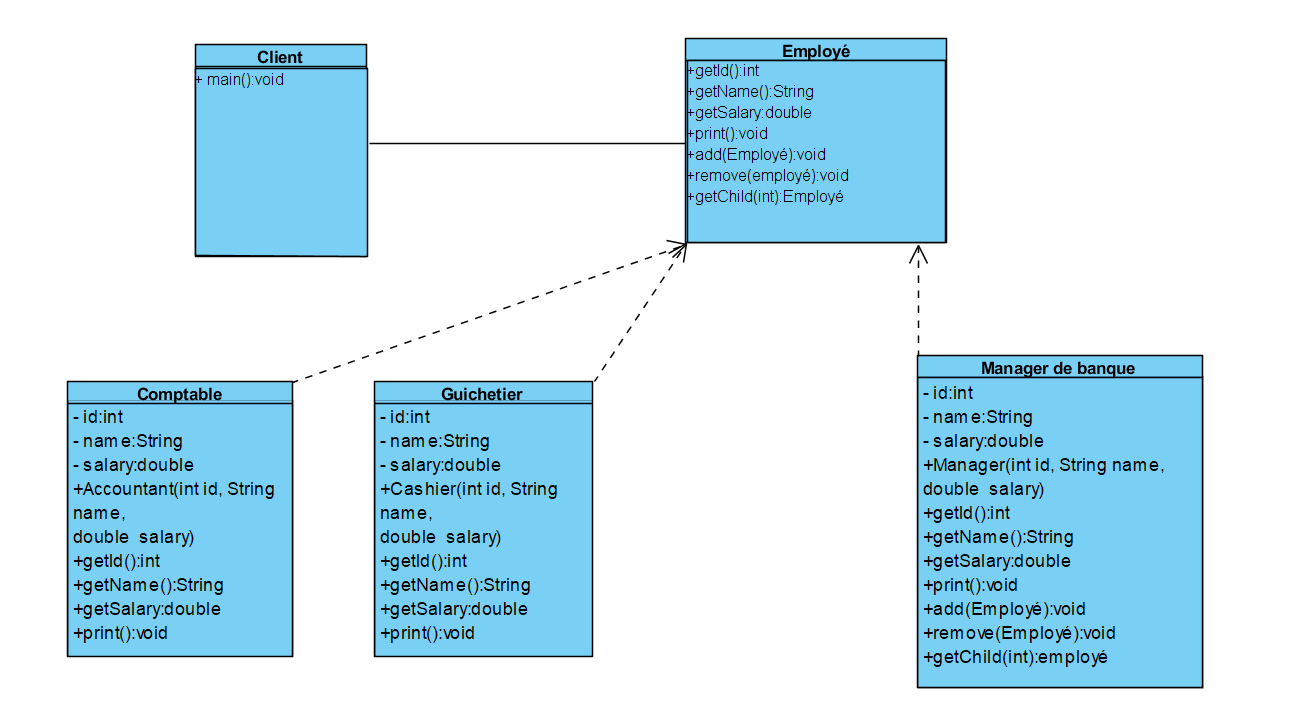
EXEMPLE CONCRET :

Composant : Employé d’une banque

Composite : Manager de la banque

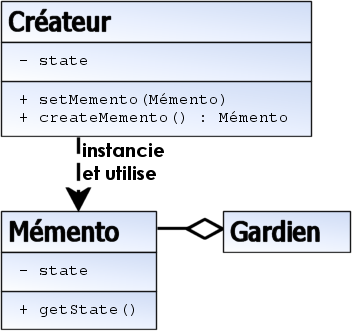
Feuille :

* Comptable
* Conseillère
* Guichetier
* …



**Memento :**  
Le patron Memento enregistre et extériorise l’état interne d’un objet pour pouvoir restaurer l’objet à un état précédant plus tard. L’état de l’objet est sauvegardé à des points stratégiques pour que le mécanisme « undo » permettent aux utilisateurs de restaurer un objet à l’état ou il était avant une certaine opération.

UML GENERAL :



UML PROGRAMME :

