

Abstract Factory

A Solução

No Abstract Factory primeiro precisamos declarar explicitamente interfaces para cada produto distinto da família de produtos (ex: veículos e aeronaves). Então podemos fazer todas as variantes dos produtos seguirem essas interfaces.

Em seguida declaramos a *fábrica abstrata*—uma interface com uma lista de métodos de criação para todos os produtos que fazem parte da família de produtos (por exemplo, `criarTransporteVeículo`, `criarTransporteAeronave`). Esses métodos devem retornar tipos abstratos de produtos representados pelas interfaces que extraímos previamente: `Veículo`, `Aeronave` e assim por diante.

E como fazer para as variantes Uber, 99 ou outros? Para cada variante de uma família de produtos nós criamos uma classe fábrica separada baseada na interface `TransporteFábricaAbstrata`. Uma fábrica é uma classe que retorna produtos de um tipo em particular. Por exemplo, a classe `UberTransport` só pode criar objetos `Carros` e `Aviões`, já a classe `NineNineTransport` só cria `Motos` e `Helicópteros`.

O código cliente tem que funcionar com ambas as fábricas e produtos via suas respectivas interfaces abstratas. Isso permite que você mude o tipo de uma fábrica que passou para o código cliente, sem quebrar o código cliente atual.