

Práctica 01

Patrones de Diseño

Patrón Observer:

Principios de diseño:

- Define dependencias de “uno a muchos”: cuando un objeto cambia de estado, sus dependientes son notificados y se actualizan automáticamente
- Al estar débilmente acoplados, los sistemas creados con el patrón Observer son más flexibles y fáciles de cambiar ya que minimizan la interdependencia entre los objetos.

Desventajas:

- Una desventaja del patrón Observer es que puede causar *memory leaks* ya que requiere que se registren y de-registren explícitamente los observadores.
- Otra desventaja es que el sujeto puede mandar actualizaciones que no sean de importancia al observador

Patrón Strategy:

Principios de diseño:

- Identificar los aspectos del código que varían y encapsularlos para que no afecten a los demás.
- Programa interfaces/supertipos: para que las clases que implementan el comportamiento en particular no sepan los detalles de su implementación.

- Favorecer composición sobre herencia: le da más flexibilidad a los sistemas ya que a parte de encapsular una serie de algoritmos, nos permite cambiar el comportamiento en tiempo de ejecución
- Programar para las interfaces/clases abstractas, no para la implementación

Desventajas:

- Se incrementa el número de objetos y clases que se crean.