

Patrón Decorador

Nombre: Sebastián Toledo

Materia: Programación Aplicada

Docente: Ing. Diego Quisi

Patrón Decorador

- Es un patrón de tipo estructural.
- Su objetivo es añadir funcionalidades a una clase u objeto ya existente en tiempo de ejecución, evitando las subclases.
- Utiliza las clases envoltorio (wrapper) para añadir las funcionalidades a una clase que ya existe (wrappee)
- Utiliza las relaciones de agregación y composición.

Implementación

- Se crea una interfaz con los métodos básicos
- Se crean dos clases, una clase concreta, y una clase abstracta que será la base para las clases decoradoras, estas dos clases deben implementar la interfaz.
- En la clase abstracta tendremos un atributo que será la clase envuelta (wrappee), y esta será heredad por las demás clases decoradoras.
- Se crean clases concretas que son las decoradoras, están implementarán funcionalidades sobre la clase original que se ejecutarán antes o después de los métodos de la clase original.

Ventajas

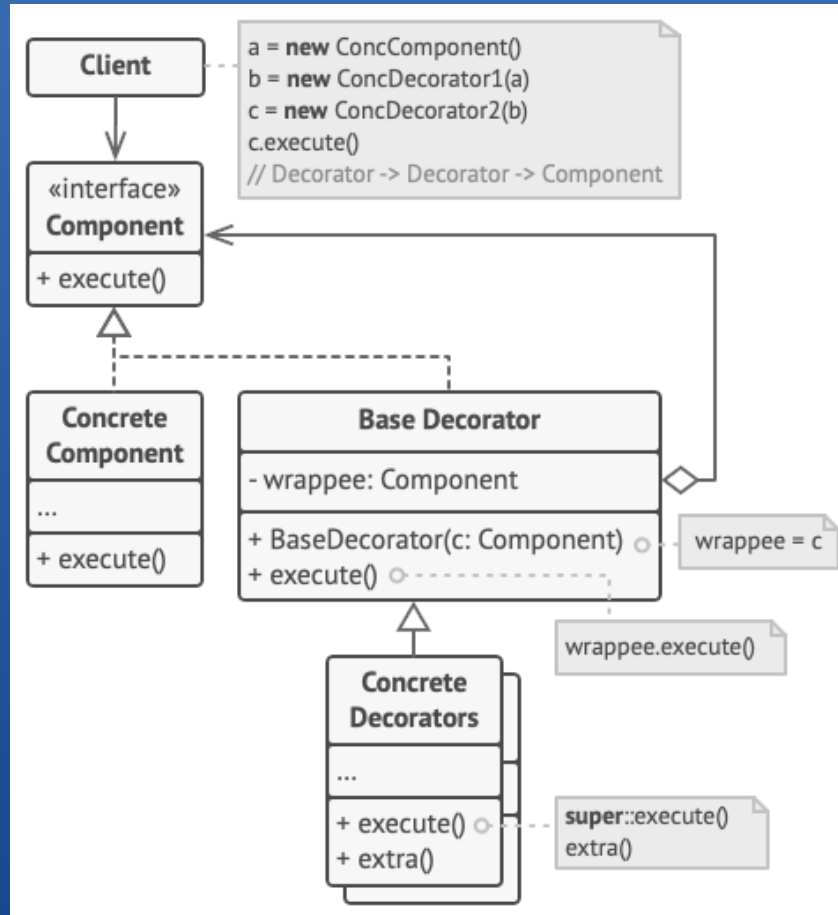
Ventajas

- Permite añadir funcionalidades a las clases en tiempo de ejecución.
- Se puede utilizar como una alternativa a las subclases. Estas añaden funcionalidades en tiempo de compilación, sin embargo, se debe tener una subclase para cada tipo de objeto que se podría dar en el programa.
- Ofrece flexibilidad en las funcionalidades, es decir, se las puede ir añadiendo a medida que las necesitemos.

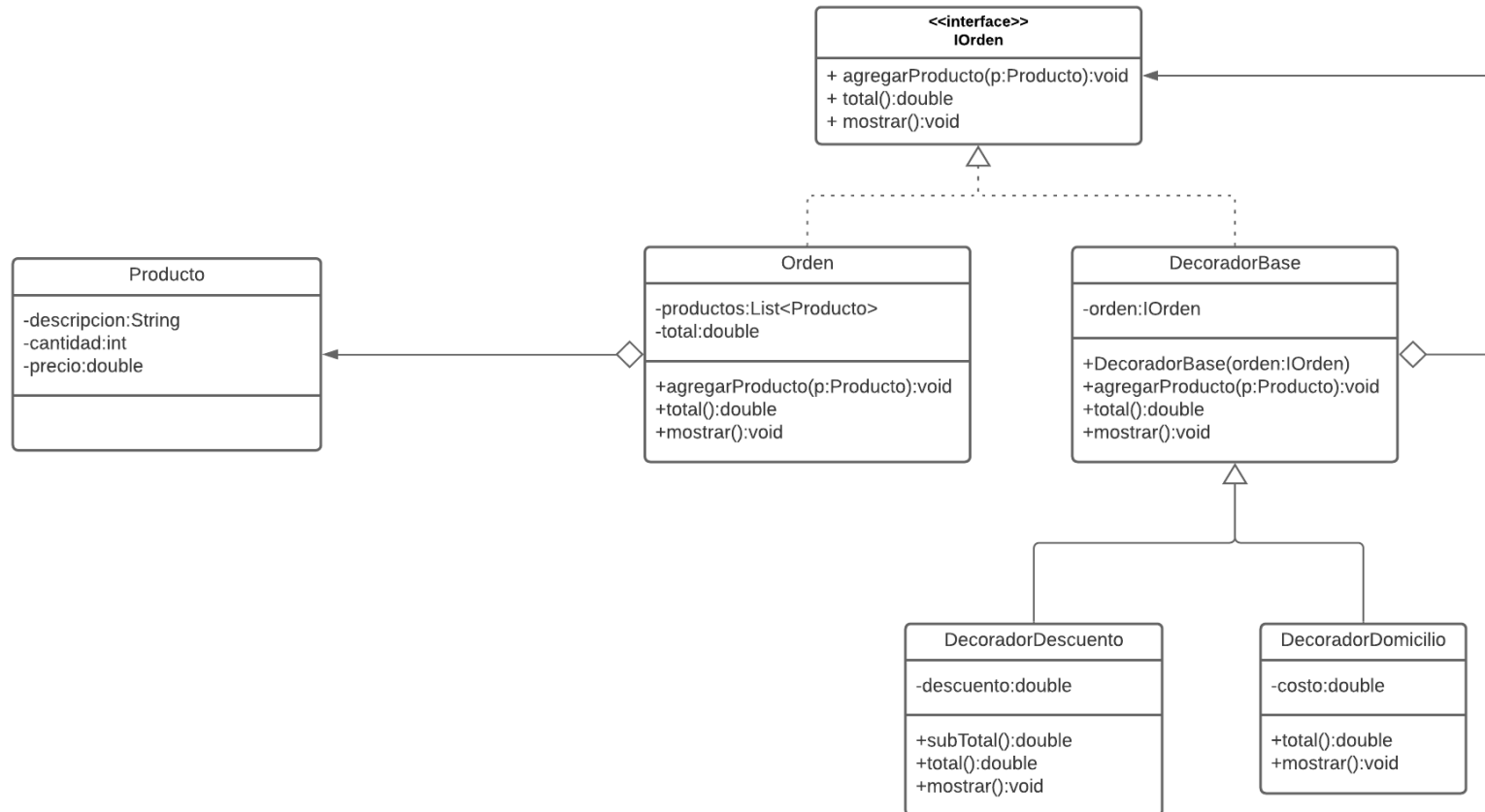
Desventajas

- Desventajas
- Se puede llegar a confundir cuando se tienen varios objetos con diferentes decoradores cada uno.
- El utilizar instancias de un mismo objeto nos puede llegar a agrandar el código ya que se necesitan crear e instanciar varios objetos con varios decoradores.
- El añadir varios decoradores a un objeto puede hacernos perder conocimiento de cuantos decoradores hemos añadido a un solo objeto.

Patrón Decorador (Modelo)



Patrón Decorador (Ejemplo)



Resultado Ejemplo

Orden

Descripción: mango Cantidad: 7 Precio: 0.5 CT: 3.5

Descripción: manzana Cantidad: 5 Precio: 0.1 CT: 0.5

Descripción: naranja Cantidad: 10 Precio: 0.25 CT: 2.5

Total: 6.5

Orden

Descripción: mango Cantidad: 7 Precio: 0.5 CT: 3.5

Descripción: manzana Cantidad: 5 Precio: 0.1 CT: 0.5

Descripción: naranja Cantidad: 10 Precio: 0.25 CT: 2.5

Total: 6.5

Descuento: 40.0%

Costo Final: 3.9

Orden

Descripción: mango Cantidad: 7 Precio: 0.5 CT: 3.5

Descripción: manzana Cantidad: 5 Precio: 0.1 CT: 0.5

Descripción: naranja Cantidad: 10 Precio: 0.25 CT: 2.5

Total: 6.5

Descuento: 40.0%

Costo Final: 3.9

Costo de entrega a domicilio: \$2.0

Precio con costo de envío: 5.9

BUILD SUCCESSFUL (total time: 0 seconds)

Referencias

- B. (2019, 11 septiembre). The Decorator Pattern in Java. Baeldung.
<https://www.baeldung.com/java-decorator-pattern>
- Design Patterns - Decorator Pattern - Tutorialspoint. (s. f.).
Tutorialspoint.
https://www.tutorialspoint.com/design_pattern/decorator_pattern.htm
- GeeksforGeeks. (2018, 12 noviembre). The Decorator Pattern | Set 2 (Introduction and Design).
<https://www.geeksforgeeks.org/the-decorator-pattern-set-2-introduction-and-design/?ref=rp>
- Refactoring.Guru. (2020, 1 enero). Decorator.
<https://refactoring.guru/design-patterns/decorator>