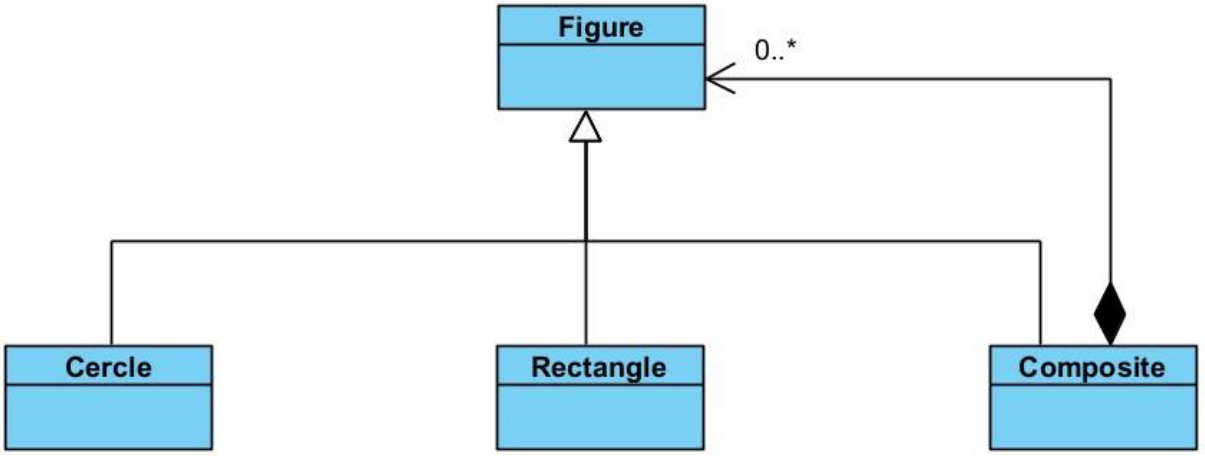
Travail à Rendre Design patterns

Halim YOUNESS

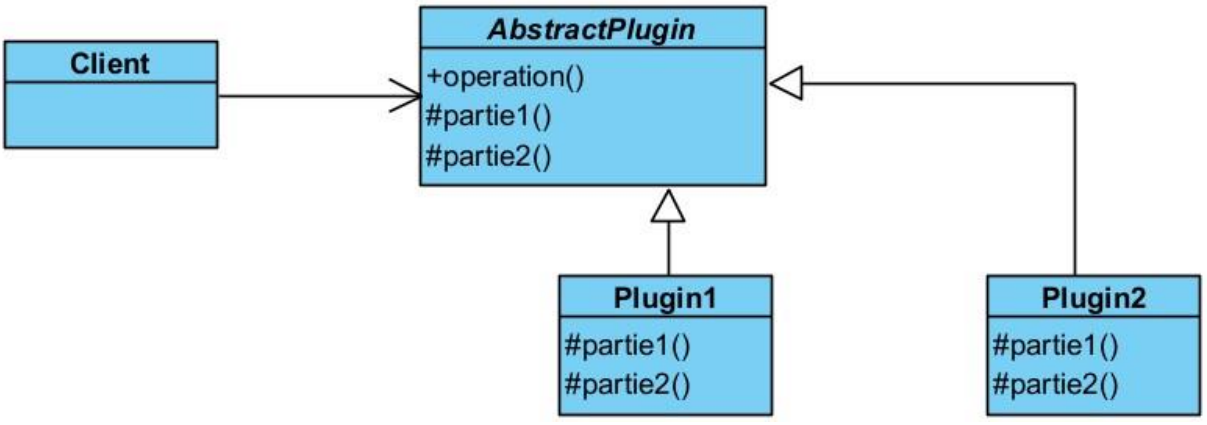
Exercice 1 :

Créer les diagrammes de classes en mentionnant les designs patterns appropriés pour les situations suivantes :

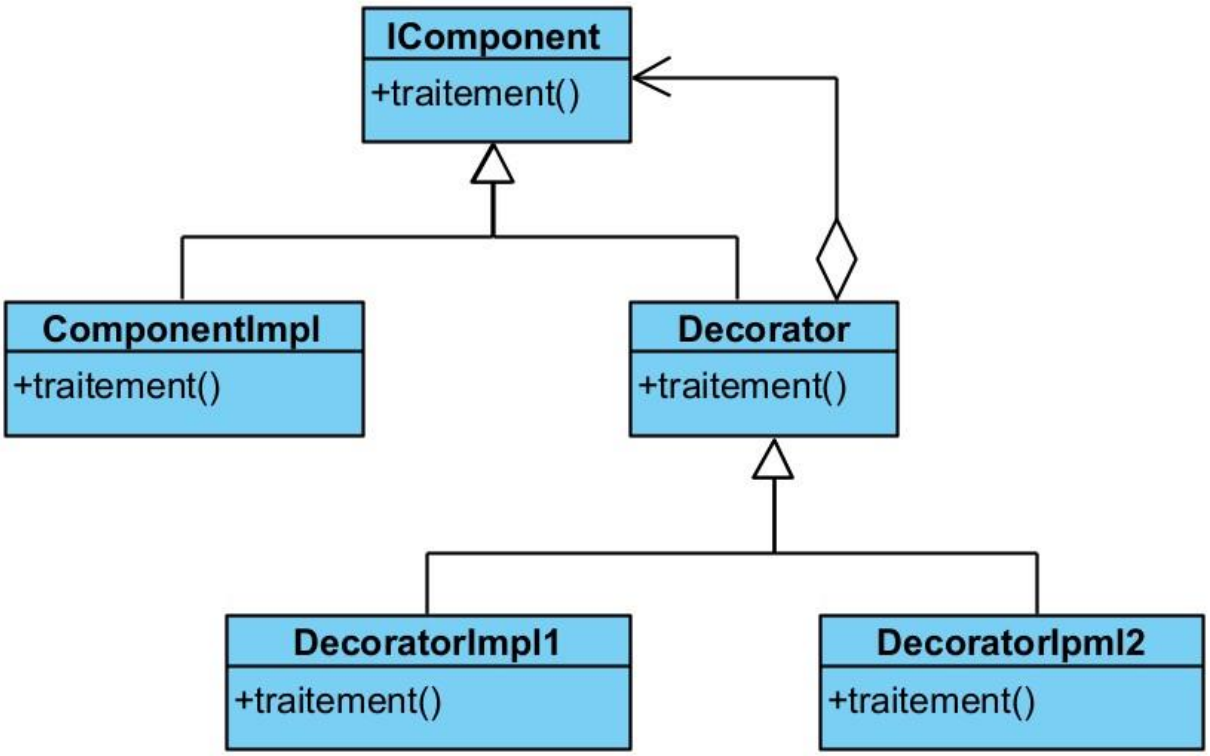
1- Une figure peut être soit un cercle, un rectangle ou un groupe de figures.



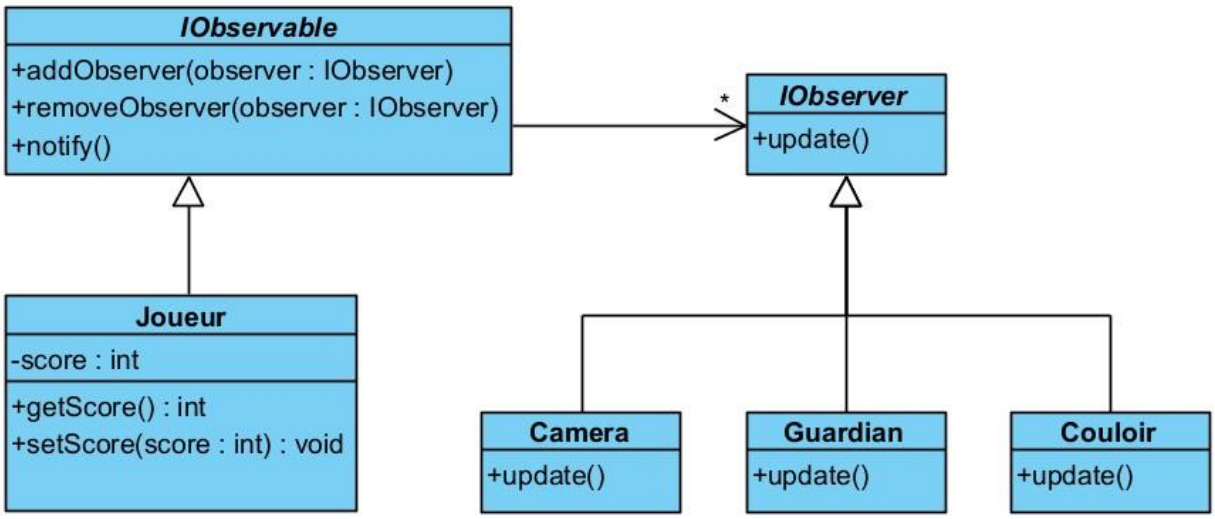
2- Un plugin contient une opération implémentant le squelette d'un algorithme dont deux parties (partie1 et partie2) sont variables. On voudrait laisser le développeur la possibilité d'implémenter les deux parties manquantes de cet algorithme et on voudrait aussi que l'application cliente puisse instancier une implémentation concrète du plugin sans connaitre sa classe d'implémentation.



3- On dispose d'un **composant** implémentant une interface qui définir une opération « *traitement()* ». On voudrait rattacher à ce composant des responsabilités supplémentaires sans modifier son code source. C'est-à-dire envelopper l'exécution de la méthode traitement par d'autres traitements avant et après son exécution.



4- On désire créer une classe **Joueur** ayant un état représenté par une variable score de type int. On voudrait que les objets de l'environnement du jeu (Couloir, Caméra et Gardien) soient informés à chaque fois que le score du joueur change tout en gardant un couplage faible entre la classe Joueur et les autres classes.



Exercice 2 :

On souhaite concevoir et développer un Framework qui permet d'effectuer des traitements sur une image. L'image étant représentée par un tableau d'entiers. Le Framework définit deux opérations principales :

· Une opération qui permet de filtrer l'image dont la signature est :

- **public int[] filter (int [] data)**

· Une opération qui permet de compresser l'image dont la signature est :

- **public int[] compresser (int [] data)**

Le Framework doit respecter les critères suivants :

· Il doit être fermé à la modification et ouvert à l'extension.

· L'opération de filtrage à effectuer peut évoluer dans le temps. Cela signifie que l'utilisateur de l'application peut, lui-même, définir de nouvelles implémentations de l'opération de filtrage.

· Au moment de l'exécution, on peut changer dynamiquement la version de l'implémentation de filtrage à appliquer à l'image.

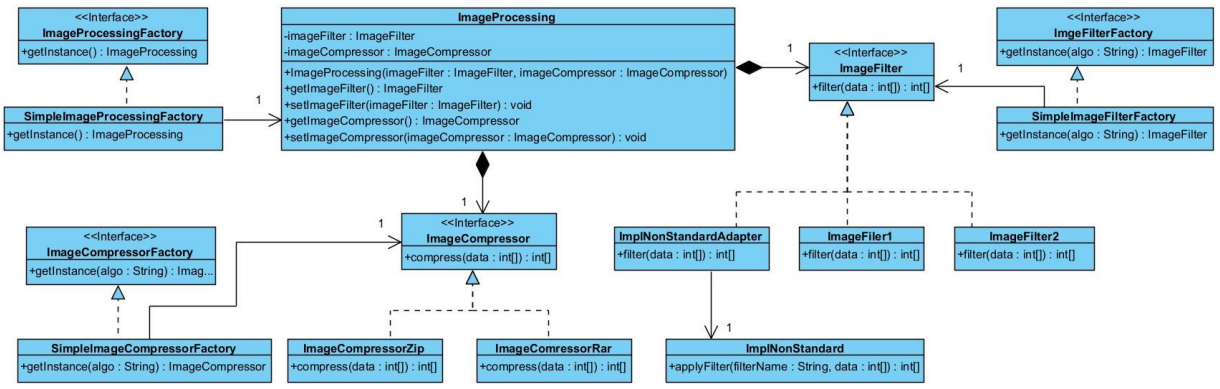
· Permettre au Framework d'utiliser une ancienne implémentation (ImplNonStandard) d'une opération de filtrage dont la signature est :

- **public int[] appliquerFiltre(String filterName, int[] data)**

· Pour l'opération de compression, on souhaite définir le squelette de l'algorithme de compression et déléguer les détails de cet algorithme aux sous classes.

Questions :

1) Etablir un diagramme de classes de ce Framework en appliquant les design patterns appropriés.



2) Ecrire une implémentation Java de ce Framework.

3) Ecrire le code d’une application qui utilise ce Framework en permettant à l’utilisateur de le nom des classes d’implémentation à utiliser pour effectuer les traitements.

2 - 3 : https://github.com/yhalim8/TP2_design_Pattern