

# Práctica 1, Strategy y Observer.

Rojas Reyes Saúl Adrián.

114006224.

1. Menciona los principios de diseño esenciales del patrón Strategy y Observer.  
Menciona una desventaja de cada patrón.

El patrón strategy consiste en programar hacia el uso de las interfaces, así cuanto tengo muchos objetos que desempeñan las mismas funciones pero de diferente manera, puedo separar lo que se mantiene constante, y dejarlo en la una clase padre, digamos Servicio. Lo que cambia lo implementaremos en una interfaz para cada tipo de comportamiento, digamos ServicioModoPago, así cada clase nueva que herede de servicio puede tener sus propios comportamientos sin generar problemas, además así no escribimos código, solo agregamos.

El patrón Observer consiste en tener un objeto como sujeto observable, el cual tiene una lista interna de observadores, estos observadores están al pendiente del sujeto pero son independientes de él, así se pueden agregar más observadores sin alterar al sujeto ni a ell comportamiento de los otros observadores en existencia. El sujeto puede notificar a todos sus observadores usando su lista interna, y cualquier objeto que implemente la interfaz de observador puede subscribirse a el sujeto.

La desventaja con el patrón strategy, es que para cada nueva clase tiene que crear 2, por lo que rápidamente aumenta el número de clases en un proyecto. En proyectos grandes esto podría llegar a causar problemas.

La desventaja del patrón Observer es que el sujeto está débilmente ligado a sus observadores, el no sabe nada de los observadores, ni ellos de él, solo los une un método de notificación, por lo que cualquier cambio no sería detectado si no se notifica, así habiendo cabida de errores.

-----  
Sobre la práctica. No se necesita ningún argumento para ser ejecutada, esta imprime la simulación de los meses en la consola, el patrón de Strategy se usa con los las clases que heredan de Servicio y cada una tiene su propia interfaz, además si se quisieran agregar más bancos es posible hacerlo, el patrón Observer lo usan los clientes, al ser siempre especificados como personas, son solo una clase y no utilizan Strategy, pero es posible que lo hagan. Todos los sujetos, o bancos pueden ser observados por clientes mientras hereden de Banco.

La práctica se ejecuta con la clase Practica1, la simulación se encuentra en su main, yo usé el método javac Practica1.java para compilarla, luego java Practica1, debería funcionar correctamente.