

第三章 二維陣列

3-1 二維陣列

3-1-1 二維陣列結構

既然陣列是由若干個相同資料型態的變數整合而成，變數的排列格式也延伸了不同的陣列格式。如果所有變數排列成一行或一列，則稱之為『一維陣列』(如 `array[]`)；如果排列成**平面形狀**，則稱為『二維陣列』(如 `array[][]`)；如果是立體形狀，則為『三維陣列』(如 `array[][][]`)；如是四度空間型態，則是『四維陣列』(如 `array[][][][]`)，其中每一維都需要一個變數作為指標，索引其相對位置。各種陣列型態都有其運用範圍，如沒有特殊情況的話，還是一維與二維陣列的應用最普遍，本書僅介紹到二維陣列，更多維數陣列的處理方式，也大致相同的。

將變數排列成『縱橫』的平面形狀，則稱之為『二維陣列』。二維陣列需要兩個位置指標，如 `score[x][y]` 變數，前面指標(如 `x`)是標示第幾行數(由 0 開始)；第二個指標(如 `y`)，則標示第幾列數；如是 $4 * 3$ 陣列共計有 12 個相同型態(`int`)變數的元素。另外，陣列被宣告完成之後(如 `int score[][] = { .. }`)。以 `score[][]` 陣列為例，java 利用 `score.length` 變數來表該陣列的行數。又 Java 允許陣列中各行的列數目不一定要相同，以 `score[x].length` 表示該行(`x`)元素的數目(即是列數目)。陣列內各元素與行列變數的表示方法，如圖 3-1 所示。

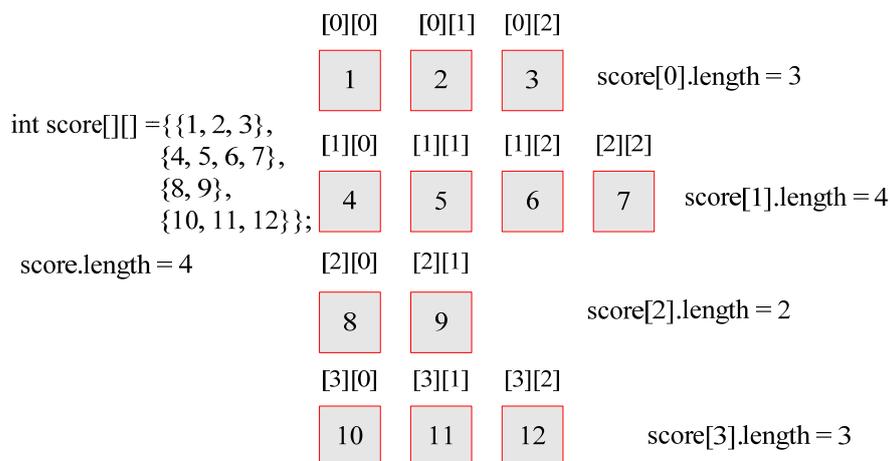


圖 3-1 二維陣列的結構

3-1-2 二維陣列宣告

陣列的宣告方法有兩種，一者先宣告陣列變數名稱與結構，當需要使用時再指定所需空間，

其稱之為動態宣告。另一者是宣告時就限定空間並給予出直，稱之為靜態宣告。語法分別如下：

	二維陣列宣告語法	範 例
動態 宣告	Data_type Array_name[][]; Array_name = new Data_type[num1][num2]; Data_type[][] name = new Data_type[num1][num2];	int score[]; score = new int[20][10]; int[][] score = new int [20][10];
靜態 宣告	Data_type Array_name[] []= { ...};	int score[][] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}, {10, 11, 12}};

第一行語法是宣告產生二維陣列，第二行則再給予初值。如敘述句如：int [] [] score = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}, {10, 11, 12}};則產生各個元素與變數內容如下：

```
score[0][0] = 1, score[0][1]=2, score[0][2]=3 =>score[0].length=3
score[1][0] = 4, score[1][1]=5, score[1][2]=6, score[1][3]=7 =>score[1].length=4
score[2][0] = 8, score[2][1]=9 =>score[2].length=2
score[3][0] = 10, score[3][1]=11, score[3][2]=12 =>score[3].length=3
score.length = 4
```

3-1-3 範例研討：印出二維陣列內容

(A) 程式功能：Ex3_1.java

請編寫一程式，印出陣列 score [][] = {{1, 2, 3}, {4, 5, 6},{7, 8, 9}, {10, 11, 12}} 內容並依照相對位置排列（平面形狀）。期望操作介面如下：

```
D:\Java2_book\chap3>javac Ex3_1.java

D:\Java2_book\chap3>java Ex3_1
    1    2    3
    4    5    6
    7    8    8
   10   11   12
```

(B) 製作技巧研討：

吾人利用 x 與 y 變數作為陣列位置索引，x 是行數指標，y 為列數指標，則任何一元素可由 score[x][y] 表示。如欲平面型態印出陣列內容，則需要二重迴圈敘述，外迴圈標示共計列印幾行 (x = 0, 1, ..., score.length); 內迴圈索引每行的列數 (y = 0, 1, 2, ..., score[x].length)。

(C) 程式範例：

```

01 //Ex3_1.java
02
03 public class Ex3_1 {
04     public static void main(String args[]) {
05         int score[][]={{1, 2, 3},
06                       {4, 5, 6},
07                       {7, 8, 8},
08                       {10, 11, 12}};
09         int x;                // 行指標 (外迴圈)
10
11         int y;                // 列指標 (內迴圈)
12         for(x=0; x<score.length; x++) {
13             for(y=0; y<score[x].length; y++)
14                 System.out.printf("\t%2d", score[x][y]);
15             System.out.printf("\n");
16         }
17     }
}

```

(D) 程式重點說明：

- (1) 第 5~8 行：『**int score[][] = { ...};**』。宣告二維陣列並給予初值。
- (2) 第 11~15 行：『**for(x=0; x<score.length; x++) {...}**』。外迴圈指定列印行數，其中 `score.length` 表示該陣列的行數。
- (3) 第 12~13 行：『**for(y=0; y<score[x].length; y++){...}**』。內迴圈指定每行的列的數目，其中 `score[x].length` 表示該行 (x) 列的數目。
- (4) 第 13 行：『**System.out.printf("\t%2d", score[x][y]);**』。列印 `score[x][y]` 元素的內容，列印格式是開始先跳一個『Tab』(\t) 定格，再以兩個位置的格式列印整數 (%2d)。
- (5) 第 14 行：『**System.out.printf("\n");**』。每行列印完成之後，跳至下一行。

3-1-4 範例研討：將二維陣列內容加倍**(A) 程式功能：Ex3_2.java**

請編寫一程式，請修改 Ex3_1 範例，將陣列 `score [][] = {{1, 2, 3}, {4, 5, 6},{7, 8, 9}, {10, 11, 12}}` 內容加 3 倍，再依照相對位置排列 (平面形狀)。期望操作介面如下：

```
D:\Java2_book\chap3>java Ex3_2
原陣列內容：
    1      2      3
    4      5      6
    7      8      8
   10     11     12
加倍後陣列內容：
    2      4      6
    8     10     12
   14     16     16
   20     22     24
```

(B) 製作技巧提示：

我們建構一個比較容易擴充的程式，將 `score[][]` 宣告成類別變數，使其允許類別內所有方法使用，又將列印陣列的功能製作成一個獨立的 `disp_arr()` 方法，程式架構如圖 3-2 所示。

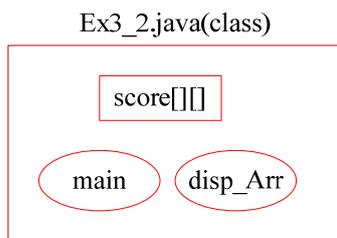


圖 3-2 Ex3-2 程式架構

(C) 程式範例

```

01 //Ex3_2.java
02 public class Ex3_2 {
03     /*宣告類別變數 score[] */
04     static int score[][]={{1, 2, 3},
05                             {4, 5, 6},
06                             {7, 8, 8},
07                             {10, 11, 12}};
08
09     /* 主程式 main() */
10     public static void main(String args[]) {
11         System.out.printf("原陣列內容：\n");
12         disp_Arr();
13         for(int x=0; x<score.length; x++) {           // 行指標 (外迴圈)
14             for(int y=0; y<score[x].length; y++)     // 列指標 (內迴圈)
15                 score[x][y] = score[x][y] * 2;
16             }
17         }
18     }

```

```

19         System.out.printf("加倍後陣列內容：\n");
20         disp_Arr();
21     }
22
23
24     /* 列印陣列方法 disp_arr() */
25     static void disp_Arr() {
26         for(int x=0; x<score.length; x++) {
27             for(int y=0; y<score[x].length; y++)
28                 System.out.printf("\t%2d", score[x][y]);
29             System.out.printf("\n");
30         }
31     }
32 }

```

3-1-5 自我挑戰：印出最高與最低成績者

(A) 程式功能：PM3_1.java

數學老師利用一個二維陣列儲存某一班級學生的成績，`score[][] = {{411101, 70}, {411102, 80}, {411103, 75}, {411104, 90}, {411105, 85}, {411106, 65}, {411107, 83}, {411108, 78}}`。請編寫一程式列印出該班成績最高與最低分數與姓名，期望操作介面如下：

```

D:\Java2_book\chap3>javac PM3_1.java

D:\Java2_book\chap3>java PM3_1
最高者 411104 成績= 90
最低者 411106 成績= 65

```

(B) 製作技巧提示：

二維陣列(`score[][]`)的每行(`score[i]`, $i = 0, 1, 2 \dots, 7$)紀錄一筆學生資料，第 0 列(`score[i][0]`)儲存學號、第 1 列(`score[i][1]`)儲存數學成績。吾人可利用兩隻一維陣列(`max[]` 與 `min[]`)儲存最高成績與最低成績，其中第 0 列(如 `max[0]`)儲存學號、第 1 列(如 `max[1]`)存放成績。圖 3-# 為搜尋運作程序，起先將最高成績的陣列設定為 0 (`max = {0, 0}`)；最低成績設定成超過成績界線 (`min = {0, 999}`)，再依序比較每位學生成績，如果高於 `max`，則該紀錄複製到 `max` 陣列內；如果低於 `min`，則複製到 `min` 上。如此由第 0 筆到最後一筆搜尋之後，`max` 與 `min` 陣列分別得到最高與最低成績的紀錄。虛擬碼提示如下：

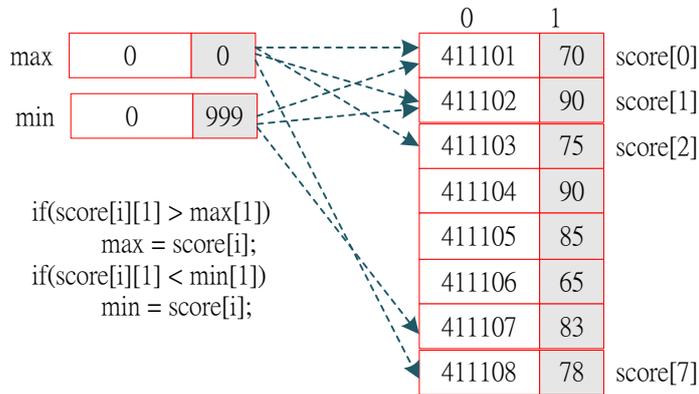


圖 3-# 最高與最低成績搜尋

```

01 宣告成績儲存陣列並給初值 ( int score[][]= {{411101, 70}, ...} );
02
03 宣告最高成績陣列 ( int max[] = {0, 0} );
04 宣告最低成績陣列 ( int min[] = {0, 9999} );
05
06 搜尋最高與最低成績者：
07     for(int i=0; i<score.length; i++) {
08         if(score[i][1] > max[1])
09             max = score[i];
10         if(score[i][1] < min[1])
11             min = score[i];
12     }
    
```

3-1-6 自我挑戰：成績查詢系統

(A) 程式功能：PM3_2.java

數學老師利用一個二維陣列儲存某一班級學生的成績，score[][] = {{411101, 70}, {411102, 80}, {411103, 75}, {411104, 90}, {411105, 85}, {411106, 65}, {411107, 83}, {411108, 78}}。請編寫程式可允許輸入學生學號，並輸出該學生的數學成績，期望操作介面如下：

```

D:\Java2_book\chap3>java PM3_2
== 成績查詢系統 ==
請輸入學生學號 =>4111120
沒有學號 = 4111120 的資料

D:\Java2_book\chap3>java PM3_2
== 成績查詢系統 ==
請輸入學生學號 =>411103
學號 = 411103, 成績= 75
    
```

(B) 製作技巧提示：

利用輸入學號作為查詢關鍵，由陣列的起頭到結尾依序比較第 0 個欄位 (score[i][0], i=0, 1, 2, ..., score.length-1)，如果相同則印出第 1 個欄位的內容 (score[i][1]，成績欄位)；虛擬碼提示如下：

```

01 宣告輸入物件 ( Scanner );
02
03 宣告成績儲存陣列並給初值 ( int score[][]= {{411101, 70}, ...} );
04
05 宣告旗號並給予初值 ( int flag=0 );
06 讀入查詢學號 ( num );
07
08 搜尋查詢成績：
09     for(int i=0; i<score.length; i++) {
10         if(score[i][0] == num){
11             輸出顯示學號與成績；
12
13             flag = 1;
           }
       }
       if(flag ==0)
           輸出顯示沒有資料；

```

3-2 陣列資料儲存

3-2-1 陣列資料型態

資料大多是由多筆紀錄所構成，每筆資料再利用若干個欄位來描述其意思 (第 8 章將詳細說明)。我們也可以利用二維陣列來儲存資料，陣列中每一行代表一筆紀錄，每行裡包含若干個欄位 (列)，存放某一特殊訊息，這就是陣列資料結構的架構。陣列資料結構最大的缺點是有元素都必須是相同的資料型態；當然還有其他表示法，以後再詳細介紹。我們用一般學生成績檔案為範例，來說明陣列資料結構的處理過程，如圖 3-9 所示。每一行儲存一筆學生資料，每行有 6 個欄位 (6 列)，分別存放學生學號、程式設計、資訊網路、資料處理、統計分析、以及總平均分數。

學 號	程式設計	資訊網路	資料處理	統計分析	總平均
492101	80	90	78	72	0
492102	90	67	54	46	0
492103	71	64	83	53	0

492104	68	89	73	73	0
492105	56	72	53	49	0
492106	96	65	78	83	0
492107	49	62	91	68	0

圖 3-3 陣列資料結構範例

3-2-2 範例研討：印出班級成績單

(A) 程式功能：Ex3_3.java

張老師利用二維陣列 `score[][] = {{492101, 80, 90, 78, 72, 0}, {492102, 90, 67, 54, 46, 0}, {492103, 71, 64, 83, 53, 0}, {492104, 68, 89, 73, 73, 0}, {492105, 56, 72, 53, 49, 0}, {492106, 96, 65, 78, 83, 0}, {492107, 49, 62, 91, 68, 0}}` 存放該班的學生成績資料，每筆資料為 {學號、程式設計、資訊網路、資料處理、統計分析、總平均}。請計算每一位同學的總平均分數，再列印出所有成績資料。期望操作介面如下：

```
D:\Java2_book\chap3>java Ex3_3
學號 程式設計 資訊網路 資料處理 統計分析 總平均
492101 80 90 78 72 80
492102 90 67 54 46 64
492103 71 64 83 53 67
492104 68 89 73 73 75
492105 56 72 53 49 57
492106 96 65 78 83 80
492107 49 62 91 68 67
```

(B) 製作技巧研討：

首先計算每位學生的總平均分數，亦填入該欄位 (`socre[k][5]`，第 `k` 位學生)，最後在印出所有內容即可。

(C) 程式範例：

```
01 //Ex3_5.java
02
03 public class Ex3_5 {
04     public static void main(String args[]) {
05
06         int score[][] = {{492101, 80, 90, 78, 72, 0},
07                         {492102, 90, 67, 54, 46, 0},
08                         {492103, 71, 64, 83, 53, 0},
```

```

09         {492104, 68, 89, 73, 73, 0},
10         {492105, 56, 72, 53, 49, 0},
11         {492106, 96, 65, 78, 83, 0},
12         {492107, 49, 62, 91, 68, 0}};
13
14     /* 計算每位學生的總平均分數 */
15     for(int i=0; i<score.length; i++)
16         score[i][5] = (score[i][1]+score[i][2]+score[i][3]+score[i][4])/4;
17
18     /* 列印所有學生成績資料 */
19
20     System.out.printf("學號 程式設計 資訊網路 資料處理 統計分析 總平均
21 \n");
22     for (int i=0; i<score.length; i++) {
23         for(int j=0; j<score[i].length; j++)
24             System.out.printf("%d \t", score[i][j]);
25             System.out.printf("\n");
26     }
    }
}

```

(D) 程式重點分析：

- (1) 第 15~16 行：『for(int i=0; i<score.length; i++){ ...}』。計算每筆資料的總平均分數。
- (2) 第 20~24 行：『for (int i=0; i<score.length; i++) { ...}』。列印出陣列所有內容。

3-2-3 自我挑戰：印出已排序成績單

(A) 程式功能：PM3_3.java

延續 Ex3_3 範例，請編寫一程式列印出依照總平均分數，由高至低順序的成績單，期望操作介面如下：

```

G:\Examples\chap7>java PM3_8
學號 程式設計 資訊網路 資料處理 統計分析 總平均
492101 80      90      78      72      80
492106 96      65      78      83      80
492104 68      89      73      73      75
492103 71      64      83      53      67
492107 49      62      91      68      67
492102 90      67      54      46      64
492105 56      72      53      49      57

```

(B) 製作技巧提示：

以總平均為基準的排序是比較總平均分數，如果大於的話，則整筆資料都要交換；因此，暫存器需要宣告成具有 6 個欄位的一維陣列 (`int[] temp = new int[6]`，每筆資料有 6 個欄位)。泡沫排序法提示如下：

```

01  /* 泡沫排序法，依照總平均分數由高至低排列 */
02
03  int temp[] = new int[6];
04  for(int i=0; i<score.length; i++) {
05      for(int j=i+1; j<score.length; j++) {
06          if(score[i][5] < score[j][5]) {
07              temp = score[i];
08              score[i] = score[j];
09              score[j] = temp;
10          }
11      }
    }

```

3-3 專題研討**3-3-1 自我挑戰：列印直式唐詩****(A) 程式功能：PM3_4.java**

請製作一套唐詩列印系統，功能是能將所將輸入的唐詩(五言四句)，分別以橫式與直式印出。期望操作介面如下：

```

G:\Examples\chap7>java PM3_4
== 五言四句唐詩 列印 ==
請輸入第 1 句(五個字)=>山中相送罷
請輸入第 2 句(五個字)=>日暮掩柴扉
請輸入第 3 句(五個字)=>春草明年綠
請輸入第 4 句(五個字)=>王孫歸不歸

== 橫式列印唐詩 ==
山 中 相 送 罷

```

```

日 幕 掩 柒 扉
春 草 明 年 綠
王 孫 歸 不 歸

== 直式列印唐詩 ==

山 日 春 王
中 幕 草 孫
相 掩 明 歸
送 柒 年 不
罷 扉 綠 歸
    
```

(B) 製作技巧提示：

本系統需要二維陣列 (poem[][])，將詩句裡每一個文字儲存於陣列的元素裡，才可分別以橫式或直式印出。其中包含兩困難點，一者是如何將一串文字輸入，分別取出每一個文字，再存入陣列元素內；另一者是如何轉換陣列的行與列印出 (直式列印)。前者，吾人將一串文字直接讀入系統並存放某字串變數 (data) 內 (利用 Scanner 物件)，再將該變數宣告成掃描物件 (Scanner 物件) 並指定分隔符號為沒有空格 (useDelimiter(""))，接著即可掃描取出每一個文字，再分別存入陣列內，如圖 3-4 所示。

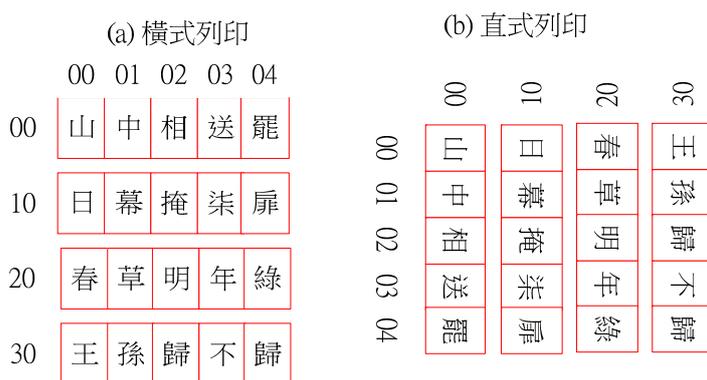


圖 3-4 poem[][] 陣列儲存內容

唐詩內所有文字都依序存入陣列後。原先是依照一行接一行填入文字，如果一行接一行印出，則如同輸入格式一樣是橫式輸出；如果將行與列倒過來印，則是直式輸出。虛擬碼提示如下：

```

01  導入相關套件 ( java.util.Scanner );
02
    
```

```

03 宣告唐詩陣列 ( String[][] poem = new String[4][5] );
04
05 讀入五言四句唐詩：
06     for(int i=0; i<4; i++) {
07         System.out.printf("請輸入第 %d 句(五個字) =>", i+1);
08         data = keyin.next();
09         Scanner s = new Scanner(data).useDelimiter("");
10         for(int j=0; j<5; j++)      // 分別讀取詩句中每一個文字
11             poem[i][j] = s.next();
12     }
13
14 列印橫式唐詩：
15
16     for(int i=0; i<4; i++) {          // 列印 4 行
17         for(int j=0; j<5; j++)      // 每行 5 個字
18             System.out.printf("%s ", poem[i][j]);
19         System.out.printf("\n");
20     }
21
22 列印直式唐詩：
23
24     for(int j=0; j<5; j++) {          // 列印 5 行
25         for(int i=0; i<4; i++)      // 每行 4 個字
26             System.out.printf("%s ", poem[i][j]);
27         System.out.printf("\n");
28     }

```

3-3-2 自我挑戰：大樂透對獎系統

(A) 程式功能：PM3_5.java

許多客戶買了大樂透彩券，一直不瞭解開獎辦法，再說彩券購買太多的話，每次對獎都要花費許多時間。彩券公司為了方便客戶對獎，於是在網路上公布一套對獎系統，客戶可輸入當期開獎號碼，或觀察當期號碼，也可輸入所購買號碼，系統會告知是否得獎，或簽中哪一獎，大樂透開獎辦法是：利用氣球吹出 6 個 1~49 之間不重複的號碼，另外多一個特別號；中獎種類有：

中獎獎項	開獎辦法
頭獎	六個號碼相同者。
貳獎	簽中五個號碼與特別號。
參獎	簽中五個號碼。
肆獎	簽中四個號碼與特別號。

伍獎	簽中四個號碼。
陸獎	簽中三個號碼與特別號。
普獎	簽中三個號碼。

(1) 期望操作介面上有 4 個選項功能如下：

```
D:\Java2_book\chap3>java PM3_5

== 歡迎光臨 六合彩對獎系統 ==
(1) 輸入開獎號碼 (2) 顯示開獎號碼
(3) 輸入對獎號碼 (4) 離開系統

    請選擇工作項目 =>
```

(2) 選擇輸入開獎號碼(選擇 1)操作如下：

```
== 歡迎光臨 六合彩對獎系統 ==
(1) 輸入開獎號碼 (2) 顯示開獎號碼
(3) 輸入對獎號碼 (4) 離開系統

    請選擇工作項目 =>1

請輸入 6 個開獎號碼 =>11 22 27 29 35 40
請輸入特別號碼 =>39
```

(3) 選擇顯示開獎號碼(選擇 2)的操作如下：

```
== 歡迎光臨 六合彩對獎系統 ==
(1) 輸入開獎號碼 (2) 顯示開獎號碼
(3) 輸入對獎號碼 (4) 離開系統

    請選擇工作項目 =>2

本期開獎號碼: 11 22 27 29 35 40
特別號: 39
```

(4) 選擇輸入對獎號碼(選擇 3)的操作如下：

```
== 歡迎光臨 六合彩對獎系統 ==
(1) 輸入開獎號碼 (2) 顯示開獎號碼
```

(3) 輸入對獎號碼 (4) 離開系統

請選擇工作項目 =>3

請輸入 6 個對獎號碼 =>11 15 18 22 27 40

4 個對中號碼:11 22 27 40

恭喜您, 伍獎

(B) 製作技巧提示：

吾人將開獎號碼(`spaNum[]`)、特別號(`special`)、與中獎號碼(`winNum[]`)的陣列、特別號(`wining`)、以及中獎數目(`winSpa`)宣告成靜態變數，再利用 4 個函數 (方法) 來製作整個系統。

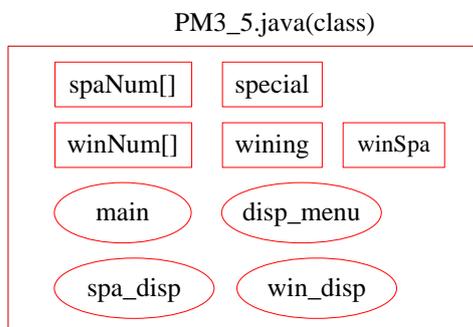


圖 3-3 PM3 5 程式架構

虛擬碼提示如下：

```

01 宣告開獎號碼、特別號變數 ( static int spaNum[], special; );
02
03 宣告中獎號碼、特別號與數目 ( static int winNum[], wining, winSpa; );
04 主方法範圍 ( main() ): {
05
06     宣告對獎號碼陣列 ( int[] num = new int[6]; );
07
08     顯示及讀入工作項目 ( select );
09     while (select !=4) {
10         switch(select) {
11             case 1:
12                 讀入六個開獎號碼 ( spaNum[0], .. spaNum[5] );
13
14                 讀入特別號 ( special );
15             case 2:
16                 顯示開獎號碼 ( spa_disp() );
17             case 3:
18                 讀入六個對獎號碼 ( num[] )
19
20                 /* 對獎程式 */
  
```

```
21         wining = 0; winSpa=0; int k=0;
22         for(int i=0; i<6; i++) {           // 6 個對獎號碼
23             for(int j=0; j<6; j++) {      // 6 個開獎號碼
24                 if(num[i] == spaNum[j]) {
25                     winNum[k] = num[i]; // 存入中獎陣列
26                     wining = wining+1; // 累增中獎數
27                     k = k+1;
28                 }
29             }
30         }
31     }
32     if(num[i] == special)                // 1 個特別號
33         winSpa = num[i];
34 }
35 /* 比對對獎數目
36 switch(wining) {
37     case 0:
38         顯示全沒簽中 ;
39     case 1:
40         呼叫顯示中獎號碼 ( win_disp() )
41         顯示沒有簽中 ;
42     case 2:
43         呼叫顯示中獎號碼 ( win_disp() )
44         顯示沒有簽中 ;
45     case 3:
46         呼叫顯示中獎號碼 ( win_disp() )
47         if (winSpa == 0)
48             顯示簽中普獎 ;
49         else {
50             顯示簽中陸獎 ;
51         }
52     case 4:
53         呼叫顯示中獎號碼 ( win_disp() )
54         if (winSpa == 0)
55             顯示簽中伍獎 ;
56         else {
57             顯示簽中肆獎 ;
58         }
59     case 5:
60         呼叫顯示中獎號碼 ( win_disp() )
61         if (winSpa == 0)
62             顯示簽中參獎 ;
63     }
```

```
else
    顯示中貳獎 ;

case 6
    呼叫顯示中獎號碼 ( win_disp() );
    顯示簽中頭講 ;
}
default:
    顯示錯誤輸入, 請重新選擇 ;
}
顯示及讀入工作項目 ( select );
}
}
宣告顯示工作項目方法 ( static void disp_menu() );
宣告顯示開獎號碼及特別號方法 ( static void spa_disp() );
宣告顯示中獎號碼及特別號方法 ( static void win_disp() );
```

(C) 重點說明：

- (1) 行號 29 ~ 59：判斷是否中獎的 `switch(){...}` 範圍。
- (2) 行號 7 ~ 64：主選單 `switch() {...}` 範圍。