

Abstract Factory

Então o padrão abstract factory auxilia na criação de objetos que se relacionam entre si ou são da mesma família.

- Princípio do aberto e fechado
 - Aberto para extensões e fechado para modificações
- Princípio da responsabilidade única.
 - Onde apenas uma classe é responsável pela criação dos objetos e a outra apenas usa esses objetos.

Participantes

- Abstract Factory
 - Concrete Factory
 - Abstract Product
 - Concrete Product
 - Client
- ❖ Extra:
- CreatorFactory

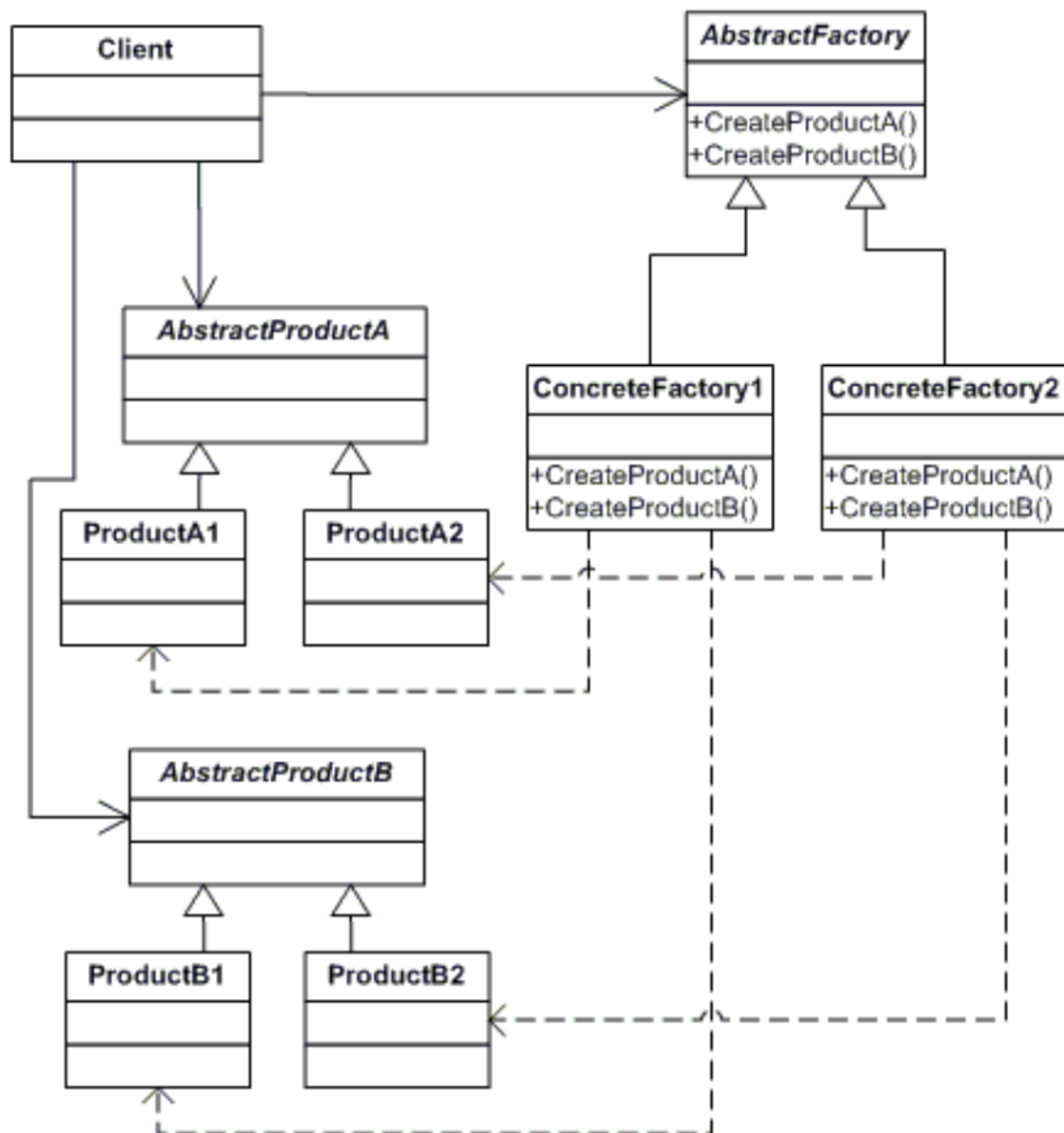
Abstract Factory é a própria classe abstrata da fábrica abstrata. É ela que tem os métodos das classes que serão criadas no final.

Concrete Factory é a classe concreta que faz a herança da fábrica abstrata (Abstract Factory). É a classe responsável pela criação das famílias das classes.

Abstract Product é o resultado final do que a classe cliente deseja retornar no final, é a classe abstrata que entra no método da fábrica abstrata.

Product class é o produto que faz a herança da Abstract Product.

A classe creator factory é ela quem será responsável para fazer a criação dos objetos é onde será feita a instância é uma classe adicional que não está diretamente relacionada ao padrão, porém é nela que fica centralizada a ação de criar os objetos.



Usando um exemplo de venda de conjunto de móveis, que no final não se relacionam entre si, porém são da mesma família e não podem ser entregues de forma misturada, partes de cada conjunto que são diferentes em um apenas.

Imagine um conjunto de mesa de centro, poltrona e um sofá com os tipos de conjunto como conjunto moderno, conjunto rústico e um conjunto vitoriano.

Usando o abstract factory teremos certeza que no final vamos ter um “Produto” (o resultado final neste caso o conjunto de classes mesa de centro, poltrona e sofá da mesma família) correto e que nunca serão feitos com tipos diferentes.