Factory Method Pattern

|  |  |
| --- | --- |
| **Tipo** | *Creazionale* |
| **Partecipanti** | *Product, ConcreteProduct, ConcreteFactory* |
| **Utilità** | *Molto utile* |

Nella programmazione ad oggetti, il **Factory Method** è uno dei design pattern fondamentali per l'implementazione del concetto di *factory*. Come altri pattern **creazionali**, esso indirizza il problema della creazione di oggetti senza specificarne l'esatta classe.

Questo pattern raggiunge il suo scopo fornendo un'interfaccia per creare un oggetto, ma lascia che le sottoclassi decidano quale oggetto istanziare.

L'utilizzo di questo pattern consiste nella creazione di una classe *factory* che gestirà la creazione di classi che implementano la stessa astrazione (interfaccia o classe astratta). Ciò significa che, se c'è un'interfaccia definita per diverse sottoclassi, la classe factory può creare qualsiasi di queste sottoclassi a seconda della logica passata ad essa.

Facciamo un esempio: creiamo una factory chiamata *BurgerFactory* che prenderà in input il parametro *typeOfBurger* (per esempio Pollo o Manzo). Il *BurgerFactory* deciderà quale tipo di oggetto Burger dovrà creare.

Supponiamo di avere un'interfaccia chiamata **IBurger** che sia il *ChickenBurger* sia il *BeefBurger* implementano, allora la classe BurgerFactory restituirà un oggetto di tipo *IBurger*. Il client non è a conoscenza di quale oggetto *IBurger* verrà creato e restituito. In questo modo riusciamo ad isolare il client dallo specifico oggetto ritornato aumentando la flessibilità e il riutilizzo del codice.