

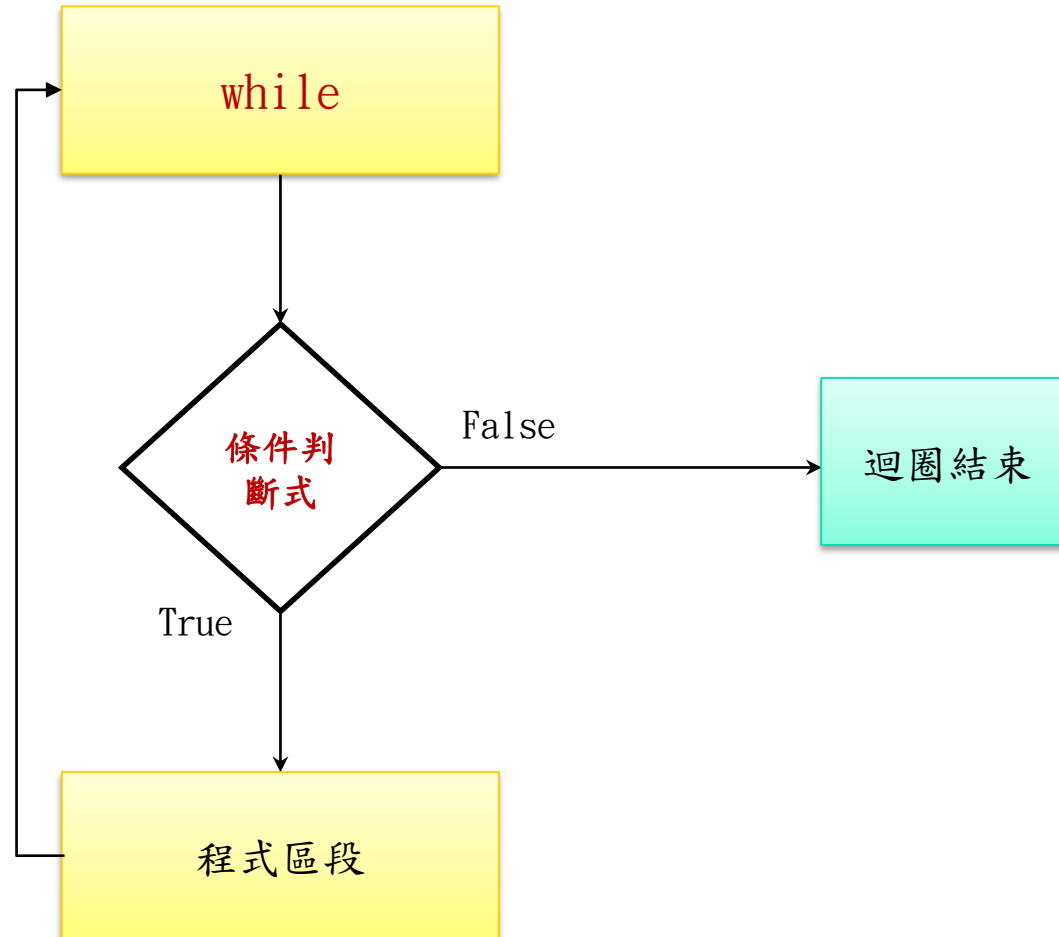
# Chapter 4-2

## 流程控制

---

- for/while迴圈
- break/continue
- Block標記用法

# 流程控制-while



# while 迴圈

## ■ while (判斷式){

//程式區段

}

```
while (a>b){  
    System.out.println("a>b");  
    System.out.println("條件如果為True且不改變"  
        + "則會一直迴圈執行!");  
}
```

一直執行

while 使用方式跟if很像，一樣是在指令後面加上 條件判斷式，唯一不同的是if條件為真只會執行一次，while條件為真則會執行無窮次(除非條件改成False)

# while跟for的差別

不確定迴圈次數或需要邏輯判斷的使用while，知道迴圈次數的使用for

- `for(int i=0;i<5;i++)` ➔ 固定執行五次
- `while (條件式)` ➔ 程式碼會一直執行到條件式改變為止

# 範例

```
Scanner input=new Scanner(System.in);  
int score=0;  
int total=0;  
  
while(score!=-1){  
    System.out.print("請輸入分數:");  
    score=input.nextInt();  
    total+=score;  
}  
  
System.out.println("總分為:"+total);
```

透過判斷score== -1  
才離開迴圈

請輸入分數:100  
請輸入分數:99  
請輸入分數:55  
請輸入分數:77  
請輸入分數:-1  
總分為:330

# 範例

- 使用while迴圈設計可以輸入n次姓名跟分數的程式，直到詢問是否離開，並輸入'y'為止。

請輸入第1位學生姓名：

jerry

請輸入jerry分數：100

是否離開？(y/n) n

請輸入第2位學生姓名：

mary

請輸入mary分數：88

是否離開？(y/n) n

請輸入第3位學生姓名：

kevin

請輸入kevin分數：77

是否離開？(y/n) y

共輸入3位學生

總分為265

平均分為88.3

詢問是否要離開，  
使用input輸入y或n

```
while(!exit.equals("y")){  
  
}
```

```
Scanner input=new Scanner(System.in);

String exit="n";
int total=0;
int count=0;

while(!exit.equals("y")) {

    System.out.println("請輸入第"++count+"位學生姓名:");
    String name=input.next();
    System.out.printf("請輸入[%s]分數:",name);
    int score=input.nextInt();
    total+=score;

    System.out.print("是否離開?(y/n)");
    exit=input.next();

}

System.out.printf("共輸入%d位學生\n",count);
System.out.printf("總分為%d\n",total);
System.out.printf("平均分為%.1f\n",(float)total/count);
```



# == or equals 比對

true	==	Equals方法
基本類型 (int, double, float...)	值的相等	
String	指向同個參考	相同類型後再比較其 內容值是否相同
物件	指向同個參考	



## ■ == 比對要為true:

- 一般類型直接比對
- 當比對物件跟字串時唯有當兩個參考指向同一個物件(記憶體位置)，且值相同時，==運算子的結果才為true。

## ■ equals()比對要為true:

- 比較兩個物件是否為相同類型的類別後再比較其內容值是否相同。

```
int a=0;
int b=0;
System.out.println(a==b);
```

```
String c="test";
String d="test";
System.out.println(c==d);
System.out.println(c==new String( original: "test"));
System.out.println(c.equals(new String( original: "test")));
```

true
true
false
true

# do while 迴圈

- 應用在至少迴圈執行一次的場合。

```
do {
```

```
    //程式區段(至少會執行一次)
```

```
}while(判斷式);
```

```
boolean first = false;
```

```
while (first) {  
    System.out.println("a:被執行了一次");  
}
```

沒有執行

```
do {  
    System.out.println("b:被執行了一次");  
} while (first);
```

至少會執行一次

b:被執行了一次

# 數字倒數

- 使用者輸入一個數字，反向輸出。

請輸入一個數字：

12345678

倒轉輸出：

87654321

```
System.out.println("請輸入一個數字:");  
Scanner input = new Scanner(System.in);  
long numbers = input.nextLong();  
System.out.println("倒轉輸出:");  
do {  
  
    System.out.print(numbers % 10);  
    numbers /= 10;  
  
} while (numbers != 0);
```

# 迴圈中斷(break)

- 迴圈中有的時候會需要在某個條件成立後跳離不繼續執行，這個時候就得需要使用break指令

```
if (count>=5)
    break;
```

```
int count=0;
do {

    System.out.print(numbers%10);
    numbers/=10;
    if(++count==5)
        break;

}while(numbers!=0);
```

請輸入一個數字：

123456789

倒轉輸出：

98765


- 使用 `while(true)+break;`
  - 改寫 `ch4_12.java`

```
Scanner input = new Scanner(System.in);

int count = 0, total = 0;

while (true) {
    System.out.println("請輸入第" + (++count) + "位學生姓名:");
    String name = input.next();
    System.out.printf("請輸入%s分數:", name);
    total += input.nextInt();
    System.out.print("是否離開?(y/n)");
    if (input.next().toLowerCase().equals("y"))
        break;
}

System.out.printf("共輸入%d位學生\n", count);
System.out.printf("總分為%d\n", total);
System.out.printf("平均分為%.1f\n", (float) total / count);
```



離開while(true)迴圈


- for迴圈也可以使用break
  - 計數1~100之間小於67的奇數和

```
int total=0;
int i;
for(i=1;i<100;i+=2) {

    if (i>66)
        break;

    total+=i;
}

System.out.printf("1~%d 總合為:%d",i,total);
```



離開for迴圈

1~67 總合為:1089|

## 迴圈回到起始(continue)

- 程式中有的時候會需要在某個條件成立後回到迴圈的起始重新執行(跳過該次)，這個時候就得需要使用continue指令

```
if (x==10)  
    continue;
```



```
Scanner input = new Scanner(System.in);
int count = 0, total = 0;

while (true) {
    System.out.println("請輸入第" + (++count) + "位學生姓名:");
    String name = input.next();
    System.out.printf("請輸入%s分數:", name);
    int score=input.nextInt();
    if (score<0 | score>100){
        System.out.println("輸入錯誤!");
        --count;
        continue;
    }
    total +=score;
    System.out.print("是否離開?(y/n)");
    if (input.next().toLowerCase().equals("y"))
        break;
}
```

```
System.out.printf("共輸入%d位學生\n", count);
System.out.printf("總分為%d\n", total);
System.out.printf("平均分為%.1f\n", (float) total / count);
```

請輸入第1位學生姓名:

jerry

請輸入jerry分數:0

是否離開?(y/n) n

請輸入第2位學生姓名:

mary

請輸入mary分數:-10

輸入錯誤!

請輸入第2位學生姓名:

mary

請輸入mary分數:77

是否離開?(y/n) y

共輸入2位學生

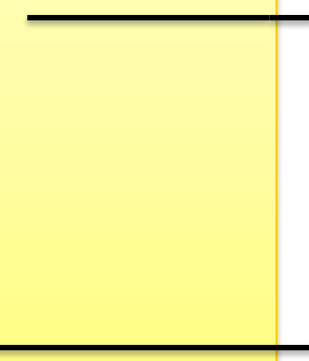
總分為77

平均分為38.5

# 標記的使用方式

在Java程式中可以使用標記+break功能跳出指定的區塊。

```
Block1:{  
    .....  
    if(true){  
        break Block1;  
    }  
    .....  
}  
.....
```

A diagram consisting of a horizontal line from the end of the 'break Block1;' statement, a vertical line extending downwards, and a horizontal line ending in an arrow pointing to the left, positioned below the closing brace of the 'Block1' block.

## ■ 範例

```
Block1:
{
    int i = 0;
    Block2:
    {
        System.out.println("目前在Block2");
        for (i = 0; i < 10; i++) {
            System.out.println(i);
            if (i == 5) {
                break Block2;
            }
        }

        System.out.println("跳出目前在Block1");
    }
}
```

目前在Block2

0

1

2

3

4

5

跳出目前在Block1

## ■ 輸出2, 3, 4

```
public static void main(String[] args) {  
    int i=0,j=0,k=0;  
    for(i=0;i<5;i++){  
        for(j=0;j<5;j++){  
            for(k=0;k<5;k++){  
                if(i==2 && j==3 && k==4){  
                    break;  
                }  
            }  
        }  
    }  
  
    System.out.println(i+" "+j+" "+k);  
}
```

5, 5, 5

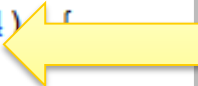
```
int i = 0, j = 0, k = 0;  
boolean done = false;  
for (i = 0; i < 5; i++) {  
    for (j = 0; j < 5; j++) {  
        for (k = 0; k < 5; k++) {  
            if (i == 2 && j == 3 && k == 4) {  
                done = true;  
                break;  
            }  
        }  
        if (done)  
            break;  
    }  
    if (done)  
        break;  
}  
  
System.out.println(i + "," + j + "," + k);
```

2, 3, 4

## ■ 使用標記

```
int i = 0, j = 0, k = 0;
Block1:
{
    for (i = 0; i < 5; i++) {
        for (j = 0; j < 5; j++) {
            for (k = 0; k < 5; k++) {
                if (i == 2 && j == 3 && k == 4) {
                    break Block1;
                }
            }
        }
    }

    System.out.println(i + "," + j + "," + k);
}
```



一次跳出三個迴圈