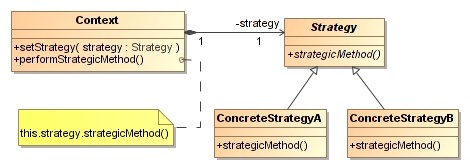
Patrón Strategy:



Problemática:

Suponiendo que para un mismo problema hay varias estrategias para resolverlo, se desea separar estas estrategias por varios motivos:

* Incluir el código de los algoritmos en los clientes hace que estos sean demasiado grandes y complicados de mantener y/o extender.
* El cliente no va a necesitar todos los algoritmos en todos los casos, de modo que no queremos que dicho cliente los almacene si no los va a usar.
* Si existiesen clientes distintos que usasen los mismos algoritmos, habría que duplicar el código, por tanto, esta situación no favorece la reutilización.

A razón de estos problemas surge el patrón Strategy. Se separan los algoritmos o estrategias distintas en otra jerarquía para que el cliente trabaje con un objeto intermediario. Esta elección puede ser tomada por el cliente o por algún tipo de lógica que elige cual sería el más adecuado para cada caso.

Componentes del patrón:

Contexto (Context) : Es el elemento que usa los algoritmos, por tanto, delega en la jerarquía de estrategias. Configura una estrategia concreta mediante una referencia a la estrategia necesaria.

Estrategia (Strategy): Declara una interfaz común para todos los algoritmos soportados. Esta interfaz será usada por el contexto para invocar a la estrategia concreta.

EstrategiaConcreta (ConcreteStrategy): Implementa el algoritmo utilizando la interfaz definida por la estrategia.

Uso:

El ejemplo que se hace es un ejemplo trivial, solo para explicar cómo se usaría. Este patrón ya lo hemos aplicado en el ejercicio de Master Mind de la primera asignatura en el que se empleaba para hacer un put de un código por el usuario o el código obtenido de forma aleatoria a modo de demo. Otro ejemplo que he visto por internet es para calcular el área de distintas figuras geométricas o un editor de texto en el que en función de lo que se quiera devuelve el texto con un formato u otro. En cualquier caso se puede aumentar el número de estrategias de forma sencilla y sin repetir código.

Conclusión:

Es una buena forma de separar las estrategias de resolución o comportamientos distintos que puede tener un problema. A mi modo de ver es muy similar a una inyección de dependencias, si no es exactamente la misma idea. En función de lo que necesite el cliente se le inyecta una estrategia o un objeto con la misma para resolver los problemas expuestos en el apartado Problemática.

Infografía:

https://es.wikipedia.org/wiki/Strategy\_(patr%C3%B3n\_de\_dise%C3%B1o)