PATRÓN DECORATOR

Laboratorio de Ingeniería de Software II

Presentado por:

Ledy Astudillo

Santiago Nieto Guaca

Profesor:

Ricardo Zambrano

Universidad del Cauca

Facultad de Ingeniería Electrónica y telecomunicaciones

Ingeniería de sistemas

Popayán, octubre 2023

1. Desarrolle la siguiente plantilla para el patrón Decorador:

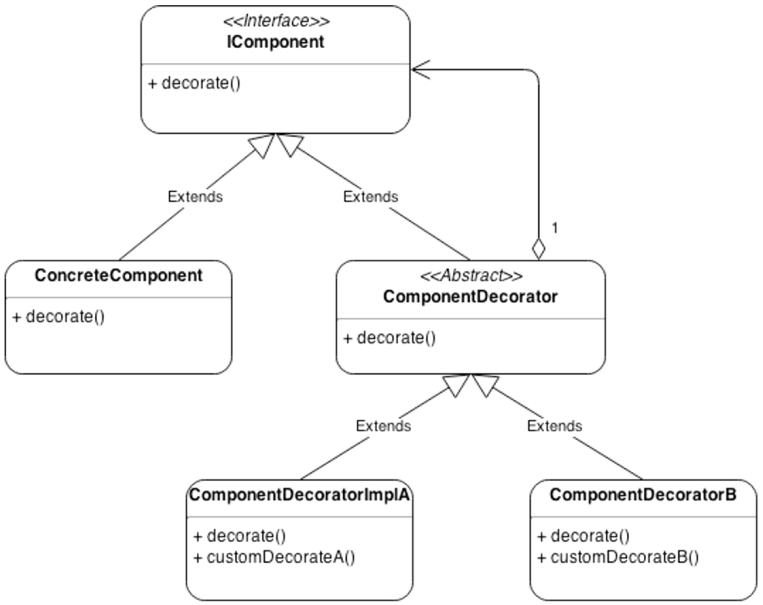
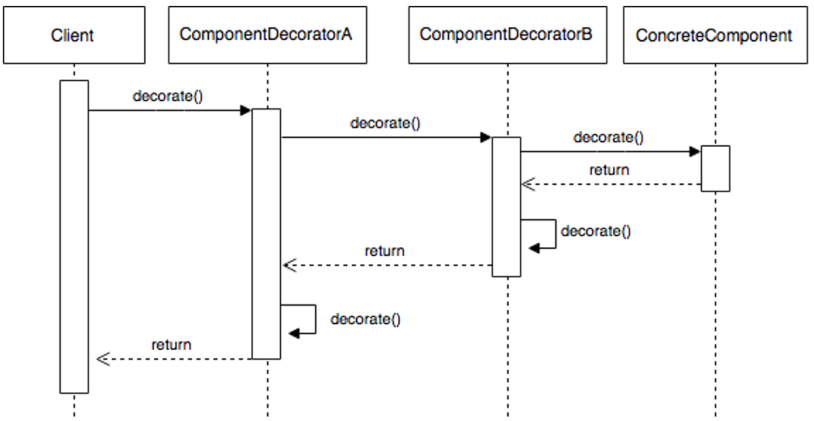
Patrón creacional: **Decorador**	
Intención	El patrón de diseño Decorador tiene la intención de agregar dinámicamente responsabilidades adicionales a un objeto. Proporciona una alternativa flexible a la herencia para extender la funcionalidad.
Problema que soluciona	Cuando se necesita agregar responsabilidades a objetos individuales, no a una clase entera, y cuando las extensiones de responsabilidad deben ser aplicables en tiempo de ejecución.
Solución propuesta	Implica un conjunto de clases decoradoras que se utilizan para envolver objetos concretos. Las clases decoradoras reflejan el tipo de objetos concretos que decoran. Heredan del objeto que decoran.
Diagrama de clases	

Diagrama de secuencia	
Participantes	• **Componente:** Define la interfaz para los objetos que pueden tener responsabilidades añadidas dinámicamente. • **Concreto Componente:** Define un objeto al que se pueden añadir responsabilidades adicionales. • **Decorador:** Mantiene una referencia a un objeto Componente y define una interfaz que cumple con la interfaz del Componente. • **Concreto Decorador:** Añade responsabilidades al Componente.
Aplicabilidad	1. Se necesita añadir responsabilidades a objetos individuales de manera dinámica y transparente, es decir, sin afectar a otros objetos. 2. Se necesita retirar las responsabilidades de un objeto. 3. La extensión mediante la herencia no es viable. En muchos casos, una cantidad excesiva de opciones independientes entre sí conduciría a una explosión de subclases para cubrir todas las combinaciones posibles.
Consecuencias	1. Más flexibilidad que la herencia estática. 2. Evita la carga de “clases con características”, es decir, clases altamente personalizadas al final del árbol de herencia. 3. Un decorador y su componente no son idénticos. 4. Muchos pequeños objetos pueden resultar difíciles de aprender y depurar.

REFERENCIAS

[1] “Patrones de Diseño (X): Patrones Estructurales - Decorator”. Programación en Castellano. Accedido el 16 de octubre de 2023. [En línea]. Disponible: https://tinyurl.com/67k7a5xh

[2] Reactive Programming - arquitectura y desarrollo de software. Accedido el 16 de octubre de 2023. [En línea]. Disponible: https://tinyurl.com/54ka9shn