

Patrones de Diseño

Universidad Tecnológica Metropolitana
Facultad de Ingeniería
Ingeniería en Software

12 de mayo de 2014

1. Introducción

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Un patrón de diseño resulta ser 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. En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares.

Debemos tener presente los siguientes elementos de un patrón:

1. Nombre.
2. El problema (cuando aplicar un patrón).
3. La solución (descripción abstracta del problema).
4. Las consecuencias (costos y beneficios).

2. Patrones de Diseño

Los patrones de diseño pretenden:

1. Proporcionar catálogos de elementos reusables en el diseño de sistemas software. Proporcionar catálogos de elementos reusables en el diseño de sistemas software.
2. Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
3. Formalizar un vocabulario común entre diseñadores.
4. Estandarizar el modo en que se realiza el diseño.
5. Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente.

Asimismo, no pretenden:

1. Imponer ciertas alternativas de diseño frente a otras.
2. Eliminar la creatividad inherente al proceso de diseño.
3. No es obligatorio utilizar los patrones, solo es aconsejable en el caso de tener el mismo problema o similar que soluciona el patrón, siempre teniendo en cuenta que en un caso particular puede no ser aplicable. "Abusar o forzar el uso de los patrones puede ser un error".

Categorías de patrones:

1. Según la escala o nivel de abstracción.
2. Patrones de arquitectura: Aquéllos que expresan un esquema organizativo estructural fundamental para sistemas de software.
3. Patrones de diseño: Aquéllos que expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas de software.
4. Dialectos: Patrones de bajo nivel específicos para un lenguaje de programación o entorno concreto.

3. Ejemplo Patrón Orientado a Objeto

3.1. Patrón de Estructura Proxy

Es un patrón estructural que tiene como propósito proporcionar un intermediario de un objeto para controlar su acceso.

El uso de un proxy introduce un nivel de in-dirección adicional con diferentes usos:

1. Un proxy remoto oculta el hecho de que un objeto reside en otro espacio de direcciones.
2. Un proxy virtual puede realizar optimizaciones, como la creación de objetos bajo demanda.
3. El proxy de protección y las referencias inteligentes permiten realizar diversas tareas de mantenimiento adicionales al acceder a un objeto.

Además, su uso también permite realizar una optimización COW (copy-on-write) , puesto que copiar un objeto grande puede ser costoso, y si la copia no se modifica, no es necesario incurrir en dicho gasto. Además el sujeto mantiene un número de referencias, y sólo cuando se realiza una operación que modifica el objeto, éste se copia. Es útil por tanto para retrasar la replicación de un objeto hasta que cambia.

3.2. Ejemplos comunes de la aplicación del patrón proxy

Consideremos un editor que puede incluir objetos gráficos dentro de un documento. Se requiere que la apertura de un documento sea rápida, mientras que la creación de algunos objetos (imágenes de gran tamaño) es costosa. En este caso no es necesario crear todos los objetos con imágenes nada más abrir el documento porque no todos los objetos son visibles. Interesa por tanto retrasar el coste de crear e inicializar un objeto hasta que es realmente necesario (por ejemplo, no abrir las imágenes de un documento hasta que no son visibles). La solución que se plantea para ello es la de cargar las imágenes bajo demanda. Pero, ¿Cómo cargar las imágenes bajo demanda sin complicar el resto del editor? La respuesta es utilizar un objeto proxy. Dicho objeto se comporta como una imagen normal y es el responsable de cargar la imagen bajo demanda.

4. Anexo

Link de Repositorio Github

1. https://github.com/victoriaMB/Patrones_de_Disenio