Suponha que você deve trabalhar em um projeto computacional com um conjunto de carros, cada um de uma determinada fábrica. Para exemplificar suponha os quatro seguintes modelos/fabricantes:

\* Palio – Fiat

\* Gol – Volkswagen

\* Celta – Chevrolet

\* Fiesta – Ford

Será necessário manipular este conjunto de carros em diversas operações, como poderíamos modelar este problema?

Uma primeira solução, mais simples, seria criar uma classe para representar cada carro, no entanto ficaria muito difícil prever as classes ou escrever vários métodos iguais para tratar cada um dos tipos de objetos.

Poderíamos então criar uma classe base para todos os carros e especializá-la em subclasses que representem cada tipo de carro, assim, uma vez definida uma interface comum poderíamos tratar todos os carros da mesma maneira. O problema surge quando vamos criar o objeto, pois, de alguma forma, precisamos identificar qual objetos queremos criar. Ou seja, precisaríamos criar uma enumeração para identificar cada um dos carros e, ao criar um carro, identificaríamos seguindo essa enumeração. Veja o código abaixo:

**public enum**  ModeloCarro {

    palio,gol, celta, fiesta

}

**public** **abstract** **class** FabricaCarro {

**public** Carro criarCarro(ModeloCarro modelo) {

**switch** (modelo) {

**case** celta:

**return** **new** Celta();

**case** fiesta:

**return** **new** Fiesta();

**case** gol:

**return** **new** Gol();

**case** palio:

**return** **new** Palio();

**default**:

**break**;

        }

    }

}

Esta implementação já corresponde a uma implementação do Factory Method, pois um método fábrica cria Objetos concretos que só serão definidos em tempo de execução. No entanto, esta implementação traz um problema quanto a manutenibilidade do código, pois, como utilizamos um switch para definir qual objeto criar, a cada criação de um novo modelo de carro precisaríamos incrementar este switch e criar novas enumerações. Como resolver este problema?