

# Escola de Arquitetos

## Design Pattern Strategy

# Intenção

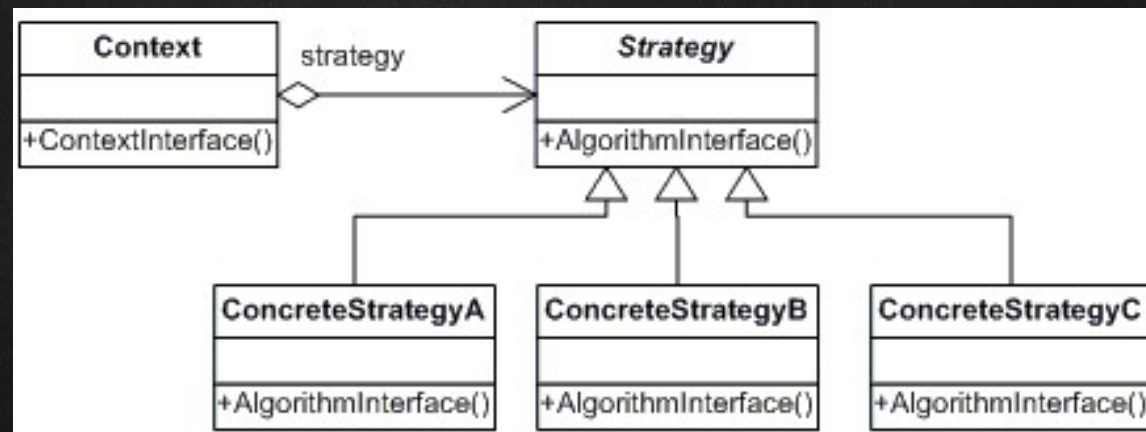
*“Definir uma família de algoritmos, encapsular cada um, e fazê-los intercambiáveis. Strategy permite que algoritmos mudem independentemente entre clientes que os utilizam.”*

*GoF (Gang of Four)*

# Traduzindo

*“Strategy nos permite configurar uma classe com um de vários comportamentos, utilizando o conceito de OO (Orientação a Objetos) chamado COMPOSIÇÃO.”*

# Estrutura



# Aplicação

*Strategy pode ser aplicado quando um objeto deve ser parametrizado com um de vários algoritmos, os quais podem ser encapsulados e representados por uma única interface.*



# Prós

- *Substitui a herança pela composição e delegação.*
- *Impede as instruções condicionais (switch, if, else...).*
- *Os algoritmos são fracamente acoplados à entidade de contexto podendo ser substituídos sem alterar a entidade de contexto.*
- *Muito fácil de ser estendido.*

# Contras

- *Aumenta a complexidade geral do código, criando várias classes adicionais.*
- *O cliente deve estar ciente das diferenças entre as estratégias para escolher uma adequada.*

# Relações com outros padrões

- *State, Strategy, Bridge (e até certo ponto Adapter) têm estruturas de solução semelhantes. Eles diferem em intenção, isto é, eles resolvem problemas diferentes.*
- *Decorator permite que você mude a pele de um objeto. Strategy permite que você mude o interior.*



# Relações com outros padrões

- *O Template Method usa herança para alterar o algoritmo estendendo suas partes em classes diferentes. Strategy usa delegação para alterar o comportamento do objeto, substituindo o objeto de estratégia aninhada. Template Method funciona no nível da classe. Strategy permite que você altere o comportamento de objetos individuais.*

# Relações com outros padrões

- *State pode ser considerado como uma extensão do padrão Strategy. Ambos os padrões usam a composição para alterar o comportamento do objeto principal, delegando o trabalho aos objetos auxiliares. O padrão Strategy torna esses objetos completamente independentes. No entanto, o padrão State permite que objetos de estado alterem o estado atual do contexto com outro estado, tornando-os interdependentes.*

# Perguntas?