UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERIA ELECTRÓNICA Y TELECOMUNICACIONES PROGRAMA DE INGENIERÍA DE SISTEMAS

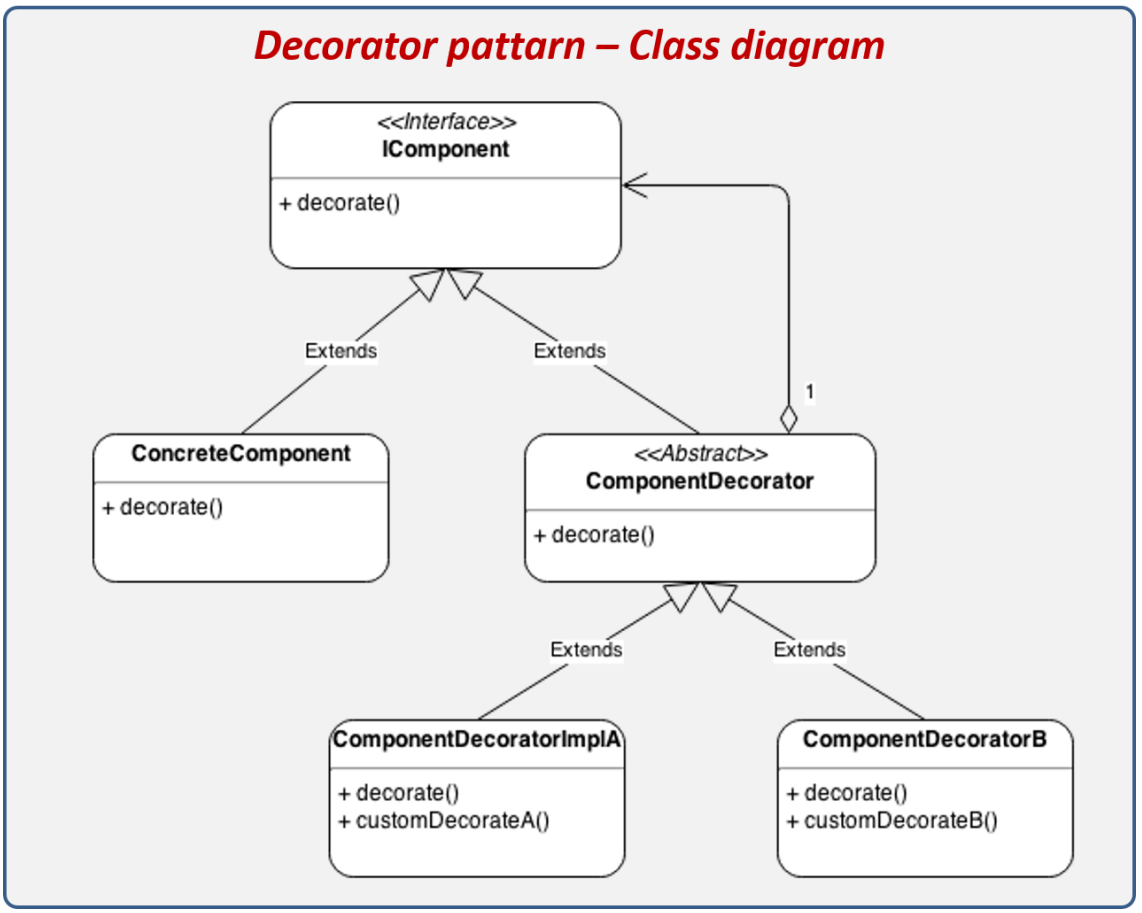
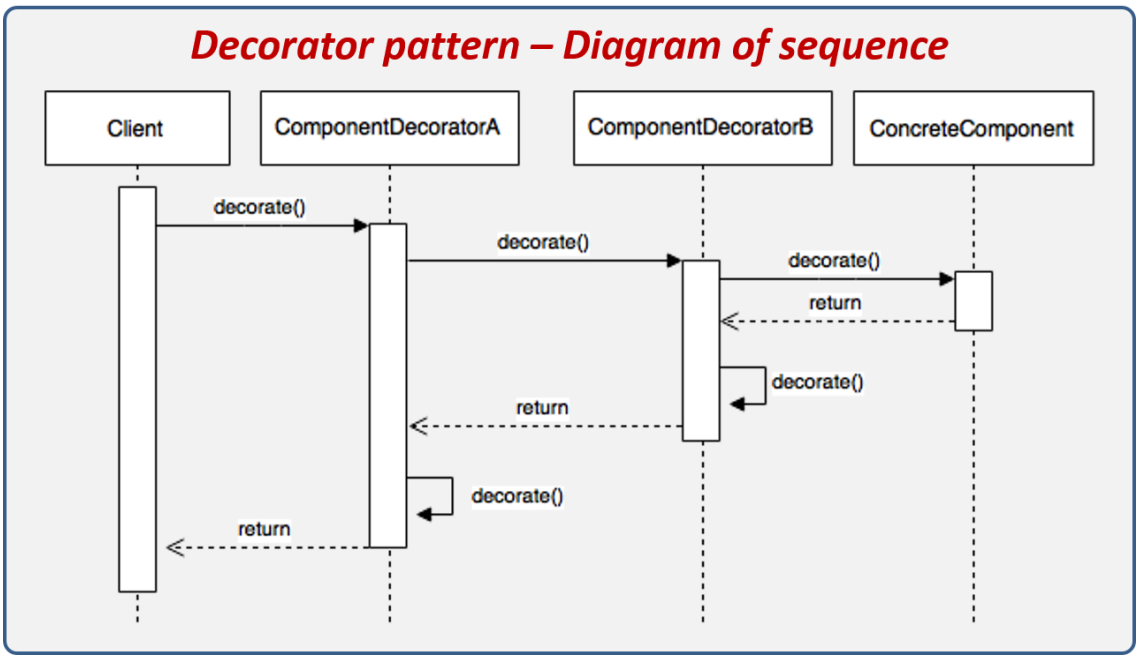
PRACTICA DE LABORATORIO No. 7
Definición del proyecto de trabajo

OBJETIVOS: Comprender e implementar el patrón estructural Decorator (Decorador).

ACTIVIDADES:

1. Desarrolle la siguiente plantilla para el patrón Decorador:

Patrón creacional: **Decorador**	
Intención	Consiste en añadir dinámicamente nuevas funcionalidades a los objetos, colocando estos objetos dentro de objetos encapsuladores especiales que contienen los comportamientos o funcionalidades
Problema que soluciona	Supongamos que una clase está encargada de realizar funciones generales, pero de alguna manera se espera que el programa evolucione y los objetos puedan implementar nuevas funcionalidades para casos específicos o de interés. Sin embargo, creando nuevas subclases especiales y metiendo todo dentro de una sola clase no es la solución ya que desbordaba el código. Y no se puede emplear la herencia debido a su comportamiento estático, ya que no se puede alterar la funcionalidad de un objeto existente durante el tiempo de ejecución.
Solución propuesta	La solución es el uso de la agregación o composición, de esta manera un objeto tiene una referencia a otro, permitiendo otorgar parte del trabajo. De este modo se puede modificar el comportamiento del objeto durante su tiempo de ejecución. Un objeto puede utilizar el comportamiento de varias clases con referencias a varios objetos, delegándoles todo tipo de tareas.

Diagrama de clases	
Diagrama de secuencia	
Participantes	IComponente (interfaz) , ComponenteConcreto(Clase concreta) ,ComponenteDecorador (Clase abstracta) , ComponeteDecoradorimplA y ComponeteDecoradorimplA (Clases que hereadan de ComponenteDecorador).

Aplicabilidad	Cuando es necesario asignar funcionalidades a los objetos en tiempo de ejecución, es buena idea utilizar el patrón decorador, pues permite estructurar la lógica en capas y componer objetos con varias combinaciones en esta lógica.
Consecuencias	• Es difícil de implementar, ya que su comportamiento no debe depender del orden de la pila de decoradores. • El código inicial no es tan legible. • El objeto core, estará oculto por lo cual acceder directamente puede ser un problema.

2. Desarrolle una implementación del patrón para la siguiente situación problemática:

La agencia de viajes "Limitless" ha decido contratarlos a ustedes para el montaje de sus sistema de configuración de planes y paquetes turísticos a la isla de Hawaii en la Polinesia. El sistema se debe configurar a partir de un paquete básico. Cada cliente tendrá dicho plan como mínimo y podrá decorarlo de acuerdo con sus gustos, preferencias y presupuesto.

El paquete base (U$ 7000 y cinco días) consta de:

- tiquetes aéreos
- hotel en habitación estándar
- alimentación
- vuelta a la isla
- recepción con lei hawaiano y camiseta de Millonarios (es el equipo favorito de los nativos)

Este plan puede ser decorado con mini paquetes de actividades que tienen diferentes costos.
Así por ejemplo, para quienes gustan de la historia y la aviación se tiene un paquete llamado Pearl Harbor (U$ 653 y 2 días adicionales) que permite visitar el museo del mismo nombre, el palacio Iolani, el museo Bishop y el USS Arizona Memorial.
A quienes les gusta la naturaleza se les ofrece el paquete Nature (U$ 720 y 5 días mas) que consta de visitas al Kualoa Ranch, el Maui Ocean Center y el Akaka Falls State Park.
Finalmente si usted es deportista extremo podrá comprar el paquete Amazing Hawaii (U$ 931 y 3 días), con el cual podrá escalar en el Waipio Valley, surfear en las playas Waikiki o Hanalei Bay y bucear en Hanauma Bay.
Ahora, si su plan es rumbear le recomendamos una discoteca para la noche final,

pero queremos que conozca Hawaii, no que el licor le haga perder la oportunidad de conocer esta maravilla.
El sistema debe permitir realizar la cotización y duración de un plan por persona y por familia.