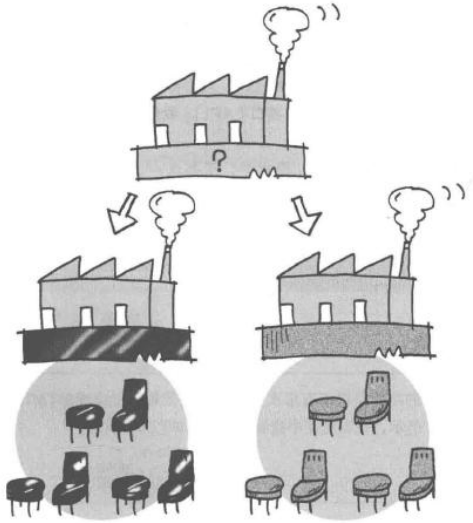
Abstract Factory (抽象工厂, Creational Pattern)



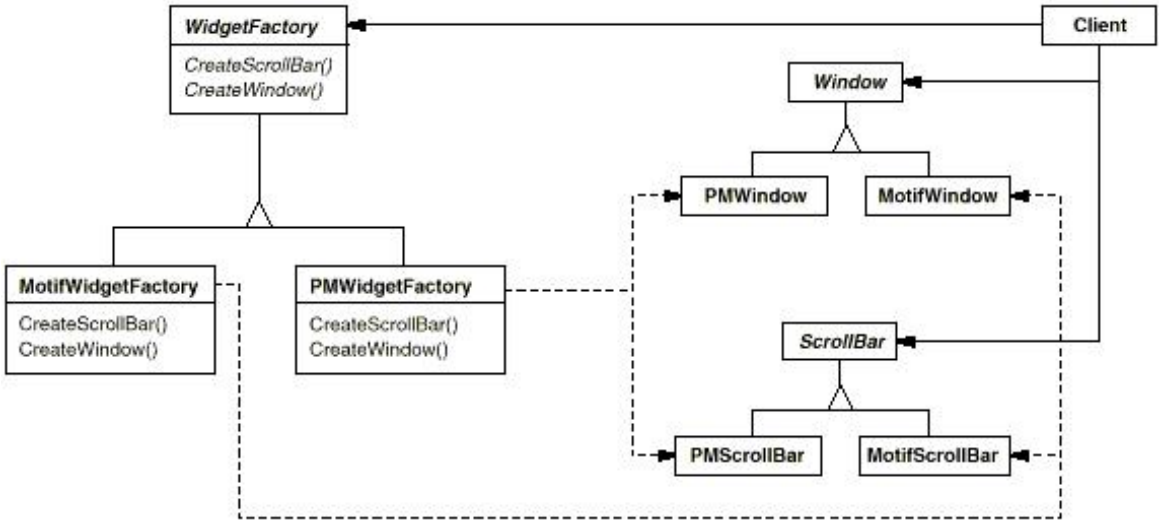
Kai SHI

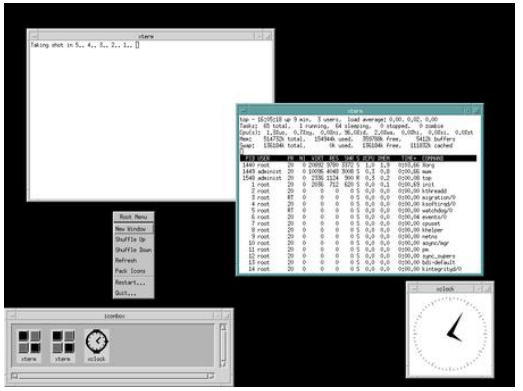
Abstract Factory

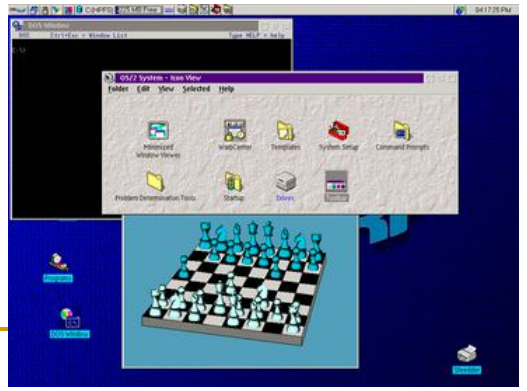
- Intent
  - Provide an interface for creating families of related or dependent objects without specifying their concrete classes.
- Also Known As
  - Kit

Motivation

- A user interface toolkit that supports multiple look-and-feel standards, such as Motif and Presentation Manager.



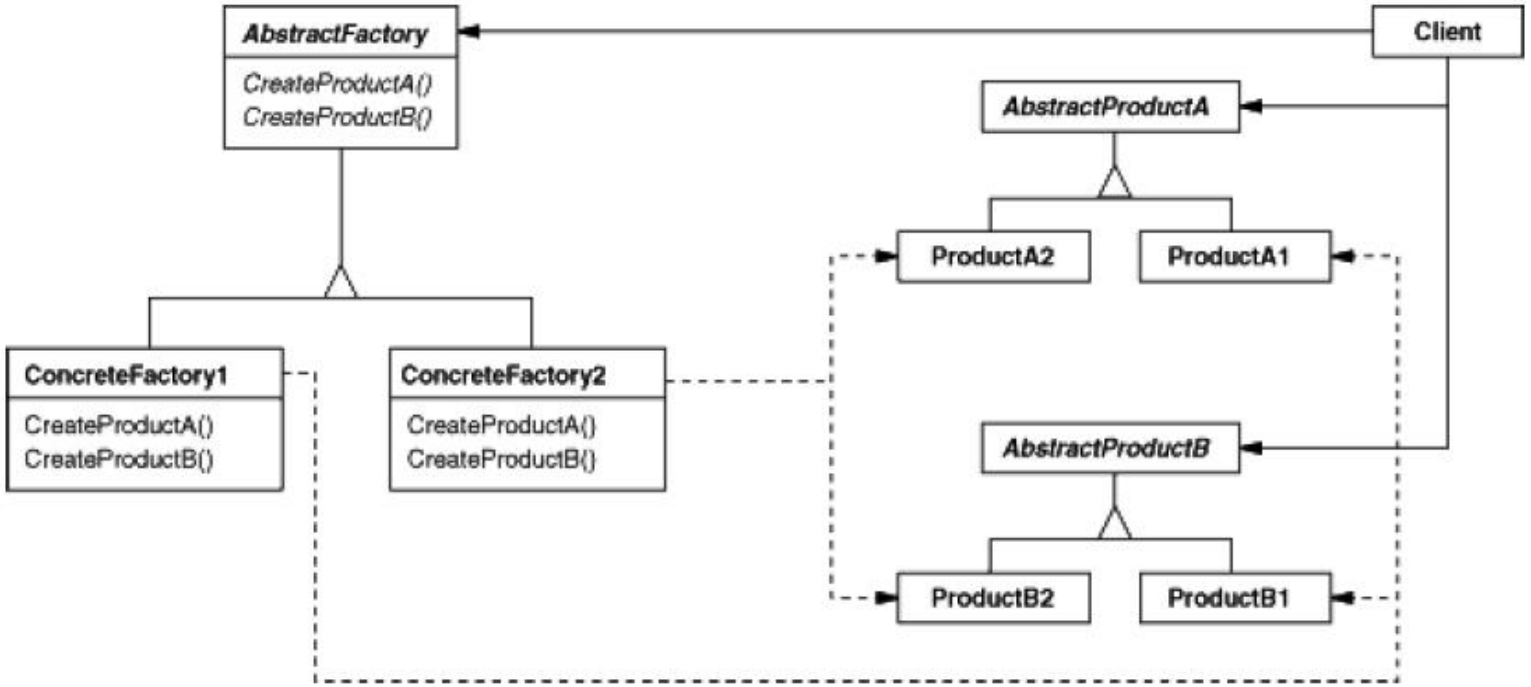




Applicability: Use the Abstract Factory pattern when

- a system should be independent of how its products are created, composed, and represented.
- a system should be configured with one of multiple families of products.
- a family of related product objects is designed to be used together, and you need to enforce this constraint.
- you want to provide a class library of products, and you want to reveal just their interfaces, not their implementations.

Structure


AbstractFactory
CreateProductA()
CreateProductB()
Client
AbstractProductA
ProductA2
ProductA1
ConcreteFactory1
CreateProductA()
CreateProductB()
ConcreteFactory2
CreateProductA()
CreateProductB()
AbstractProductB
ProductB2
ProductB1

Participants

- AbstractFactory: declares an interface for operations that create abstract product objects.
- ConcreteFactory: implements the operations to create concrete product objects.
- AbstractProduct: declares an interface for a type of product object.
- ConcreteProduct: defines a product object to be created by the corresponding concrete factory.
- Client: uses only interfaces declared by AbstractFactory and AbstractProduct classes.

Consequences

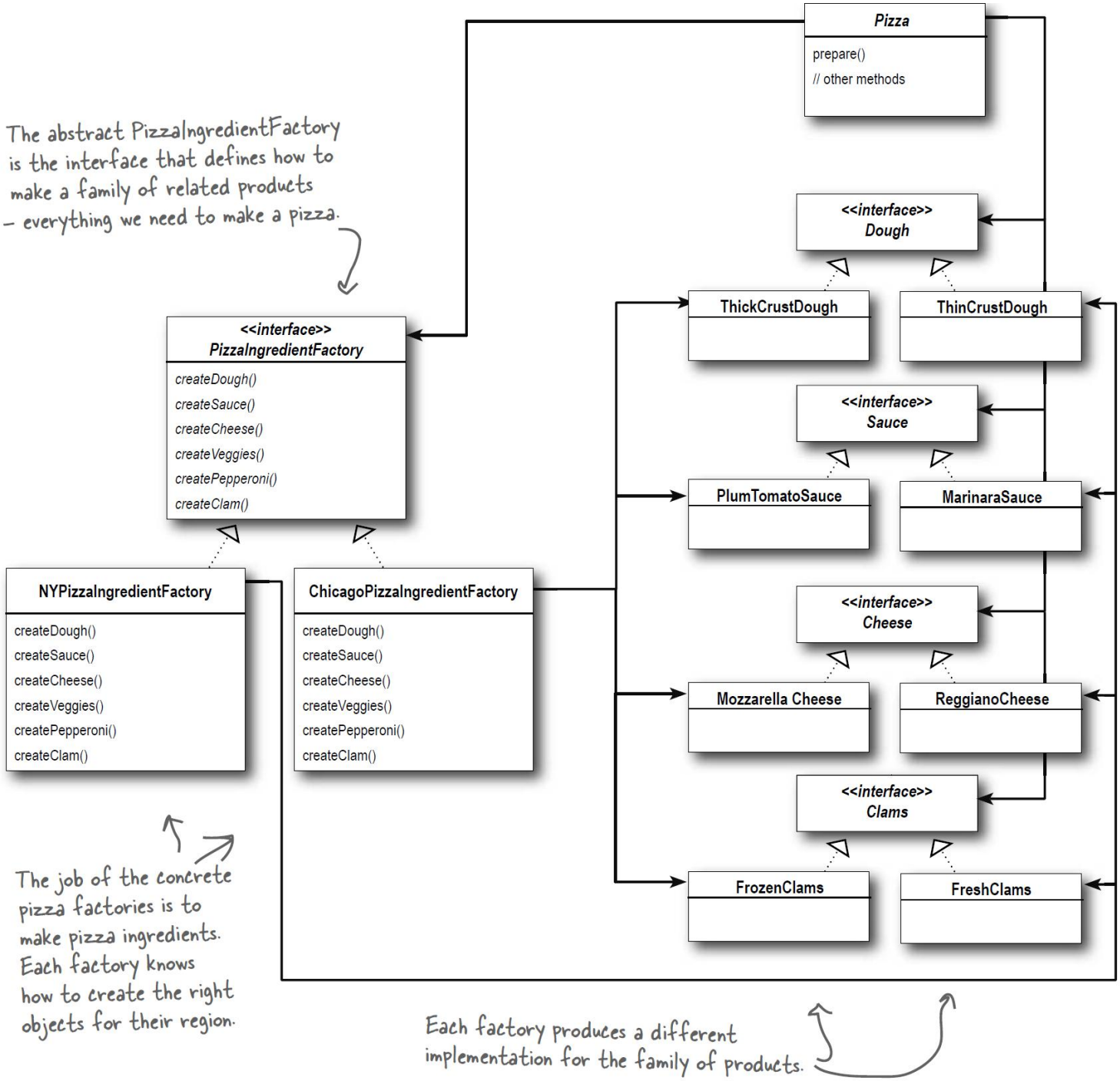
- Advantages
  - It isolates concrete classes.
  - It makes exchanging product families easy.
  - It promotes consistency among products. An application uses objects from only one family at a time.
- Disadvantage
  - Supporting new kinds of products is difficult.

Implementation

- Here are some useful techniques for implementing the Abstract Factory pattern.
  - Factories as singletons. An application typically needs only one instance of a ConcreteFactory per product family.
  - Creating the products. AbstractFactory only declares an interface for creating products. It's up to ConcreteProduct subclasses to actually create them. The most common way to do this is to define a factory method for each product.
  - Defining extensible factories. Basically, it is difficult to add new products. A more flexible but less safe design is to add a parameter to operations that create objects.

Let's Try! Problem: The Pizza Chain Again

- Requirements
  - A pizza chain (连锁店) has several stores. Each store purchases ingredient from the local ingredient factory. The factory provides dough (生面团), sauce, cheese, and clams ().
  - There are two factories: New York and Chicago.
    - New York factory provides: ThinCrustDough, MarinaraSauce, ReggianoCheese, and FreshClams.
    - Chicago factory provides: ThickCrustDough, PlumTomatoSauce, MozzarellaCheese, and FrozenClams.
- Use abstract factory, draw class diagram.


Pizza
prepare()
// other methods
The abstract PizzaIngredientFactory is the interface that defines how to make a family of related products – everything we need to make a pizza.
<<interface>>
PizzaIngredientFactory
createDough()
createSauce()
createCheese()
createVeggies()
createPepperoni()
createClam()
<<interface>>
Dough
ThickCrustDough
ThinCrustDough
<<interface>>
Sauce
PlumTomatoSauce
MarinaraSauce
<<interface>>
Cheese
Mozzarella Cheese
ReggianoCheese
<<interface>>
Clams
FrozenClams
FreshClams
NYPizzaIngredientFactory
createDough()
createSauce()
createCheese()
createVeggies()
createPepperoni()
createClam()
ChicagoPizzaIngredientFactory
createDough()
createSauce()
createCheese()
createVeggies()
createPepperoni()
createClam()
The job of the concrete pizza factories is to make pizza ingredients. Each factory knows how to create the right objects for their region.
Each factory produces a different implementation for the family of products.

Code

- net.dp.factory.abstractFactory.PizzaTestDrive

Summary of Factory Method and Abstract Factory

- Factory Method - Define an interface for creating **an** object, but let subclasses decide which class to instantiate. Factory Method lets a class defer (推迟) instantiation to the subclasses.
- Abstract Factory - Provide an interface for creating families of related or depedent objects without specifying their concrete classes.