

Java程式設計入門

流程控制－迴圈結構

鄭安翔

ansel_cheng@hotmail.com

課程大綱

1) **Java迴圈控制**

- **while-loop**
- **do-while-loop**
- **for-loop**

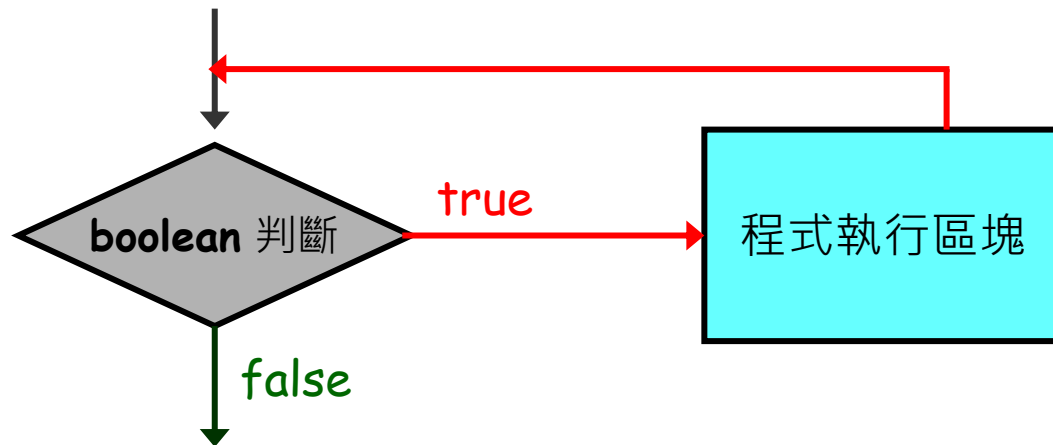
2) 迴圈進階控制

Java迴圈控制

- 處裡程式中必須重複執行的動作
- 知道停止執行的條件,不確定執行次數
 - While Loop
 - 前測型迴圈
 - 執行 0 ~ n 次
 - Do-While Loop
 - 後測型迴圈
 - 執行 1 ~ n 次
- 預先知道重複執行的次數
 - For Loop
 - 執行 n 次

while Loop

```
while (boolean) {  
    statements;  
}
```



while Loop

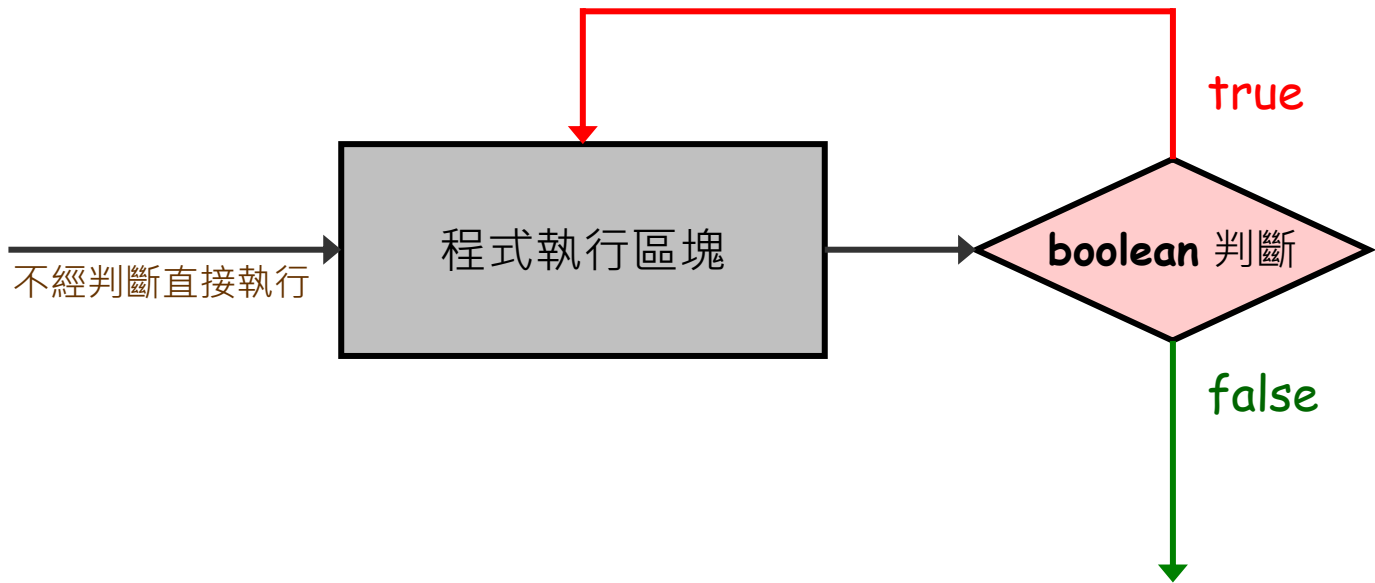
```
01 public class WhileDemo {  
02     public static void main(String[] args) {  
03         java.util.Scanner scanner =  
04             new java.util.Scanner(System.in);  
05  
06         System.out.print("輸入執行次數: ");  
07         int input = scanner.nextInt();  
08  
09         int count = 0; //計數用  
10  
11         while(count < input){ //判斷次數  
12             System.out.println("Hello World");  
13             count++; //每次執行後遞增  
14         }  
15     }  
16 }
```



```
C:\JavaClass>java WhileDemo  
輸入執行次數: 5  
Hello World!  
Hello World!  
Hello World!  
Hello World!  
Hello World!  
c:\JavaClass>
```

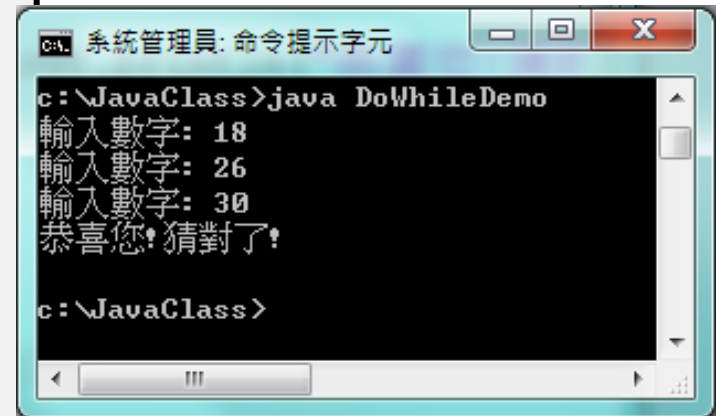
do/while Loop

```
do {  
    statements;  
} while (boolean);
```



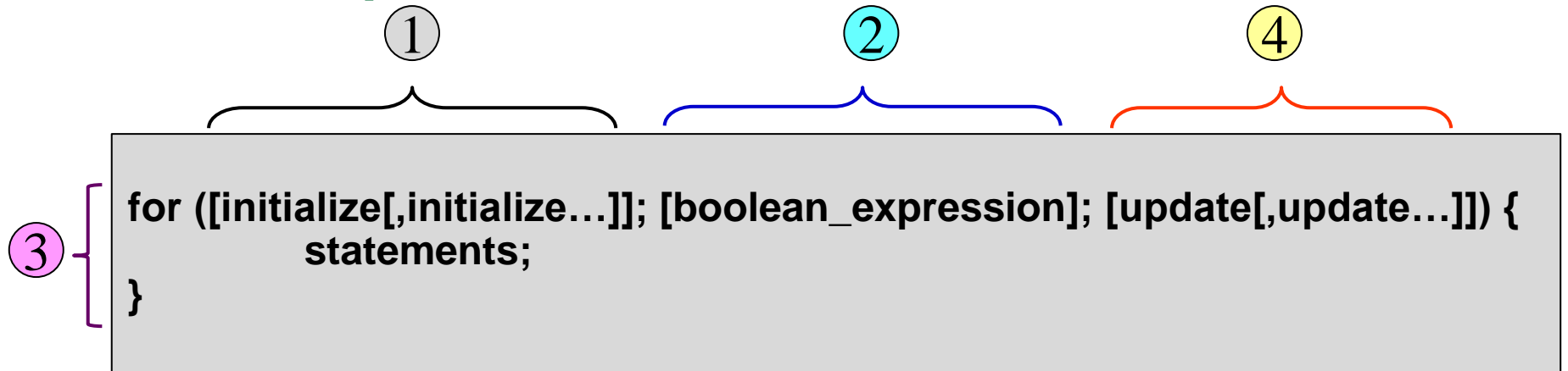
do/while Loop

```
01 public class DoWhileDemo {  
02     public static void main(String[] args) {  
03         java.util.Scanner scanner =  
04             new java.util.Scanner(System.in);  
05  
06         int input = 0;  
07         int anser = 30;  
08  
09         do{  
10             System.out.print("輸入數字: ");  
11             input = scanner.nextInt();  
12         } while ( input != anser );  
13  
14         System.out.println("恭喜您!猜對了!");  
15     }  
16 }
```



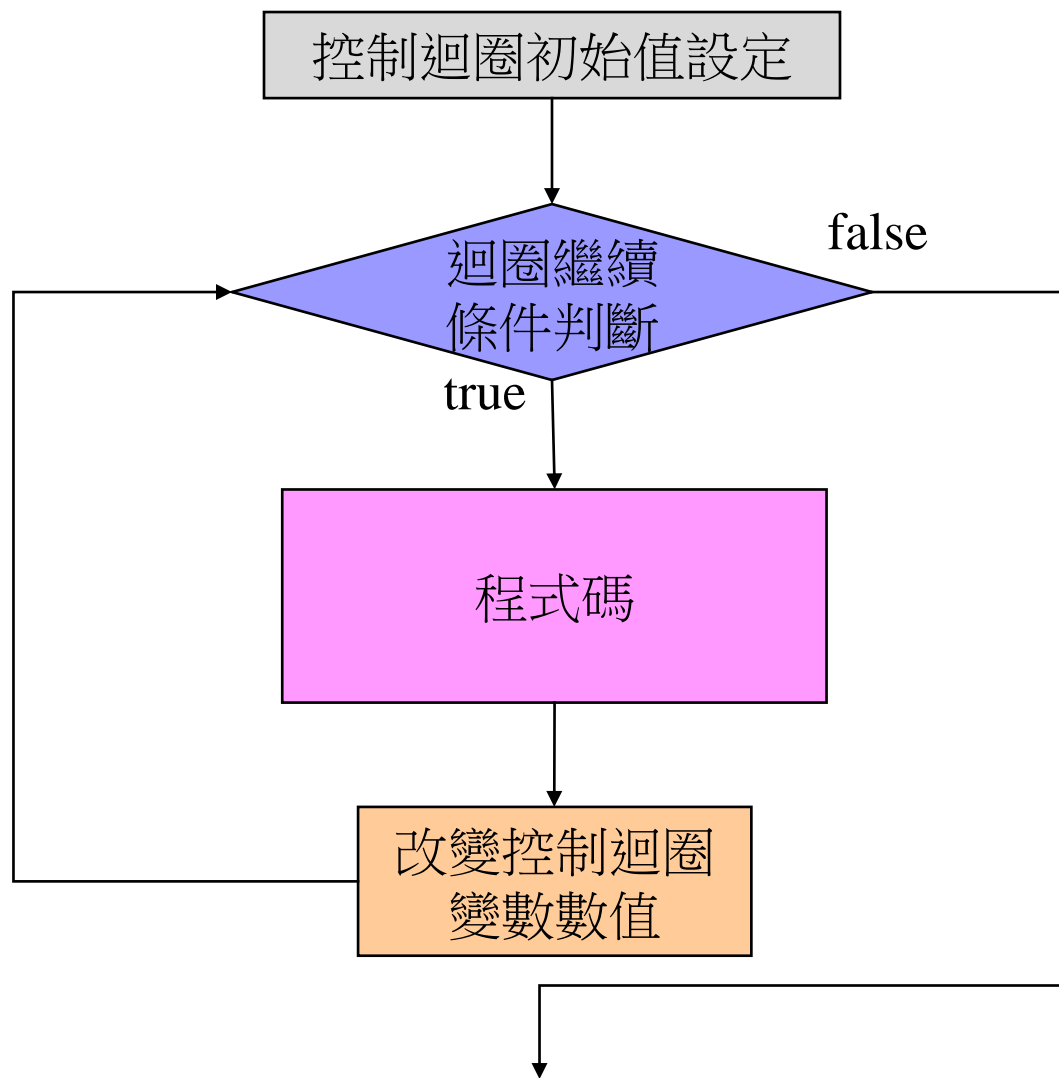
```
C:\JavaClass>java DoWhileDemo  
輸入數字: 18  
輸入數字: 26  
輸入數字: 30  
恭喜您!猜對了!  
C:\JavaClass>
```

for Loop



for迴圈執行時:

- 變數宣告. 只執行一次
- 布林判斷式. 每次迴圈開始之前
- 執行 `code_blocks`
- 更新狀態. 執行 `code_blocks`之後, 布林判斷式 之前



for Loop

```
01 public class ForDemo {  
02     public static void main(String[] args) {  
03         java.util.Scanner scanner =  
04             new java.util.Scanner(System.in);  
05  
06         System.out.print("輸入執行次數: ");  
07         int input = scanner.nextInt();  
08  
09         for(int i = 0; i < input; i++){  
10             System.out.println("Hello World!");  
11         }  
12     }  
13 }
```



```
系統管理員: 命令提示字元  
c:\JavaClass>java ForDemo  
輸入執行次數: 5  
Hello World!  
Hello World!  
Hello World!  
Hello World!  
Hello World!
```

課程大綱

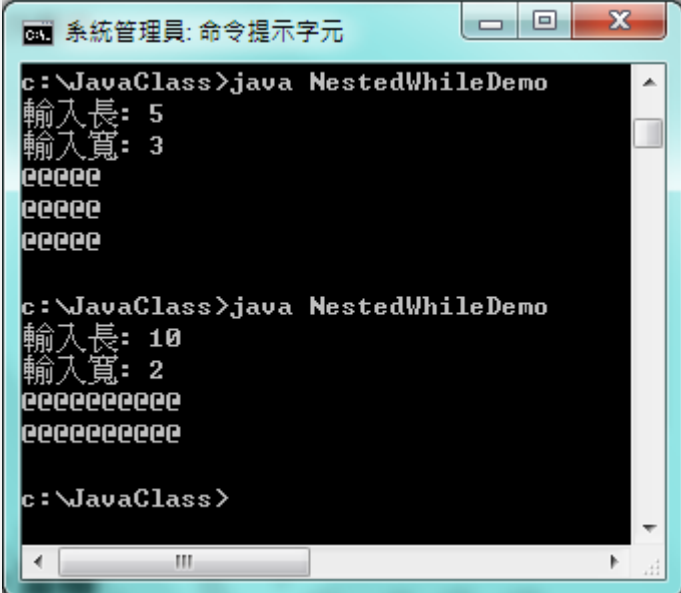
1) Java迴圈控制

2) 迴圈進階控制

- 巢狀結構
- 無窮迴圈
- **break & continue**
- **Label 標籤**

巢狀結構 while loop

```
01 public class NestedWhileDemo {
02     public static void main(String[] args) {
03         java.util.Scanner scanner =
04             new java.util.Scanner(System.in);
05
06         System.out.print("輸入長: ");
07         int length = scanner.nextInt();
08         System.out.print("輸入寬: ");
09         int width = scanner.nextInt();
10
11         int widthCount = 0;
12         while(widthCount < width){
13             int lengthCount = 0;
14             while(lengthCount < length){
15                 System.out.print("@");
16                 lengthCount++;
17             } //內層 while
18
19             System.out.println(); //換行
20             widthCount++;
21         } //外層 while
22     }
23 }
```



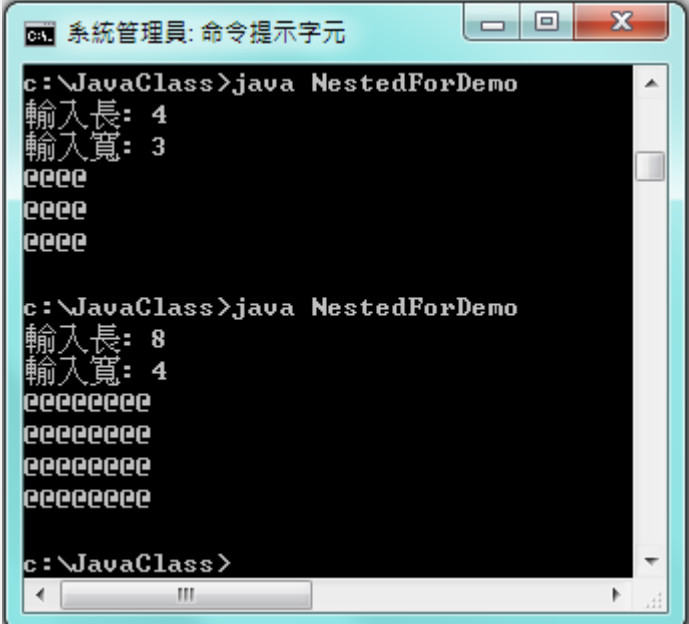
```
c:\JavaClass>java NestedWhileDemo
輸入長: 5
輸入寬: 3
#####
#####
#####

c:\JavaClass>java NestedWhileDemo
輸入長: 10
輸入寬: 2
#####
#####

c:\JavaClass>
```

巢狀結構 for loop

```
01 public class NestedForDemo {
02     public static void main(String[] args) {
03         java.util.Scanner scanner =
04             new java.util.Scanner(System.in);
05
06         System.out.print("輸入長: ");
07         int length = scanner.nextInt();
08         System.out.print("輸入寬: ");
09         int width = scanner.nextInt();
10
11         for (int i = width; i > 0; i-- ) {
12             for(int j = length; j > 0; j--){
13                 System.out.print("@");
14             } //內層 for
15
16             System.out.println(); //換行
17         } //外層 for
18     }
19 }
```



```
c:\JavaClass>java NestedForDemo
輸入長: 4
輸入寬: 3
@
@
@

c:\JavaClass>java NestedForDemo
輸入長: 8
輸入寬: 4
@@@@@@
@@@@@@
@@@@@@
@@@@@@

c:\JavaClass>
```

巢狀結構 for loop

```
for(int i=1; i<=5; i+=2){  
    for(int j=2; j<=6; j+=2){  
        System.out.print("(" + i + "," + j + ")\t");  
    }  
    System.out.print("\n");  
}
```

執行方向→

外層 for 第1次	內層 for 第1次 (1, 2)	內層 for 第2次 (1, 4)	內層 for 第3次 (1, 6)	換行
外層 for 第2次	內層 for 第4次 (3, 2)	內層 for 第5次 (3, 4)	內層 for 第6次 (3, 6)	換行
外層 for 第3次	內層 for 第7次 (5, 2)	內層 for 第8次 (5, 4)	內層 for 第9次 (5, 6)	換行

無窮迴圈

```
for (;;)
{
    // 區塊內敘述
}
```

```
while(true)
{
    // 區塊內敘述
}
```

continue & break statement

■ continue

- 略過後面敘述,繼續迴圈循環 (while/do-while/for)

■ break

- 離開正在執行的迴圈區塊 (while/do-while/for)

```
01 for(int i=1 ; i <= 10 ; i++) {  
02     if (i == 8) {  
03         break;  
04     }  
05     if (i%3 == 0) {  
06         continue;  
08     System.out.print(i);  
09 }
```

執行結果 : 1 2 4 5 7

break statement

```
01 public class BreakDemo {
02     public static void main(String[] args) {
03         java.util.Scanner scanner =
04             new java.util.Scanner(System.in);
05
06         int score = 0;
07         int sum = 0;
08         int count = -1;
09
10         while(true){
11             count++;
12             sum += score;
13             System.out.print("輸入分數(輸入-1結束):");
14             score = scanner.nextInt();
15
16             if(score == -1)
17                 break;
18         }
19         System.out.println("平均分數:" + (double)sum/count);
20     }
21 }
```



系統管理員: 命令提示字元

```
輸入分數<輸入-1結束>:80
輸入分數<輸入-1結束>:68
輸入分數<輸入-1結束>:92
輸入分數<輸入-1結束>:-1
平均分數:80.0

c:\JavaClass>
```

continue statement

```
01 public class ContinueDemo {  
02     public static void main(String[] args) {  
03         for (int i = 0; i < 10; i++) {  
04             if(i==5)  
05                 continue;  
06             System.out.println("i = " + i);  
07         }  
08     }  
09 }
```



系統管理員: 命令提示字元

```
C:\JavaClass>java ContinueDemo  
i = 0  
i = 1  
i = 2  
i = 3  
i = 4  
i = 6  
i = 7  
i = 8  
i = 9
```

標籤 Label

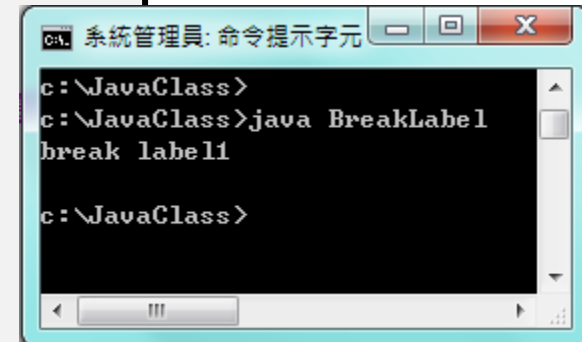
■ 標籤Label

- 將程式區塊加上標籤 (LabelName :)
- 與break和continue搭配,可指定要跳出或繼續的區段

```
01      OuterLoop :    // 標籤名稱 (迴圈的名字)
02      for( ; ; ) {   // 外層迴圈 ←
03          InnerLoop :
04          for( ; ; ) { // 內層迴圈 ←
05              break InnerLoop;
06              // 程式區塊...
07              continue InnerLoop; .....
08              // 程式區塊...
09              break OuterLoop;
10              // 程式區塊...
11              continue OuterLoop; .....
12              // 程式區塊...
13          }
14      }
```

Label + break

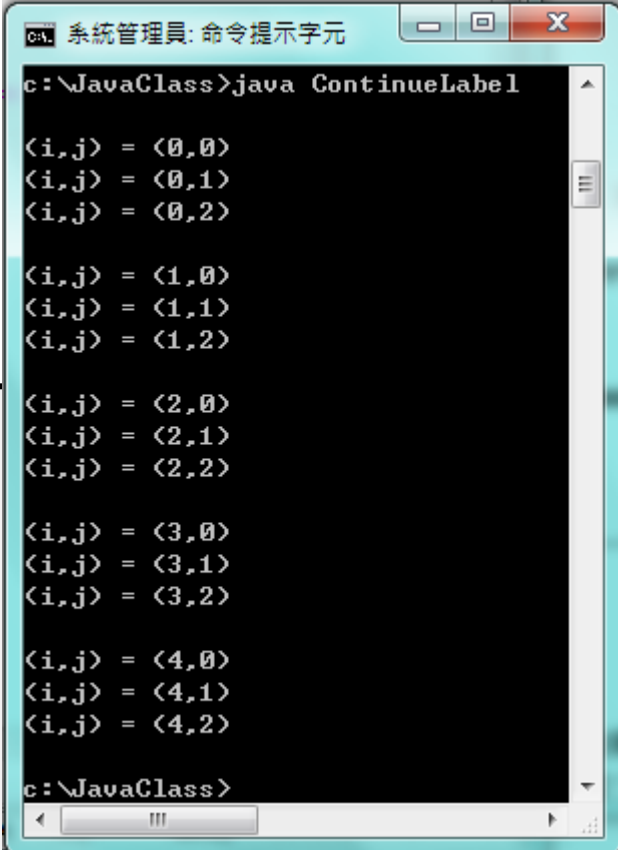
```
01 public class BreakLabel {  
02     public static void main(String[] args) {  
03         label1 : {  
04             for (int i = 0; i < 10; i++) {  
05                 if(i==9){  
06                     System.out.println("break label1");  
07                     break label1;  
08                 }  
09             }  
10             System.out.println("for迴圈之後執行");  
11         } //label1區塊結尾  
12     }  
13 }
```



```
C:\JavaClass>  
c:\JavaClass>java BreakLabel  
break label1  
c:\JavaClass>
```

Label + continue

```
01 public class ContinueLabel {  
02     public static void main(String[] args) {  
03         label1:  
04         for (int i = 0; i < 5; i++) {  
05             System.out.println();  
06             label2:  
07             for (int j = 0; j < 5; j++) {  
08                 if(j==3)  
09                     continue label1;  
10                 System.out.println("(i,j) = (" + i + ", " + j + ")");  
11             }  
12         }  
13     }  
14 }  
15 }
```



```
c:\JavaClass>java ContinueLabel  
  
<i,j> = <0,0>  
<i,j> = <0,1>  
<i,j> = <0,2>  
  
<i,j> = <1,0>  
<i,j> = <1,1>  
<i,j> = <1,2>  
  
<i,j> = <2,0>  
<i,j> = <2,1>  
<i,j> = <2,2>  
  
<i,j> = <3,0>  
<i,j> = <3,1>  
<i,j> = <3,2>  
  
<i,j> = <4,0>  
<i,j> = <4,1>  
<i,j> = <4,2>  
  
c:\JavaClass>
```