

Bridge

P

- 將抽象概念與其實作進行分離，讓兩者可以分開擴充

問題描述

在Framework的設計上，「預留擴展」(Expandable)是「充要條件」
舉例而言，像Spring Framework或是J2EE，使用者可以透過設計EJB或是Spring Bean來實作服務
這是怎麼做到的呢？

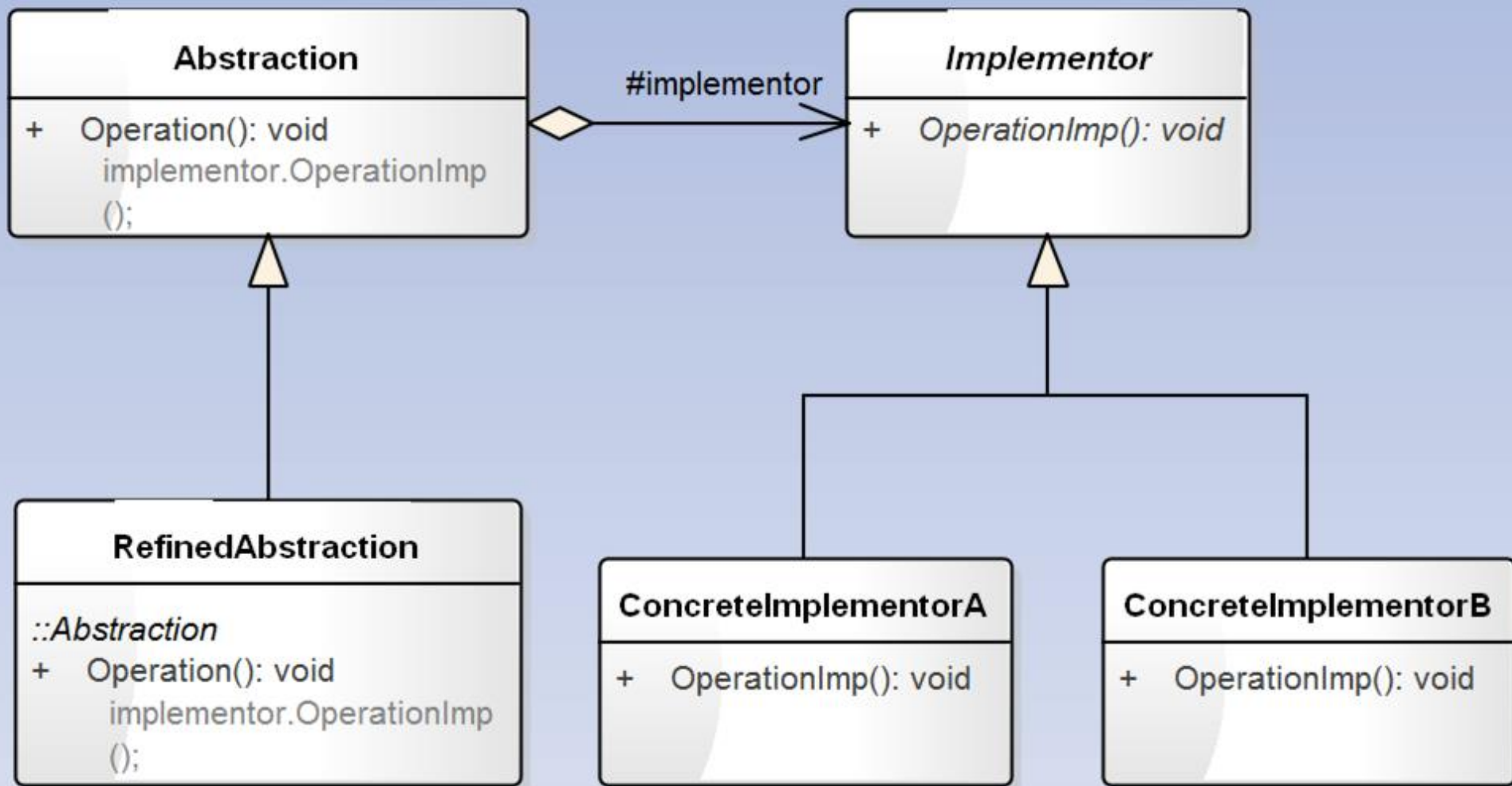
解決方案

透過介面與實作的分離，可以讓擴展變的可能

Spring Bean透過使用者自行定義介面與實作，再透過設定在Runtime由Framework自行組裝兩者，如此一來，一個Framework可以透過開發者讓其更為完整

「You complete me!!」

結構



Abstraction是屬於某個特定的問題領域中的重要概念，要注意的是，在Bridge Pattern中，Abstraction其實是一個「Concrete Class」，其Operation是包括實作的Implementation定義了Abstraction中的Operation所有需要使用到的行為(行為分解)

RefinedAbstraction並不一定要覆寫Abstraction的Operation，通常而言，若有需要擴充Abstraction的行為時，才需要有RefinedAbstraction(該擴充的行為，必須使用到Abstraction中定義的Operation)

案例描述

PXHome是一個線上購物平台，希望能夠提供客戶可以在下單後，即時查詢目前訂購貨品的狀態。

貨品狀態有幾種不同情況：

1. 若還在PXHome的倉儲內，可以透過PXHome的資料庫知道目前的商品狀態
2. 若已經出貨了，則需要透過Shipper所提供的狀態碼來判斷貨品在途的狀況
 1. 目前PXHome有兩家Shipper，分別是黑貓以及郵局，兩個Shipper的貨品狀態碼不同

問題：

是否能提供一個方式，讓上述的幾種狀況都可以只提供統一的狀態碼？

查詢訂單狀態

