

Modelado y programación  
Sección teórica: Practica 2

1- Principios de diseño: Decorator

Decorator se guía por principios como:

- Las clases deben estar abiertas a extensión, pero cerradas a modificación.
- Identificar los aspectos de que varían encapsulándolos y separarlos de los que no cambian.
- Programa a una interfaz, no a una implementación.
- Favorecer la composición sobre la herencia.

Principios de diseño: Adapter

- Programa a una interfaz, no a una implementación.

Desventajas de: Decorator

- Se incrementa el número de clases que el programa debe mantener.

Desventajas de: Adapter

- La clase que se quiera adaptar probablemente se vaya ver limitada en su funcionalidad debido a la interfaz.

README:

- En esta practica realizamos la implementación de Decorator y Adapter en un ejemplo de pizzas y baguettes.
- En la practica, para seleccionar entre pizza y baguette se utiliza p y b respectivamente.
- También para la seleccionar tanto las pizzas, como tipo de pan e ingredientes utilice números, para dejar de agregar ingredientes se utiliza el número 0.
- La implementación de la pizzeria la hice muy básica debido a que no tuve mucho tiempo de hacer la practica y pues el objetivo de la practica no era ese.
- Perdón por la falta de documentación, realmente ya no me alcanzo el tiempo.
- La practica se ejecuta ingresando a la carpeta src desde la terminal y utilizando el comando `javac *.java` y luego `java Main` (Solo lo pongo por si acaso).