Tarea 6

Descripción y beneficios de ejemplos de patrones

15/03/2018

Mónica Elizabeth Alba González 160502

a) Factory

El ejemplo de este patrón trata de una construcción de viviendas. Existen dos tipos de constructores, uno que construye casas y otro que construye departamentos. Los beneficios de usar este patrón en este ejemplo son:

- No se necesita reescribir código para describir cómo funciona cada constructor, ya que tienen métodos parecidos.
- No se reescribe código para la construcción de casas o departamentos, porque la diferencia entre ambas viviendas es muy pequeña.

En general, este patrón sirve para crear varios tipos de objetos o productos sin necesidad de reescribir lo mismo varias veces, especialmente cuando uno sabe que va a necesitar un objeto pero no sabe previamente de qué tipo va a ser.

b) Abstract Factory

El ejemplo de este patrón trata de la creación de criaturas, animales o monstruos. El uso de este patrón facilita la creación de varios constructores que pueden producir productos dependientes o relacionados. Los beneficios de usar este patrón en este ejemplo son:

- No se necesita reescribir código para describir cómo funciona cada constructor, ya que tienen métodos parecidos.
- No se reescribe código para la construcción de animales o monstruos, porque a pesar de que existe una diferencia entre ellos, están relacionados, ya que un animal deforme es considerado un monstruo, pero está muy ligado a un animal.

En general, este patrón sirve para tener diversos constructores que puedan construir distintos tipos de productos.

c) Decorator

El ejemplo de este patrón trata de una construcción de coches. En este caso, se construye un coche básico de color blanco. Con distintas clases que descienden de una clase abstracta de coche se le agregan cosas al coche para personalizarlo.

Los beneficios de usar este patrón en este ejemplo son:

- No se tiene que modificar la estructura del coche para agregarle decoraciones.

- Se pueden hacer tantas decoraciones como uno así lo necesite sin interferir con la funcionalidad principal que es crear un coche.

En general, este patrón sirve para crear un producto al que se le pueden agregar distintas funcionalidades sin modificar la esencia del producto cada vez que se le quiera modificar algo.

d) Proxy

El ejemplo de este patrón trata de una construcción de un video, que tiene distintos métodos y de un intermediario que permite al usuario interactuar únicamente con el método que reproduzca el video. Los beneficios de usar este patrón en este ejemplo son:

- Se restringe el acceso a métodos por cuestión de seguridad o para no modificarlos cuando no es necesario.

En general, este patrón sirve para limitar al usuario el acceso al objeto y que no modifique la información privada o confidencial.

e) Visitor

El ejemplo de este patrón trata de una compra de distintos tipos de moneda al que se le agrega un impuesto dependiendo de la moneda. Es decir, a parte del costo de la moneda que es distinto para cada tipo, existe un impuesto que se cobra que es distinto para cada moneda. Los beneficios de usar este patrón en este ejemplo son:

- No se necesita hacer una clase externa que sea del tipo moneda que agregue también el método agregaImpuesto().
- Se puede implementar un método o el que uno quiera sin alterar las clases.

En general, este patrón sirve para agregar funcionalidades que pueden ser distintas a distintas clases sin alterar estas clases.

f) Memento

El ejemplo de este patrón es uno más general, ya que se puede implementar en cualquier caso. En general, este patrón es una manera de almacenar los estados de un objeto de manera sencilla. El memento es el objeto que se guarda en los estados (es el que se puede

cambiar en el ejemplo). Originator obtiene los valores del memento actual y el Caretaker es el arreglo que guarda todos los mementos.

g) Adapter

El ejemplo de este patrón trata de un chofer y un auto que se conduce solo. El auto autónomo se adapta a los métodos de un chofer porque tienen la misma funcionalidad (llevar a alguien a un lugar), solo que prescinde del chofer. Los beneficios de usar este patrón en este ejemplo son:

- Sirve para no reescribir métodos o funcionalidades cuando se tienen distintos tipos pero que funcionan muy parecido o tienen el mismo fin.