Java 應用程式入門：輸出入、運算子與判斷

Deitel & Deitel《Java How to Program: Early Objects》11e, Ch.2

§ 1　一、名詞速查表

中文	English	一句話定義	科目	章節
應用程式	application	用 `java` 指令啟動 JVM 後執行的 Java 程式，必須含 `main` 方法	Java	CH2
類別	class	Java 程式的基本建構單位；每個程式由一或多個類別組成	Java	CH2
主方法	main method	`public static void main(String[] args)`，JVM 執行的進入點	Java	CH2
標準輸出物件	System.out	預先存在的物件，呼叫 `print`/`println`/`printf` 把文字顯示到螢幕	Java	CH2
跳脫序列	escape sequence	反斜線開頭的特殊字元，如 `\n`（換行）、`\t`（定位）	Java	CH2
格式化輸出	printf	依格式控制字串輸出，`%d`（整數）`%s`（字串）`%f`（浮點）	Java	CH2
匯入宣告	import declaration	`import java.util.Scanner;`，告訴編譯器要用哪個類別	Java	CH2
掃描器	Scanner	讀取鍵盤輸入的類別，`nextInt()` 讀一個整數	Java	CH2
變數	variable	記憶體中具名的位置，存放某型別的值	Java	CH2
基本型別	primitive type	Java 內建型別，如 `int`、`double`、`boolean`	Java	CH2
賦值運算子	assignment operator	`=`，把右邊的值存入左邊的變數（由右往左）	Java	CH2
取餘數	modulus / remainder	`%`，回傳整數除法的餘數，如 `7 % 5` 得 `2`	Java	CH2
運算子優先序	operator precedence	決定運算先後的規則：`* / %` 先於 `+ -`	Java	CH2
相等運算子	equality operators	`==`（等於）、`!=`（不等於），結果為 `boolean`	Java	CH2

中文	English	一句話定義	科目	章節
關係運算子	relational operators	`>` `<` `>=` `<=`，比較大小，結果為 `boolean`	Java	CH2

§ 2 二、核心概念

Java 程式由**類別（class）**組成；能獨立執行的程式（application）必須有一個 `main` **方法**當進入點。寫好 `.java` 後，先用 `javac` **編譯（compile）**成 bytecode（`.class`），再用 `java` 啟動 **JVM（Java Virtual Machine）執行（execute）**。

> 本章圍繞四件事：把資料**輸出**到螢幕（`print`/`println`/`printf`）、從鍵盤**輸入**（`Scanner`）、用**運算子**計算（算術 `+ - * / %`）、依條件**判斷**（`if` 配 `== != > < >= <=`）。中間用**變數**把值存在記憶體裡。

程式骨架固定如下，先記熟結構再填內容：

```
public class 類別名 {              // 檔名必須是「類別名.java」
    public static void main(String[] args) {    // 執行從這裡開始
        // 你的敘述（statement）寫在這裡，每句以分號 ； 結尾
    }
}
```

§ 3 三、主要內容

3.1 1. 第一個程式：輸出一行文字（Welcome1）

```
// Fig. 2.1: Welcome1.java
// 顯示一行文字的程式
public class Welcome1 {
    public static void main(String[] args) {
        System.out.println("Welcome to Java Programming!");
    }
}
```

- `public class Welcome1`：宣告類別，名稱**必須**與檔名 `Welcome1.java` 相同（含大小寫）。
- `main` 是 JVM 自動呼叫的進入點；`static` 表示不需先建立物件就能呼叫。
- `System.out.println(...)`：把括號內的字串輸出後**換行**。

註解（comment）三種：`//` 單行；`/* ... */` 跨行傳統註解；`/** ... */` Javadoc 文件註解。編譯器全部忽略。

3.2 2. 輸出三兄弟：print、println、printf

方法	行為
`System.out.print(s)`	輸出 `s`，**不換行**（游標停在原行尾）
`System.out.println(s)`	輸出 `s` 後**換行**
`System.out.printf(格式, 值...)`	依**格式控制字串**輸出，可一次排版多個值

跳脫序列讓字串包含特殊字元：

```
System.out.print("Welcome to ");
System.out.println("Java Programming!");    // 與上行接成同一行後才換行
System.out.printf("%s%n%s%n", "Welcome to", "Java Programming!");
```

`%n` 是與平台無關的換行；`%s` 代入一個字串、`%d` 代入一個整數。

3.3 3. 輸入：用 Scanner 讀鍵盤

```
import java.util.Scanner;          // 1. 匯入 Scanner 類別

Scanner input = new Scanner(System.in);    // 2. 建立 Scanner 物件，接鍵盤
System.out.print("Enter first integer: "); // 3. 提示使用者
int number1 = input.nextInt();             // 4. 讀入一個整數存進變數
```

`import` 要寫在程式最前面（`public class` 之前）。`System.in` 代表鍵盤（標準輸入）。`nextInt()` 等使用者輸入一個整數並按 Enter，回傳該整數。

3.4 4. 變數、型別與賦值

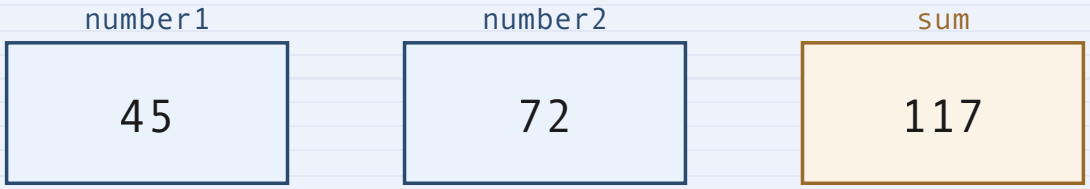
```
int number1;           // 宣告：在記憶體開一個放 int 的位置，取名 number1
number1 = 45;          // 賦值：把 45 存進 number1（= 是「存入」，不是數學等於）
int sum = number1 + number2;    // 宣告同時初始化
```

`int` 存整數；`double` 存含小數的數；`=` 把**右邊算出的值**存到**左邊的變數**。

3.5 5. 記憶體概念（本章重點）

把變數想成貼了名牌的盒子，盒子裡放當前的值：

執行 number1 = 45; number2 = 72; sum = number1 + number2;



讀取變數值（如 number1 + number2）→ 不會改變盒子內容（nondestructive）
把新值存入變數 → 覆蓋舊值、舊值消失（destructive read-in）

讀取不破壞、寫入會覆蓋：計算 `number1 + number2` 只是「讀」兩個盒子的值，不改變它們；而 `sum = ...` 會把 `sum` 盒子的舊值換成新值。

3.6 6. 算術運算子與優先序

運算	運算子	範例	結果
加	`+`	`7 + 5`	`12`
減	`-`	`7 - 5`	`2`
乘	`*`	`7 * 5`	`35`
除	`/`	`7 / 5`	`1`（**整數除法去小數**）
餘數	`%`	`7 % 5`	`2`

優先序：先 `* / %`（由左而右），再 `+ -`（由左而右）；要改變順序用**括號**。

求值：a = 2 + 3 * 4 - 10 / 2

步驟1 3 * 4 = 12 → a = 2 + 12 - 10 / 2
步驟2 10 / 2 = 5 → a = 2 + 12 - 5
步驟3 2 + 12 = 14 → a = 14 - 5
步驟4 14 - 5 = 9 → a = 9

3.7 7. 判斷：相等與關係運算子

`if` 敘述在**條件為真**時執行大括號內的敘述：

```
if (number1 == number2)
    System.out.printf("%d == %d%n", number1, number2);
if (number1 > number2)
    System.out.printf("%d > %d%n", number1, number2);
```

運算子	意義	運算子	意義
`==`	等於	`!=`	不等於
`>`	大於	`<`	小於
`>=`	大於等於	`<=`	小於等於

注意 `==`（比較）與 `=`（賦值）是**兩個不同的運算子**，最容易混。條件的結果是 `boolean`（`true`/`false`）。

§ 4 四、語法與 API 速查

程式骨架

```
import java.util.Scanner;              // 需要輸入才寫
public class 類別名 {
    public static void main(String[] args) {
        // 敘述；每句以 ； 結尾
    }
}
```

輸出：`System.out.print(s)` / `println(s)` / `printf(格式, 值...)` **跳脫序列**：`\n` 換行、`\t` 定位、`\"` 雙引號、`\\` 反斜線、`\r` 歸位 **printf 格式符**：`%d` 整數、`%s` 字串、`%f` 浮點、`%n` 換行 **輸入**：`Scanner input = new Scanner(System.in);` → `input.nextInt()` / `nextDouble()` / `next()` / `nextLine()` **算術**：`+ - * / %`（優先序 `* / %` > `+ -`，括號最先） **關係/相等**：`> < >= <= == !=`（結果 `boolean`）

§ 5 五、常見錯誤

- **檔名與 `public class` 名不同**：`public class Welcome1` 必須存成 `Welcome1.java`，否則編譯錯。
- **漏分號 `;`**：每句敘述結尾要分號。
- **用了 Scanner 卻沒 `import java.util.Scanner;`**：編譯找不到 Scanner。

- **`=` 與 `==` 混用**：`if (a = b)` 是賦值（型別錯誤），比較要 `if (a == b)`。
- **整數除法吃掉小數**：`7 / 2` 得 `3` 不是 `3.5`；要小數需用 `double`。
- **優先序誤判**：`2 + 3 * 4` 是 `14` 不是 `20`；先乘除後加減。
- **大小寫錯**：`System`（大寫 S）、`String`（大寫 S）、`main` 全小寫；Java 區分大小寫。
- **字串串接與加法混淆**：`"x" + 3 + 4` 得 `"x34"`，而 `"x" + (3 + 4)` 得 `"x7"`。

§ 6 六、練習題

例題 1：印出固定格式輸出

寫一個程式，用**一個** `printf` 印出兩行：第一行 `Welcome to`，第二行 `Java!`。

> 1. 想清楚要幾行 → 需要一個換行 2. 選 `printf`，格式字串用 `%s` 代入文字 3. 換行用 `%n` 4. 把兩段文字依序當參數傳入

```
public class Q1 {
    public static void main(String[] args) {
        System.out.printf("%s%n%s%n", "Welcome to", "Java!");
    }
}
```

輸出：

```
Welcome to
Java!
```

> 易錯：忘記 `%n` 會兩段黏在一起；`%s` 個數要與參數個數相符。

例題 2：讀兩個整數並輸出總和

從鍵盤讀入兩個整數，印出它們的和。

> 1. `import java.util.Scanner;` 2. 建立 `Scanner`、提示並 `nextInt()` 讀兩次 3. 用 `+` 相加存進變數 4. `printf` 輸出結果

```
```

```java
import java.util.Scanner;
public class Q2 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter first integer: ");
        int n1 = input.nextInt();
        System.out.print("Enter second integer: ");
        int n2 = input.nextInt();
        int sum = n1 + n2;
        System.out.printf("Sum is %d%n", sum);
    }
}
```

易錯：漏 import；把 `nextInt()` 寫成 `nextint()`；忘記提示使用者導致看起來「卡住」。

例題 3：手算運算子優先序

不執行程式，寫出 `int r = 10 - 2 * 3 + 16 / 4 % 3;` 的 `r` 值與每一步。

> 1. 先做 `* / %`（由左而右） 2. `2 * 3`、`16 / 4`、再 `% 3` 3. 最後做 `+ -`（由左而右）

- `2 * 3 = 6` - `16 / 4 = 4`，`4 % 3 = 1` - 式子變成 `10 - 6 + 1` - `10 - 6 = 4`，`4 + 1 = 5` → **`r = 5`**

> 易錯：以為 `+` 先於 `/ %`；`%` 與 `* /` 同級要由左而右。

§ 7 七、自我檢核

- [ ] 能默寫 `public class` + `main` 的程式骨架，並說出檔名規則。
- [ ] 分得清 `print` / `println` / `printf` 的差別與 `%d %s %n` 用法。
- [ ] 會用 `import` + `Scanner` + `nextInt()` 讀整數。
- [ ] 能畫出變數記憶體盒子，解釋「讀取不破壞、寫入覆蓋」。
- [ ] 能正確手算含 `+ - * / %` 與括號的算術優先序。
- [ ] 分得清 `=` 與 `==`，會用六個關係/相等運算子寫 `if`。