

觀察者模式——Youtube 訂閱機制

難度：★☆☆☆☆

需求

你要簡單撰寫一個模擬 Youtube 頻道 (Channel) 和 頻道訂閱者 (Channel Subscriber) 之間互動的程式。

1. Youtube 頻道和頻道訂閱者都擁有名稱 (name)。
2. 如果用戶對某個 Youtube 頻道有興趣，可以對其訂閱 (Subscribe)，訂閱之後，此用戶會成為該頻道的訂閱者。
3. 每當 Youtube 頻道有新的影片 (Video) 上傳 (Upload) 時，所有該頻道的訂閱者都會**接獲新影片通知**。
 1. 影片的屬性包含：標題 (Title)、敘述 (Description) 和影片長度 (Length)。
4. 每位訂閱者接獲新影片通知之後的行為都不一樣。

模擬測資

佈局 (Given)

1. 兩個 Youtube 頻道：PewDiePie 和 水球軟體學院
2. 第一位 Youtube 用戶：水球
 1. 每當水球接獲新的影片通知時，他會判斷：如果影片的長度時間 \geq 三分鐘，那麼水球就會對其影片按讚 (Like)，否則置之不理。
3. 第二位 Youtube 用戶：火球
 1. 每當火球接獲新的影片通知時，他會判斷：如果影片的長度時間 \leq 一分鐘，那麼火球就會立刻對該頻道解除訂閱 (Unsubscribe)。

測試 (When)

1. 水球訂閱 PewDiePie 和 水球軟體學院
2. 火球訂閱 PewDiePie 和 水球軟體學院
3. 水球軟體學院上傳一部影片：標題："C1M1S2"、敘述："這個世界正是物件導向的呢！"、影片長度：4 分鐘。
4. PewDiePie 上傳一部影片：標題："Hello guys"、敘述："Clickbait"、影片長度：30 秒。

5. 水球軟體學院上傳一部影片：標題："C1M1S3"、敘述："物件 vs. 類別"、影片長度：1 分鐘。
6. PewDiePie 上傳一部影片：標題："Minecraft"、敘述："Let's play Minecraft"、影片長度：30 分鐘。

預期結果 (Then)

水球 訂閱了 水球軟體學院。
水球 訂閱了 PewDiePie。
火球 訂閱了 水球軟體學院。
火球 訂閱了 PewDiePie。
頻道 水球軟體學院 上架了一則新影片 "C1M1S2"。
水球 對影片 "C1M1S2" 按讚。
頻道 PewDiePie 上架了一則新影片 "Hello guys"。
火球 解除訂閱了 PewDiePie。
頻道 水球軟體學院 上架了一則新影片 "C1M1S3"。
火球 解除訂閱了 水球軟體學院。
頻道 PewDiePie 上架了一則新影片 "Minecraft"。
水球 對影片 "Minecraft" 按讚。

設計需求

1. 對於需求 2 和 3 的實作，要遵守開閉原則 (Open-Closed Principle, OCP)；意即：在實作新的訂閱者時，Youtube 頻道必須對修改關閉、對擴充開放。

作答引導

靠自己的力量走到盡頭了嗎？那趕緊來從作答引導中，照著我的思維再走一次吧！**(爆雷警告)**

1. 針對需求 1：「你要簡單撰寫一個模擬 Youtube 頻道 (Channel) 和 頻道訂閱者 (Channel Subscriber) 之間互動的程式。」
 1. 看一下有哪些名詞，在 **Youtube** 領域中是重要的概念，將他們捕捉成類別吧！（Tip: 兩個類別）
2. 對需求 2：「每當 Youtube 頻道有新的影片 (Video) 上傳 (Upload) 時，所有該頻道的訂閱者都會接獲新影片通知。」
 1. 「上傳」是一個動詞，為可以對 Youtube 頻道的一個操作，你會如何 Model 它？
 2. 「上傳」操作會建立 Youtube 頻道和影片的關係，你覺得是何種關係？
 3. 「接獲新影片通知」也是一個動作，你會如何 Model 它？

4. 這一句需求敘述中是否存在著響應式行為的 Force？如果存在，響應的「事件」為什麼？
3. 你的初版 OOA 類別圖應該已經畫好了，你應該會有 Youtube Channel, Channel Subscriber 和 Video 三個類別，並且 Youtube Channel 和 Channel Subscriber 之間會有響應式行為，因此會有一條 1... 的關聯。
4. 對需求 3：「每位訂閱者接獲新影片通知之後的行為都不一樣。」
 1. 你認為這一句需求描述，形成了何者 Force？這道 Force 影響著哪一個類別？
5. 在設計需求中，對程式設計添加了需求：「對於需求 2 和 3 的實作，要遵守開閉原則 (Open-Closed Principle, OCP)；意即：在實作新的訂閱者時，Youtube 頻道必須對修改關閉、對擴充開放。」
 1. 在設計需求加入之後，你能否歸納目前程式設計中面對的 Forces？這些 Forces 中哪些衝突了彼此？（至少兩條）
 2. 觀察者模式是否能夠化解這兩條 Forces？
6. 是時候套用觀察者模式了，用依賴反轉之重構三步驟來化解這兩條衝突的 Forces 吧！
 1. **封裝變動之處 (Encapsulate what varies)**：（水球 & 火球訂閱後接獲新影片通知的行為不同。）
 2. **萃取共同行為 (Abstract common behaviors)**：萃取出一個表示水球和火球共同行為的介面吧！
 3. **委派 (Delegation)**
7. 套完模式之後，你應該就能夠在 Main method 中簡單的實體化 PewDiePie 和 _水球軟體學院，_以及_水球_和_火球_用戶，然後讓他們互動來建立關係吧！