

# Fábrica Abstracta (Abstract Factory)

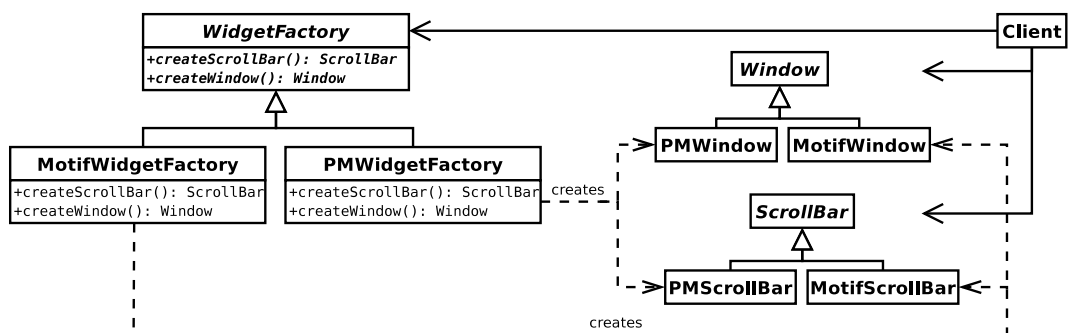
- *Patrón de Creación*

- *Propósito*

Interfaz para la creación de familias de objetos relacionados sin especificar sus clases concretas

- *Motivación*

- Interfaz de usuario con múltiples *look-and-feels* (Motif, Presentation Manager, Windows, ...)
- Cada look-and-feel define su propio juego de componentes (*widgets*: botones, ventanas, barras de desplazamiento, ...)
- Problema: Por razones de portabilidad el resto del sistema no debe hacer referencia a componentes concretos
- Solución: Clase abstracta con un método de fabricación para cada componente (*fábrica abstracta*)



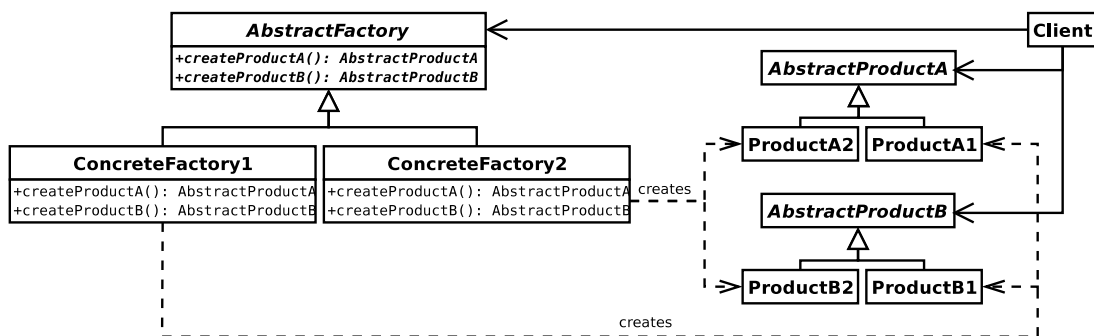
## ■ *Motivación (cont.)*

- Componentes...
  - Clase abstracta para cada componente
  - El cliente crea componentes a través de la fábrica abstracta
- La fábrica de componentes garantiza el uso consistente de componentes de un mismo look-and-feel

## ■ *Aplicabilidad*

- Un sistema debe ser independiente de la forma en que sus productos son creados, compuestos, y representados
- Un sistema debe ser configurado con una de muchas familias de productos disponibles
- Una familia de productos son diseñados para su uso conjunto, y se requiere asegurar este uso conjunto
- Se desea proporcionar una biblioteca de productos presentando su interfaz, pero no su implementación

## ■ Estructura



## ■ Participantes

- Fábrica Abstracta (AbstractFactory)
 

Declara interfaz para las operaciones que crean productos abstractos (métodos de fabricación)
- Fábrica Concreta (ConcreteFactory)
 

Implementa los métodos de fabricación de productos concretos
- Producto Abstracto (AbstractProduct<sub>i</sub>)
 

Declara la interfaz utilizada por el cliente para un tipo de producto concreto
- Producto Concreto (ConcreteProduct<sub>i,j</sub>)
 

Define un producto creado por el método de fabricación de una fábrica concreta

Implementa la interfaz Producto Abstracto *i*
- Cliente (Client)
 

Usa sólo las interfaces declaradas por Fábrica Abstracta y Productos Abstractos

## ■ *Colaboraciones*

Normalmente se crea una única Fábrica Concreta, que se encarga de crear los productos concretos

La Fábrica Abstracta difiere la creación de productos a sus subclases

## ■ *Consecuencias*

- Aísla clases concretas (no aparecen en el código del cliente)
- Facilita el intercambio de familias de productos
- Simplifica consistencia entre productos
- Difícil añadir nuevas clases de productos

## ■ *Implementación*

- Fábricas usualmente Instancia Única
- Creación de productos
  - Método de Fabricación (normal)
  - Prototipo (menos habitual, aunque evita extender Fábrica Abstracta)
- Flexibilizar fábrica mediante parametrización de método de fabricación
  - Método de fabricación puede crear *distintos tipos de componentes* en base a los parámetros del método
  - El cliente debe realizar una conversión tras crear el producto (*downcasting*)
  - (En la mayoría de los lenguajes) no hay ninguna relación formal entre valores de los parámetros y productos creados (salvo código fuente!)