

# Prototype

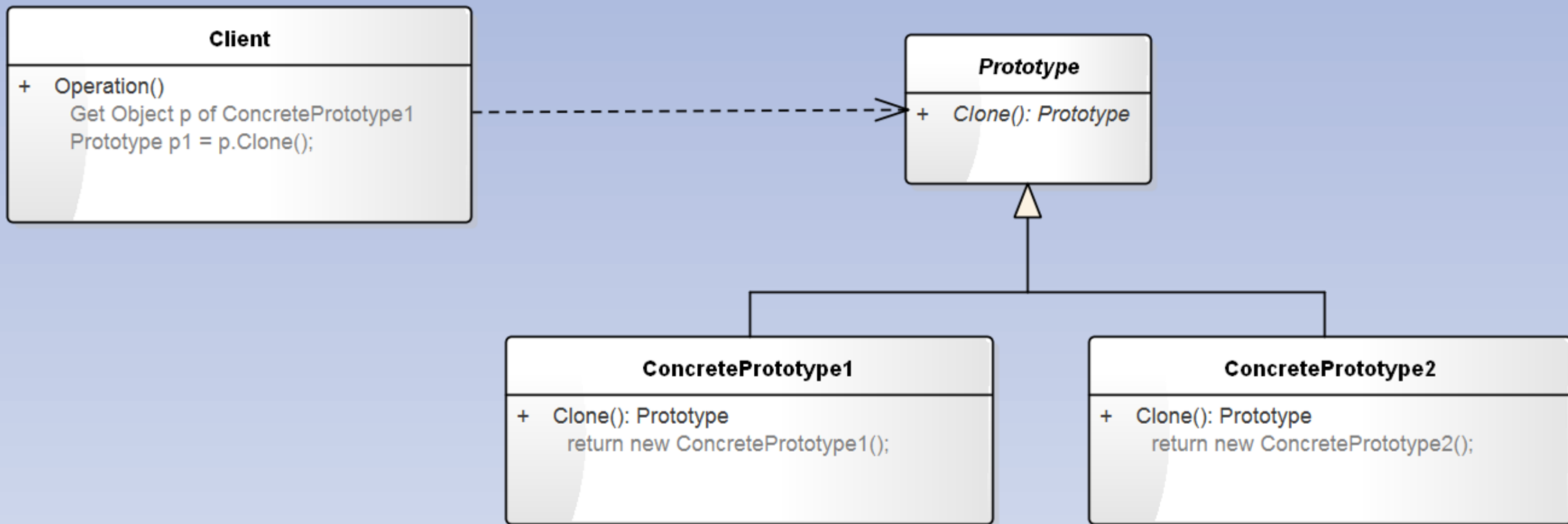
- 在Class的繼承家族中，不需要透過Subclass的建構子，直接透過Parent的clone產生Instance
- 若Class階層屬於資料型別的Class階層，可以透過Prototype來隱藏Data的結構細節，讓建構資料物件可以比較容易

# 問題描述

在Framework的設計中，由於可視性(Visibility)的問題，Client也許只知道公用介面，而不了解實際的Concrete Class。為了提供一個統一的方法讓Client端在不需知道Concrete Class的情況下，就可以產生物件，此時，可以利用Prototype來進行實作。

# 解決方案

定義一個Prototype介面，讓Concrete Prototype去實現Clone的Method  
Client端只需認識Prototype介面，透過Clone Method建構所需的Concrete Prototype的物件



Prototype樣式的重點在於使用clone method取代原本的Constructor

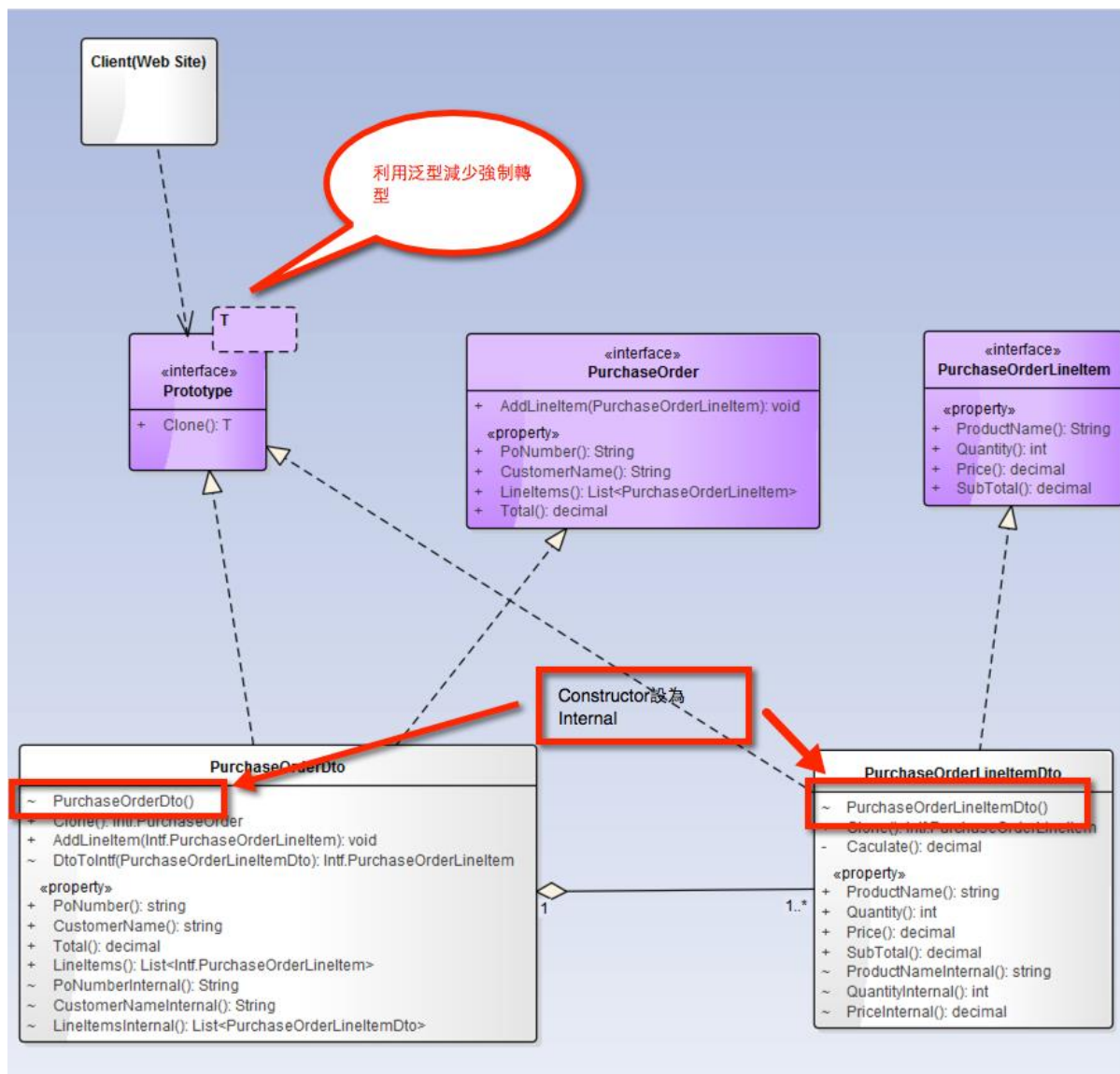
Prototype有兩種不同的實作應用：

- 只Copy物件結構，產生一個空的物件
- 同時Copy物件結構以及資料，產生一個完整的附件

以原始的Prototype的精神來說，第一種實作應用似乎是比较適當的

- 完美封裝隔絕
  - ◆ RingleFramework的設計中，所有的資料交換物件 (Data Transfer Object, DTO)都只透過介面來傳遞，當前端的MVC要新增一個特定的物件傳入到應用程式層，需要如何處理？
- Copy-Paste
  - ◆ RingleERP中，採購資料可以透過複製方式Copy一張既有的採購到新的採購，應該要如何實現這個複製的過程？

# 案例1 - 完美封裝



# 案例2 – Copy-Paste

