Patrón Estrategia

El patrón de estrategia es útil cuando hay un conjunto de algoritmos relacionados y un objeto cliente necesita poder elegir de forma dinámica un algoritmo de este conjunto que se adapte a sus necesidades actuales.

El patrón de estrategia sugiere mantener la implementación de cada uno de los algoritmos en una clase separada. . Cada uno de estos algoritmos encapsulados en una clase separada se conoce como estrategia. Un objeto que utiliza un objeto de estrategia a menudo se conoce como un objeto de contexto. Con diferentes objetos de estrategia en su lugar, cambiar el comportamiento de un objeto de contexto es simplemente una cuestión de cambiar su objeto de estrategia por el que implementa el algoritmo requerido. Para acceder a diferentes objetos de Estrategia de manera transparente, todos los objetos de Estrategia deben estar diseñados para ofrecer la misma interfaz. En el lenguaje de programación Java, esto se puede lograr diseñando cada objeto de Estrategia como un implementador de una interfaz común o como una subclase de una clase abstracta común que declara la interfaz común requerida. Una vez que el grupo de algoritmos relacionados se encapsula en un conjunto de clases de Estrategia en una jerarquía de clases, un cliente puede elegir entre estos algoritmos seleccionando e instanciando una clase de estrategia apropiada. Para alterar el comportamiento del contexto, un objeto cliente necesita configurar el contexto con la instancia de estrategia seleccionada. Este tipo de arreglo separa completamente la implementación de un algoritmo del contexto que lo usa. Como resultado, cuando se cambia la implementación de un algoritmo existente o se agrega un nuevo algoritmo al grupo, tanto el contexto como el objeto cliente (que usa el contexto) no se ven afectados.