



Design Patterns

(Padrões de Projeto)

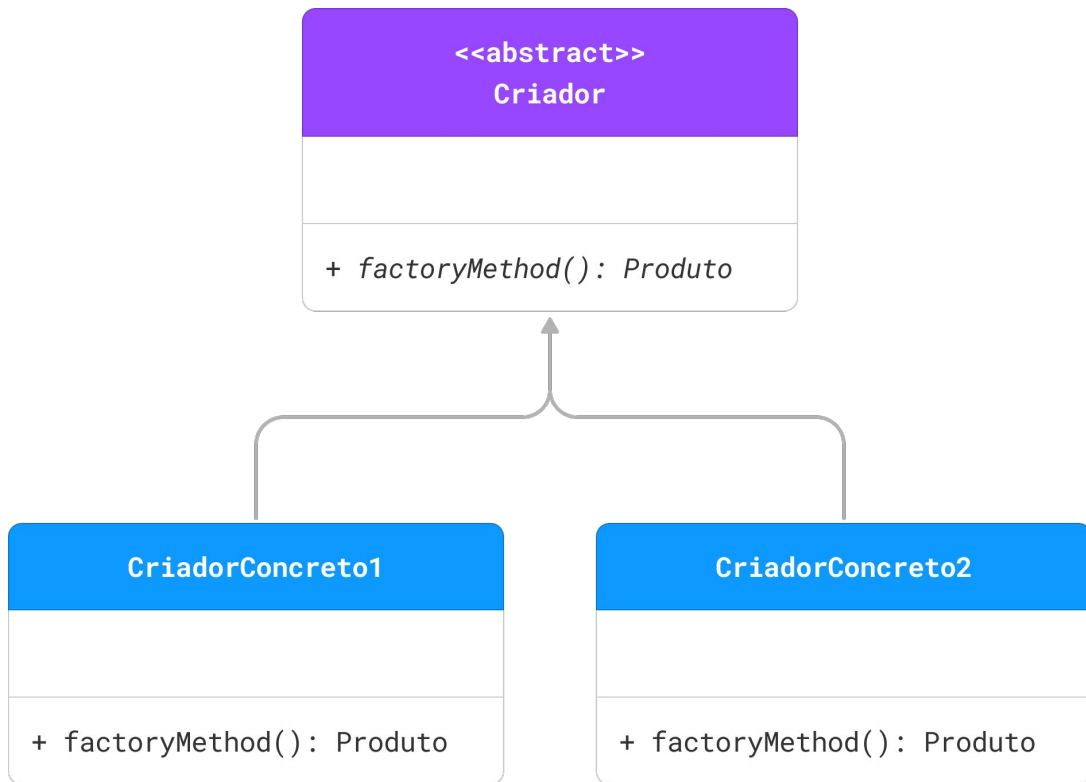


Abstract Factory

Definição

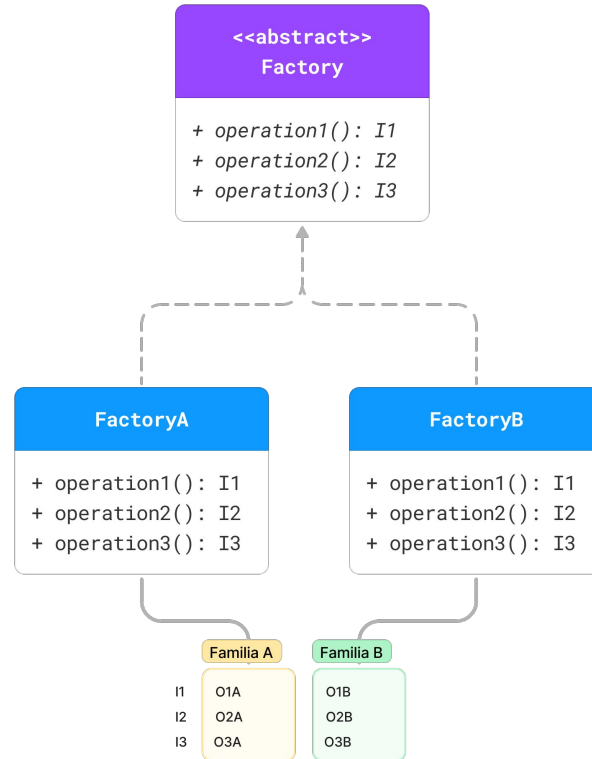
O padrão Abstract Factory fornece uma interface para *criar famílias de objetos relacionados* ou dependentes sem especificar suas classes concretas.

Factory Method



Herança

Abstract Method



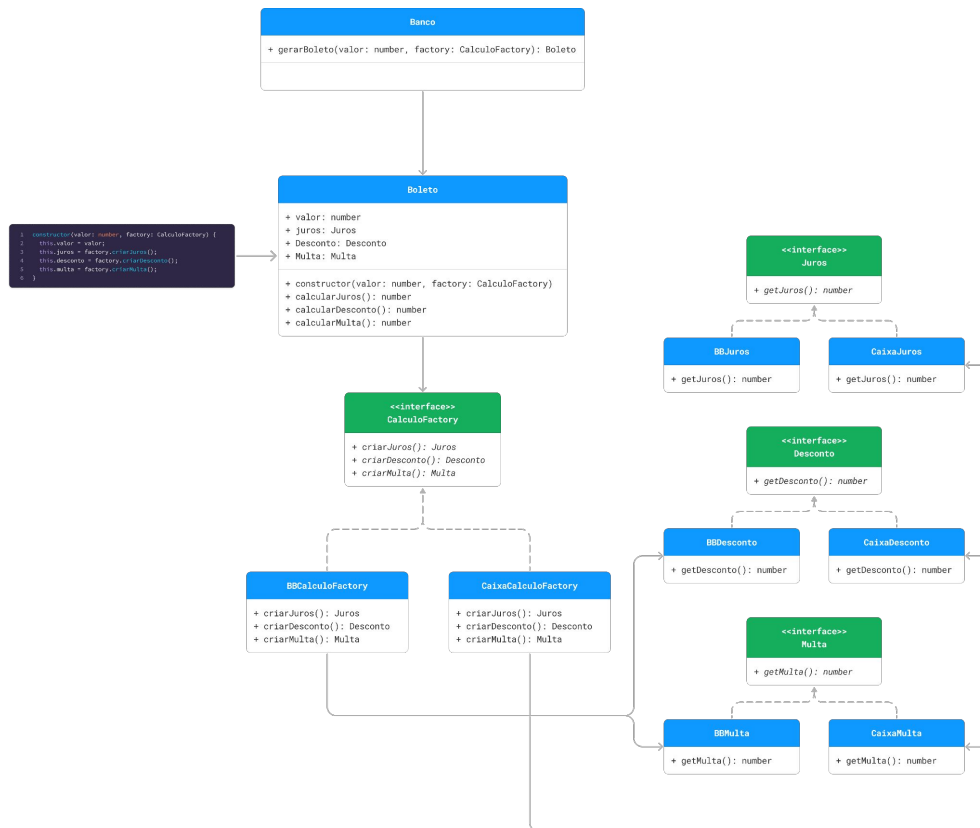


Abstract Method

Para exemplificar

- Você está criando um módulo para geração de boletos em sua aplicação.
- Diferentes taxas e cálculos para cada banco.
 - Caixa Juros 2% Desconto 10% Multa 5%
 - BB, Juros 3% Desconto 5% Multa 2%

Abstract Method

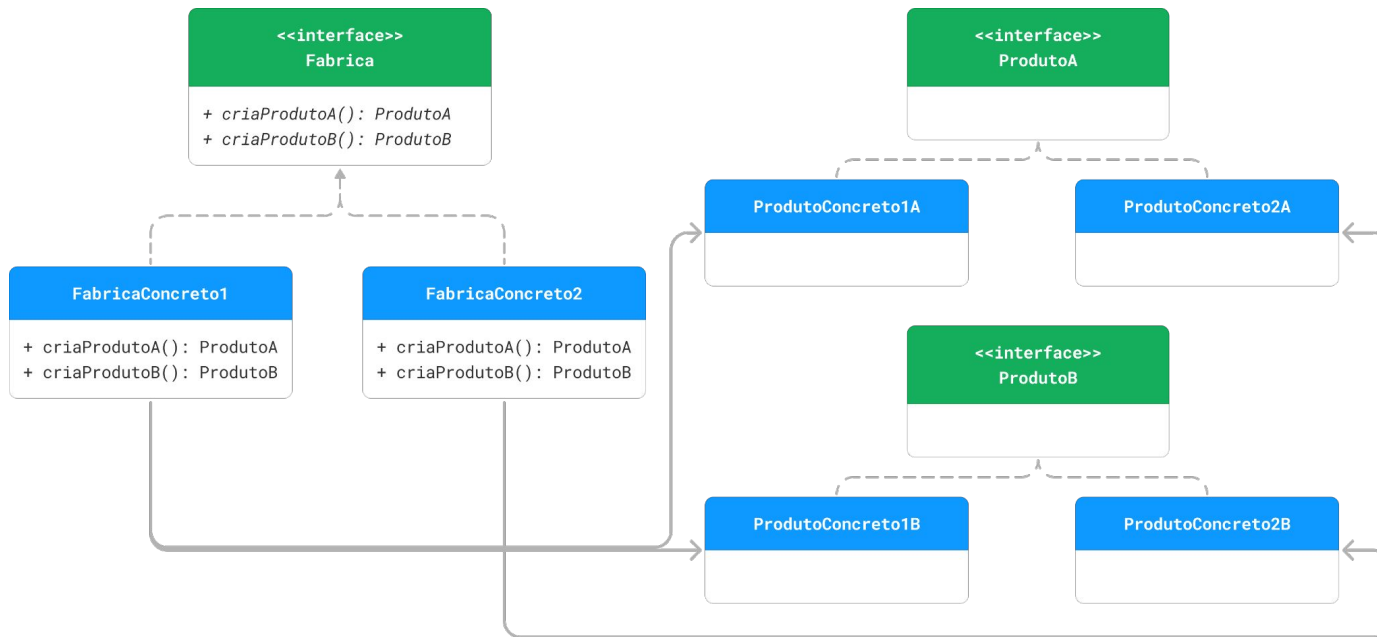


Abstract Method



Show me the code!

Abstract Method





Abstract Method

Aplicabilidade

- Quando um sistema deve ser independente de como seus produtos são criados, compostos ou representados.
- Quando um sistema deve ser configurado com uma dentre múltiplas famílias de produtos.
- Quando uma família de objetos relacionados foi projetada para ser usada em conjunto e é necessário impor essa restrição.
- Quando se deseja fornecer uma biblioteca de produtos e se deseja revelar para o cliente apenas suas interfaces e não suas implementações.



Abstract Method

Consequências

- Promove o isolamento de classes concretas.
- Facilita a troca de famílias de produtos.
- Promove a consistência entre produtos.
- Suportar novos tipos de produtos é difícil.
- Criar novos produtos de um tipo já existente é fácil.



Abstract Method

Exercício

Você está desenvolvendo um jogo de RPG onde os jogadores podem explorar diferentes mundos (como um mundo medieval e um mundo futurista). Cada mundo tem um conjunto específico de itens, como armas e armaduras, que são diferentes em aparência e características dependendo do mundo em que o jogador está.

- Crie os tipos de item `Weapon` e `Armor` com os métodos `getName`
- Crie um conjunto de itens que faça sentido para um mundo medieval e um mundo futurista