

第八章 方法與方法庫產生

8-1 物件方法

8-1-1 物件方法與類別方法

到目前為止，我們對物件的概念大多僅限於描述真實環境的應用，它使所描述的『事件』資料，具有主動處理能力。基本上，物件包含變數與方法兩種成員，『變數成員』描述真實環境的屬性，『方法成員』為存取變數成員的途徑。如果擴充方法成員，使用具有某一通用性的專屬功能(增強功能)，也使變數成員成為區域性變數(削弱功能)，又展開了物件的另一種運用。吾人可針對某些特殊運用發展出工具套件，並且可被引用產生另一個專屬工具，該物件工具則成為『物件方法庫』(Object Library)，如圖 8-1 所示；本章就針對這方面的運用來加以說明。

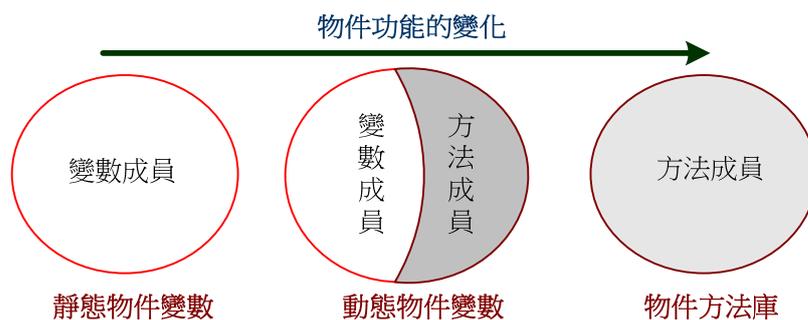


圖 8-1 方法庫的演變

利用物件導向技巧所發展出來的方法庫 (Method Library) 可區分為下列兩大類：

- (1) **物件方法**：利用類別描述技巧，包裝多個方法成員於某一類別內，而此類別必須經由 new 產生物件後，才可引用物件內的方法，該物件內的方法則稱為『物件方法』(Object method)。
- (2) **類別方法**：利用類別建構技巧，包裝多個方法於某一類別內。此類別可不用經由 new 產生物件，即可引用類別內方法，該類別內方法則稱之為『類別方法』(Class method)。

其實，類別方法就像靜態變數一樣，都是屬於靜態的(第 10 章說明)。宣告語法也類似，只要將 static 關鍵字加諸於方法名稱的前面即可。以下分別介紹這兩方法的產生與運用。

8-1-2 物件方法宣告與產生

將一些常用的程式整合於某一類別內，類別內每一方法成員，實現某一專屬功能的程式，讓其他類別 (或物件) 可以直接呼叫使用。這種做法很像傳統語言的『庫存函數』(Library)，但 Java 物件方法必須經過 new 命令產生另一個物件實體，再引用物件內的方法成員。宣告語法如下：

	宣告語法	範 例
類別內物件 方法宣告	<pre>class class_name { return_type method_name() { method_body; } }</pre>	<pre>class mathFun { int getMax(intx, int y) { 物件方法內容..... } }</pre>
物件產生	<pre>class_name object_1 = new class_name();</pre>	<pre>mathFun u1_Fun = new mathFun();</pre>
物件方法引 用	<pre>object1.method_name();</pre>	<pre>u1_Fun.getMath();</pre>

如同一般類別的宣告語法，可以將類別宣告成私有性 (private class) 或公有性 (public class)，也可將方法成員宣告成私有性 (private static) 或公有性 (public static)。

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。請製作上述 3 種時間工具，並測試其運算結果，期望操作介面如下：

```
G:\Examples\chap11\Ex8_1>java Ex8_1
設定目前時間(時/分/秒) =>9/20/30

目前是 9 時 20 分 30 秒
請輸入之前時間距離 (時/分/秒) =>2/30/50
```

```
之前是 6 時 49 分 40 秒  
  
目前是 9 時 20 分 30 秒  
請輸入之後時間距離 (時/分/秒) =>1/40/20  
之後是 11 時 0 分 50 秒  
  
目前是 9 時 20 分 30 秒  
請輸入第二點時間 (時/分/秒) =>11/40/20  
兩點時間相差 2 時 19 分 50 秒
```

(B) 製作技巧研討：

製作此系統可區分為下列 3 個主題：

- (1) **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）。
- (2) **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）：
 - (a) **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。
 - (b) **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）。
 - (c) **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）。
 - (d) **timeInterval()**：輸入兩個時間點（時、分、秒），回傳之間的距離時間量。
- (3) **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。

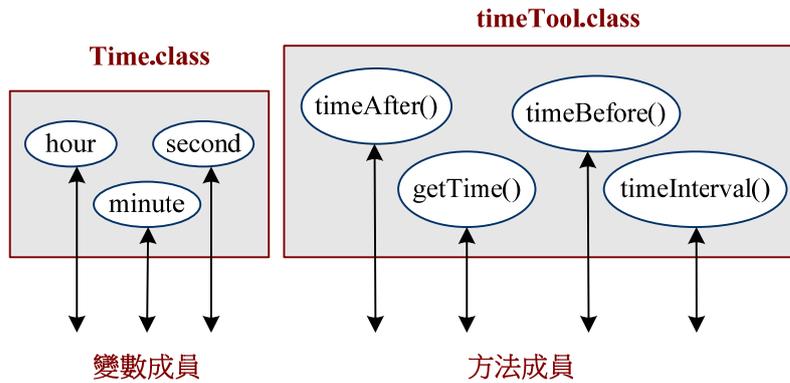


圖 8-2 Time 與 timeTool 類別的功能

```

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
13 /* 時間工具類別，包含 4 個方法 (計算器) */
14 class timeTool {
15
16     /* 轉換字串成為時間物件格式 */
17     Time getTime(String time_S){
18         Time time = new Time();
19         Scanner s = new Scanner(time_S).useDelimiter("/");
20         time.hour = s.nextInt();
21         time.minute = s.nextInt();
22         time.second = s.nextInt();
23         return time;
24     }
25
26     /* 之前時間計算器 */
27     Time timeBefore(Time now, Time value) {
28         Time time = new Time();
29         time.hour=0; time.minute=0; time.second=0;
30         time.second = now.second - value.second;
31         if (time.second < 0) {
32             time.second = time.second + 60;
33             time.minute = 1;
34         }
35     }

```

```
36         time.minute = now.minute - value.minute - time.minute;
37         if (time.minute < 0) {
38             time.minute = time.minute + 60;
39             time.hour = 1;
40         }
41         time.hour = now.hour - value.hour - time.hour;
42         if(time.hour < 0)
43             time.hour = time.hour + 24;
44         return time;
45     }
46
47     /* 之後時間計算器 */
48     Time timeAfter(Time now, Time value) {
49         Time time = new Time();
50         time.hour=0; time.minute=0; time.second=0;
51         time.second = now.second + value.second;
52         if (time.second > 60) {
53             time.second = 60 - time.second;
54             time.minute = 1;
55         }
56         time.minute = time.minute + now.minute + value.minute;
57         if (time.minute > 60) {
58             time.minute = 60 - time.minute;
59             time.hour = 1;
60         }
61         time.hour = time.hour + now.hour + value.hour;
62         if (time.hour > 24)
63             time.hour = time.hour - 24;
64         return time;
65     }
66
67     /* 兩點時間之間距離計算器 */
68     Time timeInterval(Time start, Time end) {
69         Time time = new Time();
70         time.hour=0; time.minute=0; time.second=0;
71         time.second = end.second - start.second;
72         if (time.second < 0) {
73             time.second = time.second + 60;
74             time.minute = 1;
75         }
76         time.minute = end.minute - start.minute - time.minute;
77         if (time.minute < 0) {
78             time.minute = time.minute + 60;
79             time.hour = 1;
80         }
81         time.hour = end.hour - start.hour - time.hour;
82         if(time.hour < 0)
```

```

84         time.hour = time.hour + 24;
85         return time;
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。

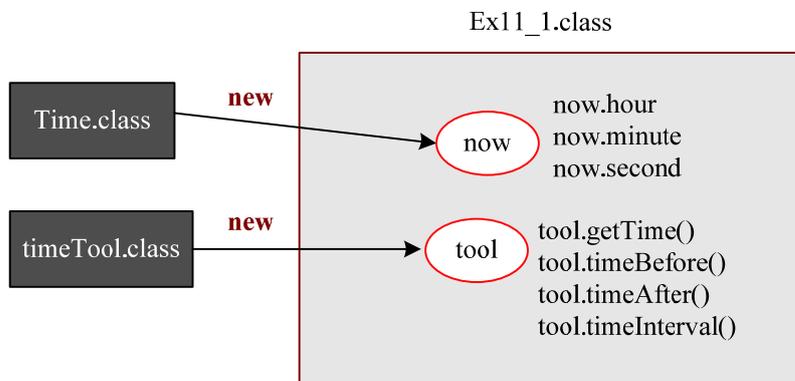


圖 8-3 Time 與 timeTool 類別產生物件

```

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
12         timeTool tool = new timeTool(); // 產生工具，包含 4 種方法
13         Time now = new Time(); // 儲存目前時間
14         Time time_T = new Time(); // 時間暫存變數
15         Time time_B = new Time(); // 時間暫存變數
16         String time_S; // 讀入時間字串
17
18
19
20
    }
}
    
```

```
21      /* 設定目前時間 */
22
23      System.out.printf("設定目前時間(時/分/秒) =>");
24      time_S = keyin.readLine();
25      now = tool.getTime(time_S);
26
27      /* 測試計算之前時間 */
28
29      System.out.printf("\n 目前是 %d 時 %d 分 %d 秒\n",
30                          now.hour, now.minute, now.second);
31
32      System.out.printf("請輸入之前時間距離 (時/分/秒) =>");
33      time_S = keyin.readLine();
34      time_T = tool.getTime(time_S);
35      time_B = tool.timeBefore(now, time_T);
36      System.out.printf("之前是 %d 時 %d 分 %d 秒\n",
37                          time_B.hour, time_B.minute, time_B.second);
38
39      /* 測試計算之後時間 */
40
41      System.out.printf("\n 目前是 %d 時 %d 分 %d 秒\n",
42                          now.hour, now.minute, now.second);
43
44      System.out.printf("請輸入之後時間距離 (時/分/秒) =>");
45      time_S = keyin.readLine();
46      time_T = tool.getTime(time_S);
47      time_B = tool.timeAfter(now, time_T);
48      System.out.printf("之後是 %d 時 %d 分 %d 秒\n",
49                          time_B.hour, time_B.minute, time_B.second);
50
51      /* 測試計算兩時間之間 距離 */
52
53      System.out.printf("\n 目前是 %d 時 %d 分 %d 秒\n",
54                          now.hour, now.minute, now.second);
55
56      System.out.printf("請輸入第二點時間 (時/分/秒) =>");
57      time_S = keyin.readLine();
58      time_T = tool.getTime(time_S);
59      time_B = tool.timeInterval(now, time_T);
60      System.out.printf("兩點時間相差 %d 時 %d 分 %d 秒\n",
61                          time_B.hour, time_B.minute, time_B.second);
62
63      }
64 }
```

程式重點說明：

- (1) 第 11 行：『timeTool tool = new timeTool();』。利用 timeTool 類別產生物件工具 tool(包

含 4 種方法)，但 `timeTool.class` 必須存在於同一目錄下。

- (2) 第 12 行：『`Time now = new Time();`』。利用 `Time` 類別產生物件變數 `now` (包含 3 個變數成員)，但 `Time.class` 必須儲存同一目錄下。
- (3) 第 21 行：『`now = tool.getTime(time_S);`』。呼叫 `tool` 物件下的 `getTime()` 物件方法，傳回 `Time` 格式的物件。
- (4) 第 29 行：『`time_B = tool.timeBefore(now, time_T);`』。呼叫 `tool` 物件下的 `timeBefore()` 方法。
- (5) 第 40 行：『`time_B = tool.timeAfter(now, time_T);`』。呼叫 `tool` 物件下的 `timeAfter()` 方法。
- (6) 第 51 行：『`time_B = tool.timeInterval(now, time_T);`』。呼叫 `tool` 物件下的 `timeInterval()` 方法。

如果依照上述技巧製作本系統，編譯後應該具有下列檔案：

```
G:\Examples\chap11\Ex8_1>dir/b
Time.class
timeTool.java
timeTool.class
Ex8_1.java
Ex8_1.class
```

8-1-4 自我挑戰：遊樂場計費系統

(A) 程式功能：PM8_1.java

『**Karoror 線上遊戲**』係採用按上線時間收費方法，遊戲者上線後開始計時，下線後在計算上線時間多寡計費，以每 30 秒 1 元計算。該公司需要一套收費資訊系統，因此在網路公開招標，期望參與投標者需製作一套雛型管理系統，應具有：設定目前時間、顯示上線遊戲者、開始遊戲、結束遊戲等功能；期望操作介面如下：

```
G:\Examples\chap11\PM8_1>java PM8_1

** Keroro 線上遊戲系統 0 時 0 分 0 秒 **

(1) 設定目前時間          (2) 顯示上線遊戲者
```

```
(3) 開始遊戲, 計費開始    (4) 結束遊戲, 繳費出場
(5) 離開系統
目前線上 0 請輸入工作選項 =>1
請輸入目前時刻(時/分/秒) =>9/12/30

** Keroro 線上遊戲系統  9 時 12 分 30 秒 **
(1) 設定目前時間          (2) 顯示上線遊戲者
(3) 開始遊戲, 計費開始    (4) 結束遊戲, 繳費出場
(5) 離開系統
目前線上 0 請輸入工作選項 =>3
參與玩家暱名 (英文) =>tsnien

** Keroro 線上遊戲系統  9 時 12 分 30 秒 **
(1) 設定目前時間          (2) 顯示上線遊戲者
(3) 開始遊戲, 計費開始    (4) 結束遊戲, 繳費出場
(5) 離開系統
目前線上 1 請輸入工作選項 =>2
姓名            開始時間
tsnien          9 時 12 分 30 秒

** Keroro 線上遊戲系統  9 時 12 分 30 秒 **
(1) 設定目前時間          (2) 顯示上線遊戲者
(3) 開始遊戲, 計費開始    (4) 結束遊戲, 繳費出場
(5) 離開系統
目前線上 1 請輸入工作選項 =>1
請輸入目前時刻(時/分/秒) =>10/23/30

** Keroro 線上遊戲系統  10 時 23 分 30 秒 **
(1) 設定目前時間          (2) 顯示上線遊戲者
(3) 開始遊戲, 計費開始    (4) 結束遊戲, 繳費出場
(5) 離開系統
目前線上 1 請輸入工作選項 =>3
.....
```

(B) 製作技巧提示：

系統必須提供登錄每位遊戲者上線的起始時間。當某位遊戲者下線時，計算當時時間與他的起始之間的相差值，得到他上線的時間，依此計算出上線費用。吾人可利用 Ex8_1 範例所建立的 timeTool (timeInterval() 方法) 物件來計算兩點差距的差距。另外，主程式虛擬碼提示如下：

(Time.class 與 timeTool.class 需與主程式存在於相同目錄下)

```
導入相關套件；
宣告遊戲者類別 class Games{
    String name;           // 遊戲者匿名
    Time start = new Time(); // 遊戲者上線時間
}
宣告主類別 public class PM8_1 {
    static Time clock;           // 靜態時鐘物件
    static BufferedReader keyin; // 靜態鍵盤輸入物件
    static timeTool tool;       // 靜態時間工具物件
    static int number;          // 參與遊戲人數
    static Games[] games;      // 紀錄遊戲者資料(暱名, 入場時間)
    宣告主方法 main {
        keyin = new BufferedReader(new InputStreamReader(System.in));
        tool = new timeTool();
        clock = new Time();
        number = 0;             // 紀錄目前上線者 0
        games = new Games[20]; // 儲存遊戲者陣列
        顯示主功能表 ( get_menu(), select )
        while (select != 5) {
            switch(select) {
                case 1: setClock();           // 設定目前時間
                case 2: dispGame();          // 顯示目前參與遊戲者
                case 3: startGame();         // 開始遊戲
                case 4: stopGame();          // 停止遊戲
            }
        }
    } //主方法結束
    主功能表選單 ( get_menu() );
}
```

```

    設定目前時間 ( setClock() );
    顯示目前參與遊戲者 ( dispGame() );
    參與遊戲, 開始計時 ( startGame() );
    結束遊戲, 繳費出場 ( stopGame() );
} // 主類別結束

```

8-2 類別方法

8-2-1 類別方法宣告與引用

類別可不經由 `new` 產生物件，類別內的方法成員可被直接引用，則稱為類別方法；宣告時只要在方法前面加入 `static` 關鍵字即可。某一類別程式呼叫其他類別方法時，兩類別實體 (`class` 檔案) 必須儲存於同一目錄下；宣告與引用語法如下：

宣告語法：	範例：
<pre> class class_name { static return_type method_name() { method_body; } } // 引用類別方法 class_name.method_name(); </pre>	<pre> class mathFun { static int Add(intx, int y) { } } // 引用類別方法 mathFun.Add(); </pre>

如同一般類別的宣告語法，可以將類別宣告成私有性 (`private class`) 或公有性 (`public class`)，也可將方法成員宣告成私有性 (`private static`) 或公有性 (`public static`)。

8-2-2 範例研討：製作日期計算工具

(A) 程式功能：Ex8_2.java、dateTool.java

許多系統都可能遇到日期計算問題，造成程式設計師困擾。因此，軟體公司裡期望建立相關日期計算工具，爾後所有工程師直接引用即可。該公司對日期計算規劃有下列工具：

- (1) `dataAfter()`：指定某一日期 (年、月、日)，再給予幾個天數之後的日期，應該是何日期。呼叫方法時，給予指定某一特定日期，與之後的天數。執行方法後回覆相對日期的年、月、日之日期。

- (2) `dataBefore()`：指定某一日期，再給予該日期之前的天數，則計算出相對應的日期。呼叫此方法時，給予指定特定日期 (年、月、日)，與之前的天數。執行後回覆相對應的年、月、日。
- (3) `dataInterval()`：指定兩個日期 (年、月、日)，則計算出兩日期之間的天數。呼叫此方法時，給予兩個日期，執行後回覆之間的天數。

期望操作及測試結果如下：

```
G:\Examples\chap11\EX8_2>java Ex8_2
設定目前日期(年/月/日) =>2004/3/12

目前是 2004 年 3 月 12 日
請輸入之前的天數 (日) =>100
之前是 2003 年 12 月 2 日

目前是 2003 年 12 月 2 日
請輸入之後的天數 (日) =>100
之後是 2004 年 3 月 12 日

目前是 2004 年 3 月 12 日
後面的第二個日期 (年/月/日) =>2005/4/6
兩點日期相差 390 天
```

(B) 製作技巧研討：

研發計算日前之前，需制定一個通用型的日期物件 (`Date`)，應包含有：年 (`year`)、月 (`month`) 與 (`day`) 等三個變數成員。另外，每個月的天數並不相同，需要利用一個陣列 (`Months[]`) 來表示每個月最高天數。還有一個重點，如欲利用數值的加減運算子，製作日期計算工具是非常困難的，採用計數累加方法，可能較為容易達成。製作技巧說明如下：

- (1) 之後天數的日期 (`dateAfter()`)：輸入第一點日期，與該日前之後的天數 (`value`)，計算並輸出第二點日期。製作方法是以第一點日期作基礎，由『日』 (`day`) 開始累計加一，當他超過該月最高天數 (`Months[]`) 後，『月』 (`month`) 則累計加一，又如果『月』超過 12 時，則『年』累計加一；如此一個循環則輸入天數 (`value`) 減一，一直到該數值為 0。

最後得到的日期即是第二點日期。

- (2) 之前天數的日期 (`dateBefore()`)：輸入第一點日期，與該日前之前的天數 (`value`)，計算並輸出第二點日期。方法是以第一點日期作基礎，由『日』開始累計減一，當它等於零，則『月』數累計減一，並將『日』設定為該月最高日數；如果『月』減一後等於 0，則『年』減一，亦將『月』設定為 12；如此一個循環則輸入天數 (`value`) 減一，依此類推，一直到輸入數值為 0。
- (3) 兩日期之間天數 (`dateInterval()`)：與 `dataAfter()` 相似，由第一個日期開始累計，每次 `value` 數值累計加一。一直到第一個日期與第二個日期相同，最後 `value` 的數值即是兩日期之間的天數。

日期計算工具大多僅簡單的輸入與輸出運算，極少牽涉到共享資料的處理，製作成靜態類別方法即可。因此我們將宣告日期物件 (`Date`) 與日期工具 (`dateTool()`) 兩類別規劃於 `dateTool.java` 檔案內，編譯後 (`javac`) 將產生 `Data.class` 與 `dateTool.class` 兩類別檔案，其中 `dateTool.class` 包含三個類別方法。另一方面，也製作了一個主程式 (`Ex8_2.java`) 測試這三個類別方法是否正確。

(C) 程式範例：dateTool.java

吾人利用 `dateTool.java` 檔案規劃 `Date` 與 `dateTool` 兩類別。`Date` 類別可宣告產生儲存日期的物件 (年、月、日)。`dateTool` 類別包含 4 個靜態方法成員與 1 個變數成員，分別說明如下：(如圖 8-4 所示)

- (a) `Date` 類別：規劃日期物件的儲存格式，包含年 (`year`)、月 (`month`) 與日 (`day`) 等變數成員。
- (b) `Month[]` 類別變數：儲存每年 12 個月，每月的日數 (12 個元素)。
- (c) `getDate()` 方法：將日期字串 (年/月/日) 轉換成日期物件格式 (年、月、日)。
- (d) `dateBefore()` 方法：計算某一指定日期，相對應之前的天數量，應該是何日前 (年、月、日)。
- (e) `dateAfer()` 方法：計算某一指定日期，相對應之後的天數量，應該是何日前 (年、月、日)。
- (f) `dateInterval()`：計算兩日期之間的天數量。

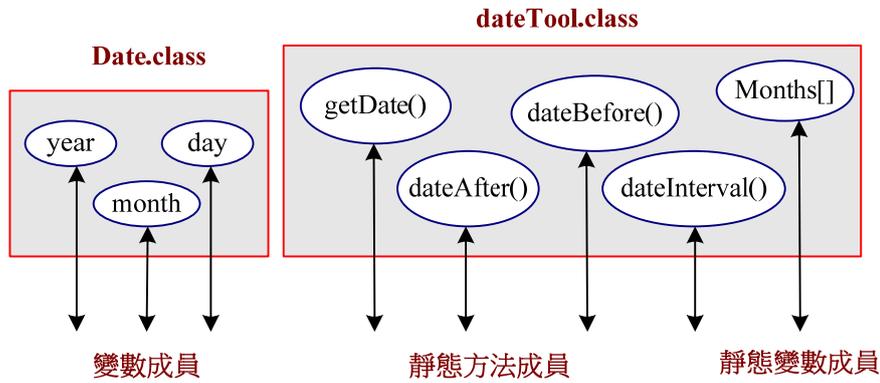


圖 8-4 Date 與 dateTool 類別的功能

```

01 //dateTool.java
02
03 import java.util.*;
04
05 /* 日期類別，產生日期物件的格式 */
06 class Date {
07     int year;
08     int month;
09     int day;
10 }
11
12
13 /* 時間工具類別，包含 4 個方法 (計算器) */
14 class dateTool {
15     /* 每月的天數 */
16     static int Months[] = {31, 28, 31, 30, 31, 30,
17                           31, 31, 30, 31, 30, 31};
18
19     /* 轉換字串成為日期物件格式 */
20     static Date getDate(String date_S){
21         Date date = new Date();
22         Scanner s = new Scanner(date_S).useDelimiter("/");
23         date.year = s.nextInt();
24         date.month = s.nextInt();
25         date.day = s.nextInt();
26         return date;
27     }
28 }
29
30 /* 之後天數的日期計算器 */
31 static Date dateAfter(Date now, int value) {
32     int k;
33     while(value >0) {
34         now.day = now.day + 1;
35         if(now.day > Months[now.month-1]) {

```

```
36         now.day = 1;
37         now.month = now.month + 1;
38         if (now.month > 12) {
39             now.month = 1;
40             now.year = now.year + 1;
41         }
42     }
43     value = value - 1;
44 }
45 return now;
46 }
47
48 /* 之前天數的日期計算器 */
49 static Date dateBefore(Date now, int value) {
50     while(value > 0) {
51         now.day = now.day - 1;
52         if (now.day < 1) {
53             now.month = now.month - 1;
54             if (now.month < 1) {
55                 now.month = 12;
56                 now.year = now.year - 1;
57             }
58             now.day = Months[now.month-1];
59         }
60         value = value - 1;
61     }
62     return now;
63 }
64
65 /* 計算兩個日期之間天數的計算器 */
66 static int dateInterval(Date start, Date end) {
67     int number=0;
68     while(!((end.year == start.year)&&
69             (end.month == start.month)&&
70             (end.day == start.day))) {
71         start.day = start.day + 1;
72         if (start.day > Months[start.month-1]) {
73             start.month = start.month + 1;
74             start.day = 1;
75             if (start.month > 12) {
76                 start.year = start.year + 1;
77                 start.month = 1;
78             }
79         }
80     }
81     number = number + 1;
82 }
83 return number;
```

```

84     }
    }
    
```

程式重點說明：

- (1) 第 6~10 行：『class Date { ... }』。宣告日期類別。產生日期物件的格式。
- (2) 第 13~84 行：『class dateTool { ... }』。宣告日期工具類別。
- (3) 第 15~16 行：『static int Months[] = { ... }』。每月最高日期陣列。

(D) 程式範例：Ex8_2.java

完成時間計算工具製作之後，吾人編寫 Ex8_2.java 程式來驗證工具是否運作正常。又 dateTool 工具內的方法皆被宣告成靜態方法，可不經由產生物件，即可直接引用執行，如圖 8-4 所示。

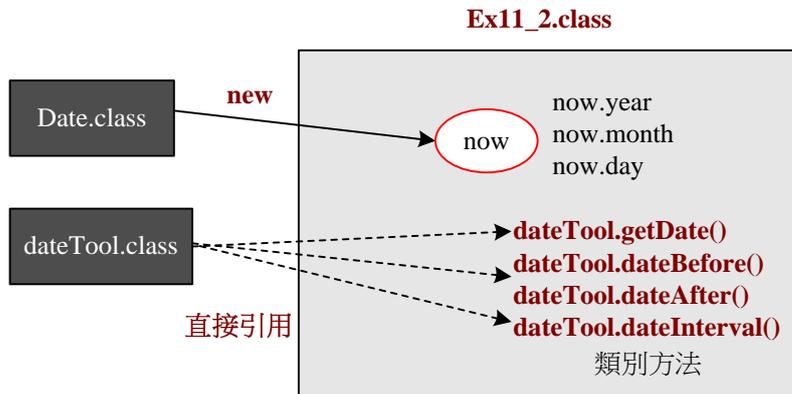


圖 8-5 Date 物件產生與 dateTool 的類別方法

```

01 //Ex8_2.java
02
03 /* dateTool.class, Date.class 需儲存於相同目錄下 */
04 import java.io.*;
05 import java.util.*;
06 public class Ex8_2 {
07     public static void main(String args[]) throws IOException {
08         BufferedReader keyin = new BufferedReader(new
09             InputStreamReader(System.in));
10
11         Date now = new Date();           // 儲存當時日期
12         Date date_T = new Date();       // 日期暫存變數
13         Date date_B = new Date();       // 日期暫存變數
14         String date_S;                  // 讀入日期字串
15         int value;
    
```

```
18
19     /* 設定目前日期 */
20
21     System.out.printf("設定目前日期(年/月/日) =>");
22     date_S = keyin.readLine();
23     now = dateTool.getDate(date_S);
24
25     /* 測試某日前之前天數的日期 */
26
27     System.out.printf("\n 目前是 %d 年 %d 月 %d 日\n",
28                     now.year, now.month, now.day);
29     System.out.printf("請輸入之前的天數 (日) =>");
30     value = Integer.parseInt(keyin.readLine());
31     date_B = dateTool.dateBefore(now, value);
32     System.out.printf("之前是 %d 年 %d 月 %d 日\n",
33                     date_B.year, date_B.month, date_B.day);
34
35
36     /* 測試某日前之後天數的日期 */
37
38     System.out.printf("\n 目前是 %d 年 %d 月 %d 日\n",
39                     now.year, now.month, now.day);
40     System.out.printf("請輸入之後的天數 (日) =>");
41     value = Integer.parseInt(keyin.readLine());
42     date_B = dateTool.dateAfter(now, value);
43     System.out.printf("之後是 %d 年 %d 月 %d 日\n",
44                     date_B.year, date_B.month, date_B.day);
45
46
47     /* 測試計算兩日前之間的天數 */
48
49     System.out.printf("\n 目前是 %d 年 %d 月 %d 日\n",
50                     now.year, now.month, now.day);
51     System.out.printf("後面的第二個日期 (年/月/日) =>");
52     date_S = keyin.readLine();
53     date_T = dateTool.getDate(date_S);
54     value = dateTool.dateInterval(now, date_T);
55     System.out.printf("兩點日期相差 %d 天\n", value);
56 }
57 }
```

8-2-3 自我挑戰：預估葡萄成長採收日期

(A) 程式功能：PM8_2.java

南投縣信義鄉種植葡萄的生長情況大致上是這樣，葡萄收成後農夫會將所有綠葉剪光，此時

讓葡萄樹休養，等到希望葡萄開始成長時，再將成長激素擦拭於藤枝的某些地方，新芽將由擦拭藥水的地方開始成長。由擦拭成長激素日開始計算，其他工作週期如下：

- (a) 7 天後會發新芽。
- (b) 30 天後會長出花苞；
- (c) 之後每隔 10 天必須噴農藥一次，連續 5 次；
- (d) 同時，每隔 20 天必須下普通有機肥料一次，連續 3 次；離最後加肥 40 天，必須加高蛋白有機肥料一次。
- (e) 最後噴農藥後，隔 20 天需葡萄果實需套裝紙袋。
- (f) 從擦拭成長激素到收割葡萄，大約需要 200 天。

農民除了需要依照整個成長過程施肥與噴灑農藥外，也需時常觀察葡萄藤與樹葉的變化，是否『落菌』感染，如發現有被感染必須立即再噴灑消除藥水，絕對不可延遲到隔日，否則就前功盡棄。種植葡萄並不容易，整個成長過程中有稍微瑕疵的話，輕者品質不良，收成當然不好；重者整片葡萄園全部作廢，血本無歸。當然農民也希望辛苦可以得到代價，葡萄可以賣到好價錢。但葡萄要賣到好價錢，除了需要細心照顧外，也要仰賴上天不要有天然災害（雨水過量、寒流、下冰雹、颱風...等），更重要的是採收日期是否剛好合時，譬如：採收當時其他水果產量不豐，或葡萄成長時期天然災害少果實賣像好，就更容易賣到好價錢。由此可見，選定何時可以採收葡萄是很重要的。掌握了這些因素之外，信義鄉梁姓與黃姓兩位年輕農民，便組織了一個葡萄產銷班，期望班友之間採收期錯開。因此，建立一個教導同班農民種植葡萄的教學網站，農民只要輸入預期何時採收葡萄，系統會告知何時擦拭成長激素的，也會一序列告知農民需要從事哪些工作的日期，也隨時提醒農民應該預備哪些工作。所有班友依照此系統種植葡萄，如成長過程與系統不符，則可能種植過程有瑕疵，或系統評估不正確。因此該產銷班必須定期開會，農民將種植心得提出意見，再來修飾系統。請您幫他們製作一個雛型系統，期望操作介面如下：（數據並非確實，僅作教學參考）

```
G:\Examples\chap11\PM8_2>java PM8_2
== 信義鄉葡萄第 # 產銷班 種植施工預估系統 ==
=>本系統評估成長及種植過程僅供農友參考 <=
=>系統有不妥或有新種植技術請與班長聯繫 <=
請輸入預定採收日期 (國曆 年/月/日) =>2008/9/15

預計採收日期 2008 年 9 月 15 日種植工程序列如下:

預估自擦拭成長激素後 200 天可採收
```

```

擦拭日期是 2008 年 2 月 27 日

發芽日期是 2008 年 3 月 6 日

開花日期是 2008 年 4 月 5 日

== 開花後噴灑農藥日期 ==
第 1 次是 2008 年 4 月 15 日
第 2 次是 2008 年 4 月 30 日
第 3 次是 2008 年 5 月 15 日
第 4 次是 2008 年 6 月 4 日
第 5 次是 2008 年 6 月 24 日

== 發芽後施放有機肥料日期 ==
第 1 次是 2008 年 4 月 10 日
第 2 次是 2008 年 4 月 30 日
第 3 次是 2008 年 5 月 20 日
最後一次是高蛋白有機肥料，請到班長處索取配方

套裝紙袋日期是 2008 年 5 月 25 日

== 列印完畢，歡迎再度使用 ==

```

(B) 製作技巧提示：

引用 Ex8_2 範例的日期工具(dateTool)，製作本系統就非常簡單了，只要將 dataTool.java 檔案複製到目前目錄下，再重新編譯即可。除了日期工具外，吾人以虛擬碼提示部份程式要點，如下：

```

/* dateTool.class, Date.class 需儲存於相同目錄下 */

    /* 預估相關成長與施工週期 */

    final int totalDays = 200;           // 所需成長時間 (天)
    final int bornLeaf = 7;             // 發芽時間
    final int bornFlower= 30;           // 開出花苞時間
    final int dust[] = {10, 15, 15, 20, 20}; // 花苞後噴灑農藥週期
    final int organic[] = {5, 20, 20}; // 發芽後有機肥料週期
    final int bagging = 50;             // 開出發苞後需套裝紙袋天數

```

```
/* 宣告所需物件與變數 */  
  
Date cropDate = new Date();           // 收成日期  
Date startDate = new Date();          // 擦拭成長激素日期  
Date leafDate = new Date();           // 發芽日期  
Date bornDate = new Date();           // 開出花苞日期  
Date tempDate = new Date();           // 日期暫存器  
String date_S;                         // 讀入日期字串  
int value;  
.....  
System.out.printf("請輸入預定採收日期 (國曆 年/月/日) =>");  
date_S = keyin.readLine();  
cropDate = dateTool.getDate(date_S);   // 類別方法引用  
列印預計採收日期;  
  
/* 計算擦拭成長激素日期 */  
  
startDate = dateTool.dateBefore(cropDate, totalDays); // 類別方法引用  
列印擦拭成長激素日期;  
  
/* 計算發芽日期 */  
  
leafDate = dateTool.dateAfter(startDate, bornLeaf);   // 類別方法引用  
列印發芽日期;  
  
/* 計算開花日期 */  
  
bornDate = dateTool.dateAfter(startDate, bornFlower); // 類別方法引用  
列印開花日期;  
  
/* 計算噴灑農藥日期 */  
tempDate.year = bornDate.year;  
tempDate.month = bornDate.month;  
tempDate.day = bornDate.day;  
for(int i=0; i<dust.length; i++) {  
    tempDate = dateTool.dateAfter(tempDate, dust[i]); // 類別方法引用  
    列印噴灑農藥日期;  
}  
.....  
.....
```

8-3 方法庫與套件包裝

8-3-1 Java 套件格式

對傳統語言的程式設計而言，一套完整的應用程式是由多個函數所構成。因此，我們可將某些特殊功能編寫成函數程式 (Function)，再編譯成『目的檔』(Object file)，讓其他程式引用，如此可減少許多應用系統開發的時程，也可達到函數重複使用的目的。

一般會將所開發的函數集，依照功能歸類某些特定功能的群組，並可指定特定群組引用，因此稱之為『庫存函數』(Library)。一套系統開發工具 (如 C 語言) 的強弱，其關鍵在於所提供庫存函數的多寡。某些開發工具除了提供標準庫存函數之外，也會提供某些特殊功能的函數庫。譬如，某一套系統具有許多多媒體相關庫存函數，則稱之為『多媒體發展系統』；如果具有許多工程繪圖庫存函數，則稱為『模具設計發展系統』；如果具有許多動畫繪圖庫存函數，則稱為『3D 動畫發展系統』...等等。

Java 也如同傳統語言一樣，可透過各種庫存函數，成為各式各樣的發展系統。更難得可貴的，大多數傳統語言的庫存函數都須購買，譬如具有視窗開發工具的 Visual 系列都必須付費購買。Java 語言則不然，許多開發工具的套件都允許免費下載使用。當然，它也有需付費的企業版販售，但一般系統或初學者，基本版本就非常夠用。

圖 8-1 為 Java 套件結構，跟目錄結構一樣，是以樹狀結構排列。根目錄名稱為 java，每一端點代表一個套件名稱，端點以下可能另一個子套件端點，或該套件下的方法成員。但導入套件時，是以『點』(.) 表示某一目錄，譬如 java.lang.Math。

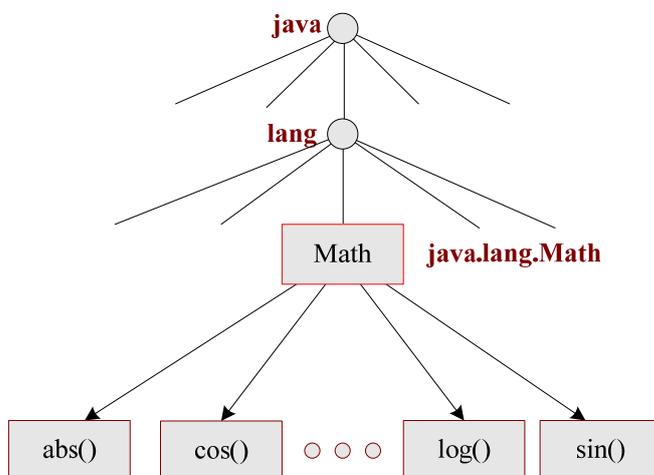


圖 8-6 Java 套件結構圖

自行製作套件 (package) 也是學習 Java 語言不可或缺的。使用者可將常用的函數編製成套件，以供他人或爾後使用。引用自行定義的套件時，必須明確的告訴系統，套件所存放的位置，一

一般都利用 classpath 變數指明套件的目錄位置。當發展出許多套件之後，儲存於電腦上也必須佔用許多空間。Java 發展出一套類似 Unix 系統上的 tar，將一大堆的檔案資料壓縮成較小的空間。Java 發展出來的是 Java ARchive (jar 檔案)。可將 package 包裝於 jar 檔案內，如此可佔用比較小的空間，當被引用時，再展開回復原來的檔案型態。本章僅介紹套件產生方式，限於篇幅不再介紹其他處理。

8-3-2 套件編譯語法 - package

(A) 套件語法 - package

基本上，製作套件與類別/物件方法非常相似。比較特殊的地方是，某些方法被包裝成套件後，大多不會限制使用者的身分如何，也都將其宣告成 public 屬性較為普遍，宣告 package 的語法如下：

套件宣告語法：	範例：
<pre>package package_name; public class class_name { public data_type variable_name; public return_type method_name{ } }</pre>	<pre>package dateTool; public class Date { public int year; public int month; public int day; }</pre>

其中：

- (1) package：指定套件名稱關鍵字。譬如『package dateTool;』則指定套件名稱為 dateTool。套件名稱存在的話，將所製作的類別檔案儲存於該名稱的目錄下；如果不存在，則新建立以套件名稱的目錄，再將類別檔案儲存該目錄底下。
- (2) public class class_name{ ... }：宣告該套件下的類別。譬如『public class Date { .. }』則建立一個 Date.class 類別，並儲存於 dateTool 目錄下。為了要讓其他檔案可以直接引用，必須宣告成 public 屬性。
- (3) public return_type method_name() { .. }：宣告該類別下的方法或物件成員。譬如『public int year;』則宣告一個整數的變數成員；也必須宣告成 public 屬性。

(B) 編譯套件 – javac -d

套件原始檔案製作完成之後，則可將其編譯成套件方法庫。編譯之前必須考慮套件的類別檔案儲存於何目錄下。基本上，在原始檔案內需指定套件名稱，編譯後類別檔案將儲存於該套件名稱的目錄下，但該套件名稱的目錄又在何處？除了編譯時必須指定儲存目錄外，也必須設定 `classpath` 環境變數，命令格式如下：

設定 <code>classpath</code> 環境變數：	範例：
> <code>set classpath=path_name</code>	> <code>set classpath=G:\Examples\timeDate;.</code>
編譯套件命令：	範例：
> <code>javac -d path_name file_name</code>	> <code>javac -d G:\Examples\timeDate Date.java</code>

上述範例中，設定 `classpath` 環境變數需特別注意，內容『`G:\Examples\timeDate;.`』表示設定套件目錄 (`G:\Examples\timeDate`) 與目前目錄 (『`.`』) 兩者。套件目錄如果不存在必須事先利用 Windows 命令建立，編譯後會將套件儲存於該目錄下。

(C) 主程式導入 – import

當然自行開發的套件也可儲存於原 `java` 套件底下，但為了方便管理大多不會這樣；如果沒有儲存原 `java` 預定目錄下，欲導入運用又有稍微麻煩。欲導入自行開發的套件，也須利用 `classpath` 指名套件 (如 `set classpath=G:\Examples\timeDate;.`) 儲存目錄，導入語法如下：

套件導入語法：	範例：
<pre>import class_name; public class main_class { }</pre>	<pre>import dateTool.dateTool; import dateTool.Date; public class Ex8_3 { }</pre>

『`import class_name`』表示導入 `class_name` 類別。譬如『`import dateTool.Date`』表示導入 `dateTool.Date` 類別，但該類別必須儲存於 `G:\Examples\timeDate` 目錄下，然而 `dateTool` 也是目錄，`Date.class` 儲存於該目錄下。

8-3-3 範例研討：時間/日期工具套件

(A) 程式功能：Ex8_3.java

請製作一套時間與日期計算工具的套件，再編寫一主程式導入該套件，並測試套件下類別方法與物件方法引用情形；期望操作結果如下：

```
G:\Examples\chap11\Ex8_3>javac Ex8_3.java

G:\Examples\chap11\Ex8_3>java Ex8_3
設定目前日期(年/月/日) =>2007/8/25
設定目前時間(時/分/秒) =>12/5/30

現在是 2007 年 8 月 25 日 12 時 5 分 30 秒

== 現在時刻輸出完畢 ==
```

(B) 製作技巧研討：

吾人可以稍加修改 Ex8_1 的時間計算工具(timeTool)，以及 Ex8_2 的日期工具(dateTool)，使其合乎套件語法再重新編譯即可。又 timeTool 類別是產生物件方法，而 dateTool 是類別方法，也剛好能滿足系統的要求。系統僅要求輸入字串格式輸入，轉換再輸出日期與時間格式，因此僅使用到 dateTool 類別中 getData() 方法，與 timeTool 類別中 getTime() 方法。

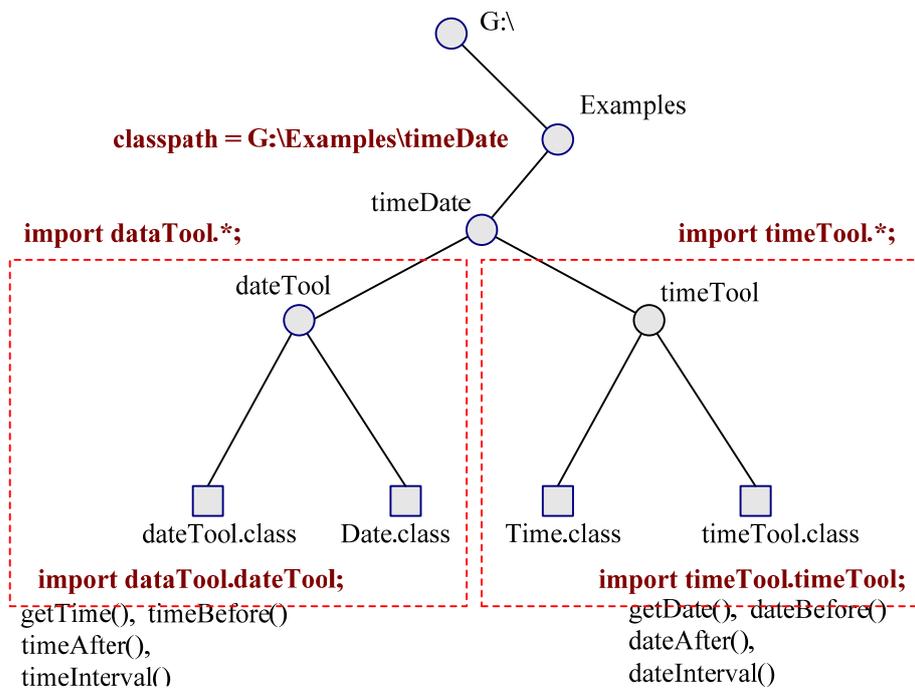


圖 8-7 timeDate 方法庫架構

吾人將 timeTool 物件方法 (Ex8_1 範例) 與 dateTool 類別方法 (Ex8_2 範例)，整合成

timeDate 方法庫，如圖 8-6 所示。首先我們利用 classpath 環境變數指定方法庫儲存目錄位置 (G:\Examples\timeDate)，接下來再修改 Ex8_1 與 Ex8_2 的程式範例，各個程式說明如下：

(C) 程式範例：Date.java

此程式原來是包含於 dateTool.java 檔案內，為了讓其他檔案可以直接引用，因此必須將它製作成獨立檔案，並宣告成 public 屬性。此程式經編譯後會產生 Date.class 類別，功能是產生儲存日期資料的物件，原始檔案如下：

```
01 // Date.java, 產生方法套件語法
02
03 // 儲存於 \Examples\timeDate 目錄下(假設於 g 磁碟機)
04 // 設定環境變數 > set classpath=g:\Examples\timeDate;.
05
06 // 編譯語法: > javac -d \Examples\timeDate Date.java
07
08 package dateTool;
09 public class Date {
10     public int year;
11     public int month;
12     public int day;
13 }
14
15
```

(1) 第 1~4 行：說明使用。

(D) 程式範例：Time.java

原來此程式式包含於 timeTool.java 內，我們將他分離成獨立程式。此程式將產生 Time.class 類別，功能是產生儲存時間資料的物件，程式範例如下：

```
01 // Time.java
02 package timeTool;
03
04 /* 時間類別, 產生時間物件的格式 */
05 public class Time {
06     public int hour;
07     public int minute;
08     public int second;
09 }
10
11
12
```

(E) 程式範例：timeTool.java

吾人還是利用 Ex8_1 範例的 timrTool.java 範例 (包含 4 個物件方法)，將他複製到目前工作目錄下 (chap8\Ex8_3)，並稍加修改使成為套件的原始程式，修改地方以『粗黑』表示，顯示部分檔案如下：

```
01 // timeTool.java
02 // 儲存於 \Examples\timeDate 目錄下(假設於 g 磁碟機)
03
04 // 設定環境變數 > set classpath=g:\Examples\timeDate;.
05 // 編譯語法: > javac -d \Examples\timeDate timeTool.java
06
07 package timeTool;
08 import java.util.*;
09
10 public class timeTool {
11
12     /* 轉換字串成為時間物件格式 */
13     public Time getTime(String time_S){
14         ....
15         ....
16     }
17
18     /* 之前時間計算器 */
19     public Time timeBefore(Time now, Time value) {
20         ....
21         ....
22     }
23
24     /* 之後時間計算器 */
25     public Time timeAfter(Time now, Time value) {
26         ....
27         ....
28     }
29
30     /* 兩點時間之間距離計算器 */
31     public Time timeInterval(Time start, Time end) {
32         ....
33         ....
34     }
35 }
```

(F) 程式範例：dateTool.java

將原來 dateTool.java (Ex8_2 範例，包含 4 個類別方法) 稍加修改使成為套件的原始程式，修改地方以『粗黑』表示，顯示部分檔案如下：

```
01 // dateTool.java, 產生方法套件語法
02
03 // 儲存於 \Examples\timeDate 目錄下(假設於 g 磁碟機)
04 // 設定環境變數 > set classpath=G:\Examples\timeDate;.
05
06 // 編譯語法: > javac -d \Examples\timeDate dateTool.java
07
08 package dateTool;
09 import java.util.*;
10
11 public class dateTool {
12
13     /* 轉換字串成為日期物件格式 */
14     public static Date getDate(String date_S){
15         Date date = new Date();
16         Scanner s = new Scanner(date_S).useDelimiter("/");
17         date.year = s.nextInt();
18         date.month = s.nextInt();
19         date.day = s.nextInt();
20         return date;
21     }
22
23     /* 之後天數的日期計算器 */
24     public static Date dateAfter(Date now, int value) {
25         ....
26         ....
27     }
28
29     /* 之前天數的日期計算器 */
30     public static Date dateBefore(Date now, int value) {
31         .....
32         .....
33     }
34
35     /* 計算兩個日期之間天數的計算器 */
36     public static int dateInterval(Date start, Date end) {
37         .....
38         .....
39     }
40 }
```

(G) 程式範例：Ex8_3.java

吾人依照系統要求，製作一個簡單程式來測試引用套件內方法是否可以成功。圖 8-7 為 Ex8_3 類別內引用方法庫的運用情形，程式範例如下：

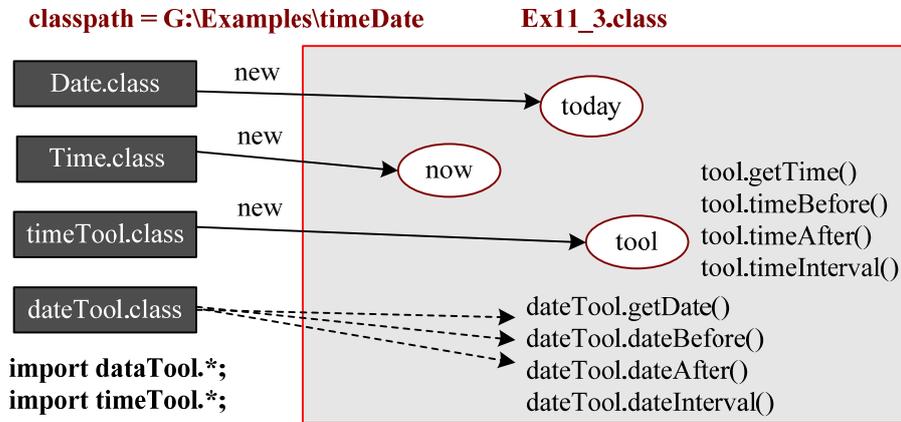


圖 8-8 Ex8_3 引用方法庫範例

```

01 // Ex8_3.java
02 // 庫存方法儲存於 G:\Examples\timeDate 目錄下(假設於 G 磁碟機)
03
04 // 設定環境變數 > set classpath=G:\Examples\timeDate;.
05
06 import dateTool.dateTool;
07 import dateTool.Date;
08 import timeTool.timeTool;
09 import timeTool.Time;
10 import java.io.*;
11 import java.util.*;
12 public class Ex8_3 {
13     public static void main(String args[]) throws IOException {
14         BufferedReader keyin = new BufferedReader(new
15             InputStreamReader(System.in));
16         /* 時間工具是物件方法 */
17
18         timeTool tool = new timeTool(); // 產生時間工具
19
20         Time now = new Time(); // 儲存目前時間
21
22         String time_S; // 讀入時間字串
23
24         /* 日期工具是類別方法 */
25
26         Date today = new Date(); // 儲存今天日期
27
28         String date_S; // 讀入日期字串

```

```

29      /* 設定今天日期 */
30
31      System.out.printf("設定目前日期(年/月/日) =>");
32      date_S = keyin.readLine();
33      today = dateTool.getDate(date_S);      // 引用類別方法
34
35
36      /* 設定目前時間 */
37
38      System.out.printf("設定目前時間(時/分/秒) =>");
39      time_S = keyin.readLine();
40      now = tool.getTime(time_S);           // 引用物件方法
41
42      /* 輸出現在時刻 */
43
44      System.out.printf("\n 現在是 %d 年 %d 月 %d 日",
                          today.year, today.month, today.day);
      System.out.printf(" %d 時 %d 分 %d 秒\n",
                          now.hour, now.minute, now.second);
      System.out.printf("== 現在時刻輸出完畢 ==\n");
    }
}

```

(H) 編譯原始程式

上述總共製作了 5 個原始檔案，其中 4 個檔案是套件程式，另一個是主程式，編譯如下：

```

G:\Examples\chap11\Ex8_3>javac -d \Examples\timeDate dateTool.java
G:\Examples\chap11\Ex8_3>javac -d \Examples\timeDate timeTool.java
G:\Examples\chap11\Ex8_3>javac -d \Examples\timeDate Time.java
G:\Examples\chap11\Ex8_3>javac -d \Examples\timeDate Date.java
G:\Examples\chap11\Ex8_3>javac Ex8_3.java
G:\Examples\chap11\Ex8_3>dir/b
dateTool.java
timeTool.java
Ex8_3.java
Date.java
Time.java
Ex8_3.class

```

我們可以發現，編譯後僅多出來 Ex8_3.class 一個檔案，其他檔案將被放置於所指定的套件目錄下(G:\Examples\timeDate\)，並且以套件名稱為目錄，儲存該套件下的類別檔案，可觀察如下：
(可查閱到 4 個類別檔案儲存位置)

```
G:\Examples\chap11\Ex8_3>cd \Examples\timeDate
```

```
G:\Examples\timeDate>dir/b/s
G:\Examples\timeDate\dateTool          【dateTool 目錄】
G:\Examples\timeDate\timeTool          【timeTool 目錄】
G:\Examples\timeDate\dateTool\Date.class  【類別檔案】
G:\Examples\timeDate\dateTool\dateTool.class  【類別檔案】
G:\Examples\timeDate\timeTool\Time.class  【類別檔案】
G:\Examples\timeDate\timeTool\timeTool.class  【類別檔案】
```

8-4 專題製作

8-4-1 自我挑戰：國際機場停車計費

(A) 程式功能：PM8_3.java

許多人為了方便，短期出國都將汽車停放於國際機場的停車場內，造成停車場內一位難求的情況，民航局只好祭出絕招，完全計時收費。計費方法是不滿 30 分鐘不收費，之後每 30 分鐘收費 20 元，無論停多久（幾天幾夜）都計時收費。因此需要一套計費系統，管理員輸入車輛進出的日期與時間之後，計算出應繳多少停車費。期望操作介面如下：

```
G:\Examples\chap11\PM8_3>java PM8_3
請輸入車輛進入日期(年/月/日) =>2007/8/20
進入時間(時/分/秒) =>12/30/40
請輸入車輛出場日期(年/月/日) =>2007/9/4
出場時間(時/分/秒) =>10/30/50

該車輛進入時刻是 2007 年 8 月 20 日 12 時 30 分 40 秒

        出場時刻是 2007 年 9 月 4 日 10 時 30 分 50 秒

停車費合計: 14300
```

(B) 製作技巧提示：

本範例可引用 Ex8_3 所製作的時間/日期的工具套件。計算兩時刻之間的差距，日期工具是

dateInterval()，時間工具是 timeInterval()。其中有個特殊的地方，如果兩者相距的時數

(tempTime.hour) 比起始時間的時數 (entraTime.hour) 高的話，表示已過了另外一天，則元計算出來的天數需要再扣除一。虛擬碼提示如下：

```
// 需設定環境變數 > set classpath=G:\Examples\timeDate;.

import dateTool.dateTool;
import dateTool.Date;
import timeTool.timeTool;
import timeTool.Time;
.....
.....
輸入車輛進入日期與時間 ( entraDate, entraTime );
輸入車輛出場日期與時間 ( outDate, outTime );
計算日期距離 ( tempDays = dateTool.dateInterval(entraDate, outDate); )
計算時間差距距離 (Time 格式) :
    tempTime = tool.timeInterval(entraTime, outTime);
    if (tempTime.hour > outTime.hour)
        tempDays = tempDays - 1;
/*計算停車費用與輸出：
    int cost;
    int value = (tempDays*24 + tempTime.hour)*60 + tempTime.minute;
    if (value < 30)
        cost = 0;
    else {
        value = value -30;
        cost = value/30 * 20;
    }
    System.out.printf("\n 停車費合計: %d\n", cost);
```

8-4-2 自我挑戰：婦女預產期計算系統

(A) 系統功能

- (1) 請幫家扶中心建立一套婦女安全期預估日期。基本上，受孕危險期是月經來朝日的前 14 天的前後 3 天，又每次月經週期為 28 天 (取平均值)。使用者只要輸入這次月經來朝日期，則系統可顯示下一週期哪幾天是安全期，哪幾天是危險期。婦女再依此訊息決定是否要受孕懷胎。

(引用 dateTool 方法套件)

(2) 家扶中心期望建立一套預產期計算系統，受孕婦女輸入上一次月經日期，則系統會計算懷孕中胎兒成長過程的重要日期，並告知婦女應注意的事項。預計系統顯示下列重要日期與注意事項：

- (a) 胎兒著床的日期：約是最後一次月經週期後的兩週 (14 天)，這是小 BABY 生命在這個世界形成的第一天喔。
- (b) 大約是在 5-10 週的時候 (顯示第 5 與 10 週兩個日期)：是最不穩定的一段時間，這時期要特別小心喔，要避免提重物或太勞累，有些人可能晨間會嘔吐或害喜的很厲害。
- (c) 懷孕的第 12 週：第一妊娠期已經過了，前三個月難過的害喜應該已經不會了，這時間應該多運動以及到醫院作一些胎兒先天性異常的篩檢。
- (d) 懷孕的第 23 週：這個時候可能有些媽媽的體重增加的很快吧，要記得來醫院作妊娠糖尿病以及妊娠毒血症的篩檢喔以及飲食要開始好好的控制了。
- (e) 懷孕的第 27 週：第二妊娠期又過了，這時應該開始準備嬰兒的用品了，醫師已經可以用超音波看出寶寶的性別了，媽媽應該開始與醫師討論生產的方式，準備自然產的媽媽應該開始參加『拉梅姿課程』了。
- (f) 懷孕的第 40 週：親愛的媽咪恭喜你囉，這是妳的預產期，可能會有一些差距，不過先跟家裡的人計劃好唷，有產兆時要如何到醫院，住院生產時要用的東西先打包好並放在隨時都方便拿到的地方喔。

家扶中心希望在網路公布此計算系統，婦女只要輸入上次月經日期，則系統會顯示上述日期與相關注意事項。(引用 dateTool 方法套件)