

Il **Factory Method** è un pattern creazionale che definisce un'interfaccia per creare un oggetto, ma lascia alle sottoclassi la decisione di quale classe istanziare. Consente a una classe di delegare l'istanziamento di oggetti a sottoclassi.

Componenti del Factory Method

- 1. **Creator (Creatore):** Una classe che dichiara il metodo factory, che ritorna un oggetto del tipo ``Product``.
- 2. **ConcreteCreator (Creatore Concreto):** Sottoclassi che implementano il metodo factory per creare oggetti ``ConcreteProduct``.
- 3. **Product (Prodotto):** Un'interfaccia o una classe astratta per oggetti che il factory method crea.
- 4. **ConcreteProduct (Prodotto Concreto):** Implementazioni specifiche del ``Product``.

Esempio con Pokémon

Immagina di voler creare diversi tipi di Pokémon (Pikachu, Charmander) utilizzando il Factory Method.

Interfaccia Product

javaCopia codice

```
public interface Pokemon {
    void attacca();
}
```

ConcreteProduct

javaCopia codice

```
public class Pikachu implements Pokemon {
    @Override
    public void attacca() {
        System.out.println("Pikachu usa Attacco Elettrico!");
    }
}

public class Charmander implements Pokemon {
    @Override
    public void attacca() {
        System.out.println("Charmander usa Attacco di Fuoco!");
    }
}
```

Creator

javaCopia codice

```
public abstract class PokemonFactory {
    public abstract Pokemon creaPokemon();
}
```

ConcreteCreator

javaCopia codice

```
public class PikachuFactory extends PokemonFactory {
    @Override
    public Pokemon creaPokemon() {
        return new Pikachu();
    }
}

public class CharmanderFactory extends PokemonFactory {
    @Override
    public Pokemon creaPokemon() {
        return new Charmander();
    }
}
```

Applicazione Principale

javaCopia codice

```
public class MainApp {
    public static void main(String[] args) {
        PokemonFactory pikachuFactory = new PikachuFactory();
        PokemonFactory charmanderFactory = new CharmanderFactory();

        Pokemon pikachu = pikachuFactory.creaPokemon();
        Pokemon charmander = charmanderFactory.creaPokemon();

        pikachu.attacca(); // Output: Pikachu usa Attacco Elettrico!
        charmander.attacca(); // Output: Charmander usa Attacco di Fuoco!
    }
}
```