

INFORMACIÓN PATRONES DE DISEÑO

¿Qué son?, ventajas y desventajas, patrones

| | |
|---------------------------------------|----------|
| ¿Qué son? | 1 |
| Ventajas y desventajas | 2 |
| Ventajas | 2 |
| Desventajas | 2 |
| Principales patrones de diseño | 2 |
| Patrones creacionales | 3 |
| Prototype | 3 |
| Singleton | 3 |
| Patrones estructurales | 3 |
| Composite | 3 |
| Proxy | 3 |
| Patrones de comportamiento | 4 |
| Iterator | 4 |
| Strategy | 4 |
| Bibliografía | 4 |

¿Qué son?

Los patrones de diseño son unas técnicas para resolver problemas comunes en el desarrollo de software.

Un patrón de diseño es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

Ventajas y desventajas

Ventajas

- **Ahorro de tiempo:** Si para cada problema al que nos enfrentamos, intentamos diseñar una solución nueva, perderíamos demasiado tiempo. Por eso es que el uso de los mismos nos ahorra el tiempo de pensar en una nueva solución cada vez, siempre y cuando tengamos en claro que patrón usar.
- **Establecen un lenguaje común:** A la hora de la solución elegida ante tu equipo de trabajo, los patrones de diseño son muy útiles, ya que facilitan el entendimiento y ayudan a descubrir si es la solución adecuada al problema o no. Pueden ser útiles a la hora de explicarle a un cliente el funcionamiento interno de su sistema.

Desventajas

- **Pueden lograrse soluciones ineficientes:** El abuso del uso de patrones de diseño puede llevar a soluciones en las que aparece la duplicación de código, o soluciones que cambian totalmente el contexto del problema a resolver. Para el uso correcto de varios patrones a la vez, se requiere mucha investigación.

Principales patrones de diseño



Patrones creacionales

Los patrones creacionales proporcionan diversos mecanismos de creación de objetos, que aumentan la flexibilidad y la reutilización del código existente de una manera adecuada a la situación. Esto le da al programa más flexibilidad para decidir qué objetos deben crearse para un caso de uso dado.

Prototype

Permite copiar objetos existentes sin hacer que su código dependa de sus clases. Se utiliza para restringir las operaciones de memoria o base de datos manteniendo la modificación al mínimo utilizando copias de objetos.

Singleton

Este patrón de diseño restringe la creación de instancias de una clase a un único objeto.

Patrones estructurales

Facilitan soluciones y estándares eficientes con respecto a las composiciones de clase y las estructuras de objetos. El concepto de herencia se utiliza para componer interfaces y definir formas de componer objetos para obtener nuevas funcionalidades.

Composite

Se usa para agrupar objetos como un solo objeto. Permite componer objetos en estructuras de árbol y luego trabajar con estas estructuras como si fueran objetos individuales.

Proxy

Se utiliza para crear objetos que pueden representar funciones de otras clases u objetos y la interfaz se utiliza para acceder a estas funcionalidades.

Patrones de comportamiento

El patrón de comportamiento se ocupa de la comunicación entre objetos de clase. Se utilizan para detectar la presencia de patrones de comunicación ya presentes y pueden manipular estos patrones.

Estos patrones de diseño están específicamente relacionados con la comunicación entre objetos.

Iterator

Su utilidad es proporcionar acceso secuencial a un número de elementos presentes dentro de un objeto de colección sin realizar ningún intercambio de información relevante.

Strategy

Permite definir una familia de algoritmos, poner cada uno de ellos en una clase separada y hacer que sus objetos sean intercambiables.

Bibliografía

https://es.wikipedia.org/wiki/Patr%C3%B3n_de_dise%C3%B1o

<https://profile.es/blog/patrones-de-diseno-de-software/>

<https://dev.to/gelopfalcon/los-7-patrones-de-diseno-de-software-mas-importantes-28l2>

<https://www.studocu.com/es-ar/document/universidad-nacional-de-rio-cuarto/analisis-y-diseno-de-sistemas/patrones-de-diseno/4267128>