

Práctica 01

Modelado y Programación

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

- Observer:
 - Definir una dependencia “uno a muchos” entre los objetos. Así cuando un objeto cambia de estado, todos sus dependientes son notificados y actualizados automáticamente.
 - Lograr obtener diseños débilmente acoplados entre objetos que interactúen.

Desventajas:

- o Con el uso de este patrón ocurre que se van a desconocer las consecuencias de una actualización, lo cual, dependiendo del problema, puede afectar en mayor o menor medida (por ejemplo, al rendimiento).
- o La información de cambios transmitida puede ser irrelevante para ciertos observadores
- o El código fuente del sujeto a menudo no revela qué observadores están siendo informados.

- Strategy:
 - Identificar los aspectos de que varían encapsulándolos y separarlos de los que no cambian.
 - Programar en una interfaz, no en una implementación
 - Favorecer la composición sobre la herencia

Desventajas

- o Si sólo tienes un par de algoritmos que raramente cambian, no hay una razón real para complicar el programa en exceso con nuevas clases e interfaces que vengan con el patrón.
- o Los clientes deben conocer las diferencias entre estrategias para poder seleccionar la adecuada.
- o Muchos lenguajes de programación modernos tienen un soporte de tipo funcional que te permite implementar distintas versiones de un algoritmo dentro de un grupo de funciones anónimas. Entonces puedes utilizar estas funciones exactamente como habrías utilizado los objetos de estrategia, pero sin saturar tu código con clases e interfaces adicionales.

Comentarios adicionales:

- Para correr el programa: Desde la carpeta src poner: "javac Main.java" para compilar, luego "java Main" para correrlo
- Dentro de nuestro archivo de tarea se encuentran los siguientes documentos adicionales:
 - Un .gitignore, para no estar constantemente generando los .class
 - Cuatro .txt que ocupamos para las pruebas