

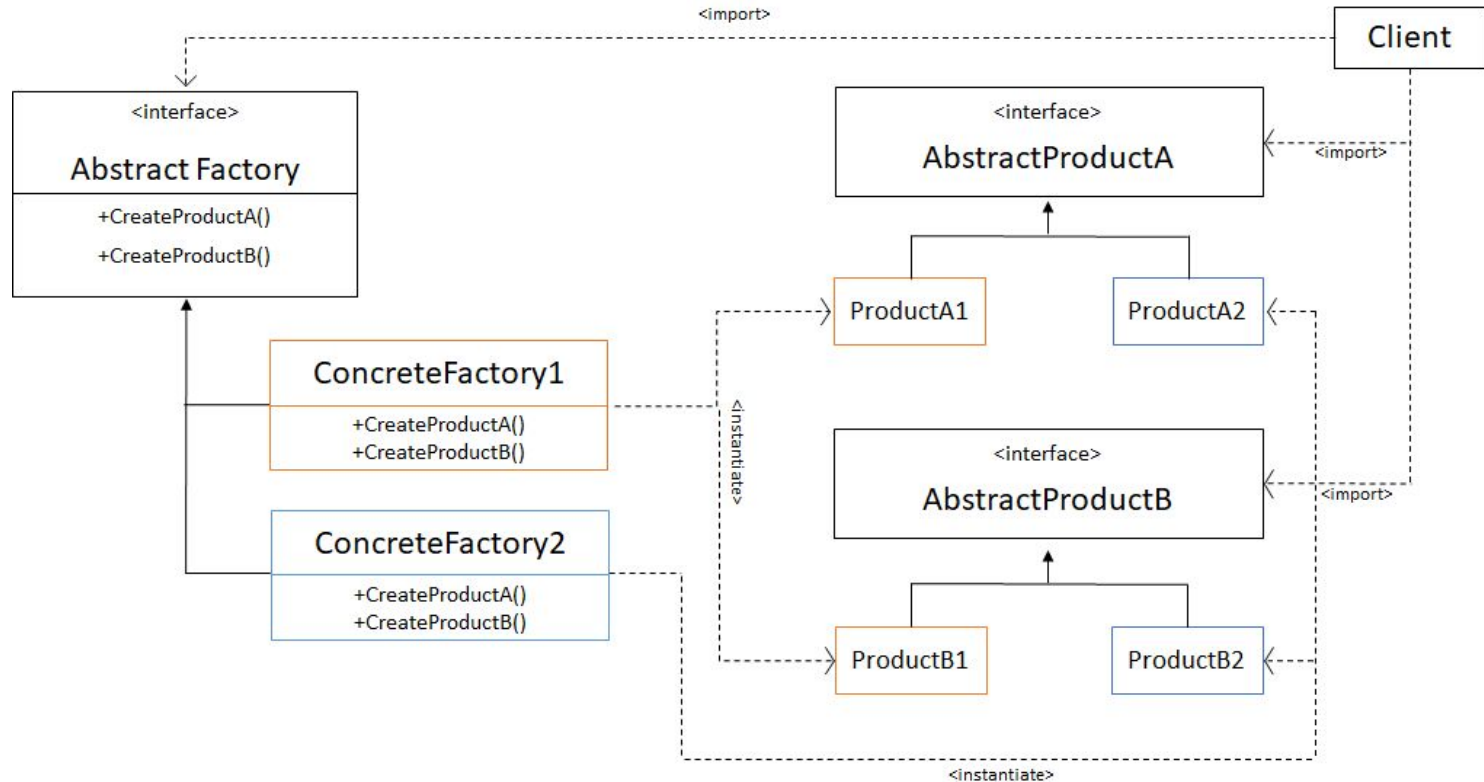
Abstract Factory Pattern

Lucas Agra de Omena
lao2@ic.ufal.br

Problem

- How can a class be independent from the objects creation that it requires?
- How can an application be independent of how its objects are created?
- How can families of related or dependent objects be created?

Solution



Benefits and Liabilities

- (+) Isolates concrete classes. The Abstract Factory pattern helps you control the classes of objects that an application creates.
 - *A factory encapsulates the responsibility and the process of creating product objects, it isolates clients from implementation classes. Clients manipulate instances through their abstract interfaces.*
- (+) Makes exchanging product families easy. The class of a concrete factory appears only once in an application.
 - *This makes it easy to change the concrete factory an application uses. It can use different product configurations simply by changing the concrete factory. As the abstract factory creates a complete family of products, the whole product family changes at once.*
- (+) Makes it easy to promote consistency among products.
 - *When product objects in a family are designed to work together, it's important that an application use objects from only one family at a time.*
- (-) Supporting new kinds of products is difficult (extending abstract factories to produce new kinds of products).
 - *It involves changing the Abstract Factory class and all of its subclasses.*