Padrão Strategy

Certified Developer
The Ultimate Tech Degree

DigitalHouse >
Coding School

Índice

1 | Contextualização

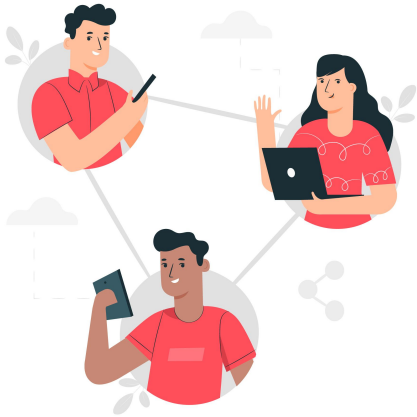
“ Vamos separar os diferentes comportamentos de uma classe e assim mudar a estratégia para quando necessário. “

Propósito

Um determinado objeto terá um comportamento que pode ser simples e sempre o mesmo, mas às vezes esse comportamento se torna mais complexo e, conforme as necessidades, muda.

O padrão **Strategy** faz com que os algoritmos variem independentemente do cliente que os está usando.

Ele propõe uma solução simples baseada em um objeto que muda e cujo comportamento é aquele que se adapta às circunstâncias.



Solucão

Existe uma interface, Estrategia (Strategy), que serve para estabelecer as assinaturas dos métodos que vamos mudar e as diferentes classes que implementam esta interface, Estrategias Concretas (ConcreteStrategy), que são as que os diferentes algoritmos possuem para fazer a tarefa.

Podemos ver alguns casos em que queremos que a Estratégia também tenha algum atributo que seja comum a cada Estratégia específica. Neste caso, a Estratégia será uma classe abstrata e as Estratégias concretas serão herdadas.

Vantagens e desvantagens

✓	Usar o padrão fornece uma alternativa para a herança de classe, uma vez que uma mudança dinâmica na estratégia pode ser feita.
✓	Se um algoritmo usa informações que os clientes não deveriam saber, o uso do padrão Strategy evita a exposição dessas estruturas.
✓	Esse padrão nos ajuda a trocar várias estratégias possíveis.
✗	O número de objetos criados aumenta, então há um aumento na troca de objetos entre estratégia e contexto

2 | Diagrama UML

Diagrama Solução Padrão Strategy

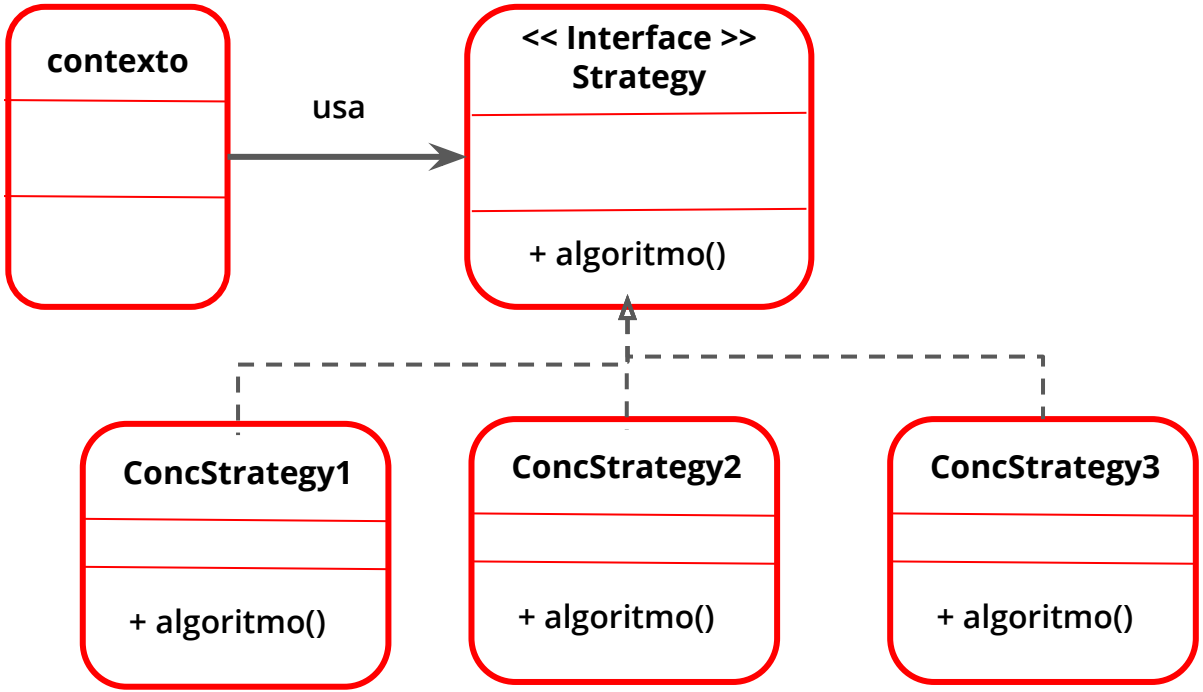


Diagrama Solução Padrão Strategy

- **Contexto:** É o elemento que usa os algoritmos, delegados na hierarquia das estratégias. Você configura uma estratégia concreta referindo-se à estratégia necessária e pode, portanto, alterar a estratégia de acordo com suas necessidades.
- **Interface Strategy:** declara uma interface comum para todos os algoritmos suportados. Essa interface será usada pelo contexto para invocar a estratégia específica. Uma classe do tipo Abstract também pode ser usada no caso de especificar variáveis ou métodos comuns às diferentes estratégias.
- **ConcStrategy:** implementa o algoritmo usando a interface definida pela estratégia. É usado para ter lógica nesta classe que pode ser reutilizada e isolada do contexto.

Como funciona?

Quando o contexto deseja usar um algoritmo específico, ele primeiro define a variável (que terá um objeto que implementa a interface de Estratégia) com um valor adequado. Em seguida, você invoca o método que deseja executar. Aquele que recebe esta invocação será um objeto ConcreteStrategy que a executa com seu próprio algoritmo.


DigitalHouse>
Coding School