

Introducción a los patrones de diseño

Un patrón de diseño es la solución a un problema de diseño, el cual debe haber comprobado su efectividad resolviendo problemas similares en el pasado, también tiene que ser reutilizable, por lo que se deben poder usar para resolver problemas parecidos en contextos diferentes.

Un poco de historia en el libro de Patter Language (1979) por Christopher Alexander, en este libro plantea en tener:

- Un vocabulario (nombre del patrón que amplía el vocabulario)
- Gramática (descripción del problema en términos simples)
- Sintaxis (describe el problema utilizado en el vocabulario previamente definido)

En 1994, sale un libro llamado Design Patterns, que cambiaría para siempre el concepto de patrones de diseño.

LA IMPORTANCIA DE LOS PATRONES DE DISEÑO

- Demuestra las madures de un programador de software}
- Evita reinventar la rueda
- Agiliza el desarrollo de software
- Se basa en las mejores particas de programación
- Permite utilizar un vocabulario común
- Proporcionar un catalogo de soluciones probadas de diseño para problemas comunes conocidos.
- Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente
- Crear u lenguaje estándar entre los desarrolladores
- Facilita el aprendizaje a nuevas generaciones de programadores

QUE NO SE BUSCA CON LOS PATRONES DE DISEÑO

- Imponer ciertas alternativas de diseño frente a otras
- Imponer la solución definitiva a un problema de diseño

- Eliminar la creatividad inherente al proceso de diseño

PATRONES DE DISEÑO Y AL EVOLUCION PROFESIONAL

- SOLUTION ARCHITECT: Habla de proceso, componentes e integraciones
- AQUITECTO: Habla de patrones y componentes
- DEVELOPER SR: Mezcla patrones con código
- DEVELOPER JR: Habla en código

TIPOS DE PATRONES DE DISEÑO

Patrones:

- PATRONES DE DISEÑO: Tienen un contexto más pequeño, pues se concentran en forma en que los objetos se crean, estructuran o interactúan con el resto.
 - CREACIONALES: Controlan la forma en que los objetos son creados.
 - ESTRUCTURALES: Define a la forma en que las clases deben estructurarse
 - COMPORTAMIENTO: Define la forma en que los objetos deben de comportarse en Runtime.
- PATRONES ARQUITECTONICOS: Afecta la forma de trabajar del todo el componente, impone restricciones, así como la forma con que se comunican con otros componentes.
 - INTEGRACION
 - SEGURIDAD
 - WEB

PRINCIPALES PATRONES DE DISEÑO

CREACIONALES

- FACTORY METHOD
- ABSTRACT FACTORY
- SINGLETON
- BUILDER
- PROTOTYPE
- OBJECT POOL

ESTRUCTURALES

- ADAPTER
- BRIDGE
- COMPOSITE
- DECORATOR
- AFCADE
- FLYWEIGHT
- PROXY

COMPORTAMIENTO

- ITERATOR
- COMMANDO
- OBSERVER
- TEMPLATE METHOD
- STRATEGY
- CHAIN OF RESPONSABILITY
- INTERPRETER
- MADIATOR
- MEMENTO
- STATE
- VISITOR

IMPLEMENTACION

Se necesita:

POO

1.Encapsulamiento

2.Abtraccion

3.Herencia

4.Polimorfismo

5.UML

