

變數的有效範圍 (Scope)

- Java 允許在程式的任何地方宣告變數
- 宣告後的變數並非永久可用
- 變數僅在其**有效範圍**存在
- 方法內的區域變數有效範圍
 - 在方法內宣告的變數是為**區域變數**
 - 變數在經過宣告及給予初始值之後才可使用
 - 一旦離開宣告該變數的方法，此變數便失效
 - 若該區塊裡面又有小區塊，則小區塊也能使用該變數，即內層區塊不能再重定義該變數

變數的有效範圍 (Scope) (Cont.)

```
13 public class PassRef{
14     public static void main(String[] argv){
15         /*
16         TestA a = new TestA();
17         TestB b = new TestB();
18         a.show();
19         b.changTestA(a, 20);
20         a.show();
21         */
22         int x = 1;
23         {
24             int y = 20;
25             {
26                 int z = 300;
27                 System.out.println("x = "+x);
28                 System.out.println("y = "+y);
29                 System.out.println("z = "+z);
30                 System.out.println();
31             }
32             int z = 40;
33             System.out.println("x = "+x);
34             System.out.println("y = "+y);
35             System.out.println("z = "+z);
36             System.out.println();
37         }
38         int y = 2;
39         int z = 3;
40         System.out.println("x = "+x);
41         System.out.println("y = "+y);
42         System.out.println("z = "+z);
43         System.out.println();
44     }
45 }
```

x = 1
y = 20
z = 300

x = 1
y = 20
z = 40

x = 1
y = 2
z = 3

z = 300

x = 1
y = 2
z = 3

y = 20
z = 40

宣告方法中的參數

```
1 /* program name: ParaTest.java
2  * author: Yung-Chen Chou
3  * Date: Apr. 20, 2009
4  */
5 import java.io.*;
6 public class ParaTest{
7     public static void main(String[] argv) throws IOException{
8         ParaTest pt = new ParaTest();
9         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
10        System.out.print("請輸入起始數字:");
11        int v1 = Integer.parseInt(br.readLine());
12        System.out.print("請輸入結束數字:");
13        int v2 = Integer.parseInt(br.readLine());
14        pt.productResults(v1, v2);
15    }
16    void productResults(int start, int end){
17        int start;
18        int i=0, j = 0;
19        for(i=start; i<=end; i++){
20            for(j=start; j<=end; j++){
21                System.out.print(i*j+"\t");
22            }
23            System.out.println();
24        }
25    }
26 }
```

請輸入起始數字:1

請輸入結束數字:9

Exception in thread "main" java.lang.Error: 尚未解決的編譯問題:
區域變數 start 重複

at ParaTest.productResults(ParaTest.java:17)

at ParaTest.main(ParaTest.java:14)

宣告方法中的參數 (Cont.)

- **不能**在方法主體中宣告和參數同名的區域變數，因為，已經宣告在方法中的參數，在整個方法的主體中都有效
- 類別一旦產生為物件，其成員變數即與物件共存亡，只要物件未被銷毀其成員變數就仍然有效
- **★注意★** 如果類別的方法中宣告了與類別中的成員變數名稱相同的變數，則在該方法中取到的是方法中的變數資料而非成員變數資料，這種情況稱之為『**名稱遮蔽(Shadowing of Name)**』

宣告方法中的參數 (Cont.)

```
5 import java.io.*;
6 class Car{
7     double capacity;
8     void show(){
9         System.out.println("車載油量為 "+capacity+" 耗油率為每公升跑 "+eff+"公里。");
10    }
11    double eff;
12 }
13 public class ParaTest{
14     public static void main(String[] argv) throws IOException{
15         Car oldCar = new Car();
16         oldCar.capacity = 100;
17         oldCar.eff = 12.3D;
18         oldCar.show();
19     }
20 }
```

車載油量為 100.0 耗油率為每公升跑 12.3公里。

宣告方法中的參數 (Cont.)

```
5 import java.io.*;
6 class Car{
7     double capacity;
8     double eff;
9     void drive(double eff){
10         System.out.println("本車載油量為 "+capacity+" 以耗油率 "+eff);
11         System.out.println("計算，可跑"+capacity*eff+"公里");
12     }
13     void show(){
14         System.out.println("車載油量為 "+capacity+" 耗油率為每公升跑 "+eff+"公里。");
15     }
16 }
17 public class ParaTest{
18     public static void main(String[] argv) throws IOException{
19         Car oldCar = new Car();
20         oldCar.capacity = 100;
21         oldCar.eff = 12.3D;
22         oldCar.show();
23         oldCar.drive(5.89D);
24     }
25 }
```

車載油量為 100.0 耗油率為每公升跑 12.3公里。
本車載油量為 100.0 以耗油率 5.89
計算，可跑589.0公里

宣告方法中的參數 (Cont.)

- 如果，在類別的方法中存在一個變數與類別的成員變數相同，而在方法中又必須取得成員變數的值，此時必須使用『**this**』這個保留字
- 語法是：**this. 成員變數**
- **this** 的意思指的是『目前這個物件』

宣告方法中的參數 (Cont.)

```
5 import java.io.*;
6 class Car{
7     double capacity;
8     double eff;
9     void drive(double eff){
10         System.out.print("本車載油量為 "+capacity+" 以耗油率 "+eff);
11         System.out.println("計算，可跑"+capacity*eff+"公里");
12         System.out.println("drive方法下的變數 eff 值為 "+eff);
13         System.out.println("成員變數 eff 的值為 "+this.eff);
14     }
15     void show(){
16         System.out.println("車載油量為 "+capacity+" 耗油率為每公升跑 "+eff+"公里。");
17     }
18 }
19 public class ParaTest{
20     public static void main(String[] argv) throws IOException{
21         Car oldCar = new Car();
22         oldCar.capacity = 100;
23         oldCar.eff = 12.3D;
24         oldCar.show();
25         oldCar.drive(5.89D);
```

```
<已終止> ParaTest [java 應用性例] C:\Program Files\Java\jdk1.6.0_05\bin\java.exe (2008/04/28 14:00:00)
車載油量為 100.0 耗油率為每公升跑 12.3公里。
本車載油量為 100.0 以耗油率 5.89計算，可跑589.0公里
drive方法下的變數 eff 值為 5.89
成員變數 eff 的值為 12.3
```


宣告在 for 迴圈中的變數

- for 迴圈使用時, 在 for 的初始運算式中宣告迴圈變數, 此迴圈變數只在整個 for 區塊中有效, 一旦離開 for 迴圈, 該變數就無法使用 (相同的規則也適用於 foreach 迴圈)
- 如果需在迴圈結束後取得迴圈變數的值, 就必須在 for 迴圈之前自行宣告變數

宣告在 for 迴圈中的變數

```
19 public class ParaTest{
20     public static void main(String[] argv) throws IOException{
21         /*Car oldCar = new Car();
22         oldCar.capacity = 100;
23         oldCar.eff = 12.3D;
24         oldCar.show();
25         oldCar.drive(5.89D);*/
26         ParaTest pt = new ParaTest();
27         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
28         System.out.print("請輸入起始數字:");
29         int v1 = Integer.parseInt(br.readLine());
30         System.out.print("請輸入結束數字:");
31         int v2 = Integer.parseInt(br.readLine());
32         pt.productResults(v1, v2);
33     }
34     void productResults(int start, int end){
35         for(int i=start; i<=end; i++){
36             for(int j=start; j<=end; j++){
37                 System.out.print(i*j+"\t");
38             }
39             System.out.println();
40         }
41         System.out.println("i = "+i+"\t j = "+j);
42     }
43 }
44 }
```

```
請輸入起始數字:1
請輸入結束數字:9
Exception in thread "main" java.lang.Error: 尚未解決的編譯問題:
    i 無法解析
    j 無法解析

at ParaTest.productResults(ParaTest.java:41)
at ParaTest.main(ParaTest.java:32)
```

匿名陣列 (Anonymous Array)

```
45 class Sort{
46     int[] bubbleSort(int[] data, String flag){
47         if(flag.equals("d")){
48             for(int i=0;i<data.length;i++){
49                 for(int j=i+1;j<data.length;j++){
50                     if(data[i] < data[j]){
51                         int tmp1 = data[i];
52                         data[i] = data[j];
53                         data[j] = tmp1;
54                     }
55                 }
56             }
57         }else{
58             for(int i=0;i<data.length;i++){
59                 for(int j=i+1;j<data.length;j++){
60                     if(data[i] > data[j]){
61                         int tmp1 = data[i];
62                         data[i] = data[j];
63                         data[j] = tmp1;
64                     }
65                 }
66             }
67         }
68         return data;
69     }
70 }
```

```
72 public class OOPTest{
73     public static void main(String[] argv) throws IOException{
74         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
75         Sort st = new Sort();
76         System.out.print("請輸入資料個數: ");
77         int eleNum = Integer.parseInt(br.readLine());
78         System.out.print("遞增(a) or 遞減(d) 排序? ");
79         String flag = br.readLine();
80         int[] data = new int[eleNum];
81         for(int i=0;i<eleNum;i++){
82             System.out.print("data["+i+"]->");
83             data[i] = Integer.parseInt(br.readLine());
84         }
85         data = st.bubbleSort(data, flag);
86         for(int i=0;i<data.length;i++){
87             System.out.print(data[i]+"\\t");
88         }
89     }
90 }
```

```
> java OOPTest [java 應用程式] C:\Program Files\Java\jdk1.6.0_02\bin
請輸入資料個數: 7
遞增(a) or 遞減(d) 排序? d
data[0]->56
data[1]->54
data[2]->55
data[3]->1
data[4]->32
data[5]->54
data[6]->87
87      56      55      54      54      32      1
```

```
> java OOPTest [java 應用程式] C:\Program Files\Java\jdk1.6.0_02\bin
請輸入資料個數: 7
遞增(a) or 遞減(d) 排序? a
data[0]->5
data[1]->25
data[2]->94
data[3]->57
data[4]->1
data[5]->2
data[6]->34
1        2        5        25       34       57       94
```

宣告一個整數陣列

呼叫方法時將陣列傳出

若陣列資料只用一次沒有其他用途時也要宣告一個陣列，其他作法？

匿名陣列 (Anonymous Array)

```
72 public class OOPTest{
73     public static void main(String[] argv) throws IOException{
74         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
75         Sort st = new Sort();
76         //System.out.print("請輸入資料個數: ");
77         int eleNum = 3;
78         //Integer.parseInt(br.readLine());
79         System.out.print("遞增 (a) or 遞減 (d) 排序? ");
80         String flag = br.readLine();
81         int[] data = new int[eleNum];
82         /*
83         for(int i=0;i<eleNum;i++){
84             System.out.print("data["+i+"]->");
85             data[i] = Integer.parseInt(br.readLine());
86         }*/
87         data = st.bubbleSort(new int[] {23, 10, 79, 1, 98, 4, 5, 101}, flag);
88         for(int i=0;i<data.length;i++){
89             System.out.print(data[i]+"\\t");
90         }
```

<已終止> OOPTest [Java 應用程式] C:\Program Files\Java\jre6\bin\javaw.exe (2009/4/21 上午 8:00)

遞增 (a) or 遞減 (d) 排序? d

101 98 79 23 10 5 4 1

<已終止> OOPTest [Java 應用程式] C:\Program Files\Java\jre6\bin\javaw.exe (2009/4/21 上午 8:04)

遞增 (a) or 遞減 (d) 排序? a

1 4 5 10 23 79 98 101

匿名陣列 (Anonymous Array) (Cont.)

- 在呼叫方法時, 如果所需的參數是陣列, 往往必須先宣告一個陣列變數, 才能傳遞給**方法**
- 假設陣列除了作為參數傳遞以外, 並沒有其他的用途, 則不易為陣列變數取個好名字
- 有可能多佔記憶體
- 解決困擾, Java 提供了**匿名陣列(Anonymous Array)**的程式寫法
- 達到不用宣告陣列變數, 即可達到陣列傳遞的目的