

認識 Java 語法與建立類別

定義類別及其成員

1. 宣告類別

使用「class」關鍵字宣告類別：

語法：

```
[modifiers] class class_identifier {  
    class_code_block  
}
```

// [modifiers] 是修飾詞，非必要

範例：

```
class Shirt {  
  
}
```

- 一個最簡單的類別只要關鍵字「class」，加上類別名稱，最後加上大括號 {} 的符號，就可以通過編譯，其他內容其實都非必要

2. 宣告欄位

類別內可以宣告欄位 (fields)，但非必要：

(1) 來自物件的「屬性」

(2) 屬性有值 (value) 或狀態 (state)，建立欄位時必須宣告「型別」。說明該欄位屬於何種型態。整數？浮點數？或是其他？

(3) 宣告欄位時可以一併給值

範例：

```
public class Shirt {  
    public int size;        // 欄位size是類別的屬性，型別為整數(int)  
    public double price = 100.5;    // 欄位price是類別的屬性，型別為雙精度浮點數(double)  
}
```

3. 宣告方法

類別內可以宣告方法 (methods) · 但非必要：

(1) 來自物件的「行為」，需描述其「內容」

(2) 執行後，結果可分成兩種：

- 有回傳 (return) 結果，必須宣告其結果的「型別」
- 沒回傳 (return) 結果，必須宣告其結果為「void」

語法：

```
[ modifiers ] return_type method_identifier ([ arguments ]){  
    method_code_block  
}
```

範例：

```
public class Shirt {  
    public int size;  
    public double price = 100.5;  
    public double getPrice(){  
        return price;  
    }  
  
    public void display(){  
        System.out.println("size = " + size);  
        System.out.println("price = " + price);  
    }  
}
```

4. 使用註解

類別內可以使用註解 (comments) · 但非必要。程式執行時會自動略過

```
/* this is a class  
 * to show you what a class is  
 */  
public class Shirt {  
    // this is a field  
    public int size;  
    public double price = 100.5;5.  
    // this is a method  
    public double getPrice() {  
        return price;  
    }  
}
```

- 以「/*」開頭，以「*/」結尾的文字段落，都是註解，可以跨行

- 以「//」開頭，餘下到行尾都是註解，屬於單行註解，無法跨行

5. 程式裡的特殊符號

類別內常見的特殊符號意義如下：

符號	名稱	用途
{ }	大括號	一段程式碼
()	小括號	方法名稱後，可放參數
;	分號	一個敘述 (statement)的結束
,	逗號	分離變數和值
' '	單引號	用於字元
" "	雙引號	用於字串
//	雙斜線符號	用於註解

認識 Java 關鍵字

「關鍵字」在程式中已被編譯器賦予特殊意義，等同「保留字」，不可用在類別、欄位或方法的命名，否則編譯無法成功：

關鍵字都是以小寫開頭

abstract	continue	for	new	switch
assert	default	goto	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while

認識 main 方法

「main」方法是 Java 內特殊的方法，用來啟動 Java SE 的程式。此方法不屬於物件行為，類別可有可無。其宣告必須滿足以下條件：

- (1) 名稱必須是「main」，且都是小寫
- (2) 修飾詞必須有「public」
- (3) 修飾詞必須有「static」，所以不屬於物件的行為
- (4) 方法參數必須是「字串陣列」

語法：

```
public static void main (String[] args){  
    method_code_block  
}
```

- 方法宣告的修飾詞如 public 、static 是形容詞，其順序先後不影響 main 方法的意義

範例：

```
public class Shirt {  
    public static void main(String[] args){  
  
    }  
}
```

編譯並執行 Java 程式

(1) 建立可執行的程式：要有 main 方法，不然無法執行

(2) 編譯並執行程式：

- 程式碼開發後，可以使用 Java 的編譯器 (Java Compiler，亦即執行檔 javac.exe，位於 java 安裝目錄的 bin 資料夾內) 進行編譯 (compile) 的工作，編譯後產生 *.class 檔案：

語法：

```
javac 程式檔.java
```

- 完成編譯後，可以使用 Java 的執行檔 (java.exe，位於 java 安裝目錄的 bin 資料夾內) 執行程式：

語法：

```
java 程式編譯檔 (不能加 class 的副檔名)
```

- 執行的程式中，必須有 main 方法作為程式進入點，否則會出錯。

必須注意的是，Java 的程式檔案名稱和檔案內定義的 class 有若干關係：

- a. 程式碼檔案副檔名必須是「.java」，否則編譯器將拒絕編譯

b. 程式碼檔案內可以定義多個「非 public」的 class，檔案名稱不一定要和這些 class 名稱相同。編譯後，檔案內宣告的每個 class

都會各自產生附檔名是「.class」的編譯檔

c. 程式碼檔案內若有 public 的 class，此時一個程式碼檔案內只允許有一個 public class，且檔名必須和該 public class 的名稱相同