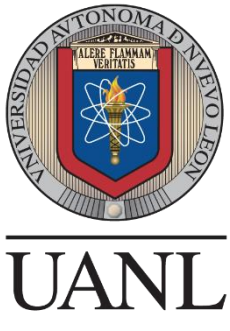

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE CIENCIAS FISICO MATEMÁTICAS

Laboratorio de Diseño Orientado a Objetos

Armando Rodrigo Botello Alanis

Grupo: 009 Matrícula: 1722884

Patrones de diseño

Maestro: Miguel Salazar

Día: 07 de marzo de 2017

Contenido

¿Qué es?

"Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software."

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: su nombre, el problema (cuando aplicar un patrón), la solución (descripción abstracta del problema) y las consecuencias (costos y beneficios).

¿Para qué sirven?

Nos ayudan a cumplir muchos de estos principios o reglas de diseño. Programación SOLID, control de cohesión y acoplamiento o reutilización de código son algunos de los beneficios que podemos conseguir al utilizar patrones.

¿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.

Preguntas de Reflexión

Patrón MVC (Modelo-Vista-Controlador): proviene del principio de que dos aplicaciones se pueden dividir en tres áreas separadas:

Modelo: los datos utilizados en la aplicación

Vista: cómo se representan los datos al usuario

Controlador: cómo se procesa la información en la interfaz del usuario

Proxy: es el patrón que define el objeto intermediario que pide un objeto remoto y que es transparente para el usuario.

Singleton: (instancia única en inglés) es un patrón de diseño diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto.

Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella.

El patrón singleton se implementa creando en nuestra clase un método que crea una instancia del objeto sólo si todavía no existe alguna. Para asegurar que la clase no puede ser instanciada nuevamente se regula el alcance del constructor (con modificadores de acceso como protegido o privado).

El patrón Abstract Factory está aconsejado cuando se prevé la inclusión de nuevas familias de productos, pero puede resultar contraproducente cuando se añaden nuevos productos o cambian los existentes, puesto que afectaría a todas las familias creadas.

El patrón Composite sirve para construir objetos complejos a partir de otros más simples y similares entre sí, gracias a la composición recursiva como. Un panal de abejas

Esto simplifica el tratamiento de los objetos creados, ya que al poseer todos ellos una interfaz común, se tratan todos de la misma manera. Dependiendo de la implementación, pueden aplicarse procedimientos al total o una de las partes de la estructura compuesta como si de un nodo final se tratara, aunque dicha parte esté compuesta a su vez de muchas otras. Un claro ejemplo de uso extendido de este patrón se da en los entornos de programación 2D para aplicaciones gráficas. Un videojuego puede contener diferentes capas "layers" de sprites (como una capa de enemigos) pudiéndose invocar un método que actúe sobre toda esta capa de sprites a la vez (por ejemplo, para ocultarlos, darles un filtro de color etc.).

Bibliografía

- **Design Patterns (Elements of Reusable Object-Oriented Software)**, por Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides.
- **Abstract Factory Pattern in VB .Net**, por Rajesh V. S., http://www.vbdotnetheaven.com/Code/Jun2003/2014.asp .
- **Implementing the Command Pattern in .Net**, http://blogs.vbcity.com/jspano/articles/198.aspx .
- **Implementing the Visitor Design Pattern In .Net**, http://blogs.vbcity.com/jspano/articles/199.aspx .