

Padrão de Projeto Strategy

Rafael N. Romeu

Centro de Ciências Computacionais – Universidade Federal do Rio Grande (FURG)

Rio Grande – RS – Brasil

rafaelromeufurg@gmail.com

***Resumo.** Este relatório tem o intuito de analisar o padrão de projeto strategy descrevendo seus pontos fortes e fracos e em quais casos sua utilização é recomendada e explicar seu funcionamento.*

1. Introdução

Para solucionar problemas que ocorrer recorrentemente em um determinado contexto durante o processo de desenvolvimento de softwares foram concebidos os padrões de projeto. Cada padrão tem por objetivo a solução de uma “família” de problemas.

Em geral, um padrão tem quatro elementos essenciais:

- O nome do padrão: É uma referência para podermos descrever um problema de projeto, suas soluções e suas consequências;
- O problema: Descreve a situação em que o padrão deve ser utilizado;
- A solução: Fornece uma descrição abstrata de classes, objetos e relações para a solução do problema;
- As consequências: São os resultados das análises de vantagens e desvantagens da aplicação do padrão.

2. Strategy

O que é strategy? Em quais casos é utilizado? Quais são seus pontos fortes e fracos?

“Definir uma família de algoritmos, encapsular cada uma delas e torná-las intercambiáveis. Strategy permite que o algoritmo varie independentemente dos clientes que o utilizam” Gamma, E. “Padrões de projeto: Soluções reutilizáveis de software orientado a objetos”

O padrão strategy, além de encapsular os algoritmos de mesma família também permite a reutilização do código, outra vantagem é a facilidade para a extensão das funcionalidades.

O padrão Strategy é recomendado quando:

- Muitas classes relacionadas diferem somente no comportamento.
- Há necessidade de variantes de um algoritmo.
- Um algoritmo dados que o cliente não deveria ter conhecimento.
- Uma classe possui muitos comportamentos e estes aparecem como múltiplos “ifs”.

O strategy possui 3 principais entidades.

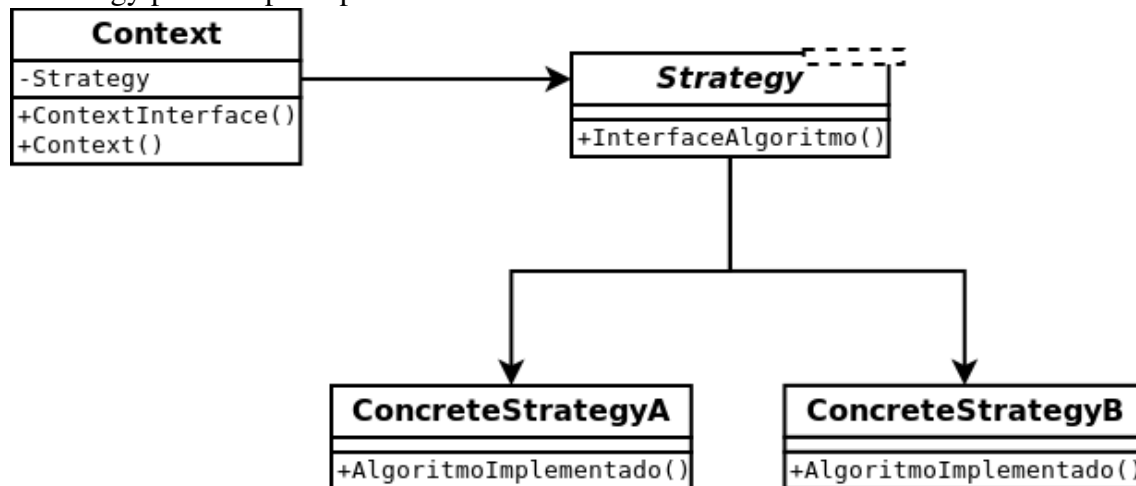


Figure 1. Diagrama Strategy

Context:

- É configurado com um objeto concreteStrategy;
- Mantém uma referência para um objeto Strategy;
- Envia dados ao Strategy ou pode definir uma interface que permite a Strategy acessar seus dados

Strategy:

- Define uma interface comum para todos os algoritmos suportados, Context utiliza essa interface para chamar o objeto definido por uma concreteStrategy.

ConcreteStrategy:

- Implementa algoritmo usando a interface strategy.

References

Gamma, E. and Helm, R. (2000) “Padrões de projeto: Soluções reutilizáveis de software orientado a objetos”