

第九次作業

1. CrazyPrinter

分別使用 Thread 和 Runnable 實現兩個執行緒，這兩個執行緒都實現如下相同的功能：列印從 1 到 1000。（要能區分是誰列印的）。

啟動這兩個執行緒觀看程式輸出。給靠後 start 的執行緒設置優先順序為 8，再執行一次。並思考結果出現差距的原因。

注：將 2 次執行輸出的結果分別保存在 2 個檔中，第一個檔案名為 writer.txt，第二個檔案名為 writer8.txt。

2. ChaoticSum

編寫一個程式，體驗一下執行緒互斥的概念。在 main 函數中創建一新執行緒，然後 main 執行緒和這個新執行緒共用同一個變數，如 sum。在 main 函數中，編寫一個迴圈語句，總共迴圈 20 次，在每次迴圈中，先把 sum 的值加 1，然後休眠 1 毫秒。在新執行緒中，也編寫一個迴圈語句，總共迴圈 20 次，在每次迴圈中，先把 sum 賦值給一個臨時變數 t，然後把 t 加 1，然後休眠 1 毫秒，最後把 t 的值賦給 sum。當上述工作全部完成後，在 main 函數中列印 sum 的值。請運行這個程式，看看 sum 的值是多少，並思考其中的原因是什麼。

3. ParallelSum

編寫一個程式，採用多執行緒的方法計算一個陣列的各個元素之和。具體來說，該陣列為整型陣列 a，有 300 個元素。在 main 函數中，對該陣列的每個元素進行賦值，值為一個 1-100 的亂數。然後創建 3 個執行緒，第 1 個執行緒負責計算 a[0] ~ a[99] 這 100 個元素之和；第 2 個執行緒負責計算 a[100] ~ a[199] 這

100 個元素之和；第 3 個執行緒負責計算 $a[200] \sim a[299]$ 這 100 個元素之和。最後在 `main` 函數中彙集各個執行緒的計算結果，然後算出最終的結果，即 $a[0] \sim a[299]$ 這 300 個元素的總和，並且列印出來。