

Práctica 04 - Tarea 06

Hernández Leyva Mirén Jessamyn
Ruiz López Jorge Antonio

1. Menciona los principios de diseño esenciales de los patrones Decorator, Proxy y Facade. Menciona una desventaja de cada patrón.

Decorator: Permite agregar determinado comportamiento a un objeto individual, dinámicamente, sin afectar el comportamiento de otros objetos de la misma clase, permite añadir y eliminar responsabilidades en tiempo de ejecución.

Desventajas:

- Puede haber problemas con la identidad de los objetos. Un decorador se comporta como un envoltorio transparente. Pero desde el punto de vista de la identidad de objetos, estos no son idénticos, por lo tanto no deberíamos apoyarnos en la identidad cuando estamos usando decoradores.
- Genera gran cantidad de objetos pequeños y parecidos

Proxy: Tiene como propósito proporcionar un intermediario de un objeto para controlar su acceso, creando una interfaz idéntica al objeto, con la cual el usuario pueda interactuar sin tocar el objeto real directamente, para que una vez que el cliente deje de usarla se pueda revisar que todo este bien con la interfaz y ahora si pasar los cambios al objeto real.

Desventajas:

- Copiar un objeto grande puede ser costoso.
- Se deben hacer tareas extra de mantenimiento al acceder al objeto.
- Introduce un nivel de induración adicional.

Facade: Estructura un entorno de programación y reduce su complejidad con la división en subsistemas, minimizando las comunicaciones y dependencias entre estos, creando una "fachada" en la cual los clientes pueden interactuar de

manera más sencilla sin tocar realmente las clases, ya que no necesitan conocerlas.

Desventaja:

Si varios clientes necesitan acceder a subconjuntos diferentes de la funcionalidad que provee el sistema, podrían acabar usando sólo una pequeña parte de la fachada, por lo que sería conveniente utilizar varias fachadas más específicas en lugar de una única global.