

# Abstract Factory

Padrão de Projeto de Criação

**INTEGRANTES:** Arthur Monteiro, Brenda Stefany, Bruna Letícia, Elenice Florentina,  
Eduarda Faria e Marco Aurélio de Faria

01

# O problema

- **Cenário:** Um sistema precisa criar objetos que pertencem a diferentes "famílias" de produtos compatíveis entre si.
- **Dificuldade:** Instanciar objetos diretamente no código cliente (`new ProdutoConcreto()`) gera um forte acoplamento com as implementações específicas.
- **Consequências:**
  - A troca ou adição de novas famílias de produtos exige modificação direta no código cliente.
  - Violação do Princípio Aberto/Fechado (OCP), que preza por código aberto para extensão, mas fechado para modificação.
  - Risco de usar accidentalmente produtos de famílias diferentes, causando inconsistências.

02

# A solução

- **Abordagem:** Introduzir uma camada de abstração para a criação dos objetos.
- **Interface *AbstractFactory*:** Declara métodos para criar cada produto de uma família.
- **Classes *ConcreteFactory*:**
  - Implementam a interface para instanciar uma família específica de produtos.
- **Resultado:**
  - O código cliente passa a depender apenas das interfaces abstratas.
  - A troca de famílias de produtos se torna simples, alterando apenas a fábrica instanciada.

# 03

## Estrutura

- **AbstractFactory:** Interface que declara os métodos para criar produtos abstratos (Ex: FabricaGUI).
- **ConcreteFactory:** Classe que implementa a AbstractFactory para uma família específica (Ex: FabricaWindows).
- **AbstractProduct:** Interface que define um produto genérico (Ex: Botao).
- **ConcreteProduct:** Implementação específica de um produto para uma família (Ex: BotaoWindows).
- **Client:** Usa apenas as interfaces da fábrica e dos produtos, sem conhecer as implementações.

# 04

## Vantagens

- Consistência
- Baixo Acoplamento
- Flexibilidade
- Centralização

## Desvantagens

- Complexidade
- Rigidez

# 06

## Aplicabilidade do Padrão

- O sistema deve ser independente de como seus produtos são criados e representados.
- O código precisa trabalhar com múltiplas famílias de produtos sem depender de suas classes concretas.
- É necessário garantir que produtos de uma mesma família sejam sempre utilizados em conjunto.
- Você quer fornecer uma biblioteca de classes expondo apenas suas interfaces, e não as implementações.

# Conclusão

- O Abstract Factory é uma solução robusta para criar famílias de objetos de forma coesa e desacoplada.
- Promove a manutenibilidade e escalabilidade do código em sistemas que lidam com múltiplas variantes de produtos.