

Informe Ejercicio 2

Principios de diseño (SOLID) utilizados en esta práctica:

- Principio de responsabilidad única: Consiste en que una clase solo realice una única acción o trabajo. Las clases que cumplen este principio son: Acción, Cliente, MercadoAcciones.
- Principio de sustitución de Liskov: Se basa en usar una clase hija que adopta el comportamiento de su padre. Gracias a esto, conseguimos utilizar dicha clase hija sin saber cuál es en concreto.
- Principio de inversión de la dependencia: Se apoya en el uso de interfaces a la hora de hacer el código, en lugar de utilizar clases concretas. La clase que cumple este principio es: Cliente.
- Principio de segregación de interfaces: Consiste en tener diferentes implementaciones de una misma interfaz.
- Principio de abierto/cerrado: Establece que todo el código debería estar abierto a su extensión, pero cerrado a su modificación.

Patrones de diseño empleados:

A la hora de realizar esta práctica se ha usado el patrón Observador con el cual "Cliente" delega en la interfaz "Observer" la elección de que algoritmo usar. La clase "MercadoAcciones" juega el papel de Sujeto, la interfaz "Observer" de Java funciona como Observador, la clase "Accion" como Sujeto Concreto y "Cliente" como Observador Concreto.

