Patrón Factory Method

DigitalHouse >
Coding School

Certified Tech Developer
The Ultimate Degree

Patrón Factory

El patrón de diseño **Factory** es uno de los principales patrones de diseño y uno de los más utilizados en la mayoría de los lenguajes de programación en la actualidad.

Tiene dos variaciones:

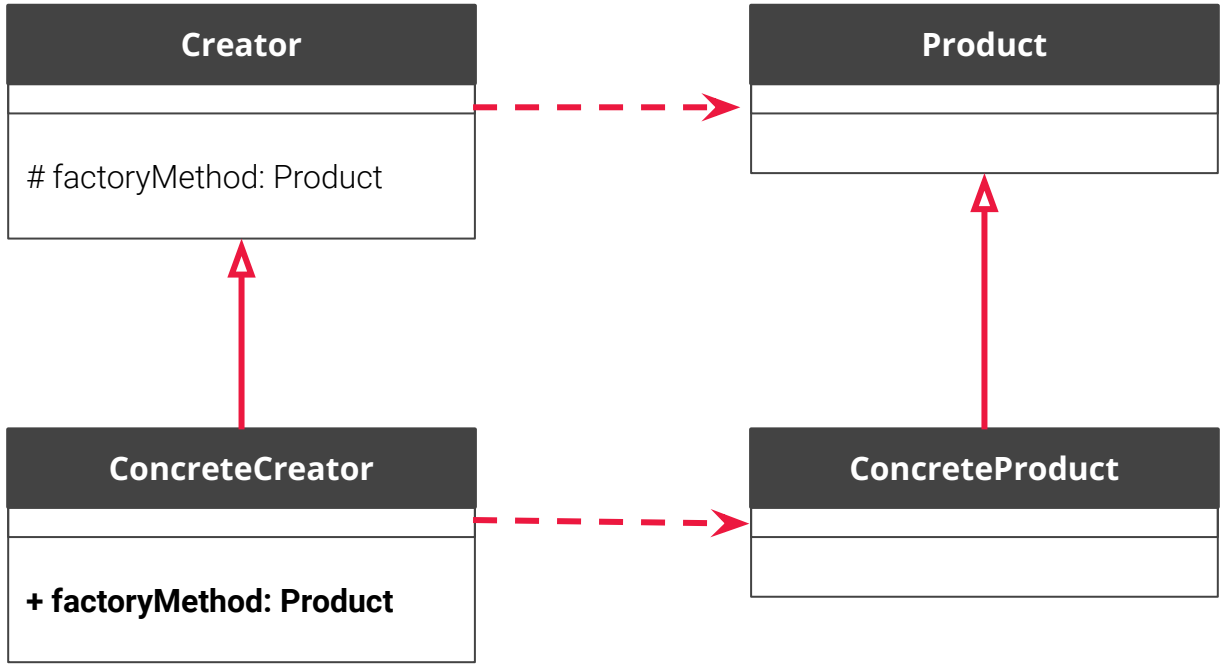
- ***Factory Method***
- ***Abstract Factory***

Variaciones

Factory Method	Abstract Factory
Es un patrón que define una interfaz para crear un objeto, pero permite que las subclases decidan qué clase instanciar.	Es un patrón que proporciona una interfaz para crear familias de objetos dependientes o relacionados sin especificar sus clases concretas.

Factory Method

UML

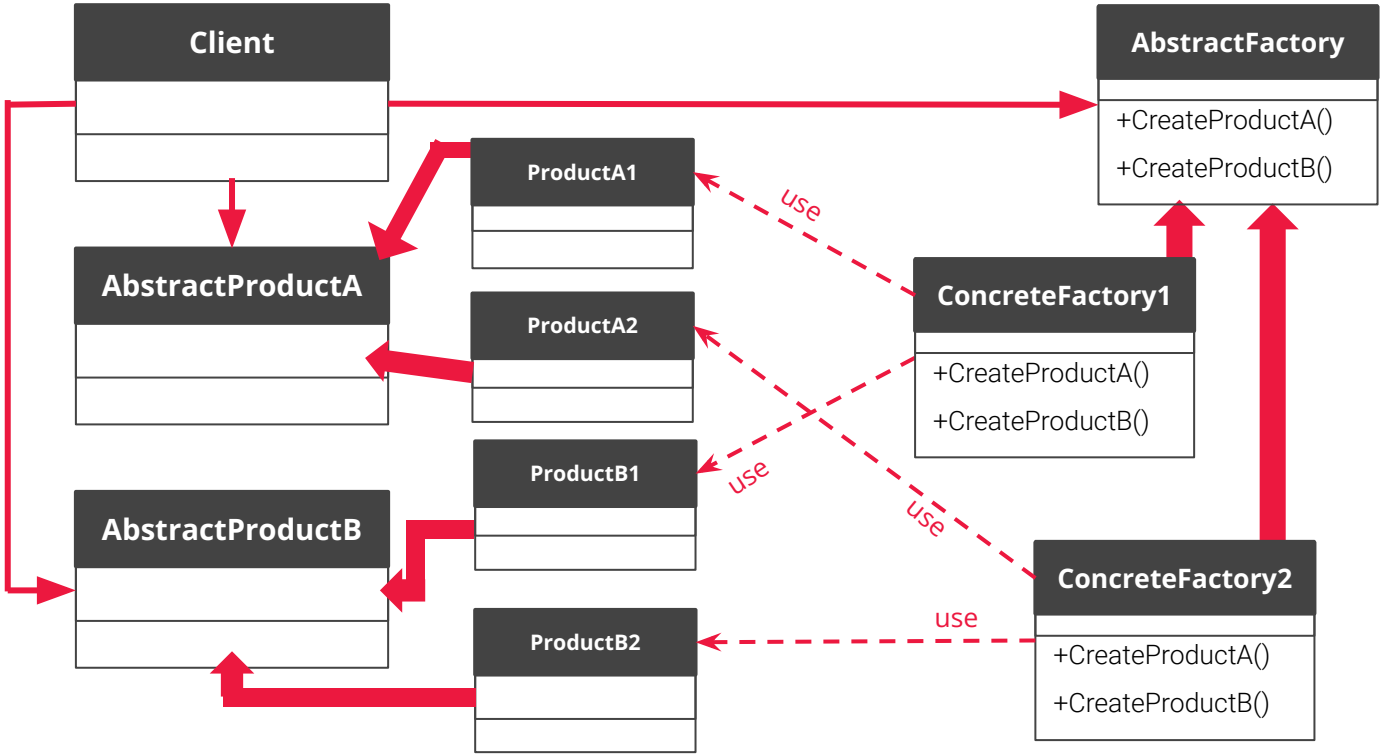


Factory Method

- **Creator (Creador abstracto):** declara el ***Factory Method* (método de fabricación)** que retorna el método de la clase **Producto**. Este elemento también puede definir una implementación base que devuelve un objeto de la clase **ConcreteProduct**.
- **ConcreteCreator (Creador concreto):** sobreescribe el método que fabrica el **Product** y nos permite instanciar el **ConcreteProduct** sin hacer referencia directa al mismo.
- **Product (Produto abstracto):** define una interfaz para los objetos creados por el ***factory method***.
- **ConcreteProduct (Producto concreto):** representa una implementación para la interfaz del producto.

Abstract Factory Method

UML



Abstract Factory

- **AbstractFactory:** Su propósito es declarar métodos de creación de tipo ***AbstractProduct***, que son implementados por una clase de tipo ***ConcreteFactory***, que hereda o implementa a ***AbstractFactory***.
- **AbstractProduct:** Declara métodos implementados por clases de tipo ***ConcreteProduct***. *ConcreteFactory* crea internamente un objeto de tipo *ConcreteProduct*, pero este objeto se devuelve como *AbstractProduct.*
- **ConcreteFactory:** Implementa los métodos declarados en *AbstractFactory*, creando un objeto de tipo ***ConcreteProduct*** y devolviéndolo como *AbstractProduct.*
- **ConcreteProduct:** Es la clase que especifica la instancia correcta a crear. Implementa los métodos declarados en *AbstractProduct.*

Conclusión

El propósito del patrón Factory es **crear objetos**, por lo que se considera un patrón de creación.

Básicamente, la lógica de creación está encapsulada dentro de la fábrica (*FactoryMethod*) y se proporciona un método que devuelve un objeto (Metodo **Factory** predeterminado) o la creación del objeto se delega a una subclase (método **Abstract Factory** predeterminado).


DigitalHouse>
Coding School