Design Patterns: Solução comum para um problema recorrente ao utilizar paradigmas de orientação a objetos

Padrões de projeto são soluções genéricas para problemas recorrentes do desenvolvimento de software orientado a objetos;

Categorias: Criacionais, estruturais e comportamentais

**Stratagy**: Sempre que eu tenho um algoritmo que varia de acordo com determinado parâmetro, usamos para eliminar o if else separando cada implementação especifica por classe

Existem três principais categorias de padrões de projeto:

Comportamentais (que serão vistos neste treinamento)

Estruturais

Criacionais

* Como diminuir a complexidade do nosso código: Trocando múltiplas condicionais por classes, esta técnica é chamada de **Strategy**, que é um dos padrões de projeto

Sempre que uma nova funcionalidade for implementada, o ideal é que possamos criar código novo e editar o mínimo possível de código já existente. Este é um dos principais pontos do princípio Aberto-Fechado (Open-Closed Principle) do SOLID. Ao editar código existente, podemos acabar quebrando funcionalidades já implementadas e funcionais.

O padrão **Template Method** favorece o reaproveitamento de códigos comuns entre classes, evitando assim duplicações de códigos.

Aplicamos o pattern **Template Method** com a criação de um método concreto na classe “mãe”, que chama métodos abstratos implementados nas classes “filhas”