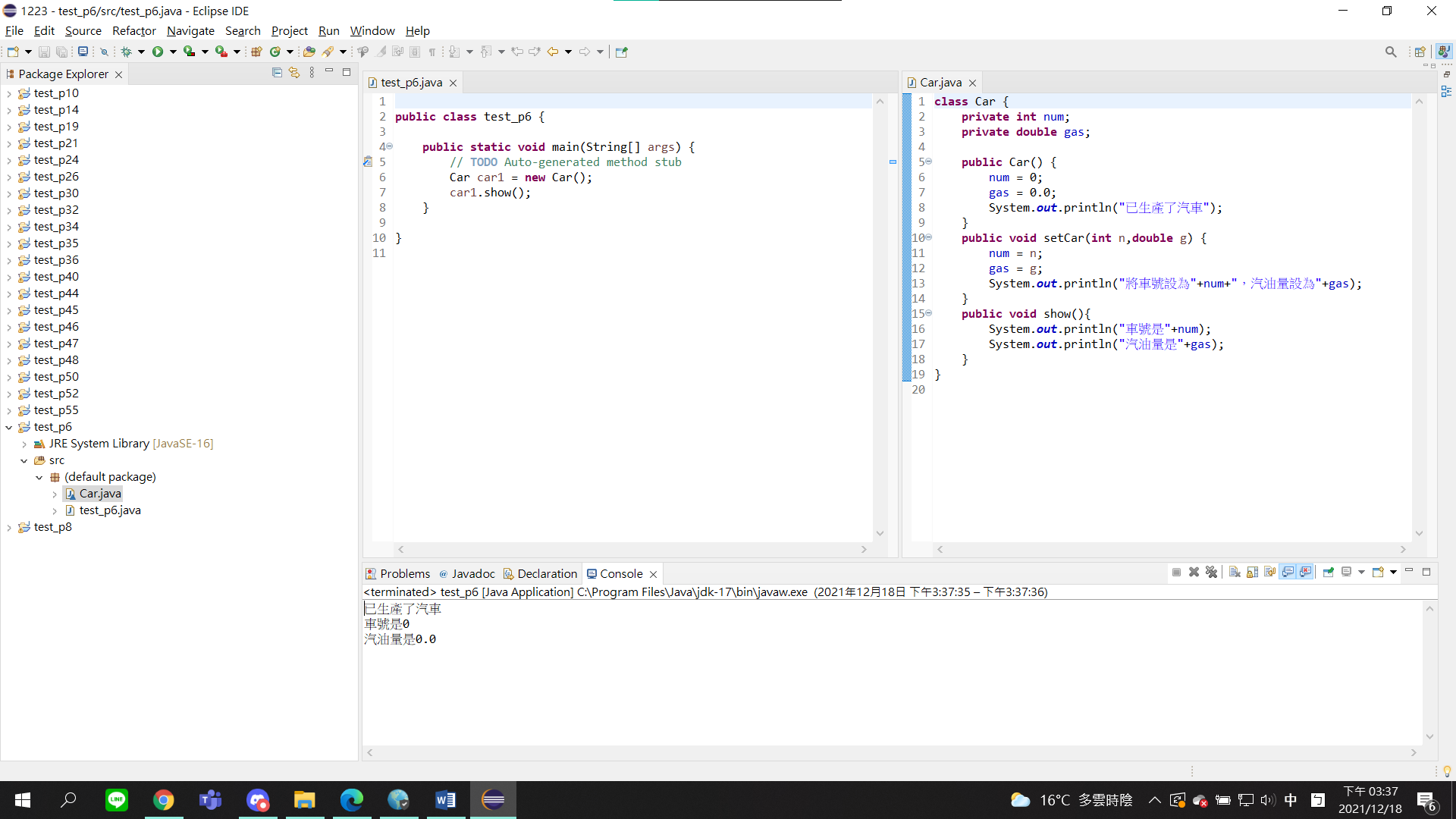
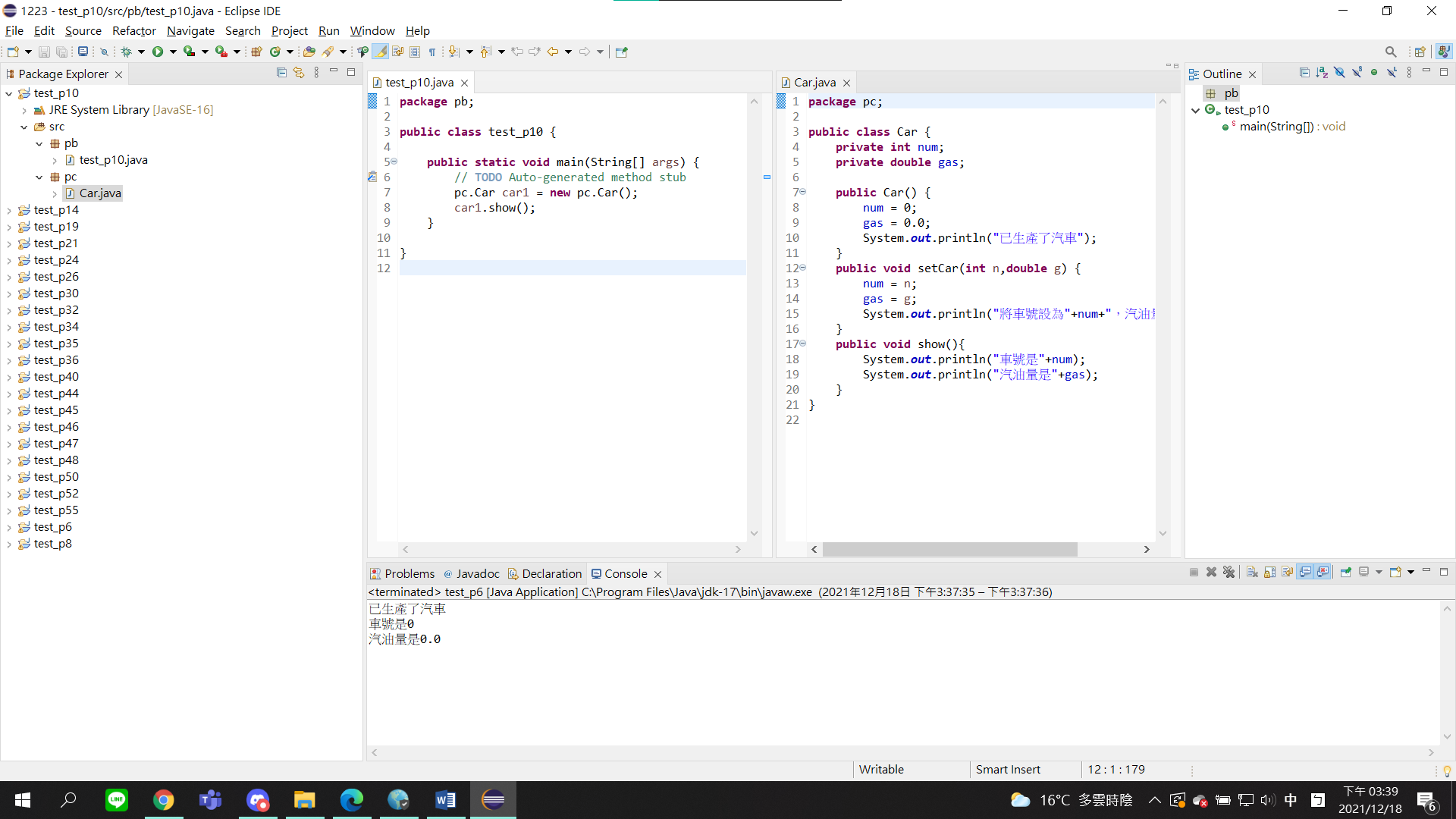
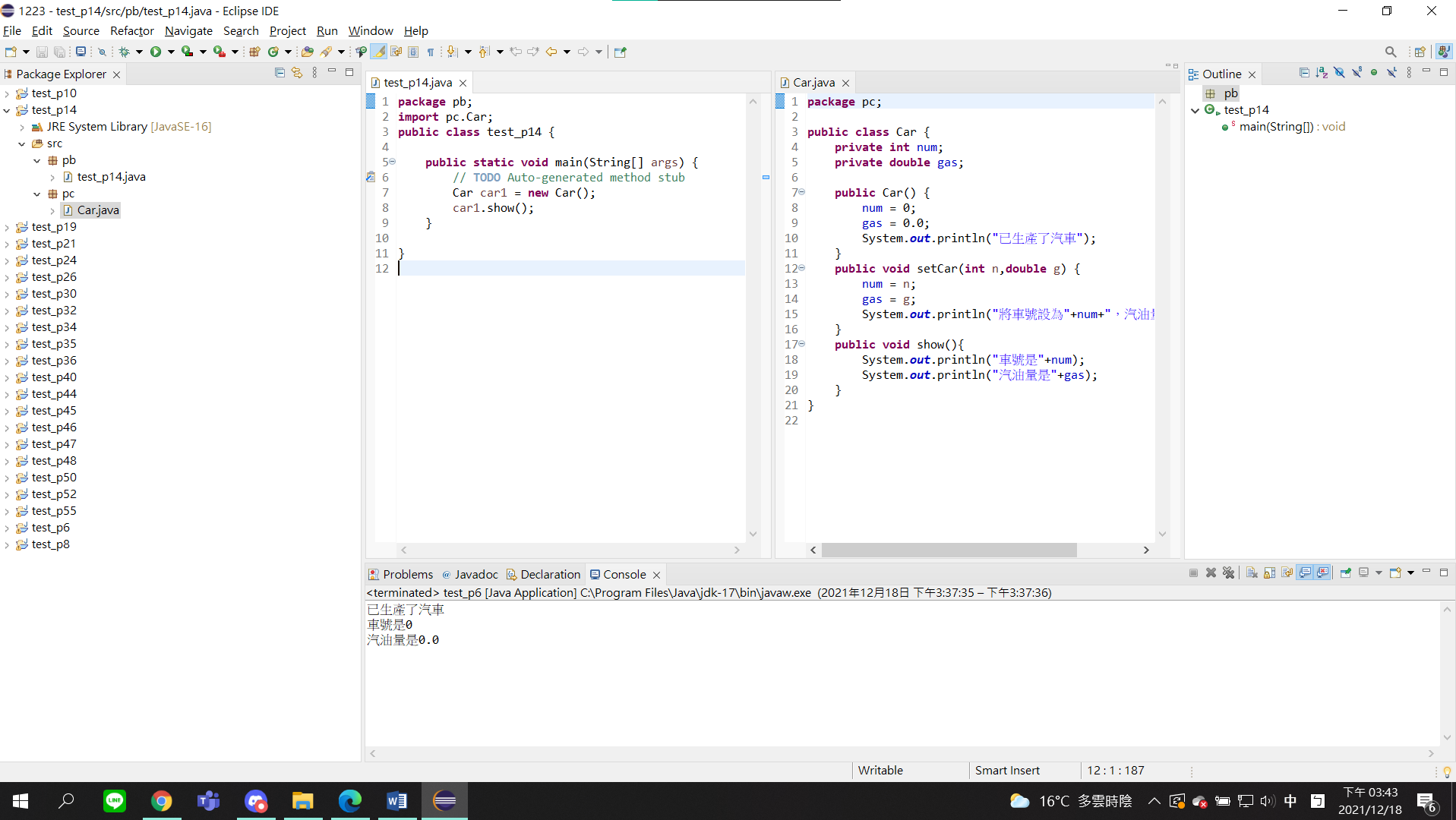
HW5-心得報告



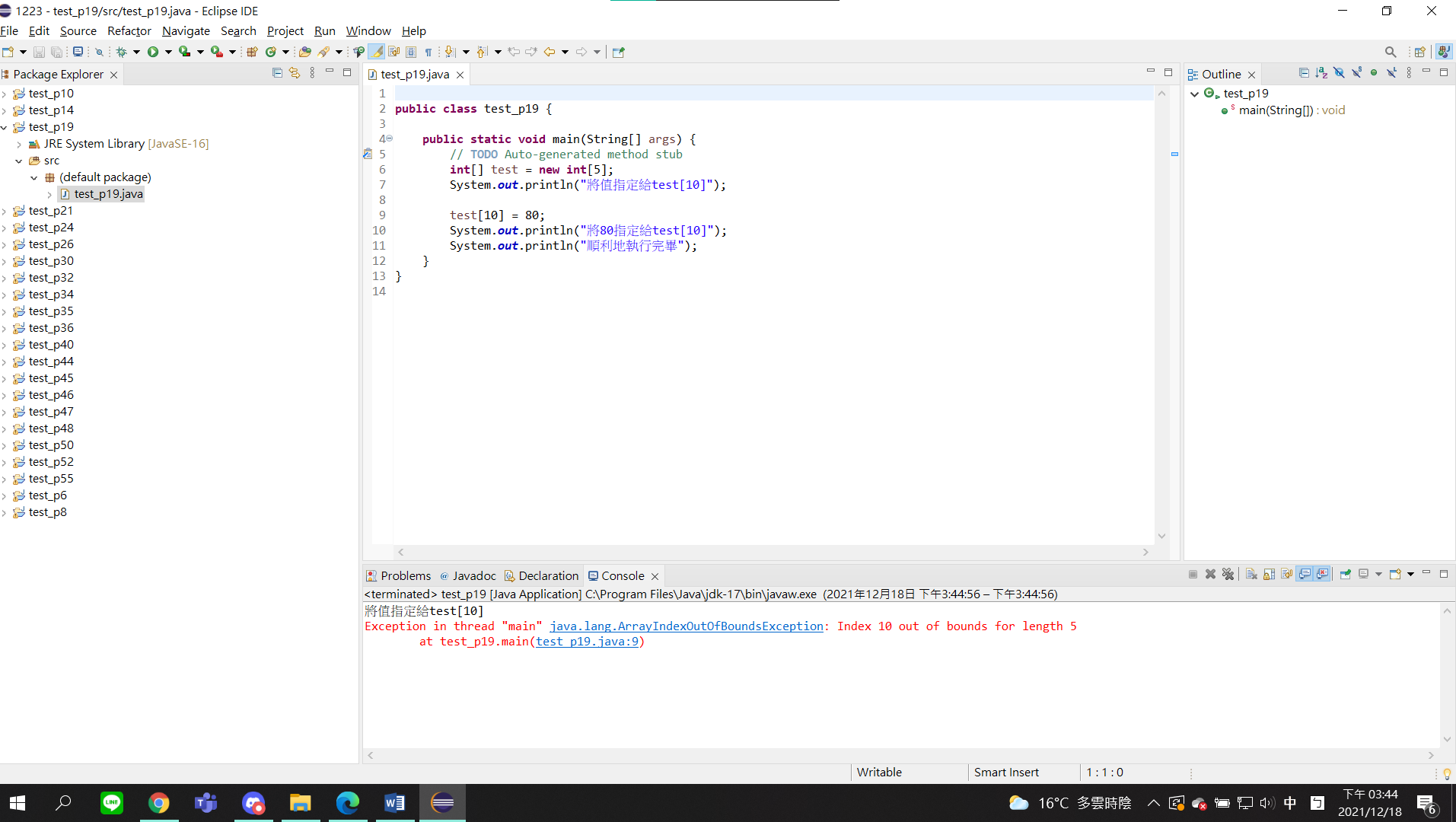
在同一個package底下，將各個類別分別寫在不同的檔案中。



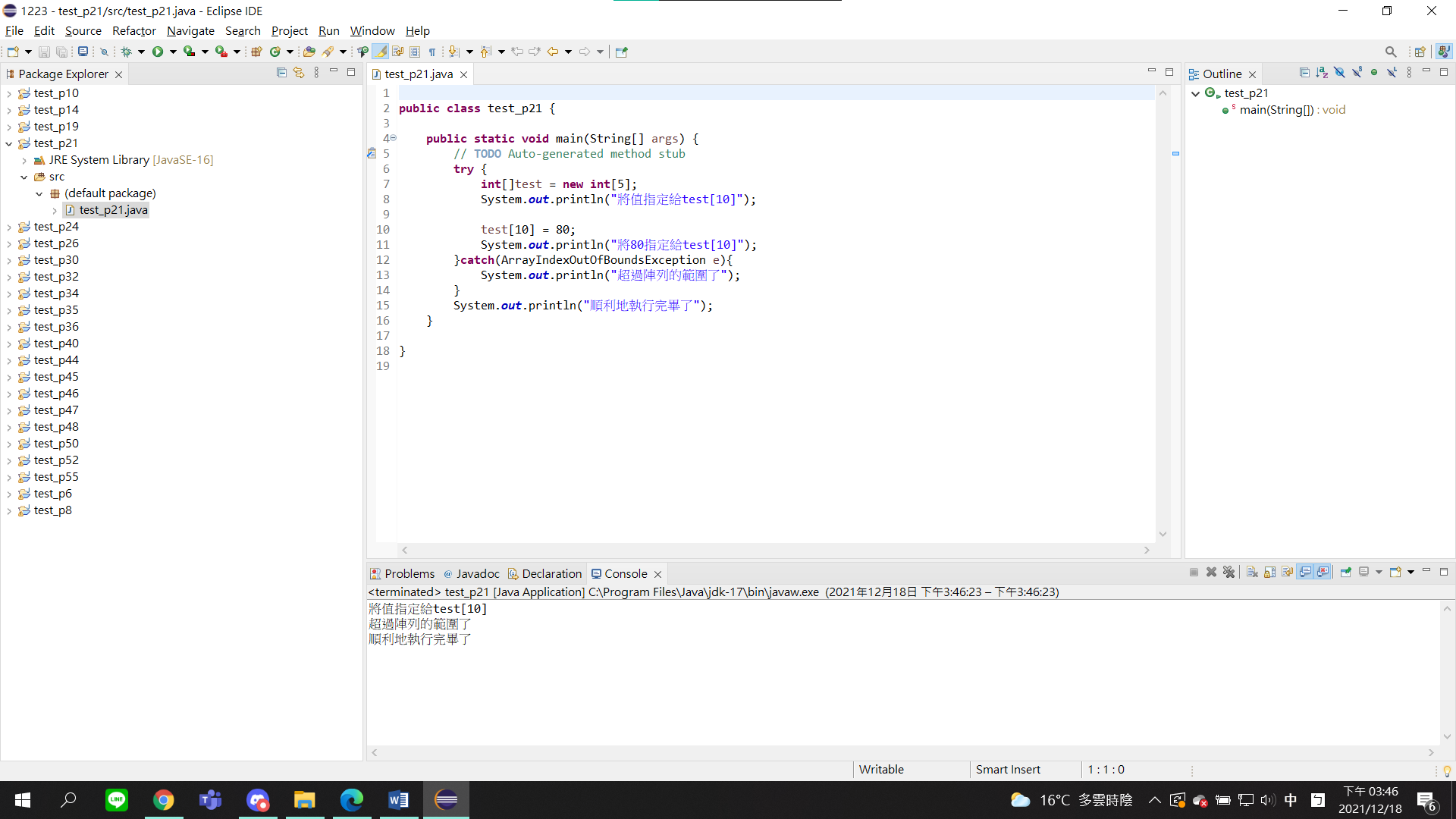
被存取的類別要加上public修飾子才能被其他套件使用，存取其他套件類別時，須加上套件名稱



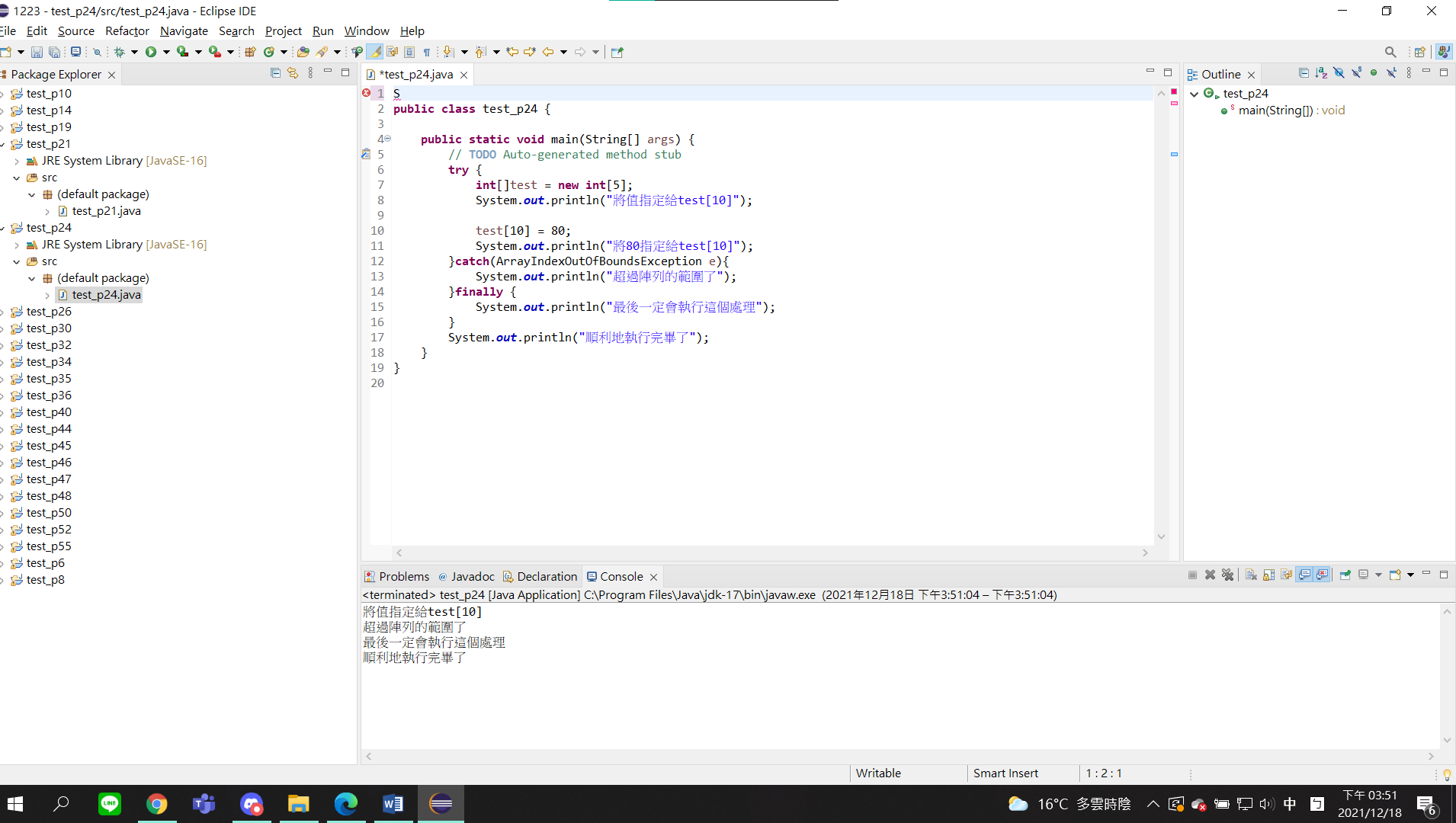
為了省去每次存取都要加上套件名稱的麻煩，則可以在程式開頭先匯入要存取的套件名稱.類別名稱



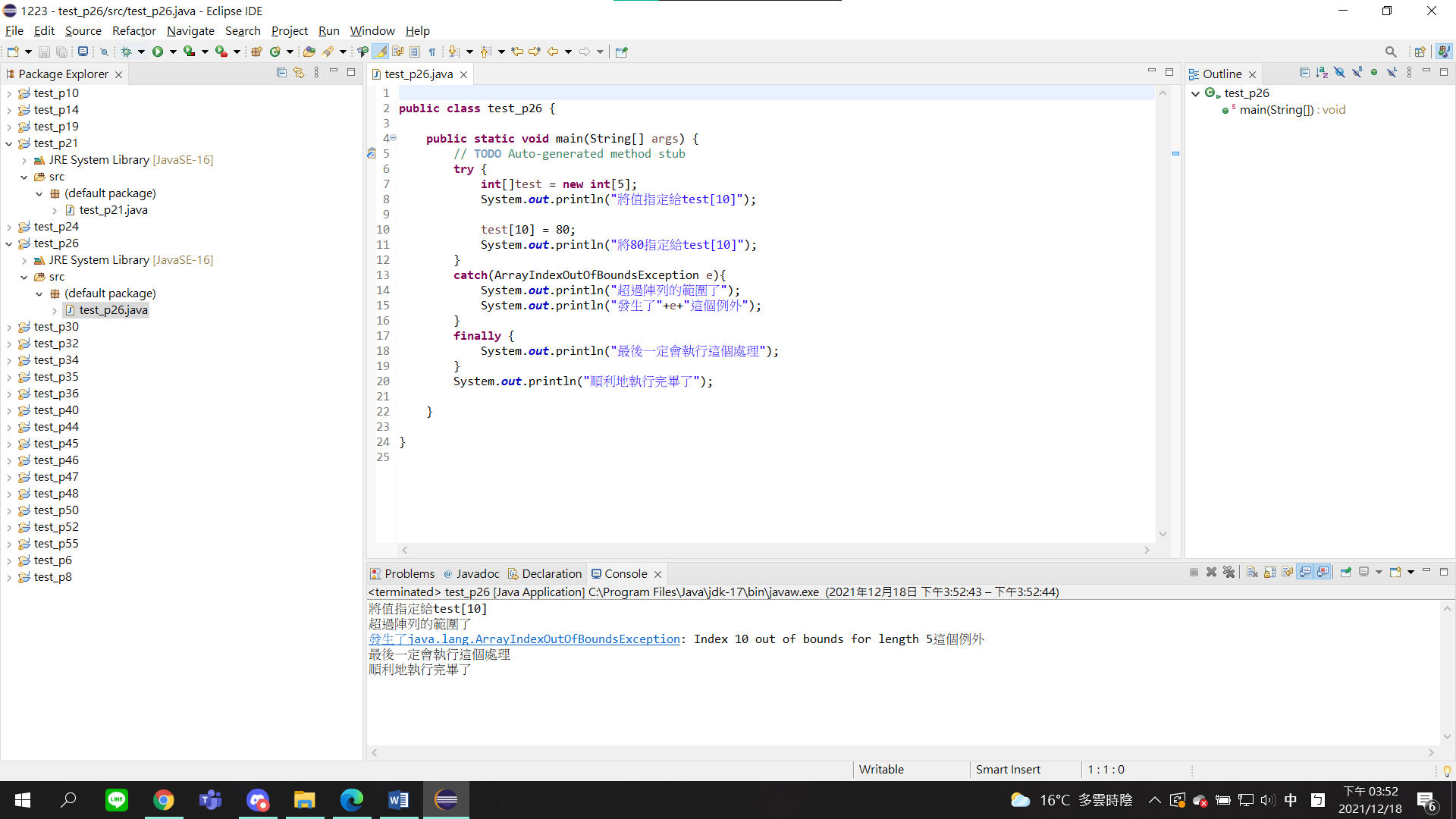
發生例外時，若不做其他處理則接下來的程式不會被執行。



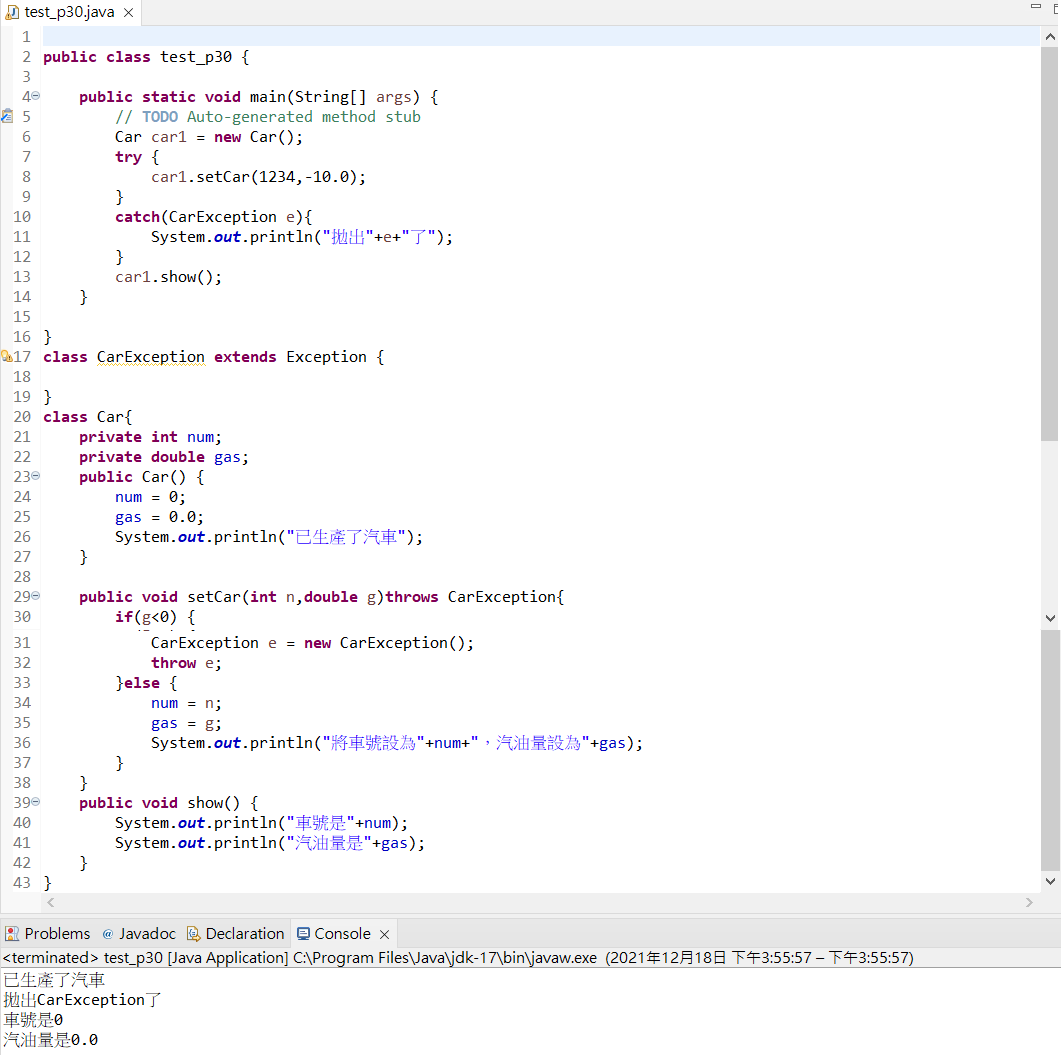
包在try當中的程式會被檢查是否發生例外，例外會在catch區塊作處理且接下來的程式會繼續執行。



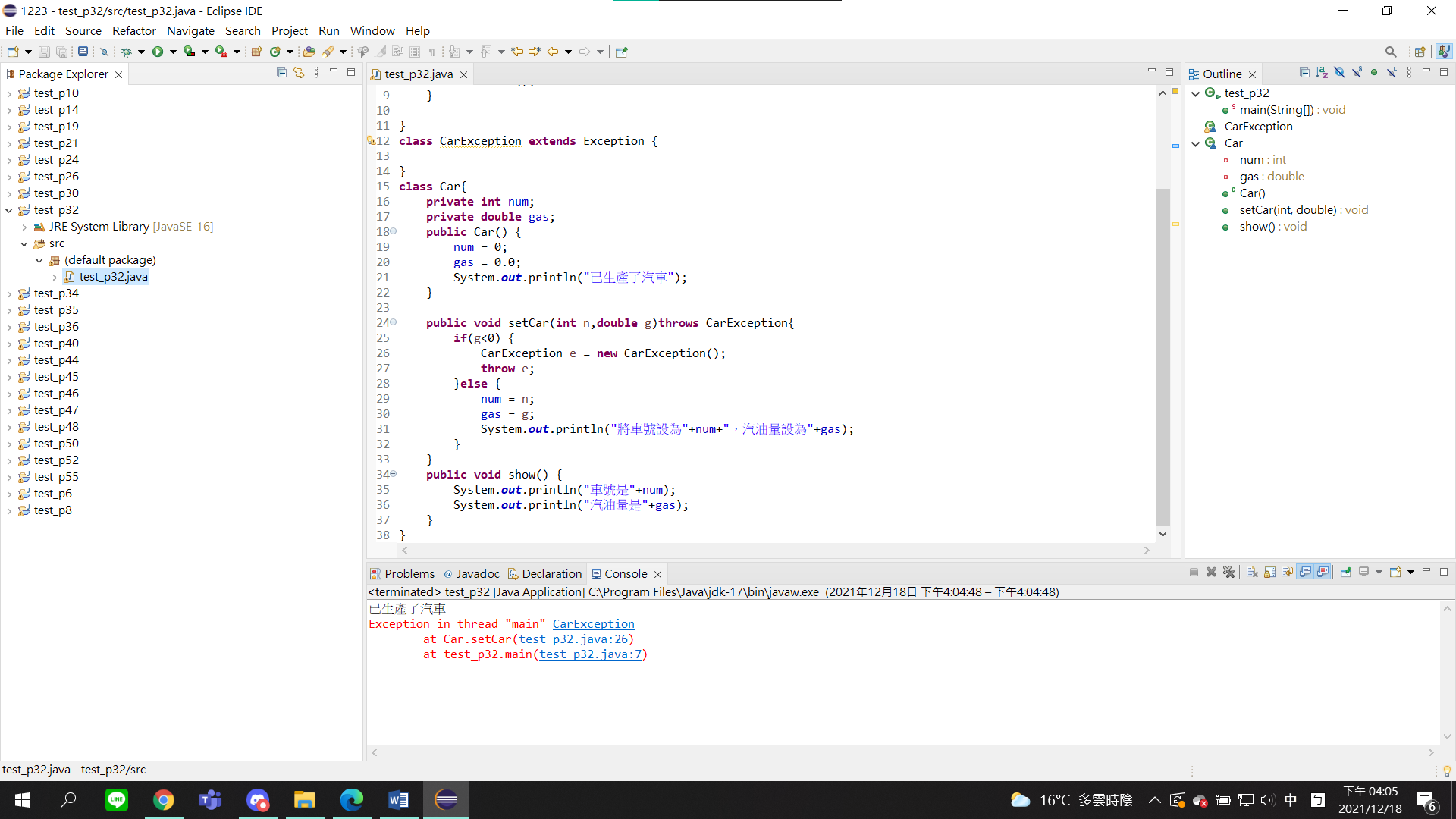
包在finally當中的程式無論有沒有發生例外都會被執行



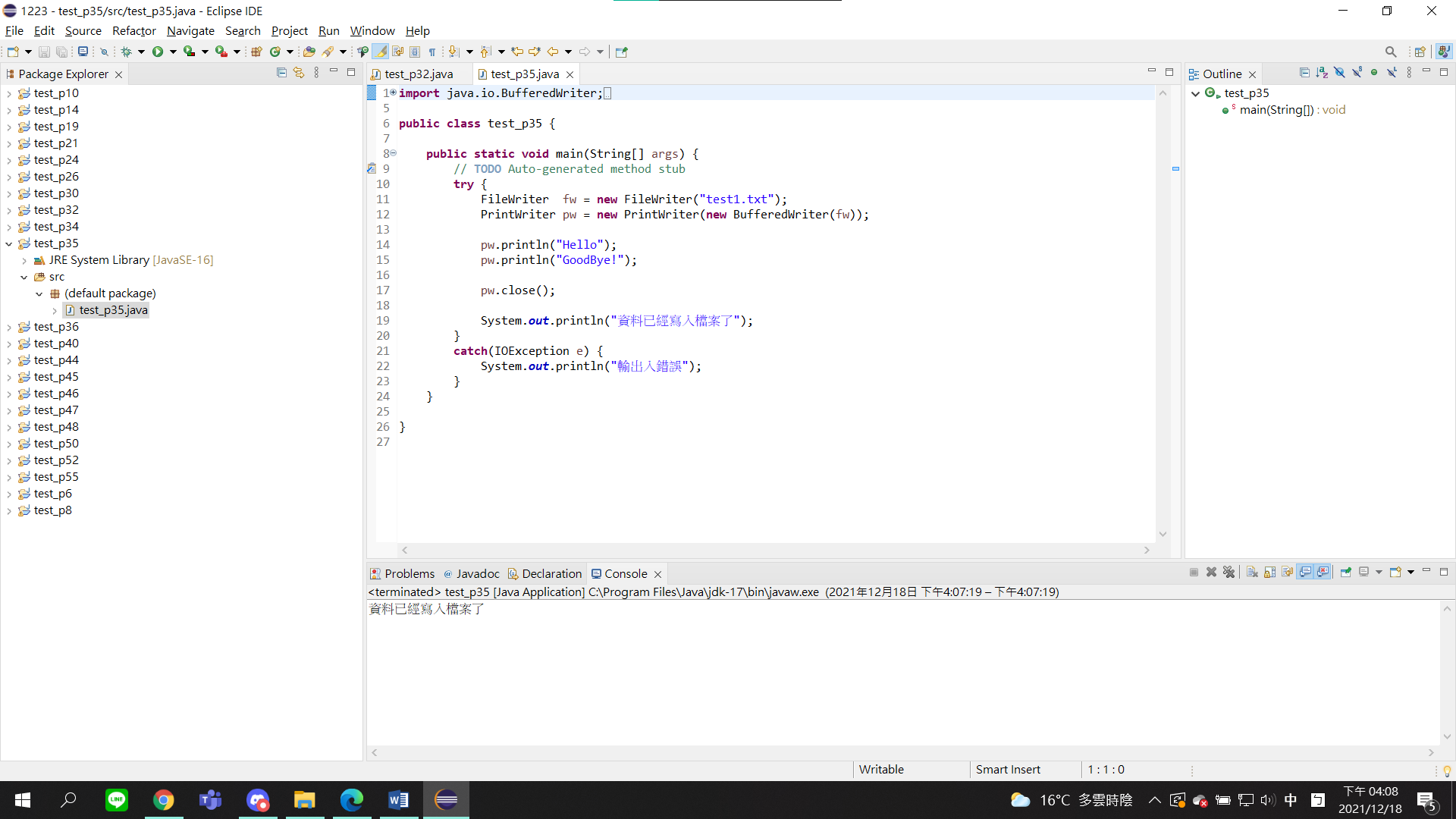
在catch的()內指定例外的類別並設定接收該物件的變數



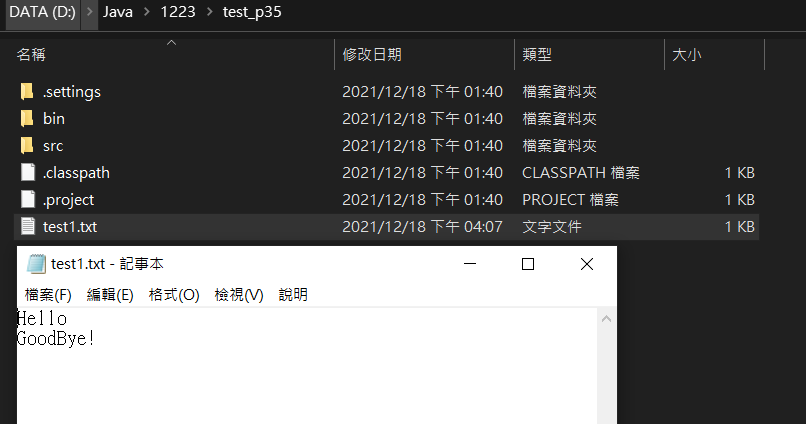
宣告可能拋出例外的方法，使用try~catch的方式呼叫該方法來拋出例外

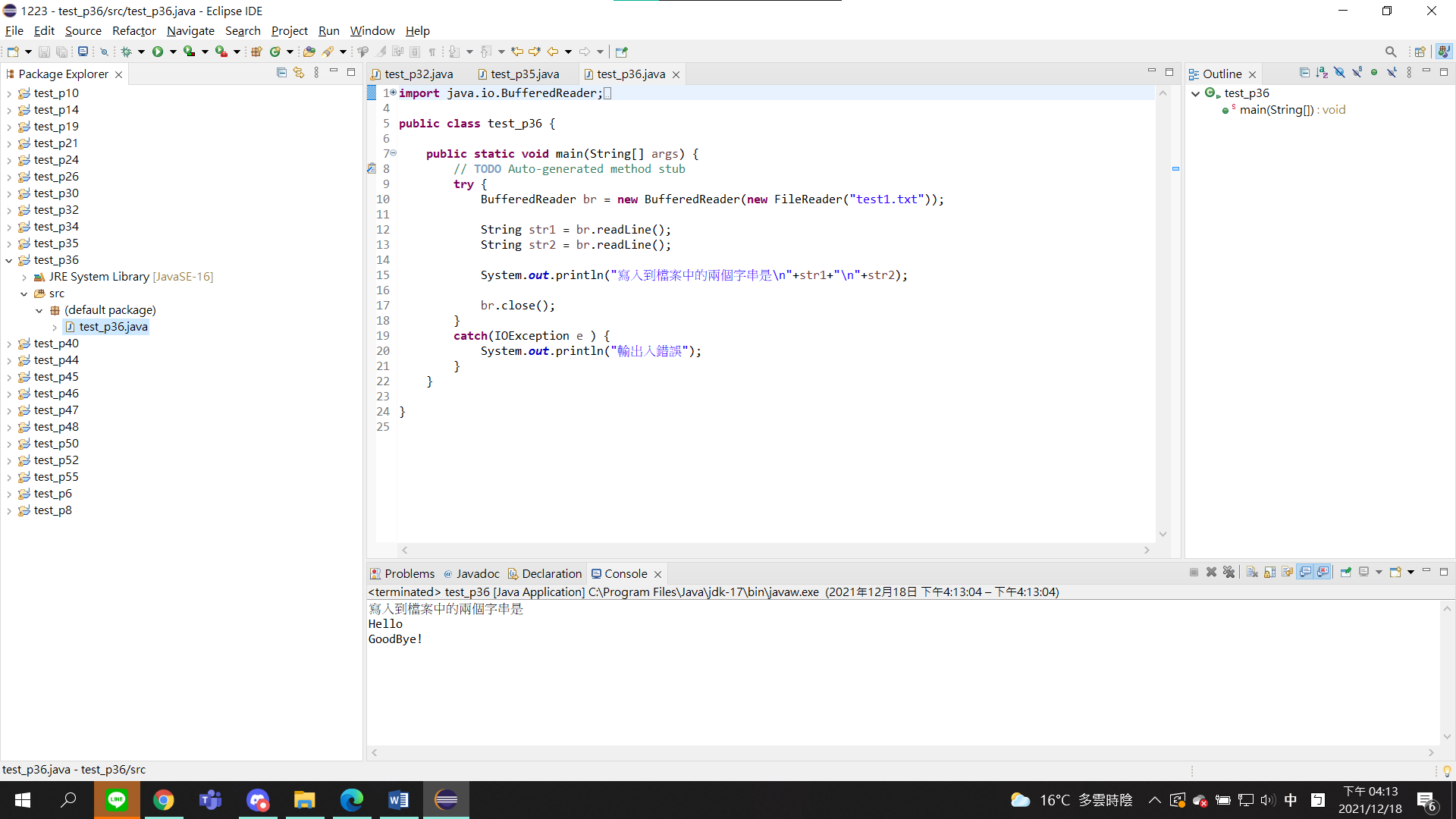


用throws的方式，把處理例外的工作交給原呼叫程式所在的方法

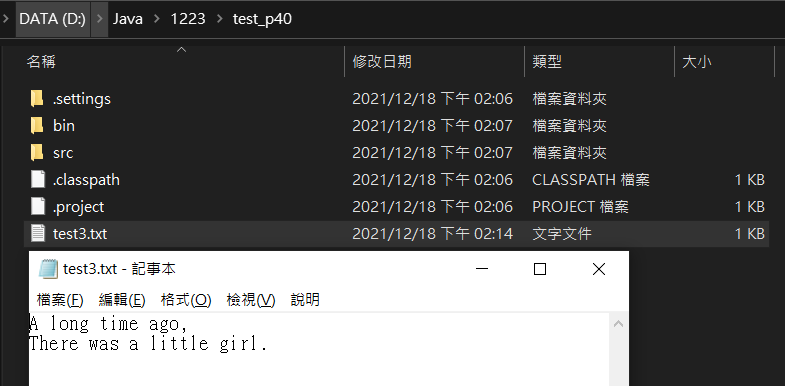


寫入資料時，要先指定檔案名稱、透過緩衝區輸出資料，最後記得關閉檔案

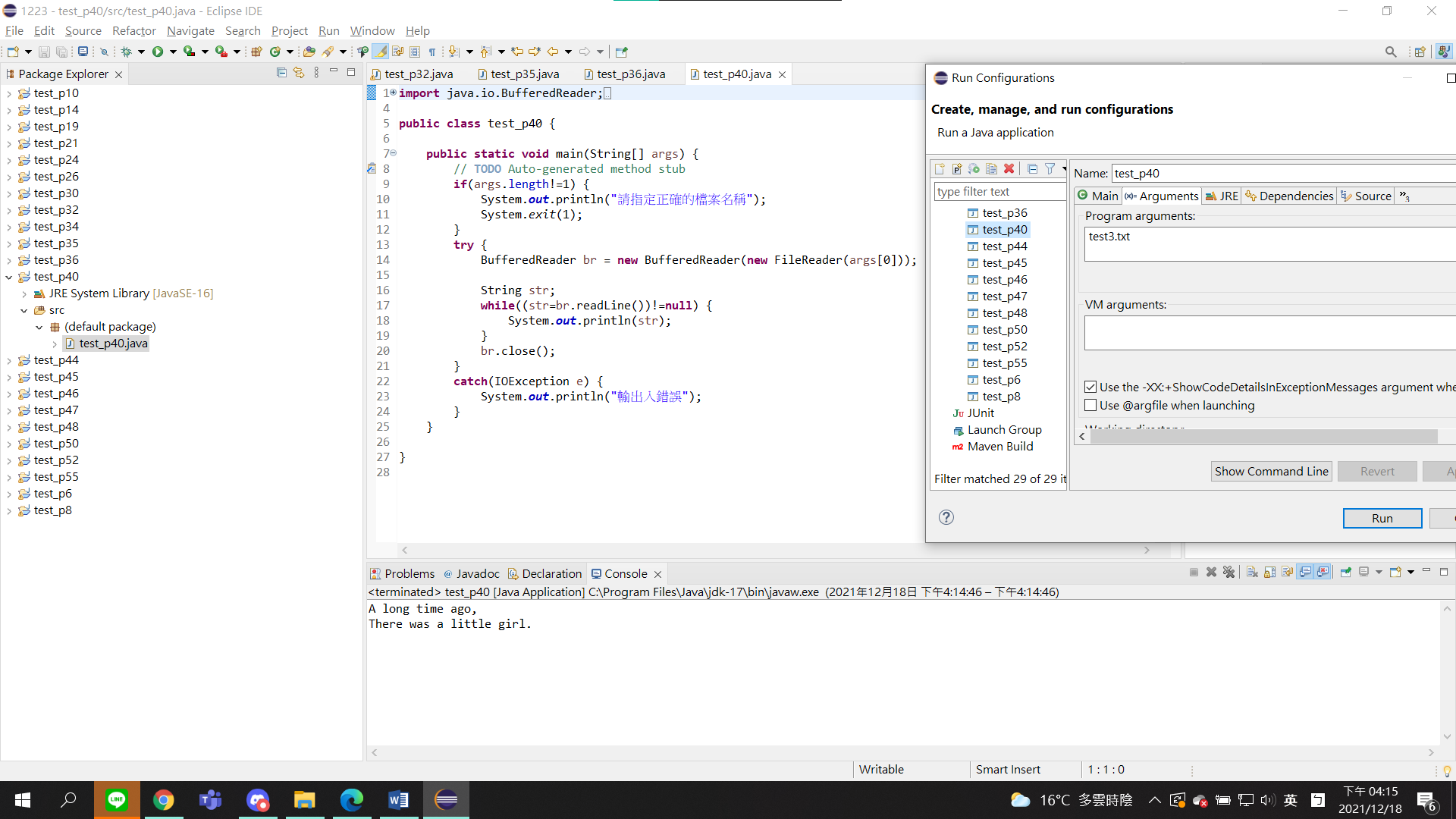




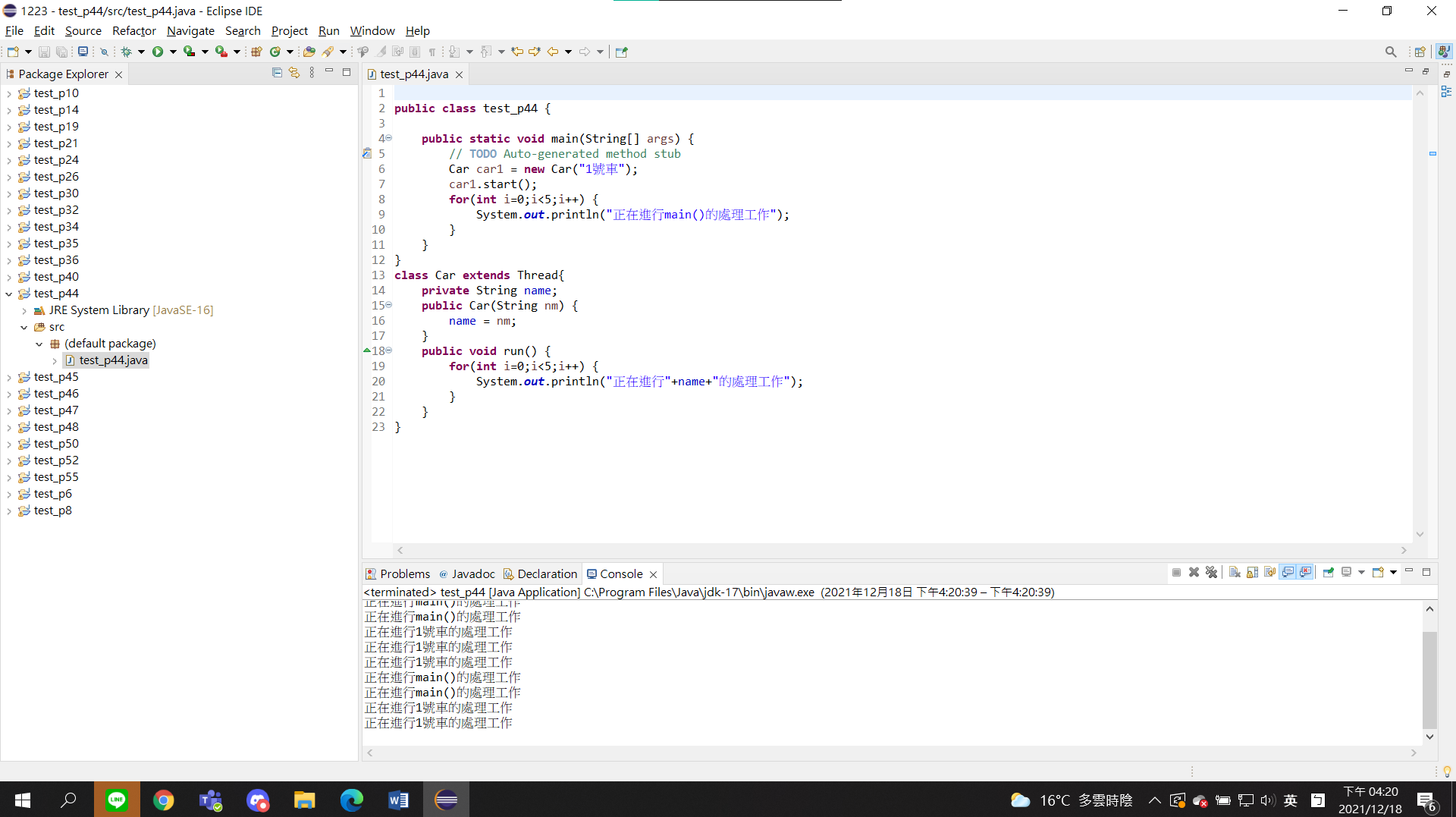
讀取資料時先指定檔案名稱並透過緩衝區讀入資料



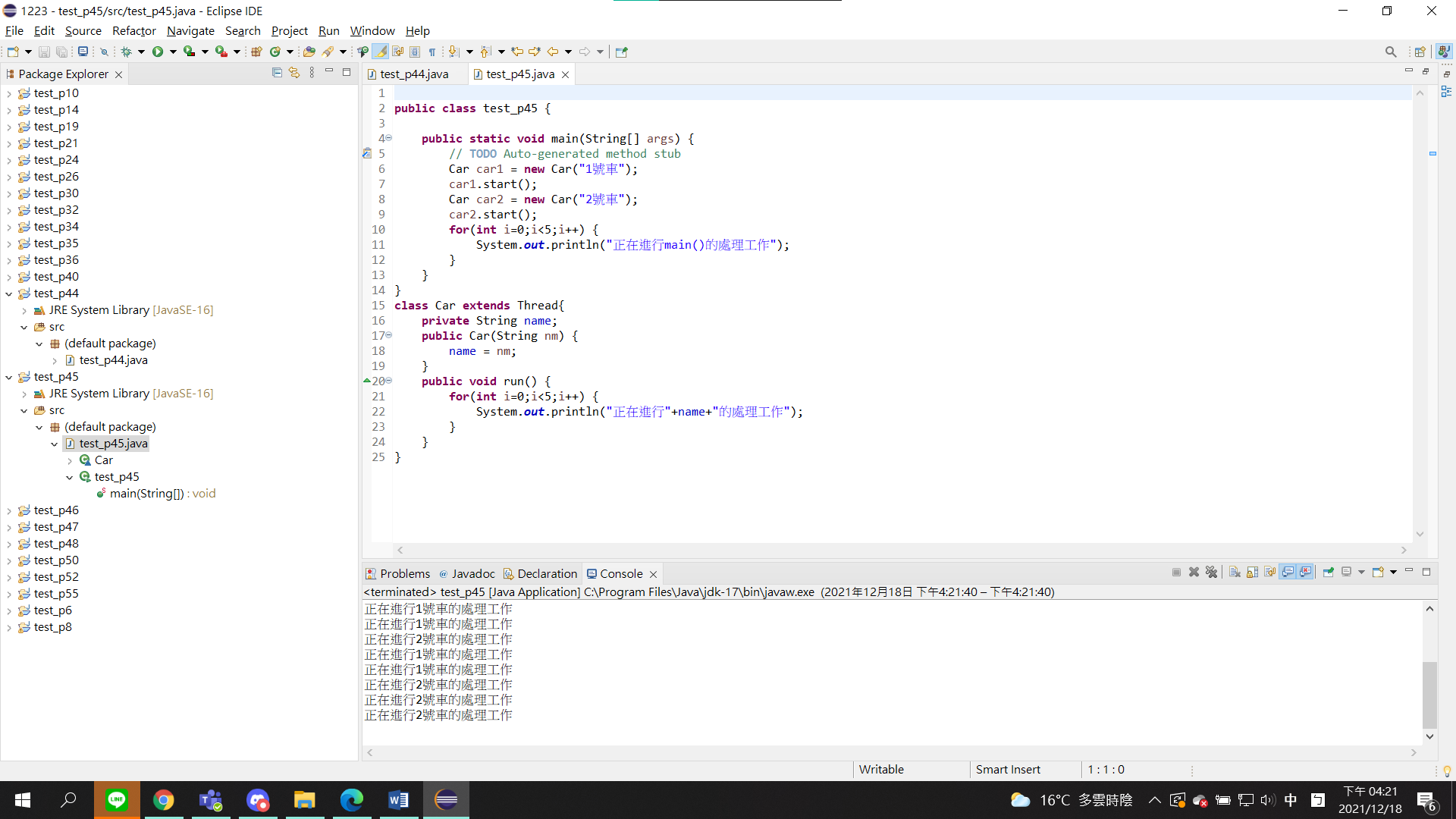
先在目錄下建立好檔案



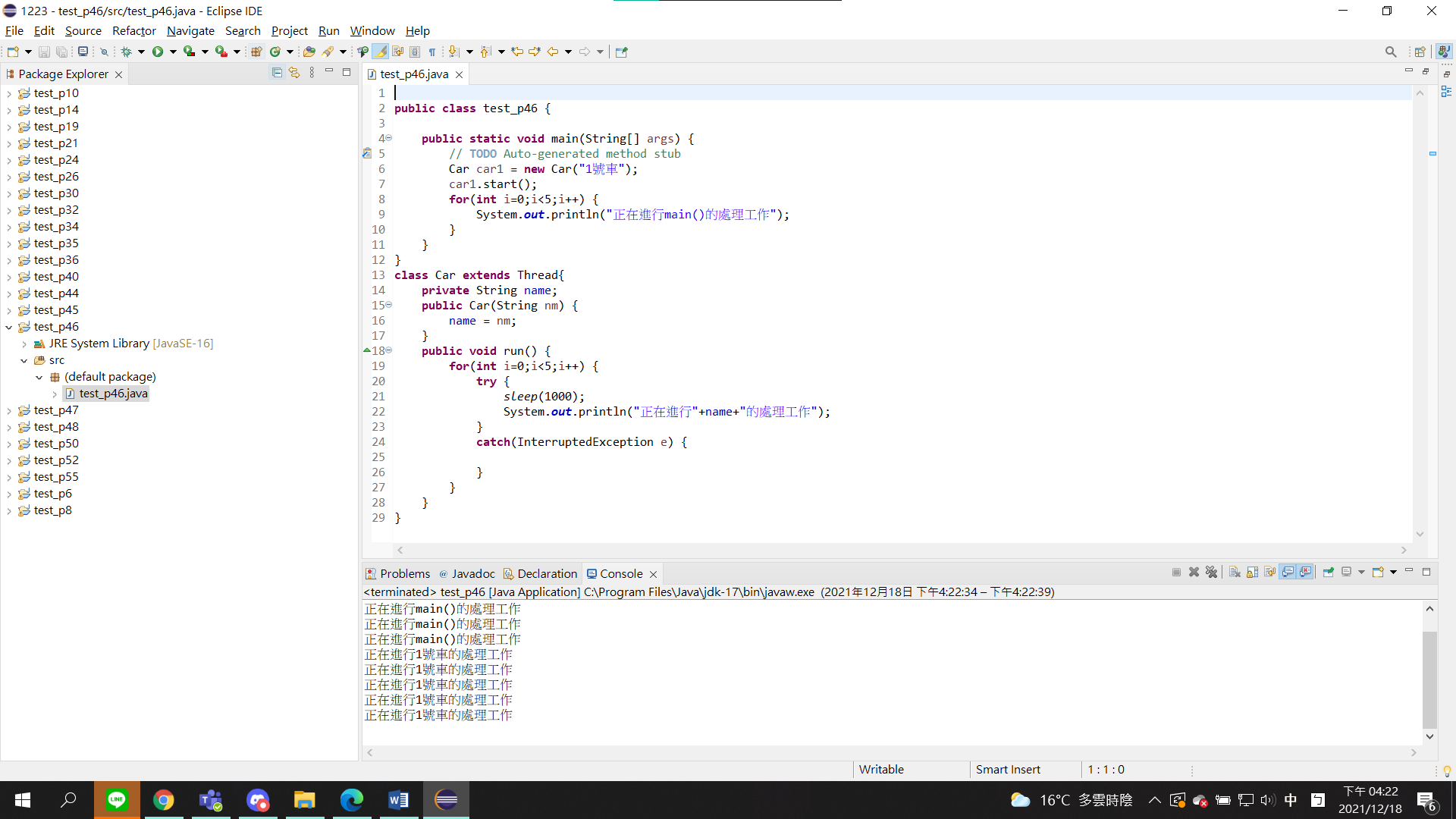
再透過Run Configurations傳入引數



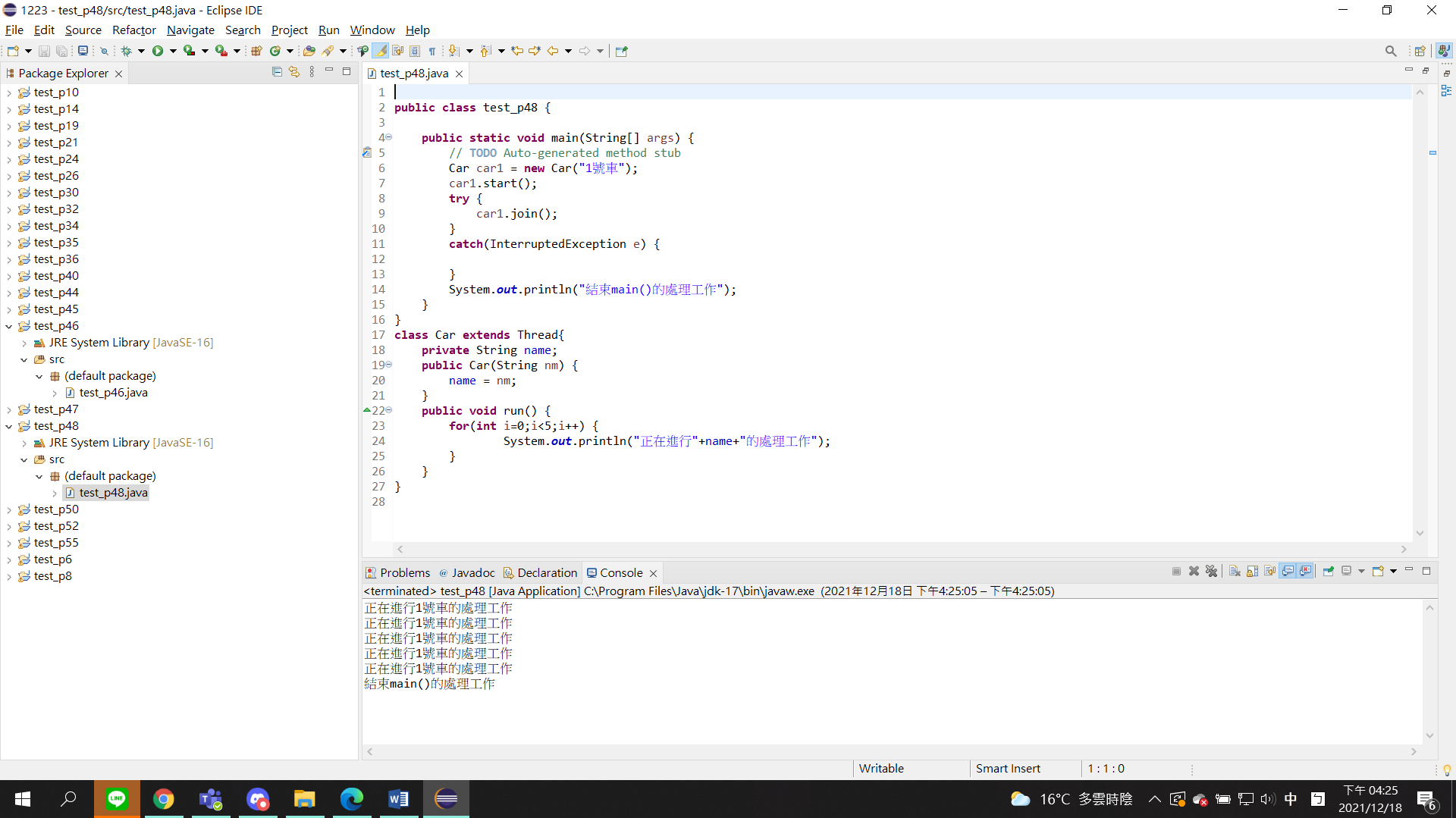
繼承Thread類別並覆寫run方法



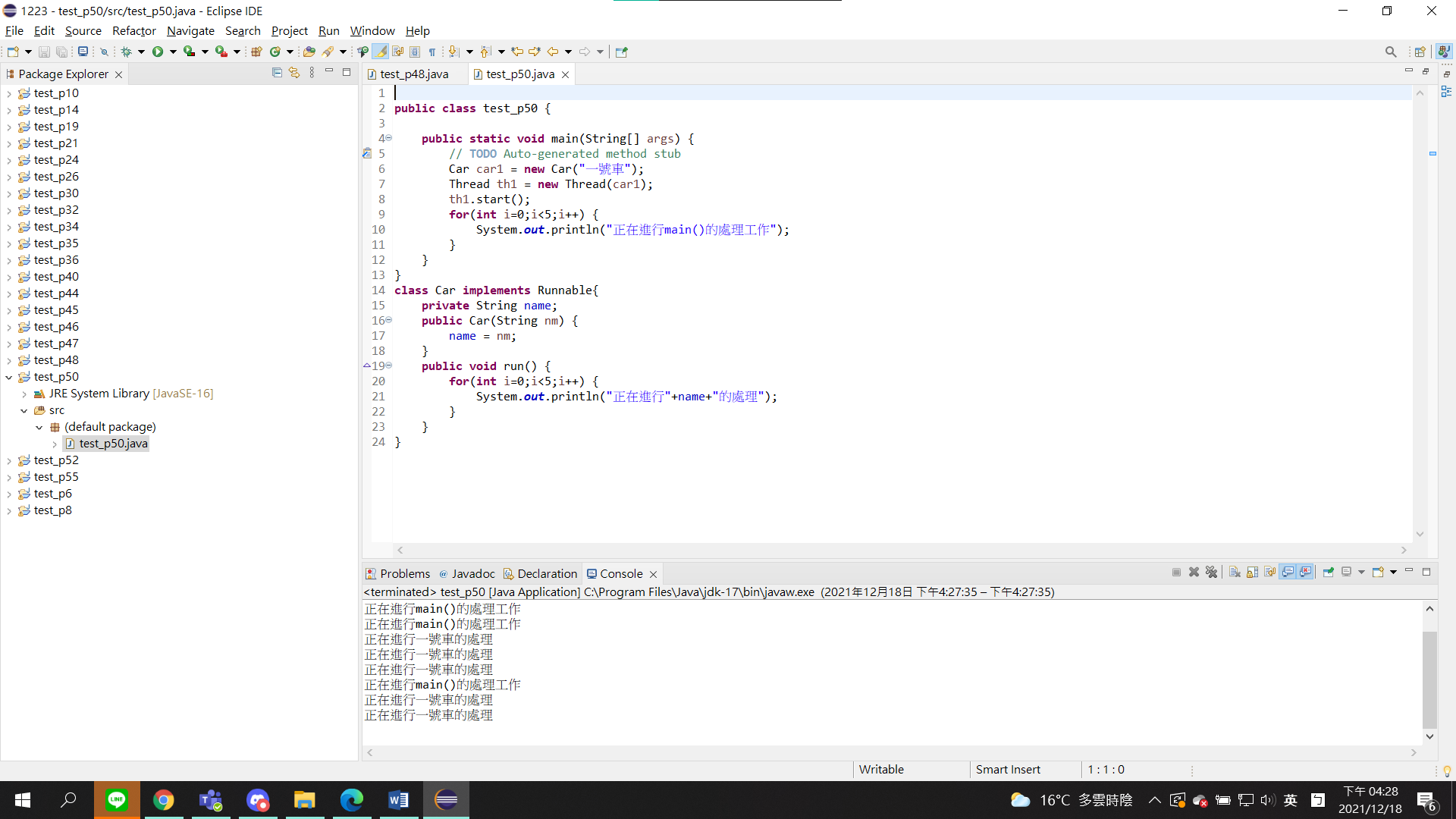
啟動多個執行緒



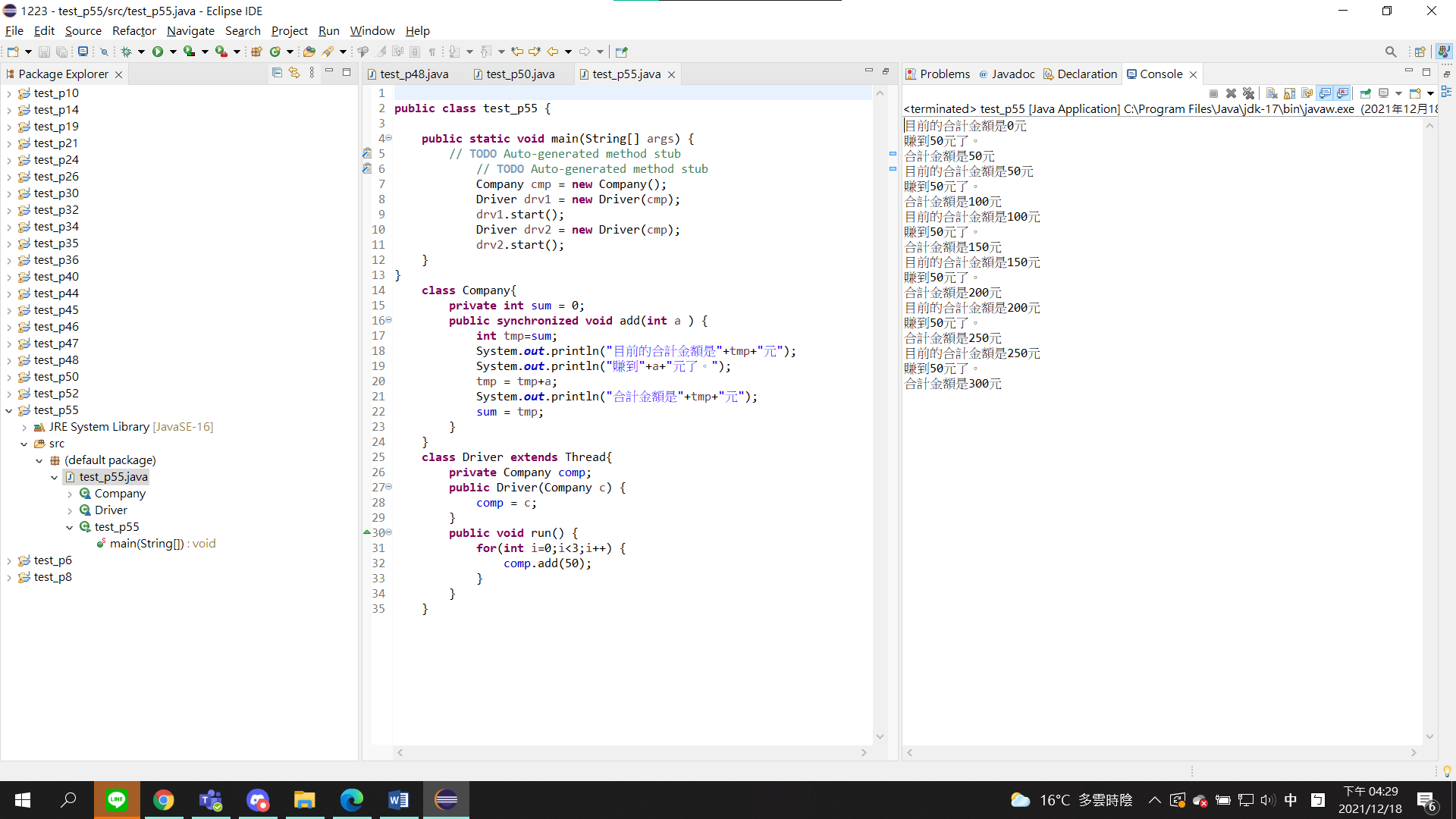
透過sleep使得每次執行這個敘述句時會先暫停1000毫秒



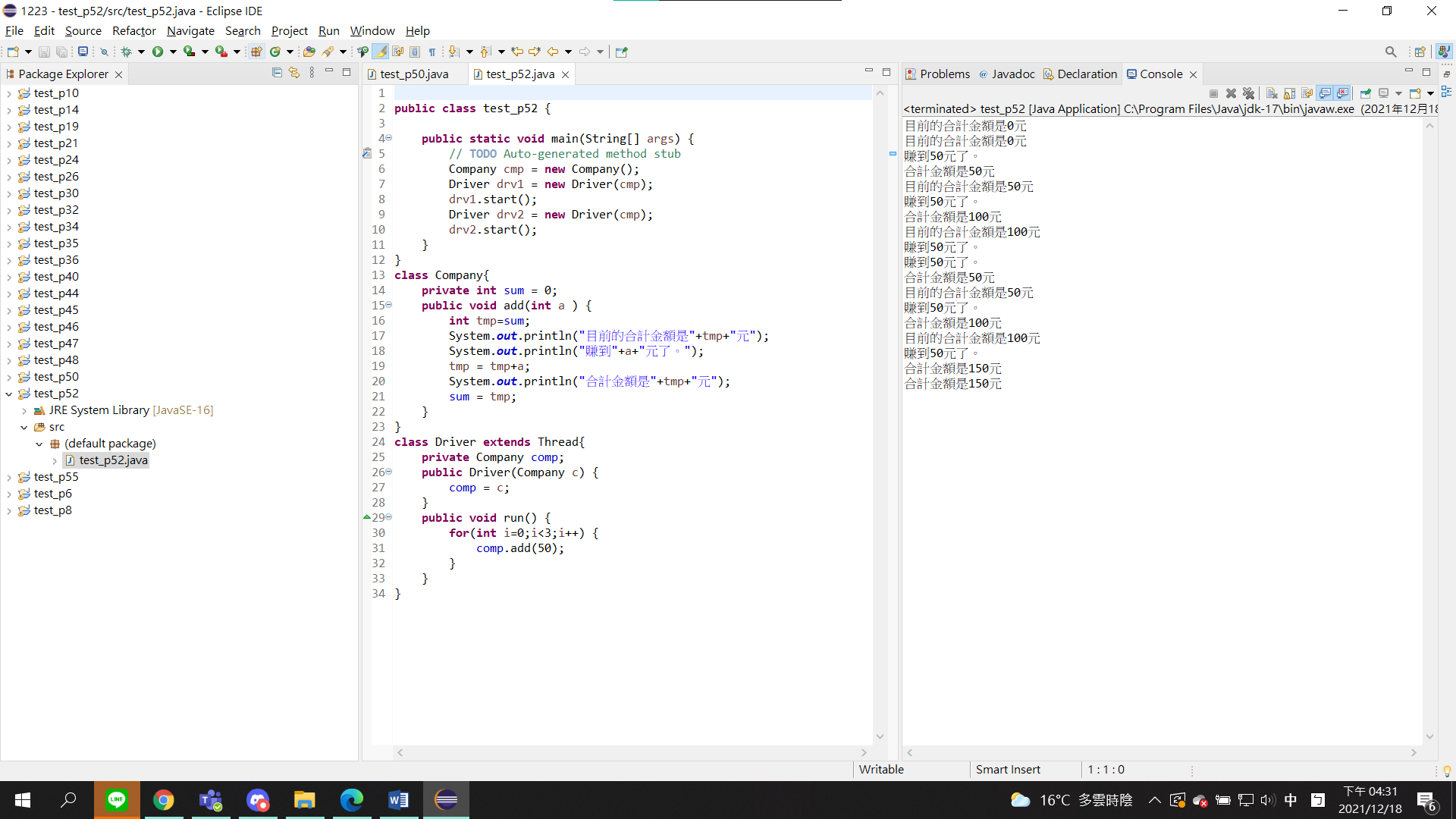
中斷現在再處理的執行緒，先處理完car1執行緒才繼續執行



要同時從不同類別繼承Thread類別時，可以透過實作Runnable介面來實現



透過加上synchronized修飾子，來避免同時呼叫導致錯誤的結果



如果沒有加上synchronized，會因為有兩個物件同時使用到add方法而導致錯誤

github網址:

https://github.com/YaXuan724/HW-CH5.git