

# PROBLEMA

CRIAR UMA LISTA DE ALUNOS PARA SER APRESENTADA EM UMA TELA. PORÉM O USUÁRIO PODERÁ ESCOLHER ENTRE VÁRIAS FORMAS DE ORDENAÇÃO. A LISTA PODERÁ ESTAR ORDENADA:

- POR NOME;
- POR SOBRENOME;
- POR MATRICULA;
- POR CURSO;
- ETC.

APRESENTAR O PRIMEIRO EXEMPLO SEM O PADRÃO.



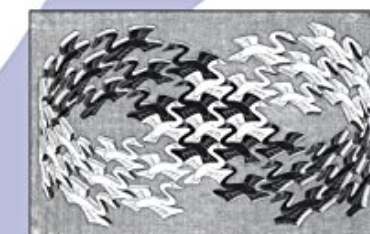
# PADRÕES DE PROJETO

## TEMPLATE METHOD

**Prof. Me Eugênio Júlio Messala Cândido Carvalho**  
**eugeniojuliomessala@gmail.com**

**Padrões de  
Projeto**

Soluções reutilizáveis de software  
orientado a objetos



ERICH GAMMA  
RICHARD HELM  
RALPH JOHNSON  
JOHN VLISSIDES

**Design Patterns**



# TEMPLATE METHOD

## ***INTENÇÃO***

Definir o esqueleto de um algoritmo em uma operação, postergando alguns passos para as subclasses.

Template Method permite que subclasses redefinam certos passos de um algoritmo sem mudar a estrutura do mesmo.

# TEMPLATE METHOD

## ***MOTIVAÇÃO***

Um método-template define um algoritmo em termos da operação abstrata que as subclasses redefinem para fornecer um comportamento concreto. As subclasses da aplicação definem os passos do algoritmo.

# TEMPLATE METHOD

## ***APLICABILIDADE***

O padrão Template Method pode ser usado:

- para implementar as partes invariantes de um algoritmo uma só vez e deixar para as subclasses a implementação do comportamento que pode variar;
- para controlar extensões de subclasses. Você pode definir um método-template que chama operações “gancho” em pontos específicos, desta forma permitindo extensões somente nesses pontos.

# TEMPLATE METHOD

## ***APLICABILIDADE***

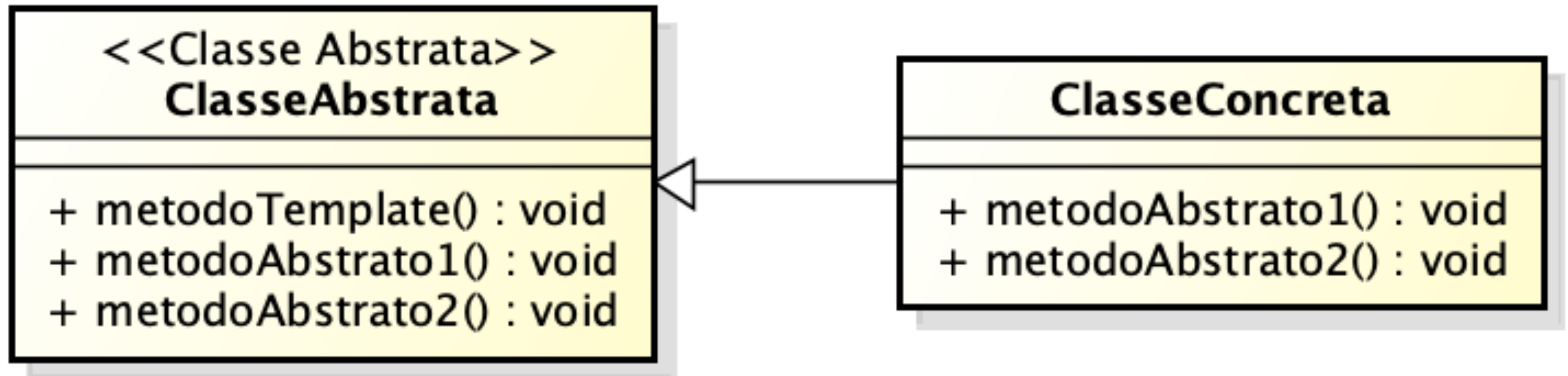
O padrão Template Method pode ser usado:

- quando o comportamento comum entre subclasses deve ser fatorado e concentrado numa classe comum para evitar a duplicação de código. Primeiramente, você identifica as diferenças no código existente e então separa as diferenças em novas operações. Por fim, você substitui o código que apresentava as diferenças por um método-template que chama uma dessas novas operações.



# TEMPLATE METHOD

## *ESTRUTURA*



# BIBLIOGRAFIA

- GAMMA, Erich et al. **Padrões de Projeto: soluções reutilizáveis de software orientado a objetos.** Porto Alegre: Bookman, 2007.
- KERIEVSKY, Joshua. **Refatoração para padrões.** Porto Alegre: Bookman, 2008.