

# 第三章 算術運算式

## 3-1 Java 運算式

### 3-1-1 Java 運算子彙集

一只程式係由多個『敘述句』( **Statement** ) 構成，『敘述句』也許是電腦的單一命令動作，也許是由多個敘述句結合而成的『敘述區塊』。各種程式語言都會制訂一些基本敘述句，描述期望電腦處理哪些動作。基本上，處理程序可區分為若干個類型，此即為敘述句的型態。較常見的型態有『運算式』( 本章介紹 )、『判斷式』( 第四章介紹 )、『迴圈控制式』( 第五章介紹 ) 等等。

**表 3-1 運算子彙集**

運算子	描 述	運算子	描 述
指定運算子		位元邏輯運算子	
=	指定變數內容	^	位元 XOR
算術運算子		&	位元 AND
+, -	一元 ( unary ) 正負符號		位元 OR
+, -, *, /, %	加、減、乘、除、餘數	條件邏輯組合運算子	
++, --	遞增、遞減	&&	邏輯 AND
判斷運算子			邏輯 OR
==	是否相等	移位運算子	
!=	是否不相等	<<	右移運算子
<	是否小於	>>	左移運算子
<=	是否小於或等於		
>	是否大於		
>=	是否大於或等於		
!	是否邏輯否定		

### 3-1-2 一元與二元運算式

『運算式』是程式的最基本敘述句，無論『判斷式』或『迴圈控制式』都需要它來達成。運

算式的功能是将一個或兩個變數，經過某只『運算子』( Operate ) 處理之後，得到一個運算結果；參與運算的數值或變數，則稱為『運算元』( Operant )。如果運算式內僅有一個運算元，則稱為『一元運算式』；否則稱為『二元運算式』，格式如下所示：

a + b	兩個運算元 a 與 b ( 變數 )，經由『+』運算子處理
count - 5	變數 count 與 5 兩運算元，經由『-』運算子處理
flag == True	變數 flag 與 True 兩運算元，經由『==』運算子處理
count >= 10	變數 count 與整數 10，經由『>=』運算子處理
- data	單一變數 data，經由『-』運算子處理

### 3-1-3 運算子種類

表 3-1 為 Java 語言運算子彙集表，無論運算式、判斷式或迴圈控制式都需要利用這些運算子來達成；每一種運算子代表電腦可能處理的某一種動作，分類說明如下：

- (1) **指定運算子**：指定變數內容，語法為『變數 = 數值』；或稱『將某一數值填入變數內』。譬如將 3 填入變數 x 內，則語法為 x = 3。
- (2) **算數運算子**：兩個或一個變數可經由算數運算子處理後，得到另一個結果。譬如，兩變數相乘 ( x \* y )，則表示分別取出 x 與 y 的內容，兩者再相乘得其結果。
- (3) **判斷運算子**：兩個或一個變數經由判斷運算子處理後，得到一個如同布林變數的邏輯判斷『真』( true · 1 ) 或『否』( false · 0 ) 的結果。譬如 x 是否大於 y ( x > y )，如果成立的話，運算結果是『真』( 1 )，否則為『否』( 0 )。另一範例，判斷 x 是否沒有大於 y，敘述句是『! ( x > y )』，得到與前敘述句相反的結果。而『!』則表示邏輯否定的意思。
- (4) **條件組合邏輯運算子**：如果敘述句內需要多個條件判斷的話，則可利用 && ( AND ) 與 || ( OR ) 兩運算子達成。譬如，敘述句需要 x > y 與 y > z 兩條件是否都成立，則利用 && 則合成為『(x>y) && (y>z)』；另一方面，如任一條件成立即可，則為『(x > y) || ((y > z)』。
- (5) **位元邏輯運算子**：兩變數內容做位元處理 ( AND、OR、XOR )，得其結果。
- (6) **位元移位運算子**：變數內容以位元為單位，左移 ( >> ) 或右移 ( << ) 多少位元。

本章僅介紹指定運算子與算術運算子所構成的運算式，其他運算子將會在爾後章節介紹。

## 3-2 指定運算子

變數除了宣告時，給予某一特定初值外，程式運作當中也可能隨時改變其內容，以下將介紹變更變數內容方法。

### 3-2-1 等於與變數指定

程式最容易讓人搞混莫過於，變數與『等於』( = ) 運算子之間的關係。程式語言的等於 ( = ) 運算子，絕對不是『相當於』( equal ) 的意思，而是『指定』某一變數的內容，稱之為『指定運算子』( Assignment operator )。簡單的說，即是將某一數值『填入』變數內。

基本上，宣告產生某一變數後，其內容不可能是個『空值』，存在某一不可預測的數值 ( 由 0 與 1 構成的亂數 )。將數值填入變數後，他將會覆蓋原來的內容；當變數內容被指定後，如再重新指定，也會覆蓋原來的內容，由指定運算子所構成之運算式的格式如下：

var1 = 10;	將數值 10 填入變數 var1 內，並覆蓋原來內容。
var2 = 20;	將數值 20 填入變數 var2 內，並覆蓋原來內容。
var3 = var1 + var2;	將變數 var1 與 var2 內容取出，相加後將結果填入 var3 內，但 var1 與 var2 的內容不會改變。
var3 = var3 + 20;	將 var3 的內容取出，再加 20 後將結果存回 var3；原來 var3 的內容便被覆蓋掉。

指定運算式的功能是，將指定式 ( = ) 右邊的敘述結果，填入左邊的變數內；右邊可以是一個數值或運算式，但左邊必須是單一變數。譬如  $x = 10 + x$ ，則表示取出變數  $x$  內容，再與 10 執行『+』的運算後，再存回  $x$  變數內。如果編寫成  $x + 10 = x$ ，則是錯誤的敘述。

### 3-2-2 範例研討：收銀機計算

#### ( A ) 程式功能：Ex3\_1.java

請製作一套模擬超商收銀機系統，假設客戶僅購買汽水、餅乾、與御便當，程式要求輸入各項金額，並隨時顯示累計金額，最後計算總金額數量。期望系統操作模式如下：

超商收銀機、請注意是否歸零 = 0

```
請輸入汽水的金額 =>20
    目前累進金額為 = 20
請輸入餅乾的金額 =>10
    目前累進金額為 = 30
請輸入御便當的金額 =>50
    總金額為 = 80 謝謝光臨 !!!
```

## ( B ) 製作技巧分析：

由操作介面可以看出，系統是輸入某項購買金額後，立即計算出目前累進金額，也很合乎一般計算器的操作模式。吾人可設定一個變數作為客戶購買金額的累加器 ( `int total;` )，系統顯示要求輸入各項產品金額，操作人員輸入後 ( `item = keyin.nextFloat();` )，則累計到累加器內 ( `total = total + item` )。

## ( C ) 程式範例：Ex3\_1.java

```
01 // Ex3_1.java
02
03 import java.util.Scanner;
04 public class Ex3_1 {
05     public static void main(String args[]) {
06         Scanner in = new Scanner(System.in);
07         int total=0, item;           // 設 total 累積器初值為 0
08
09         System.out.printf("超商收銀機、請注意是否歸零 = %d\n", total);
10
11         System.out.printf("請輸入汽水的金額 =>");
12         item = in.nextInt();
13         total = total + item;
14         System.out.printf("\t目前累進金額為 = %d\n", total);
15
16         System.out.print("請輸入餅乾的金額 =>");
17         item = in.nextInt();
18         total = total + item;
19         System.out.printf("\t目前累進金額為 = %d\n", total);
20
21         System.out.print("請輸入御便當的金額 =>");
22         item = in.nextInt();
23         total = total + item;
24         System.out.printf("\t總金額為 = %d 謝謝光臨 !!!\n", total);
25     }
26 }
```

## (D) 程式重點說明：

- 第 3 行：『import java.util.Scanner;』。導入 Scanner 掃描輸入套件。
- 第 6 行：『Scanner in = new Scanner(System.in);』。利用 Scanner 類別產生一個標準輸入 (System.in, 鍵盤) 物件，名稱為 in。
- 第 9 行：『item = in.nextFloat();』。利用 Scanner 物件 (in) 由鍵盤讀入一個浮點數 (物件方法為 in.nextFloat())，並將其存入 item 變數內 (指定運算子的功能)。
- 第 10 行：『total = total + item;』。將所讀取 item 數值累積存入 total 變數內，為目前累積購買金額。
- 第 11 行：『System.out.printf(“\t 目前累進金額為 = %.2f\n\n”, total);』。此為格式化輸出方法 (printf())，功能是：跳一個 tab 空白鍵 (\t)，再印出『目前累進金額 =>』，接著以浮點數格式印出 total 內容，而且取兩位小數點 (%.2f)，最後執行兩次跳行 (\n\n) 動作。

## 3-3 算術運算子

### 3-3-1 算術運算子彙集

基本算術運算子包含一般數學的加法 (+)、減法 (-)、乘法 (\*) 與除法 (/)，在計算機運算裡增加一個除法餘數 (%，或稱『模數』運算子)，而其資料型態可以是 byte、short、int、long、float、double 與 char，其中 char 型態於運算時會轉成 ASCII 碼，表 3-1 是一些基本算術符號，可能會出現正、負號與加、減符號相同，運算式會自動去判斷。

**表 3-2 基本算術運算子**

運算符號	說明
+、-	正、負號
+、-、*、/、%	加、減、乘、除、模數 (求餘數)
++、--	遞增、遞減
(...)	運算式集合

算術運算子大多必須與『指定運算子』(=) 配合使用，常用運算式型態彙集如下：

$x = x + y;$	取出 $x$ 與 $y$ 兩變數內容，相『加』後再填入變數 $x$ 內。
$x = -x + y;$	變數 $x$ 內容取負數，與變數 $y$ 相『加』後再填入變數 $x$ 內。
$x = x * y;$	取出 $x$ 與 $y$ 兩變數內容，相『乘』後再填入變數 $x$ 內。
$z = x / y;$	變數 $x$ 除以 $y$ ，所得到的商存入變數 $z$ 內。
$z = x \% y;$	變數 $x$ 除以 $y$ ，所得到的『餘數』存入變數 $z$ 內；如 $x = 5$ 、 $y = 3$ ，則 $x \% y$ 得到餘數為 2。
$x++;$	相當於 $x = x + 1;$
$x--$	相當於 $x = x - 1;$
$x += y;$	相當於 $x = x + y;$
$x -= y;$	相當於 $x = x - y;$
$x *= y;$	相當於 $x = x * y;$
$x /= y;$	相當於 $x = x / y;$
$x \% = y;$	相當於 $x = x \% y;$
$(2 + 3) * (4 + 5)$	先處理括號內得到 $5 * 9$ ，最後結果為 45。

### 3-3-2 範例探討：計算股票平均價格

#### (A) 程式功能：Ex3\_2.java

當分析師選定某一支股票作為標的物後，則需紀錄該股票每天的股價多寡，並計算 5 日（一般都需紀錄 5、10、20、30 日）股價平均價格如何。請您幫他製作一套登錄及計算系統，可連續輸入 5 個交易日的收盤價，並隨時顯示當日的平均價格為何，最後輸出 5 日平均股價。期望系統運作模式如下：

```

*** 計算股票平均價系統 ***
請輸入第一個交易日股價 =>40.5
(目前平均價 = 40.50)請輸入第二個交易日股價 =>48.6
(目前平均價 = 44.55)請輸入第三個交易日股價 =>42.7
(目前平均價 = 43.93)請輸入第四個交易日股價 =>49.2
(目前平均價 = 45.25)請輸入第五個交易日股價 =>50.2
五日平均價 = 46.24

```

## (B) 製作技巧研討：

股票 5 日平均價算法是當天之前 5 個交易日價錢的平均，本系統執行之前並沒有資料，只好每天輸入股價後，隨時計算出平均價，到第 5 天則可得到 5 日平均價；因此，每天輸入價格後，計算出從第 1 天到該日的總額及平均價

## (C) 程式範例：

```
01 // Ex3_2.java
02 import java.util.Scanner;
03 public class Ex3_2 {
04     public static void main(String args[]) {
05         Scanner keyin = new Scanner(System.in);
06         float ave, sum=0, cost;
07         int number=0;
08         System.out.printf("*** 計算股票平均價系統 ***\n");
09
10         System.out.printf("請輸入第一個交易日股價 =>");
11
12         cost = keyin.nextFloat();
13         sum = sum + cost;
14         number++;
15         ave = sum / number;
16
17         System.out.printf("(目前平均價 = %.2f)請輸入第二個交易日股價 =>", ave);
18
19         cost = keyin.nextFloat();
20         sum = sum + cost;
21         number++;
22         ave = sum / number;
23
24         System.out.printf("(目前平均價 = %.2f)請輸入第三個交易日股價 =>", ave);
25
26         cost = keyin.nextFloat();
27         sum = sum + cost;
28         number++;
29         ave = sum / number;
30
31         System.out.printf("(目前平均價 = %.2f)請輸入第四個交易日股價 =>", ave);
32
33         cost = keyin.nextFloat();
34         sum = sum + cost;
35         number++;
36         ave = sum / number;
37
38         System.out.printf("(目前平均價 = %.2f)請輸入第五個交易日股價 =>", ave);
39
40         cost = keyin.nextFloat();
41         sum = sum + cost;
```

```
38         number++;
39         ave = sum / number;
40         System.out.printf("五日平均價 = %.2f\n", ave);
41     }
42 }
43
44
45
```

### (D) 程式重點分析：

- 吾人可以發現計算 5 日平均價還稍可接受，如計算 10、20、30 個交易日，程式將變得非常長，而大多書寫同樣的敘述句。因此，本系統應該使用迴圈敘述句，可容易許多，本書第五章將會介紹到。
- 第 7 行：『float ave, sum=0F, cost;』。表示同時宣告 3 個浮點變數，並給 sum 初值 0，但浮點數的數值必須多加 F (sum=0F)。
- 第 11 行：『sum = sum + cost;』。將 cost 內容累積增加到 sum 變數內。
- 第 12 行：『number++;』。功能是 number 變數累增 1 (number = number + 1)。
- 第 13 行：『ave = sum / number;』。計算當天之前連續交易日的平均股價。

## 3-3-3 自我挑戰：學期成績計算程式

### (A) 程式功能：PM3\_1.java

資管系二年級這學期開了電腦概論 (2 學分)、程式設計 (3 學分)、離散數學 (3 學分)、國文 (2 學分)、英文 (2 學分)，共計 5 門課 12 學分。請建立一套系統，讓學生自行輸入各科成績，能計算並印出該同學學期總平均多寡。期望使用操作介面如下：

```
*** 學期成績計算系統(各科學分數) ***
請輸入電腦概論成績(2 學分) =>80
請輸入程式設計成績(3 學分) =>87
請輸入離散數學成績(3 學分) =>72
請輸入國文成績(2 學分) =>65
請輸入英文成績(2 學分) =>74
```

```
學期總平均分數 = 76.25
```

```
四捨五入後成績 = 76
```

### ( B ) 製作技巧提示：

吾人可利用一個累積變數 ( sum )，每筆資料讀入，再與它的學分數相乘後，累計到累積變數內；最後再乘以總學分數，即可得到總平均分數，程式重點提示如下：

```

01 .....
02     System.out.printf("請輸入程式設計成績(3 學分) =>");
03     value = keyin.nextInt();
04     sum = sum + value * 3;
05 .....
06 .....
07 .....
08     ave = sum/12.0F;
09     System.out.printf("學期總平均分數 = %.2f\n", ave);
10     ave = ave + 0.5F;
11     System.out.printf("四捨五入後成績 = %d\n", (int)ave);
12 .....
13 .....

```

題目要求平均分數必須包含小數點第二位，平均分數的變數需採用浮點數 ( float )，因為整數相除只能得到整數，所以必須讓整數除以浮點數 ( 12.0F )，才會得到浮點數的結果。

### 3-3-4 範例研討：超商找錢工具

#### ( A ) 程式功能：Ex3\_3.java

請建立一個超商找錢工具，系統要求輸入購買總金額之後，再要求輸入已收金額 ( 大於購買金額 )，請輸出 100 元、50 元、10 元、5 元、1 元的零錢各需找多少。期望操作介面如下：

```

**** 超商找錢工具 ****
請輸入消費的金額 =>354
請輸入繳納金額 ( 大於消費額 354 ) =>500
應找金額 = 146 各種零錢數量如下:
    100 元零錢 = 1 張
    50 元硬幣 = 0 個
    10 元硬幣 = 4 個

```

5 元硬幣 = 1 個

1 元硬幣 = 1 個

### ( B ) 製作技巧研討：

本書到目前為止還未介紹到判斷大小的敘述句，因此假設繳納金額大於消費金額；繳納金額扣除消費金額，則為應找金額。吾人利用整數除以整數，僅得到整數結果的特性（譬如， $431 / 100 = 4$ ），來計算各項零錢的數目，再利用乘法計算剩餘的錢（ $431 - 100 * 4 = 31$ ）多寡（也可利用求餘數 % 運算子計算）。

### ( C ) 程式範例：

```
01 // Ex3_3.java
02 import java.util.Scanner;
03 public class Ex3_3 {
04     public static void main(String args[]){
05         Scanner keyin = new Scanner(System.in);
06         int total, receipt, value, value1, handre, fifty, ten, five, one;
07
08         System.out.printf("**** 超商找錢工具 ****\n");
09
10         System.out.print("請輸入消費的金額 =>");
11         total = keyin.nextInt();
12         System.out.printf("請輸入繳納金額 ( 大於消費額 %d ) =>", total);
13         receipt = keyin.nextInt();
14
15         value = receipt - total;
16         value1 = value;
17         handre = value / 100;
18         value = value - handre * 100;
19
20         fifty = value / 50;
21         value = value - fifty * 50;
22
23         ten = value / 10;
24         value = value - ten * 10;
25
26         five = value / 5;
27         value = value - five * 5;
28
29         one = value;
30         System.out.printf("應找金額 = %d 各種零錢數量如下:\n", value1);
31
32
```

```

33     System.out.printf("\t 100 元零錢 = %d 張 \n", hundre);
34
35     System.out.printf("\t 50 元硬幣 = %d 個 \n", fifty);
36     System.out.printf("\t 10 元硬幣 = %d 個 \n", ten);
37
        System.out.printf("\t 5 元硬幣 = %d 個 \n", five);
        System.out.printf("\t 1 元硬幣 = %d 個 \n", one);
    }
}

```

### (D) 程式重點分析：

- 第 17 行：『`hundre = value / 100;`』。計算 100 元零錢的數目；整數相除僅得到整數，剩下的餘數則被取捨掉。
- 第 18 行：『`value = value - hundre*100;`』。計算找 100 元零錢數目後，剩下的找錢數目多寡；其實是求餘數的功能，讀者可將其改為 `value = value % 100` 執行看看，結果是否相同。

## 3-3-5 自我挑戰：超商收銀機系統

### (A) 程式功能：PM3\_2.java

請製作一套超商收費系統，假設客戶僅購買衛生紙（每包 32 元）、口香糖（每包 12 元）、可樂（每瓶 18 元）、與熱狗（每支 17 元），系統會依序顯示各項產品單價，要求輸入購買數量，輸入完成後系統顯示總金額多寡。接著系統要求輸入客戶繳交金額，再計算出應該找多少零錢（100、50、10、5、1 元的數量）。期望系統操作介面如下：

```

***** 超商收銀機系統 *****
衛生紙(每包 32 元) 購買數量 =>3
口香糖(每包 12 元) 購買數量 =>5
可樂(每瓶 18 元) 購買數量 =>2
熱狗(每支 17 元) 購買數量 =>3
請輸入繳納金額(大於購買總金額 = 243) =>500
應找金額 = 257 各種零錢數量如下:
    100 元零錢 = 2 張
    50 元硬幣 = 1 個

```

```

10 元硬幣 = 0 個
5 元硬幣 = 1 個
1 元硬幣 = 2 個

```

## (B) 製作技巧提示：

此系統也需要累積計算來達成，吾人宣告一個累積變數 `total=0` ( 初始值為 0 )。接著輸入購買產品數量，則計算該項金額 ( 單價 \* 數量 ) 後累加到累積變數上 ( 如 `total = total + item * 32` )；最後計算出總購買金額，並要求輸入『繳納金額』。繳納金額扣除購買金額，剩下的即是應找金額，再參考範例 `Ex3_3.java`，計算出各個零錢數量多寡。重點提示如下

```

01 .....
02 ....
03         System.out.printf("***** 超商收銀機系統 *****\n");
04
05         System.out.printf("衛生紙(每包 32 元) 購買數量 =>");
06         item = in.nextInt();
07         total = total + item * 32;
08         .....
09
10         System.out.printf("請輸入繳納金額(大於購買總金額 = %d) =>", total);
11         item = in.nextInt();
12         value = item - total;
13         value1 = value;
14         .....

```

## 3-4 資料型態的轉換

### 3-4-1 資料型態變換時機

#### (A) 自動轉換

不同資料型態的變數之間，也可能經由運算子計算後，產生另一個數值表示，而其結果應該是哪一種資料型態。一般程式語言的規則如下：

- (1) 整數與整數運算 ( + 、 - 、 \* 、 / ) 後輸出為整數，如  $x = y + z$ ，如  $y$ 、 $z$  為整數，則輸出  $x$  為整數。
- (2) 字元或字串大多以 ASCII 碼 ( 整數型態 ) 表示，字元之間或字元與整數之間計算後，所

得結果也是 ASCII 碼 ( 整數 )。

### (3) 整數與浮點數計算後輸出為浮點數。

值得注意的是，不同變數之間經過運算後，大多必須結果存入內一個變數內；如果被存入變數的資料型態不符合的話，編譯程式時 ( Compiler, javac ) 會告知資料型態錯誤。

## (B) 強迫轉換

在許多情況下，我們為了計算正確需要轉換原來變數的資料型態，或是運算後轉換成比較合乎真實情況所需的資料型態。為了合乎這些需求，而必須將變數的資料型態轉換成其他資料型態，稱為『強迫轉換』；敘述格式如下：

( 新資料型態 ) 變數名稱;

上述功能為，無論原變數是任何資料型態，取出其內容並轉換成新的資料型態，範例如下：

<code>int a = (int) value;</code>	取出 value 內容，轉換成整數型態，再存入變數 a 內。 假設 value = 4.5，則 a = 4。
<code>float a = (float) value;</code>	取出 value 內容，轉換成浮點數型態，再存入 a 內。 假設 value = 5，則 a = 5.0。

在物件導向的程式語言上，物件類別與基本資料型態具有相同的屬性。我們可以任意轉換變數的資料型態；相同的，也可以任意轉換物件的類別型態。我們爾後再介紹此功能應用。

## 3-4-2 範例研討：兩整數相除結果

### (A) 程式功能：Ex3\_4.java

由鍵盤輸入兩個整數，以兩數相除的範例，驗證是否有轉換資料型態可能產生不同的結果。期望程式操作介面如下：

```
請輸入兩個整數(value1 value2) =>45 23
未轉換 => 45/23 = 1.000000
已轉換 => (float)45/23 = 1.956522
已轉換 => 45/(float)23 = 1.956522
```

## (B) 製作技巧分析：

兩整數相除僅能得到整數的結果，如果將其結果存入浮點數內，小數點部分也僅能補零 (如  $45/13 = 3.00$ )；如欲得到小數點部分，則必須將其中某一整數轉換成浮點數。

## (C) 程式範例：

```
01 // Ex3_4.java
02
03 import java.util.Scanner;
04 public class Ex3_4 {
05     public static void main(String args[]) {
06         Scanner keyin = new Scanner(System.in);
07         int div1, value1, value2;
08         float div;
09         System.out.printf("請輸入兩個整數(value1 value2) =>");
10         value1 = keyin.nextInt();
11         value2 = keyin.nextInt();
12
13         // 整數相除得到整數，存入浮點數變數
14         div = value1 /value2;
15         System.out.printf("未轉換 => %d/%d = %f\n", value1, value2, div);
16
17         // 一個整數轉換成浮點數，再除以另一個整數，結果存入浮點數變數
18         div = (float)value1 /value2;
19         System.out.printf("已轉換 => (float)%d/%d = %f\n", value1, value2, div);
20
21         // 強迫轉換另一個整數變數
22         div = value1 /(float)value2;
23         System.out.printf("已轉換 => %d/(float)%d = %f\n", value1, value2, div);
24
25     }
26 }
27 }
```

## (D) 程式重點分析：

- 第 10、11 行：『value1 = keyin.nextInt();』。利用 Scanner 函數讀入兩個整數，正常情況是使用者輸入兩個整數，再敲入『Enter』鍵，如果僅輸入一個整數，則系統會等待使用者再輸入一個整數。
- 第 18 行：『div = (float)value1/vaule2;』。將 value1 內容轉換成浮點數。

- 第 22 行：『div = value1/(float)value2;』。將 value2 內容轉換成浮點數。

### 3-4-3 自我挑戰：紀錄棒球打擊率

#### (A) 程式功能：PM3\_3.java

美國大聯盟某一球隊看上了熊隊的林智勝先生，派遣一位球探到台灣，觀察林智勝每次打擊，並紀錄他的打擊率。打擊率是累積方式計算出來的，每一球季開始到計算日當天，之前打擊次數除以安打數（打擊率 = 打擊次數 / 安打次數）；球探每次輸入林智勝打擊次數與安打數，則計算出當時的打擊率如何（假設可連續輸入 2 次比賽、之前打擊次數為 50、安打數為 15 支）。期望系統操作介面如下：

```
****  記錄棒球打擊率工具  ****
目前打擊=50數 安打=15支、打擊率是 0.3000
      請輸入第一場 (打擊次數 安打次數) =>3 1
目前出場=53數 安打=16支、打擊率是 0.3019
      請輸入第二場 (打擊次數 安打次數) =>4 0
目前出場=57數 安打=16支、打擊率是 0.2807
```

#### (B) 程式製作提示：

首先設定打擊累積次數 ( int total = 50 )、安打累積次數 ( int hits = 15 )、打擊率 ( float batting )、以及每場打擊次數 ( int numbers ) 與安打次數 ( int bingles )，並給予適當初值。接著計算與顯示出之前打擊率多寡，再輸入每場比賽結果，並計算出當時打擊率；程式重點提示如下：

```
01  .....
02  ....
03      System.out.printf("目前打擊=%d數 安打=%d支、打擊率是 %.4f\n", total, hits, batting);
04
05
06      System.out.printf("\t請輸入第一場 (打擊次數 安打次數) =>");
07      numbers = keyin.nextInt();
08      bingles = keyin.nextInt();
09      total = total + numbers;
10      hits = hits + bingles;
11      batting = (float)hits/total;
12      System.out.printf("目前出場=%d數 安打=%d支、打擊率是 %.4f\n", total, hits, batting);
13  .....
```

上述中『`batting = (float)hits/total;`』功能是強迫轉換變數型態。該敘述運作如下：原來 `hits` 為整數變數，取出他的內容再將他轉換成浮點數 (`(float)hits`)，再除以 `total`，得到的結果是浮點數型態，最後將其存入 `batting` 變數內。

### (C) 擴充紀錄『長打率』功能：

僅紀錄『打擊率』並無法表示出打擊手的功力如何，球探增加紀錄林智勝的『長打率』變化情形。計算方式：長打率 = 壘打數 / 打數次數，其中壘打與打擊次數都是累加計算的，一壘打則累加 1 分、二壘打則 2 分、三壘打則 3 分、全壘打為 4 分。假設目前記錄為打擊累積次數 (`total = 50`)、壘打數 (`base = 30`)，請擴充上述範例，增加可記錄『長打率』(目前 `ops = 30/50 = 0.6`) 的功能。

## 3-5常用的數學套件 - Math

### 3-5-1 內定數學套件

大多數程式語言的編譯器，大多會將較常用的數學函數製作成程式庫存函數，使用者可以直接引用，不用為這些函數的產生而苦惱。傳統語言則稱之為『數學庫存函數』；然而，在 Java 的編譯器則稱為『數學套件』。也就是說，Java 編譯器將一些較常用的程式函數，依照其屬性分別製作成多個程式套件。另外，依照檔案儲存方式的樹狀結構，來存放與引用套件。儲存架構以 `java` 為樹根，在它之下再分別建立各種不同屬性的套件，如 `java.lang`、`java.util`、`java.network...`等；接著再由某一套件下延伸出子套件，如 `java.util.Scanner...`等；依此類推，最後節點為子套件下的類別方法。

在 `java.lang` 套件中的一個 `Math` 共用類別，該目錄下 (`java.lang.Math`) 包含了許多有關數學函數的類別方法，表 3-1 中列出一些較常用的類別方法。又 `java.lang` 套件的類別方法，大多是較常用到的，因此 Java 編譯器會自動匯入，而不需宣告導入 (`import java.lang`)。

表 3-1 常用 Math 套件的類別方法

方法	說明
<code>abs(x)</code>	回傳 <code>x</code> 的絕對值
<code>exp(x)</code>	回傳 $e^x$

log(x)	回傳 log x
max(x, y)	回傳 x, y 的較大值
min(x, y)	回傳 x, y 的較小值
pow(x, y)	回傳 $x^y$
sqrt(x)	回傳 x 的平方根值

### 3-5-2 範例研討：找出最大/最小的數

#### (A) 程式功能：Ex3\_5.java

人的自然行為中，由一堆數字中找出最大或最小數值，是易如反掌的；但要電腦來處理可就不容易了。此範例功能是使用者任意輸入三個浮點數，電腦分別輸出其中最大與最小的數值為何。期望使用者操作介面如下：

```
*** 三個浮點數比較大小工具 ***
請輸入三個浮點數(val1 val2 val3) =>45.6 34.9 98.6
最大數 => 98.60
最小數 => 34.90
```

#### (B) 製作技巧分析：

由上述操作介面，表示需要連續輸入 3 個浮點數，再找出其中最大與最小數值。連續輸入資料則需利用 Scanner 套件才有此功能，因此程式開始需導入 java.util.Scanner，但僅寫 java.util.\* 也可（表示導入 util 以下所有類別套件）。3 個數值輸入後，吾人再利用數學套件的大小比較函數來製作。找出較大函數 max(x, y)，功能是回傳 x 與 y 兩者之間較大的數值；如果  $x = 5$ 、 $y = 8$ ，執行  $k = \max(x, y)$  之後，得到  $k = 8$  之結果；執行  $k = \min(x, y)$ ，得到  $k = 5$  結果。

#### (C) 程式範例：Ex3\_5.java

```
01 // Ex3_5.java
02
03 import java.util.*;
04 public class Ex3_5 {
05     public static void main(String args[]) {
06         Scanner keyin = new Scanner(System.in);
07         float value1, value2, value3;
08         float max, min;
09         System.out.printf("*** 三個浮點數比較大小工具 ***\n");
10
11         System.out.printf("請輸入三個浮點數(val1 val2 val3) =>");
12         value1 = keyin.nextFloat();
13         value2 = keyin.nextFloat();
14         value3 = keyin.nextFloat();
15
16         max = Math.max(value1, Math.max(value2, value3));
17         min = Math.min(value1, Math.min(value2, value3));
18
19         System.out.printf("最大數 => %.2f\n", max);
20
21         System.out.printf("最小數 => %.2f\n", min);
22     }
}
```

### (D) 程式重點分析：

- 第 11~13 行：『value1 = keyin.nextFloat();』。連續讀入 3 個浮點數，如果鍵盤沒有輸入的話（或不足 3 個），程式會等待輸入。
- 第 15 行：『max = Math.max(value1, Math.max(value2, value3));』。功能是先取出 value2 與 value3 兩者之間較大的數值，該數值再與 value1 比較，得到兩者之間較大值並存入 max 變數內。
- 第 16 行：『min = Math.min(value1, Math.min(value2, value3));』。如同第 15 行功能，但取出較小數值。
- 第 18 行：『System.out.printf("最大數 => %.2f\n", max);』。列印 max 變數內容的格式是具有 2 個小數點位數的浮點數（%.2f）。

### 3-5-3 自我挑戰：麻將骰子比大小

#### (A) PM3\_4：程式功能

麻將桌上常有擲骰子比大小來決定取牌順序，請編寫一程式可輸入東、西、南、北所擲骰子的大小，最後輸出最大骰子的數目多少，操作如下：

```
請輸入東方所擲數目 (3~18) =>12
請輸入西方所擲數目 (3~18) =>15
請輸入南方所擲數目(3~18) =>6
請輸入北方所擲數目(3~18) =>17
最大數目 => 17
```

## (B) 程式提示

```
....
Scanner keyin = new Scanner(System.in);
int value1, value2, value3, value4;
int max;

System.out.printf("請輸入東方所擲數目 (3~18) =>");
value1 = keyin.nextInt();
...
```

## 3-6 專題研討

### 3-6-1 自我挑戰：溫度轉換工具

#### (A) 系統功能：PM3\_5.java

我們需要一套可以轉換溫度攝氏與華式之間的工具，期望操作如下：

```
*** 攝氏/華式溫度轉換工具 ***
請輸入華式溫度 =>103.5
華氏 103.50 度= 攝氏 39.72 度
請輸入攝式溫度 =>32.1
攝氏 32.10 度= 華氏 89.78 度
```

#### (B) 製作技巧提示：

首先我們必須了解兩者溫度之間的轉換，華式溫度(F)轉換攝氏溫度(C)公式是  $(C = (F - 32) * 5 / 9)$

5/9，需包含小數點)；攝氏溫度轉(C)換成華氏溫度(F)的計算公式是  $(F = 32 + (9 * C)/5)$ 。程式片段如下：

```
01 ....
02     System.out.printf("*** 攝氏/華式溫度轉換工具 ***\n");
03
04     System.out.printf("請輸入華式溫度 =>");
05     value_F = keyin.nextFloat();
06     value_C = (value_F - 32) * 5.0F/9.0F;
07 ....
```

### 3-6-2 自我挑戰：手機計費工具

#### (A) 系統功能：PM3\_6.java

手機通訊費率大多採用秒數計算，請製作一套計算費率系統。系統要求輸入通話時間是『時 分 秒』方式（請引用 Scanner 套件），接著系統要求輸入每秒通話費（如 0.5 元），則系統需計算出總通話秒數，以及通話費多寡，期望操作介面如下：

```
*** 手機通話費計算工具 ***
請輸入通話時間(時 分 秒) =>2 30 50
通話時間總共 9050 秒
請輸入每秒計費(如 0.5) =>0.4
通話費總共 3620 元
```

#### (B) 製作技巧提示：

```
01 ....
02     times = (hour * 60 + minute) * 60 + second;
03     ....
04     total = units * times;
05     System.out.printf("通話費總共 %d 元\n", (int)total);
06     ....
```

### 3-6-3 範例研討：登革熱診斷系統(一)

#### (A) 程式功能：Ex3\_6.java

高雄地區的登革熱病蚊傳播嚴重，衛生局常要求醫生隨時通報案例發生，同時也鼓勵民眾初步自行診斷是否有被感染，因此期望在網站上公佈一個簡單的診斷工具，如有病情徵兆時，再到醫

院進一步檢查。診斷系統要求使用者填入各種徵兆項目，再依照這些參數的比重推論病情可能被感染比例。徵兆項目如下：( 假設症狀與事實不完全相符 )

- 發高燒 ( x , 比重 a=6 )。
- 筋骨酸痛 ( y , 比重 b=3 )。
- 感冒症狀 ( z , 比重 c=8 )。
- 貪睡症狀 ( k , 比重 d=3 )。

上述各種症狀的徵兆數值為 ( 0 ~ 10 ) : 不明顯 ( 0 )、稍明顯 ( 4 )、明顯 ( 6 )、很明顯 ( 10 )。感染登革熱指數 ( 0 ~ 10 ) =  $(x * 6 + y * 3 + z * 8 + k * 3) = (6 + 3 + 8 + 3)$ ，指數越高則表示被感染可能性越高，民眾自我診斷之後，再自行判斷是否到醫院確實診治。期望系統的操作介面如下：

```
*** 登革熱自我診斷系統 ***
請依照程式詢問步驟, 輸入各種症狀明顯度如何 (0 ~ 10)
    如: 不明顯(0),稍明顯 (4) 明顯(6),很明顯(10)
請輸入發燒症狀明顯否(0~10) =>5
請輸入筋骨酸痛症狀明顯否(0~10) =>8
請輸入感冒症狀明顯否(0~10) =>6
請輸入嗜睡症狀明顯否(0~10) =>7

您感染登革熱指數為(0 ~ 10)= 6.33
```

## ( B ) 製作技巧分析：

讀入每一種症狀的明顯度 ( 0 ~ 10 )，再乘以該症狀的加權比重；將所有症狀明顯度的總合，在除以總加權比重，即可得到感染登革熱的指數為何。

## ( C ) 程式範例：Ex3\_6.java

```
01 // Ex3_6.java
02
03 import java.util.*;
04 public class Ex3_6 {
05     public static void main(String args[]){
06         Scanner keyin = new Scanner(System.in);
07         final int a=8, b=6, c=4, d=3;
08         int weights = a+b+c+d;
09         int x, y, z, k;
10         float value;
11         System.out.printf("*** 登革熱自我診斷系統 ***\n");
12         System.out.printf("請輸入各種症狀明顯度如何 (0 ~ 10)\n");
13         System.out.printf("\t如: 不明顯(0),稍明顯 (4) 明顯(6),很明顯(10)\n");
14
15
16         System.out.printf("請輸入發燒症狀明顯否(0~10) =>");
17         x = keyin.nextInt();
18         System.out.print("請輸入筋骨酸痛症狀明顯否(0~10) =>");
19         y = keyin.nextInt();
20         System.out.print("請輸入感冒症狀明顯否(0~10) =>");
21         z = keyin.nextInt();
22         System.out.print("請輸入嗜睡症狀明顯否(0~10) =>");
23         k = keyin.nextInt();
24
25         value = (float)(x*a + y*b + z*c + k*d)/weights;
26         //value=Math.sqrt((a*x*x + b*y*y + c*z*z + d*k*k)/(double)(a+b+c+d));
27         System.out.printf("\n您感染登革熱指數為(0 ~ 10)= %.2f\n", value);
28     }
29 }
30
31
32 }
```

## ( D ) 程式重點分析：

- 第 8 行：『final int a=8, b=6, c=4, d=3;』。設定各項症狀的加權比重值，數值越大表示該症狀對是否感染登革可能性越高。
- 第 9 行：『int weights = a+b+c+d;』。所有加權比重的總合。
- 第 28 行：『value = (float)(x\*a + y\*b + z\*c + k\*d)/weights;』。利用加權比重計算出感染登革熱指數的多寡。

### 3-6-4 自我挑戰：登革熱診斷系統之二

#### (A) 程式功能：PM3\_7.java

許多疾病的症狀是大同小異的，當民眾到醫院時，醫生診斷病患是否感染登革熱，或是罹患其他疾病，就顯得非常困難。雖然選擇正確的症狀與加權比重，對於判斷的正確性越高，但加權比重的計算方式，也會影響到最後的判斷正確否；請您幫醫生建立一套更精密的計算方式，來提高判斷的正確性。有另一種均方根值的數理推論計算方式，說不定可以提高正確判斷，計算公式如下：

$$\sqrt{\frac{ax^2 + by^2 + cz^2 + dk^2}{a + b + c + d}}$$

圖 3-1

吾人期望的操作介面如下：

```

*** 登革熱自我診斷系統 ***
請依照程式詢問步驟，輸入各種症狀明顯度如何 (0 ~ 10)
    如：不明顯(0),稍明顯 (4) 明顯(6),很明顯(10)
請輸入發燒症狀明顯否(0~10) =>5
請輸入筋骨酸痛症狀明顯否(0~10) =>8
請輸入感冒症狀明顯否(0~10) =>6
請輸入嗜睡症狀明顯否(0~10) =>7

您感染登革熱指數為(0 ~ 10)= 6.45

```

#### (B) 程式製作提示：

輸出入介面大致上與 Ex3\_6.java 相同，我們只要變更感染指數的計算公式即可，程式提示如下：

```

value=Math.sqrt((a*x*x + b*y*y + c*z*z + d*k*k)/(double)(a+b+c+d));
System.out.printf("\n 您感染登革熱指數為(0 ~ 10)= %.2f\n", value);

```

### 3-6-5 自我挑戰：憂鬱症自我診斷工具

## ( A ) 程式功能：PM3\_8.java

依照醫學學術單位研究，在這緊張社會環境裡，民眾大多有憂鬱症的傾向；衛生局期望建立一套自行探討系統，能讓國人隨時瞭解自己憂鬱症傾向如何。在網路上公布一套對談選單方式，讓國人自行填入各項參數，再推論憂鬱症傾向比重如何。推論參數如下：各項皆 0~10 分。

- (1) 想哭 ( 比重 5 )
- (2) 情不好 ( 比重 4 )
- (3) 以前更容易發脾氣 ( 比重 5 )
- (4) 我睡不好 ( 比重 3 )
- (5) 我覺得不想吃東西 ( 比重 2 )
- (6) 我覺得想事情或做事情比以前慢 ( 比重 3 )
- (7) 我比較會往壞的方向想 ( 比重 4 )
- (8) 我覺得身體不舒服 ( 比重 2 )
- (9) 我覺得自己很沒用 ( 比重 3 )
- (10) 我很想不開、甚至想死 ( 比重 5 )

為了增加系統推論的正確性，提供有兩種演算法，依照上述症狀推論是否罹患憂鬱症，演算法如下：

- **總和計算**：每題 0~10 分，以總和分數判斷：正常 ( 0~30 )、輕微憂鬱症 ( 30~60 )、明顯憂鬱症 ( 60~100 ) ( 沒有使用比重加權 )。
- **加權比重計算**：每題 0~10，乘以加權比重後的總和，再除以加權的總和 ( 如同學校成績的學分計算方式 )。總平均 ( 0~10 ) 判斷：正常 ( 0~3 )、輕微憂鬱症 ( 3~6 )、明顯憂鬱症 ( 6~10 )。

期望操作介面如下：

```
***** 憂鬱症自我診斷系統 *****
請依序回答下列問題，相似度 (0~10)
不明顯(0)，稍明顯(2)，明顯(6)，很明顯(10)

我常覺得想哭 ( 比重 5 ) =>4
```

```

我覺得心情不好 ( 比重 4 ) =>3
我覺得比以前更容易發脾氣 ( 比重 5 ) =>9
我睡不好 ( 比重 3 ) =>8
我覺得不想吃東西 ( 比重 2 ) =>8
我覺得想事情或做事情比以前慢 ( 比重 3 ) =>7
我比較會往壞的方向想 ( 比重 4 ) =>9
我覺得身體不舒服 ( 比重 2 ) =>5
我覺得自己很沒用 ( 比重 3 ) =>7
我很想不開、甚至想死 ( 比重 5 ) =>5

總合計算 ( 0 ~ 100), 判斷結果: 正常 ( 0 ~ 30 )
    輕微憂鬱症 ( 30 ~ 60), 明顯憂鬱症 ( 60~100 )
您憂鬱症指數為 =>65

加權指數計算 ( 0 ~ 10), 判斷結果: 正常 ( 0 ~ 3 )
    輕微憂鬱症 ( 3 ~ 6), 明顯憂鬱症 ( 6~10 )
您憂鬱症指數為 =>6.39

```

## ( B ) 製作技巧提示 :

分別讀入各項問題的分數，各項分數的總和 ( sum1 )，即是總和計算的判斷法；如分別乘以加權比重，累加其總和 ( sum2 )，再除以加權比重的合計 ( weights )，即可加權指數計算方式。提示如下：

```

.....
System.out.printf("\n 總合計算 ( 0 ~ 100), 判斷結果: 正常 ( 0 ~ 30 ) \n");
System.out.printf("\t 輕微憂鬱症 ( 30 ~ 60), 明顯憂鬱症 ( 60~100 ) \n");
System.out.printf("您憂鬱症指數為 =>%d\n", sum1);

System.out.printf("\n 加權指數計算 ( 0 ~ 10), 判斷結果: 正常 ( 0 ~ 3 ) \n");
System.out.printf("\t 輕微憂鬱症 ( 3 ~ 6 ), 明顯憂鬱症 ( 6~10 ) \n" );
System.out.printf("您憂鬱症指數為 =>%.2f\n", (float)sum2/weights);

```