

第六章 檔案輸入與輸出運用

6-1 檔案輸入/輸出模式

資訊系統大多需要檔案來儲存資料。當使用者輸入建立資料之後，需將所登錄的資料儲存到檔案內，下次管理資料時再由檔案讀入原來資料，如此才能管理龐大資料量。檔案是資料儲存於磁碟機的空間結構，不同的儲存結構也可能影響到資料存取效率，也需要相對應的存取方法來配合。一般資訊系統的檔案型態可區分為順序檔案與紀錄檔案兩大類，前者大多採用串流存取模式；後者大多採用緩衝器存取技巧。以下分別介紹這兩種檔案型態的特性與宣告產生方式。

6-1-1 順序檔案與串流讀寫

將資料依序寫入、讀出檔案。好像錄製錄音帶一樣，由起頭開始依序寫入磁帶，最後標示一個特殊記號表示結束，讀出時也要從頭開始順序讀取，此種儲存檔格式稱為『順序檔案』(Sequence File)。簡單的說，順序檔案允許資料以連串方式，一點一滴依序寫入磁碟檔案內，也允許一點接一點依序由檔案讀出，吾人則將此存取技巧稱之為『串流』(stream) 讀寫。早期順序檔案主要是針對磁帶機的存取方式，但最近網路運用普遍，電腦之間透過網路傳輸資料，大多不喜歡資料傳輸完畢後，再執行下一步驟的處理，大多希望一邊傳輸資料，一邊處理已傳輸部分資料，資料全部傳輸完畢後，也剛好全部處理完畢。譬如，收聽網站上音樂，音樂檔案必須由網站下載到個人電腦上，再由個人電腦撥放該音樂檔案。我們當然希望一邊傳輸音樂檔案，一邊撥放該音樂，全部傳輸完畢後，也剛好收聽完畢，因此就必須利用順序格式讀取音樂檔案，這就是串流的概念。其實，網路上除了音樂檔案之外，無論是視訊、動畫、影像等等檔案也大多如此。也就是說，多媒體檔案大多採用順序檔案格式與串流存取方式，尤其透過網路傳輸也顯得需要。

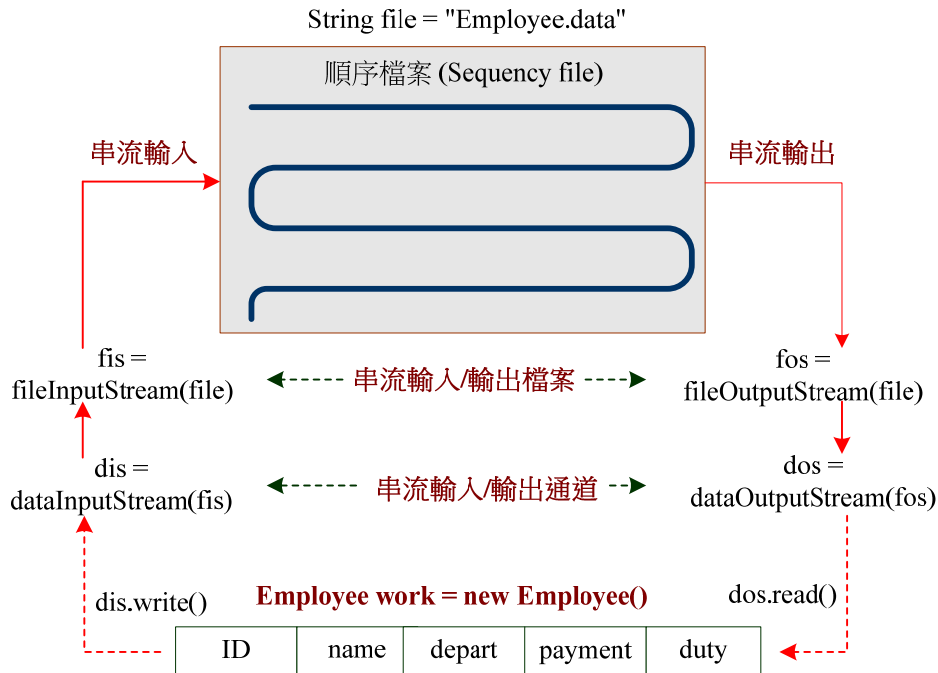


圖 6-1 串流檔案輸入/輸出物件

順序檔案是利用串流 (Stream) 方式，連續性的寫入或讀出。產生檔案時必須宣告它成為串流輸入/輸出檔案；另一方面，也必須將寫入檔案的記憶體資料宣告成串流型態。圖 6-1 是將 Employee.data 檔案宣告成串流輸入/輸出檔案，並產生串流輸入/輸出通道。

6-1-2 紀錄檔案與緩衝器讀寫

描述真實環境現象的資料大多有一定的規範，每樣事件大多由一筆資料記載它的屬性，此筆資料就稱為『紀錄』(Record)。記載許多事件而必須寫入磁碟檔案時，就需要一筆紀錄接一筆，將紀錄依序寫入檔案系統，此格式則稱為『紀錄檔案』(Record file)。一般作業系統為了提高效益，並不立即將紀錄寫入檔案內，而是將紀錄資料組裝成『緩衝器』(Buffer) 格式，暫存於主記憶體的某一固定空間內，等待電腦較有空閒或已累積多筆『緩衝器』資料時，再一併寫入檔案 (或稱磁碟機)。由檔案讀出資料也是一樣，電腦一次由磁碟檔案裡讀出多筆『緩衝器』資料 (儲存於主記憶體內)，再依序由『緩衝器』讀入執行中的程式。一次讀取一筆資料模式大多運用於管理資訊系統的大量資料儲存。也就是說，一般『電子化』管理系統的資料儲存大多採用記錄檔案格式與緩衝器存取模式。

簡單的說，紀錄檔案的存取都需透過緩衝器的運作，因此宣告順序檔案物件時，也須宣告緩衝器物件，並將兩者連結起來成為『程式』的輸入/輸出檔案物件。圖 6-2 為產生紀錄檔

案的宣告方式，它是屬於 java.io 套件的類別。以下分別說明輸入與輸出檔案的宣告方法。

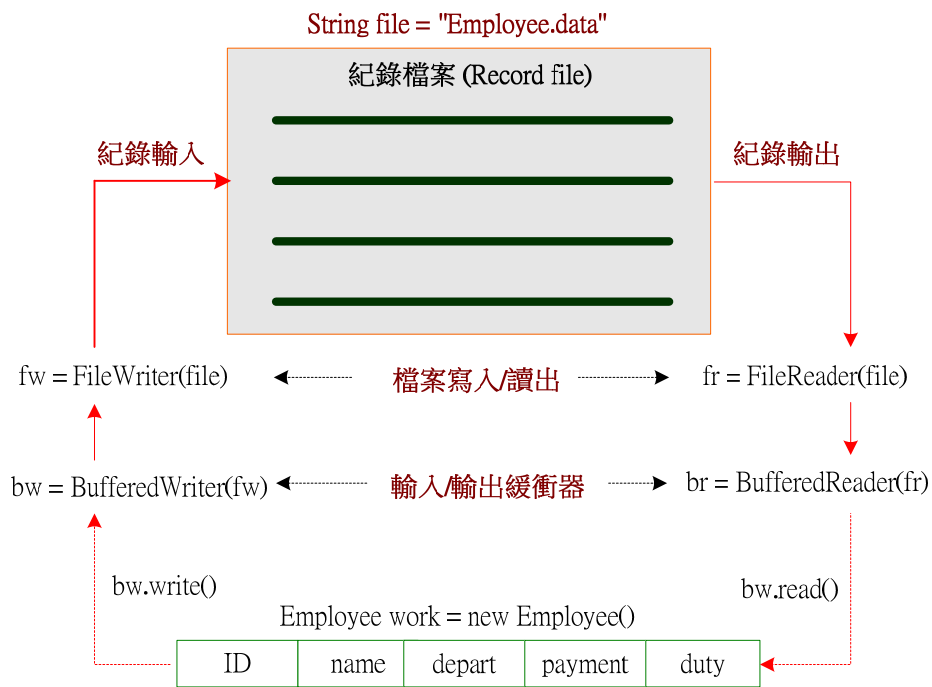


圖 6-2 紀錄檔案輸入/輸出物件

6-2 記錄檔案的存取介面

6-2-1 檔案寫入介面

宣告可寫入記錄檔案的命令(檔案為 Employee.data)，與相關存取方法如下：(如圖 6-2)

宣告『可寫入』紀錄檔案的語法範例：	功能說明：
import java.io.*;	導入 java.io 套件
String file = "Employee.data"	檔案名稱 (範例：Employee.data)
FileWriter fw = new FileWriter(file);	產生可供寫入的紀錄檔案
BufferedWriter bw = new BufferedWriter(fw);	產生可供寫入的緩衝器
void bw.write()	將字元、字串或位元寫入檔案。
void bw.flush()	強迫寫入檔案。
void bw.newLine()	於檔案內開啟新行 (行分隔)。
void bw.close()	關閉檔案。

重點說明如下：

- (1) `fw = FileWriter(file)`：檔案 `file` (如 `Employee.data`) 存在的話，開啟它並清除其內容；不存在的話，則建立該檔案，再將檔案物件存入 `fw` 變數內。
- (2) `bw = BufferedWriter(fw)`：宣告一個可供寫入緩衝器的物件，並連結到可寫入檔案 `fw` 物件；再將緩衝器物件存入 `bw` 變數內。
- (3) 可寫入緩衝器物件的方法有：`write()`、`newLine()`、`close()`。

6-2-2 檔案輸出介面

產生可讀取檔案與可寫入檔案的步驟大致上相同。以 `Employee.data` 檔案為例，產生的宣告步驟如下：(如圖 6-2 所示)

宣告『可讀取』紀錄檔案的語法範例：	功能說明：
<code>import java.io.*;</code>	導入 <code>java.io</code> 套件
<code>String file = "Employee.data"</code>	檔案名稱 (範例： <code>Employee.data</code>)
<code>File fileID = new File(file);</code>	開啟檔案，並回傳檔案識別碼
<code>FileReader fr = new FileReader(fileID);</code>	產生可供讀取的紀錄檔案
<code>BufferedReader br = new BufferedReader(fr);</code>	產生可供讀取的緩衝器
<code>br.read()</code>	緩衝器讀取方法之一

重點說明如下：

- (1) `fileID = new File(file)`：利用 `File` 類別開啟 `file` (`Employee.data`) 檔案物件，如果 `file` 檔案存在的話，則回傳一個整數的檔案識別碼 (存入 `fileID` 物件內)；否則回傳整數值為 `-1`。`File` 物件的方法有 (節錄較常用的；詳請參考 java.sun.com 網站)：

檔案物件的方法：	說明：
<code>boolean canRead()</code>	測試檔案是否可讀取。
<code>boolean canWrite()</code>	測試檔案是否可寫入。
<code>boolean exists()</code>	測試檔案是否存在
<code>boolean isFile()</code>	測試是否是一般檔案。

- (2) `fr = new FileReader(fileID)`：產生一個可讀取的檔案物件，並連結到 `fileID` (檔案識別碼)。
- (3) `br = new BufferedReader(fr)`：產生一個可讀取的緩衝器物件，並連結到 `fr` (可讀取檔案物件變數)。
- (4) 可讀取緩衝器物件的方法有 (節錄較常用的)：

『可讀取』檔案的方法：	說明：
<code>void close()</code>	關閉檔案。
<code>void mark()</code>	於檔案內標示某一位置。
<code>int read()</code>	讀取以字串或位元組格式讀取檔案。
<code>String readLine()</code>	由檔案讀取一行 (或一筆) 紀錄。
<code>void reset()</code>	重置檔案回標示位置。

6-3 紀錄檔案的輸出運用

6-3-1 物件陣列與記錄檔案

電子化管理系統的資料大多有固定格式，每一筆資料就成為一個紀錄。譬如，某一公司的員工管理系統，每一筆紀錄登錄一位員工的資料；也依照系統需求，每一筆資料由若干個屬性所構成。紀錄型態資料大多以一筆接一筆紀錄型態儲存於檔案內，資料系統需要處理這些資料時，再由紀錄檔案讀入主機電腦的記憶體內。然而，紀錄資料於記憶體內儲存方式又有許多格式，譬如陣列、樹狀或鏈路格式等等，即是資料結構的型態。

圖 6-3 為紀錄檔案與陣列資料之間的存取方法。每一陣列元素是一筆紀錄資料 (物件變數)，其中包含了若干屬性的變數資料。陣列資料輸出到紀錄檔案的方法是，每筆資料是一個屬性接一個屬性寫入，屬性之間利用一個分隔符號 (Delimit) 區分，可能是『tab』(`\t`) 空白鍵、分號 (`;`) 或單一空白鍵等等。一筆資料輸出完畢後，再跳到下一行繼續輸出，與輸出到螢幕的方法非常類似。

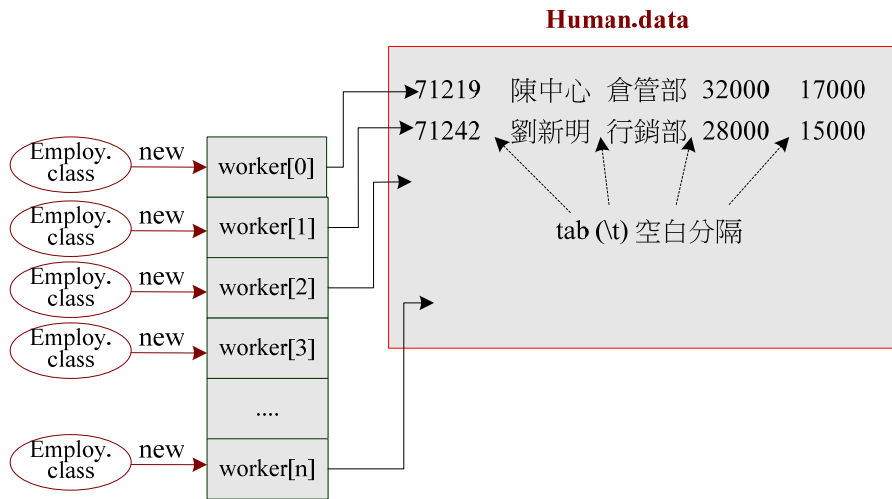


圖 6-3 陣列內物件變數輸出到紀錄檔案

6-3-2 範例研討：登錄員工資料

(A) 程式功能：Ex6_1.java

請幫志明電器製造公司建立一套人事管理系統，該系統允許輸入員工資料，並儲存於資料檔案內 (Human.data)。假設登錄員工資料包含有：員工代號 (int)、姓名 (String)、工作部門 (String)、底薪 (int)、職務加級 (int)。期望操作介面，以及檔案輸出結果如下：

```
G:\Examples\chap9\Ex6_1>java Ex6_1
是否繼續輸入員工資料 (yes/no) =>yes
***** 輸入員工資料 *****
輸入員工姓名 =>陳中心
輸入所屬部門 =>倉管課
輸入員工代號 =>71219
輸入員工底薪 =>32000
輸入薪資加級 =>17000
是否繼續輸入員工資料 (yes/no) =>yes
***** 輸入員工資料 *****
輸入員工姓名 =>劉明新
輸入所屬部門 =>行銷課
輸入員工代號 =>71242
輸入員工底薪 =>31000
```

```

輸入薪資加級 =>15000
是否繼續輸入員工資料 (yes/no) =>no
***** 將員工資料輸出到 Human.dada 檔案內 *****
***** 輸出完畢 *****

G:\Examples\chap9\Ex6_1>dir/b
Employee.class
Human.data
Ex6_1.java
Ex6_1.class

G:\Examples\chap9\Ex6_1>type Human.data          【type 顯示檔案內容命令】
71219   陳中心   倉管課   32000   17000
71242   劉明新   行銷課   31000   15000

```

(B) 製作技巧研討：

本系統製作要點是，宣告產生一個物件陣列，將連續輸入的員工資料寫入其中，輸入完畢之後，再將物件陣列內容一筆接一筆寫入輸出檔案上。須注意的是，輸出檔案上每一行存放一筆員工資料，資料的每個屬性分別由一個欄位表示。虛擬碼提示如下：

```

宣告物件陣列的類別 ( Employee.data );
主類別 ( Ex6_1 ) 範圍：{
    宣告產生物件陣列 ( Employee[] worker; );
    宣告輸出檔案物件：
        String file = "Human.data";
        BufferedWriter data = new BufferedWriter(new FileWriter(file));
    連續讀取多筆員工資料並存入物件陣列中 ( while() { ... } );
    將物件陣列輸出到檔案；
}

```

(C) 程式範例：

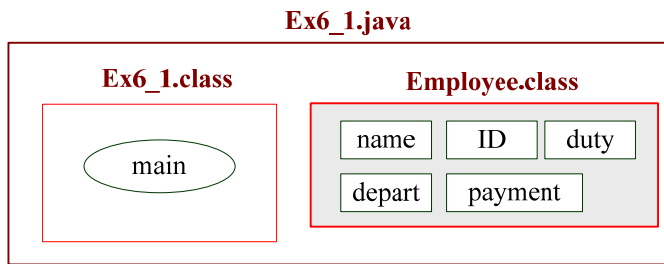


圖 6-4 Ex6_1 程式架構

```

01 //Ex6_1.java
02 /*系統允許輸入員工資料，並儲存於資料檔案內(Human.data)*/
03
04 import java.util.Scanner;
05 import java.io.*;
06
07 class Employee {      /* 物件陣列的類別宣告 */
08     int ID;           // 員工代號
09     String name;     // 員工姓名
10     String depart;   // 所屬部門
11     int payment;     // 底薪
12     int duty;        // 職務加級
13 }
14
15 public class Ex6_1 {
16     public static void main(String args[]) throws IOException {
17         Scanner keyin = new Scanner(System.in);
18         Employee[] worker = new Employee[20];
19
20         /* 宣告輸出檔案物件 */
21         String file = "Human.data";
22         BufferedWriter data = new BufferedWriter(new FileWriter(file));
23         int k=0;
24         String sel;
25
26         /* 讀入資料並存入物件陣列內 */
27         System.out.printf("是否繼續輸入員工資料 (yes/no) =>");
28         sel = keyin.nextLine();
29         while(sel.equals("yes") && (k < 20)) {
30             worker[k] = new Employee();
31             System.out.printf("***** 輸入員工資料 *****\n");
32             System.out.printf("輸入員工姓名 =>");
  
```



```

38         worker[k].name = keyin.nextLine();
39         System.out.printf("輸入所屬部門 =>");
40         worker[k].depart = keyin.nextLine();
41         System.out.printf("輸入員工代號 =>");
42         worker[k].ID = keyin.nextInt();
43         System.out.printf("輸入員工底薪 =>");
44         worker[k].payment = keyin.nextInt();
45         System.out.printf("輸入薪資加級 =>");
46         worker[k].duty = keyin.nextInt();
47         keyin.nextLine(); // 清除鍵盤輸入
48         System.out.printf("是否繼續輸入員工資料 (yes/no) =>");
49         sel = keyin.nextLine();
50         k = k+1;
51     }
52
53     /* 將物件陣列內容寫入檔案內 */
54     System.out.printf("*** 將員工資料輸出到 Human.dada 檔案內 **\n");
55     for (int i=0; i<k; i++) {
56         data.write(worker[i].ID + "\t");
57         data.write(worker[i].name + "\t");
58         data.write(worker[i].depart + "\t");
59         data.write(worker[i].payment + "\t");
60         data.write(worker[i].duty + "\n");
61     }
62     data.close();
63     System.out.printf("***** 輸出完畢 *****\n");
64 }

```

(D) 程式重點說明：

- (1) 第 22~23 行：『String file = "Human.data"; BufferedWriter data ...』。宣告產生一個緩衝器型的輸出檔案物件，指定該檔案位置為 Human.data。
- (2) 第 24 行：『int k;』。利用變數 k 紀錄目前輸入幾筆資料。
- (3) 第 53~59 行：『for(int i=0; i<k; i++) { ...}』。將物件陣列內資料一筆接一筆連續寫入 data 檔案物件；其中 data.write() 為檔案物件的寫入方法。資料屬性之間以『tab』鍵空格位置 (\t)，最後屬性輸出後跳至下一行 (\n)。

(4) 第 60 行：『data.close();』。關閉檔案。目的是強迫將緩衝器內容輸出到檔案上。

6-3-3 自我挑戰：超商列印購物清單

(A) 程式功能：PM6_1.java

擴充範例 Ex5_4.java 程式功能 (本書第五章)。請擴充『春嬌生鮮超市』販賣系統的功能，除了可由螢幕點選客人購買商品與數量外，最後可由螢幕顯示總購買金額，並印出收據 (寫入檔案)。假設僅販賣：可口餅乾 (20 元)、味全鮮乳 (30 元)、御便當 (50 元)、黑松汽水 (20 元)、頻果西打 (30 元) 與脆笛酥 (20 元)，期望操作介面如下：

```
D:\Java2_book\chap6\PM6_1>javac PM6_1.java

D:\Java2_book\chap6\PM6_1>dir/b
Element.class
List.data
PM6_1.class
PM6_1.java
D:\Java2_book\chap6\PM6_1>java PM6_1
**** 歡迎光臨 春嬌超商 請選擇購買商品 ****
(1) 可口餅乾 20 元      (2) 味全鮮乳 30 元      (3) 御便當 50 元
(4) 黑松汽水 20 元      (5) 頻果西打 30 元      (6) 脆笛酥 20 元
(7) 結算金額
      請輸入選項 =>1
      購買數量 =>24
**** 歡迎光臨 春嬌超商 請選擇購買商品 ****
(1) 可口餅乾 20 元      (2) 味全鮮乳 30 元      (3) 御便當 50 元
(4) 黑松汽水 20 元      (5) 頻果西打 30 元      (6) 脆笛酥 20 元
(7) 結算金額
      請輸入選項 =>3
      購買數量 =>15
**** 歡迎光臨 春嬌超商 請選擇購買商品 ****
(1) 可口餅乾 20 元      (2) 味全鮮乳 30 元      (3) 御便當 50 元
(4) 黑松汽水 20 元      (5) 頻果西打 30 元      (6) 脆笛酥 20 元
(7) 結算金額
```

```

    請輸入選項 =>5
    購買數量 =>8
    **** 歡迎光臨 春嬌超商 請選擇購買商品 ****
(1) 可口餅乾 20 元      (2) 味全鮮乳 30 元      (3) 御便當 50 元
(4) 黑松汽水 20 元      (5) 頻果西打 30 元      (6) 脆笛酥 20 元
(7) 結算金額
        請輸入選項 =>7
總購買金額 = 1470

** 採購清單列印中 (List.data) **
** 列印完畢 (List.data) **

G:\Examples\chap9\PM9_1>type List.data      【type 顯示檔案內容命令】
== 春嬌超商 購物清單 歡迎再度光臨 ==
品名            單價      數量      小計
-----
可口餅乾            20        24        480
御便當              50        15        750
頻果西打            30         8        240
總購買金額 = 1470 元整
    
```

(B) 製作技巧提示：

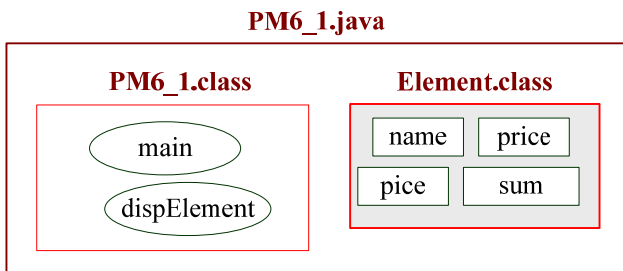


圖 6-5 PM6 1 程式架構

吾人僅修改 Ex5_4.java 範例即可，原來購物清單是由螢幕顯示，將轉換到檔案輸出即可，虛擬碼提示如下：

```
/* 原 Ex5_4.java 程式內容 */
....
宣告產生檔案輸出物件：
    String file = "List.data";
    BufferedWriter data = new BufferedWriter(new FileWriter(file));
.....
將物件陣列內容輸出到檔案內：
    data.write("== 春嬌超商 購物清單 歡迎再度光臨 ==\n");
    data.write("品名\t 單價 \t 數量\t 小計\n");
    data.write("-----\n");
    for (int i=0; i<k; i++) {
        data.write(cust[i].name + "\t");
        data.write(cust[i].price + "\t");
        data.write(cust[i].pice + "\t");
        data.write(cust[i].sum + "\n");
    }
    data.write("總購買金額 = "+total+" 元整");
    data.close();
/* 原 Ex5_4.java 程式內容 */
.....
```

6-4 紀錄檔案的輸入/輸出運用

6-4-1 記錄檔案輸出格式

讀取紀錄檔案大多是每次讀取一筆資料，即是每次讀取檔案內的一行資料，而不論每行包含多少屬性資料。各個屬性都有自己的資料型態，因此讀取時，只好將該筆資料可為一個字串 (String)。圖 6-4 是讀取紀錄檔案的基本運作模式。每次讀取一筆資料並以字串格式儲存於 inData 變數內，亦可直接輸出到螢幕上。(如何轉換成陣列變數資料，下一個範例再說明，Ex6_2_1.java)

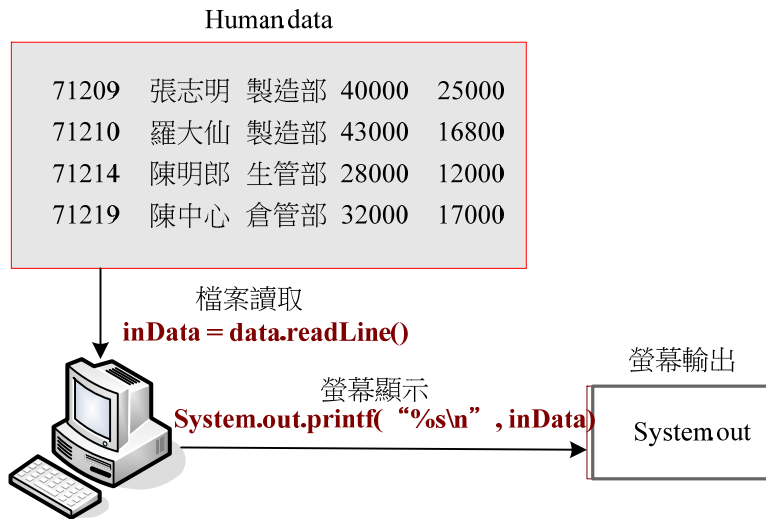


圖 6-6 紀錄檔案整筆輸入，並整筆輸出螢幕顯示

6-4-2 範例研討：建立員工薪資資料

(A) 程式功能：Ex6_2.java

擴充 Ex6_1.java 程式功能。請幫志明電器製造公司建立一套人事管理系統，該系統允許輸入員工資料，並儲存於資料檔案內 (Human.data)。當輸入完畢之後，再由檔案內讀出並顯示螢幕上，觀察資料是否建立正確。假設登錄員工資料包含有：員工代號 (int)、姓名 (String)、工作部門 (String)、底薪 (int)、職務加級 (int)；期望操作介面，以及檔案輸出結果如下：

```
G:\Examples\chap9\Ex6_2>java Ex6_2
是否繼續輸入員工資料 (yes/no) =>yes
***** 輸入員工資料 *****
輸入員工姓名 =>張志明
輸入所屬部門 =>製造課
輸入員工代號 =>71209
輸入員工底薪 =>40000
輸入薪資加級 =>15000
是否繼續輸入員工資料 (yes/no) =>yes
.....
.....
是否繼續輸入員工資料 (yes/no) =>no
** 將資料輸出到 Human.dada 檔案內 **
```

```

**** 輸出完畢 ****

== 由 Human.data 讀取並顯示 ==
代號    姓名    部門    底薪    加級
71209   張志明  製造課  40000   15000
71210   羅大仙  製造課  43000   16800
71214   陳名郎  生管課  28000   12000
71219   蔡中心  倉管課  32000   17000
71242   劉新明  行銷課  28000   15000

== Human.data 輸入完畢 ==
    
```

(B) 製作技巧研討：

本系統包含檔案 (Human.data) 輸出與輸入處理，其運作情形如圖 6-5 所示。也針對檔案輸出與輸入，分別產生下列兩個物件，及其方法如下所示：

檔案名稱	String file = "Human.data";	
輸出物件	BufferedWriter data = new BufferedWriter(new FileWriter(file));	
	寫入方法	data.write()
	關閉方法	dada.close()
輸入物件	BufferedReader data_R = new BufferedReader(new FileReader(fileID));	
	讀取方法	data_R.read()
	關閉方法	data_R.cloas()

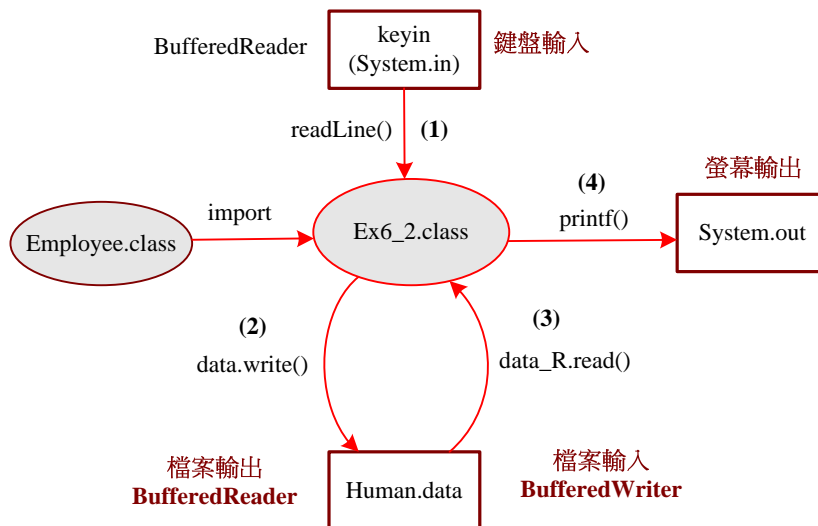


圖 6-7 範例 Ex6_2 程式架構

Ex6_2 程式運作如下：程式開始時宣告產生鍵盤(標示 (1))與輸出檔案物件(標示 (2))，以及一只儲存資料的物件陣列，接著利用 while 迴圈連續讀取多筆資料，系統也隨時存入物件陣列內。操作者輸入完畢之後後，則將物件陣列內容，一筆接一筆寫入檔案 (data.write() ，標示 (2))。輸出完畢後，系統再由檔案讀出 (data_R.read() ，標示 (3))，也一筆接一筆顯示到螢幕上 (標示 (4))。

(C) 程式範例：

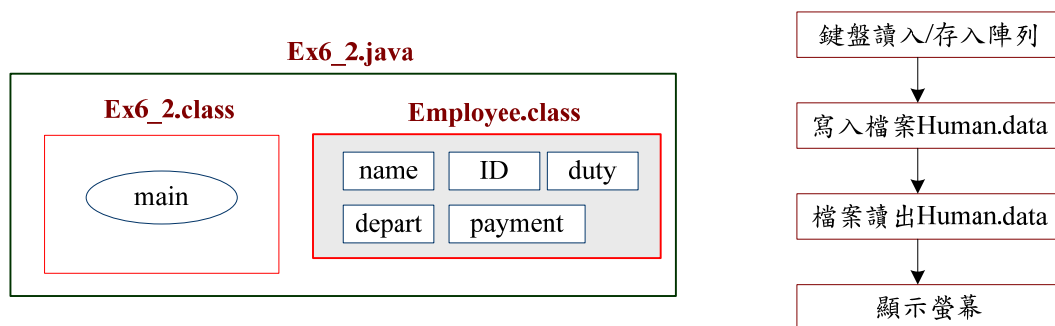


圖 6-8 Ex6_2 程式架構

```

01 //Ex6_2.java
02 /* 擴充 Ex6_1 程式功能，將輸出的結果寫入檔案 */
03
04 import java.util.Scanner;
05 import java.io.*;
06
07 class Employee {      /* 物件陣列的類別宣告 */
08     int ID;           // 員工代號
09     String name;     // 員工姓名
10     String depart;   // 所屬部門
11     int payment;     // 底薪
12     int duty;        // 職務加級
13
14 }
15
16
17 public class Ex6_2 {
18     public static void main(String args[]) throws IOException {
19         Scanner keyin = new Scanner(System.in);
20         Employee[] worker = new Employee[20];
21
22
  
```

```
23      /* 宣告輸出檔案物件 */
24      String file = "Human.data";
25      BufferedWriter data = new BufferedWriter(new FileWriter(file));
26      int k=0;
27      String sel;
28
29
30      /* 讀入資料並存入物件陣列內 */
31      System.out.printf("是否繼續輸入員工資料 (yes/no) =>");
32      sel = keyin.nextLine();
33      while(sel.equals("yes") && (k <20)) {
34          worker[k] = new Employee();
35          System.out.printf("***** 輸入員工資料 *****\n");
36
37          System.out.printf("輸入員工姓名 =>");
38          worker[k].name = keyin.nextLine();
39          System.out.printf("輸入所屬部門 =>");
40          worker[k].depart = keyin.nextLine();
41          System.out.printf("輸入員工代號 =>");
42          worker[k].ID = keyin.nextInt();
43          System.out.printf("輸入員工底薪 =>");
44          worker[k].payment = keyin.nextInt();
45          System.out.printf("輸入薪資加級 =>");
46          worker[k].duty = keyin.nextInt();
47          keyin.nextLine();           //清除鍵盤緩衝器
48          System.out.printf("是否繼續輸入員工資料 (yes/no) =>");
49          sel = keyin.nextLine();
50          k = k+1;
51      }
52
53      /* 將物件陣列內容寫入檔案內 */
54      System.out.printf("*** 將資料輸出到 Human.dada 檔案內 ***\n");
55      for (int i=0; i<k; i++) {
56          data.write(worker[i].ID + "\t");
57          data.write(worker[i].name + "\t");
58          data.write(worker[i].depart + "\t");
59          data.write(worker[i].payment + "\t");
60          data.write(worker[i].duty + "\n");
61      }
62      data.close();
63      System.out.printf("***** 輸出完畢 *****\n");
64
```



```

69
70     /* 檔案輸出程式, 每次讀取一筆資料 */
71
72     System.out.printf("\n== 由 Human.data 讀取並顯示 ==\n");
73     String inData;
74     File fileID = new File(file);           //開啟 Human.data 檔案
75
76     if (fileID.exists()) {
77         BufferedReader data_R = new BufferedReader(new FileReader(fileID));
78         System.out.print("代號\t 姓名\t 部門\t 底薪\t 加級\n");
79         while ((inData=data_R.readLine()) != null)
80             System.out.printf("%s\n", inData);
81         data_R.close();
82     }
83     else
84         System.out.printf("%s 檔案讀取錯誤\n", file);
85
86     System.out.printf("== Human.data 輸入完畢 ==\n");
87 }

```

(D) 程式重點說明：

- (1) 第 3~59 行：與 Ex8_4.java 程式相同，功能是連續由鍵盤輸入資料，並隨時存入物件陣列內；輸入完畢後，再將物件陣列內容輸出到 Human.data 檔案內。
- (2) 第 65 行：『File fileID = new File(file);』。宣告一個檔案物件，以供輸入使用，該檔案的名稱是 『String file = "Human.data"』。
- (3) 第 66 行：『if(fileID.exist()) { ... }』。如果檔案開啟成功的話，則繼續讀取檔案內容；否則通知該檔案不存在。由此可見，此程式欲能正常運作需事先該檔案，無論是否已建立資料。備註：可利用下列方法 (或利用記事本) 建立空白檔案：

```

G:\Examples>copy con: Human.data
^Z
複製了          1 個檔案。

```

- (4) 第 67~68 行：『BufferedReader data_R = new BufferedReader(new FileReader(fileID));』。如果檔案存在的話，則利用該檔案建立一個可供讀取的緩衝器物件，名稱為 data_R。
- (5) 第 70~71 行：『while ((inData=data_R.readLine()) != null){...}』。如果利用

`data_R.readLine()` 方法讀取資料成功的話，則將該資料顯示到螢幕上。這裡利用 `readLine()` 方法讀取，表示每次由檔案內讀取一行，也隨之將該行資料輸出螢幕上。

(6) 第 72 行：『`data_R.close();`』。讀取完畢後，關閉該檔案。

6-4-3 自我挑戰：列印員工薪資表

(A) 程式功能：PM6_2.java、Employee.class

志明電器公司已建立完成員工薪資管理系統 (Ex6_2.java)，該系統允許管理者輸入員工薪資資料，並可除儲存於 `Human.data` 檔案內。請編寫一套程式可供列印公司所有員工的薪資表，員工資料直接由 `Human.data` 檔案讀取。薪資表包含員工代號、姓名、服務部門，薪資總額 (= 底薪 + 職務加級)、預扣稅額 (= 薪資總額 * 0.1) 與應領薪資 (薪資總額 - 預扣稅額); 期望操作如下：

```
D:\Java2_book\chap6\PM6_2>javac PM6_2.java

D:\Java2_book\chap6\PM6_2>dir/b
Employee.class
Human.data
PM6_2.class

D:\Java2_book\chap6\PM6_2>java PM6_2
== 志明電器公司 員工薪資表 ==
員工資料                薪資總額    預扣稅額    應領薪資
張志明(71209 製造課)    55000      5500       49500
羅大仙(71210 製造課)    59800      5980       53820
陳名郎(71214 生管課)    40000      4000       36000
蔡中心(71219 倉管課)    49000      4900       44100
劉新明(71242 行銷課)    43000      4300       38700

薪資總額 = 246800 元整  總稅額 = 24680 元整  總應額 = 222120 元整
```

(B) 製作技巧研討：

本範例最大的特點是，每筆資料由檔案讀入後，如何儲存於物件陣列內，處理方式如

圖 6-6 所示。首先系統利用 `readLine()` 方法讀入整筆資料後儲存於 `inData` 變數內，該資料還是屬於字串格式 (String)。接著將 `inData` 導入 `Scanner` 類別，宣告其物件 (`data_R`)，其中必須設定字串資料之中各個欄位的『分界符號』(Demiliter)，再利用 `Scanner` 物件方法，分別讀取各欄位資料再寫入陣列物件的變數成員內。相關之物件及其方法如下：

範例：	說明：
<code>inData=data.readLine()</code>	由檔案讀取一筆資料。
<code>Scanner s = new Scanner(inData).useDelimiter("\t");</code>	宣告成 <code>Scanner</code> 物件並設定分界符號 (<code>\t</code>)，物件名稱為 <code>s</code> 。
<code>worker[k].ID = s.nextInt();</code>	讀取一整數，再存入陣列物件的變數成員內。

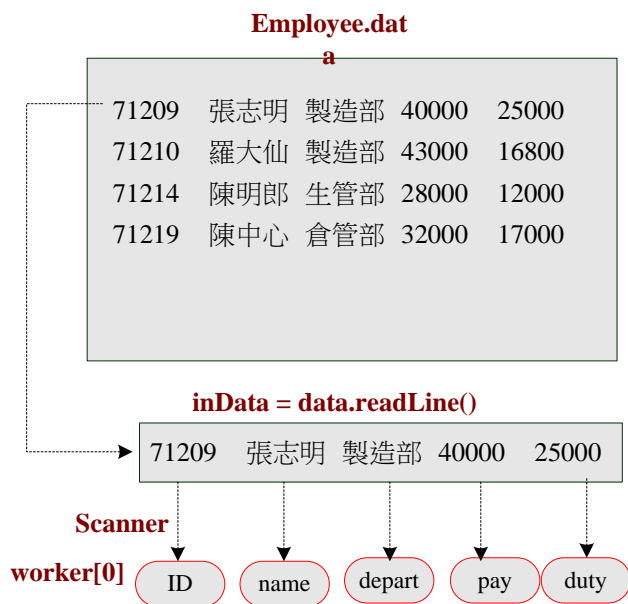


圖 6-9 紀錄檔案讀入，每筆儲存於物件陣列內

(C) 程式片段：

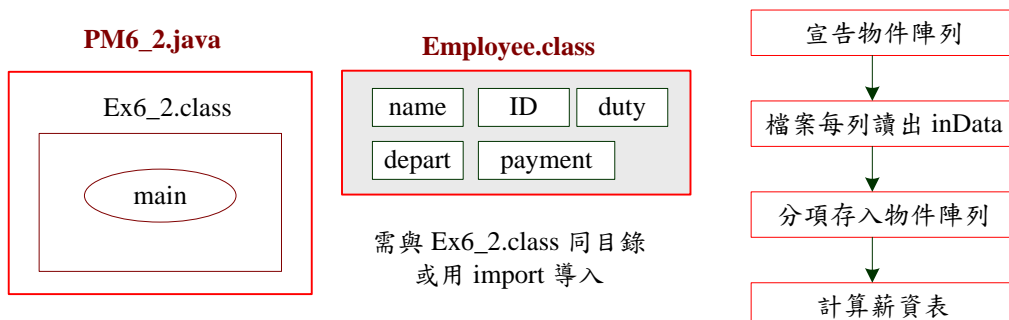


圖 6-10 PM6 2 程式架構

```
01 //PM6_2.java, Human.data 檔案需存在
02
03 /* Employee.class 需存在於同目錄下 */
04 import java.io.*;
05 import java.util.Scanner;
06 public class PM6_2 {
07     public static void main(String args[]) throws IOException {
08         Employee[] worker = new Employee[20];
09         String file = "Human.data";
10         String inData;
11         int k=0;
12         File fileID = new File(file);
13         if (fileID.exists()) {
14
15             /* 由 Humam.data 讀取員工資料並存入 worker[] 陣列 */
16             BufferedReader data = new BufferedReader(new
17                 FileReader(fileID));
18             while ((inData=data.readLine()) != null) {
19                 Scanner s = new Scanner(inData).useDelimiter("\t");
20                 worker[k] = new Employee();
21                 worker[k].ID = s.nextInt();
22                 worker[k].name = s.next();
23                 worker[k].depart = s.next();
24                 worker[k].payment = s.nextInt();
25                 worker[k].duty = s.nextInt();
26                 k = k +1;
27             }
28             data.close();
29
30             /* 計算並列印薪資表 */
31             System.out.printf("== 志明電器公司 員工薪資表 ==\n");
32             System.out.print("員工資料\t\t 薪資總額\t 預扣稅\t 應領薪資\n");
33             .....
34             .....
35             .....
36             .....
37         else
38             System.out.printf("%s 檔案不存在\n", file);
39     }
40 }
```

(D) 程式重點分析：

- (1) 第 17~26 行：『while ((inData=data.readLine()) != null) { ... }』。每次讀取一筆資料，分別填入陣列物件的變數成員內。
- (2) 第 18 行：『Scanner s = new Scanner(inData).useDelimiter("\t");』。宣告產生一個 Scanner 物件，其名稱為 s，也指定欄位分界符號為 『\t』。
- (3) 第 19 行：『worker[k] = new Employee();』。產生一個 Employee 物件，並填入 worker 陣列的第 k 元素 (worker[k]) 內。
- (4) 第 20~24 行：『worker[k].ID = s.nextInt(); ...』。利用 s.nextInt() 物件方法，讀取一個整數，再填入 worker[k].ID 變數成員內。

6-5 專題製作

6-5-1 範例研討：建立人事管理資訊系統

(A) 程式功能：Ex6_3.java、Personal.java

『展鵬網路行銷公司』希望建立一個電子化管理系統，其中包含若干個子系統，各子系統所使用的員工資料都能夠統一處理，才不至於發生各系統之間不一致的問題。請您幫該公司規劃員工資料的屬性，並建立『人事資料管理系統』(Ex6_3.java) 可讓管理人員『新增』、『修改』與『刪除』員工資料，爾後其他子系統也可引用該資料 (Human.data)。目前預估員工資料包含有：員工代號、姓名、服務部門、性別、出生、底薪、職務加級等屬性，並由獨立檔案製作而成 (Personal.java)，經過編譯後產生 Personal.class。

至於員工資料請以 Human.data 檔案名稱儲存。第一次執行沒有 Human.data 檔案，則利用編輯工具(如：notepad++) 開啟一個新檔，直接另存新檔並將檔名取 “Human.data”(雙引號包起來，不要輸入任何內容)，再儲存於本程式的同一目錄下即可。

期望操作介面與執行結果如下：

- (1) 執行該檔案之前，需先建立 Human.dat，如果該檔案不在，則會要求先建立後再執行，如下：(系統啟動時，會將 Human.dat 檔案內容讀入)

```
D:\Java2_book\chap6\Ex6_3>java Ex6_3
```

```
Human.data 檔案不存在, 請先建立它
按<enter>鍵離開 =>
```

- (2) 系統包含 6 個工作選項，無論增加、修改或刪除員工資料之後，都必儲存資料才有效，亦是，離開系統(選項 6) 之前必須儲存資料(選項 5)，選單如下：

```
D:\Java2_book\chap6\Ex6_3>java Ex6_3
=== 人事資料管理系統 ===
(1) 顯示所有資料      (2) 增加員工資料      (3) 修改員工資料
(4) 刪除員工資料      (5) 儲存資料          (6) 離開系統
請選擇工作項目 =>
```

- (3) 選擇 (1) 表顯示系統內所有員工資料，如下：

```
請選擇工作項目 => 1
= 列印所有員工資料 ==
代號   姓名   部門   性別   出生   底薪   加級
101    王力宏 會計課 男     63/12/5 80000  40000
102    張惠妹 業務課 女     64/3/12 90000  70000
103    蔡依玲 製造課 女     67/3/15 80000  60000
```

- (4) 工作選項(2)是增加員工資料，操作介面如下：

```
請選擇工作項目 => 2
請輸入員工編號 =>7106
請輸入姓名 =>劉有得
請輸入服務部門 =>資訊部
請輸入性別 (男/女) =>男
請輸入生日 (西元/月/日) =>71/09/12
請輸入底薪 =>54000
請輸入職務加級 =>11000
```

- (5) 工作選項(3)是修改員工資料，操作介面如下：

```
請選擇工作項目 => 3
```

```

請輸入欲修改員工的代號 =>7105
請輸入 張有德 姓名 =>張有得
請輸入 資訊部 服務部門 =>製造部
請輸入性別 男 (男/女) =>男
請輸入生日 75/09/23 (年/月/日) =>76/09/23
請輸入 60000 底薪 =>50000
請輸入 20000 職務加級 =>20000
    
```

(6) 工作選項(4)是刪除員工資料，操作介面如下：

```

請選擇工作項目 => 4
請輸入欲刪除員工的代號 =>7105
7105 資料已刪除 !!
    
```

(7) 工作選項(5)是將目前員工資料儲存到檔案內，操作介面如下：

```

請選擇工作項目 => 5
** 將儲存檔案 (Human.dada) **
***** 儲存完畢 *****
    
```

(B) 製作技巧研討：

吾人將製作本系統的重點歸類成下列重點：（如圖 6-7 所示）

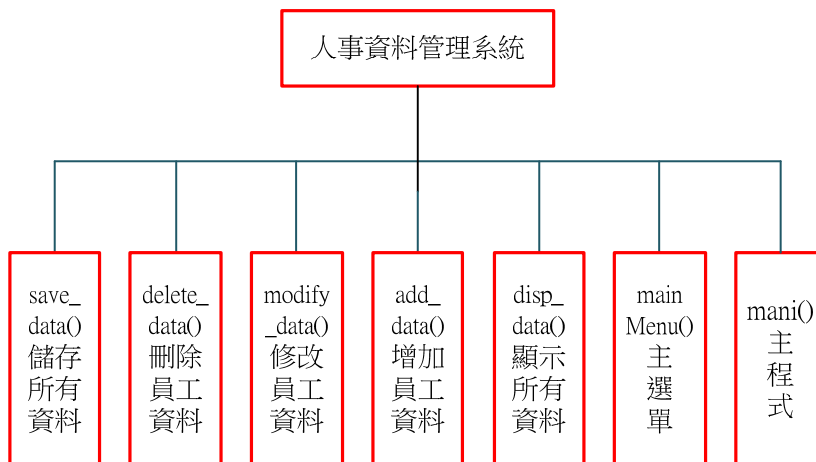


圖 6-11 人事資料管理系統架構

(1) 建立一個專屬員工資料類別的原始檔案 (Personal.java)，類該類別內的變數成員較能符

合其他管理系統運用。

- (2) 需要針對 Human.data 檔案宣告產生輸入與輸出物件。
- (3) 可依照本系統所提供的功能，劃分多個子系統再分別實現。為了讓子系統能直接處理員工資料，而將其物件陣列 (Personal[] emp;) 與登記所儲存資料筆數的變數 (int k;)，設定成類別靜態變數 (static)。
- (4) 主方法(main()):宣告檔案(Human.data)輸入與輸出物件、儲存資料的物件陣列(emp[])。讀取 Human.data 檔案內容，並分別將各個欄位內容填入相對應的物件陣列的變數成員內。再利用 while/switch/case 迴圈讀取操作者的工作選項。
- (5) 主選單程式 (main_menu()): 顯示本系統所提供的工作項目。
- (6) 顯示資料子程式 (disp_data()): 將目前物件陣列 (emp[]，靜態變數) 登錄的員工資料，輸出顯示到螢幕上。
- (7) 增加資料子程式 (add_data()): 可輸入增加一位員工資料到物件陣列 (emp[]) 上。
- (8) 修改資料子程式 (modify_data): 修改物件陣列 (emp[]) 內某位員工的資料。
- (9) 刪除資料子程式 (delete_data): 刪除物件陣列 (emp[]) 內某位員工的資料。
- (10) 儲存資料子程式 (save_data): 依序將物件陣列 (emp[]) 各筆員工資料輸出到檔案 (Human.data) 上。

(C) 程式範例：(執行前，須建立 Human.data 檔案)

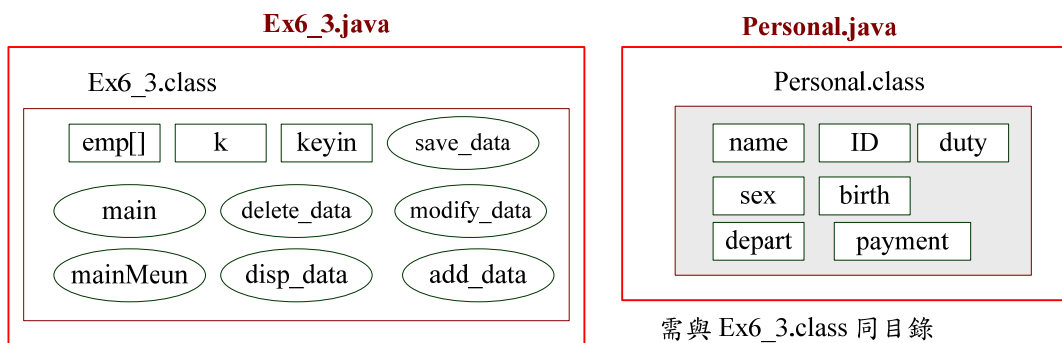


圖 6_12 Ex6_3 程式架構

(a) Personal 類別的原始檔案

```
01 //Personal.java
02
03 class Personal {
04     int ID;           // 員工代號
05     String name;     // 員工姓名
06     String depart;   // 服務部門
07     String sex;      // 性 別
08     String birth;    // 出生日期
09     int payment;     // 底 薪
10     int duty;        // 職務加級
11
12 }
}
```

(b) 主類別的原始檔案

```
01 //Ex6_3.java
02 /* 請建立一套人事資料管理系統，可讓管理人員新增、修改、刪除員工資料 */
03
04 import java.io.*;
05 import java.util.Scanner;
06 public class Ex6_3 {
07     static Personal[] emp;           // 員工資料的物件陣列
08     static int k;                   // 紀錄儲存筆數
09     static Scanner keyin;           // 鍵盤輸入物件
10
11     public static void main(String args[]) throws IOException {
12         keyin = new Scanner(System.in);
13
14         /* 讀取 Human.data 檔案資料，再存入 emp 物件陣列中 */
15         emp = new Personal[50];
16         k=0;                          // 讀取資料的筆數
17         String inData;
18         String file_R = "Human.data";
19         File fileID = new File(file_R); // 產生輸入檔案物件
20         if (fileID.exists()) {
21             BufferedReader data = new BufferedReader(new
22                 FileReader(fileID));
23         }
24     }
25 }
26 }
```

```
27         while ((inData=data.readLine()) != null) {
28             Scanner s = new Scanner(inData).useDelimiter("\t");
29             emp[k] = new Personal();
30             emp[k].ID = s.nextInt();
31             emp[k].name = s.next();
32             emp[k].depart = s.next();
33             emp[k].sex = s.next();
34             emp[k].birth = s.next();
35             emp[k].payment = s.nextInt();
36             emp[k].duty = s.nextInt();
37             k = k +1;
38         }
39         data.close();
40     }
41     else {
42         System.out.printf("%s 檔案不存在，請先建立它\n", file_R);
43
44         System.out.printf("按任何鍵離開 =>");
45         keyin.nextLine();
46         System.exit(1);
47     }
48     mainMenu();
49     int select = keyin.nextInt();
50     while (select !=6) {
51         switch (select) {
52             case 1:      /* 顯示員工資料 */
53                 disp_data();
54                 break;
55             case 2:      /* 增加員工資料 */
56                 add_data();
57                 break;
58             case 3:      /* 修改員工資料 */
59                 modify_data();
60                 break;
61             case 4:      /* 刪除資料 */
62                 delete_data();
63                 break;
64             case 5:      /* 儲存資料 */
65                 save_data();
66                 break;
67             default:
68                 System.out.printf("錯誤輸入，請重新選擇 !!\n");
69         }
70     }
71 }
72 }
```

```
73         mainMenu();
74         select = keyin.nextInt();
75     }
76 }
77 public static void mainMenu() {
78     System.out.printf("\n=== 人事資料管理系統 ===\n");
79
80     System.out.printf("(1) 顯示所有資料\t");
81     System.out.printf("(2) 增加員工資料\t");
82     System.out.printf("(3) 修改員工資料\n");
83     System.out.printf("(4) 刪除員工資料\t");
84     System.out.printf("(5) 儲存資料\t");
85     System.out.printf("(6) 離開系統\n");
86     System.out.printf("請選擇工作項目 => ");
87
88 }
89
90 public static void disp_data() {
91     System.out.printf("== 列印所有員工資料 ==\n");
92     System.out.printf("代號\t姓名\t部門\t性別\t出生\t底薪\t加級\n");
93     for (int i=0; i<k; i++) {
94         System.out.printf("%d\t", emp[i].ID);
95         System.out.printf("%s\t", emp[i].name);
96         System.out.printf("%s\t", emp[i].depart);
97         System.out.printf("%s\t", emp[i].sex);
98         System.out.printf("%s\t", emp[i].birth);
99         System.out.printf("%d\t", emp[i].payment);
100        System.out.printf("%d\n", emp[i].duty);
101    }
102 }
103
104 public static void add_data() throws IOException {
105     emp[k] = new Personal();
106     System.out.printf("請輸入員工編號 =>");
107     emp[k].ID = keyin.nextInt();
108     keyin.nextLine();           //清除鍵盤緩衝器
109
110     System.out.printf("請輸入姓名 =>");
111     emp[k].name = keyin.nextLine();
112     System.out.printf("請輸入服務部門 =>");
113     emp[k].depart = keyin.nextLine();
114     System.out.printf("請輸入性別 (男/女) =>");
115     emp[k].sex = keyin.nextLine();
116 }
117
118 }
```

```
119         System.out.printf("請輸入生日 (年/月/日) =>");
120             emp[k].birth = keyin.nextLine();
121         System.out.printf("請輸入底薪 =>");
122             emp[k].payment = keyin.nextInt();
123         System.out.printf("請輸入職務加級 =>");
124             emp[k].duty = keyin.nextInt();
125         k = k+1;
126     }
127     public static void modify_data() throws IOException {
128         System.out.printf("請輸入欲修改員工的代號 =>");
129         int num = keyin.nextInt();
130         keyin.nextLine();           //清除鍵盤緩衝器
131         int flag = 0;
132         int i = 0;
133         while(i < k) {
134             if (emp[i].ID == num) {
135                 flag = 1;
136                 break;
137             }
138             i = i + 1;
139         }
140         if (flag == 0) {
141             System.out.printf("test here 5\n");
142             System.out.printf("沒有 %d 資料, 拒絕處理 !!\n", num);
143             return;
144         }
145         System.out.printf("請輸入 %s 姓名 =>", emp[i].name);
146             emp[i].name = keyin.nextLine();
147         System.out.printf("請輸入 %s 服務部門 =>", emp[i].depart);
148             emp[i].depart = keyin.nextLine();
149         System.out.printf("請輸入性別 %s (男/女) =>", emp[i].sex);
150             emp[i].sex = keyin.nextLine();
151         System.out.printf("請輸入生日 %s (年/月/日) =>", emp[i].birth);
152             emp[i].birth = keyin.nextLine();
153         System.out.printf("請輸入 %d 底薪 =>", emp[i].payment);
154             emp[i].payment = keyin.nextInt();
155         System.out.printf("請輸入 %d 職務加級 =>", emp[i].duty);
156             emp[i].duty = keyin.nextInt();
157     }
158     public static void delete_data() throws IOException {
```

```
165         System.out.printf("請輸入欲刪除員工的代號 =>");
166         int num = keyin.nextInt();
167         int flag = 0, m = 0;
168         while(m < k) {
169             if (emp[m].ID == num){
170                 flag = 1;
171                 break;
172             }
173             m = m + 1;
174         }
175         if (flag == 0){
176             System.out.printf("沒有 %d 資料，拒絕處理 !!\n", num);
177             return;
178         }
179         for(int i=m; i<k; i++)
180             emp[i]=emp[i+1];
181         k = k-1;
182         System.out.printf("%d 資料已刪除 !!\n", num);
183     }
184 }
185 public static void save_data() throws IOException {
    String file = "Human.data";
    BufferedWriter outData = new BufferedWriter(new FileWriter(file));
    System.out.printf("** 將儲存檔案 (Human.dada) **\n");
    for (int i=0; i<k; i++) {
        outData.write(emp[i].ID + "\t");
        outData.write(emp[i].name + "\t");
        outData.write(emp[i].depart + "\t");
        outData.write(emp[i].sex + "\t");
        outData.write(emp[i].birth + "\t");
        outData.write(emp[i].payment + "\t");
        outData.write(emp[i].duty + "\n");
    }
    outData.close();
    System.out.printf("***** 儲存完畢 *****\n");
}
}
```

(D) 程式重點分析：

- (1) 第 6~8 行：『static ...』。將物件陣列 emp[]、紀錄筆數變數 k 與鍵盤輸入物鍵 keyin 宣告成靜態變數，該類別內所有方法（或稱函數）都可以直接存取。
- (2) 第 13~41 行：系統被啟動後，立即將 Human.data 檔案所儲存的員工資料讀入 emp[]

陣列內，以後才可以針對 emp[] 陣列作資料處理。如果 Human.data 檔案不存在的話，系統會直接離開不處理，因此第一次執行之前，必須先建立一個名為 Human.data 空白檔案（可利用記事本產生）。

- (3) 第 31 行：『k = k+1;』。累計目前已讀取第幾筆資料；由此可見，變數 k 非常重要，是物件陣列 emp[] 的指標器。
- (4) 第 94~112 行：『public static void add_data() throws IOException {..}』。增加員工資料函數，執行該函數時須由鍵盤輸入資料，而且鍵盤輸入是利用 BufferedReader 物件，他必須指定發生異常狀態時，需將訊息擲向何處（IOException）。
- (5) 第 155~158 行：『if (flag == 0) { .. return; }』。如果變數 flag 為 0 的話，則該函數直接返回，就不會再執行往下的敘述句。
- (6) 第 159~160 行：『for(int i=m; i<k; i++) emp[i]=emp[i+1];』。欲刪除第 m 筆資料的動作，是將後面的資料往前移一個位置，該筆資料自然被覆蓋掉。
- (7) 第 164~180 行：『public static void save_data() ...』。將物件陣列 emp[] 內容寫入檔案，如果沒有執行寫入便離開系統，則所處理的動作將不會影響到檔案內容。

6-5-2 自我挑戰：超商商品管理系統

(A) 程式功能：PM6_3.java、Element.java

請幫『春嬌生鮮超市』建立一套商品登錄系統。該系統功能有『顯示所有資料』、『增加商品資料』、以及『儲存資料』。庫存資料儲存於 storage.data 檔案內。假設描述商品屬性（Element.java）為：商品編號（String no）、商品名稱（String name）、單價（int price）、單位（String unit）、庫存量（int stock）與製造商（String maker）。

至於庫存資料請以 storage.data 檔案名稱儲存。第一次執行沒有 storage.data 檔案，則利用編輯工具(如：notepad++) 開啟一個新檔，直接另存新檔並將檔名取“storage.data”(雙引號包起來，不要輸入任何內容)，再儲存於本程式的同一目錄下即可。

期望操作介面與結果如下：

- (1) 系統啟動時，會由 storage.data 檔案讀入目前資料內容，如果該檔案不在，則會要求先建立後再執行，如下：

```
D:\Java2_book\chap6\PM6_3>java PM6_3
storage.data 檔案不存在, 請先建立它
按 <Enter> 鍵離開 =>
```

- (2) 進入系統後，包含 5 個工作選項，如下：

```
D:\Java2_book\chap6\PM6_3>java PM6_3

== 春嬌超市 商品管理系統 ==

(1) 顯示所有資料          (2) 增加商品資料          (3) 修改商品資料
(4) 儲存資料              (5) 離開系統

請選擇工作項目 =>
```

- (3) 進入系統後，包含 5 個工作選項，如下：

```
D:\Java2_book\chap6\PM6_3>java PM6_3

== 春嬌超市 商品管理系統 ==

(1) 顯示所有資料          (2) 增加商品資料          (3) 修改商品資料
(4) 儲存資料              (5) 離開系統

請選擇工作項目 =>
```

- (4) 工作選項(1) 為顯示儲存的產品資訊，如下：

```
請選擇工作項目 => 1
== 列印所有商品資料 ==
代號    品名          單價    單位    庫存量    製造商
A1001   黑松汽水      20      個      210       黑松公司
A1002   可口奶滋      50      個      310       統一食品
```

- (5) 工作選項(2) 是增加商品資料的操作，如下：

```
請選擇工作項目 => 2
請輸入商品編號 =>1005
```

```

請輸入商品名稱 =>台灣啤酒
請輸入單價 =>60
請輸入單位 (個/箱/公斤) =>個
請輸入安全庫存量 =>50
請輸入製造商 =>台灣菸酒公司
    
```

(6) 工作選項(3)修改商品資訊的操作，如下：

```

請選擇工作項目 => 3
請輸入欲修改的商品編碼 =>A1005
[台灣啤酒]請輸入商品名稱 =>金門高粱
[60]請輸入單價 =>500
[個]請輸入單位 (個/箱/公斤) =>個
[50]請輸入庫存量 =>30
[台灣菸酒公司]請輸入製造商 =>金門酒廠
    
```

(7) 工作選項(4) 是儲存所有資料到檔案，如下：

```

請選擇工作項目 => 4
** 將儲存檔案 (storage.data) **
***** 儲存完畢 *****
    
```

(B) 製作技巧提示：

為了方便製作，吾人可將此系統劃分為若干個子系統 (方法或函數) 來分別實現，如

圖 6-8 所示；虛擬碼提示如下：

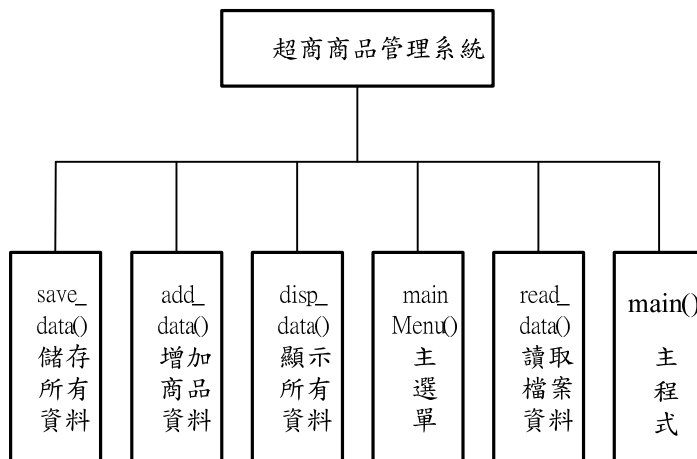


圖 6-13 超商商品管理系統架構

```
宣告商品類別 ( class Element { ... } );
宣告主類別範圍 ( class PM9_2 ) {
    宣告商品物件陣列 ( static Element[] art );
    宣告紀錄資料筆數的變數 ( static int number );
    宣告庫存檔案的變數 ( static String file );
    宣告鍵盤輸入物件 ( static BufferedReader keyin );
    宣告主方法範圍 ( main() ) {
        產生物件陣列範圍 ( art = new Element[50] );
        設定類別變數內容 ( number=0、file="storage.data" );
        呼叫執行讀取檔案資料方法 ( read_data() );
        呼叫主功能選單 ( mainMenu() ) 並讀取功能選項 ( select );
        while (select != 4) {
            switch(select) {
                case 1: disp_data() /* 顯示所有資料 */
                case 2: add_data() /* 增加商品資料 */
                case 3: modify_data() /* 修改商品資料 */
                case 4: save_data() /* 儲存資料 */
            }
            呼叫主功能選單 ( mainMenu() ) 並讀取功能選項 ( select );
        }
    } // 主方法結束
    /* 主工作項目選單 */
    宣告主功能表選單方法 ( mainMenu() { ....} );
    宣告檔案讀取方法 ( read_data() );
        /* 將 stroage.data 檔案讀入並存入 art[] 陣列 */
    宣告顯示所有資料方法 ( disp_data() );
    宣告增加商品資料方法 ( add_data() );
    宣告儲存商品資料方法 ( save_data() );
```

```

        /* 將 art[] 陣列內容輸出至 storage.data 檔案 */
    } // 主類別結束

```

(C) 程式片段提示

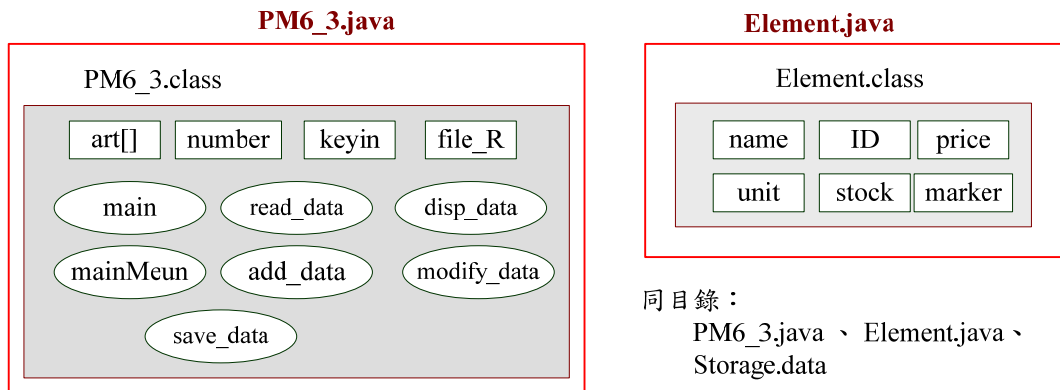


圖 6-14 PM6 3 程式架構

- **Element 類別的原始檔案：(Element.java)**

```

01 //Element.java
02
03 class Element {
04     String ID;           // 商品編號
05
06     String name;        // 商品名稱
07
08     int price;          // 單價
09
10     String unit;        // 單位
11
12     int stock;          // 庫存數量
13
14     String maker;       // 製造商
15 }

```

- **主類別的原始檔案片段：(PM6_3.java)**

```

01 //PM6_3.java
02 /* 請建立一套商品登錄系統，功能有:顯示所有資料、增加商品資料、儲存資料；
03  * 資料儲存於 storage.data 檔案內 */
04
05
06 /* Element.class 須於同目錄下 */
07 import java.io.*;

```

```
08 import java.util.Scanner;
09 public class PM6_3 {
10     static Element[] art;           // 商品資料的物件陣列
11
12     static int number;              // 紀錄儲存筆數
13
14     static Scanner keyin;          // 鍵盤輸入物件
15
16     static String file_R;          // 庫存檔案
17
18     public static void main(String args[]) throws IOException {
19         keyin = new Scanner(System.in);
20
21         art = new Element[50];
22         number=0;                   // 儲存資料的筆數
23         file_R = "storage.data";
24         read_data();
25
26         mainMenu();
27         int select = keyin.nextInt();
28         keyin.nextLine();
29         while (select !=5) {
30             switch (select) {
31                 case 1:             /* 顯示所有資料 */
32                     disp_data();
33                     break;
34                 case 2:             /* 增加商品資料 */
35                     add_data();
36                     break;
37                 case 3:             /* 修改商品資料 */
38                     modify_data();
39                     break;
40                 case 4:             /* 儲存資料 */
41                     save_data();
42                     break;
43                 default:
44                     System.out.printf("錯誤輸入，請重新選擇 !!\n");
45             }
46             mainMenu();
47             select = keyin.nextInt();
48             keyin.nextLine();
49         }
50     }
51 }
52
53 /* 主工作項目選單 */
```

```
54     public static void mainMenu() {
55         System.out.printf("\n== 春嬌超市 商品管理系統 ==\n");
56         ....
57         ....
58     }
59
60
61
62     /* 將 stroage.data 檔案輸入至系統 */
63     public static void read_data() throws IOException {
64         File fileID = new File(file_R);    // 產生輸入檔案物件
65         String inData;
66         if (fileID.exists()) {
67             BufferedReader data = new BufferedReader(new
68                                                         FileReader(fileID));
69             while ((inData=data.readLine()) != null) {
70                 Scanner s = new Scanner(inData).useDelimiter("\t");
71                 art[number] = new Element();
72                 art[number].ID = s.next();
73                 art[number].name = s.next();
74                 art[number].price = s.nextInt();
75                 art[number].unit = s.next();
76                 art[number].stock = s.nextInt();
77                 art[number].maker = s.next();
78                 number = number + 1;
79             }
80             data.close();
81         }
82         else {
83             System.out.printf("%s 檔案不存在，請先建立它 \n", file_R);
84             System.out.printf("按 <Enter> 鍵離開 =>");
85             keyin.nextLine();
86             System.exit(1);
87         }
88     }
89
90
91
92     /* 顯示所有資料 */
93     public static void disp_data() {
94         System.out.printf("== 列印所有商品資料 ==\n");
95         System.out.printf("代號\t品名\t單價\t單位\t安全庫存量\t製造商\n");
96         for (int i=0; i<number; i++) {
97             System.out.printf("%s\t", art[i].ID);
98             System.out.printf("%4s\t", art[i].name);
```

```
100         System.out.printf("%d\t", art[i].price);
101         System.out.printf("%s\t", art[i].unit);
102         System.out.printf("%d\t\t", art[i].stock);
103         System.out.printf("%s\n", art[i].maker);
104     }
105 }
106
107 /* 增加商品資料 */
108 public static void add_data(){
109     int k=number;
110     art[k] = new Element();
111     System.out.printf("請輸入商品編號 =>");
112     art[k].ID = keyin.nextLine();
113     System.out.printf("請輸入商品名稱 =>");
114     art[k].name = keyin.nextLine();
115     System.out.printf("請輸入單價 =>");
116     art[k].price = keyin.nextInt();
117     keyin.nextLine();
118     System.out.printf("請輸入單位 (個/箱/公斤) =>");
119     art[k].unit = keyin.nextLine();
120     System.out.printf("請輸入庫存量 =>");
121     art[k].stock = keyin.nextInt();
122     keyin.nextLine();
123     System.out.printf("請輸入製造商 =>");
124     art[k].maker = keyin.nextLine();
125     number = number+1;
126 }
127
128 /* 修改商品資料 */
129 public static void modify_data(){
130     System.out.printf("請輸入欲修改的商品編碼 =>");
131     String num = keyin.nextLine();
132     int flag = 0, test;
133     int i = 0;
134     while(i < number) {
135         test = art[i].ID.compareTo(num);
136         if (test == 0) {
137             flag = 1;
138             break;
139         }
140         i = i + 1;
141     }
142 }
```

```
146         if (flag == 0) {
147             System.out.printf("沒有 %s 資料, 拒絕處理 !!\n", num);
148             return;
149         }
150
151         System.out.printf("[%s]請輸入商品名稱 =>", art[i].name);
152         art[i].name = keyin.nextLine();
153         System.out.printf("[%d]請輸入單價 =>", art[i].price);
154         art[i].price = keyin.nextInt();
155         keyin.nextLine();
156         System.out.printf("[%s]請輸入單位 (個/箱/公斤) =>", art[i].unit);
157         art[i].unit = keyin.nextLine();
158         System.out.printf("[%d]請輸入庫存量 =>", art[i].stock);
159         art[i].stock = keyin.nextInt();
160         keyin.nextLine();
161         System.out.printf("[%s]請輸入製造商 =>", art[i].maker);
162         art[i].maker = keyin.nextLine();
163     }
164
165     /* 儲存商品資料至 storage.data */
166
167     public static void save_data() throws IOException {
168         BufferedWriter outData = new BufferedWriter(new FileWriter(file_R));
169         System.out.printf("*** 將儲存檔案 (storage.data) **\n");
170         for (int i=0; i<number; i++) {
171             outData.write(art[i].ID + "\t");
172             outData.write(art[i].name + "\t");
173             outData.write(art[i].price + "\t");
174             outData.write(art[i].unit + "\t");
175             outData.write(art[i].stock + "\t");
176             outData.write(art[i].maker + "\n");
177         }
178         outData.close();
179         System.out.printf("***** 儲存完畢 *****\n");
180     }
181 }
```

6-5-3 自我挑戰：倉儲管理系統

(A) 程式功能：PM6_4

(擴充 PM6_3.java 功能) 請幫『春嬌超商連鎖店』建立一套『倉儲管理系統』, 可供隨

時登錄各連鎖店進出貨量。該系統功能有『顯示所有資料』、『增加商品資料』、『出貨登錄』、『進貨登錄』、以及『儲存資料』。庫存資料儲存於 storage.data 檔案內。當點選『出貨』或『進貨』功能時，系統會將目前所建立的商品資料顯示於螢幕上，管理者點選產品再輸入數量即可。假設描述商品屬性(Element.class)為：商品編號(String no)、商品名稱(String name)、單價 (int price)、單位 (Sting unit)、庫存量 (int stock) 與製造商 (String maker)。

- (1) 除了 PM6_3.java 的 5 個功能外，還要擴充『進貨資料登錄』與『出貨資料登錄』兩個工作選項，如下：

```
D:\Java2_book\chap6\PM6_4>javac PM6_4.java
D:\Java2_book\chap6\PM6_4>java PM6_4

== 春嬌超市 商品管理系統 ==
(1) 顯示所有資料          (2) 增加商品資料          (3) 修改商品資料
(4) 進貨資料登錄          (5) 出貨資料登錄          (6) 儲存資料
(7) 離開系統

      請選擇工作項目 =>
```

- (2) 當選擇『進貨資料登錄』後，系統會出現目前庫存狀況，再選擇進貨那一種商品，並輸入進貨數量。輸入後再選擇『顯示所有資料』(選擇 4 與 1) 的結果如下：

```
      請選擇工作項目 => 4
==== 請點選 (進出貨) 商品 ====
(1) 黑松汽水 260 (2) 可口奶滋 410 (3) 可口可樂 500 (4) 台灣啤酒 200
(5) 滿漢大餐 200
請輸入商品選項 =>4
進貨數量 =>50

= 春嬌超市 商品管理系統 =
1) 顯示所有資料          (2) 增加商品資料          (3) 修改商品資料
4) 進貨資料登錄          (5) 出貨資料登錄          (6) 儲存資料
7) 離開系統

      請選擇工作項目 => 1
```

```
= 列印所有商品資料 ==
```

代號	品名	單價	單位	安全庫存量	製造商
1001	黑松汽水	20	個	260	黑松公司
1002	可口奶滋	50	個	410	統一食品
1003	可口可樂	15	個	500	英商太古
1004	台灣啤酒	60	個	250	台灣菸酒
1005	滿漢大餐	50	個	200	統一食品

- (3) 當選擇『出貨資料登錄』後，系統同樣會出現目前庫存狀況，再選擇出貨那一種商品與數量。輸入後再選擇『顯示所有資料』（選擇 5 與 1）的結果如下：

```
請選擇工作項目 => 5
=== 請點選 (進出貨) 商品 ===
(1) 黑松汽水 260 (2) 可口奶滋 410 (3) 可口可樂 500 (4) 台灣啤酒 250
(5) 滿漢大餐 200
請輸入商品選項 =>3
出貨數量 =>200

== 春嬌超市 商品管理系統 ==
(1) 顯示所有資料 (2) 增加商品資料 (3) 修改商品資料
(4) 進貨資料登錄 (5) 出貨資料登錄 (6) 儲存資料
(7) 離開系統

請選擇工作項目 => 1
== 列印所有商品資料 ==
```

代號	品名	單價	單位	安全庫存量	製造商
A1001	黑松汽水	20	個	260	黑松公司
A1002	可口奶滋	50	個	410	統一食品
A1003	可口可樂	15	個	300	英商太古
A1004	台灣啤酒	60	個	250	台灣菸酒
A1005	滿漢大餐	50	個	200	統一食品

- (4) 離開系統之前，記得『儲存資料』（選項 6），如下：


```

    請選擇工作項目 => 6

    ** 將儲存檔案 (storage.data) **

    ***** 儲存完畢 *****
    
```

(B) 製作技巧提示：

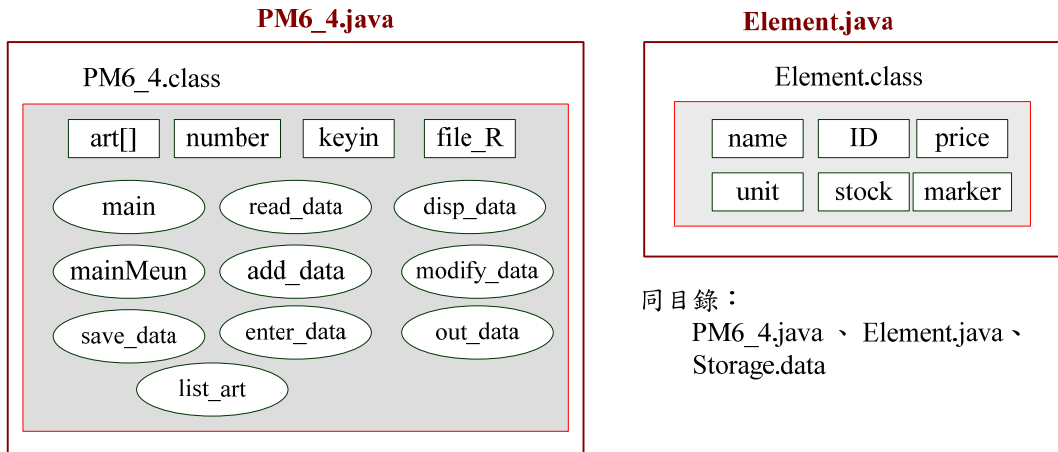


圖 6-15 PM6 4 程式架構

本系統大致上與 PM6_3.java 相似，僅增加了『進貨登錄』與『出貨登錄』功能選項。當執行這兩項功能時，系統必須將目前已登錄產品與數量，顯示於螢幕上；操作者點選後，再輸入進出貨數量。因此，必須另製作一只顯示產品名稱與數量，並可選擇輸入的子程式 (list_art())，兩功能子程式 (enter_date() 與 out_data()) 分別呼叫它，再傳回正確的產品選項。這三個子程式提示如下：

```

/* 進貨登錄子程式 */
public static void enter_data(){
    int num = list_art();
    if (num == 999) {
        return;
    }
    System.out.printf("進貨數量 =>");
    int enter = keyin.nextInt();
    keyin.nextLine();
    art[num].stock = art[num].stock + enter;
}
/* 出貨登錄子程式 */
public static void out_data(){
    int num = list_art();
    
```

```
        if (num == 999) {
            return;
        }
        System.out.printf("出貨數量 =>");
        int outer = keyin.nextInt();
        keyin.nextLine();
        art[num].stock = art[num].stock - outer;
    }

    /* 顯示商品清單及選擇商品子程式 */
    public static int list_art() {
        System.out.printf("==== 請點選 (進出貨) 商品 ==\n");
        for (int i=0; i<number; i++) {
            System.out.printf("(%d) %s %d  ", i+1, art[i].name, art[i].stock);
            if ((i+1) % 4 == 0)
                System.out.printf("\n");
        }
        System.out.printf("\n 請輸入商品選項 =>");
        int num = keyin.nextInt();
        keyin.nextLine();
        num = num - 1;
        if ((num >= number) || (num < 0)) {
            System.out.printf("錯誤輸入!! 按任何鍵回主選單\n");
            keyin.readLine();
            return 999;
        }
        return num;
    }
}
```