

Factory Method

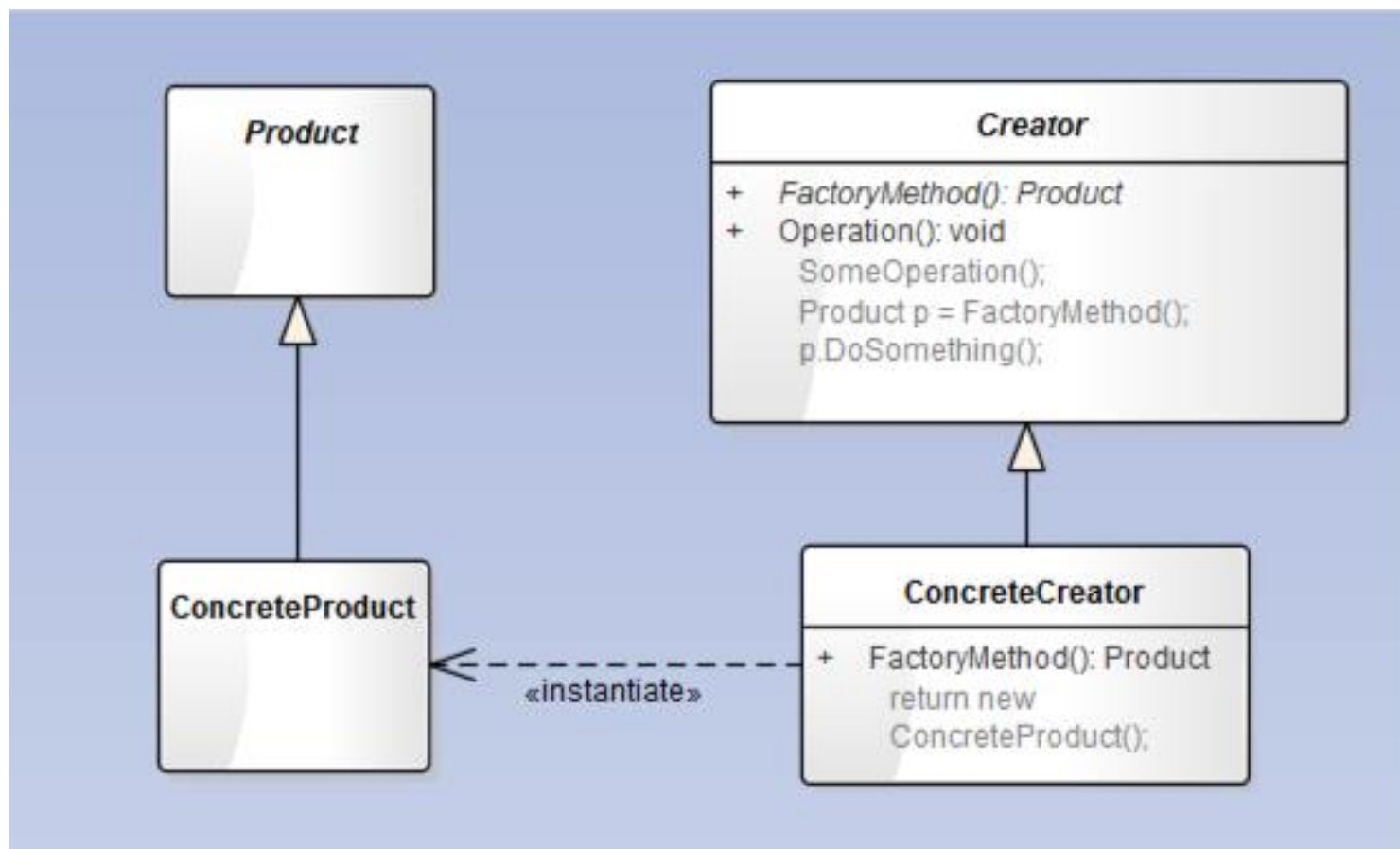
- 當知道建立某種物件的時機，但是不知道如何建立該物件的邏輯，此時，可以透過建立一個虛擬的Factory Method，把建立物件的邏輯的時間點遞延至實際開發時，由後續的開發者來處理

問題描述

在進行Framework設計時，Framework的Designer不可避免會遇到一個議題：我知道「何時」要產生某個類型的物件，並且針對他進行操作，但是到底該物件是什麼？在開發時並不知道。要如何才能解決這個問題呢？

解決方案

建立Creator跟Product兩個Abstract Class。
在Creator中，建立一個Factory Method的Abstract Method。
在Creator的Template Method(Concrete Method)中，直接使用Factory Method產生Product的Instance，然後針對該Product進行操作。



Creator與Product必然有關係，或是Association關係，或是Whole-Part關係，或是Dependency關係。

Creator與Product的行為必須抽象出來，並且在Creator中，必然有一個Template Method去使用到Product，為了要在Design階段就可以使用Product的Instance，因此才需要在Creator中設計Factory Method。

Factory Method產生的是Product的Instance，同時，Creator也只能使用Product的相關Operation。

案例描述

傳統的ERP程式中，有許多的維護(Maintain)性質的使用案例，如維護客戶資料，維護廠商資料，維護產品資料....等。

在這些維護性質的使用案例中，大部份都是提供四大類的Function，也就是傳統說的C(Create), R(Read), U(Update)以及D>Delete)。

針對Read的功能來說(假設取得所有列表)，其主要的操作邏輯為：

1. 建立存取的Dao
2. 請Dao進行存取
3. 回傳處理的結果

如何設計一套可以進行這些處理的Framework，讓日後的開發者在使用時可以更為簡易地新增新的維護性質的功能？

維護資料功能

