

Práctica 3

David Pérez Jacome
Etzael Iván Sosa Hedding
Modelado y Programación 2020-2
Grupo 7031

27 de abril de 2020

1. Ejercicios

1. Menciona los principios de diseño esenciales de los patrones Decorator y Adapter. Menciona una desventaja de cada uno.

Decorator: Este patrón de diseño nos permite añadir funciones o características a un objetos, es decir, modificarlo dinámicamente. Esto lo hacemos mediante un objeto base que podemos ir .envolviendo con lo que necesitemos. Se podría decir que envolvemos un objeto en otro para añadir cualidades al original.

Una desventaja de este diseño es que si no tenemos herencia, no es una opción viable el implementarla. Por otra parte, nos genera gran cantidad de clases y objetos más pequeños que pueden dificultar el uso de referencias puesto que después de añadir suficientes envoltentes, puede volverse confuso.

Adapter: Se encarga de adaptar o transformar un objeto en otro con comportamiento en común pero no necesariamente el mismo. Dicho de otra forma, permite que dos o más clases incompatibles trabajen juntas, aparentando ser el mismo objeto, mediante una interfaz que implementa una clase mediadora. Esta última extendida por sus hijas se encarga de darles forma a todas para que sean aceptadas por la interfaz que, a su vez es implementada directamente por objetos de la forma correcta para la interfaz.

Una desventaja es que se puede dificultar la comprensión de la diferencia entre objetos y el uso de las clases, ya que sólo se aparenta un objeto distinto pero sigue teniendo su comportamiento original.

2. Para la compilación de la práctica:

ant

3. Para la ejecución de la práctica:

java -jar practica3.jar

4. Descripción y observaciones:

Para esta práctica no encontramos muchas complicaciones. Empezamos definiendo nuestros diagramas y discutiéndolos logramos encontrar una estructura que funciona y nos define ambos patrones.

Como se puede observar, creamos un objeto Baguette y elegimos el tipo de pan (pues hay 2) y mandamos un menú con todas las opciones de condimentos para el emparedado. Una vez elegido creamos una instancia de ese condimento y envolvemos el baguette con él, aumentando el precio y la descripción. En los diagramas se ve como una clase abstracta que es extendida por CondimentoBaguette que se encarga de hacer el baguette y también extendida por los tipos de pan y condimentos.

Con respecto a las pizzas, extiende a Baguette y PizzaAdapter se encarga de aparentar ser un baguette con precio y descripción pero en realidad crea una Pizza de cualquiera de las 5 variedades.

Algunos posibles problemas que podemos hayar es que al momento de ordenar una pizza e imprimir el ticket se finaliza el programa y también la clase abstracta Baguette podría ser una interfaz.