

## 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：

- `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。
- `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。
- `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。

### (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：

- 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。
- 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。

- 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。

### (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：

- 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。
- Month[] 類別變數：儲存每年 12 個月，每月的日數 (12 個元素)。
- getDate() 方法：將日期字串 (年/月/日) 轉換成日期物件格式 (年、月、日)。
- 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)。
- dateAfere() 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)。
- dateInterval()：計算兩日期之間的天數量。

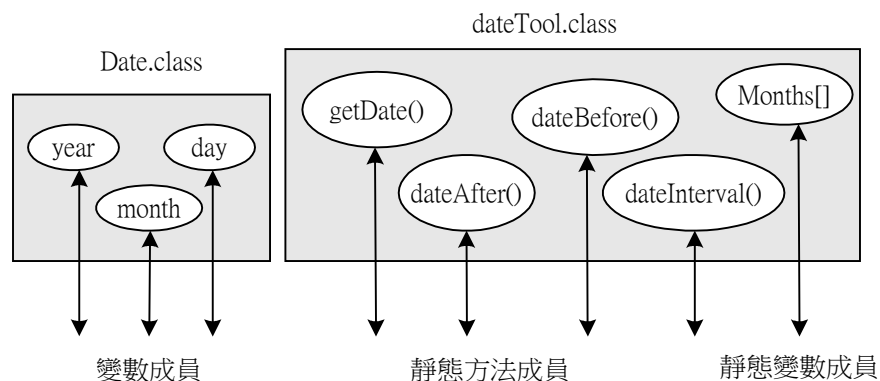


圖 8-4 Date 與 dateTool 類別的功能

```
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 /* 時間工具類別，包含 4 個方法 (計算器) */
13 class dateTool {
14     /* 每月的天數 */
15     static int Months[] = {31, 28, 31, 30, 31, 30,
16                             31, 31, 30, 31, 30, 31};
17
18     /* 轉換字串成為日期物件格式 */
19     static Date getDate(String date_S){
20         Date date = new Date();
21         Scanner s = new Scanner(date_S).useDelimiter("/");
22         date.year = s.nextInt();
23         date.month = s.nextInt();
24         date.day = s.nextInt();
25         return date;
26     }
27
28     /* 之後天數的日期計算器 */
29     static Date dateAfter(Date now, int value) {
30         int k;
31         while(value >0) {
32             now.day = now.day + 1;
33             if(now.day > Months[now.month-1]) {
34                 now.day = 1;
35                 now.month = now.month + 1;
36                 if (now.month > 12) {
37                     now.month = 1;
38                     now.year = now.year + 1;
39                 }
40             }
41             value = value - 1;
42         }
43         return now;
44     }
45
46     /* 之前天數的日期計算器 */
47     static Date dateBefore(Date now, int value) {
48         while(value > 0) {
49             now.day = now.day - 1;
50             if (now.day < 1) {
51                 now.month = now.month - 1;
52                 if (now.month < 1) {
53                     now.month = 12;
54                     now.year = now.year -1;
55                 }
56                 now.day = Months[now.month-1];
57             }
58         }
59     }
60 }
```

```
58         }
59         value = value - 1;
60     }
61     return now;
62 }
63
64 /* 計算兩個日期之間天數的計算器 */
65 static int dateInterval(Date start, Date end) {
66     int number=0;
67     while(!((end.year == start.year)&&
68             (end.month == start.month)&&
69             (end.day == start.day))) {
70         start.day = start.day + 1;
71         if (start.day > Months[start.month-1]) {
72             start.month = start.month + 1;
73             start.day = 1;
74             if (start.month > 12) {
75                 start.year = start.year + 1;
76                 start.month = 1;
77             }
78         }
79         number = number + 1;
80     }
81     return number;
82 }
83 }
84 }
```

程式重點說明：

- 第 6~10 行：『class Date { ...}』。宣告日期類別。產生日期物件的格式。
- 第 13~84 行：『class dateTool { ...}』。宣告日期工具類別。
- 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。

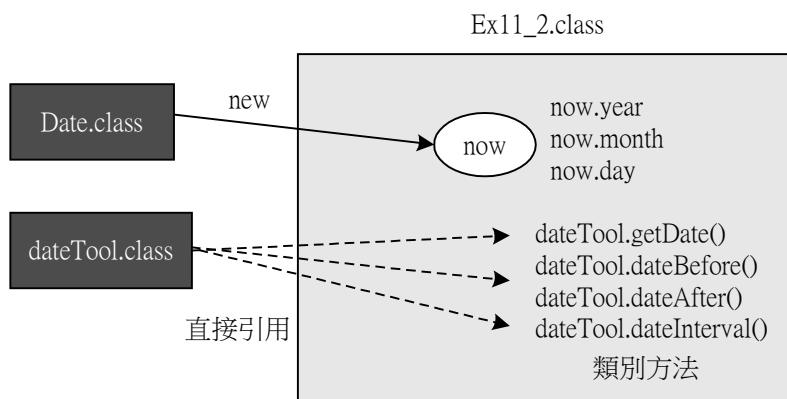


圖 8-5 Date 物件產生與 dateTool 的類別方法

```

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;
16
17         /* 設定目前日期 */
18         System.out.printf("設定目前日期(年/月/日) =>");
19         date_S = keyin.readLine();
20         now = dateTool.getDate(date_S);
21
22         /* 測試某日前之前天數的日期 */
23         System.out.printf("\n目前是 %d 年 %d 月 %d 日\n",
24             now.year, now.month, now.day);
25
26         System.out.printf("請輸入之前的天數 (日) =>");
27         value = Integer.parseInt(keyin.readLine());
28         date_B = dateTool.dateBefore(now, value);
29         System.out.printf("之前是 %d 年 %d 月 %d 日\n",
30             date_B.year, date_B.month, date_B.day);
31
32         /* 測試某日前之後天數的日期 */
33         System.out.printf("\n目前是 %d 年 %d 月 %d 日\n",
34             now.year, now.month, now.day);
35
36         System.out.printf("請輸入之後的天數 (日) =>");
37         value = Integer.parseInt(keyin.readLine());
38         date_B = dateTool.dateAfter(now, value);
39         System.out.printf("之後是 %d 年 %d 月 %d 日\n",
40             date_B.year, date_B.month, date_B.day);

```

```
41
42     /* 測試計算兩日前之間的天數 */
43     System.out.printf("\n 目前是 %d 年 %d 月 %d 日\n",
44                       now.year, now.month, now.day);
45     System.out.printf("後面的第二個日期 (年/月/日) =>");
46     date_S = keyin.readLine();
47     date_T = dateTool.getDate(date_S);
48     value = dateTool.dateInterval(now, date_T);
49     System.out.printf("兩點日期相差 %d 天\n", value);
50
51     }
52 }
```