

Projet Programmation Système A3

« Master Chef Info »

Choix des différents design pattern :

1. Factory :

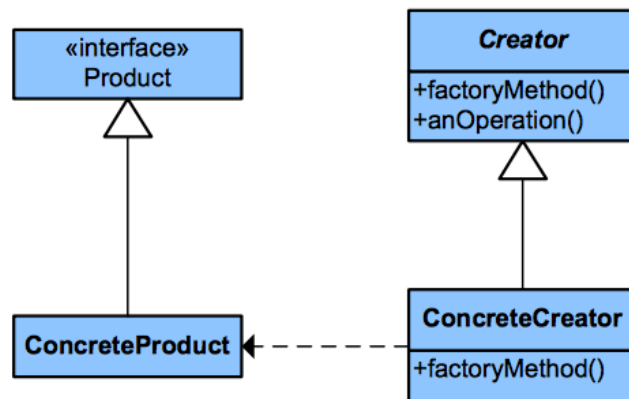
a) Rappel :

Factory Method

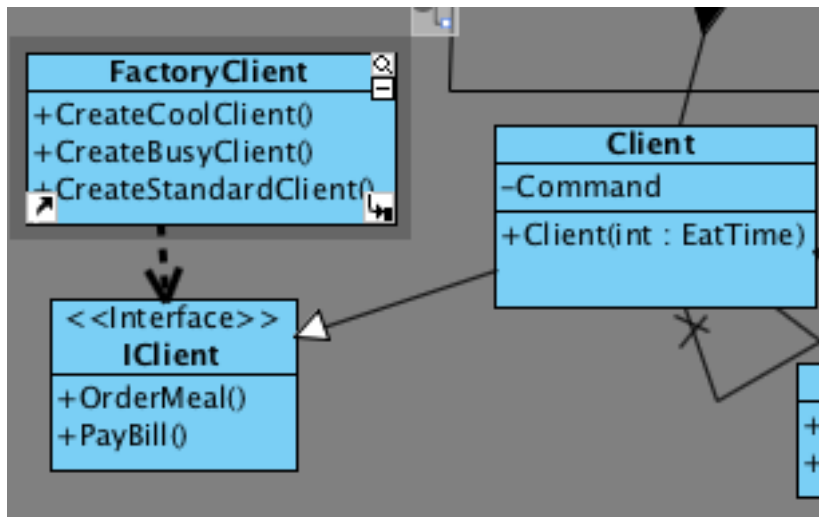
Type: Creational

What it is:

Define an interface for creating an object, but let subclasses decide which class to instantiate. Lets a class defer instantiation to subclasses.



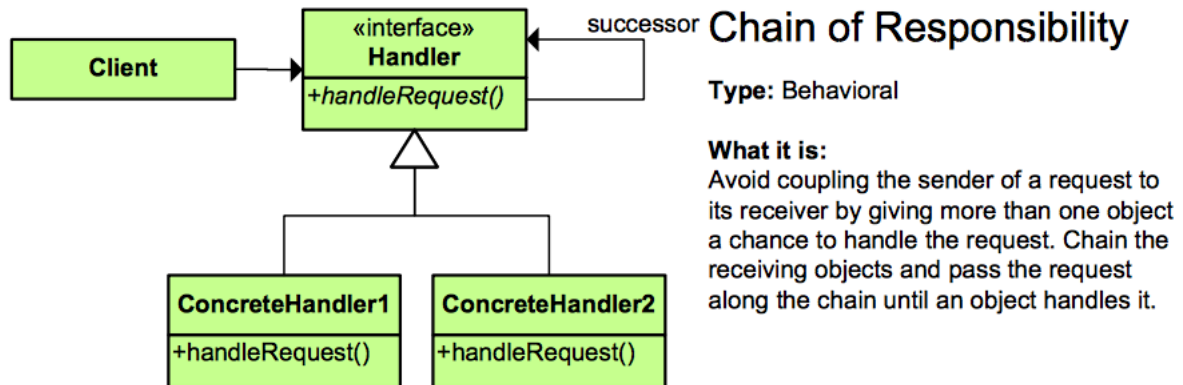
b) Mise en place :



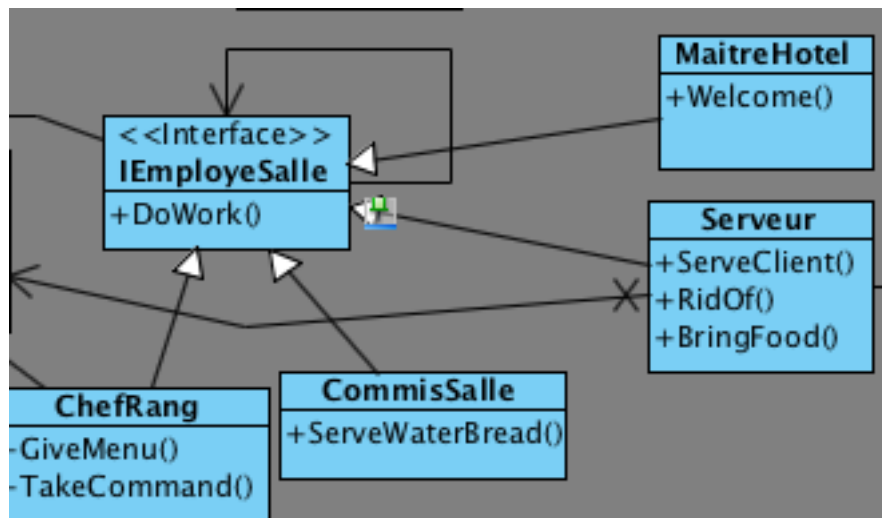
On définit une factory de client, qui nous permettra de créer à la volé des clients de différent type, ici 'cool', 'pressé', 'normal' On aura donc des IClient qui seront créé.

2. Chain of responsibility :

a) Rappel :



b) Mise en place :



La chaine de responsabilité nous permet de préciser à qui passer un ou plusieurs objets vers un destinataire précis et ainsi de suite.

3. Observer :

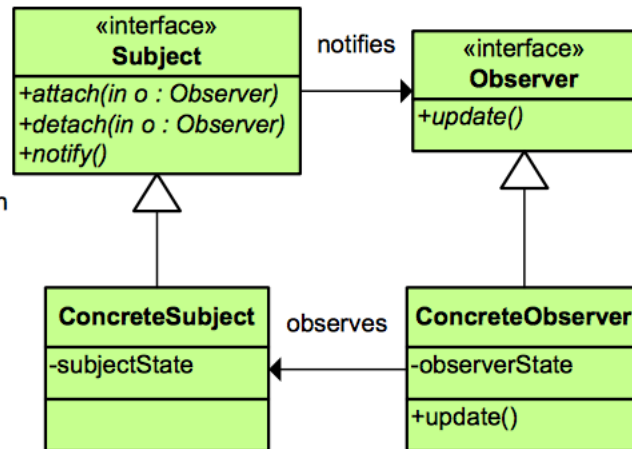
a) Rappel :

Observer

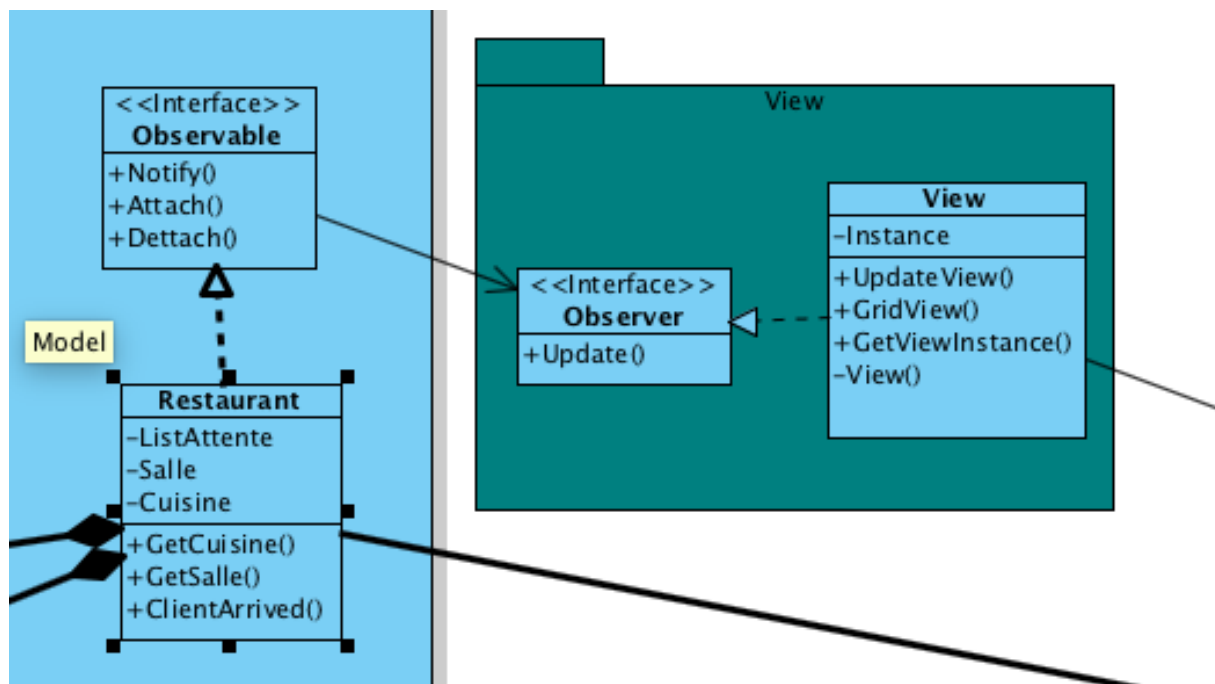
Type: Behavioral

What it is:

Define a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically.



b) Mise en place :



On implémente l'observer afin de modifier la vue quand un élément est modifier/ajouter/supprimer... dans notre model.

4. Strategy :

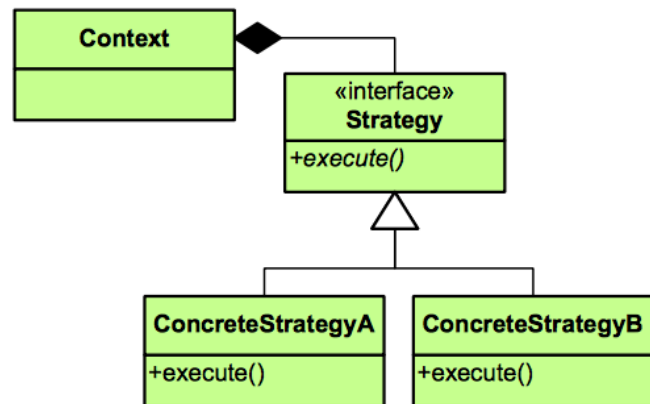
a) Rappel :

Strategy

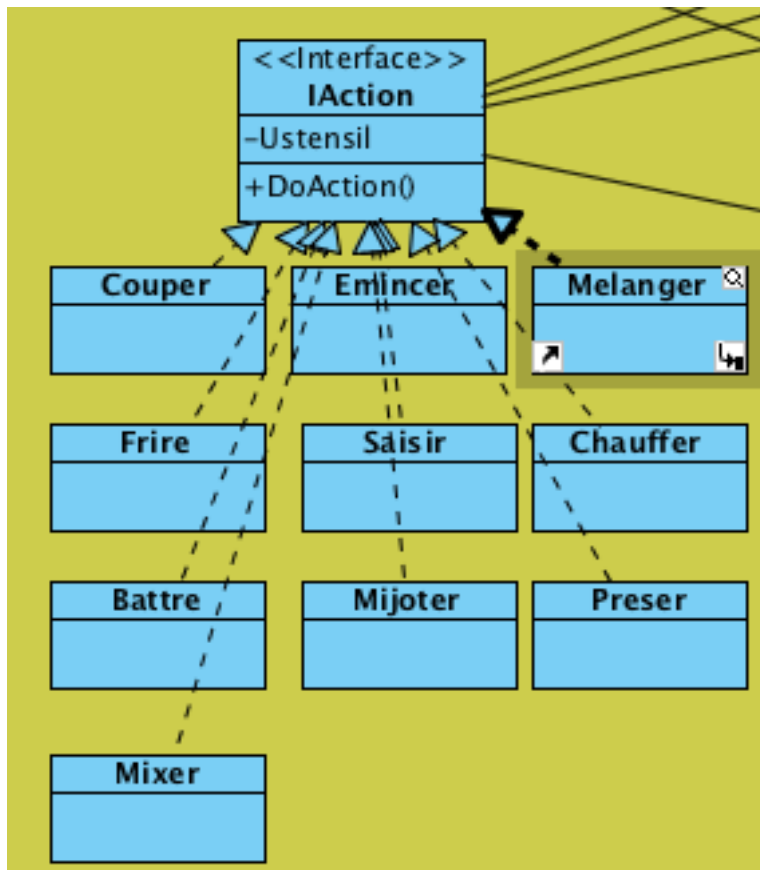
Type: Behavioral

What it is:

Define a family of algorithms, encapsulate each one, and make them interchangeable. Lets the algorithm vary independently from clients that use it.



b) Mise en place :



Le design pattern strategy nous permet de changer à la volé la méthode utiliser en fonction de l'action à faire sur la recette.

Ex : -Quand on veut découper une carotte, on passera donc par la classe 'Couper'

-Quand on veut cuire un steak, on passera par la classe 'Chauffer'

5. Singleton

a) Rappel :

Singleton

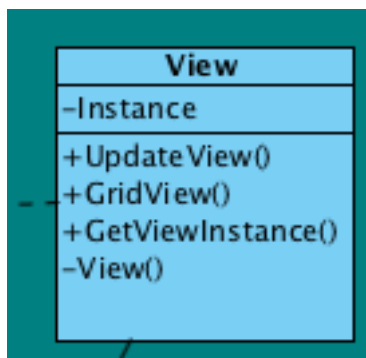
Type: Creational

What it is:

Ensure a class only has one instance and provide a global point of access to it.

Singleton
-static uniqueInstance -singletonData
+static instance() +SingletonOperation()

b) Mise en place :



On aura un singleton de notre vue afin d'être sûr d'en avoir qu'une seule instance