

# 控制敘述 (上) : 判斷與 while 迴圈

Deitel & Deitel 《Java How to Program: Early Objects》11e, Ch.4

## § 1 一、名詞速查表

中文	English	一句話定義	科目	章節
演算法	algorithm	解決問題的步驟與順序	Java	CH4
虛擬碼	pseudocode	用接近自然語言描述演算法，不是真程式	Java	CH4
控制結構	control structure	控制執行流向的三種結構：順序、選擇、重複	Java	CH4
選擇結構	selection	依條件決定走哪條路： <code>if</code> 、 <code>if-else</code> 、 <code>switch</code>	Java	CH4
重複結構	repetition / loop	反覆執行某段： <code>while</code> 、 <code>for</code> 、 <code>do-while</code>	Java	CH4
單選	if	條件成立才做，否則跳過	Java	CH4
雙選	if-else	條件成立做 A、否則做 B	Java	CH4
while 迴圈	while loop	條件為真時反覆執行迴圈體	Java	CH4
計數器控制	counter-controlled	已知重複次數，用計數器數到固定次數	Java	CH4
哨兵控制	sentinel-controlled	不知次數，用特殊值（哨兵）當結束訊號	Java	CH4
巢狀	nested	一個控制結構放在另一個裡面	Java	CH4
複合賦值	compound assignment	<code>+=</code> <code>-=</code> <code>*=</code> <code>/=</code> <code>%=</code> ， <code>x += 3</code> 等於 <code>x = x + 3</code>	Java	CH4
遞增／遞減	increment / decrement	<code>++</code> 加一、 <code>--</code> 減一	Java	CH4
前置／後置	prefix / postfix	<code>++x</code> 先加再用； <code>x++</code> 先用再加	Java	CH4

## § 2 二、核心概念

程式的執行流向只由三種控制結構組成：**順序**（由上而下）、**選擇**（分岔）、**重複**（迴圈）。任何程式都是這三種的組合與堆疊。

本章學會「**做選擇**」 (`if`、`if-else`) 和「**重複做**」 (`while`)。重複又分兩型：**知道要做幾次**用計數器控制；**不知道幾次、靠特殊訊號停用**哨兵控制。

選擇與重複都靠一個 `boolean` **條件** (`true/false`) 決定流向，條件用 CH2 的關係/相等運算子 (`>`、`<`、`>=`、`<=`、`==`、`!=`) 寫成。

## § 3 三、主要內容

### 3.1 1. `if` 單選與 `if-else` 雙選

```
if (grade >= 60) // 單選：成立才做
    System.out.println("Passed");

if (grade >= 60) // 雙選：二擇一
    System.out.println("Passed");
else
    System.out.println("Failed");
```

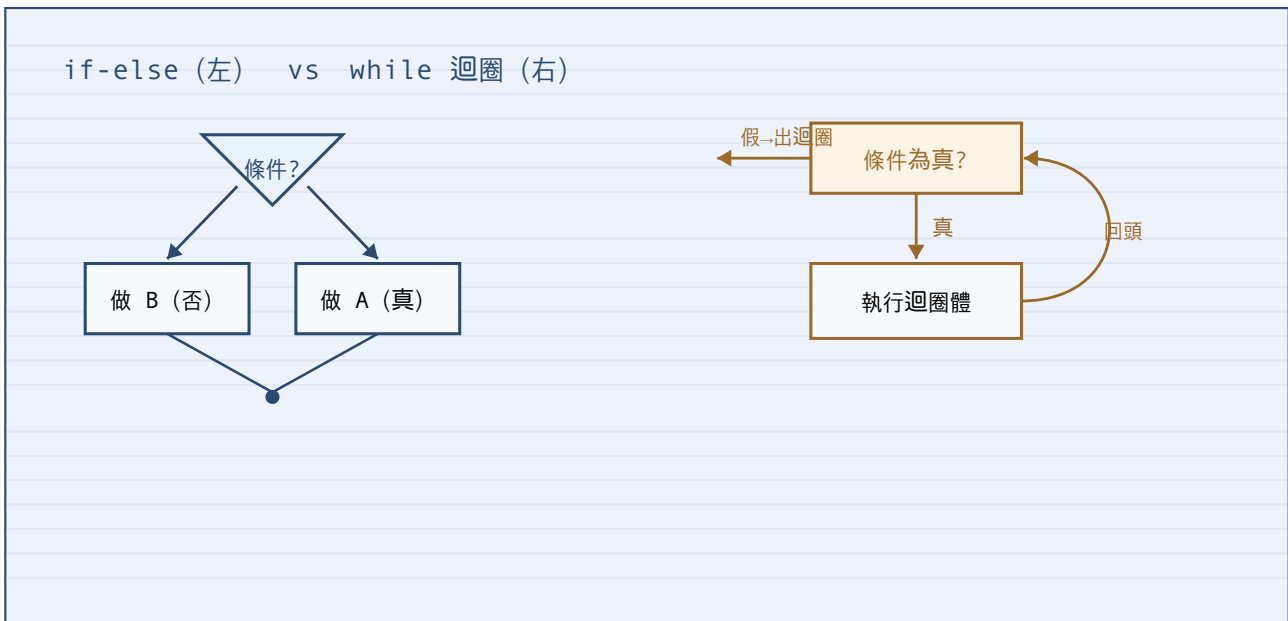
多條敘述要用大括號 `{ }` 包成一個區塊：

```
if (grade >= 60) {
    System.out.println("Passed");
    passes = passes + 1;
}
```

**條件運算子 `?:`** (`if-else` 的精簡寫法)：

```
System.out.println(grade >= 60 ? "Passed" : "Failed");
```

## 3.2 2. 控制流程圖



## 3.3 3. while 重複

條件為真時，**反覆**執行迴圈體；每輪結束回頭再檢查條件，直到條件為假才離開。

```

int product = 3;
while (product <= 100)    // 條件
    product = 3 * product; // 迴圈體 (一定要讓條件最終變假，否則無窮迴圈)
  
```

## 3.4 4. 計數器控制重複 (知道做幾次)

求 10 位學生的平均分：

```

int total = 0;           // 累加總和
int counter = 1;        // 計數器從 1 開始
while (counter <= 10) { // 數到 10
    int grade = input.nextInt();
    total = total + grade;
    counter = counter + 1; // 每輪 +1，邁向結束
}
int average = total / 10; // 注意：整數除法
  
```

計數器迴圈追蹤 (grade 依序 80, 90, 70)

counter	grade	total	條件 counter≤3
1	80	0→80	真, 做
2	90	80→170	真, 做
3	70	170→240	真, 做
4	-	240	假, 停

離開後 average = total / 3 = 240 / 3 = 80

### 3.5 5. 哨兵控制重複 (不知道做幾次)

不知有幾位學生，約定輸入 **-1 (哨兵值)** 代表結束：

```
int total = 0, count = 0;
int grade = input.nextInt(); // 先讀第一筆
while (grade != -1) {       // 不是哨兵就繼續
    total = total + grade;
    count = count + 1;
    grade = input.nextInt(); // 讀下一筆
}
double average = (double) total / count; // 轉 double 才有小數
```

- 哨兵值要選**不可能出現在正常資料**的值 (分數不會是 -1)。
- `(double)` 是**強制轉型**，讓除法保留小數 (否則整數除法去小數)。

### 3.6 6. 複合賦值與遞增遞減

```
total += grade; // 等於 total = total + grade; 另有 -= *= /= %=
counter++;      // 等於 counter = counter + 1
counter--;      // 等於 counter = counter - 1
```

**前置 vs 後置** (差在「先用還是先變」)：

```
int a = 5, b;
b = ++a; // 先 a 變 6, 再給 b → a=6, b=6
int c = 5, d;
d = c++; // 先給 d 用舊值 5, 再 c 變 6 → c=6, d=5
```

## § 4 四、語法與 API 速查

```
if (條件) 敘述;  
if (條件) { 區塊 } else { 區塊 }  
條件 ? 真值 : 假值 // 條件運算子  
while (條件) { 迴圈體 } // 條件為真時反覆
```

- **計數器控制**：初始化計數器 → `while (counter <= N)` → 體內處理 + `counter++`
- **哨兵控制**：先讀一筆 → `while (值 != 哨兵)` → 處理 + 讀下一筆
- **複合賦值**：`+= -= *= /= %=`
- **遞增遞減**：`++ --` (前置 `++x` 先變後用；後置 `x++` 先用後變)
- **整數除法保留小數**：`(double) total / count`

## § 5 五、常見錯誤

- **= 寫成比較**：`while (x = 5)` 是賦值；條件要用 `==` (Java 會因型別不符報錯，比 C 安全)。
- **無窮迴圈**：迴圈體沒讓條件最終變假 (忘了 `counter++`)，程式卡住。
- **差一錯誤 (off-by-one)**：`counter <= 10` 跑 10 次、`counter < 10` 只跑 9 次，邊界要算清楚。
- **整數除法算平均**：`total / count` 兩個 int 相除去小數；要 `(double) total / count`。
- **懸置 else (dangling else)**：`else` 配對最近的未配對 `if`；多層用大括號釐清。
- **漏大括號**：`if (c) a; b;` 的 `b;` 不在 `if` 內、一定會執行；多敘述要 `{ }`。
- **前置後置搞混**：`x++` 與 `++x` 單獨用一樣，但放在運算式裡值不同。

## § 6 六、練習題

### 例題 1：if-else 判斷及格

讀一個整數分數， $\geq 60$  印 `Passed`，否則印 `Failed`。

1. Scanner 讀 grade 2. `if (grade >= 60)` 印 Passed 3. `else` 印 Failed


```
import java.util.Scanner;
public class Q1 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int grade = input.nextInt();
        if (grade >= 60)
            System.out.println("Passed");
        else
            System.out.println("Failed");
    }
}
```

易錯：`>=` 寫成 `=>`；else 後又寫條件。





## § 7 七、自我檢核

---

- [] 能說出三種控制結構，並分辨選擇與重複。
- [] 會寫 `if`、`if-else`，知道多敘述要用 `{ }`。
- [] 會寫 `while`，並確保條件最終會變假（不無窮迴圈）。
- [] 分得清計數器控制與哨兵控制，各自何時用。
- [] 算平均時知道要 `(double)` 轉型避免整數除法。
- [] 會用 `+=`、`++`、`--`，並分辨前置與後置的差別。