

第四章

結構化程式設計的流程控制



4-5 選擇結構(Switch)

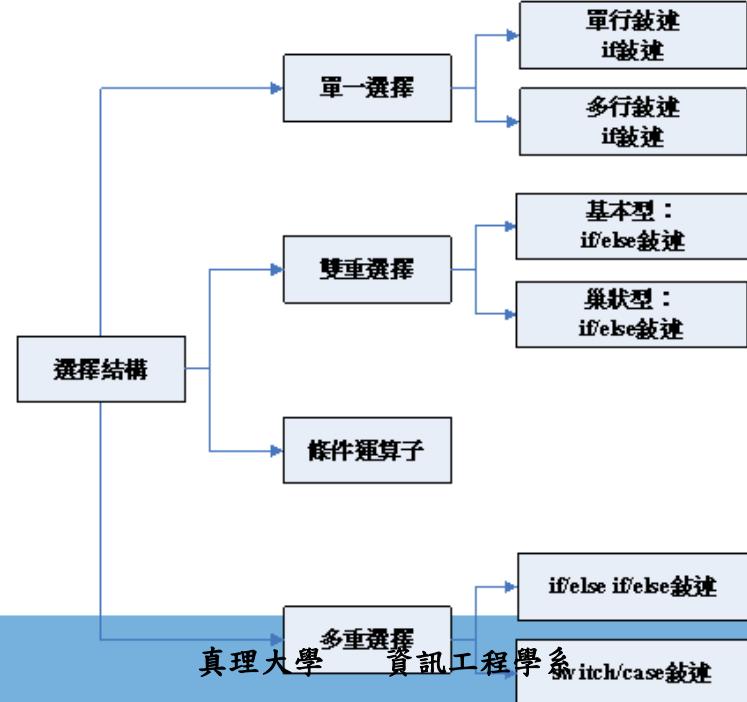


【定義】是指根據「條件式」來選擇不同的執行路徑。

【使用時機】

1. 只過濾某一種狀況。
2. 執行狀況有兩種或兩種以上。

【分類架構圖】



【例如】判斷及格與不及格、判斷奇數與偶數、判斷成績的等第...等情況，

【延伸學習】

- 一、機器人軌跡車(判斷白線或黑線)使用if/else
- 二、機器人讀顏色(針對不同的顏色，唸出不同的英文音)使用switch/case

【分類】「單一選擇結構」、「雙重選擇結構」與「多重選擇結構」



4-5.1 單一選擇結構(IF-Then)



【定義】

if的中文意思就是「如果...就...」。亦即只會執行「條件成立」時的敘述。

【示意圖】

如果...就...(單一選擇)	單一路徑的結構

【分類】

(一)單行敘述

【定義】指當條件式成立之後，所要執行的敘述式只有一行稱之。

【語法1】

```
if (條件式) 敘述;
```

【語法2】

```
if (條件式)
```

```
{
```

```
    敘述;
```

```
}
```

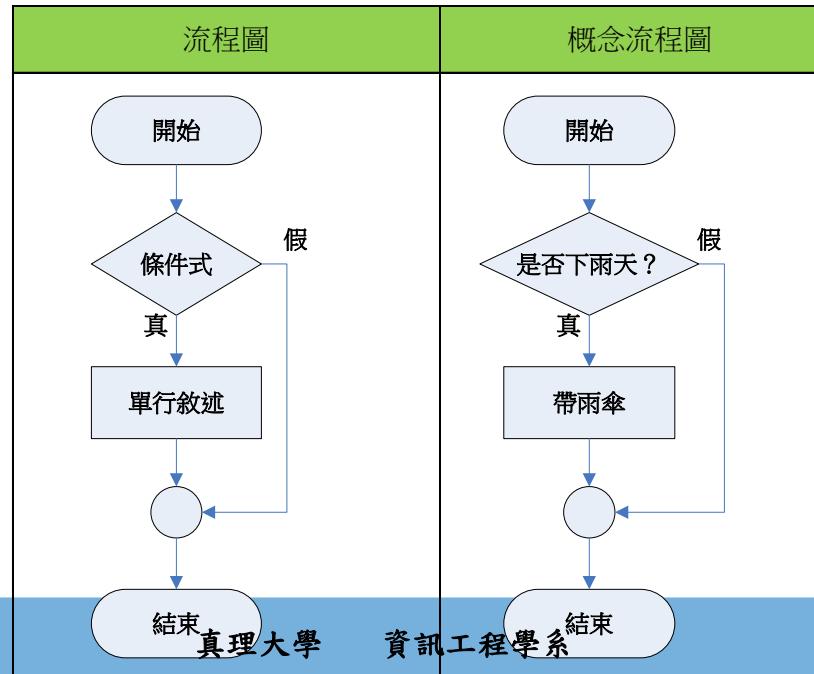
其中 <條件式> 是一關係運算式 或 邏輯運算式

【注意】單行敘述可省略上下兩個大括號。

- 【說明】
1. 以if為首的條件式必須放在()之內，之後的敘述放在它後面。
 2. 如果「條件式」成立(True)，就執行後面的「敘述」。
如果「條件式」不成立(False)，就跳過不執行。
 3. 若if條件成立時，希望執行的敘述不只有一行時，請用{}括起來。若只有一行，則可以省略。

【使用時機】當條件式成立之後，所要執行的敘述式只有一行。

【流程圖】



【實作1】請利用if分岔結構來判斷機器人的觸碰感測器是否被壓下，如果是的話，則前進。

【舉例】只對成績「及格」的學生有興趣。



行號	第一種寫法(成立敘述在後面) 程式碼：ch4_5_1A.java
01	<code>public class ch4_5_1A</code>
02	<code>{</code>
03	<code> public static void main(String[] args)</code>
04	<code> { //第一種方法</code>
05	<code> int Score=70;</code>
06	<code> if (Score >= 60) System.out.println("通過了");</code>
07	<code> }</code>
08	<code>}</code>

【說明】單行敘述可直接放在條件式後面。



行號	第二種寫法(成立敘述在{}之中)	程式碼：ch4_5_1B.java
01	<code>public class ch4_5_1B</code>	
02	<code>{</code>	
03	<code> public static void main(String[] args)</code>	
04	<code> { //第二種方法</code>	
05	<code> int Score=70;</code>	
06	<code> if (Score >= 60)</code>	
07	<code> {</code>	
08	<code> System.out.println("通過了");</code>	
09	<code> }</code>	
10	<code> }</code>	
11	<code>}</code>	

【說明】單行敘述可省略上下兩個大括號。因此，行號08與09可省略。

(二)多行敘述

【定義】指當條件式成立之後，所要執行的敘述式超過一行以上則稱之。

【語法】

```
if (條件式)
{
    敘述 1;
    敘述 2;
    .....
    敘述 n;
}
```

其中 (條件式) 是一關係運算式 或 邏輯運算式

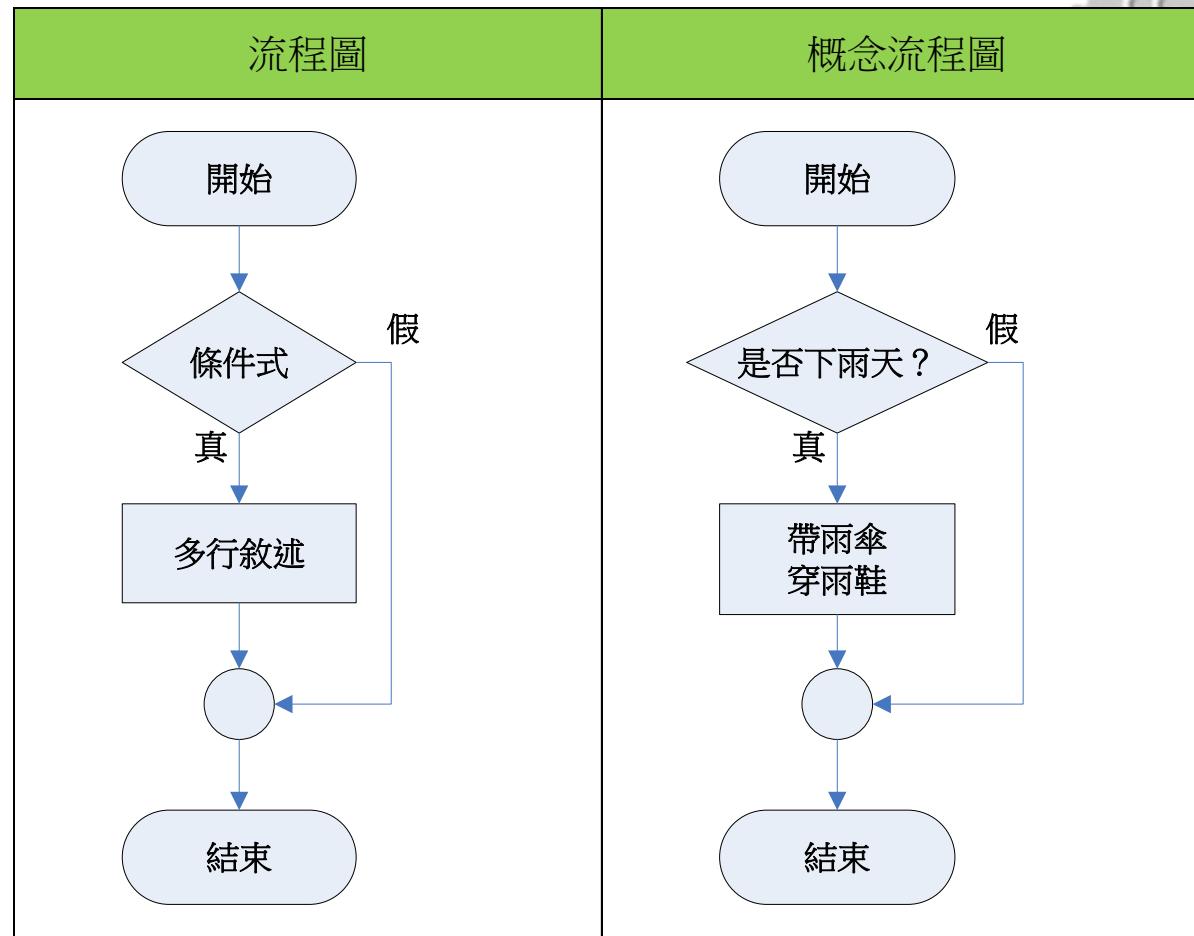
【注意】多行敘述不可以省略上下兩個大括號。

【說明】(1)如果「條件式」成立，就執行後面的「多行敘述」。

(2)若if條件式成立時，希望執行的敘述不只有一行時，請用{}
括起來。

【使用時機】當條件式成立之後，所要執行的敘述式是多行。

【流程圖】



【舉例】只對成績「不及格」的學生有興趣並且給予鼓勵。

題目：計算平均成績	程式檔案名稱	ch4_5_1C.java
01 public class ch4_5_1C 02 { 03 public static void main(String[] args) 04 { 05 //宣告 06 int Score=50; 07 //處理 08 if (Score <= 60) 09 {//輸出 10 System. out .println("您必須補考！"); 11 System. out .println("請再多加油哦！"); 12 System. out .println("希望補考能夠順利通過！"); 13 } 14 } 15 }		

【執行結果】

您必須補考！

請再多加油哦！

希望補考能夠順利通過！

4-5-2 雙重選擇結構



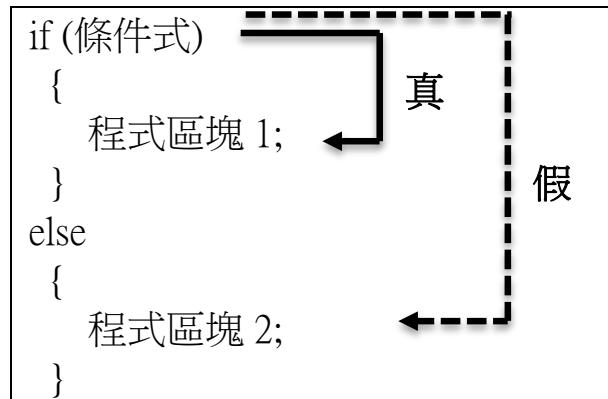
【定義】是指依照「條件式」成立以否，來執行不同的敘述。

【例如】判斷及格與不及格、判斷奇數與偶數、判斷男生與女生...等情況。

【示意圖】

如果...就...(雙重選擇)	雙向路徑的結構

【語法】



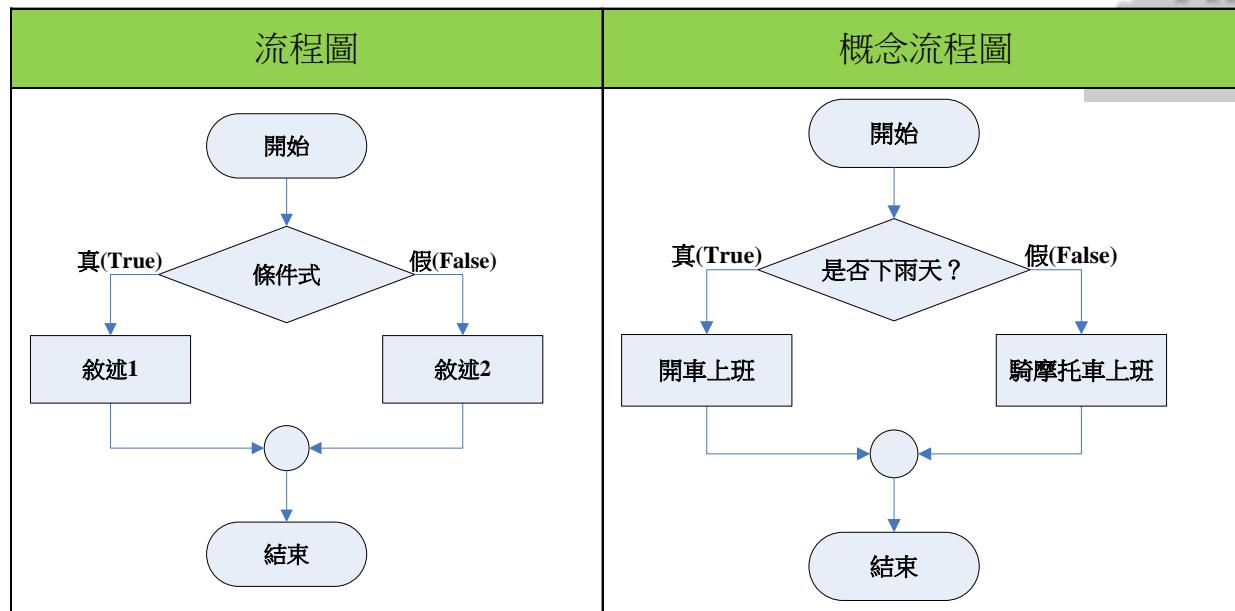
其中<條件式> 是一關係運算式 或 邏輯運算式

【說明】如果「條件式」成立(真) · 就執行後面的「敘述區塊1」 ·
否則就執行「敘述區塊2」 ·

【注意】當「敘述區塊」內的敘述只有一行時 · 則可以省略左右大括號 ·

【使用時機】當條件只有二種情況 ·

【流程圖】



【實作1】依照平均成績來判斷是「及格」與「不及格」兩種情況。

【分析】(1)輸入：國文及英文成績

(2)處理：計算平均分數，並依平均分數判斷「及格」或「不及格」。

(3)輸出：顯示兩科成績、平均成績及公佈結果。

【撰寫程式】



行號	程式檔名：ch4-5-2A.java
01	<code>import java.util.Scanner; //載入Scanner類別套件</code>
02	<code>public class ch4_5_2A</code>
03	<code>{</code>
04	<code> public static void main(String[] args)</code>
05	<code> { //輸入</code>
06	<code> Scanner inScore=new Scanner(System.in);</code>
07	<code> System.out.print("請輸入國文成績：");</code>
08	<code> String chiScore=inScore.next(); //國文成績</code>
09	<code> System.out.print("請輸入英文成績：");</code>
10	<code> String engScore=inScore.next(); //英文成績</code>
11	<code> //處理</code>
12	<code> int chi=Integer.parseInt(chiScore);</code>
13	<code> int eng=Integer.parseInt(engScore);</code>
14	<code> int average=(chi+eng)/2;</code>
15	<code> String result;</code>
16	<code> if (average >= 60)</code>
17	<code> result="及格";</code>
18	<code> else</code>
19	<code> result="不及格";</code>
20	<code> //輸出</code>
21	<code> System.out.println();//換行</code>
22	<code> System.out.println("國文：" + chi);</code>
23	<code> System.out.println("英文：" + eng);</code>
24	<code> System.out.println("平均：" + average);</code>
25	<code> System.out.println("結果：" + result);</code>
26	<code> }</code>
27	<code>}</code>

【執行結果】

請輸入國文成績：50

請輸入英文成績：60

國文：50

英文：60

平均：55

結果：不及格

【實作2】請設計一個判斷「奇數或偶數」的程式。

【分析】(1)輸入：某一個整數資料(num)

(2)處理：判斷此整數除以2，餘數是否為0

(3)輸出：當餘數為0時，則顯示「偶數」，否則顯示「奇數」。

【撰寫程式】

行號	程式檔名：ch4-5-2B.java
01	<code>import java.util.Scanner; //載入Scanner類別套件</code>
02	<code>public class ch4_5_2B</code>
03	<code>{</code>
04	<code> public static void main(String[] args)</code>
05	<code> { //輸入</code>
06	<code> Scanner inNumber=new Scanner(System.in);</code>
07	<code> System.out.print("請輸入某一個整數 : ");</code>
08	<code> String Number=inNumber.nextInt();</code>
09	<code> //處理</code>
10	<code> int num=Integer.parseInt(Number);</code>
11	<code> String result;</code>
12	<code> if ((num % 2) == 0)</code>
13	<code> result="偶數";</code>
14	<code> else</code>
15	<code> result="奇數";</code>
16	<code> //輸出</code>
17	<code> System.out.println();//換行</code>
18	<code> System.out.println("整數 : " + num + " 為 " + result);</code>
19	<code> }</code>
20	<code>}</code>

【執行結果】



請輸入某一個整數：10

整數：10 為 偶數

【實作3】：請設計可以求絕對值的程式。

【分析】 (1)輸入：某一個整數資料

(2)處理：判斷此整數是否大於0，當小於0時，則該數乘於負
1，否則就會直接顯示。

(3)輸出：正整數

【撰寫程式】

行號	程式檔名：ch4-5-2C.java
01	<code>import java.util.Scanner; //載入Scanner類別套件</code>
02	<code>public class ch4_5_2C</code>
03	<code>{</code>
04	<code> public static void main(String[] args)</code>
05	<code> { //輸入</code>
06	<code> Scanner inNumber=new Scanner(System.in);</code>
07	<code> System.out.print("請輸入某一個整數：");</code>
08	<code> String Number=inNumber.next();</code>
09	<code> //處理</code>
10	<code> int num=Integer.parseInt(Number);</code>
11	<code> if (num< 0)</code>
12	<code> num = (-1) * num;</code>
13	<code> //輸出</code>
14	<code> System.out.println(); //換行</code>
15	<code> System.out.println(" " + Number + " = " + num);</code>
16	<code> }</code>
17	<code>}</code>

【執行結果】

請輸入某一個整數：`-100`

`|-100|= 100`



4-5.3 條件運算子結構

【定義】是由三個運算子(=?:)組合而成的選擇結構。

【功能】與雙重選擇結構相同，都是使用在選擇結構的情況只有兩種。
因此，條件運算子結構也可以取代if/else的功能。

【語法】

```
變數名稱 = 條件式 ? 運算式1 : 運算式2
```

【說明】如果當條件式成立，會執行運算式1，否則執行運算式2。

【實例1】利用條件運算子結構，可以求某數的絕對值。

我們也可以將上面的程式(ch4-5-2C.java)改用條件運算子結構。

【分析】(1)輸入：某一個整數資料(num)

(2)處理：`result = num < 0 ? -num : num;`

(3)輸出：`result`

【撰寫程式】

行號	程式檔名：ch4-5-3A.java
01	<code>import java.util.Scanner; //載入Scanner類別套件</code>
02	<code>public class ch4_5_3A</code>
03	<code>{</code>
04	<code> public static void main(String[] args)</code>
05	<code> { //輸入</code>
06	<code> Scanner <u>inNumber</u>=new Scanner(System.in);</code>
07	<code> System.out.print("請輸入某一個整數：");</code>
08	<code> String Number=<u>inNumber</u>.next();</code>
09	<code> //處理</code>
10	<code> int num=Integer.parseInt(Number);</code>
11	<code> num= num<0 ? -num :num;</code>
12	<code> //輸出</code>
13	<code> System.out.println();//換行</code>
14	<code> System.out.println(" " + Number + " =" + num);</code>
15	<code> }</code>
16	<code>}</code>

【執行結果】

請輸入某一個整數：`-100`

`| -100 | = 100`

【實例2】利用條件運算子結構，求成績是否及格。

【分析】(1)輸入：某一個整數成績(score)

(2)處理：`result= score>=60 ? "及格" : "不及格";`

(3)輸出：result

【撰寫程式】

行號	程式檔名：ch4-5-3B.java
01	<code>import java.util.Scanner; //載入Scanner類別套件</code>
02	<code>public class ch4_5_3B</code>
03	<code>{</code>
04	<code> public static void main(String[] args)</code>
05	<code> { //輸入</code>
06	<code> Scanner inScore=new Scanner(System.in);</code>
07	<code> System.out.print("請輸入某一個整數：");</code>
08	<code> String Score=inScore.nextInt();</code>
09	<code> //處理</code>
10	<code> int intScore=Integer.parseInt(Score);</code>
11	<code> String result;</code>
12	<code> result= intScore>=60 ? "及格" : "不及格";</code>
13	<code> //輸出</code>
14	<code> System.out.println();//換行</code>
15	<code> System.out.println("結果：" + result);</code>
16	<code> }</code>
17	<code>}</code>

【執行結果】



請輸入某一個整數：60

結果：及格

4-5-4 多重選擇結構

【引言】

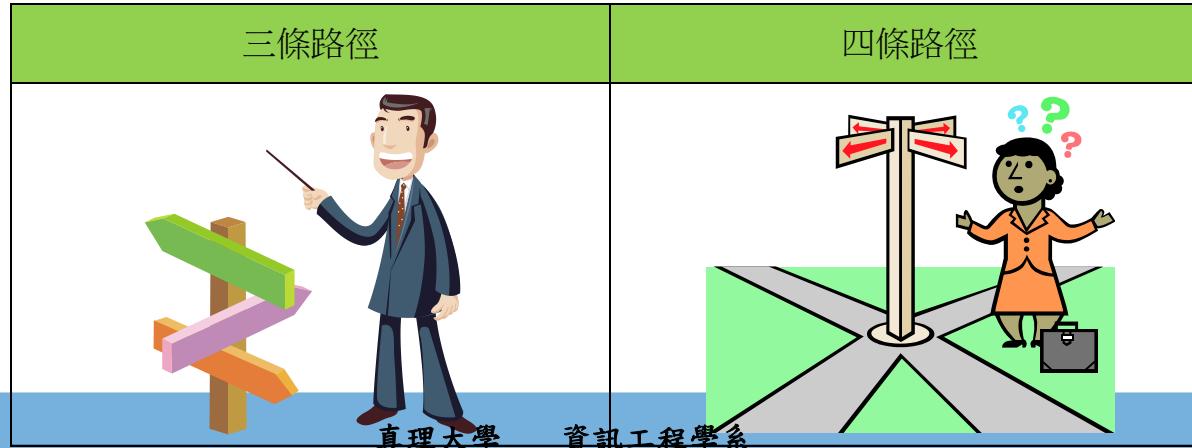
在日常生活中，我們所面臨的決策可能不只一種情況，也有可能兩種情況，甚至兩種以上不同情況。

【定義】是指「條件式」的情況是兩種以上時，稱之。

【例如】1. 國小成績分等第(優、甲、乙、丙...)。

2. 電影分級制(普通級、保護級、輔導級、限制級)。

【示意圖】





【作法】

- 1.逐一比對結構if/else if/else
- 2.逐一比對結構switch/case

4-5-4-1 逐一比對結構if/else if/else



【定義】此種結構是雙重結構的改良版，它可以使用於多種選擇情況。

【示意圖】

雙重結構	雙重結構的改良版(多種選擇結構)
	



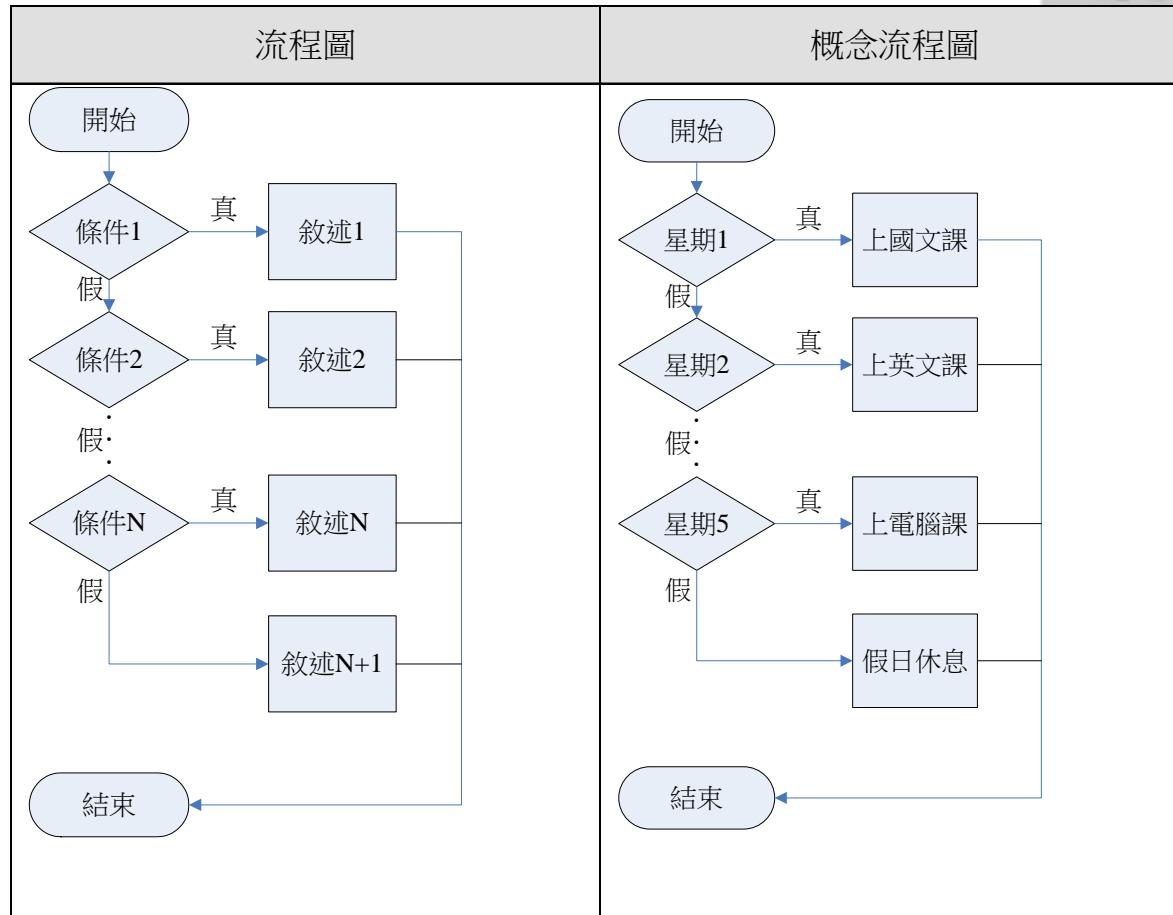
【語法】

```
if (條件式 1)
{
    敘述區塊 1
}
else if (條件式 2)
{
    敘述區塊 2
}
.
.
.
else if (條件式 n)
{
    敘述區塊 n
}
else
{
    敘述區塊 n+1
}
```

【說明】如果「條件式1」不成立，就繼續往下判斷「條件式2」，依樣
畫葫蘆的判斷下去，直到所有的條件式判斷完為止，否則就執
行「敘述區塊n+1」。

【使用時機】當條件式有兩種以上時。

【流程圖】



【實作】請計算國文(chi)與英文(eng)成績兩科的平均成績(average)，
並且依平均成績來分等第。



【分析】(1)輸入：國文(chi)與英文(eng)成績

(2)處理：計算平均成績(Average)並且依平均成績來分等第。

其條件如下：

- ① 60分以下為：E等
- ② 60~69分為：D等
- ③ 70~79分為：C等
- ④ 80~89分為：B等
- ⑤ 90(含)分以上為：A等

(3)輸出：平均成績與等第的結果

【撰寫程式】

行號	程式檔名：ch4-5-4-1.java
01	<code>import java.util.Scanner; //載入Scanner類別套件</code>
02	<code>public class ch4_5_4_1</code>
03	<code>{</code>
04	<code> public static void main(String[] args)</code>
05	<code> { //輸入</code>
06	<code> Scanner inScore=new Scanner(System.in);</code>
07	<code> System.out.print("請輸入國文成績：");</code>
08	<code> String chiScore=inScore.next(); //國文成績</code>
09	<code> System.out.print("請輸入英文成績：");</code>
10	<code> String engScore=inScore.next(); //英文成績</code>
11	<code> //處理</code>
12	<code> int chi=Integer.parseInt(chiScore);</code>
13	<code> int eng=Integer.parseInt(engScore);</code>
14	<code> int average=(chi+eng)/2;</code>
15	<code> String result;</code>
16	<code> if (average <60)</code>
17	<code> result="E等";</code>
18	<code> else if (average >= 60 && average < 70)</code>
19	<code> result="D等";</code>
20	<code> else if (average >= 70 && average < 80)</code>

```
21         result="C等";
22     else if (average >= 80 && average < 90)
23         result="B等";
24     else
25         result="A等";
26 //輸出
27     System.out.println(); //換行
28     System.out.println("國文：" + chi);
29     System.out.println("英文：" + eng);
30     System.out.println("平均：" + average);
31     System.out.println("結果：" + result);
32 }
33 }
```

【執行結果】

請輸入國文成績：80

請輸入英文成績：70

國文：80

英文：70

平均：75

結果：C等

4-5-4-2 逐一比對結構Switch Case



【引言】

if/else if/else與switch/case結構具有相同的功能，但如果條件很多時，使用if/else if/else結構就很容易混亂了。

【定義】是指使用於多種選擇情況。

【適用時機】當程式中的條件式超過兩個以上時。

【目的】程式較為精簡且可讀性較高。

【示意圖】

雙重結構	雙重結構的改良版(多種選擇結構)
	

【語法】

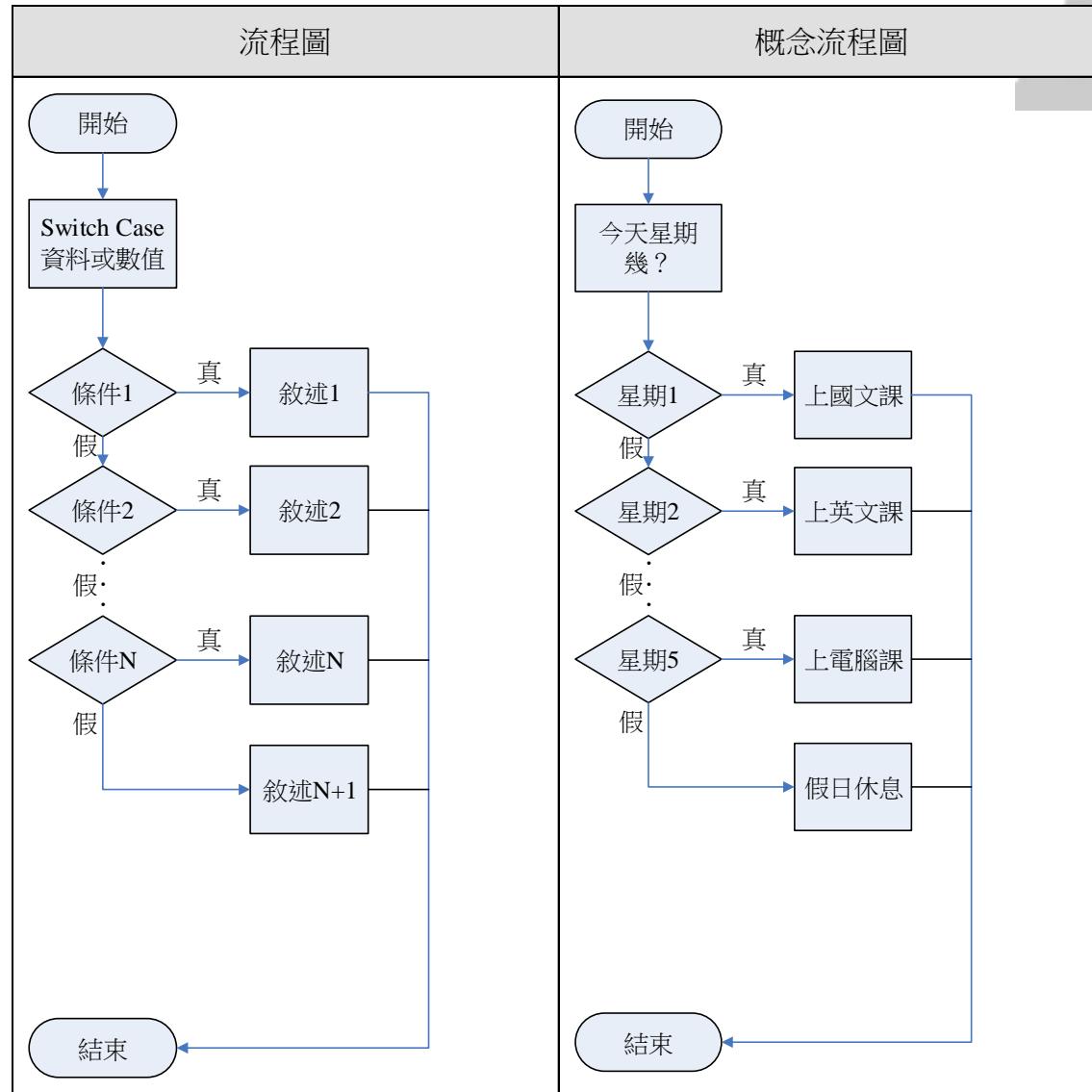


```
switch(資料或運算式)
{
    case 常數 1;
        敘述區塊 1;
        break;
    case 常數 2;
        敘述區塊 2;
        break;
    .....
    .....
    case 常數 N;
        敘述區塊 N;
        break;
    default:
        敘述區塊 N+1;
        break;
}
```

【說明1】當(資料或運算式)的值，符合常數1時，則執行敘述區塊1，符合常數2時，則執行敘述區塊2，直到遇到break敘述，才會離開switch，如果所有的條件式都不能符合時，則會執行敘述區塊N+1。

【說明2】switch後面的運算式可以是整數或字串資料。並且case後面的常數可以是整數或字串資料。

【流程圖】



【實作】承上一題，請利用switch/case結構，依平均成績來分等第。

行號	程式檔名：ch4-5-4-2.java
01	<code>import java.util.Scanner; //載入Scanner類別套件</code>
02	<code>public class ch4_5_4_2</code>
03	<code>{</code>
04	<code> public static void main(String[] args)</code>
05	<code> { //輸入</code>
06	<code> Scanner inScore=new Scanner(System.in);</code>
07	<code> System.out.print("請輸入國文成績：");</code>
08	<code> String chiScore=inScore.next(); //國文成績</code>
09	<code> System.out.print("請輸入英文成績：");</code>
10	<code> String engScore=inScore.next(); //英文成績</code>
11	<code> //處理</code>
12	<code> int chi=Integer.parseInt(chiScore);</code>
13	<code> int eng=Integer.parseInt(engScore);</code>
14	<code> int average=(chi+eng)/2;</code>
15	<code> String result;</code>
16	<code> switch ((int) (average/10))</code>
17	<code> {</code>
18	<code> case 10:</code>
19	<code> case 9:</code>
20	<code> result="A等";</code>
21	<code> break;</code>
22	<code> case 8:</code>
23	<code> result="B等";</code>
24	<code> break;</code>
25	<code> case 真理大學 資訊工程學系</code>

```
26             result="C等";
27             break;
28         case 6:
29             result="D等";
30             break;
31         default:
32             result="E等";
33             break;
34     }
35     //輸出
36     System.out.println(); //換行
37     System.out.println("國文：" + chi);
38     System.out.println("英文：" + eng);
39     System.out.println("平均：" + average);
40     System.out.println("結果：" + result);
41 }
42 }
```

【執行結果】

請輸入國文成績：80

請輸入英文成績：70

國文：80

英文：70

平均：75

結果：C等