

1. 類別之間的相依性過高：

要降低類別間的相依性，最基本的原則，就是盡量避免某個類別的名稱出現在另一個類別的原始碼中，且盡可能讓介面中所含的成員越少越好，這樣才能避免此介面和其他類別相介接的機會。

當環境一更新希望所有有關心環境的客戶都可以自動收到通知的狀況，這樣物件間的一對多關係適合以觀察者模式來實行，環境實行被觀察者；而客戶實行觀察者，如此一來觀察者不必隨時要去檢查環境是否有更新，因為當環境更新時會自動通知客戶們。

2. 如果有新的客戶加入，或是現有的客戶要退出，必須要修改 EnvironmentData 的程式碼：

由於新增或減少客戶的過程中，EnvironmentData 的行為並沒有改變(都是接收資料後通知客戶，改變的只有客戶數量)，因此我在

EnvironmentData 這個 class 裡增加了一個以 Observer 為 object 的

ArrayList 用於建立客戶清單，紀錄已註冊的客戶。因此，每當有舊客戶要退訂或新客戶要訂閱時，只需要讓 EnvironmentData 取消註冊

此位舊客戶，同時 EnvironmentData 會將此位舊客戶從客戶清單

remove 掉；讓新客戶訂閱 EnvironmentData，同時 EnvironmentData 會註冊此位新客戶並將之新增進客戶清單。

3. 通知客戶的方法沒有統一規範，管理員必須知道每個客戶的方法名稱 (updateData、recieveData、update)：

EnvironmentData 身兼管理員的角色，在起初新增客戶時便紀錄此客戶的要求，當資料被更新的時候，就告知所有客戶可以更新的資料，僅僅是告知 EnvironmentData 有更新，讓客戶自己決定要取得那些資料。

這樣便能完成假如客戶 A 要溫度濕度風速而客戶 B 只要溫度也無需特別修改方法內容的要求。

4. 別的應用上如果需要使用 EnvironmentData 類別的資料但不需要管理員功能時，必須大幅修改 EnvironmentData 的程式碼。：

在 EnvironmentData 這個 class 裡我有各別寫 get 溫度、濕度、風速的方法假如別的應用上如果需要使用 EnvironmentData 類別的資料，只需要透過以上的方法忽略管理員功能就能取得 EnvironmentData 類別的資料，不必大幅修改 EnvironmentData 的程式碼。

5. 每個客戶都要定義一個變數去紀錄，如果要逐一通知會非常麻煩：

如同第二點，在 EnvironmentData 這個 class 裡增加了一個以 Observer 為 object 的 ArrayList 用於建立客戶清單，當資料被更新的時候，很輕易就能透過一個方法一個迴圈告知所有客戶可以更新的資料，並讓他們透過一開始新增時紀錄的要求資料讓客戶自行取得他們所需的資料。