

第二章

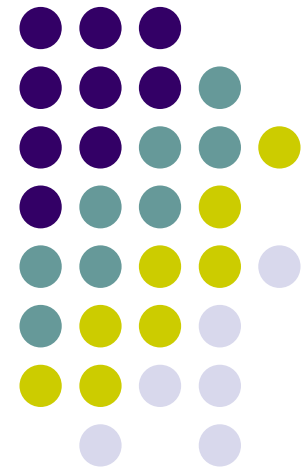
簡單的Java程式

認識Java的基本語法

認識Java的識別字與關鍵字

學習如何在Java的環境裡偵錯

學習如何提高程式的可讀性



簡單的 Java 程式

2.1 一個簡單的例子



- 下面的程式碼可印出兩行字串：

```
01 // app2_1, 簡單的 Java 程式
02 public class app2_1
03 {
04     public static void main(String args[]) // main() method, 主程式開始
05     {
06         int num; // 宣告整數 num
07         num=2; // 將 num 設值為 2
08         System.out.println("I have "+num+" dogs"); // 印出字串及變數內容
09         System.out.println("You have "+num+" dogs, too");
10     }
11 }
```

public是用來表示該類別為共有

註解以「//」記號開始，至該行結束

類別，後面接上類別名稱

// 定義 public 類別 app2_1

程式的起點

main()的主體

main() 函數在此結束

類別app2_1的定義在此結束

```
public class aaa
{
    ...
}

class bbb
{
    ...
}
```

aaa.java

若將類別宣告成public，則檔案名稱也必須命名為這個類別的名稱

/* app2-1 OUTPUT---

```
I have 2 dogs
You have 2 dogs, too
-----*/
```



類別

- Java程式是由類別（class）所組成
 - 下面的程式片段即為定義類別的範例：

```
public class myjava          // 定義public 類別 myjava
{
    ...
}
```

- public指的是對於類別的存取方式為共有
- 在完整的Java程式裡，至少需要有一個類別



大括號、區塊及本體

- 左大括號（{）為類別本體的開始，至右大括號（}）結束
- 指令敘述結束時，以分號「;」做結尾
- 指令的敘述不只一行時，以一對大括號（{ }）將這些敘述括起來，形成區塊（block）



程式區塊及本體

- 程式區塊與本體的範圍：

/* app2-2 OUTPUT---

變數 i 的值小於 8

i=5

-----*/

```
01 // app2_2, 本體與區塊
02 public class app2_2 // 定義 public 類別 app2_2
03 {
04     public static void main(String args[])
05     {
06         int i;
07         i=5;
08         if(i<8)
09         {
10             System.out.println("變數 i 的值小於 8"); // 印出字串內容
11             System.out.println("i="+i); // 印出變數
12         }
13     }
14 }
```

public 類別 app2_2 的本體

main() method 的本體

if 敘述的程式區塊



程式執行的起始點

- 每一個Java程式必須有一個main() method，而且只能有一個
- 通常main() method會是如下面的敘述片段：

```
public static void main(String args[]) // main() method, 主程式開始
{
    ...
}
```

- main() method之前必須冠上修飾子 public static void



變數使用的規則

- 變數的宣告：

- `int num;` // 宣告num為整數變數
- `int num, num1, num2;` // 同時宣告num, num1, num2為整數變數

- 變數的資料型態：

- `char` 字元 如 `'a'`、`'A'` 等
 - `String` 字串 如 `"Have a nice day!!"` 等
 - `int` 整數
 - `long` 長整數
 - `short` 短整數
 - `float` 單精度浮點數
 - `double` 倍精度浮點數
- 如12、-27 等
- 如12.762、-37.483 等



變數名稱與其限制

- 請依個人的喜好來決定變數的名稱
 - 通常變數會以其所代表的意義來取名
 - 不能使用到關鍵字
 - 簡單的變數名稱會增加閱讀及除錯的困難度
- 變數名稱可以是英文字母、數字或底線
 - 名稱中不能有空白字元
 - 第一個字元不能是數字
 - 變數有大小寫之分



literal

- 出現在Java中的各種資料，包括字元、字串、數值、常數等，通稱為literal
- literal是
 - 數字（整數、浮點數）與
 - 文字（字元、字串）的總稱
- 有些作者把它譯為「文數字」



變數的設值 (1/2)

- 宣告的時候設值

```
int num=2;
```

```
// 宣告變數，並直接設值
```

- 宣告後再設值

```
int num1,num2;
```

```
// 宣告變數
```

```
char ch;
```

```
num1=2;
```

```
// 設值給變數
```

```
num2=30;
```

```
ch = 'm';
```



變數的設值 (2/2)

- 在程式中適當的位置宣告變數並設值

```
for (int num=1; num<=10; num++)
```

```
{
```

```
...
```

```
}
```

需要用到變數時，再行宣告與設值



為什麼要宣告變數

- 宣告變數有許多好處，諸如：
 - 方便編譯器找到錯誤的變數名稱
 - 避免變數名稱打錯（如數字0與英文字母O）
 - 除錯容易
 - 增加程式的可讀性
 - 方便管理變數



println() method

- 使用println() 印出變數與字串：

```
01 // app2_3,使用 println()  
02 public class app2_3 // 定義 public 類別 app2_3  
03 {  
04     public static void main(String args[]) // main() method,主程式開始  
05     {  
06         int num=2;  
07         System.out.println("I have "+num+" dogs"); // 印出變數及字串內容  
08     }  
09 }
```

/* app2-3 OUTPUT---

I have 2 dogs

-----*/



有引數的情形

- 下面的範例可接收兩個引數 "Tom" 與 "Jerry"：

```
01 // app2_4,有引數的情形
02 public class app2_4
03 {
04     public static void main(String args[])
05     {
06         System.out.println( + " & " + );
07     }
08 }
```

"Tom"由args[0]接收

"Jerry"由args[1]接收

/* app2-4 OUTPUT---

Tom & Jerry

-----*/



識別字 (identifier)

- 變數、常數、類別或者是method的名稱為識別字
- 識別字的習慣命名原則：

識別字	命名原則	範例
常數	全部字元皆由英文大寫字母及底線組成	PI MAX_NUM
變數	英文小寫字母開始，若由數個英文單字組成，則後面的英文字由大寫起頭，其餘小寫	radius circleArea myPhoneNumber
method	英文小寫字母開始，若由數個英文單字組成，則後面的英文字由大寫起頭，其餘小寫	show addNum mouseClicked
類別	英文大寫字母開始，若由數個英文單字組成，則後面的英文字由大寫起頭，其餘小寫	Caaa CCustomer MaxSize



關鍵字 (keyword)

- 關鍵字是編譯程式本身所使用的識別字
- Java提供的關鍵字如下：

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



程式錯誤的分類

- 語法錯誤（syntax error）
 - 程式含有不合語法的敘述，它無法被編譯程式翻譯
- 語意錯誤（semantic error）
 - 語意錯誤（又稱邏輯錯誤），就是程式的執行結果非我們所願



語法錯誤

- 找出程式中的錯誤

```
01 // app2_5,有錯誤的程式
02 public class app2_5                // 定義 public 類別 app2_5
03 {
04     public static void main(String args[])
05     {
06         int num1=2;                // 宣告整數變數 num1，並設值為 2
07         int num2=3;                // 宣告整數變數 num2，並設值為 3
08
09         System.out.println("I have "+num1+" dogs");
10         System.out.println("You have "+num2+" dogs")
11     }
12 }
```

/* app2-5 OUTPUT 除錯後的結果 ---

I have 2 dogs
You have 3 dogs

-----*/



語意錯誤

- 下面是語意錯誤的程式：

/* app2-6 OUTPUT---

I have 2 dogs
You have 3 dogs
We have -1 dogs

-----*/

```
01 // app2_6,語意錯誤的程式
02 public class app2_6 // 定義 public 類別 app2_6
03 {
04     public static void main(String args[])
05     {
06         int num1=2;    // 宣告整數變數 num1，並設值為 2
07         int num2=3;    // 宣告整數變數 num2，並設值為 3
08
09         System.out.println("I have "+num1+" dogs");
10         System.out.println("You have "+num2+" dogs");
11         System.out.println("We have "+(num1-num2)+" dogs");
12     }
13 }
```



提高程式的可讀性 (1/2)

- 將程式碼縮排，可提高可讀性

```
01 // app2_7, 有縮排的程式碼
02 public class app2_7
03 {
04     public static void main(String args[])
05     {
06         int i;
07
08         for(i=1;i<=3;i++)    // 呼叫 for 迴圈
09         {
10             System.out.print("i="+i+", ");
11             System.out.println("i*i="+i*i);
12         }
13     }
14 }
```

```
01 // app2_8, 沒有縮排的程式碼
02 public class app2_8{
03     public static void main(String args[]){
04         int i;for(i=1;i<=3;i++){
05             System.out.print("i="+i+", ");
06             System.out.println("i*i="+i*i);}}}
```

/* app2-7, 2-8 OUTPUT---

i=1, i*i=1

i=2, i*i=4

i=3, i*i=9

-----*/



提高程式的可讀性 (2/2)

- 將程式碼加上註解，有助於程式的閱讀與偵錯

```
// app2 9, examples  
// created by wien hong
```

} 以「//」符號註解

```
/* This paragraph demonstrate the capability  
   of comments used by Java */
```

} 於「/*」和「*/」符號
之間的文字均是註解