

COMPOSITE

JUAN IVAN MEDINA MARTINEZ

Que es?

Sirve para construir objetos complejos a partir de unos mas simple y similares entre si, gracias a la composición recursiva y a una estructura en forma de árbol, Todos tienen una interfaz en común por lo que se tratan de la misma manera.

Dependiendo de la implementación pueden tratarse a todos o solo a una parte al mismo tiempo, como si fueran nodos.

Este patrón da una solución elegante y sencilla al problema

Cuando se trata de datos estructurados en un árbol, los programadores a menudo tienen que discriminar entre un nodo hoja y una rama. Esto hace que el código sea mas complejo, y por lo tanto, propenso a errores.

La solución es una interfaz que permite traer objetos complejos y primitivos de manera uniforme.

En la programación orientada a objetos, un compuesto es un objeto diseñado como una composición de uno o mas objetos similares, todos exhibiendo una funcionalidad similar.

El concepto clave es que puede manipular una sola instancia del objeto tal como manipularía un grupo de ellos

Participantes en el patrón...

1.- Componente: Componente declara la interfaz para los objetos en la composición y para acceder y administrar sus componentes secundarios. También implementa un comportamiento predeterminado para la interfaz común a todas las clases, según corresponda.

2. Leaf : La hoja define el comportamiento de los objetos primitivos en la composición. Representa objetos de hoja en la composición.

3. Compuesto: Almacena componentes secundarios e implementa operaciones relacionadas con los hijos en la interfaz del componente.

4. Cliente: el cliente manipula los objetos en la composición a través de la interfaz del componente.

CLIENTE

El cliente usa la interfaz de clase de componente para interactuar con objetos en la estructura de composición. Si el destinatario es una hoja, la solicitud se gestiona directamente. Si el destinatario es un compuesto, generalmente reenvía la solicitud a sus componentes secundarios, posiblemente realizando operaciones adicionales antes y después del reenvío.

Cuando usarlo?

El patrón compuesto debe usarse cuando los clientes necesitan ignorar la diferencia entre composiciones de objetos y objetos individuales.

Si los programadores descubren que están usando múltiples objetos de la misma manera, y a menudo tienen un código casi idéntico para manejar cada uno de ellos, entonces el compuesto es una buena opción, en esta situación es menos complejo tratar los primitivos y los compuestos como homogéneos.

- 1.- Menos cantidad de objetos reduce el uso de memoria, y logra mantenernos alejados de errores relacionados con la memoria como `java.lang.OutOfMemoryError`.
- 2.- Aunque crear un objeto en Java es realmente rápido, aún podemos reducir el tiempo de ejecución de nuestro programa compartiendo objetos.

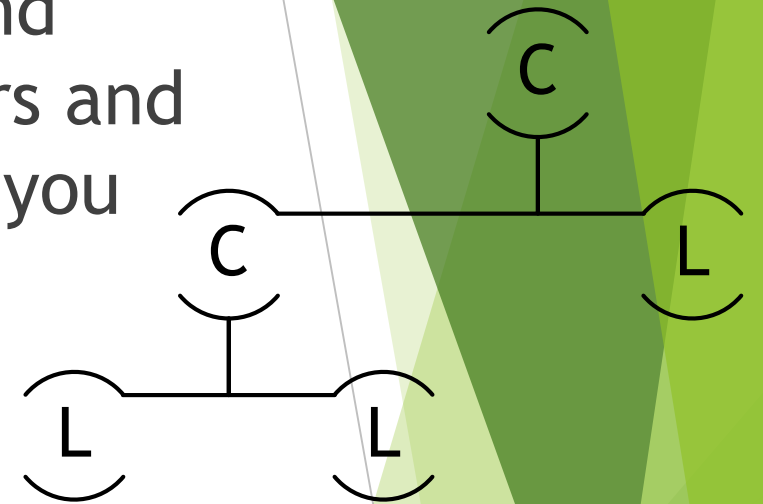
Cuando no usarlo?

1. El patrón de diseño compuesto dificulta la restricción del tipo de componentes de un compuesto. Por lo tanto, no debe usarse cuando no desee representar una jerarquía total o parcial de objetos.

2. El patrón de diseño compuesto puede hacer que el diseño sea demasiado general. Hace más difícil restringir los componentes de un compuesto. A veces quieres que un compuesto tenga solo ciertos componentes. Con Composite, no puede confiar en el sistema de tipos para imponer esas restricciones por usted. En su lugar, tendrá que usar verificaciones en tiempo de ejecución.

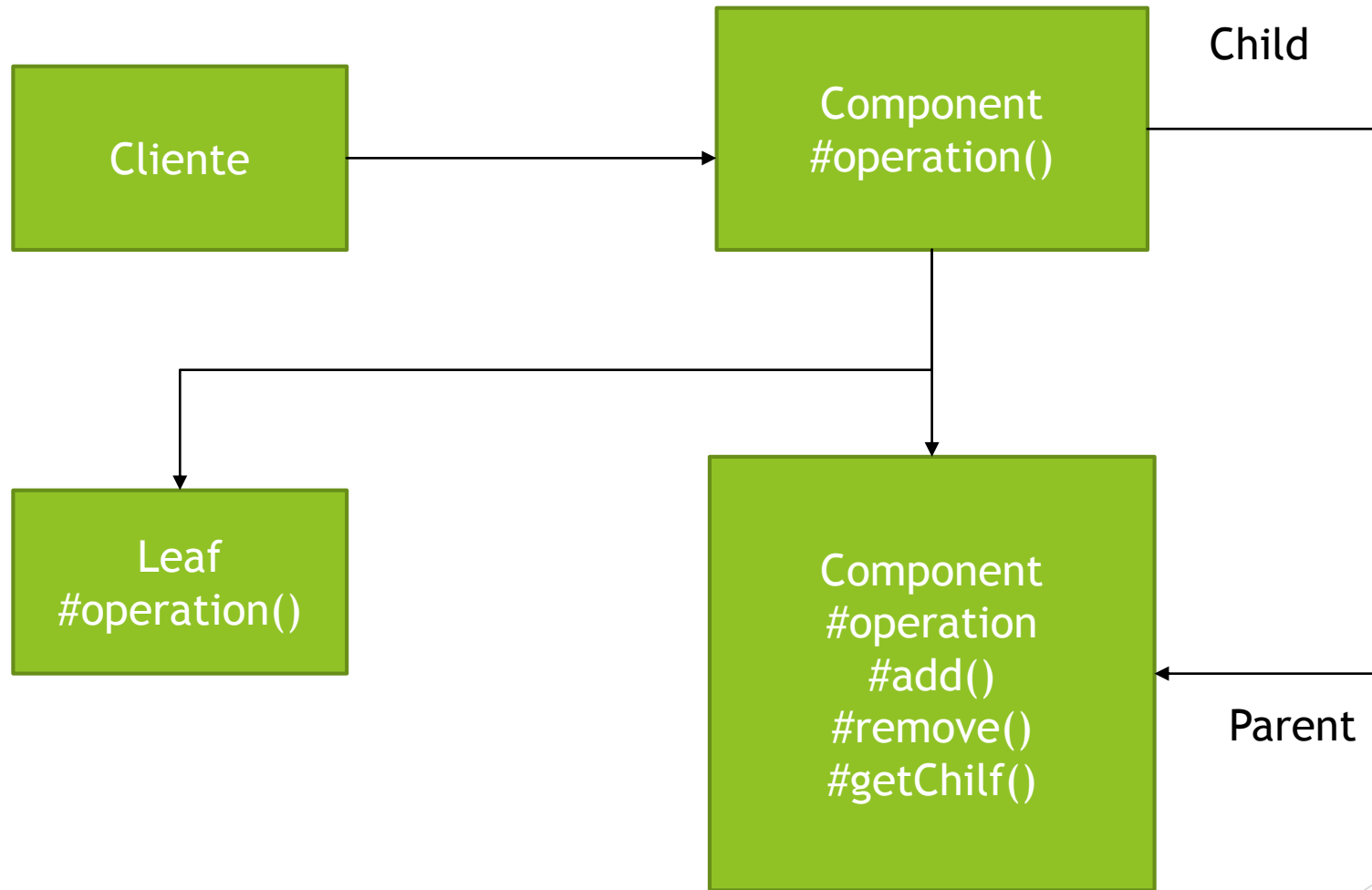
Real Life example

In an organization, It have general managers and under general managers, there can be managers and under managers there can be developers. Now you can set a tree structure and ask each node to perform common operation like `getSalary()`.



Composite design pattern treats each node in two ways:

- 1) **Composite** - Composite means it can have other objects below it.
- 2) **leaf** - leaf means it has no objects below it.



CODE...

<https://www.geeksforgeeks.org/composite-design-pattern/>

[https://es.wikipedia.org/wiki/Composite_\(patr3n_d e_diseo\)](https://es.wikipedia.org/wiki/Composite_(patr%C3%B3n_de_dise%C3%B1o))