

設計想法：

寫了一個抽象類別 BeverageMaker，讓各種飲料製作機皆能繼承它，覆寫裡面的方法或做使用，形成多型的應用，我將 prepare()規範了製作飲料的過程放在抽象類別 BeverageMaker 這樣就能避免保證未來某個 subclass 不小心把製作流程順序搞錯了，因為沒有機會讓它能調整順序。

因此繼承的 subclasses 可以按照我們制訂的製作流程規範走，卻又能彈性的自己決定每個步驟該做什麼，也能選擇要不要加冰塊（如果不加冰塊便在覆寫後的方法裡不給予任何作為）。

例如題目給的舉例：CoffeeMaker

首先將他繼承抽象類別 BeverageMaker，覆寫裡面的 brew()、addIce() 和 addCondiments()，自己選擇實作細節和要不要實作，而 boilWater() 和 pourInCup()都一樣，因此每個繼承抽象類別 BeverageMaker 的 class 就不覆寫。

在 Main class 裡我們以 BeverageMaker 抽象類別 new 一個我們想建立的飲料製作機 b 並直接呼叫 BeverageMaker 裡的 prepare()，b.prepare() 他會直接按照製作流程印出製作飲料步驟，並且依照我們前面 new 的飲料製作機的 class 印出覆寫的方法，區別了各種飲料製作機的 brew()和 addCondiments()也能避免製作流程順序搞錯並且彈性於步驟

的可以可無。

最後，每個 class 一個檔案將之分開放進一個叫 assignment2 的資料夾

並且在每個程式之開頭加上 package assignment2 ;

這樣代表每個檔案皆包在 assignment2 裡面，也不用特別 import。