



第 3 章 選擇性敘述與迴圈

本章學習目標

- ✚ 認識程式的結構設計
- ✚ 學習各種迴圈的用法
- ✚ 學習選擇性敘述與多重選擇敘述的用法





5.1 程式的結構設計

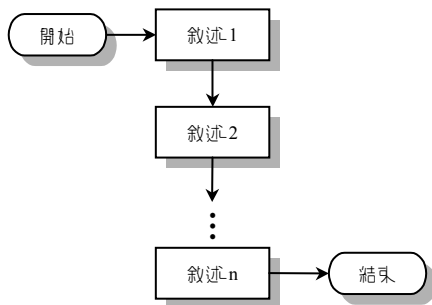
程式的結構包含有下面三種：

1. 循序性結構 (sequence structure)
2. 選擇性結構 (selection structure)
3. 重複性結構 (iteration structure)



5.1.1 循序性結構

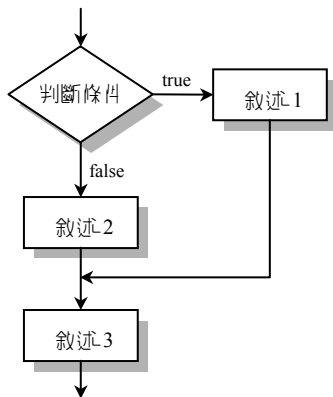
循序性結構的流程圖如下所示：





5.1.2 選擇性結構

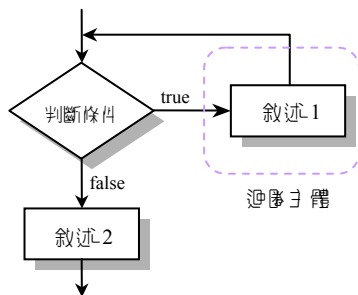
選擇性結構的流程圖如下：





5.1.3 重複性結構

重複性結構的流程圖如下圖所示：



Java 的重複性結構有 for、while 及 do while 三種迴圈。



5.2 我的程式會轉變--選擇性敘述

選擇性結構包括了 if、if-else 及 switch 敘述。

5.2.1 if 敘述

if 敘述的格式如下：

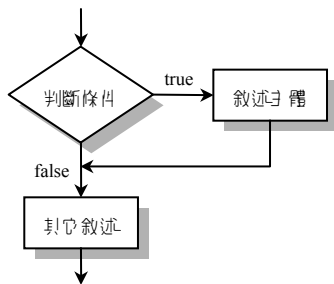
```
if (判斷條件)  
{  
    敘述語彙;  
}
```

格式 5.2.1

if 敘述的格式



if 敘述的流程圖如下圖所示：





5.2.2 if-else 敘述

if-else 敘述的格式如下：

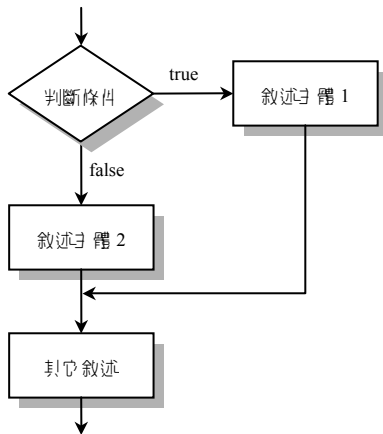
```
if (判斷條件)
{
    敘述語 體 1;
}
else
{
    敘述語 體 2;
}
```

格式 5.2.2

if-else 敘述的格式



if-else 敘述的流程圖如下所示：





下面是 if-else 的範例：

```
01 // app5_1,if-else 敘述
02 public class app5_1
03 {
04     public static void main(String args[])
05     {
06         int a=15;
07
08         if (a%2==0) // 如果可被 2 整除
09             System.out.println(a+" is an even number"); // 印出 a 為偶數
10         else
11             System.out.println(a+" is an odd number"); // 印出 a 為奇數
12     }
13 }
```

/* app5_1 OUTPUT---

15 is an odd number

-----*/



5.2.3 更多的選擇—巢狀 if 敘述

巢狀 if 敘述 (nested if) 的格式：

```
if (判斷條件 1)
```

```
{
```

```
    if (判斷條件 2)
```

```
    {
```

```
        敘述子體;
```

```
    }
```

```
    ...
```

```
    其它敘述;
```

```
}
```

} 若判斷條件 2
成立，則執行
這個部份

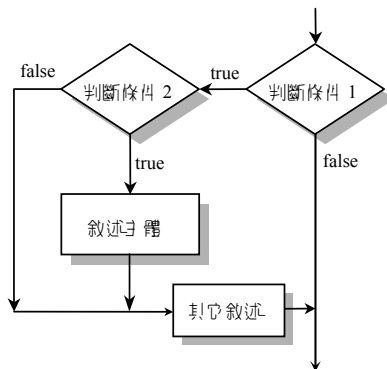
} 若判斷條件 1 成立，
則執行這個部份

格式 5.2.3

巢狀 if 敘述的格式



巢狀 if 敘述的流程圖：





5.2.4 if 的簡潔版--條件運算子

條件運算子（conditional operator）可以代替 if-else 敘述：

條件運算子	意義
?:	根據條件的成立與否，來決定結果 為?或:後的運算式

表 5.2.1

條件運算子的說明

條件運算子的格式如下：

傳回值 = 判斷條件 ? 運算式 1 : 運算式 2 ;

格式 5.2.4

?: 的敘述格式



試寫練習用條件運算子撰寫一程式：

```
01 // app5_2, 條件運算子 ? : 的使用
02 public class app5_2
03 {
04     public static void main(String args[])
05     {
06         int a=8,b=3,max;
07
08         max=(a>b)?a:b; // a>b時,max=a, 否則max=b
09
10         System.out.println("a="+a+", b="+b);
11         System.out.println(max+"是較大的數");
12     }
13 }
```

/* app5_2 OUTPUT----

a=8, b=3

8 是較大的數

-----*/



5.3 迴圈

需要重複執行某項功能時，迴圈就是最好的選擇。

5.3.1 for 迴圈

for 迴圈的敘述格式如下：

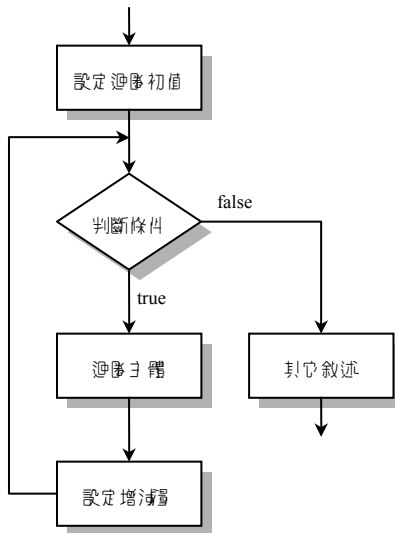
```
for (設定迴圈初值；判斷條件；設定增減量)
{
    迴圈主體；
}
這兒不可以加分號
```

格式 5.3.1

for 迴圈敘述格式



for 迴圈 流程圖：





程式 app5_3 是 for 迴圈的範例：

```
01 // app5_3, for 迴圈
02 public class app5_3
03 {
04     public static void main(String args[])
05     {
06         int i, sum=0;
07
08         for(i=1; i<=10; i++)
09             sum+=i; // 計算 sum=sum+i
10         System.out.println("1+2+...+10="+sum); // 印出結果
11     }
12 }
```

/* app5_3 OUTPUT---

1+2+...+10=55

-----*/



for 迴圈裡的區域變數

在迴圈裡宣告的變數只是區域變數，如下面的範例：

```
01 // app5_4, 區域變數
02 public class app5_4
03 {
04     public static void main(String args[])
05     {
06         int sum=0;
07
08         for(int i=1;i<=5;i++) // 在迴圈內宣告變數 i
09         {
10             sum=sum+i;
11             System.out.println("i="+i+", sum="+sum);
12         }
13     }
14 }
```

```
/* app5_4 OUTPUT---
```

```
i=1, sum=1
i=2, sum=3
i=3, sum=6
i=4, sum=10
i=5, sum=15
```

```
-----*/
```

} 變數 i 的有效範圍



5.3.2 while 迴圈

while 迴圈的格式如下：

```
設定迴圈初值；  
while (判斷條件) {  
    迴圈主體；  
    設定增減量；  
}
```

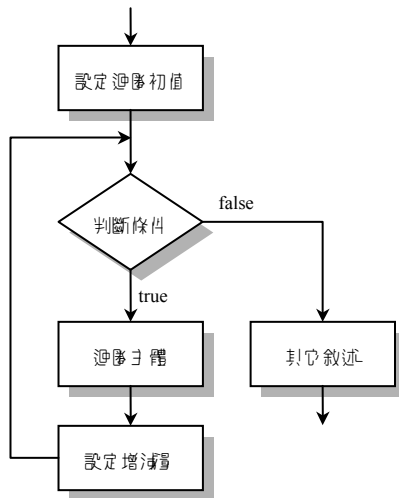
→ 這兒不可以加分號

格式 5.3.2

while 迴圈敘述格式



while 迴圈 流程圖：





下面的例子是利用 `while` 迴圈計算 1 累加到 10：

```
01 // app5_5,while 迴圈
02 public class app5_5
03 {
04     public static void main(String args[])
05     {
06         int i=1,sum=0;
07
08         while(i<=10)
09         {
10             sum+=i; // 累加計算
11             i++;
12         }
13         System.out.println("1+2+...+10="+sum); // 印出結果
14     }
15 }
```

/* app5_5 OUTPUT---

1+2+...+10=55

-----*/



5.3.3 do while 迴圈

do while 迴圈的格式如下：

設定迴圈初值；

do

{

 迴圈主體；

 設定增減量；

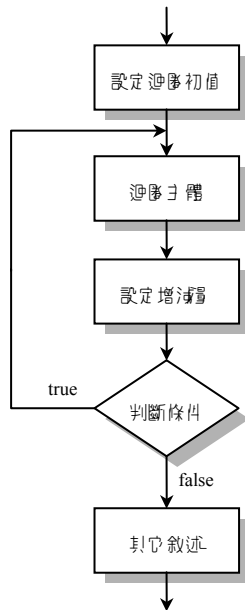
} **while** (判斷條件) ; → 要加分號

格式 5.3.3

do while 迴圈敘述格式



do while 迴圈 流程圖：





app5_6 是利用 do while 迴圈設計的程式：

```
01 // app5_6, do while 迴圈
02 import java.io.*;
03 public class app5_6
04 {
05     public static void main(String args[]) throws IOException
06     {
07         int n,i=1,sum=0;
08         String str;
09         BufferedReader buf;
10
11         buf=new BufferedReader(new InputStreamReader(System.in));
12
13         do{
14             System.out.print("請輸入累加的數大值：");
15             str=buf.readLine();
16             n=Integer.parseInt(str);
17         }while(n<1); // 輸入 n,n 要大於等於 1,否則會一直重複輸入
18
19         do
20             sum+=i++; // 計算 sum=sum+i,然後 i 值再加 1
21         while(i<=n);
22
```




```
23         System.out.println("1+2+...+"+n+"="+sum); // 印出結果
24     }
25 }
```

/* app5_6 OUTPUT-----

請輸入累加的最大值：**-8**

請輸入累加的最大值：**10**

1+2+...+10=55

-----*/



5.3.4 巢狀迴圈

當迴圈敘述中入有其它迴圈敘述時，就稱為巢狀迴圈：

```
01 // app5_7, 巢狀 for 迴圈求 9*9 乘法表
02 public class app5_7
03 {
04     public static void main(String args[])
05     {
06         int i,j;
07
08         for (i=1;i<=3;i++)          // 外層迴圈
09         {
10             for (j=1;j<=3;j++)      // 內層迴圈
11                 System.out.print(i+"*"+j+"="+ (i*j)+"\t");
12             System.out.println();
13         }
14     }
15 }
```

```
/* app5_7 OUTPUT ----
1*1=1   1*2=2   1*3=3
2*1=2   2*2=4   2*3=6
3*1=3   3*2=6   3*3=9
-----*/
```

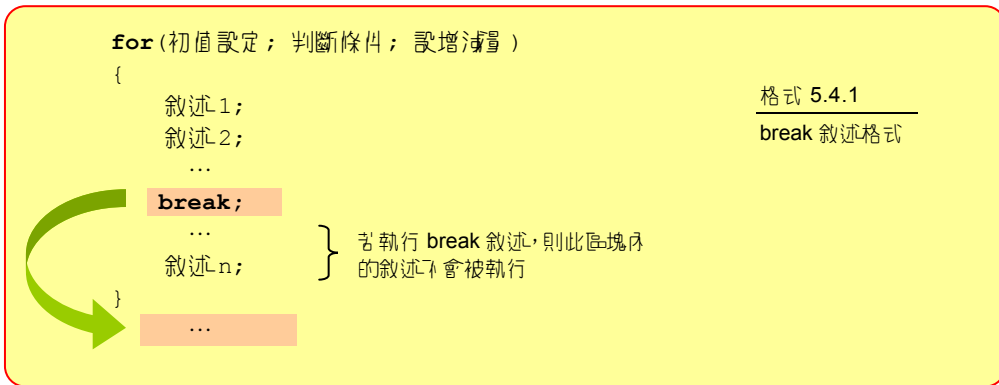


5.4 迴圈的跳離

5.4.1 break 敘述

程式執行到 **break** 時，即會離開迴圈，繼續執行迴圈外的下一個敘述。

下圖為 **for** 迴圈的 **break** 敘述格式：





app5_8 是利用 break 敘述跳離迴圈的範例：

```
01 // app5_8,break 的使用
02 public class app5_8
03 {
04     public static void main(String args[])
05     {
06         int i;
07
08         for (i=1;i<=10;i++)
09         {
10             if(i%3==0)    // 判斷 i%3 是否為 0
11                 break;
12             System.out.println("i="+i);    // 印出 i 的值
13         }
14         System.out.println("when loop interrupted,i="+i);
15     }
16 }
```

/* app5_8 OUTPUT-----

i=1

i=2

when loop interrupted,i=3

-----*/



5.4.2 continue 敘述

continue 敘述可以強迫程式跳到迴圈的起頭。

下圖為 for 迴圈的 continue 敘述格式：



```
for (初值設定; 判斷條件; 設增減量)
```

```
{  
    敘述1;  
    敘述2;  
    ...  
    continue;  
    ...  
    敘述n;  
}
```

} 若執行 **continue** 敘述，則此區塊內的敘述不會被執行

格式 5.4.2

continue 敘述格式



app5_9 是將 app5_8 中的 break 敘述改為 continue 敘述：

```
01 // app5_9, continue 的使用
02 public class app5_9
03 {
04     public static void main(String args[])
05     {
06         int i;
07
08         for (i=1;i<=10;i++)
09         {
10             if(i%3==0) // 判斷 i%3 是否為 0
11                 continue;
12             System.out.println("i="+i); // 印出 i 的值
13         }
14         System.out.println("when loop interrupted,i="+i);
15     }
16 }
```

```
/* app5_9 OUTPUT-----
i=1
i=2
i=4
i=5
i=7
i=8
i=10
when loop interrupted,i=11
-----*/
```



5.5 貝好貝的多重選擇--switch 敘述

switch 敘述的格式如下：

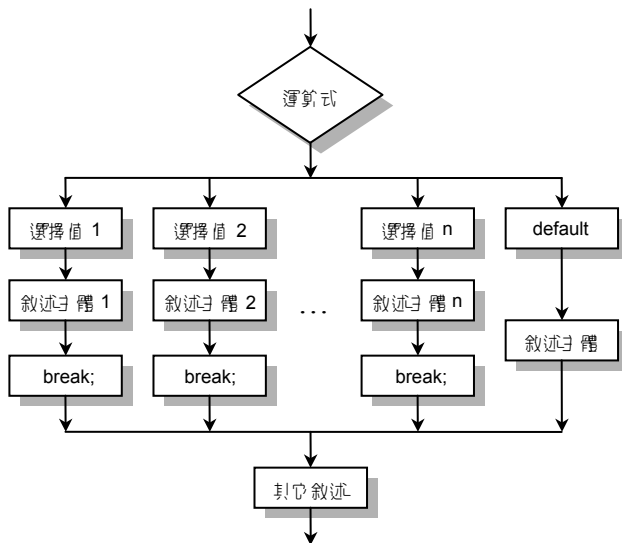
```
switch (運算式)
{
    case 選擇值 1:
        敘述句體 1;
        break;
    case 選擇值 2:
        敘述句體 2;
        break;
    ...
    case 選擇值 n:
        敘述句體 n;
        break;
    default:
        敘述句體;
}
```

格式 5.5.1

switch 敘述的格式



switch 敘述的流程圖如下：





下面的程式是利用 **switch** 敘述依據選擇值來進行簡單的四則運算：

```
01 // app5_10,switch敘述
02 public class app5_10
03 {
04     public static void main(String args[])
05     {
06         int a=100,b=7;
07         char oper='/';
08
09         switch (oper)
10         {
11             case '+':        // 印出 a+b
12                 System.out.println(a+"+"+b+"="+ (a+b));
13                 break;
14             case '-':        // 印出 a-b
15                 System.out.println(a+"-"+b+"="+ (a-b));
16                 break;
17             case '*':        // 印出 a*b
18                 System.out.println(a+"*"+b+"="+ (a*b));
19                 break;
20             case '/':        // 印出 a/b
21                 System.out.println(a+"/"+b+"="+ ((float) a/b));
22                 break;
```



```
23         default:      // 印出字串
24             System.out.println("Unknown expression!!");
25     }
26 }
27 }
```

/* app5_10 OUTPUT--

100/7=14.285714

-----*/



-The End-