

### 8-1-3 範例研討：製作時間運算工具

#### (A) 程式功能：Ex8\_1.java

許多應用系統需要相關時間計算工具，譬如：兩個時間點之間相差多久 (timeInterval())、一個時間點再過後某些時間之後是何時 (timeAfter())、以及一個時間點之前的某些時間數，又應該是何時 (timeBefore())。時間運算是屬於 60 進位方式，不同於一般 10 進位計算。

#### (B) 製作技巧研討：

製作此系統可區分為下列 3 個主題：

- 時間物件格式 (Time.class)：需製作一個時間類別，規劃時間物件的規格；利用此類別所產生的物件變數之間，才可以做運算處理 (於 timeTool.java)。
- 時間運算工具類別 (timeTool.class)：此類別內包含所需的時間計算器 (或稱方法)，再利用它產生 (new) 物件方法，就可以做運算處理事項。本範例包含下列四個工具 (於 timeTool.java)：
  - getTime()：輸入時間字串 (時/分/秒)，轉換成時間物件格式 (時、分、秒) 並回傳。
  - timeAfter()：輸入現在時間點與之前時間量，回傳之前的相當時間點 (時、分、秒)。
  - timeBefore()：輸入現在時間點與之後時間量，回傳之後的相當時間 (時、分、秒)。
  - timeInterval()：輸入兩個時間點 (時、分、秒)，回傳之間的距離時間量。
- 編譯主程式 (Ex8\_1.class)：測試所製作的工具是否可用。

#### (C) 程式範例：timeTool.java

吾人利用 timeTool.java 檔案製作 Time 與 timeTool 兩只類別；兩類別的功能如圖 8-2 所示。

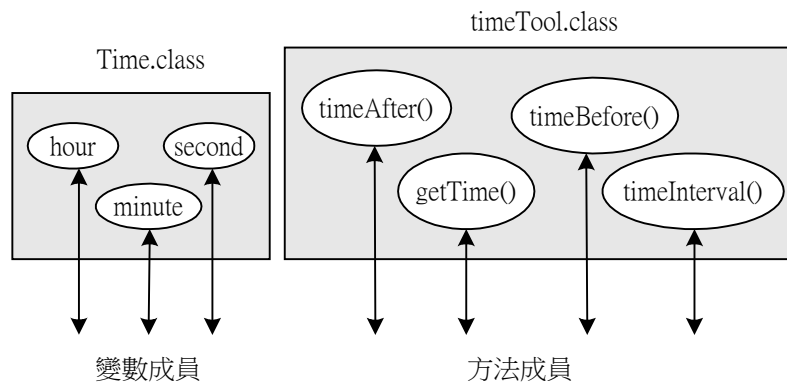


圖 8-2 Time 與 timeTool 類別的功能

```

01 //timeTool.java
02
03 import java.util.*;
04
05 /* 時間類別，描述時間物件的規格 */
06 class Time {
07     int hour;
08     int minute;
09     int second;
10 }
11
12 /* 時間工具類別，包含 4 個方法 (計算器) */
13 class timeTool {
14
15     /* 轉換字串成為時間物件格式 */
16     Time getTime(String time_S){
17         Time time = new Time();
18         Scanner s = new Scanner(time_S).useDelimiter("/");
19         time.hour = s.nextInt();
20         time.minute = s.nextInt();
21         time.second = s.nextInt();
22         return time;
23     }
24
25     /* 之前時間計算器 */
26     Time timeBefore(Time now, Time value) {
27         Time time = new Time();
28         time.hour=0; time.minute=0; time.second=0;
29         time.second = now.second - value.second;
30         if (time.second < 0) {
31             time.second = time.second + 60;
32             time.minute = 1;
33         }
34         time.minute = now.minute - value.minute - time.minute;
35         if (time.minute < 0) {
36             time.minute = time.minute + 60;
37             time.hour = 1;
38         }
39         time.hour = now.hour - value.hour - time.hour;
40         if(time.hour < 0)
    
```

```
41         time.hour = time.hour + 24;
42         return time;
43     }
44
45     /* 之後時間計算器 */
46     Time timeAfter(Time now, Time value) {
47         Time time = new Time();
48         time.hour=0; time.minute=0; time.second=0;
49         time.second = now.second + value.second;
50         if (time.second > 60) {
51             time.second = 60 - time.second;
52             time.minute = 1;
53         }
54         time.minute = time.minute + now.minute + value.minute;
55         if (time.minute > 60) {
56             time.minute = 60 - time.minute;
57             time.hour = 1;
58         }
59         time.hour = time.hour + now.hour + value.hour;
60         if (time.hour > 24)
61             time.hour = time.hour - 24;
62         return time;
63     }
64
65     /* 兩點時間之間距離計算器 */
66     Time timeInterval(Time start, Time end) {
67         Time time = new Time();
68         time.hour=0; time.minute=0; time.second=0;
69         time.second = end.second - start.second;
70         if (time.second < 0) {
71             time.second = time.second + 60;
72             time.minute = 1;
73         }
74         time.minute = end.minute - start.minute - time.minute;
75         if (time.minute < 0) {
76             time.minute = time.minute + 60;
77             time.hour = 1;
78         }
79         time.hour = end.hour - start.hour - time.hour;
80         if(time.hour < 0)
81             time.hour = time.hour + 24;
82         return time;
83     }
84 }
85
86
```

#### (D) 程式範例：Ex8\_1.java

完成 Time 與 timeTool 兩類別之後，吾人利用 Ex8\_1 範例驗證這兩類別所產生的物件，觀察其功能是否滿足所需。圖 8-3 為兩類別產生物件的情況，利用 Time 類別產生了 now 物件，該物件內具有 now.hour、now.minute 與 now.second 等 3 個變數成員，則利用這些成員記錄時間的時、分與秒。同樣的，timeTool 類別產生 tool 物件後，該物

件具有 `tool.getTime()...` 等 4 個方法成員，也利用這 4 個方法庫計算時間長短。

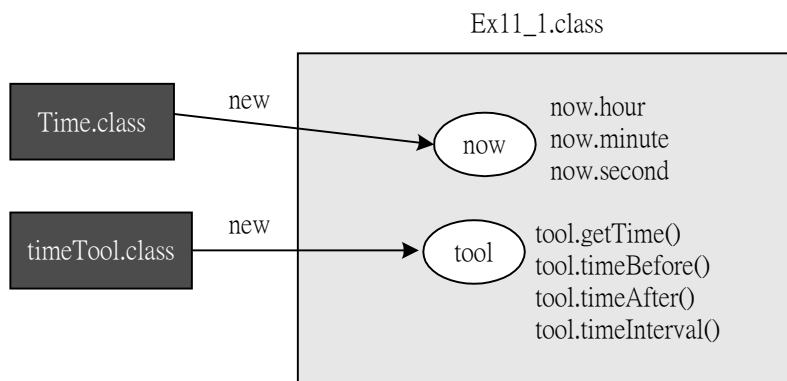


圖 8- 3 Time 與 timeTool 類別產生物件

```

01 //Ex8_1.java
02
03 /* timeTool.class, Time.class 需儲存於相同目錄下 */
04 import java.io.*;
05 import java.util.*;
06 public class Ex8_1 {
07     public static void main(String args[]) throws IOException {
08         BufferedReader keyin = new BufferedReader(new
09             InputStreamReader(System.in));
10
11         timeTool tool = new timeTool(); // 產生工具，包含 4 種方法
12         Time now = new Time(); // 儲存目前時間
13         Time time_T = new Time(); // 時間暫存變數
14         Time time_B = new Time(); // 時間暫存變數
15         String time_S; // 讀入時間字串
16
17
18         /* 設定目前時間 */
19         System.out.printf("設定目前時間(時/分/秒) =>");
20         time_S = keyin.readLine();
21         now = tool.getTime(time_S);
22
23         /* 測試計算之前時間 */
24         System.out.printf("\n 目前是 %d 時 %d 分 %d 秒\n",
25             now.hour, now.minute, now.second);
26         System.out.printf("請輸入之前時間距離 (時/分/秒) =>");
27         time_S = keyin.readLine();
28         time_T = tool.getTime(time_S);
29         time_B = tool.timeBefore(now, time_T);
30         System.out.printf("之前是 %d 時 %d 分 %d 秒\n",
31             time_B.hour, time_B.minute, time_B.second);
32
33         /* 測試計算之後時間 */
34         System.out.printf("\n 目前是 %d 時 %d 分 %d 秒\n",
35             now.hour, now.minute, now.second);
36         System.out.printf("請輸入之後時間距離 (時/分/秒) =>");
37         time_S = keyin.readLine();
38
    
```

```
39     time_T = tool.getTime(time_S);
40     time_B = tool.timeAfter(now, time_T);
41     System.out.printf("之後是 %d 時 %d 分 %d 秒\n",
42                       time_B.hour, time_B.minute, time_B.second);
43
44     /* 測試計算兩時間之間 距離 */
45     System.out.printf("\n目前是 %d 時 %d 分 %d 秒\n",
46                       now.hour, now.minute, now.second);
47     System.out.printf("請輸入第二點時間 (時/分/秒) =>");
48     time_S = keyin.readLine();
49     time_T = tool.getTime(time_S);
50     time_B = tool.timeInterval(now, time_T);
51     System.out.printf("兩點時間相差 %d 時 %d 分 %d 秒\n",
52                       time_B.hour, time_B.minute, time_B.second);
53
54     }
55 }
```