

# Decorator

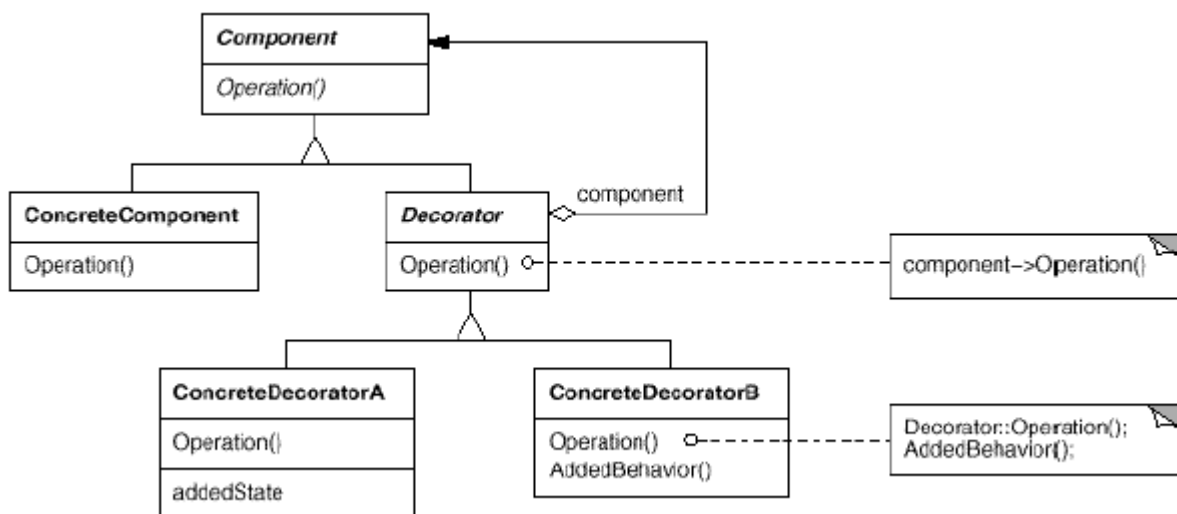
## Intention:

- Attache dynamiquement des responsabilités à un objet. Les décorateurs fournissent une alternative souple à la dérivation, pour étendre les fonctionnalités.

## Indication d'utilisation :

- Pour ajouter dynamiquement des responsabilités à des objets individuels, ceci d'une façon transparente, c.à.d. sans affecter les autres objets.
- Pour des responsabilités qui doivent être retirées.
- Quand l'extension par dérivation est impraticable. Trop de combinaisons possibles.

## Structure:



## Conséquence d'utilisation:

- Offre plus de souplesse que l'héritage statique
- Evite de surcharger des fonctionnalités les classes situées en haut de la hiérarchie.
- Un composant n'est pas identique
- Multitude petits objets, difficile à comprendre.