Facultad de Ciencias Físico Matemáticas

Nombre: Gabriel Omar Sanchez Reyes

Matricula: 1664322

Materia: Diseño orientado a objetos

Tarea 6

Un patrón de diseño es:

- una solución estándar para un problema común de programación
- una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios
- un proyecto o estructura de implementación que logra una finalidad determinada
- un lenguaje de programación de alto nivel
- una manera más práctica de describir ciertos aspectos de la organización de un programa
- conexiones entre componentes de programas • la forma de un diagrama de objeto o de un modelo de objeto.

¿Para qué sirven?

- Cuando se quiere extender y reutilizar la funcionalidad de una clase SIN UTILIZAR LA HERENCIA
- Definir un comportamiento independiente de donde vaya a ser utilizado. Sirve para indicar atributos semánticos de una clase.
- Asegurar que una clase tiene una sola instancia y proporcionar un punto de acceso global a ella
- Convertir la interfaz de una clase en otra interfaz esperada por los clientes. Permite que clases con interfaces incompatibles se comuniquen
- Componer objetos en jerarquías todo-parte y permitir a los clientes tratar objetos simples y compuestos de manera uniforme
- El patrón FACADE simplifica el acceso a un conjunto de clases proporcionando una única clase que todos utilizan para comunicarse con dicho conjunto de clases.
- Definir una dependencia 1 n de forma que cuando el objeto 1 cambie su estado, los n objetos sean notificados y se actualicen automáticamente
- Encapsular algoritmos relacionados en clases y hacerlos intercambiables. – Se permite que la selección del algoritmo se haga según el objeto que se trate.
- Proporcionar una forma de acceder a los elementos de una colección de objetos de manera secuencial sin revelar su

representación interna. Define una interfaz que declara métodos para acceder secuencialmente a la colección.

Ejemplos:

Patrón DELEGATION

Utilidad:

Cuando se quiere extender y reutilizar la funcionalidad de una clase SIN UTILIZAR LA HERENCIA

Ventajas:

• En vez de herencia múltiple • Cuando una clase que hereda de otra quiere ocultar algunos de los métodos heredados • Compartir código que NO se puede heredar

Patrón INTERFACE

Utilidad y Ventajas:

Utilidad Definir un comportamiento independiente de donde vaya a ser utilizado

Ventajas

Desacople entre comportamiento y clase. Realización de clases “Utilities”

Patrón MARKER INTERFACE

Utilidad y Ventajas

• Utilidad

– Sirve para indicar atributos semánticos de una clase.

• Ventajas:

– Se puede preguntar si un objeto pertenece a una clase de un determinado tipo o no. – Habitualmente se utiliza en clases de

utilidades que tienen que determinar algo sobre objetos sin asumir que son instancias de una determinada clase o no.

Patrón SINGLETON

• Utilidad – Asegurar que una clase tiene una sola instancia y proporcionar un punto de acceso global a ella • Ventajas – Es necesario cuando hay clases que tienen que gestionar de manera centralizada un recurso – Una variable global no garantiza que sólo se instancie una vez

¿Cómo se documentan?

Los patrones de diseño se documentan utilizando plantillas formales con unos campos estandarizados. Existen varias plantillas ampliamente aceptadas, aunque todas ellas deben al menos documentar las siguientes partes de un patrón:

- **Nombre**: describe el problema de diseño.
- **El problema**: describe cuándo aplicar el patrón.
- **La solución**: describe los elementos que componen el diseño, sus relaciones, responsabilidades y colaboración.

La plantilla más utilizada es la plantilla GOF que consta de los siguientes campos:

- Nombre del patrón: Ayuda a recordar la esencia del patrón.
- Clasificación: Los patrones originalmente definidos por GOF se clasifican en tres categorías "Creacional", "Estructural", "Comportamiento".
- Propósito / Problema: ¿Qué problema aborda el patrón?
- También conocido como...: Otros nombres comunes con los que es conocido el patrón.
- Motivación: Escenario que ilustra el problema.
- Aplicabilidad: Situaciones en las que el patrón puede ser utilizado.

- Estructura: Diagramas UML que representan las clases y objetos en el patrón.
- Participantes: Clases y/o objetos que participan en el patrón y responsabilidades de cada uno de ellos
- Colaboraciones: ¿Cómo trabajan en equipo los participantes para llevar a cabo sus responsabilidades?
- Consecuencias: ¿Cuáles son los beneficios y resultados de la aplicación del patrón?
- Implementación: Detalles a considerar en las implementaciones. Cuestiones específicas de cada lenguaje OO.
- Código de ejemplo
- Usos conocidos: Ejemplos del mundo real.
- Patrones relacionados: Comparación y discusión de patrones relacionados. Escenarios donde puede utilizarse en conjunción con otros patrones.