

Pont

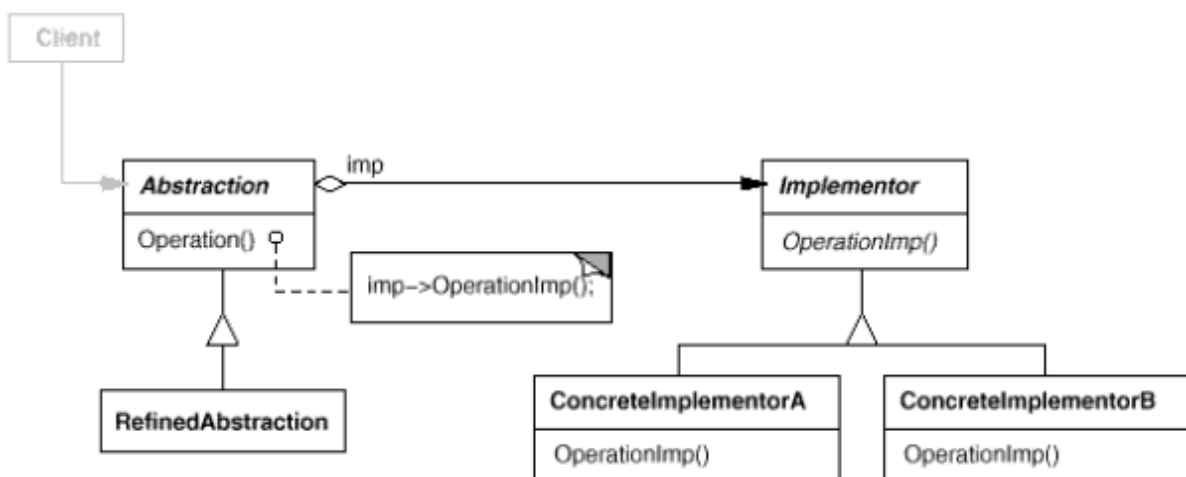
Intention:

- Découple une abstraction de son implémentation afin que les deux éléments puissent être modifiés indépendamment l'un de l'autre.

Indication d'utilisation :

- On souhaite éviter un lien définitif entre l'abstraction et son implémentation.
- Il faut que les abstractions et les implémentations soient toutes deux étendues par dérivation.
- Il ne faut pas que les modifications apportées à l'implémentation d'une abstraction aient un impact sur le code client, en particulier le code ne doit pas être recompilé.
- On souhaite cacher au client l'implémentation d'une abstraction.
- On est confronté à une prolifération de classes.
- On veut faire partager une même implémentation à plusieurs objets et cela doit être caché au client.

Structure:



Conséquence d'utilisation:

- Découplage de l'interface et de l'implémentation. Une implémentation n'est pas liée de façon permanente à une interface. L'implémentation d'une abstraction peut être composée à l'exécution. Il est même possible qu'un objet change d'implémentation lors de l'exécution.

- Capacité d'extension accrue. Les hiérarchies d'Abstraction et d'implémentation peuvent être étendues indépendamment l'une de l'autre.
- Dissimulation des détails d'implémentation aux clients. On peut masquer au client les détails d'implémentations.