RAZAFIMANANTSOA José Harisson IGGLIA 5 N°12

RANAIVOHARITIANA Fiarenantsoa Andry Nomena IGGLIA 5 N°06

**Desing Pattern :** [**Chain of responsability**](https://www.bart-konieczny.com/fr/blog/design-patterns/chain-of-responsability)

Souvent le traitement d'un élément ne passe pas par une seule personne. Prenons le cas d'un jury d'admission à une formation. Tous les membres doivent alors se prononcer en ce qui concerne les aptitudes d'élève à rejoindre une formation. Dans le développement le même traitement à la chaîne est présente dans chain of responsability.  
  
Le design pattern chain of responsability (chaîne de responsabilité) est une bonne illustration informatique du traitement à la chaîne. Son fonctionnement repose sur la décomposition d'une opération en plusieurs tâches. Chaque de ces tâches est effectuée par un objet séparé. Le résultat final correspond à la réalisation de toutes les tâches.  
  
Un des gros avantages de la chaîne de responsabilité est la flexibilité. On peut décomposer une opération en nombre de tâches indéfinie. Cela peut être utile dans des opérations plus complexes, comme par exemple analyse des images avec plusieurs filtres graphiques. On peut alors imaginer que la première tâche de traitement consiste éliminer le bruit d'une image (lissage). Ensuite, sur une image plus pure, on pourra introduire la transparence à l'image et ainsi de suite. Le nombre d'opération est donc infinie.  
  
La seule limite d'exécution non fixée explicitement dans le code, peut-être une erreur d'exécution. Imaginons que notre système de filtrage lance une exception non captée au deuxième filtre sur 10. Alors l'exécution d'autres filtres serait comprise. Il faudra prévoir cette possibilité de plantage dans le code, afin de garder le résultat final très près de celui attendu (même en cas d'échec d'un maillon de la chaîne).  
  
Grâce à cette gestion d'exceptions réfléchie, on peut rajouter des opérations à la chaîne sans se préoccuper de la stabilité de la chaîne. Or même en rajoutant un objet potentiellement défaillant, on est sûr qu'il ne compromettra pas l'opération générale. Cette idée de séparation garantit également un meilleur niveau d'isolation. Les modifications d'un objet n'impacteront pas le comportement d'un autre objet. Les deux sont pourtant indépendants.  
  
La chaîne de responsabilité est composée de 3 acteurs :   
- handler (gestionnaire) : définition de l'interface qui va gérer la requête à la chaîne.  
- concretehandler (gestionnaire concret) : l'implémentation concrète de l'handler. Il peut traiter la requête d'une manière différente ou alors se contenter de la transmettre à l'objet suivant dans la chaîne.  
- client : l'objet qui va initialiser le traitement.