

TEMA 2 - DESCRIBIR LOS PATRONES

1. Qué son necesarios para describir un patrón?

- La plantilla de descripción de patrones

El nombre y su clasificación:

- El nombre: es la manera más corta y clara de representar el patrón.
- Clasificación: para clasificar el patrón según su propósito y su ámbito



Propósito: ¿Qué hace el patrón? ¿Qué es su razón y su intención? ¿Qué tema y problema concreto trata?

También conocido como

Otro nombre si lo tiene

Motivación

- Es el escenario de un problema concreto de diseño y cómo el patrón resuelve el problema.
- Para entender una descripción más abstracta del patrón

Aplicabilidad

- Las situaciones en que el patrón se puede aplicar.
- Los ejemplos de diseños no buenos que el patrón puede tratar
- Cómo reconocer estas situaciones



Estructura

Es una representación gráfica de las clases del patrón

- Clases: las clases en el patrón con una notación de Técnica de Modelado de Objetos (OMT)
- Objetos: Los objetos en el patrón y sus interacciones con diagramas

Participantes

- Las clases y/o los objetos que participan en el patrón

Colaboraciones

- Cómo los participantes colaboran para cumplir sus responsabilidades

Consecuencias

- las ventajas y desventajas y resultados usando este patrón



Implementación

- Trucos, indicaciones o técnicas de realizar el patrón
- Temas específicos del lenguaje

Códigos de muestras

- Fragmentos de códigos para una posible implementación del patrón en C++ o Smalltalk

Usos conocidos

- Ejemplos de patrones en sistemas existentes. Por lo menos dos ejemplos

Patrones relacionados

- Los patrones de diseño estrechamente relacionados
- Las diferencias más importantes entre ellos
- Otros patrones se deben utilizar éste



TEMA 3. LA CLASIFICACIÓN DE PATRONES DE DISEÑO

3.1. Los criterios de clasificación: los objetivos y áreas de aplicarlos

- Propósitos: para qué sirve el patrón
 - Creativo: sirven para el proceso de creación de objetos
 - Estructural: tratan la composición de clases u objetos
 - Comportamiento: especifican las maneras en que las clases o los objetos interaccionan y distribuyen sus responsabilidades



- El ámbito: El área de aplicar que especifica si el patrón aplica a clases o a objetos
 - 1) Patrones de clases: tratan relaciones entre las clases y sus subclases
 - Las relaciones se establecen por inherencias y son estáticas en el tiempo de compilación
 - 2) Patrones de objetos: tratan las relaciones entre los objetos
 - Estas relaciones se pueden cambiar en el tiempo de ejecución y mucho más dinámica

Comentarios:

- La mayoría de patrones están en el área de objetos
- Casi todos los patrones utilizan inherencia
- Los patrones con el área clase son los que cuidan las relaciones entre las clases



		Propósito		
		Creativo	Estructural	Comportamiento
Ámbito	Clases	Factory Method	Adapter	Interpreter Template Method
	Objetos	Abstract Factory Builder Prototype Singleton	Adapter(objeto) Bridge Composite Decorator Facade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Memento Observer State Strategy Visitor



- El ámbito:
 - 1) Patrones de creación: todos son para crear objetos
 - En ámbito de clases: delega la creación de objetos a sus subclases (Factory method)
 - En ámbito de objetos: delega la creación de objetos a otros objetos
 - 2) Patrones estructurales:
 - Para clases: utilizan herencias para componer clases
 - Para objetos: describen las maneras de ensamblar los objetos
 - 3) Patrones de comportamiento
 - Para clases: usan herencias para describir los algoritmos y flujos de control
 - Para objetos: describen cómo un grupo de objetos cooperan para realizar una tarea que ningún objeto individual puede hacerlo sólo.
- Relaciones entre los patrones



El Tema 4.

El catálogo del patrones de diseño

