

# Padrão de projeto Factory

## Descrição

Este é um dos padrões mais utilizados em modernas linguagens de programação.

- Define uma interface para criar objetos, mas permite que subclasses decidam qual classe instanciar.
- Refere-se ao objeto recém-criado através de uma interface comum.

## Descrição do problema

A maioria das linguagens orientadas a objeto têm grande flexibilidade no nível de chamada do método por meio do polimorfismo. No lado da criação, no entanto, é preciso especificar `new X ()`, onde X é um tipo específico.

## Descrição da solução

O padrão de projeto factory lida com o problema de criar objetos (produtos) sem especificar a classe exata de objeto que será criada. O método de fábrica lida com esse problema definindo um método separado para criar os objetos, que as subclasses podem então substituir para especificar o tipo derivado de produto que será criado.

## Consequências

A principal razão pela qual o padrão factory é usado é que ele introduz uma separação entre o aplicação e uma família de classes (ele introduz um baixo

acoplamento em vez de um alto acoplamento que esconde as classes concretas da aplicação).

### **Vantagens**

- Fornece uma maneira simples de estender a família de produtos com pequenas alterações no código do aplicativo.
- Fornece ganchos de personalização. Quando os objetos são criados diretamente dentro da classe, é difícil substituí-los por objetos que estendem sua funcionalidade. Se uma fábrica é usada para criar uma família de objetos, os objetos personalizados podem facilmente substituir os objetos originais, configurando a fábrica para criá-los.

### **Desvantagens**

- A fábrica deve ser usada para uma família de objetos. Se as classes não estenderem a classe base ou interface comum, elas não poderão ser usadas em um modelo de projeto factory.