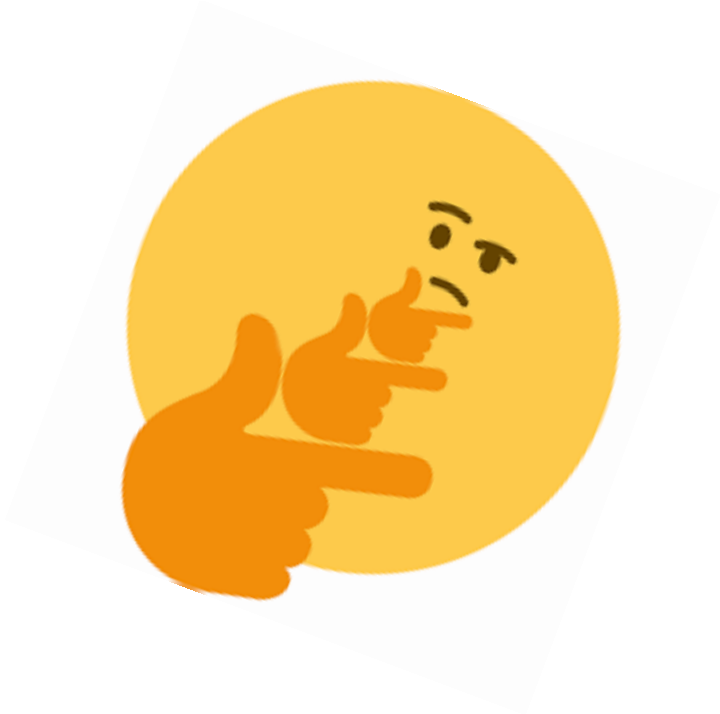
Design Patterns: Decorator

Intencion

Permite modificar, retirar o agregar responsabilidades/funcionalidad a un objeto/clase existente, evitando heredar sucesivas clases para incorporar la nueva funcionalidad.

Otros nombres

Decorator es comúnmente conocido como Wrapper.



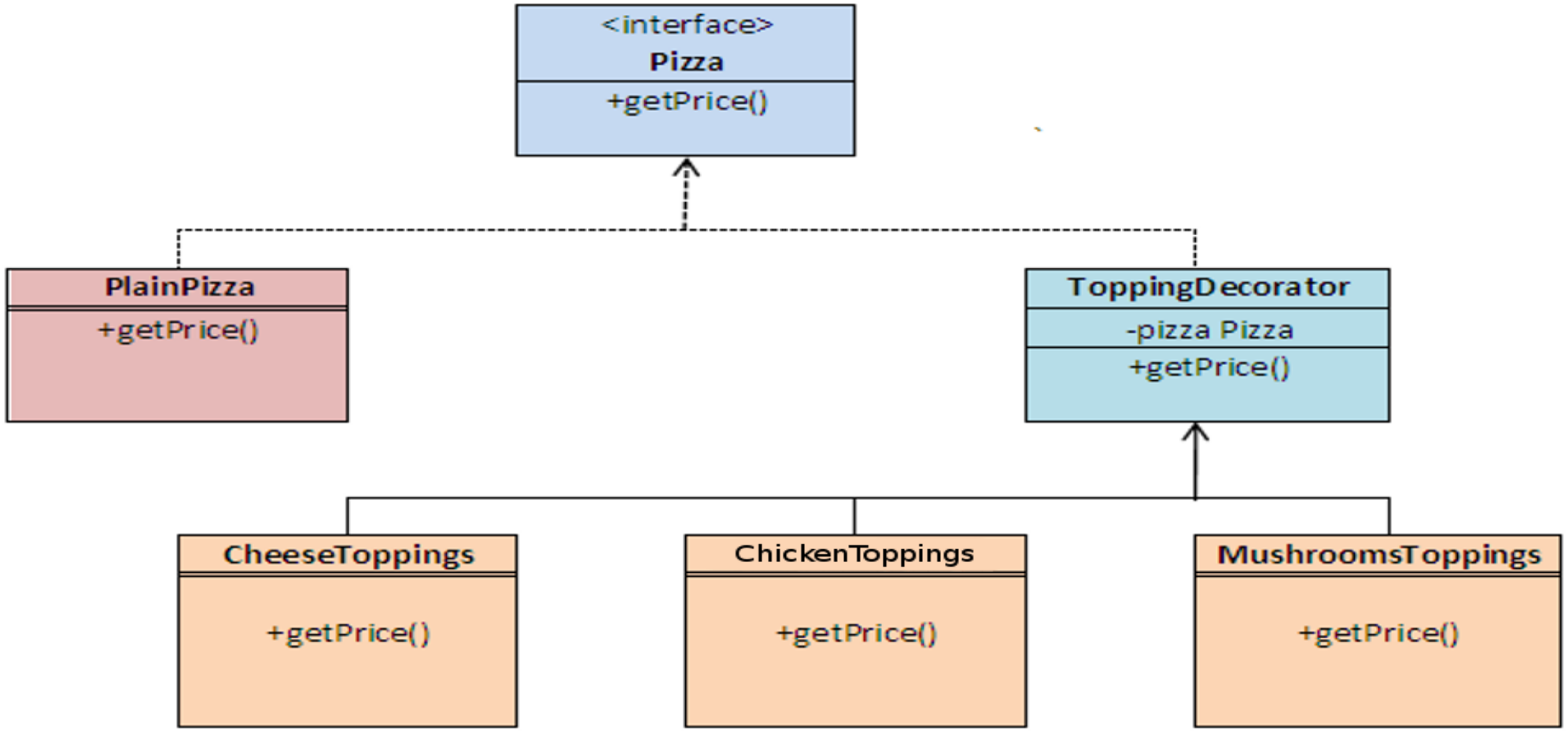
Motivacion

A veces se quiere añadir funcionalidad a un objeto concreto, no a una clase entera.

Aplicacion

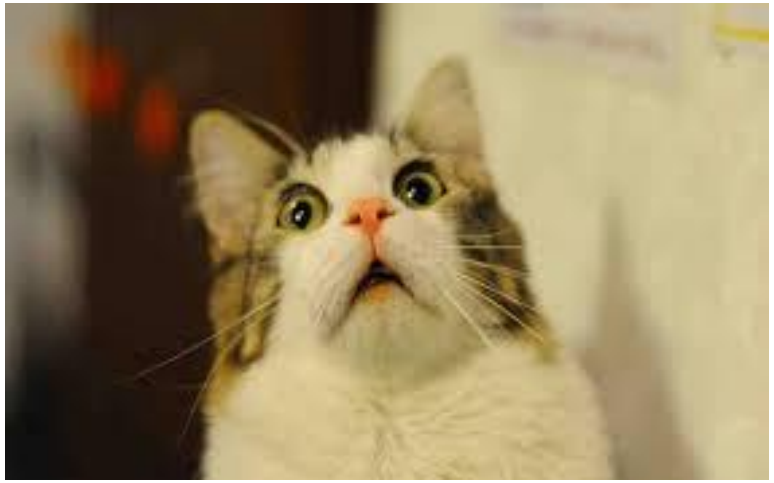
- Cuando se necesite añadir/quitar funcionalidades a una clase de forma dinámica(ejecución script), evitando las jerarquías de clases que se tienen que construir.

Estructura


<interface>
Pizza
+getPrice()
PlainPizza
+getPrice()
ToppingDecorator
-pizza Pizza
+getPrice()
CheeseToppings
+getPrice()
ChickenToppings
+getPrice()
MushroomsToppings
+getPrice()

Participantes/colaboraciones

- **Component:** Clase abstracta que define los métodos que tendrá en común los componentes.
- **ConcreteComponent:** Clase la cual extiende del componente e implementa sus métodos.
- Decorator:mantiene una referencia a un objeto Component y define una interfaz que se ajusta a la interfaz de Component.
- **ConcreteDecorator:** Extiende de la clase decorador y tiene los cambios necesarios que impactan en el componente.



Consecuencias

Pros:

- Más flexibilidad que con la herencia.
- Evita que las clases más altas en la jerarquía estén demasiado cargadas de funcionalidad y sean complejas

Contras:

- Provoca la creación de muchos objetos pequeños parecidos y encadenados, complicando la depuración

Implementacion

- Se puede omitir la clase *Decorator* si solo se va a definir una sola responsabilidad.

Patrones relacionados

- Adapter
- Strategy

Refernecias

https://yos.io/2013/07/05/decorator-pattern/
https://www.thecodeship.com/patterns/guide-to-python-function-decorators/
http://codejavu.blogspot.mx/2013/07/ejemplo-patron-de-diseno-decorator.html
https://auraham.wordpress.com/2013/05/07/decoradores-en-python/https://dzone.com/articles/understanding-python

https://stackoverflow.com/questions/1549743/when-to-use-the-decorator-pattern