

Arquitetura de Software

Padrões de Projetos – Padrões Criacionais

Nairon Neri Silva

Factory Method

Método Fábrica / Construtor Virtual

Intenção

- Definir uma interface para criar um objeto, mas deixar as subclasses decidirem que classe instanciar. O Factory Method permite adiar a instanciação para subclasses
- Também conhecido como *Virtual Constructor*

Motivação

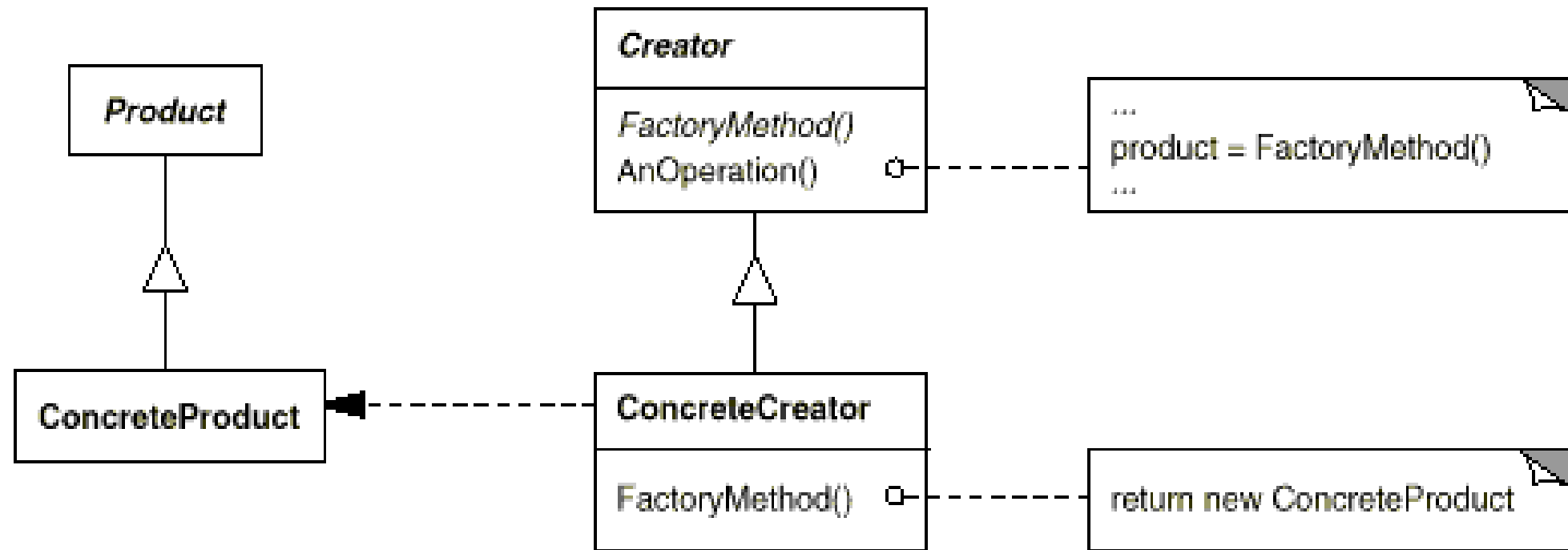
- Os frameworks usam classes abstratas para definir e manter relacionamentos entre objetos. Um framework é frequentemente responsável pela criação destes objetos.
- Considere um framework para aplicações que podem representar múltiplos documentos para o usuário. Duas abstrações chave são: Aplicação e Documento. As duas classes são abstratas, e os clientes devem prover subclasses para realizar suas implementações específicas para a aplicação

Motivação

- Por exemplo: para criar uma aplicação de desenho, definimos as classes `DrawingApplication` e `DrawingDocument`.
- A classe `Application` é responsável pela administração de `Documents` e irá criá-los conforme exigido – quando o usuário seleciona `Open` (abrir) ou `New` (novo), por exemplo em um menu.

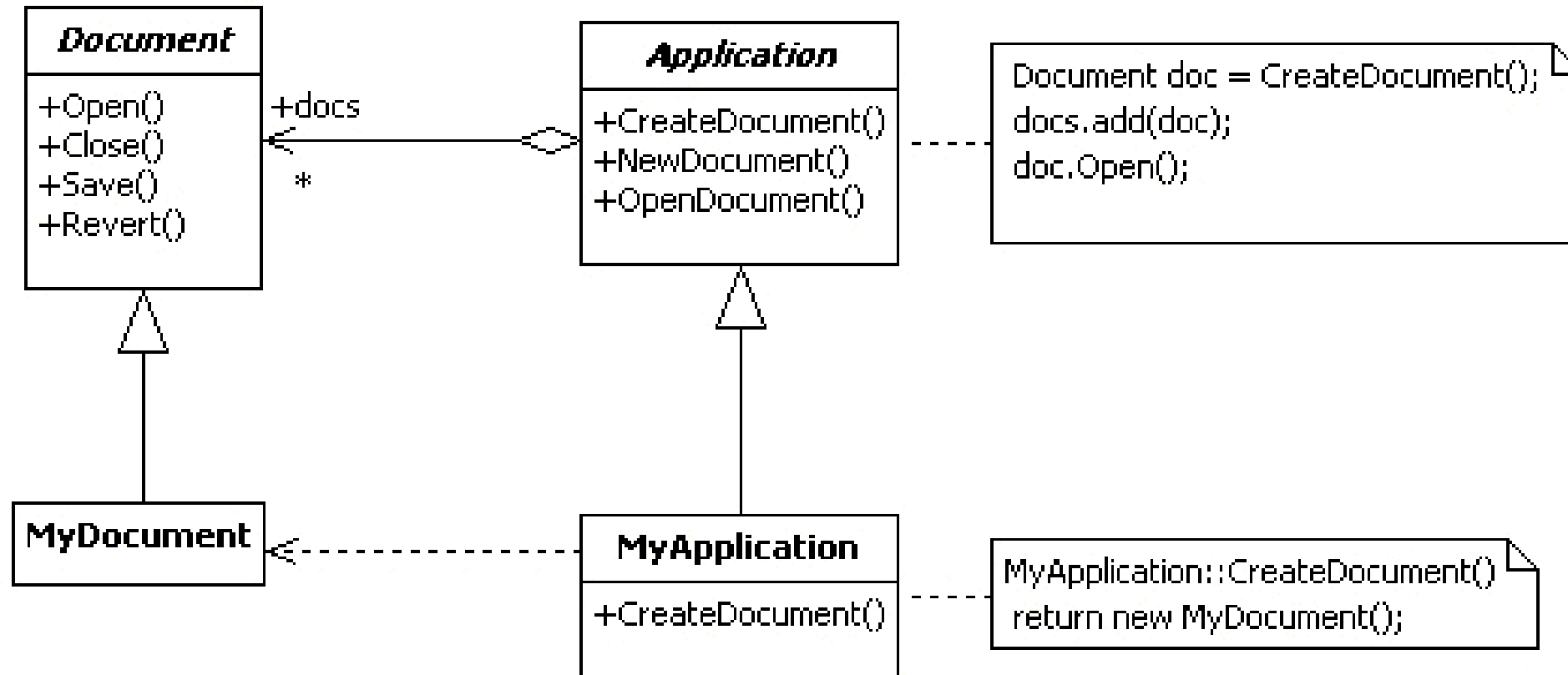
Aplicabilidade

1. Quando uma classe não pode antecipar a classe de objetos que deve criar
2. Quando uma classe quer que suas subclasses especifiquem os objetos que criam
3. Quando classes delegam responsabilidade para uma dentre várias subclasses auxiliares, e você quer localizar o conhecimento de qual subclasse auxiliar que é a delegada

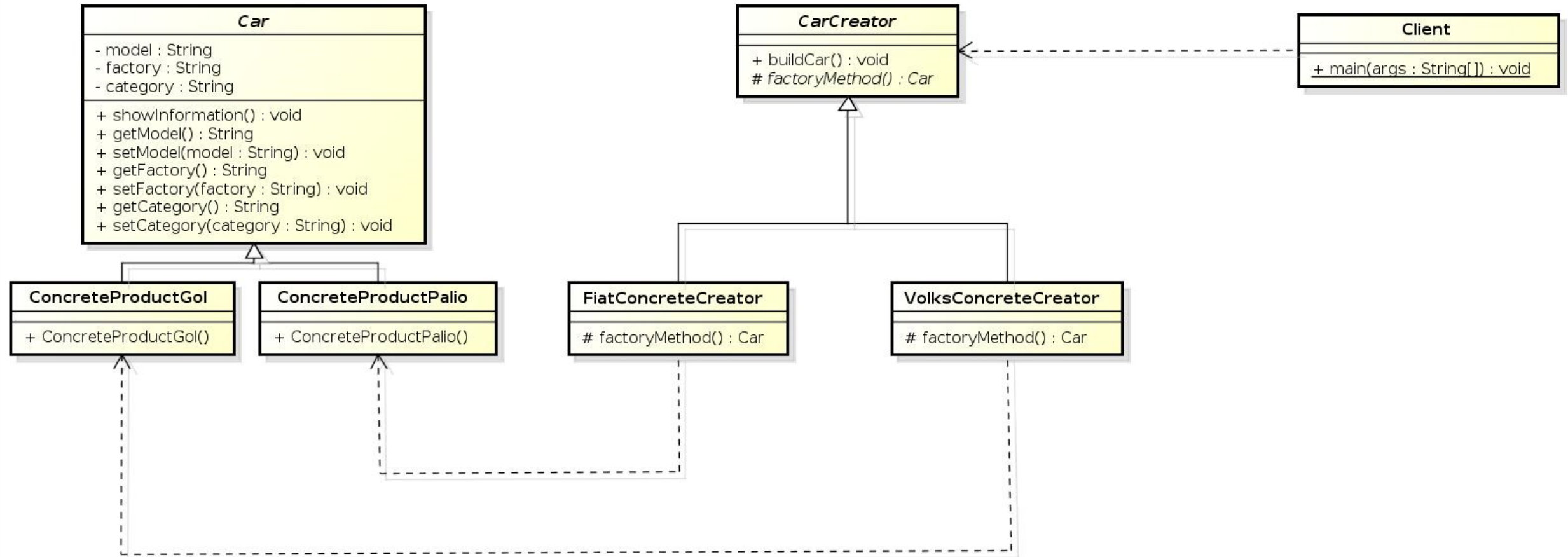


Participantes

- **Product** (Document)
 - Define a interface de objetos que o método fábrica cria
- **ConcreteProduct** (MyDocument)
 - Implementa a interface de Product
- **Creator** (Application)
 - Declara o método fábrica o qual retorna um objeto do tipo Product
- **ConcreteCreator** (MyApplication)
 - Redefine (sobrescreve) o método fábrica para retornar uma instância de um ConcreteProduct



pkg factorymethod2



Fonte: <http://patrick.ison.com.br/questions/5/patterns/factory-method>

Consequências

1. Fornece “ganchos” para subclasses
 - Pontos de extensão (*hot spots*)
2. Conecta hierarquias de classes paralelas

Saiba Mais...

- <https://refactoring.guru/pt-br/design-patterns/factory-method>
- <https://www.youtube.com/watch?v=1rB0PhvAwiU>