

實作交友配對系統 ★

策略模式——你的配對策略是什麼呢？

難度：★☆☆☆☆

你正在開發一個交友配對系統 (Matchmaking System)。

1. 你手邊有一大群對象 (Individual) 資料，每一筆資料皆記載著以下欄位：
 - a. 編號 (ID)：正整數 (>0)，每位對象的編號都不同。
 - b. 性別 (Gender)：MALE 表示男生、FEMALE 表示女生
 - c. 年紀 (Age)：正整數；至少 18 歲。
 - d. 自我介紹 (Intro)：長度約 0~200 的文字。
 - e. 興趣 (Habits)：可以為多個興趣，每個興趣以 1~10 長度的文字表示，每個興趣之間以一個逗號隔開。例如：打籃球，煮菜，玩遊戲
 - f. 座標 (Coord)：以 (x, y) 表示該用戶所在的 x 軸和 y 軸的位置。
2. 你要撰寫一個配對系統 (Matchmaking System)，系統會幫每位用戶配對最適合他的用戶，而你的系統主要提供了兩種配對策略，並且未來會持續新增新的策略：
 - a. **距離先決 (Distance-Based)**：配對與自己距離最近的對象（距離相同則選擇編號較小的那位）。
 - i. 假設自己的座標為 (x, y) 而對象的座標為 (x', y') ，則距離公式為：
$$\sqrt{(y - y')^2 + (x - x')^2}$$
 - b. **興趣先決 (Habit-Based)**：配對與自己興趣擁有最大交集量的對象（興趣交集量相同則選擇編號較小的那位）。
3. 不過，也有用戶喜歡認識更多元的人，因此你的系統還要為你每個策略提供「**反向 (reverse)**」的實作版本，像是把距離先決的策略改成是配對與自己距離最遠的對象，而把興趣先決改成是配對與自己興趣擁有**最小交集量**的對象。

設計需求（非功能性需求）

由於客戶會不斷研發新的配對策略，我們希望系統能夠允許 Client (程式中系統的使用方，通常會是你的 **Main method**) 能在外部抽換系統的配對策略。

道館作答叮嚀

請務必模仿「景點 - 策略模式 | 基礎水平行為擴充」這部影片中的思路示範，要遵照底下思路來套用設計模式：

1. OOA 的圖除了建模之外，還要捕捉底下幾者：
 - a. 抓到至少一道「行為變動性」的 Force (以紅色箭頭或是便條紙表達 Force)，將之擺在行為變動之處 (如：該操作或是行為便條紙旁)。並用文字清楚說明：是哪一種行為引起變動？未來有可能會添加何等新行為？
 - b. 然後再抓到另外一道「擴充性」的 Force，並用文字清楚說明：在擴充「什麼行為」時，希望能不必修改「哪些類別」？
 - c. 接著抓上述兩道 Forces 的衝突點，並總結成一句 Problem (可以以便條紙簡單標註)
2. 接著在 OOD 時，套用設計模式來解決上述的 Problem，已經解決完的 Forces 將其顏色改成「綠色」。

至少先以「模仿」開始，到後面就會慢慢內化、了解其精髓。

如果沒有以「Forces → Problem → Pattern」的思路來套用設計模式的話，是不會通過道館測驗的唷。

進階挑戰題

你想得到如何遵守以下條件來來實踐需求 (3) —— 反向策略嗎？

1. 「配對系統類別/距離先決策略/興趣先決」類別不能直接耦合「反向 (Reverse)」的概念，也不能使用 boolean 欄位 (e.g., `reverse: boolean`) 或是參數 (e.g., `match(..., ..., reverse: boolean)`) 來區分正反向。
 - a. 以下這些都是被禁止的錯誤示範：
 - i. `new MatchmakingSystem(new DistanceBasedMatchmakingStrategy(), Reverse=True)` (X)

- ii. `matchMakingSystem.match(individual, Reverse=True)` (X)
- iii. `distanceBasedStrategy.reverse()` (X)
- iv. `new DistanceBasedStrategy(Reverse=True)` (X)
- v. `matchMakingSystem.reverse()` (X)

2. 不組合爆炸：意即你不能讓程式中出現攸關於「正向距離先決、反向距離先決、正向興趣先決、反向興趣先決」概念的四個類別，只要你有四個配對策略的類別，就是組合爆炸了，是不被允許的。

試試看吧！

作答要求

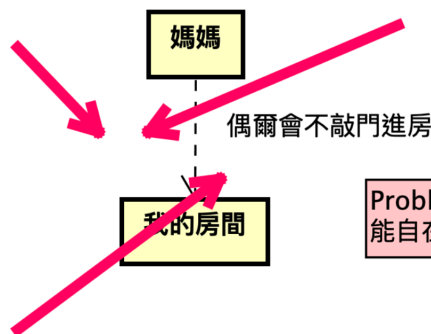
在你實際作答之前，我想要對你做一些要求，來幫助你儘早了解這門課的精華。

由於這一題是「策略模式」的題目，所以你肯定知道這一題就是要套「策略模式」。不過，在這門旅程中要學的是「精準設計」，所以你要趁在這一題比較簡單的題目中，就趕緊試試看「精準設計」的思路。

也就是說，你不能只是套用策略模式，你必須先「察覺 Forces (拘束)」，從需求和非功能性需求中，你辨識出了哪些已在策略模式的案例影片中學過的 Forces 呢？

請在 UML 上使用「紅色的箭頭」來繪製 Forces，把 Forces 放在相關的行為/結構附近，並將衝突的 Forces 定義成一道 Problem，用便條紙來表達 Problem，看起來就像這個樣子：

Force 3：我媽很強勢，鎖門一定會被罵， Force 2：我媽會不敲門突然進房，這使我必須時時刻刻保持謹慎，很累！
要求他要敲門也會被罵



Problem：到底該如何在不自鎖門、不要求我媽，且不被我媽發現的前提下能自在在我房間做我想做的事啊？

Force 1：我常常在房間中做一些不可告人的事，不想被家人發現。

如果你真的佐證了策略模式的必要之處，那你一定至少會在某些類別附近，察覺了至少兩道 Conflicting Forces，並基於這兩道 Forces 來定義 Problem。而這兩道

Forces 符合策略模式所解決的 Conflicting Forces，所以才證明了你能夠套用策略模式。

試試看吧！

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. 我們先來做 OOA（記得一定先自己試試看）：
 - a. 一句話一句話閱讀，「你正在開發一個**交友配對系統** (Matchmaking System)。」這一句話有幾個名詞？要不要考慮將此名詞捕捉進類別圖中？
 - b. 「你手邊有一大群對象 (Individual) 資料，每一筆資料皆記載著以下欄位：……」第一句話中有哪一個名詞是配對領域中重要的觀念？將他開成類別吧！然後這句話也指示著此類別中具備多項屬性。
 - c. 再來是需求 2：「你要撰寫一個配對系統 (Matchmaking System)，**系統會幫每位用戶配對最適合他的用戶……**」從第二句話來看，「配對」是否是一個系統的動作？
 - d. 我們可以從需求 2a 和 2b 發現**配對策略**的行為變動性，將此行為變動性的兩道行為變種記錄在便條紙中，作為「配對行為」的註釋。
 - e. 需求 3：「因此你的系統還要為你每個策略提供「反向 (reverse)」的實作版本……」正反向與否也多增添了一道行為變動的方向，將它加進「配對行為」的註釋便條紙中。
2. 如此一來就畫完 OOA 的類別圖囉！再來我們做 OOD：
 - a. 「配對行為」的註釋中描述著兩道行為變種：距離先決和興趣先決配對策略。結合設計需求 1，你察覺到了哪些 Conflicting Forces（至少兩道）？
 - b. 這組 Forces 之間哪裡衝突？請針對衝突之處，將其定義成一道 Problem。
 - c. 說明為什麼「策略模式」能解決這組 Conflicting Forces？
 - d. 怎麼套策略模式？首先**依賴反轉之重構三步驟**：
 - i. Encapsulate what varies：將 2a, 2b 的行為變種各開一個類別封裝起來。

- ii. Abstract common behaviors：觀察一下 2a, 2b 這兩個行為變種有什麼「共同行為」，創建一個介面 (Interface)——
MatchmakingStrategy 是一個好名字，並將此共同行為萃取至介面中。
- iii. Delegation / Composition：原本是 Matchmaking System 要負責實現配對策略，現在將此職責委派給你開出來的 MatchmakingStrategy 介面。
- e. 然後至程式碼中套 **依賴注入模式**——將 Matchmaking System 的依賴 (MatchmakingStrategy) 從 Matchmaking System 的建構子或是 setter 傳進去。
 - i. 所以你可以撰寫這種程式碼：

```
new MatchmakingSystem(new DistanceBasedMatchmakingStrategy())
```

提交挑戰

提交狀態

未提交

物件導向分析 (OOA)

上傳檔案

上傳你的 OOA 領域類別圖 (png, jpg, bmp, gif 檔案)

循序圖設計 (OOD)

上傳檔案

上傳你的 OOD 循序圖設計 (png, jpg, bmp, gif 檔案)

物件導向設計 (OOD)

上傳檔案

上傳你的 OOD 類別圖 (png, jpg, bmp, gif 檔案)

專案壓縮檔

上傳檔案

請將你專案的原始碼壓縮至 ZIP 中並且上傳。(只包含原始碼檔案就好囉,不要包含編譯後的檔案。)(zip 檔案)

提交

放棄挑戰