

MultiMedia Systems Laboratory

CHAPTER 2



條件處理(ch5) 迴圈敘述(ch6) 陣列(ch7)





本節介紹

- 條件式的表示法
- 關係運算子
- if敘述
- 常見的if錯誤
- switch敘述
- 常見的switch錯誤
- switch進行文字條件判斷
- switch敘述與if敘述
- 邏輯運算子
- 條件運算子

關鍵詞彙

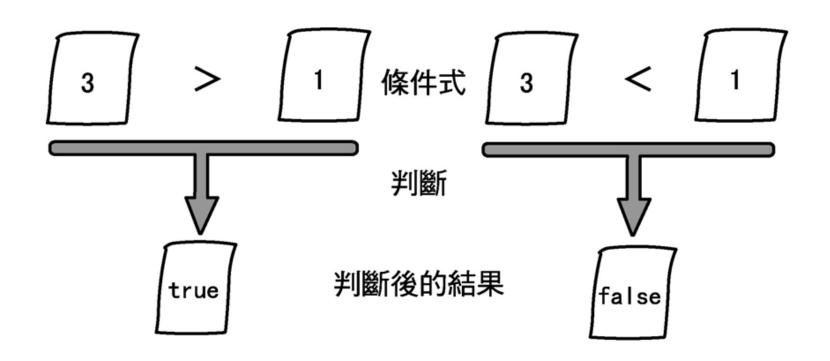
- ◆ 條件式
- ◆ 關係運算子
- ◆ if敘述
- ◆ switch敘述
- ◆ 邏輯運算子
- ◆ 條件運算子



■ 條件式的表示法

經過條件判斷最後得到true或false的程式,稱之為「條件式」。

- 「true」代表條件式的結果為真
- 「false」代表條件式的結果為假





■ 關係運算子(1/2)

為了要比較大小關係而使用的大於(>)、小於(<)、等於(==)等符號,被稱為「關係運算子」(relational operator)。

關係運算子	條件式的結果要等於true的條件		
==	左邊的值等於右邊的值		
!=	左邊的值不等於右邊的值		
>	左邊的值大於右邊的值		
>=	左邊的值大於或等於右邊的值		
<	左邊的值小於右邊的值		
<=	左邊的值小於或等於右邊的值		





■ 關係運算子(2/2)

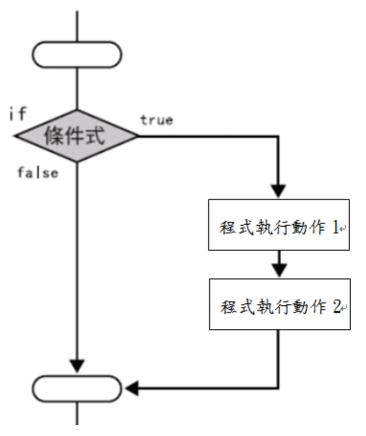
關係運算子	用途	範例	結果
==	等於	20 == 20	true
		15 == 13	false
!=	不等於	20 != 20	false
		15 != 13	true
>	大於	20 > 20	false
		15 > 13	true
<	小於	20 < 20	false
		15 < 13	false
>=	大於或等於	20 >= 20	true
		15 >= 13	true



≝ if敘述(1/2)

在if敘述中處理2行以上的的程式敘述句,必須在if敘述的內部用{}將所有的程式敘述句框住。

```
語法
if (條件式)
{
    程式執行動作1;
    程式執行動作2;
    ...
}
```



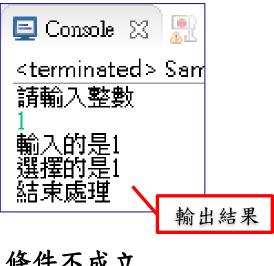




≝ if敘述(2/2)

```
🚺 Sample2_1.java 🖂
  1⊕ import java.io.BufferedReader;□
                                      讀取字元緩衝區裡的資料
  5
    public class Sample2_1
  6
  7⊜
      public static void main(String[] args) throws IOException
  8
  9
       System.outprintln("請輸入整數");
 10
 11
       BufferedReader br =
 12
          new BufferedReader(new InputStreamReader(System.in));
 13
 14
       int num = Integer.parseInt(br.readLine());
 15
 16
       if(num = = 1)
                                          使用者從鍵盤輸入的值,
 17
 18
         System.outprintln("輸入的是1");
                                              再指定給變數num
         System.outprintln("選擇的是1");
 20
 21
 22
       System.outprintln("結束處理");
 23
 24
                               當使用者輸入值為1時,會依序
 25
                                執行程式區塊內的所有程式碼
```

條件成立



條件不成立





🧧 常見的if錯誤

當 if 敘述內部有二行以上的程式敘述句時,但程式敘述句忘記加上{}時會發生:

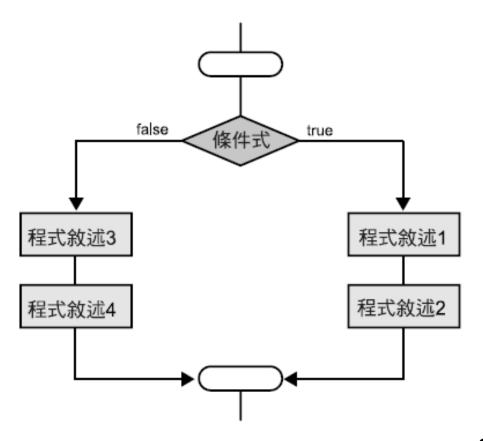
```
🚺 Sample2_2.java 💢
  1⊕ import java.io.BufferedReader;□
    public class Sample2_2
  6
      public static void main(String[] args) throws IOException
                                                                        📃 Console 💢
       System.outprintln("請輸入整數");
                                                                        <terminated> San
 10
 11
        BufferedReader br =
                                                                        請輸入整數
 12
          new BufferedReader(new InputStreamReader(System.in));
 13
                                                                        選擇的是1
 14
       int num = Integer.parseInt(br.readLine());
 15
                                                                        結束處理
                                         只有這個敘述句會
 16
       if(num = = 1)
 17
         System.outprintln("輸入的是1");
                                         變成if敘述的內容
                                                                                        輸出結果
 18
         System.outprintln("選擇的是1");
 19
 20
                                         這個敘述句變成if
 21
        System.out.println("結束處理");
                                          敘述之外的處理
 22
 23
 24
```



≝ if ~ else敘述(1/2)

使用if~else敘述時,條件式的結果不管是true還是false,都有各自不同的程式執行方向。內部的程式區塊也可以撰寫多行程式碼。

```
語法
  if (條件式) {
   程式敘述1;
   程式敘述2;
  else
   程式敘述3;
   程式敘述4;
```

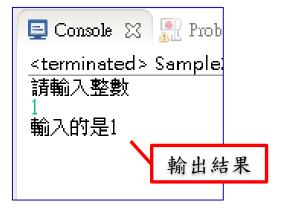




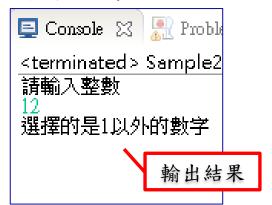
≝ if ~ else敘述(2/2)

```
🚺 Sample2_3.java. 🖂
  1⊕ import java.io.BufferedReader;□
  4
     public class Sample2_3
  6
      public static void main(String[] args) throws IOException
        System.outprintln("請輸入整數");
 10
 11
        BufferedReader br =
 12
           new BufferedReader(new InputStreamReader(System.in));
 13
 14
        int num = Integer.parseInt(br.readLine());
 15
 16
        if(num = = 1)
 17
 18
          System.out.println("輸入的是1"); a
                                             當使用者輸入值為1
 19
                                                   會執行此區塊
 20
        else
 21
 22
23
          System.outprintln("選擇的是1以外的數字");
 24
                                          當使用者輸入值1以外的
 25
                                          數字時,會執行此區塊
```

條件成立



條件不成立

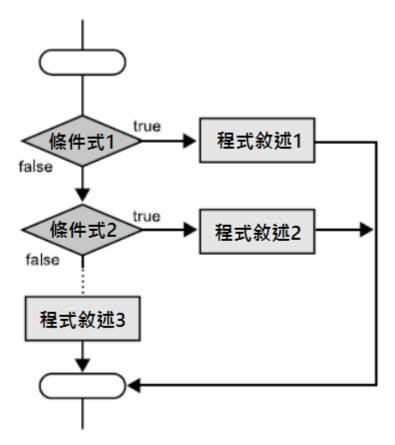




if ~ else if ~ else敘述(1/2)

之前介紹的if敘述都只有一個條件式,如果要用到多個條件是就約透過if~else if~else敘述達成。

```
語法
  if (條件式1) {
    程式敘述1;
  else if (條件式2) {
    程式敘述2;
  else {
    程式敘述3;
```





≝ if~else if~else敘述(2/2)

```
🚺 Sample2_4.java. 🖂
  1⊕ import java.io.BufferedReader;□
     public class Sample 24
  6
  70
      public static void main(String[] args) throws IOException
  8
  9
        System.out.println("請輸入整數");
 10
 11
        BufferedReader br =
 12
           new BufferedReader(new InputStreamReader(System.in));
 13
 14
        int num = Integer.parseInt(br.readLine());
 15
                                           使用者輸入1時,
 16
        if(num == 1)
 17
                                             會執行此區塊
 18
         System.out.println("輸入的是1");
 19
 20
        else if (num == 2)
                                          使用者輸入2時,
 21
 22
         System.out.println("輸入的是2");
                                             會執行此區塊
 23
 24
        else
 25
                                        使用者輸入1或2以外的
 26
         System.out.println("請輸入1或2");
 27
                                        數字時,會執行此區塊
 28
 29
```



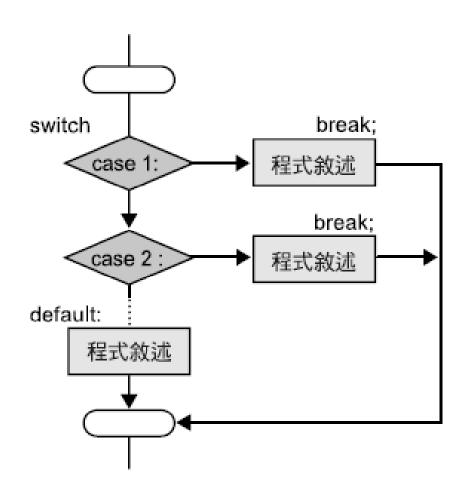


≤ switch敘述(1/2)

要使用多重條件的敘述時,我們也能使用switch敘述。

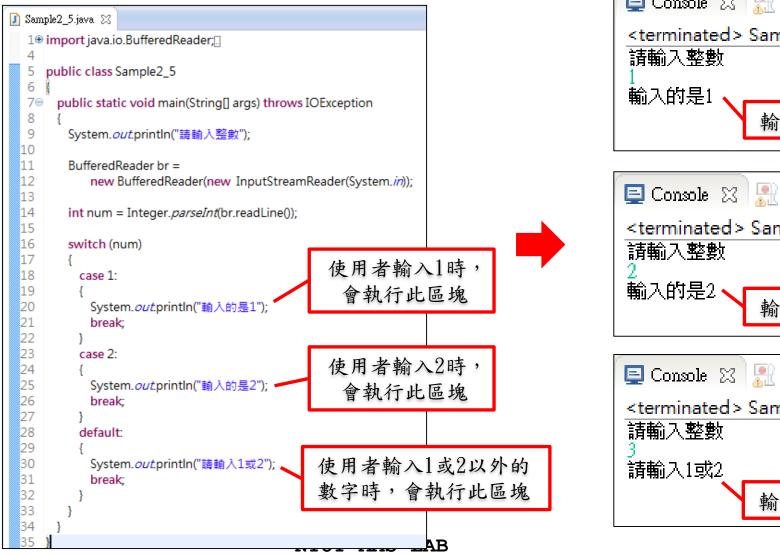
語法

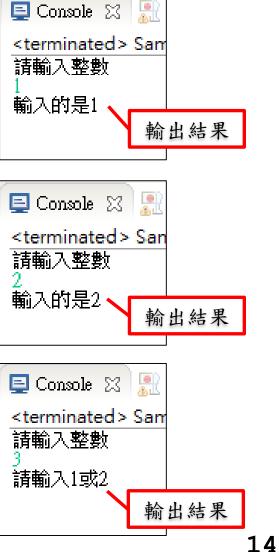
```
switch (運算式) {
 case 值1:
   程式敘述1;
   break;
 case 值2:
   程式敘述2;
   break;
 default:
   程式敘述d;
   break;
```





🧧 switch敘述(2/2)



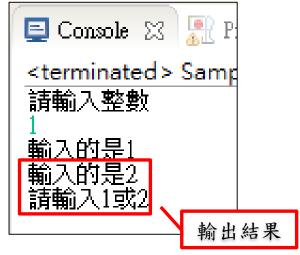




🧧 常見的switch錯誤

使用switch 敘述時,忘記加入break會發生:

```
🚺 Sample2_5.java 🖂
  1⊕ import java.io.BufferedReader; ...
     public class Sample 25
  6
      public static void main(String[] args) throws IOException
  8
        System.out.println("請輸入整數");
 11
        BufferedReader br =
 12
           new BufferedReader(new InputStreamReader(System.in));
 13
 14
        int num = Integer.parseInt(br.readLine());
 15
 16
        switch (num)
 17
 18
          case 1:
 19
 20
           System.out.println("輸入的是1");
 21
 22
          case 2:
 23
 24
           System.out.println("輸入的是2");
                                                  若未加break,則程式
 25
 26
          default:
                                                 會繼續往下執行,例如
 27
 28
           System.out.println("請輸入1或2");
                                                  跑完casel就會繼續往
 29
                                                 下跑case2和default。
 30
 31
```





Switch進行文字條件判斷

```
    Sample2_6.java 

    Sample2    Sam
            1⊕ import java.io.BufferedReader;□
                         public class Sample2_6
             6
                                public static void main(String[] args) throws IOException
             8
             9
                                        System.out.println("請輸入a或b");
        10
       11
                                         BufferedReader br =
       12
                                                      new BufferedReader(new InputStreamReader(System.in));
       13
        14
                                       String str = br.readLine();
                                                                                                                                                                                    使用String類別的
       15
                                        char letter = str.charAt(0):
                                                                                                                                                                                     charAt類別方法來取
       16
       17
                                                                                                                                                                                      出索引值為0的字元,
                                        switch (letter)
       18
                                                                                                                                                                                     在這裡表示取輸入的
        19
                                               case 'a':
                                                                                                                                                                                      第一個字母。
       20
       21
                                                      System.out.println("輸入的是a")
        22
                                                       break;
        23
        24
                                               case 'b':
       25
       26
                                                      System.out.println("輸入的是b");
        27
                                                       break:
        28
       29
                                               default:
       30
       31
                                                      System.out.println("請輸入a或b");
        32
                                                       break;
       33
       34
       35
                                                                                                                                                                                                                                                                                       B
        36
```





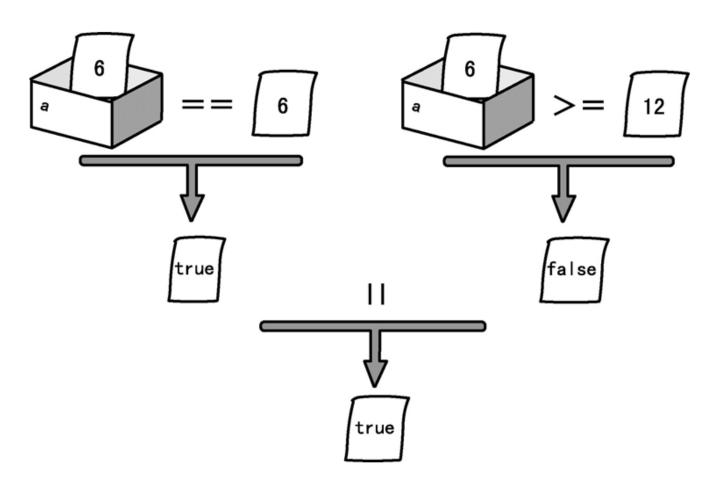
≤ switch敘述與if敘述

- ✓ if敘述和switch敘述都是讓程式流程做選擇的敘述,故合稱為選擇敘述 (selection statement)。
- ✓ switch敘述較具整體性,故使用單一運算式之值讓程式流程 做選擇的時候,選用switch敘述會比較好。
- ✓switch敘述可讀性較高。



■ 邏輯運算子(1/2)

邏輯運算子&&(and)、||(or)、!(not)可以用來設定更複雜的條件判斷。





≦ 邏輯運算子(2/2)

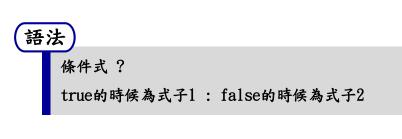
```
請問你是男生嗎?
🚺 Sample2 7.java 🖂
                                                                           請輸入Y或N
  1⊕ import java.io.BufferedReader;□
  4
                                                                           你是男生阿
    public class Sample2_7
  6
      public static void main(String[] args) throws IOException
  9
       System.out.println("請問你是男生嗎?");
 10
       System.out.println("請輸入Y或N");
                                                                          📮 Console 💢
 11
       BufferedReader br =
 12
          new BufferedReader(new InputStreamReader(System.in));
 13
 14
       String str = br.readLine();
                                                                           請輸入Y或N
 15
       char letter = str.charAt(0);
                                        使用者輸入Y或y時,
 16
                                                                           你是女生阿!
 17
       if ( letter == 'Y' || letter == 'y' )
                                           會執行此區塊
 18
 19
         System.out.println("你是男生阿!");
 20
                                        使用者輸入N或n時,
 21
       else if ( letter = = 'N' || letter = = 'n' )
 22
                                                                          📮 Console 💢
                                           會執行此區塊
 23
         System.out.println("你是女生阿!");
 24
 25
       else
 26
                                        使用者輸入Y、y、
                                                                           諳輸入Y或N
 27
         System.out.println("請輸入Y或N")
                                        N 或n之外的字元時,
 28
                                                                           請輸入Y或N
 29
                                            會執行此區塊
 30
```

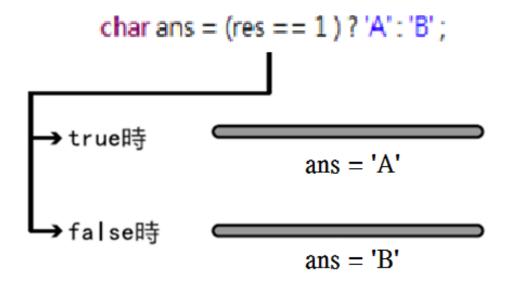




■ 條件運算子(1/2)

簡單的條件運算可以使用條件運算子(?:),條件運算子會根據條件式當中設定的值(此處是num==1)判斷結果到底是true還是false,再決定應該執行哪一個程式碼。







■ 條件運算子(2/2)

```
🚺 Sample2 8.java 🖂
                                                                          📃 Console 💢 🔝 Pro
  1⊕ import java.io.BufferedReader;□
                                                                          <terminated > Sample
                                                                          請問要選哪條路線?
     public class Sample2_8
                                                                          請輸入整數
  6
       public static void main(String[] args) throws IOException
                                                                          選擇了A路線
        System.out.println("請問要選哪條路線?");
                                                                                            輸出結果
        System.out.println("請輸入整數");
 10
 11
        BufferedReader br =
 12
            new BufferedReader(new InputStreamReader(System.in));
 13
 14
        String str = br.readLine();
                                                                          📃 Console 💢 🧦 Pro
 15
        int res =Integer.parseInt(str);
 16
                                                                          <terminated > Sample
 17
        char ans = (res == 1)? 'A': 'B';
                                                                          請問要選哪條路線?
                                               此寫法相當於
 18
                                                                          請輸入整數
                                                 if (res == 1)
 19
        System.out.println("選擇了" + ans + "路線");
                                                   ans ='A';
 20
                                                                          選擇了B路線
 21
                                                 else
                                                                                            輸出結果
                                                   ans = 'B';
```





本節介紹

- for迴圈敘述
- while迴圈敘述
- do~while迴圈敘述
- 巢狀迴圈
- break
- continue

關鍵詞彙

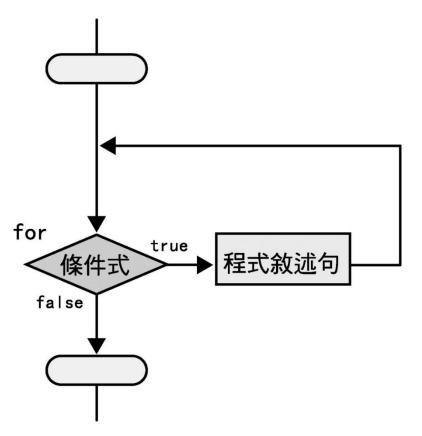
- ◆ 迴圈
- **♦** for
- **♦** while
- ♦ do ~ while
- **♦** break
- **♦** Continue
- ◆ 巢狀迴圈



for迴圈敘述(1/2)

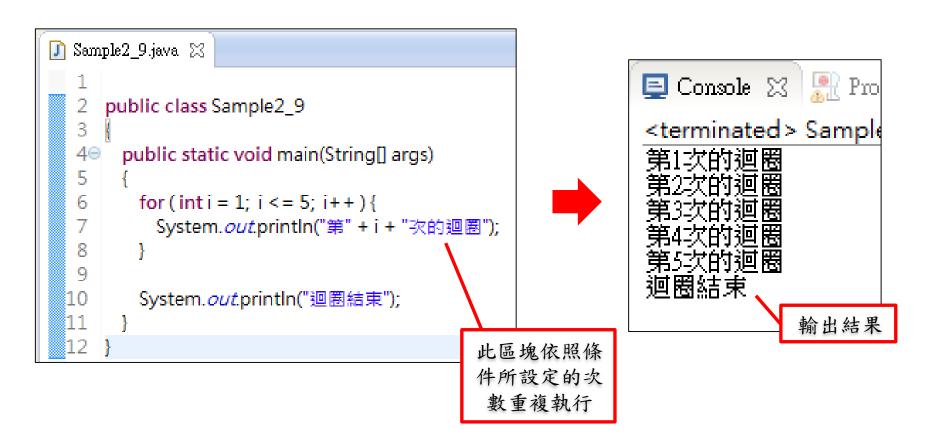
Java中有專門處理重複事件的指令,這些指令稱為迴圈敘述,首先我們來介紹for敘述。

```
語法
for(起始值; 判斷是否要重複執行的條件式; 遞增或遞減運算){
程式敘述句;
...
}
```





for迴圈敘述(2/2)



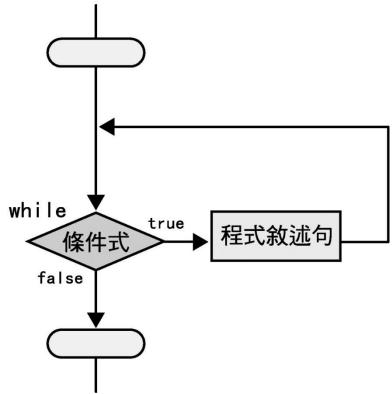




■ while 迴圈敘述(1/2)

只要while敘述後面的條件視為真(true),那while敘述下方程式區塊的內容就會不斷執行。

```
語法
while (條件式) {
    程式敘述;
    ...
}
```





≤ while 迴圈敘述(2/2)

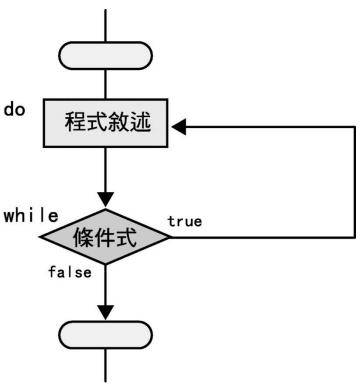




≝ do~while迴圈敘述(1/2)

- ✓ do...while迴圈敘述式類似while迴圈敘述式
- ✓ while迴圈敘述式,程式敘述執行之前就會檢查條件式,條件成立才執行
- ✓ do...while 迴圈敘述式,先執行程式敘述後,再檢查條件式,因此do...while 迴圈本體至少會執行一次

```
語法
do {
    程式敘述1;
    ...
}while (條件式);
```





≝ do ~ while迴圈敘述(2/2)

```
Sample2_11.java ⊠
                                                          📮 Console 💢 -
    public class Sample2_11
                                                           <terminated > Sample
     public static void main(String[] args)
       int i = 1:
       do {
        System.out.println("第" + i + "次的迴圈");
        i++;
10
       \} while (i <= 5);
                                                           迴圈結束
11
12
       System.out.println("迴圈結束");
                                                                           輸出結果
13
                                     此區塊依照條
14
                                     件所設定的次
                                      數重複執行
```



■ 巢狀迴圏(1/2)

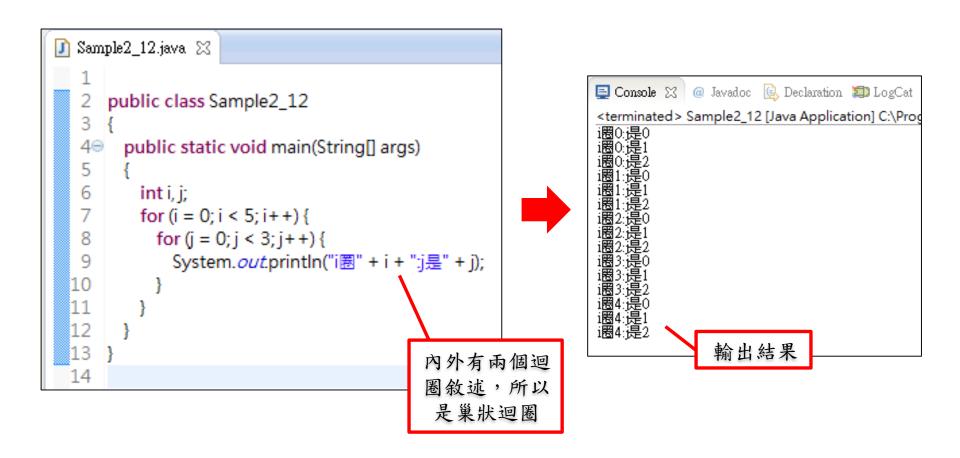
如果迴圈之內還有其他迴圈,就稱為「<mark>巢狀迴圈</mark>」。就像以下的範例當中, for迴圈內部還有另一個for迴圈就屬於巢狀迴圈的一種。

語法

```
for (起始值1;判斷是否要重複執行的條件式1;遞增或遞減運算1) {
...
for (起始值2; 判斷是否要重複執行的條件式2; 遞增或遞減運算2) {
...
}
```



■ 巢狀迴圏(2/2)





break敘述(1/2)

使用break指令,可強制程式從目前執行中的迴圈跳離。



```
for(int i=1; i<=10; i++){
    if(i==res)
        break;
}
```



break敘述(2/2)

```
☑ Sample2_13.java 
☒
   1⊕ import java.io.BufferedReader;□
   5
      public class Sample2_13
  7
        public static void main(String[] args) throws IOException
  9
 10
         System.out.println("請問要在第幾次處理結束迴圈呢?(1~10)");
 11
         BufferedReader br = new
 12
             BufferedReader(new InputStreamReader(System.in));
 13
 14
         String str = br.readLine();
 15
         int res = Integer.parseInt(str);
 16
 17
         for (int i = 1; i < = 10; i++) {
 18
           System.out.println("第" + i + "次的處理。");
 19
           if(i == res)
 20
             break;
 21
 22
                                   條件成立即
 23
                                     結束迴圈
```



輸出結果



■ continue敘述(1/2)

continue指令可以「讓目前執行中的迴圈暫時停住不往下執行,而是回到迴圈繼續下一個執行」。



```
for(int i=1; i<=10; i++){

if(i==res)

continue;
}
```



continue敘述(2/2)

```
Console 🖂 🔝 Problems

■ Sample2_14.java 

□

                                                                            <terminated > Sample2_14 [Jav
  1⊕ import java.io.BufferedReader,□
                                                                            要跳過第幾次的處理呢?(1~10)
  4
                                                                            第1次的處理。
     public class Sample2_14
                                                                            第4次的處理。
  7
  80
       public static void main(String[] args) throws IOException
  9
 10
        System.out.println("要跳過第幾次的處理呢?(1~10)");
                                                                                                     輸出結果
                                                                            第9次的處理。
 11
        BufferedReader br = new
                                                                            第10次的處理。
 12
            BufferedReader(new InputStreamReader(System.in));
 13
 14
        String str = br.readLine();
 15
        int res = Integer.parseInt(str);
 16
                                                                           Console 🔀 🧖 Problems @
                                      條件成立則暫停
 17
        for (int i = 1; i < = 10; i++) {
                                                                            <terminated > Sample2_14 [Jav
                                       目前動作,繼續
 18
                                                                            要跳過第幾次的處理呢?(1~10)
 19
          if(i == res)
                                        做下一個迴圈
 20
                                                                            第1次的處理。
            continue;
 21
22
          System.out.println("第" + i + "次的處理。");
 23
24
 25
                                                                                                     輸出結果
```





本節介紹

- 陣列
- 陣列的好處
- 陣列宣告
- 另一種陣列宣告
- 常見的陣列索引值錯誤
- 陣列初始值設定
- 陣列的應用
- 陣列的排序
- Java提供的排序法
- 多維陣列

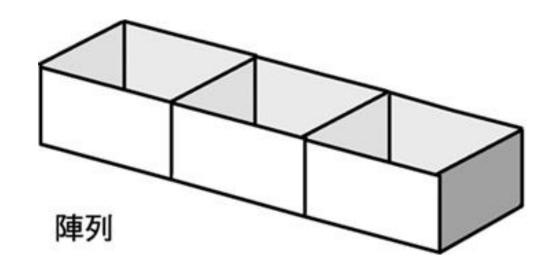
關鍵詞彙

- ◆ 陣列
- ◆ 宣告陣列
- ◆ 索引值
- ◆ 排序
- ◆ 多維陣列
- ◆ 陣列長度



■ 陣列

- ✓ 陣列是將「相同型態的多個值存放在相同名稱的記憶體位置」
- ✓ 陣列具有「一次可以存放大量性質相同的資料」的特性,因此,使用陣列可以「免除大量變數命名的問題」,使得程式具有較高的可讀性





🧧 陣列的好處

- ✓ 問題:宣告50筆的測試資料
 - 如果沒有使用陣列需要宣告50個變數,才能使用50筆資料。

• 使用陣列,只需要宣告1個宣告,即可使用50筆資料程式碼簡單且把聚集相同資料。

```
int test[50];  // test[0]代表第1筆的測試資料  // test[1]代表第2筆的測試資料
```



■ 陣列宣告(1/2)

```
語法
資料類型 陣列變數名稱[];
陣列變數名稱 = new 資料類型[元素的個數];
```

- ✓ 宣告使用一維陣列
 - int test [50]; //50筆測試資料(test), 型態為整數(int)
 - double test_avg[50]; // 50筆測試資料的平均(test_avg),型態為浮點數
- ✓ 每個陣列中的第一個元素皆是第零個元素 (zeroth element)
 - test[0] = 80; //在test [50] 陣列中, 測試資料0為80
 - test[1] = 60; //在test [50] 陣列中,測試資料1為60
 - test[20] = 50; //在test [50] 陣列中, 測試資料20為50

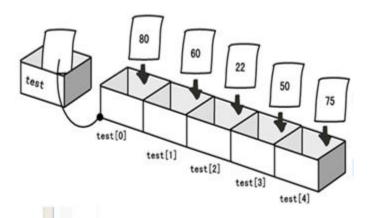


■ 陣列宣告(2/2)

```
public class Sample2_15
   3
  4⊝
       public static void main(String[] args)
   5
   6
       ① int [] test;
       ② test = new int[5];
         test[0] = 80;
                                  陣列的初始化,
         test[1] = 60;
                                      可改寫成
         test[2] = 22;
         test[3] = 50;
                                  int [] test = new int [5];
         test[4] = 75;
 14
15
         for (int i = 0; i < 5; i++)
 16
 17
           System.out.println("第" + (i+1) + "個人的分數是"+ test[i] + "分");
 18
 19
 20
21
```



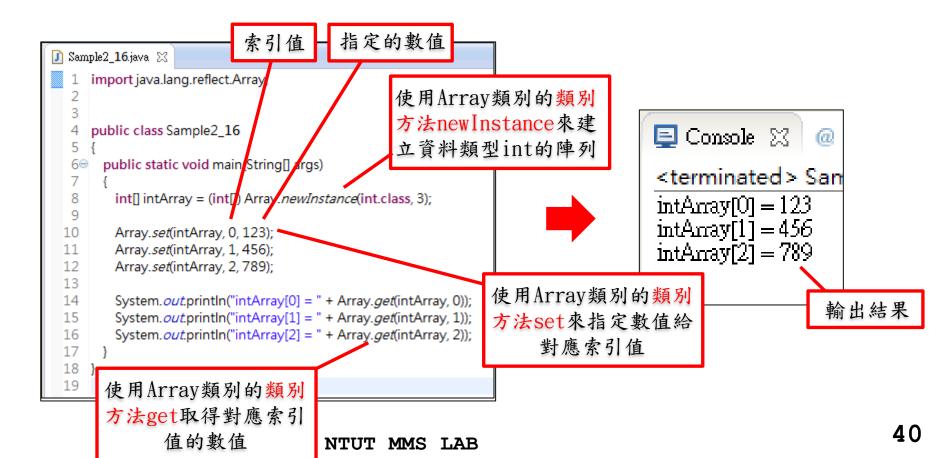
NTUT MMS LAB





■ 另一種陣列宣告

```
語法
資料類型[] 陣列變數名稱; = (資料類型[]) Array. newInstance(資料類型的類別, 陣列長度)
```

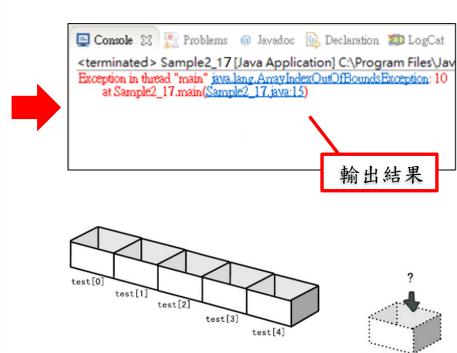




■ 常見的陣列索引值錯誤

使用陣列索引值時,您應該特別注意,「陣列索引值,不能超過陣列本身的長度」,否則將會讀不到任何資料。

```
🚺 Sample2_17.java 🖂
     public class Sample2_17
  3
       public static void main(String[] args)
         int [] test;
        test = new int[5];
        test[0] = 80;
        test[1] = 60:
 10
 11
        test[2] = 22;
                                    索引值超過範圍,
 12
        test[3] = 50;
 13
        test[4] = 75;
                                    執行到這行時會
 14
                                         發生例外
 15
        test[10] = 75;
 16
 17
        for (int i = 0; i < 5; i++)
 18
 19
          System.out.println("第" + (i+1) + "個人的分數是" + test[i] + "分");
 20
 21
```





☑ 陣列初始值設定

- \checkmark int test[5] = {80,60,22,50,75}
- ✓ int num[10] = {0}; //陣列中10個元素的初始值都為0
- ✓ 如果給定的初始值的個數小於陣列的元素個數,則剩下的元素將自動指定初始值為零

```
🚺 Sample2_18.java 🖂
     public class Sample2_18
  3
       public static void main(String[] args)
  5
                                                 陣列的初始化
  6
         int [] test = {80, 60, 22, 50, 75};
  8
         for (int i = 0; i < 5; i++)
 10
           System.out.println("第" + (i+1) + "個人的分數是" + test[i] + "分");
 11
 12
 13
 14 }
```







■ 陣列的應用

有關於陣列內元素的個數,在Java中有一個更專業的名詞稱為「陣列長度

,使用下列的方法可以取得陣列的長度: 語法 陣列變數名稱. length ■ Console

Problems <terminated > Sample2_19 🚺 *Sample2_19.java 🖂 public class Sample 219 public static void main(String[] args) int [] test = $\{80, 60, 22, 50, 75\}$; 輸出陣列的元素個數 for (int i = 0; i < test.length; i++) 輸出結果 9 test[0] System.out.println("第" + (i+1) + "個人的分數是"+ test[i] + "分"); test[1] 11 test[2] 12 test[3] 60 13 14 System.out.println("考試人數為"+ test.length + "人"); test[4] 22 75 16 test. length 43

NTUT MMS LAB



■ 陣列的排序

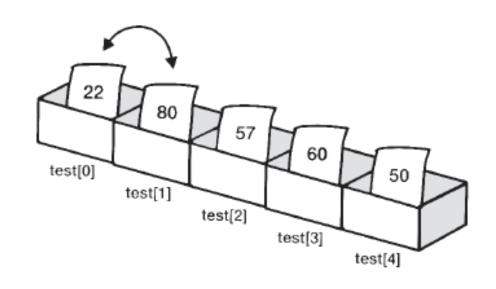
- ✓ 陣列的排序(Sorting)資料是照特定的順序放置資料
 - 遞增或遞減順序是電腦最重要的應用

原本陣列: 26, 5, 81, 7, 63

遞增排序: 5, 7, 26, 63, 81

遞減排序: 87, 63, 26, 7, 5

- ✓ 陣列的排序(Sorting)種類
 - Bubble Sort
 - Selection Sort
 - Insertion Sort
 - Quick Sort
 - Heap Sort





≦ 氣泡排序

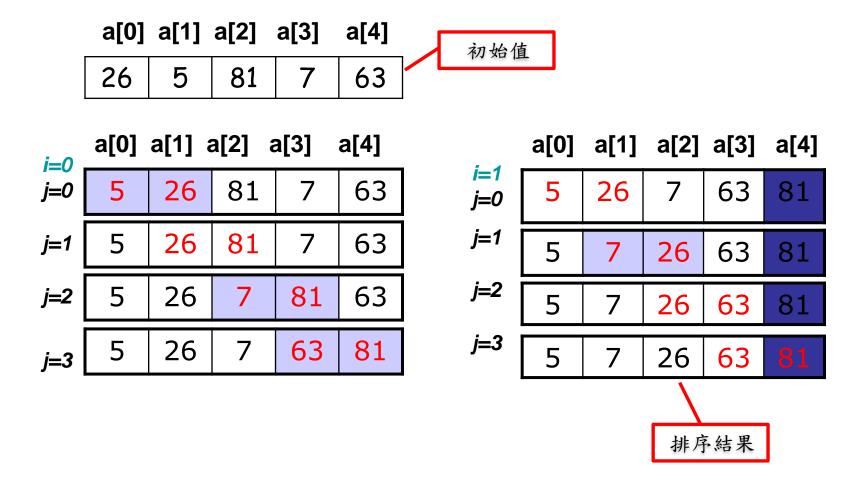
40

氣泡排序 (bubble sort或sinking sort),因為較小的數值將會如氣泡浮出水面一樣,慢慢地上升至陣列的頂點,而較大的數值則會沉到陣列的尾端。

```
Sample2_20 java. ⋈
  1 mport java.io. BufferedReader,□
     public class Sample2_20
       public static void main(String[] args) throws IOException
         BufferedReader br =
 10
            new BufferedReader(new InputStreamReader(System.in));
 11
                                                                                            Console 🖂
                                                                                                                   @ Javadoc
         System.out.println("請輸入5個人的分數");
 12
 13
         int test[] = new int[5];
                                                                                            <terminated > Sample2_20 [Jav
 14
        for (int i=0; i<test.length; i++)
 15
                                                                                            請輸入5個人的分數
 16
 17
          String str = br.readLine():
 18
          test[i]=Integer.parseInt(str);
                                                    依序讀
 19
 20
                                                    入數字
 21
        for (int s=0; s<test.length-1; s++)
 22
 23
          for (int t=s+1; t<test.length; t++)
 24
 25
            if (test[t]>test[s])
 26
 27
              int tmp = test[t];
 28
              test[t] = test[s];
 29
              test[s] = tmp;
 30
 31
                                                 氣泡排序
 32
                                                                                                                           輸出結果
 33
 34
         for (int i=0; i<test.length; i++)
 35
 36
          System.out.println("第"+(i+1)+"名的分數是"+test[i]);
 37
 38
                                                                 LAB
 39
```



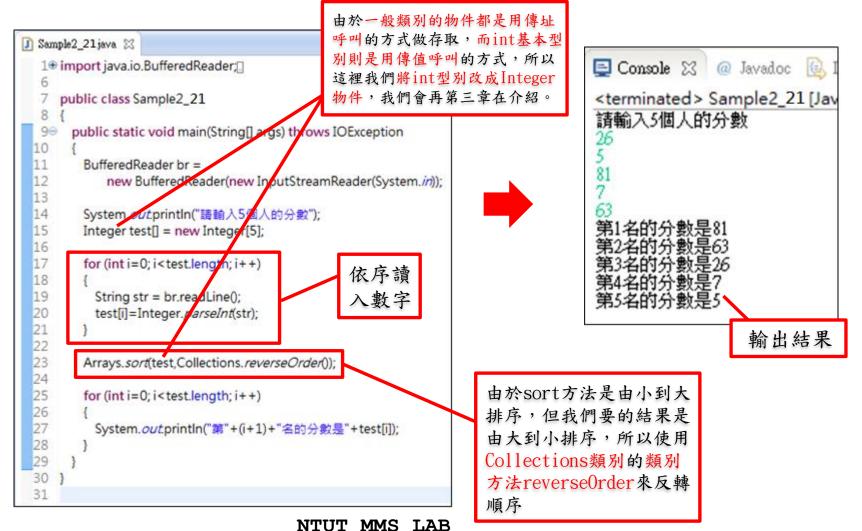
₫ 排序過程





■ Java提供的排序法

Java中Arrays類別提供一個類別方法sort讓使用者可以作排序(快速排序法)。



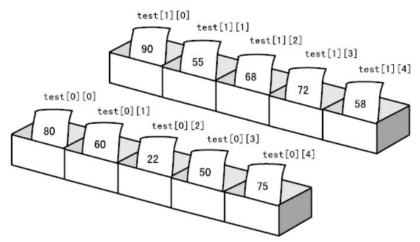


■ 多維陣列(1/2)

如果這些盒子(陣列元素)不止排成一列,而是好幾列在一起,形成一個 矩形面積,則稱為「二維陣列」。

✓ 宣告使用二維陣列

- 宣告整數 2x5 array,未設定初始值 int a[2][5]
- 宣告整數 2x5 array,設定初始值 int a[2][5] = {{80,60,22,50,75},{90,55,68,72,58}};





■ 多維陣列(2/2)

```
    Sample2_22.java 

    Sample2_22.java 

    Sample2_3
    Sample2_3
    Sample3
    Sample4
    Sample5
    Sample5

                                  public class Sample2_22
                 3
                                             public static void main(String[] args)
                 5
                 6
                                                      int [][] test;
                                                      test = new int[2][5];
                                                      test[0][0]=80;
         10
                                                      test[0][1]=60;
         11
                                                      test[0][2]=22;
         12
                                                      test[0][3]=50;
         13
                                                      test[0][4]=75;
         14
                                                      test[1][0]=90;
         15
                                                      test[1][1]=55;
         16
                                                      test[1][2]=68;
         17
                                                      test[1][3]=72;
         18
         19
                                                      for (int i=0; i<test[1].length; i++)
         20
         21
                                                                 System.out.println("第"+(i+1)+"個人的國語分數是"+test[0][i]+"分");
         22
23
                                                                 System.out.println("第"+(i+1)+"個人的數學分數是"+test[1][i]+"分");
         24
        25
          26
```

