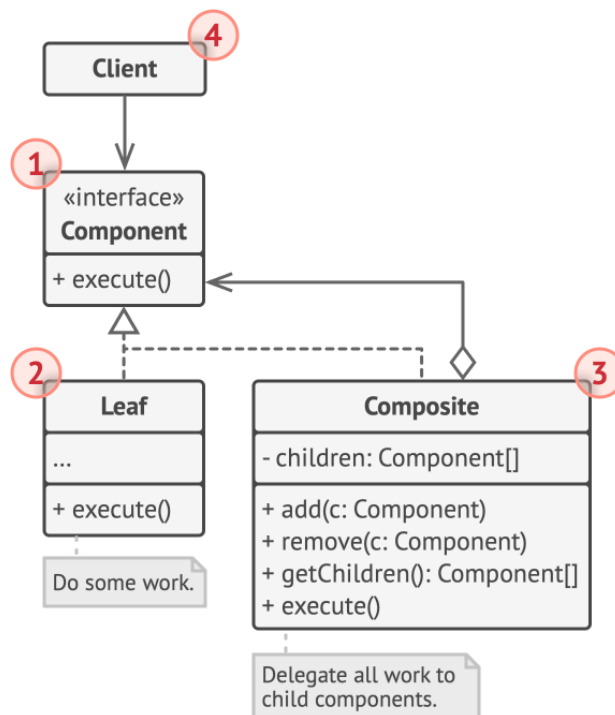


# Composite

E' un pattern strutturale che permette di comporre oggetti in strutture ad albero e di lavorare con tali come se fossero oggetti individuali.

## Struttura



- L'interfaccia Composite descrive le operazioni comuni sia agli elementi semplici che a quelli complessi dell'albero;
- La foglia è un elemento base di un albero che non ha sottoelementi;
- Il contenitore è un elemento che ha sottoelementi: foglie o altri contenitori. Un contenitore non conosce le classi concrete dei suoi figli;
- Il client interagisce con tutti gli elementi tramite l'interfaccia del componente.

## Applicabilità

- Usare il pattern quando è necessario implementare una struttura di oggetti ad albero;
- Usare il modello quando si desidera che il codice client tratti in modo uniforme sia gli elementi semplici che quelli complessi.

## Vantaggi

- E' possibile lavorare con strutture ad albero complesse in modo più pratico sfruttando il polimorfismo e la ricorsione;
- Principio open/closed.

## Svantaggi

- Potrebbe essere difficile fornire un'interfaccia comune per classi le cui funzionalità differiscono eccessivamente.

## Relazioni con altri pattern

- E' possibile usare Builder durante la creazione di alberi compositi complessi;
- Chain of Responsibility viene spesso usata insieme a Composite. In questo caso, quando un componente foglia riceve una richiesta, può inoltrarla attraverso la catena di tutti i componenti padre fino alla radice dell'albero degli oggetti;
- E' possibile usare gli iteratori per attraversare gli alberi ;
- E' possibile usare Visitor per eseguire un'operazione su un intero albero;
- E' possibile implementare nodi foglia condivisi dell'albero composito come Flyweight per risparmiare RAM;
- Composite e Decorator hanno strutture simili poichè entrambi si basano sulla composizione ricorsiva per organizzare un numero aperto di oggetti, ma

Decorator ha un solo componente figlio e aggiunge responsabilità aggiuntive all'oggetto incapsulato.

- I progetti che fanno uso di Composite e Decorator possono trarre vantaggio dall'uso di Prototype.