

Programación web con PHP



Patrones de diseño

Temario

¿Qué es un patrón?.....	4
Patrones de diseño.....	4
Clasificación de patrones.....	5
Creacionales.....	5
Estructurales.....	5
De comportamiento.....	5
Patrones GoF.....	5
Elementos de un patrón.....	6
Nombre del patrón.....	6
Problema.....	6
Solución.....	6
Consecuencias.....	6

¿Qué es un patrón?

Un patrón una solución probada que se puede aplicar con a un determinado tipo de problemas que aparecen reiteradamente en el desarrollo de sistemas software.

Otras definiciones de patrón:

- “Un patrón es un pedazo de información con nombre, instructivo y significativo, que captura la esencia de una familia exitosa y completa de soluciones a un problema recurrente en un contexto dado” Brad Appleton
- “Cada patrón es una regla de tres partes, la cual expresa una relación entre un contexto dado, un conjunto de fuerzas que ocurren repetitivamente en ese contexto y cierta configuración de software que permite a esas fuerzas resolverse por si mismas” Richard Gabriel

Existen diferentes tipo de patrones dependiendo del nivel de abstracción de la solución y de la etapa de desarrollo a la cual se aplican, entre ellos tenemos:

- Patrones de arquitectura
- Patrones de diseño
- Patrones de Idiomas
- Patrones de análisis
- Patrones para ambientes distribuidos
- Patrones de negocios
- Patrones de procesos y organizacionales

Para que una solución se considere un patrón debe tener ciertas características características:

- Solucionar un problema
- Debe ser comprobado
- Debe ser reutilizable

Patrones de diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre si cuyo propósito es resolver un problema de diseño.

Cada uno de los patrones de diseño es una solución concreta a un problema específico, que puede ser aplicado a dificultades que aparecen frecuentemente en el desarrollo orientado a objetos.

Explican la manera de resolver un problema utilizando un número pequeño de clases relacionadas entre si.

Los patrones facilitan la reutilización de clases y la programación de clases reutilizables.

Por otro lado los patrones ya implementados son difíciles de reutilizar, una vez implementado es difícil reconocer un patrón en nuestro código y aunque son soluciones genéricas sus implementaciones suelen ser mas concretas y por lo tanto difíciles de reutilizar.

Aunque suelen generar un código mas ordenado la utilización de patrones necesitan mas tiempo de implementación ya que se generan mas clases y relaciones,

Clasificación de patrones

Estos patrones pueden clasificarse según su propósito de la siguiente manera:

Creacionales

Tratan del proceso de creación de objetos.

Se utilizan para controlar la instanciación de las clases.

Estructurales

Tratan de la composición de clases y objetos.

Se utilizan para crear clases u objetos que están incluidos dentro de estructuras mas complejas.

De comportamiento

Tratan de la interacción y responsabilidades de las clases y objetos.

Se utilizan para controlar como interactúan las clases entre ellas.

Patrones GoF

Se conoce como patrones GoF a los patrones descritos en el libro “Design Patterns: Elements of Reusable Object-Oriented Software” escrito en 1994 por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides conocidos como “Gang of Four”.

Este libro tuvo un gran impacto en la ingeniería de software y hoy en día se sigue editando estando ya en su trigésima novena edición.

	Creacionales	Estructurales	Comportamiento
Clase	Factory Method	Adapter	Interpreter Template Method
Objeto	Abstract Factory Builder Prototype Singleton	Adapter Bridge Composite Decorator Facade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Memento Observer State Strategy Visitor

Elementos de un patrón

Existen diferentes maneras de expresar los elementos de un patrón que dependen del autor y del nivel de abstracción.

Algunos de estos elementos son:

Nombre del patrón

Nombre corto y sencillo que ayude a ver a que problema se aplica y sus soluciones.

Problema

Describe las circunstancias y el contexto de utilización del patrón

Solución

Describe los elementos necesarios para el diseño de la solución explicado de manera genérica sus relaciones, colaboraciones y responsabilidades utilizando interfaces, objetos y clases

Consecuencias

Describen como afectaría la implementación de este patrón en el diseño. Se tienen en cuenta características como flexibilidad, extendibilidad y portabilidad.