

# Flyweight

- 把物件的狀態分為「不變的」與「可變的」，預先建立「不變的」物件，讓其可以在整個系統環境中被共享
- 「可變的」物件讓Client自行去Hold住，其生命週期與Client相同
- 「不變的」物件的生命週期則與系統相同
- 這也是屬於「資源管控」的樣式

# 問題描述

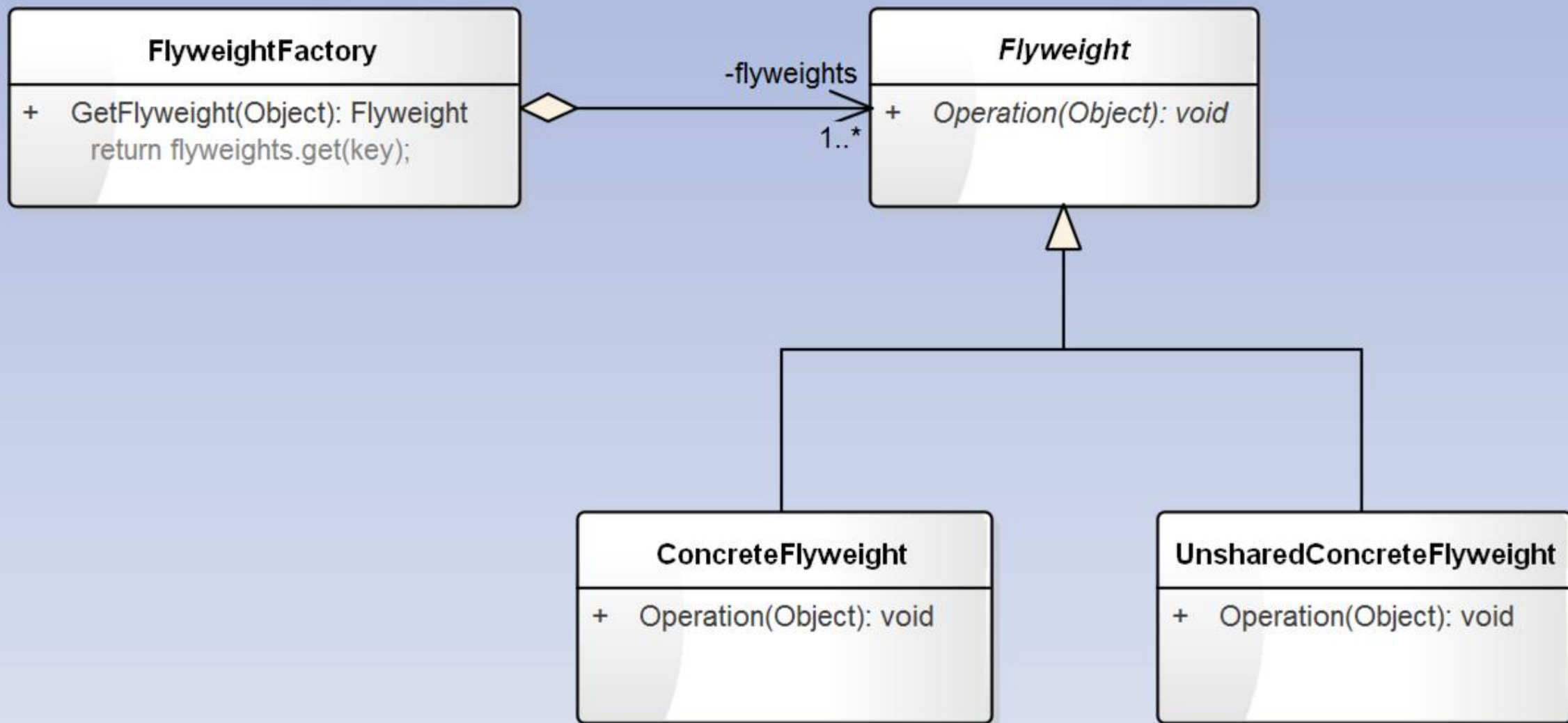
根據物件的特性，大致可以分成兩大類，一類的物件一旦誕生後，無論大環境如何改變，該類型的物件是恆久不變的(*immutable*)；另一類的物件，其狀態會不斷地改變。

針對第一種類型的物件，是否有辦法在其誕生後，就由系統Handle住其狀態，如此，可以大幅度減少產生物件的Effort，讓系統更有效率(*efficiency*)。

# 解決方案

利用一個輕量工廠(FlyweightFactory)負責Hold住所有不變類型的物件。  
至於會不斷變化的物件，則也透過輕量工廠產生，但不Hold住他。

# 結構



通常來說，Flyweight代表的是一個抽象化的介面，並且根據該抽象化的介面，把其實作(ConcreteObject)分成不變的(ConcreteObject)以及可變的(UnsharedConcreteObject)。  
FlyweightFactory中的flyweights，則只包括不變的部分。  
在應用上，通常可以用在Pooling以及Cache的實作上。

# 案例描述

在許多系統的開發中，Dropdownlist中的值(Value)與呈現文字(Text)會放在Code Table中。

但仔細分析後，會發現Code其實分成兩大類，其中一大類是不大會改變的，而另一大類則是會根據某個主要的Code類別而改變其實際的值的集合。

以地址舉例來說，地址中的城市別，基本上是不大會改變的，而當選了城市別後，呈現該城市的地區(區, 鄉, 鎮, 市)，則會根據城市別而有所變動。

能否設計一個模型，讓存取DB的次數最少，以發揮最佳的效能？

# Code處理

