Tarea 6

Alberto Acosta López
ITAM

Marzo 2018

Factory Pattern
Este patrón es uno de los mejores para crear objetos ya que éstos se crean sin exponer la creación lógica de los objetos al cliente y solo se hacen referencias a los objetos mediante una interfaz común.
Un beneficio alcanzado es que puedes crear varias clases que hacen cosas distintas, pero tienen una base común i.e. la interfaz. No es necesario tener que implementar todos los métodos dentro de una sola clase.

Abstract Factory Pattern
Trabaja sobre una fábrica que fabrica otros objetos, o puede ser una fábrica de fábricas. Una interfaz es responsable de crear una fábrica de objetos relacionados sin definir explícitamente sus clases. Cada fábrica generada funciona como el Factory Pattern.
Un beneficio alcanzado es que se pueden crear funcionalidades distintas, que requieran un tipo de operación específica para funcionar. No es necesario crear proyectos relacionados, sino que todo se crea mediante una sola interfaz.

Decorator
Proporciona la opción de añadir nuevas funcionalidades a un objeto existente sin alterar su estructura . Funciona como un envolvente a dicha estructura y mantiene los métodos intactos de dicha estructura.
Un beneficio fue poder crear funcionalidades nuevas a los objetos, como si fuera un avance iterativo sin tener que afectar el trabajo hecho.

Proxy
En este patrón, una clase representa la funcionalidad de otra clase. Se tiene una clase que se muestra como su propia interfaz y funcionalidades ante el mundo.
Un beneficio alcanzado fue poder representar las funcionalidades de una clase sin tener que alterar la clase original. Así pruebas de manera directa la funcionalidad y se le pueden añadir nuevas funcionalidades.

Visitor
Se utiliza una clase visitante que cambia la ejecución de los algoritmos de un elemento de una clase. Así, la función visitada se puede probar con parámetros distintos, aunque éstos deben ser aceptados por el objeto a probar.
Un beneficio alcanzado fue poder revisar las funcionalidades de distintas clases dentro del patrón, así como poder mantener un mejor control de dichas funcionalidades para tener la menor cantidad de errores.

Memento
Este patrón es utilizado para restaurar el estado de un objeto a uno anterior.
Un beneficio es que puedes revisar los cambios realizados al objeto y poder restaurar cambios que se pudiesen creer perdidos.

Adapter
Este patrón combina las funcionalidades de 2 objetos incompatibles, funcionando como un puente. Tiene una clase responsable de unir las funcionalidades.
Un beneficio alcanzado fue poder comprobar distintas opciones y problemas que pudiesen surgir en el programa, así se alcanzan los menores errores posibles y se cumplen con más pruebas unitarias.

Java RX
Es una forma de programación mediante la cual se construye el código mediante bloques de programación funcional. La complejidad de un programa al desarrollarse de manera síncrona, puede llegar a ser extrema. En el momento que el programa empieza a ser asíncrona, se complica demasiado el código.
El problema se complica cuando se tienen varios casos y empiezan a crecer de manera exponencial.
Esto es bueno cuando se tienen requisitos o “pedidos” que se hacen a los programas. Si la conexión se realiza directamente con el servidor, se tomaría un tiempo en exceso, con lo cual resulta en problemas de información en el programa. Es una forma fácil de dejar de hacer el trabajo cuando la interfaz no está respondiendo.

Aquí surge una facilidad para poder transformar los datos. Por ejemplo, si los datos están almacenados en otro tipo de “envase”, es más fácil que se enfoque en el “envase” que en el contenido. Otra forma de transformar los datos es transmitiendo los datos.

Tiene como ventaja la concurrencia. Existen varios problemas con las interfaces, ya que cada una puede ser tan compleja como distinta, controlando varios aspectos de la interfaz. Es una forma fácil de mover el trabajo fuera de la interfaz principal para evitar que se trabe o retrase el proceso y cálculo.

Es una manera de realizar trabajo mediante entregas continuas, buscando cubrir los mayores problemas para la funcionalidad del programa. Las propiedades clave de ésto son:

1. Modelan valores que varían a través del tiempo.
2. Modelan eventos que tienen ocurrencias en tiempos discretos o específicos en el tiempo.
3. El sistema puede ser cambiado para responder a esos eventos.

Un modelo ordinario tiene muy pocas capacidades para los programas interactivos, ya que no tienen la habilidad de “correr” los programas sin mapear desde los datos iniciales a los datos finales. Éstos problemas se pueden solucionar:

1. Creando una estructura de datos de acciones que aparecen como los datos finales. Las acciones pueden correr por un intérprete externo.
2. Usar acciones que tienen identidades que les permiten mantener datos mutables separados.

3. Correr las acciones pero recibir los datos finales después, para usar una interacción entre el evento y el programa.

Existen 2 tipos de de sistemas: push-based y pull-based.
Los push-based toman los eventos y los envían a través de una red para obtener un resultado. Los push-based esperan hasta que el resultado es demandado y trabajan de “reversa” a través de la red para obtener el valor demandado.

Los problemas que surgen es que son muy intensos computacionales para procesar muestras en un intervalo regular y la red tiene que esperar la duración del intervalo y descubrir los cambios que surgieron.
Expresión Lambda:
Este programa muestra distintas formas de asignar valores a expresiones al momento en que son declaradas. Tiene la opción de ser declarada con y sin los tipos i.e. int, incluso con distintas formas de declaraciones.
La funcionalidad que fue creada es para expresar las distintas maneras de declarar variables y valores, así como explicar y ejemplificar formas sencillas de utilizar esta expresión.
Una ventaja es declarar las variables y realizar operaciones sin necesidad de crear métodos adicionales.

Map Reduce
Este programa divide la tarea en partes pequeñas y se las asigna a varias computadoras para luego ser recolectadas en un solo lugar e integrados como un dataset. Toma un bloque de datos y los convierte en otro tipo de datos donde los elementos son convertidos en tuplas y los agrupa en pares para tener un número menor de tuplas.
La funcionalidad que fue creada es para encontrar el año mínimo y máximo en un bloque de datos formados como tablas. Esto se realiza en poco tiempo.
Una ventaja es que seccionan los datos, los comparan por separado y el resultado se obtiene de una forma demasiado rápida, por lo que la cantidad de procesamiento es mínima.