



## 習題

### 15.1 認識執行緒

1. 如果您在上網與好友 msn 的同時，一邊編譯 java 程式，一邊聽 mp3 音樂，則此時的作業系統是在做單一執行緒的處理，還是多執行緒？
2. 試依下面的步驟逐步完成程式的需求：

(a) 試撰寫一個 CTest 類別，繼承自 Thread 類別。請在 CTest 類別裡建立 CTest(String str) 建構元，用來設定 CTest 類別的資料成員 id 之值為 str。

(b) 請在 CTest 類別內設計 run() 函數，其內容如下：

```
public void run()
{
    for(int i=1;i<=5;i++)
    {
        for(int j=0;j<100000000;j++);
        System.out.println(id+" "+i);
    }
}
```

(c) 請在 main() 中，宣告 2 個 CTest 類別的物件 hi，id 為 "Hello"；另一個物件 bye，id 為 "Good bye"。並分別利用這 2 個物件呼叫 run()。

(d) 執行結果為何？試解釋此一現象。

3. 試修改習題 2，使得它可以同時啟動多個執行緒。
4. 習題 3 可同時啟動兩個執行緒來執行。試以下列的語法來建立物件，並利用 morning 物件呼叫 start()，使其同時啟動 3 個執行緒：

```
CTest morning=new CTest("Good morning");
```

5. 接續習題 4，請試著用下面的語法再加上一個新的物件，並利用 night 物件呼叫 start()，使 4 個執行緒同時啟動：

```
CTest night=new CTest("Good night");
```



## 15.2 實作 Runnable 介面來建立執行緒

6. 試設計 Add 類別，其資料成員與建構元如下：

```
class Add
{
    private int n;
    private int sum=0;
    public Add(int a)
    {
        n=a;
    }
}
```

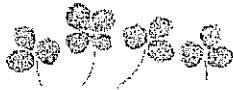
請在 Add 類別中加入可以計算  $1+2+\dots+n$  的程式，並以多執行緒的方式，分別計算  $1+2+\dots+5$  與  $1+2+\dots+10$  的值。本程式請以實作 Runnable 介面的方式建立執行緒。

7. 試修改習題 4，以實作 Runnable 介面的方式來建立執行緒。
8. 試修改習題 5，以實作 Runnable 介面的方式來建立執行緒。

## 15.3 執行緒的管理

9. 試撰寫一程式，由實作 Runnable 介面的方式建立 t1 與 t2 兩個執行緒。t1 執行緒每隔 1 秒便印出 "Thread 1 is running" 的字串，t2 執行緒每隔 2.5 秒便印出 "Thread 2 is running" 字串，直到每個執行緒執行 run() 10 次為止。
10. 試利用 sleep() 改寫習題 8，使得多執行緒小睡 0~2 秒鐘。
11. 試修改習題 3，以實作 Runnable 介面的方式建立 hi 與 bye 兩個執行緒。hi 執行緒每隔 1 秒便分別印出 "Hello 1" ~ "Hello 5" 的字串，bye 執行緒每隔 2.5 秒便分別印出 "Good bye 1" ~ "Good bye 5" 字串。
12. 試修改習題 8，請限制執行緒的流程，並在所有執行緒結束後，印出 "All Threads are finished" 字串。其執行緒的排程如下：

hi 執行緒-->morning 執行緒-->night 執行緒-->bye 執行緒



#### 15.4 同步處理

13. 試利用 Runnable 介面改寫 app15\_8，並加入 Synchronized 關鍵字，使二執行緒同步。
14. 小華去便利商店買了張面值 200 元的電話卡，他分別在 3 個不同的公共電話使用這張電話卡。請完成下列程式的需求：
  - (a) 請撰寫 PrePaid 類別，繼承自 Thread 類別，資料成員為 sum，用來記錄電話卡的餘額。
  - (b) 請在 run() 中，利用亂數產生 0~99 的整數 fee，用來當成單次使用電話卡的通話費。當電話卡的餘額 sum 大於 10 元時，即呼叫 talk(fee)，傳入此次打電話的通話費，計算並顯示電話卡的餘額。
  - (c) 請在 talk() 中加入 sleep()，故意拖延時間，使程式發生同步問題。

#### 綜合練習

15. 試依下面的步驟逐步完成兩個排序程式的執行緒：
  - (a) 試撰寫一個 Sort 類別，並在 Sort 類別裡建立兩個類別函數 (class method) bubbleSort(double data[]) 與 quickSort(double data[])，分別利用氣泡排序法 (bubble sort) 與快速排序法 (quick sort) 來排序倍精度浮點數的資料，並測試程式碼的正確性 (關於這兩個排序的演算法，請參閱資料結構或演算法等相關書籍)。
  - (b) 請用 Math.random() 產生十萬筆 0~10000 之間 double 型態的亂數，並以試著分別以 bubbleSort() 與 quickSort() 來排序之。
  - (c) 試撰寫多執行緒，讓 bubbleSort() 與 quickSort() 同時排序十萬筆 0~10000 之間 double 型態的亂數，並由執行的結果來說明哪一種排序法較好。
16. 試撰寫兩個執行緒，其中一個執行緒可用來計算 2~100000 之間質數 (prime number) 的個數，另一個執行緒則可用來計算 100000~200000 之間質數的個數，並回答下列問題：
  - (a) 哪一個執行緒先跑完？
  - (b) 2~100000 之間質數多還是 100000~200000 之間的質數多？