

第五章 重複性流程

5-1 重複性流程簡介

重複性流程表是在符合某一條件下，重複的執行特定『敘述區塊』(**Body**)，又稱之為『迴圈』。執行方法是，首先判斷條件是否成立，如成立則執行迴圈一次；否則跳過迴圈而不執行。迴圈與選擇敘述之間最大不同是，條件成立的話，選擇敘述僅執行一次；迴圈敘述則不然，執行『敘述區塊』之後，會再回來測試條件是否再成立，如成立則再執行迴圈一次，如此重複執行迴圈，一直到條件不成立才結束。如果條件一直都成立的話，則程式可能永遠無法離開迴圈敘述，一般程式語言都有下列三種迴圈：

- (1) **for 迴圈 (for loop)**。
- (2) **while 迴圈 (while loop)**。
- (3) **do/while 迴圈 (do/while loop)**。

雖然以上三種迴圈都可以達到一般重複敘述的功能，但之間還是有稍微不同的地方，並沒有孰好孰壞的問題，完全看程式設計師自己喜好決定。

5-2 for 迴圈控制

5-2-1 for 迴圈敘述

最典型的迴圈功能莫過於 **for 迴圈 (for loop)** 了，它明確表示迴圈可能出現的狀況。圖 5-1 為 for 迴圈的流程圖，首先設定初始條件，接著再測試條件是否成立，條件不成立 (no) 則立即結束該敘述；成立 (yes) 的話，則執行 for 敘述區塊。執行完畢後，則計算條件參數的『增減量』，接著再次測試條件是否成立 (yes/no)；如此重複執行 for 敘述區塊，一直到條件不成立 (no) 才結束該 for 敘述。

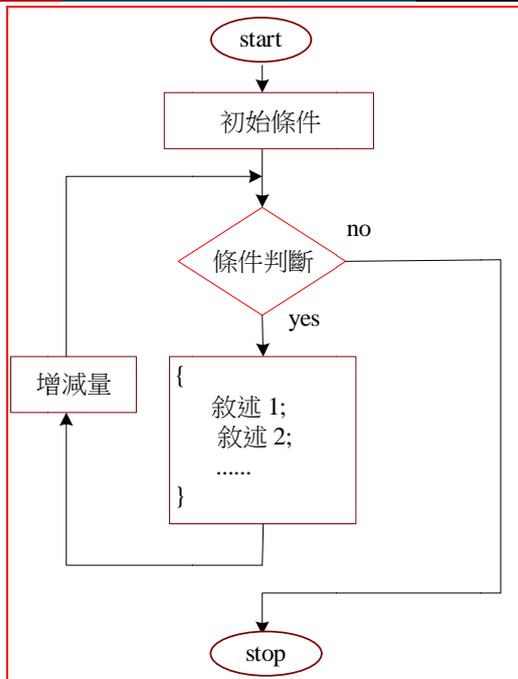


圖 5-1 for 敘述的流程圖

為了達成條件敘述的演算功能，在 for 敘述內必須指定一個或一個以上的條件變數，並初始設定條件變數內容、測試內容是否滿足條件、以及增減變數內容的數值。For 敘述句的語法與範例如下：

語法：	範例：由 1 累加到 10
<pre> for (條件初始; 條件判斷; 增減量) { Statement 1; Statement 2, } </pre>	<pre> int sum = 0; for (k=1; k <= 10; k = k +1) { sum = sum + k; //for body } System.out.println("Sum = " + sum); </pre>

For 敘述的小括號內包含 3 個敘述句，之間以『分號』(;) 分隔。第一個敘述句為條件變數的初值設定 (如 k = 1); 第二個為測試變數內容是否滿足條件 (如 k <=10); 第三個敘述句為執行 for 區塊敘述 (如 sum = sum + k) 後，再針對條件變數做增量或減量的變動 (如 k = k +1)。另一重點，如果區塊敘述超過一行的話，則必須利用左右大掛號包起來 ({ })。

在上述範例中，變數 k 作為 for 迴圈的計數基數，此變數有僅運用於 for 迴圈內，完成迴圈敘述後，該變數即失效了。許多情況下，為了減少變數宣告數量，期望宣告判斷變數僅存活於迴圈內就好，離開迴圈該變數自動消失，而將計數 (或判斷) 變數宣告成迴圈區域變數 (本書第 6-3 節介紹)。宣告方法是在 for 迴圈控制敘述的小掛號內指定，如 for(int k=0; k<=10; k++) { ... }。

5-2-2 範例研討：連續累加程式

(A) 程式功能：

請製作一只由 1 累加到 10 ($\text{sum} = 1 + 2 + 3 + \dots + 10$) 的程式，並輸出其運算結果；期望操作介面如下：

```
1 + 2 + .. + 10 = 55
```

(B) 製作技巧研討：

計算大串無聊複雜的數字，是『**計算機**』(**Computer**) 最主要的能耐，由 1 加到 10、100、1000、10000...等，對自然人來說有差別，對程式設計師來講卻都是一樣的，只要下對指令，電腦會不厭其煩地默默工作，絕不偷懶。累加程式需準備一個稱之為『**累積器**』(**Accumulator**，如 **sum**) 的箱子，將**連續數字累積加進去** ($\text{sum} = \text{sum} + \text{count}$)，如圖 5-2 所示。

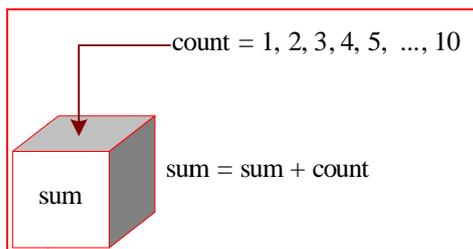


圖 5-2 累積器功能

吾人可利用 for 敘述句來製作連續由 1 到 10 重複運作，宣告產生一個整數變數 **sum** 作為累加器，並設定初始值為 0 ($\text{sum} = 0$)。再利用 **count** 變數作為計數指標 ($\text{count} = 1, 2, \dots, 10$)，也作為條件判斷數值。首先將指標設定初始值 1 ($\text{count} = 1$)、運作到 10 為止，表示滿足 $\text{count} \leq 10$ 條件，需重複執行迴圈，指標每次增值 1 ($\text{count} = \text{count} + 1$)；條件成立則執行『敘述區塊』，其內容為 $\text{sum} = \text{sum} + \text{count}$ 。將上述技巧套入 for 敘述句內，如圖 5-3 所示。

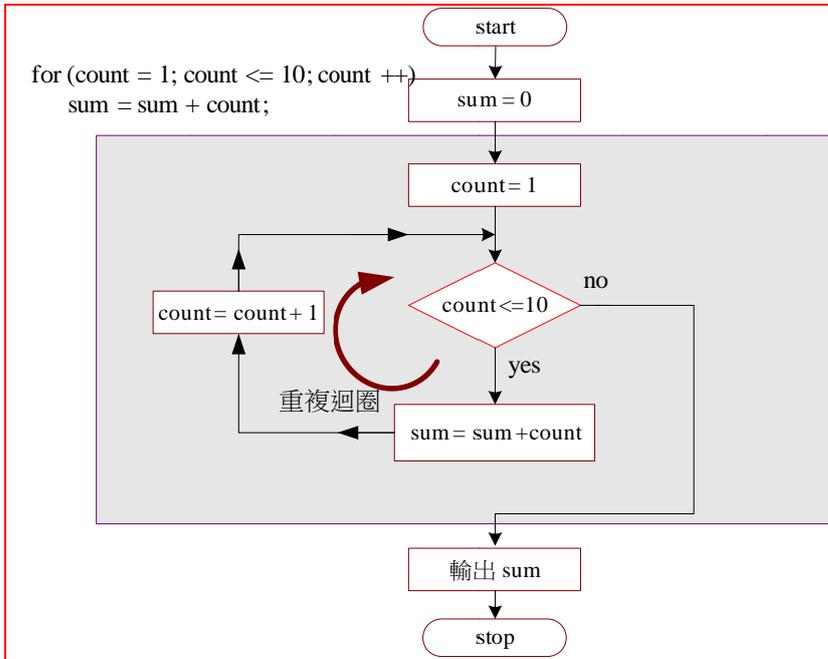


圖 5-3 連續累加流程圖

吾人將圖 5-3 運作流程，製表如下所示：

次數	count	count <=10 ?	sum = sum +count	count = count + 1
1	1	yes	sum = 0 +1 =1	count = 1+1=2
2	2	yes	sum = 1 +2 =3	count = 2+1=3
3	3	yes	sum = 3 +3 =6	count = 3+1=4
4	4	yes	sum = 6 +4 =10	count = 4+1=5
5	5	yes	sum = 10 +5 =15	count = 5+1=6
6	6	yes	sum = 15 +6 =21	count = 6+1=7
7	7	yes	sum = 21 +7 =28	count = 7+1=8
8	8	yes	sum = 28 +8 =36	count = 8+1=9
9	9	yes	sum = 36 +9 =45	count = 9+1=10
10	10	yes	sum = 45 +10 =55	count = 10+1=11
11	11	no		

(C) 程式範例：Ex5_1.java

```

01 // Ex5_1.java
02
03 public class Ex5_1 {
04     public static void main(String args[]) {
05         int count, sum;
06         sum = 0;
07         for (count = 1;count <= 10; count = count +1) {
08             sum = sum + count;
09         }
10         System.out.printf("1 + 2 + .. + 10 = %d\n", sum);
  
```

```

11     }
12 }

```

(D) 程式重點分析：

- 第 6 行：『sum = 0;』。累加器初值設定為 0。
- 第 7~9 行：『for (...) {}』。For 敘述句，其中 count = count + 1 可改為 count++；又敘述區塊僅有一條敘述句，則左右大括號 ({ ... }) 可以省略。
- 第 8 行：『sum = sum + count;』。累加計算功能，將原來 sum 變數的內取出與 k 相加後，再存入 sum 變數內。

5-2-3 自我挑戰：累加程式變化

累加程式最能表現出程式設計的特性。在現實環境中，資料越多或數值越大的情況下，處理事物起來就會越困難；但利用電腦處理則不然，無論處理資料多寡，都是一樣的，只不過時間較長一點而已（如電腦速度過快，也不會有很大的感覺）。以銀行查詢哪些客戶存款不足為範例，如利用人工查詢，當然客戶越多則越困難；但如利用電腦查詢，無論客戶有 100、1000、或 10000 筆，對程式設計師而言都沒什麼分別。我們利用 Ex5_1 範例，增加一點變化，吾人就可以瞭解其中奧妙。

(A) 程式變化(一)：PM5_1_1.java

功能：修改 Ex5_1.java 程式中一個地方，使其功能為 $sum = 1 + 2, \dots, 1000$ ，執行結果如下：

```
1 + 2 + .. + 1000 = 500500
```

(B) 程式變化(二)：PM5_1_2.java

功能：修改 Ex5_1.java，使其功能為 $sum = 100 + 99 \dots, + 1$ ，執行結果如下：

```
10099 + .. + 1 = 5050
```

(C) 程式變化(三)：PM5_1_3.java

功能：修改 Ex5_1.java，使其功能為 $sum = 0 + 2 + 4 + \dots, 1000$ ，其結果如下：

```
0 + 2 + .. + 1000 = 250500
```

(D) 程式變化(四)：PM5_1_4.java

功能：修改 Ex5_1.java，使其功能為 $sum = 1 + 3 + 5 + \dots + 999$ ，執行結果如下：

```
1 + 3 + .. + 999 = 250000
```

5-2-4 範例研討：連續累乘程式

(A) 程式功能：Ex5_2.java

請編寫一個 $total = 1 * 2 * 3 * \dots * 10$ (10!) 程式，並顯示每次累乘的計算結果。期望操作介面模式如下：

```
total * 1 = 1
total * 2 = 2
total * 3 = 6
total * 4 = 24
total * 5 = 120
total * 6 = 720
total * 7 = 5040
total * 8 = 40320
total * 9 = 362880
total * 10 = 3628800
```

(B) 製作技巧研討：

其實，連續累乘程式與累加程式非常相同，都是利用一個累積器，連續累積或累乘進入；兩者之間最大不同是，累加程式的累積器初始值必須設定為 0 ($sum = 0$)；累乘程式的累積器必須設定為 1 ($total = 1$)，否則任何數與 0 相乘結果都是 0。另外，**累乘計算的敘述句為 $total = total * count$** 。吾人利用虛擬碼規劃程式設計概要，如下：

```
宣告計數器 ( count , 1, 2, ..., 10 );
宣告累積器並設定初值為 1 ( total = 1 );
for (count = 1; count <= 10; count++) {
    total = total * count;
    列印 total, count;
}
```

(C) 程式範例：

```
01 // Ex5_2.java
```

```
02
03 public class Ex5_2 {
04     public static void main(String args[]) {
05         int count, total;
06         total= 1;
07         for (count = 1;count <= 10; count++) {
08             total = total * count;
09             System.out.printf("total * %d = %d\n", count, total);
10         }
11     }
12 }
```

(D) 程式重點分析：

- 第 7~10 行：『for(count = 1; count<=10; count++) { ..}』。For 敘述句，條件初始值設定為 1，當它小於或等於 10 則執行 for 程式區塊，執行完後再執行 count++ (增量 1)，再回去測試條件是否成立；如條件不成立，則離開 for 迴圈敘述。
- 第 8 行：『total = total * count;』。累乘程式功能，首先將 total 內容取出，與 count 相乘後結果，再存回 total 變數內。

5-2-5 自我挑戰：九九乘法評分系統

(A) 程式功能：PM5_2.java

請幫國民小學老師製作一套九九乘法的測試系統，系統會連續出現 10 題兩個 1~9 之間亂數，學生輸入自己計算相乘的結果，最後評定學生得分多寡 (每題 10 分)。期望操作介面如下：

```
== 歡迎進入九九乘法評分系統 ==
請輸入 9 * 7 = 63
請輸入 3 * 7 = 21
請輸入 3 * 3 = 9
請輸入 8 * 3 = 24
請輸入 6 * 3 = 15
錯了!! 答案是 18
請輸入 5 * 9 = 45
請輸入 3 * 2 = 7
錯了!! 答案是 6
```

```
請輸入 8 * 6 = 48
請輸入 3 * 7 = 21
請輸入 1 * 5 = 5
您得到 80 分
```

(B) 製作技巧提示：

系統要求連續出現 10 次九九乘法練習題，這方面可利用 for 迴圈達成，其中，計數器(count) 由 1 到 10，每次增量 1 (count++，或 count = count + 1)。迴圈內必須產生兩個 1~9 的亂數，如此需要導入 java.util.Random 套件；另外，也需要導入鍵盤輸入套件，吾人可選用 java.util.Scanner 套件。兩套件都是屬於 java.util 套件底下，因此只要導入該套件即可，即是『import java.util.*;』。系統虛擬碼如下：(參考 Ex4_3.java)

```
導入相關套件 ( import java.util.*; );
宣告與產生亂數物件 ( Random random )
宣告與產生輸入物件 ( Scanner keyin );
宣告相關變數；
for (count = 1; count <= 10; count++) {
    產生第一個 1 ~ 9 亂數 ( ran1 = 1 + random.nextInt(9) );
    產生第二個亂數 ( ran2 );
    計算正確答案 ( value = ran1 * value2 )
    顯示兩亂數並要求輸入答案 ( ran1, ran2 );
    讀入學生答案 ( answer = keyin.nextInt() );
    if (answer == value)
        累加學生答對題數 ( correct = correct + 1 );
    else
        顯示學生答錯了，並給予正確答案;
}
列印學分得到分數 ( correct * 10 );
```

5-3 while 迴圈敘述

5-3-1 while 敘述的流程

圖 5-4 為 while 敘述的運作流程圖，首先判斷條件是否成立，成立 (yes) 則執行 while 迴圈實體，否則 (no) 不執行 while 迴圈，並離開此 while 敘述；當條件成立並執行迴圈實體之後，再測試條件是否成立，如成立則繼續執行迴圈，否則跳離迴圈。

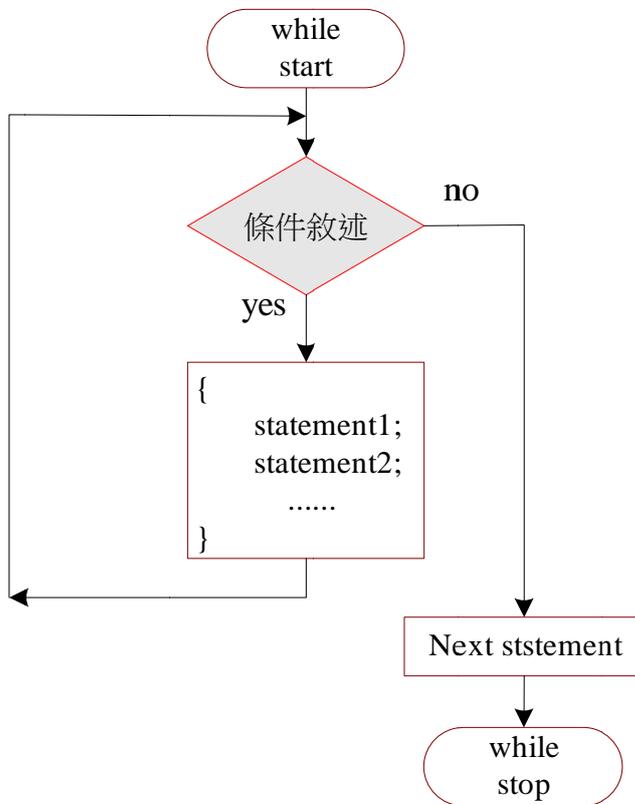


圖 5-4 While 敘述的流程圖

由圖 5-4 可以看出，while 敘述比 for 迴圈簡單了許多，並不設定條件初值，也不計算條件變數的增減量，而僅判斷條件是否成立，來決定是否重複執行迴圈敘述區塊。While 敘述的語法與範例如下：

語法：	範例：
<pre>while (條件判斷) { statement 1; statement 2; }</pre>	<pre>k=1; while(k <= value) { sum = sum + k; pro = pro * k; k = k +1; }</pre>

5-3-2 範例研討：累加/累乘程式

(A) 程式功能：Ex5_3.java

製作一個具有累加與累乘功能的計算器，系統要求輸入一個整數 n，再分別計算 1 累加到 n (sum = 1 + 2 + 3 + ... + n) 與 1 累乘到 n (pro = 1 * 2 * 3 * ... * n)，並輸出兩則計算結果。期望操作介面如下：

```

請輸入一個整數 =>20
1+2+3+4+...+20 = 210
1*2*3*... *20 = 2432902008176640000

```

(B) 製作技巧研討：

無論採用何種迴圈敘述句，連續累加或累乘程式大致上與圖 5-3 一樣。吾人修改圖 5-3，使其符合 while 迴圈敘述並加入累乘功能，如圖 5-5 所示 (陰暗部分)。進入迴圈之前，將計數器設定為 1 (count = 1)，進入 while 敘述句立即判斷 count 是否小於或等於 10 (count <=10)，如是則執行累加與累乘的敘述句，計數器也累加 1 (count++)，之後再回來測試條件是否成立。條件成立 (count <=10) 則再執行 while 程式區塊，否則離開 while 敘述句。

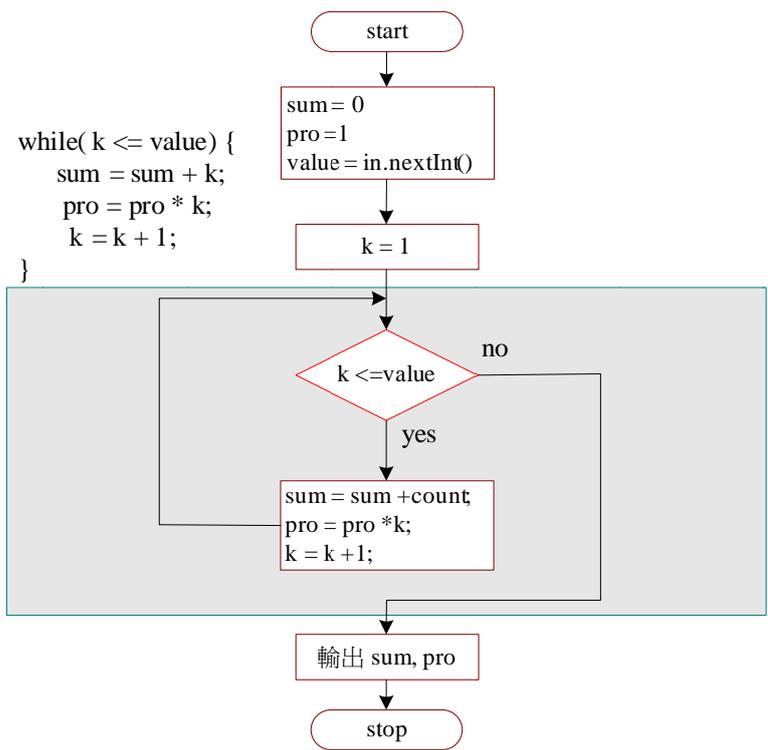


圖 5-5 while 敘述製作累加/累乘程式

(C) 程式範例：

```

01 //Ex5_3.java
02
03 import java.util.*;
04 public class Ex5_3 {

```

```

05     public static void main(String args[]) {
06         Scanner keyin = new Scanner(System.in);
07         long sum=0, pro=1, value, k;
08         System.out.printf("請輸入一個整數 =>");
09         value = keyin.nextInt();
10         k=1;
11         while(k <= value) {
12             sum = sum + k;
13             pro = pro * k;
14             k = k + 1;
15         }
16         System.out.printf("1+2+3+4+...+%d = %d\n", value, sum);
17         System.out.printf("1*2*3*... *%d = %d\n", value, pro);
18     }
19 }

```

(D) 程式重點說明：

- 第 7 行：『long sum=0, pro=1;』。分別設定累加與累乘之累積器的初始值。
- 第 10 行：『k=1;』。初始值設定，使用於 while 敘述的條件判斷式。
- 第 11~15 行：『while (k <=value) {}』。While 敘述句，判斷 k <= value 條件成立的話，則執行迴圈內敘述區塊；否則結束 while 敘述。
- 第 14 行：『k = k+1;』。條件值增量，如果沒有此敘述，while 迴圈將永遠做不完。

5-3-3 範例研討：製作電子算盤

(A) 程式功能：Ex5_4.java

早期傳統『柑仔店』大多利用算盤計算『人客』購買商品的總額。它的計算方法是一筆接一筆輸入購買金額，也一筆一筆累計總額，輸入結束後立即得到購買金額多寡。請編寫製作一套模擬算盤程式，吾人可連續輸入購買金額，也能隨時累計總額；再利用輸入價格為『零』，表示結束輸入，並顯示總計金額。期望操作介面如下：

```

歡迎光臨 == 春嬌柑仔店 ==

已購買 0 請輸入產品價格 (0 結束) =>50
已購買 50 請輸入產品價格 (0 結束) =>76

```

```
已購買 126 請輸入產品價格 (0 結束) =>230
已購買 356 請輸入產品價格 (0 結束) =>0
人客您總共買 356 元 謝謝!! 愛過來ㄜ
```

(B) 製作技巧研討：

如果沒有設定一個關鍵條件，算盤好像會無止境的一直累加不停；本範例利用價格為 0 作為停止計算的關鍵因素，此類型系統利用 `while` 敘述句，最恰當不過了。吾人將系統重點部份以虛擬碼規劃如下：

```
導入輸入套件；
宣告相關變數，並設定適當初值；
顯示目前購買小計 ( total )，並要求輸入下一個產品價格；
讀入產品價格 ( cost )；
while (cost > 0) {
    累計購買小計 ( total = total + cost )；
    顯示目前購買小計 ( total )，並要求輸入下一個產品價格；
    讀入產品價格 ( cost )；
}
顯示客戶總購買金額；
```

(C) 程式範例：

```
01 //Ex5_4.java
02
03 import java.util.*;
04 public class Ex5_4 {
05     public static void main(String args[]) {
06         Scanner keyin = new Scanner(System.in);
07         int total=0, cost;
08         System.out.printf("歡迎光臨 == 春嬌柑仔店 ==\n\n");
09         System.out.printf("已購買 %d 請輸入產品價格 (0 結束) =>", total);
10         cost = keyin.nextInt();
11         while (cost > 0) {
12             total = total + cost;
13             System.out.printf("已購買 %d 請輸入產品價格 (0 結束) =>", total);
14             cost = keyin.nextInt();
15         }
16     }
```

```

17     System.out.printf("人客您總共買 %d 元 謝謝!! 愛過來ㄝ\n", total);
18     }
19 }

```

5-3-4 自我挑戰：超商收銀機

(A) 程式功能：PM5_2.java

利用算盤結算總額最大缺點是，客戶購買相同產品時，也是需要一個接一個輸入價格。目前超商收銀機大多允許輸入單價及數量，它會自動計算並累加總額。請您幫超商建立一套收銀機系統，系統會連續要求輸入貨品單價與數量，以及顯示目前小計多寡，如果輸入貨品單價為 0 時，表示結束輸入，之後顯示購買總金額。期望操作介面如下：

```

*** 歡迎光臨 春嬌超商 ***
累計 0.0 請輸入貨品單價 (0 結束) =>56
    請輸入購買數量 =>2
累計 112.0 請輸入貨品單價 (0 結束) =>67
    請輸入購買數量 =>4
累計 380.0 請輸入貨品單價 (0 結束) =>238
    請輸入購買數量 =>2
累計 856.0 請輸入貨品單價 (0 結束) =>0
總計 => 856.0

```

(B) 製作技巧提示：

吾人稍微修改 Ex5_4 範例，即可滿足系統要求，虛擬碼重點提示如下：

```

...
設定購買總額初值 ( total =0 )
顯示目前購買小計 ( total )，並要求輸入下一個產品價格；
讀入產品價格 ( cost )；
while (cost > 0) {
    顯示輸入購買數量，並讀取 ( number )；
    累計購買小計 ( total = total + cost*number )；
    顯示目前購買小計 ( total )，並要求輸入下一個產品價格；
}

```

```

        讀入產品價格 ( cost );
    }
    顯示客戶總購買金額 ;
    
```

5-4 do/while 迴圈控制

5-4-1 do/while 敘述流程

do/while 敘述與 while 迴圈非常相似，皆僅判斷條件與執行迴圈實體而已，但兩者還有稍微不同的地方。do-while 敘述是先執行迴圈實體後再測試判斷條件，如果成立則繼續執行，否則跳離迴圈；而 while 迴圈是先測試條件，條件成立則繼續執行迴圈實體，否則跳離迴圈。由此可見，do-while 迴圈至少會執行迴圈實體一次。

圖 5-6 為 do/while 敘述的流程圖。當程式進入 do/while 敘述後，立即執行迴圈實體，之後再測試條件，如成立的話，則再重複執行；否則結束該敘述。語法與範例如下：

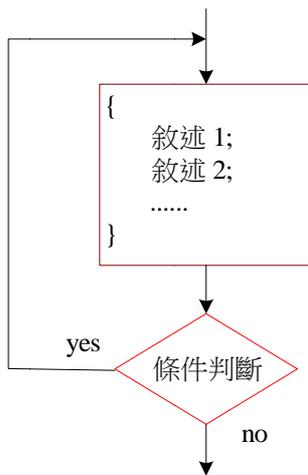


圖 5-6 do/while 敘述的流程圖

語法：	範例：
<pre> do { Statement 1; Statement 2; }while (條件判斷); </pre>	<pre> do { sum = sum + k; k = k+1; }while(k <100); </pre>

5-4-2 範例研討：do/while 累加程式

(A) 程式功能：Ex5_5.java

請利用 do-while 迴圈編寫一只 $sum = 100 + 99 + 98 + \dots + 1$ 的程式，並印出其結果。期望操作介面如下：

100 +... + 1 = 5050

(B) 製作技巧研討：

如同 for 與 while 迴圈，do/while 敘述句亦可以製作連續累加或累乘程式。但此題目希望由高累加到低數字，判斷條件的計數器必須由高遞減到低 (count--)，吾人將程式流程繪製如圖 5-7 所示。

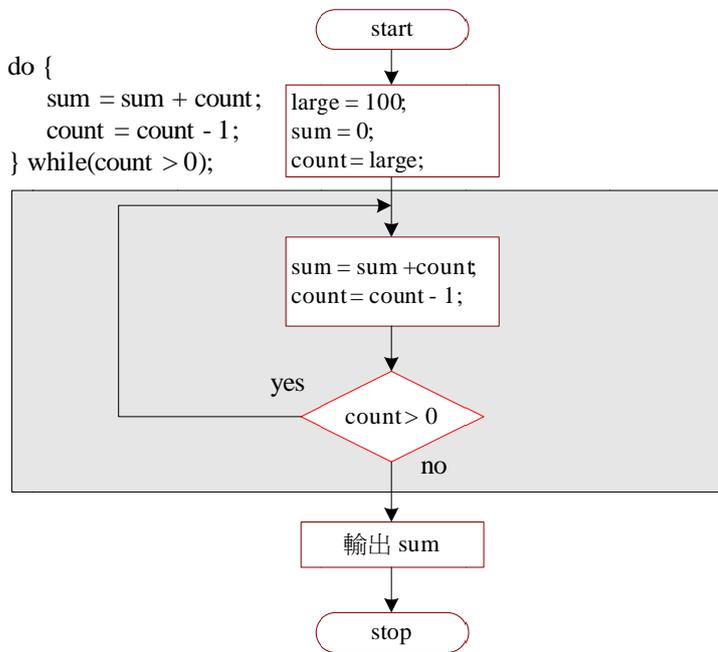


圖 5-7 do/while 累加程式的流程圖

(C) 程式範例：

```

01 // Ex5_5.java
02
03 public class Ex5_5 {
04     public static void main(String args[]) {
05         int count, limit = 100, sum=0;
06         count = limit;
07         do {
08             sum = sum + count;
09             count = count - 1;
10         } while (count > 0);
11         System.out.printf("%d +... + 1 = %d\n", limit, sum);
  
```

```

12     }
13     }

```

(D) 程式重點分析：

- 第 6 行：『count = limit;』。將判斷條件用的計數器設定最高數值 (100)。
- 第 7~10 行：『do {}while(count > 0);』。Do/while 敘述句。
- 第 9 行：『 count = count -1;』。計數器遞減 (如同 count--)。

5-4-3 範例研討：分期貸款試算系統

(A) 程式功能：Ex5_6.java

目前銀行大多有提供定額還款的貸款業務，無論客戶承貸多少金額 (如 100 萬元)，可選擇每個月還款多寡 (如 2 萬元)，其中包含利息 (如年率 4 %)，一直到債務還清為止。許多年輕人不瞭解自己還款能力，擴張貸款額度可能造成一輩子的債務人。因此，銀行希望製作一套分期貸款試算系統，客戶自行輸入貸款額度，目前年利率、預計每月還款金額；因年利率隨時調變，系統僅能輸出大約需要還款多少期 (月)。如果預估期數超 20 年 (240 期)，系統將自動拒絕承做，否則歡迎洽臨行員。期望系統操作介面如下：

情況一：允許貸款

```

== 歡迎光臨藝術銀行貸款試算系統 ==
請輸入貸款金額 (萬) =>100
請輸入年利率 (如0.04) =>0.04
期望每月還款額度 (萬) =>2
您需要 54 期(月) 才可還款完畢
本行願意承做, 歡迎洽詢行員

```

情況二：拒絕貸款

```

== 歡迎光臨藝術銀行貸款試算系統 ==
請輸入貸款金額 (萬) =>200
請輸入年利率 (如0.04) =>0.08
期望每月還款額度 (萬) =>1.5
您需要 330 期(月) 才可還款完畢

```

還款期限超過 240 請另選貸款額度

(B) 製作技巧研討：

雖然貸款是以年利率計算，但客戶是每月清還利息，因此每月餘額計算時需將原餘額再加上當月的利息。吾人利用虛擬碼規劃系統大綱如下：

```
宣告相關物件與變數；
讀入客戶貸款額度 ( credit )、年利率 ( rate ) 與每月還款 ( pay )；
設定累進期數初始值 ( count=0 )；
設定初始餘額 ( balance = credit )；
Do {
    計算目前餘額 ( balance = balance + (balance*rate)/12 - pay )
    累加還款期數 ( count = count + 1 )
} while(是否還剩餘額 ( balance >0))；
輸出顯示還款期數 ( count )；
If (期數超過 240)
    顯示 "拒絕承做"；
Else
    顯示 "願意承做"；
```

(C) 程式範例：

```
01 //Ex5_6.java
02
03 import java.util.*;
04 public class Ex5_6 {
05     public static void main(String args[] ) {
06         /* 宣告相關物件與變數 */
07         Scanner keyin = new Scanner(System.in);
08         final int limit = 240; //最高期限
09
10         double credit; // 貸款額度(萬)
11         double pay; // 每月攤還金額(萬)
12         double rate; // 年利率
13         int count=0; // 累計繳納期數
```

```
16     double balance;           // 目前貸款餘額
17
18
19     /* 讀取客戶輸入資料 */
20     System.out.printf("== 歡迎光臨藝術銀行貸款試算系統 ==\n");
21
22     System.out.printf("請輸入貸款金額 (萬) =>");
23     credit = keyin.nextDouble();
24     System.out.printf("請輸入年利率 (如 0.04) =>");
25     rate = keyin.nextDouble();
26     System.out.printf("期望每月還款額度 (萬) =>");
27     pay = keyin.nextDouble();
28     /* 計算還款期數 */
29     balance = credit;
30     do {
31         balance = balance + (balance*rate)/12;
32         balance = balance - pay;
33         count = count + 1;
34     }while(balance > 0);
35     System.out.printf("您需要 %d 期(月) 才可還款完畢\n", count-1);
36     if (count > limit)
37         System.out.printf("超過期數 %d 請另選貸款額度\n", limit);
38     else
39         System.out.printf("本行願意承做, 歡迎洽詢行員\n");
40
41
42
43
44 }
```

(D) 程式重點分析：

- 第 27~31 行：『do { ... }while(balance >0);』。Do/while 敘述句。
- 第 32 行：『System.out.printf(..., count-1);』。當 balance 已經小於 0 了，但還是執行 count = count + 1 之後再判斷，因此還要再扣除回來。

5-4-4 自我挑戰：學貸償還試算系統

(A) 程式功能：PM5_4.java

目前許多學生都有申請學生貸款，依政府規定畢業後一年須開始償還貸款，並按年金法按月平均攤還本息，償還期限以申請期間為準，每一學期得以一年計，依此類推。如此限定期間內還款

又稱為『定期貸款』。承辦銀行希望能讓學生理解繳款狀況，建立一套『學貸償還試算系統』，讓學生提早規畫畢業後，如何償還貸款。系統要求學生輸入貸款額度(如 40 萬)、返還期限(如大學四年則 8 年)、銀行利息(如年利率 4%)，則預估每月應繳多少錢(請列出前 10 個月)？以及到償還完畢後，延伸多少利息？期望操作介面如下：

對於經濟能力較好的客戶總是希望能快速還完貸款，銀行也提供定期返還貸款辦法；方法是客戶每月平均返還本金，再加上當月的利息。如此一來，客戶每月繳交的金額將不同，前面幾期繳交越多，最後幾期利息越少。銀行也希望客戶辦理貸款之前，能先評估自己還款能力如何，因此希望建立一套『定期貸款試算系統』。客戶自行輸入貸款額度、年利率、預估還款期數，則系統能列印出每月應該繳納的金額。期望操作介面如下：

```
== 學貸償還試算系統 ==  
請輸入貸款金額 (萬) =>40  
請輸入年利率 (如0.04) =>0.04  
償還年限(貸款學期數) =>8  
前 10 期應繳納金額如下：  
您第 1 期需繳納 0.55 萬元  
您第 2 期需繳納 0.55 萬元  
您第 3 期需繳納 0.55 萬元  
您第 4 期需繳納 0.55 萬元  
您第 5 期需繳納 0.54 萬元  
您第 6 期需繳納 0.54 萬元  
您第 7 期需繳納 0.54 萬元  
您第 8 期需繳納 0.54 萬元  
您第 9 期需繳納 0.54 萬元  
到清償為止，總共延伸利息 6.47 萬元
```

(B) 製作技巧提示：

此系統需先計算出客戶每個分攤多少貸款額度，每月計算利息時再扣除已還清的本金，由餘額計算出當月應繳納的利息；分攤本金加上當月利息，就是該月需繳納金額。虛擬碼規劃如下：

```
宣告相關物件與變數；
```

```
讀入客戶貸款額度 ( credit )、年利率 ( rate ) 與預計返還期數 ( stages );
計算每月平均分攤本金 ( pay_month = credit/stages );
設定初始餘額 ( balance = credit );
Do {
    本月延伸利息 ( interest = (balance*rate)/12 );
    計算目前該月應繳金額 ( pay = pay_month +interest );
    計算本月餘額 ( balance = balance – pay_month );
    累加還款期數 ( count = count + 1 );
    累加產生利息(sum = sum + interest);
    輸出顯示本月期數與應繳金額 ( count, pay );
} while(是否已到期 ( count < stages );
...
```

5-4-5 範例研討：錢莊借貸試算系統

(A) 系統功能：Ex5_7

一般錢莊(或稱資產管理公司、應收款管理公司)提供顧客緊急貸款需求，客戶能提供抵押品大多非常薄弱，譬如：舊汽車、機車、手錶等等不值錢的物品，甚至給人頭保證的信用貸款(備注：如果足夠抵押品就向銀行貸了)。相對應的，它的放款金額較低，利率也比較高，並且須按月繳還利息，否則利息加入下個月的本金(俗稱利息滾本金)。其利息算法如下：

- (1) 照政府規定年利率不得超過 48%，即是月利率 4%(俗稱四分期)。也有 10 天為一期 2%(俗稱二分期)。
- (2) 照政府規定手續費不得超過 5%。
- (3) 實際取得金額 = 借貸金額 – 第一個月利息 – 手續費。
- (4) 按月繳交利息，否則加入本金借貸，及下個月是本金=上月本金 + 利息+ 手續費。

許多人貪一時之快，造成滿身債務。政府希望市民能了解此運作情況，建立一套試算系統，功能是，市民輸入貸款金額(如 10 萬)，實際取得多少錢(扣除第一個月利息與手續費)？爾後每月應繳多少利息？如隔多少個月(如 10 個月)未繳利息後，則本金成長為多少？之後每月應繳多少利息。期望操作介面如下：

```
== 歡迎光臨錢莊貸款試算系統 ==  
請輸入貸款金額 (萬) =>20  
請輸入月利率 (如0.04) =>0.04  
多個月沒繳交利息(如 10 個月) =>20  
列印相關借貸數據如下：  
實際取得貸款金額 = 18.20 萬元  
每個月應繳交計息 = 0.80 萬元  
未繳利息 20 個月後，本金累積為 112.09 萬元  
    之後每月利息 = 4.48 萬元
```

(B) 程式範例：

```
01 //Ex5_7.java
02
03 import java.util.*;
04 public class Ex5_7 {
05     public static void main(String args[]) {
06         /* 宣告相關物件與變數 */
07
08         Scanner keyin = new Scanner(System.in);
09         double credit;           // 貸款額度(萬)
10         double rate;            // 月利率
11         int num;                // 未繳交利息月數
12         double total;           // 貸款取得金額
13
14
15
16         /* 讀取客戶貸款資料 */
17
18         System.out.printf("== 歡迎光臨錢莊貸款試算系統 ==\n");
19
20         System.out.printf("請輸入貸款金額 (萬) =>");
21         credit = keyin.nextDouble();
22
23         System.out.printf("請輸入月利率 (如0.04) =>");
24         rate = keyin.nextDouble();
25
26         System.out.printf("多個月沒繳交利息(如 10 個月) =>");
27         num = keyin.nextInt();
28
29         System.out.printf("列印相關借貸數據如下 : \n");
30         total = credit - credit * 0.04 - credit * 0.05;
31
32         System.out.printf("實際取得貸款金額 = %.2f 萬元\n", total);
33         System.out.printf("每個月應繳交計息 = %.2f 萬元\n", credit*0.04);
34         for(int i=0; i<num; i++)
35             credit = credit + credit * 0.04 + credit * 0.05;
36         System.out.printf("未繳利息 %d 個月後 · 本金累積為 %.2f 萬元\n",num, credit);
37         System.out.printf("        之後每月利息 = %.2f 萬元\n", credit*0.04);
38     }
39 }
```

5-5 多重迴圈敘述

5-5-1 多重迴圈流程

多重迴圈表示迴圈內還有子迴圈，依此延伸下去。圖 5-8 是利用兩個 for 敘述構成了雙重迴

圈，外迴圈利用變數 a 作為迴圈條件變數並由 1 到 10，迴圈執行一次，則累增加 1。內迴圈的條件變數是 b。也是迴圈執行一次，由 1 到 10 每次累增加 1。然而，外迴圈執行一次，則內迴圈必須由 1 到 10 執行 10 次迴圈實體。其運作情況如下：

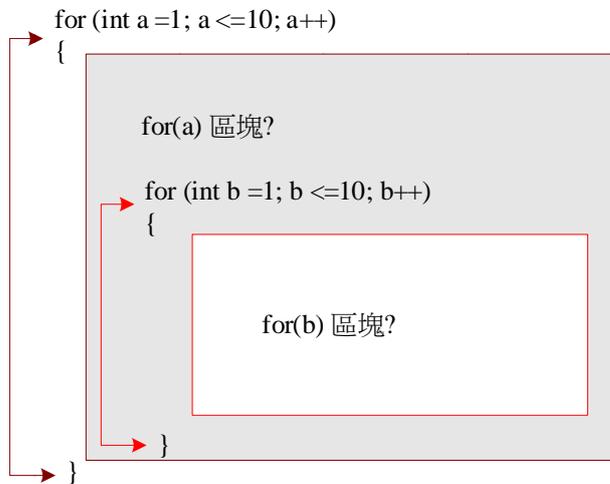


圖 5-8 利用 for 敘述建構雙重迴圈

外迴圈執行順序	內迴圈執行順序
a = 1	b= 1, 2, 3, 4, 5, 6, 7, 8, 9 10
a = 2	b= 1, 2, 3, 4, 5, 6, 7, 8, 9 10
a = 3	b= 1, 2, 3, 4, 5, 6, 7, 8, 9 10
a = 4	b= 1, 2, 3, 4, 5, 6, 7, 8, 9 10
a = 5	b= 1, 2, 3, 4, 5, 6, 7, 8, 9 10
a = 6	b= 1, 2, 3, 4, 5, 6, 7, 8, 9 10
a = 7	b= 1, 2, 3, 4, 5, 6, 7, 8, 9 10
a = 8	b= 1, 2, 3, 4, 5, 6, 7, 8, 9 10
a = 9	b= 1, 2, 3, 4, 5, 6, 7, 8, 9 10
a = 10	b= 1, 2, 3, 4, 5, 6, 7, 8, 9 10

由此範例可以看出，外迴圈實體需執行 10 次才結束，而相對的，內迴圈實體已執行了 100 次。同樣的結構敘述三重迴圈的話，最內層迴圈必須執行 1000 次、中層迴圈 100 次、而外層迴圈 10 次。利用 for 敘述建構雙重迴圈的語法與範例如下。

語法：	範例：
<pre>for (a=1; a<=10; a++) { statement 1; statement 2; for(b=1; b<=10; b++) { statement 3; statement 4; } }</pre>	<pre>for (j = 1; j <= 10; j++) { System.out.printf(" "); for (k = 1; k <= 10; k++) { System.out.printf("*"); } System.out.print("\n"); }</pre>


```
01 //Ex5_7.java
02
03 import java.util.*;
04 public class Ex5_7 {
05     public static void main(String args[]) {
06         /* 宣告相關物件與變數 */
07         Scanner keyin = new Scanner(System.in);
08         int height;           // 矩形的高度
09         int wide;            // 矩形的寬度
10
11
12         /* 要求輸入矩形高與寬 */
13
14         System.out.printf("請輸入矩形高度 =>");
15         height = keyin.nextInt();
16         System.out.printf("請輸入矩形寬度 =>");
17         wide = keyin.nextInt();
18
19
20         /* 繪製矩形圖形 */
21         for(int i=1; i<=height; i++) {
22             System.out.printf("    ");
23             for(int j=1; j<=wide; j++) {
24                 System.out.printf("*");
25             }
26             System.out.printf("\n");
27         }
28     }
29 }
```

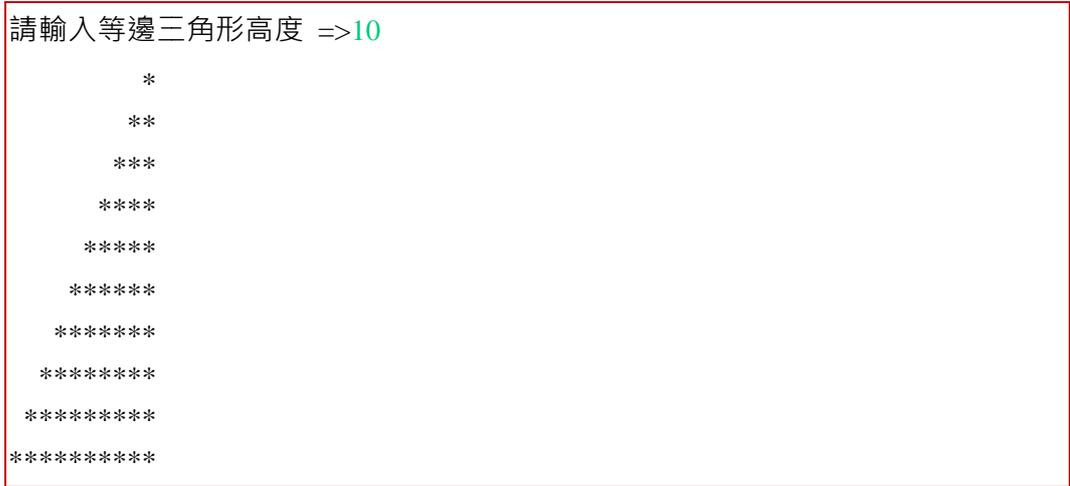
(D) 程式重點分析：

- 第 20~26 行：『for(int i=1; i<=height, i++) { ... }』。指定矩形高度的外迴圈，每執行一次迴圈，則繪出一行，共計執行 height (= 5) 次。
- 第 21 行：『System.out.printf(" ");』。每行起頭繪出 4 個空格。
- 第 22~24 行：『for (int j=1; j<=wide; j++){ ..}』。指定矩形寬度的內迴圈；每執行一次迴圈，畫出一個星號 (*)，共計執行 wide (= 8) 次。
- 第 25 行：『System.out.printf("\n");』。一行畫製完成之後，送出一個跳行控制字元，準備畫製下一行。

5-5-3 自我挑戰：畫三角形程式

(A) 程式功能：PM5_5.java

請製作畫右邊等齊的三角形，高度可以由使用者輸入指定，期望操作介面如下：



(B) 製作技巧提示：

還是需要利用雙重迴圈來製作此程式，外迴圈決定三角形的高度，但需要兩個內迴圈來製作三角形的寬度，一則畫空白格；另一迴圈輸出星號，如圖 5-10 所示。虛擬碼提示如下：

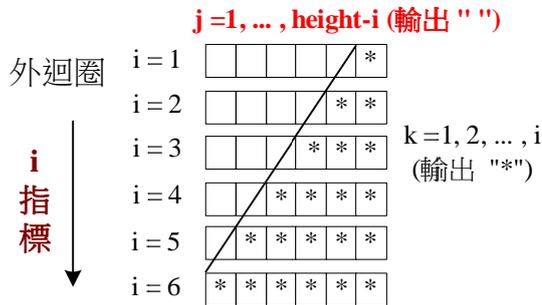


圖 5-10 三角形繪製程序

```

.....
讀入使用者輸入高度 ( height );
for(int i=1; i<=height; i++) {           // 外迴圈，指定高度
    for(int j=1; j<=(height -i); j++) // 第 1 個內迴圈，輸出空格
        System.out.printf(" ");
    for(int k=1; k<=i; k++)           // 第 2 個內迴圈，輸出星號
        System.out.printf("*");
    System.out.printf("\n");         // 換行
}

```

5-5-4 自我挑戰：畫各種等邊三角形

(A) 程式功能：PM5_6.java

請製作畫四種等邊三角形的圖型，高度可以由使用者輸入指定，期望操作介面如下：

```

請輸入等邊三角形高度 =>10
畫出左邊的等邊三角形高度：
      *
     **
    ***
   ****
  *****
 *****
*****
*****
*****
*****

畫出反向左邊的等邊三角形高度：
*****
*****
 *****
  *****
   *****
    *****
     *****
      *****
       *****
        *****
         *****

畫出右邊的等邊三角形高度：
*
**
***
****
*****
*****
*****
*****
*****
*****
*****
*****

畫出反向右邊的等邊三角形高度：
*****
*****
*****

```



```

*****
****
***
**
*
    
```

(B) 製作技巧提示：(請自行思考)

5-5-6 範例研討：for 列印九九乘法表

(A) 程式功能：Ex5_9.java

請利用 for 雙重迴圈，製作九九乘法表，期望輸出格式如下：

2*1= 2	3*1= 3	4*1= 4	5*1= 5	6*1= 6	7*1= 7	8*1= 8	9*1= 9
2*2= 4	3*2= 6	4*2= 8	5*2=10	6*2=12	7*2=14	8*2=16	9*2=18
2*3= 6	3*3= 9	4*3=12	5*3=15	6*3=18	7*3=21	8*3=24	9*3=27
2*4= 8	3*4=12	4*4=16	5*4=20	6*4=24	7*4=28	8*4=32	9*4=36
2*5=10	3*5=15	4*5=20	5*5=25	6*5=30	7*5=35	8*5=40	9*5=45
2*6=12	3*6=18	4*6=24	5*6=30	6*6=36	7*6=42	8*6=48	9*6=54
2*7=14	3*7=21	4*7=28	5*7=35	6*7=42	7*7=49	8*7=56	9*7=63
2*8=16	3*8=24	4*8=32	5*8=40	6*8=48	7*8=56	8*8=64	9*8=72
2*9=18	3*9=27	4*9=36	5*9=45	6*9=54	7*9=63	8*9=72	9*9=81

(B) 製作技巧研討：

列印九九乘法表，最能表現出雙重迴圈的特性，幾乎所有初學者都必須學會此程式。

習慣上，九九乘法表大多是直列式排序，這剛好與電腦顯示方法相反。除非是繪圖程式，否則一般文字模式都是一行顯示完再接著顯示下一行，無法做到顯示下一行後再回到上一行。因此，列印九九乘法表時也須依照表內文字，一行接一行列印。

圖 5-11 為此程式的概要說明，每行由 2 到 9，共有 8 句文字，我們利用 i 做為指標變數 (j = 2, 3, 4, ..., 9)；相對的，每列有 9 句文字，我們利用 j 做為指標變數 (i = 1, 2, ..., 9)。程式必須一行接一行列印，總共有 9 行，因此，必須外迴圈須以 i 做為指標；每行列印 8 句，因此內迴圈變數為 j；每句文字是 j * i = ij。

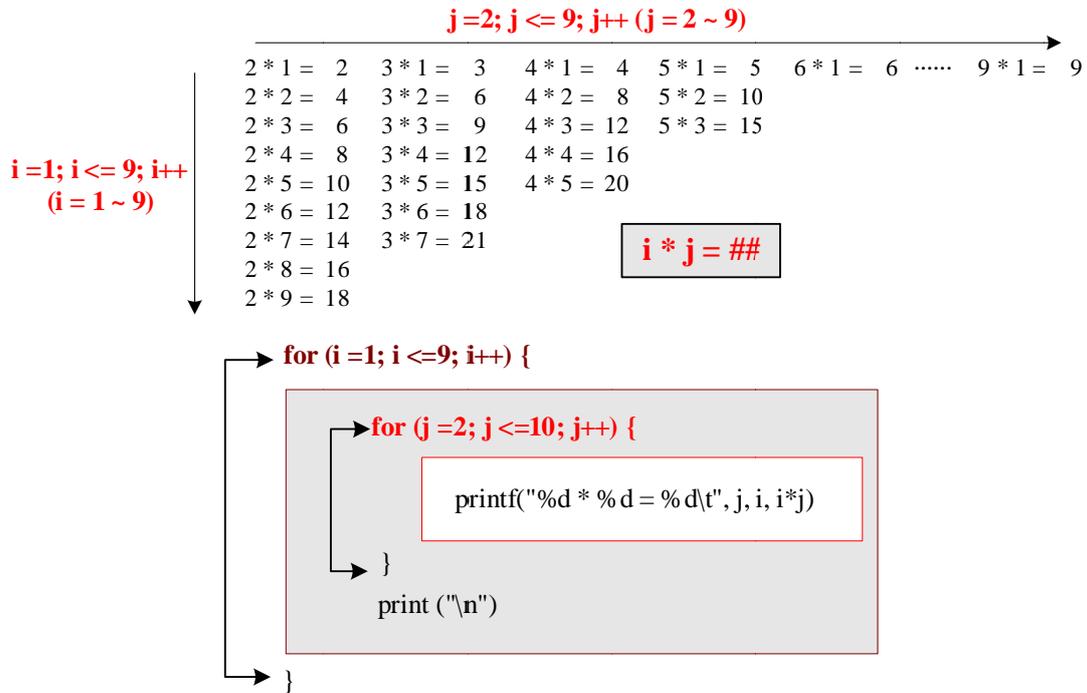


圖 5-11 九九乘法表製作技巧

(C) 程式範例：

```

01 //Ex5_9.java
02
03 public class Ex5_9 {
04     public static void main(String args[]) {
05         for (int i=1; i<=9; i++) {
06             for (int j=2; j<=9; j++)
07                 System.out.printf("%d*%d=%2d\t", j, i, i*j);
08             System.out.printf("\n");
09         }
10     }
11 }

```

(D) 程式重點分析：

- 第 5~9 行：『for (i=1; i<=9; i++) { ... }』。外迴圈指定九九乘法表的行數。
- 第 6~7 行：『for (j=2; j<=9; j++) System.out.printf(...)』。內迴圈。
- 第 7 行：『System.out.printf("%d*%d=%2d\t", j, i, i*j)』。製作九九乘法語句的格式，其中 %2d 表示利用兩個字元格式印出整數。
- 第 8 行：『System.out.printf("\n");』。每行最後加入跳行的控制字元。

5-5-7 自我挑戰：while 列印九九乘法表

(A) 程式功能：PM5_8.java

請利用 while 敘述句製作列印九九乘法表程式。

(B) 製作技巧提示：

無論採用何種敘述句 (for、while、do/while) 製作九九乘法表，技巧都是相同的，如圖 5-# 所示。利用 while 敘述句的提示如下：

```
int i = 1;
while (i <= 9) {
    int j = 2;
    while (j <= 9) {
        System.out.printf("%d*%d=%2d\t", j, i, i*j);
        j++;
    }
    i++;
    System.out.printf("\n");
}
```

(C) 腦力激盪：

請利用 do/while 敘述句，製作列印九九乘法的程式。

5-6 迴圈中斷與連續

除了條件判斷是否繼續重複執行迴圈實體 (body) 外，在迴圈實體內也依照某些因素決定是否繼續執行實體敘述，或直接跳離開該迴圈敘述。一般程式語言都有提供 break 與 continue 兩種敘述，執行到前者敘述時，則直接中斷該迴圈敘述；後者是立即再回到迴圈判斷式，再決定是否繼續執行迴圈實體。以下分別介紹這兩種敘述語法。

5-6-1 break 敘述的流程

無論在迴圈 (for、while、do/while) 或『方法』(method) 的實體區塊內，執行到 break 敘述後，立即停止繼續往下執行，並離開該迴圈或方法。如果是多重迴圈的話，僅會離開 break 敘述所屬迴圈，繼續下一個外圍迴圈敘述。圖 5-12 為 break 敘述的流程，我們以 while 迴圈為例，

當程式執行到 break 敘述後則跳離開，繼續 while 敘述後的下一個敘述句。

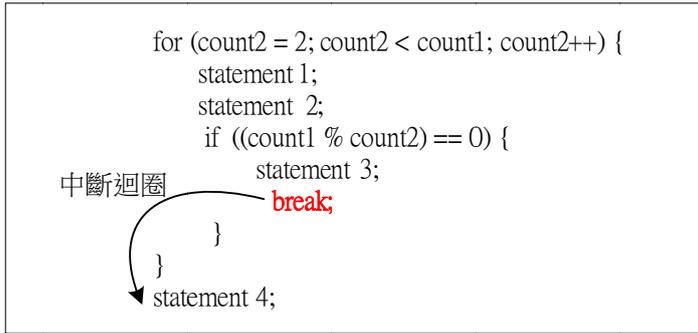


圖 5-12 break 敘述的流程圖

5-6-2 continue 敘述的流程

程式執行到 continue 敘述後，立即停止執行迴圈實體其他敘述，強迫回到迴圈的起頭，再測試條件判斷敘述，決定是否繼續重複執行迴圈。圖 5-13 是以 while 迴圈為例，當執行到 continue 後，立即回到 while 敘述起頭，再判斷是否繼續執行。(請參考本書 7-4-2 節 (Ex7_3.java) 範例介紹)

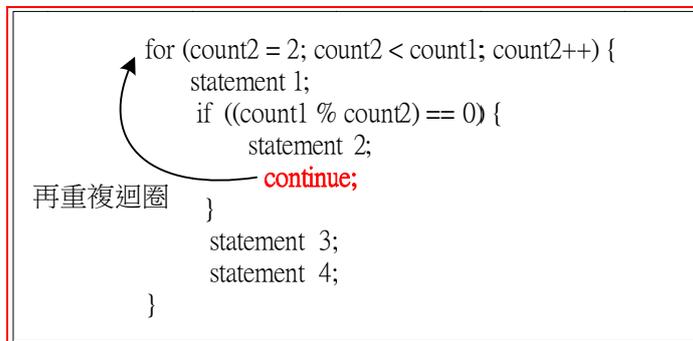


圖 5-13 continue 敘述的流程圖

5-6-3 範例研討：尋找質數程式

(A) 程式功能：Ex5_10.java

請編寫能找出 100 以內質數的程式，期望操作介面如下：

```

列印出 1~100 間的質數如下：
1  2  3  5  7  11  13  17  19  23
29 31  37  41  43  47  53  59  61  67
71 73  79  83  89  97
    
```

(B) 製作技巧研討：

『質數』表示僅能 1 和自己整除的數，即是不能被其他小於自己的數整除。譬如：2 僅能被 1 和 2 整除，其為質數，3、5、7 皆是如此。

而 4 可以被 2 整除，所以不是質數。由此可見 6、8、9、10...等偶數皆是如此，通通不是質數。

吾人編寫找出 1~100 之間(利用 while 迴圈達成)質數的方法是，利用一個旗號 prime 表示是否被測試數字是否為質數。首先假設被測試數字是質數，接著該數(如 7)再除以 2 到自己減 1(如 6，利用 while 迴圈達成)，其中任一個數可以整除則標示不是質數(prime=0)就離開；否則即為質數(prime=1)。離開迴圈後，利用 prime 內容判斷該數是否為質數，如是則印出。

(C) 程式範例：

```
01 //Ex5_10.java
02 /* 請編寫能找出100以內質數的程式 */
03
04 public class Ex5_10 {
05     public static void main(String args[]) {
06         final int limit = 100;      // 設定尋找最高數值
07
08         int prime;                  // 質數旗號
09         int i=1;
10         int count=0;
11         System.out.printf("列印出 1~100 間的質數如下：\n");
12         while (i <= limit) {
13             prime = 1;
14             int j = 2;
15             while (j <= i-1) {
16                 if ((i%j) == 0) {    // 判斷是否可以整除
17                     prime = 0;
18                     break;
19                 }
20                 j++;
21             }
22             if (prime == 1){
23                 System.out.printf("%d ", i);
24                 count = count + 1;
25                 if (count%10 == 0)    // 每 10 個數跳行
26                     System.out.printf("\n");
27             }
28         }
29     }
30 }
```

```

29         }
30         i++;
31     }
    }
}

```

(D) 程式重點說明：

- 第 7~22 行：『int i=1; while(i < limit) { i++;}』。外回圈功能是，指定質數尋找範圍，由 1 到 limit (=100) 之間。
- 第 10~17 行：『int j=2; while(j<=i-1) { ... j++;}』。內迴圈功能是所指定的數 (i)，除以由 2 到自己減一 (i-1)。
- 第 12~15 行：『if(i%j) == 0』 { ...}』。可以整除的話，則不是質數 (prime=0)，並中斷 (break) 往下繼續找。
- 第 18~19 行：『if (prime == 1) System.out.printf(...)]』。如果是該數值 (i) 是質數的話，則印出。

5-6-4 自我挑戰：心算過五關遊戲

(A) 程式功能：PM5_9.java

心算教室為了鼓舞小朋友學習心算的興趣，期望製作一套『心算過五關遊戲』，讓小朋友自行測試心算程度如何。系統中包含 5 關兩數相加，學生輸入正確表示通過該關。第一關為兩個 0~5 之間數字相加、第二關為 0~25 之間數字相加、第三關為 0~5*5*5 之間、第四關是 0~5*5*5*5、第五關是 0~5*5*5*5*5，期望操作介面如下：

情況一：通過五關

== 歡迎進入心算過五關評量系統 ==

第 1 關輸入 2 + 0 = 2

第 2 關輸入 17 + 16 = 33

第 3 關輸入 120 + 84 = 204

第 4 關輸入 194 + 321 = 515

第 5 關輸入 2477 + 1769 = 4246

恭喜您 !! 已通過五關測試

情況二：僅通過第二關

```
== 歡迎進入心算過五關評量系統 ==  
第 1 關輸入 1 + 3 = 4  
第 2 關輸入 24 + 9 = 33  
第 3 關輸入 117 + 90 = 205  
錯了!! 答案是 207  
您已完成第 2 關  
加油 !! 再來一次
```

(B) 製作技巧提示：

吾人利用虛擬碼提示程式重點，如下所示：

```
導入相關套件 ( import java.util.*; );  
....  
設定亂數基數 ( base = 1 );  
設定通過五關旗號 ( flag = 1 )  
for (count = 1; count <= 5; count++) {  
    累成亂數基數 ( base = base*5 )  
    產生第一個 0 ~ base 亂數 ( ran1 = random.nextInt(base) );  
    產生第二個亂數 ( ran2 );  
    計算正確答案 ( value = ran1 + value2 )  
    顯示兩亂數並要求輸入答案 ( ran1, ran2 );  
    讀入學生答案 ( answer = keyin.nextInt() );  
    if (answer != value)  
        顯示正確答案，及告知通過幾關 ( count-1 );  
        取消通過旗號 ( flag = 0 );  
        中斷程式 ( break );  
}  
if (flage == 1)  
    顯示 "恭喜通過五關測試";  
else  
    顯示 "加油 !! 再來一次";
```

5-7 專題製作

5-7-1 範例研討：列印月曆

(A) 系統功能：Ex5_11.java

請製作一只列印月曆的工具，使用者輸入當月 1 日星期幾(譬如 2，為星期二)與當月的天數(如 31 天)，則系統會列印該月的月曆，如下：

```
請輸入該月 1 日星期幾(1~7)=>2
請輸入該月的天數(28~31)=>31
  Mon Tue Wed Tue Fri Sat San
      1  2  3  4  5
  6  7  8  9 10 11 12
 13 14 15 16 17 18 19
 20 21 22 23 24 25 26
 27 28 29 30 31
```

(B) 製作技巧

請自行研討。

(C) 程式範例：

```
01 //Ex5_11.java
02
03 import java.util.Scanner;
04 public class Ex5_11 {
05     public static void main(String args[]) {
06         Scanner keyin = new Scanner(System.in);
07         int first, number, now=0, day;
08         System.out.printf("請輸入該月 1 日星期幾(1~7)=>");
09         first = keyin.nextInt();
10         System.out.printf("請輸入該月的天數(28~31)=>");
11         number = keyin.nextInt();
12         System.out.printf("\tMon\tTue\tWed\tTue\tFri\tSat\tSan\n");
13         for(int i=0; i<first; i++){
14             System.out.printf("\t");
15             now = now + 1;
16         }
17         day = 1;
18         while(day<=number){
```

```

20     System.out.printf("\t%d", day);
21     now = now + 1;
22     if (now%7 == 0){
23         System.out.printf("\n");
24         now = 0;
25     }
26     day = day + 1;
27 }
28 while(now < 7){
29     System.out.printf("\t");
30     now = now + 1;
31 }
32 }
33 }

```

5-7-2 自我挑戰：零存整付試算工具

(A) 程式功能：PM5_10.java

銀行預鼓勵民眾儲存現金，提供零存整付的儲蓄業務，顧客可選擇 6、9、12、24 個月，每月存入小額存款，每月計算利息，將利息加入本金，成為下個月的本金(即是複利計算)，到期後全部返還。銀行需要建構一套工具，讓客戶查詢到期後可領回多少錢。客戶只要輸入每個存入金額(如 10000)、選擇期數(如 12 個月)，以及當時年利率(如 1.2%)，則系統會顯示到期後可領回金額，如下：

```

*** 歡迎光臨零存整付試算系統 ***
預定每月儲存金額(如 10000)=>10000
存款期限(6,9,12,24)=>12
目前儲蓄年利率(如 0.15)=>0.2
到期可領回 130126 元
增加 10126 元利息

```

(B) 製作技巧與提示

一般利息皆採用年利率，每月利率為年利率/12，每月本金成長為(原本金 + 當月利息)，程式提示如下：

```

01     ....
02     for (int i=0; i<month; i++){
03         balance = balance + (int)((balance*rate)/12);

```

```

04         balance = balance + credit;
05     }
06     ....
07

```

5-7-3 自我挑戰：籌備退休金試算工具

(A) 系統功能：PM5_11.java

許多人為了籌備退休金而困擾，請您製作一套每月準備退休計算方式。假設 60 歲退休；到達平均年齡 80 歲 (20 年退休生活)，退休後生活品質分為每月 2、3、4、5 萬元。開始儲蓄退休金年紀分為 30、35、40、45 歲；暫不計算存款利息與通貨膨脹，以表格顯示方法，每個月應該儲備多少錢才可達成。試算系統操作介面如下：

```

*** 退休金試算表(請提早準備) ***
*** 假設 60 歲退休、活到 80 歲退休生活 ***
*退休後每月生活費、幾歲開始儲存、每月應存金額 *
      2萬  3萬  4萬  5萬
30 歲  1.33  2.00  2.67  3.33
35 歲  1.60  2.40  3.20  4.00
40 歲  2.00  3.00  4.00  5.00

```

5-7-4 自我挑戰：預存評估退休生活

(A)系統功能：PM5_12.java

我們可以倒過來評估退休後可過何種生活。系統要求輸入目前年紀與預備每月儲蓄多少錢，預計多少年紀退休，預估可活到 80 歲，退休後每月可以開銷多少，來評估爾後的生活，如下：

- (1) 5000 元以下：等待救濟。
- (2) 5000 ~ 10000 元：免強可以糊口。
- (3) 10,000 ~ 20,000：安逸舒適。
- (4) 20,000 ~ 40,000：富貴生活。
- (5) 40,000 以上：遨遊世界。

我們預計採用退休專案，它允許每月以零存整付的方式儲蓄，到退休年紀可領多少錢，期望系統操

作介面如下：

情況一：等待救濟生活

```
***    評估退休後生活品質系統    ***  
請輸入幾歲開始儲存退休金(如30)歲 =>30  
請問幾歲準備退休(如 60)歲 =>55  
請輸入預估年利率(如 0.14) =>0.15  
請輸入預估每月儲蓄(如 10000)元 =>2000  
到退休之前可儲存 615506 元  
平均每個月有 2052 元生活費  
退休生活品質：等待救濟。
```

情況二：富貴生活

```
***    評估退休後生活品質系統    ***  
請輸入幾歲開始儲存退休金(如30)歲 =>30  
請問幾歲準備退休(如 60)歲 =>65  
請輸入預估年利率(如 0.14) =>0.2  
請輸入預估每月儲蓄(如 10000)元 =>10000  
到退休之前可儲存 4371567 元  
平均每個月有 24286 元生活費  
退休生活品質：富貴生活。
```

(A)製作技巧與提示

一般利息皆採用年利率，每月利率為年利率/12，每月本金成長為(原本金 + 當月利息)，從開始到結束可儲存金額之程式提示如下：

```
01 ....  
02     month = (year_p - year_n)*12;  
03     balance = credit;  
04     for(int i=0; i<=month; i++){  
05         balance = balance + (balance*rate/12);  
06         balance = balance + credit;  
07     }  
08 ....
```