

# 第二章

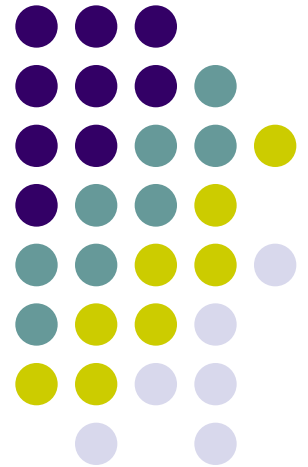
## 簡單的Java程式

認識Java的基本語法

認識Java的識別字與關鍵字

學習如何在VSCode的環境裡偵錯

學習如何提高程式的可讀性





# 簡單的 Java 程式

- 下面的程式碼可印出一行字串：

```
01 // Ch2_1, 簡單的 Java 程式
02 public class Ch2_1{      // 定義 public 類別 Ch2_1
03     public static void main(String[] args){    // main() 函數，主程式開始
04         int num=2;        // 宣告整數 num
05         System.out.println(num+" cats are running"); // 印出字串及變數內容
06     }
07 }
```

```
/* output-----
2 cats are running
-----*/
```



# 程式解說(1/4)

- 「//」記號開始至該行結束，用來表示註解的文字
- `class`為「類別」之意，後面接上類別名稱
- `public`是用來表示該類別為公有
- 類別若是宣告成`public`，檔案名稱必須和這個類別名稱相同

```
public class Ch2_1{  
    ...  
}  
  
class myClass{  
    ...  
}
```

Ch2\_1.java

如果將類別宣告成`public`，則檔案名稱也必須命名為這個類別的名稱



# 程式解說(2/4)

- 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 Ch2_1    // 定義public類別Ch2_1
{
    // 類別的本體，一般會縮排3~4個空格
}
```

將左大括號獨立成一行 ——— {

右大括號 ——— }

└—— 左大括號和右大括號垂直對齊

- 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
- 每個Java程式最多只能有一個main()



# 程式解說(3/4)

- `main()` 之前必須冠上修飾子 `public static void`
  - `public`屬性是將`main()` 設成公有
  - `static`是把`main()` 宣告成靜態函數，使得在程式一啟動時，便可自動的執行`main()`
  - `main()` 沒有傳回值，所以要加上`void`
- `main()` 的引數`String[] args` 表示引數由字串型別的陣列`args` 存放

按此處可執行或偵錯程式

```
1  // Ch2_1, 簡單的Java程式
2  public class Ch2_1{           // 定義public類別Ch2_1
3      public static void main(String[] args){ // main() 函數,主程式開始
4          int num=2;             // 宣告整數num
5          System.out.println(num+" cats are running"); // 印出變數及字串內容
6      }
7  }
```



# 程式解說(4/4)

- 「//」記號開始至該行結束，用來表示註解的文字
- `class`為「類別」之意，後面接上類別名稱
- `public`是用來表示該類別為公有
- 類別若是宣告成`public`，檔案名稱必須和這個類別名稱相同

```
public class Ch2_1{  
    ...  
}  
  
class myClass{  
    ...  
}
```

Ch2\_1.java

如果將類別宣告成`public`，則檔案名稱也必須命名為這個類別的名稱



# 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

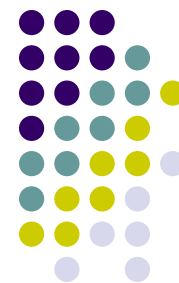
- 請依程式的需求來決定變數的名稱
  - 通常變數會以其所代表的意義來取名
  - 不能使用到關鍵字
  - 簡單的變數名稱會增加閱讀及除錯的困難度
- 變數名稱可以是英文字母、數字或底線
  - 名稱中不能有空白字元（錯誤：my cats）
  - 第一個字元不能是數字（錯誤：2cats）
  - 變數有大小寫之分（mycats與MyCats是不同變數）





# literal

- Java把諸如整數12與字元 'a' 等，一眼便能看出其內容的數值稱為literal
- literal是
  - 數字（整數、浮點數）與
  - 文字（字元、字串）的總稱
- 我們偏好把literal翻成「字面值」，因為單看literal的字面，就可以知道它的內容為何



# 變數的設值

- 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  
num1=2;              // 設值給變數  
num2=30;
```

- 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() 函數

- 使用println() 印出變數與字串：

```
01 // Ch2_2, 變數的宣告
02 public class Ch2_2{    // 定義類別 Ch2_2
03     public static void main(String[] args){
04         int a;          // 宣告變數 a，但不設定初值
05         int b=4, sum;    // 宣告 b 和 sum 兩個變數，但只將 b 的值設為 4
06
07         a=5;            // 設定 a 的值為 5
08         sum=a+b;        // 將 sum 的值設為 a+b
09         System.out.println("sum= "+sum); // 印出 sum 的值
10     }
11 }
```

執行結果：

```
sum= 9
```



# 使用println()

- 下面的範例可印出字串及變數內容：

```
01 // Ch2_3, 使用 println()函數
02 public class Ch2_3{    // 定義類別 Ch2_3
03     public static void main(String[] args){
04         int a=2, b=5;    // 宣告變數 a 和 b
05         System.out.println(a+"*"+b+"="+a*b); // 印出運算結果
06     }
07 }
```

印出 2      印出 \* 號      印出 5      印出 = 號      印出 10

執行結果：

2\*5=10



# 識別字 (identifier)

- 變數、常數、類別或者是函數的名稱為識別字
- 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
函數	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。函數和變數的命名方式相同，不同的是函數名稱後面會加上()	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	strictfp
switch				



# 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
  - 編譯器可以檢查出語法錯誤
- 語意錯誤（semantic error）
  - 又稱邏輯錯誤，就是語法正確，但執行結果不對
  - 編譯器無法檢查出語意錯誤





# 語法錯誤(1/2)

- 找出程式中的錯誤

```
01 // Ch2_4,有錯誤的程式
02 public class ch2_4{
03     public static void main(String[] args){
04         int num1=2;           // 宣告整數變數 num1，並設值為 2
05         int num2=3;           // 宣告整數變數 num2，並設值為 3
06
07         System.out.println("I have "+num1+" dogs");
08         System.out.println("You have "+num2+" dogs")
09     )
10 }
```

- 「問題」窗格內可以看到相關的錯誤訊息





# 語法錯誤(2/2)

- 於「終端機」窗格內查看錯誤的訊息

The screenshot shows an IDE interface with a 'Terminal' window (highlighted with a red dashed box) displaying a Java compilation error. The error message is: 'Exception in thread "main" java.lang.Error: Unresolved compilation problems: Syntax error on token ")", ; expected Syntax error, insert "}" to complete ClassBody at ch2\_4.main(Ch2\_4.java:9)'. To the right of the terminal window, there are two numbered annotations: '1. 按下 Proceed 按鈕' (Press the Proceed button) pointing to the 'Proceed' button in the 'Build failed' dialog box, and '2. 編譯後得到的錯誤訊息' (Error message after compilation) pointing to the error text in the terminal. The 'Build failed' dialog box also shows the source as '來源: Debugger for Java (延伸模組)'.

問題 3 輸出 偵錯主控台 終端機 Run:

kospaceStorage\b47a781450edf7edde535065ef8df634\redhat.java\jdt\_ws\Ch2\_4\_a57a3897\bin' 'ch2\_4'

Exception in thread "main" java.lang.Error: Unresolved compilation problems:

Syntax error on token ")", ; expected  
Syntax error, insert "}" to complete ClassBody

at ch2\_4.main(Ch2\_4.java:9)

2. 編譯後得到的錯誤訊息

1. 按下 Proceed 按鈕

Build failed, do you want to continue?

來源: Debugger for Java (延伸模組)

Proceed Fix... Cancel

```
/* output-----  
I have 2 dogs  
You have 3 dogs  
-----*/
```



# 語意錯誤

- 下面是語意錯誤的程式：

```
01 // Ch2_5,語意錯誤的程式
02 public class Ch2_5{
03     public static void main(String[] args) {
04         int a=2;        // 宣告整數變數 a，並設值為 2
05         int b=3;        // 宣告整數變數 b，並設值為 3
06         System.out.println("a+b="+a+b);
07     }
08 }
```

執行結果：

a+b=23



# 提高程式的可讀性 (1/4)

- 未將程式碼縮排，閱讀起來較為困難

---

```
01 // Ch2_6, 沒有縮排的程式碼
02 public class Ch2_6{    // 定義 public 類別 Ch2_6
03     public static void main(String[] args){
04
05         System.out.println("Hello Java!");
06         System.out.println("Hello world!");
07     }
08 }
```

執行結果：

```
Hello Java!
Hello world!
```

---



# 提高程式的可讀性 (2/4)

- 將程式碼縮排，可提高可讀性

01 // Ch2\_7, 有縮排的程式碼

02 public class Ch2\_7{

03 public static void main(String[] args){

04 System.out.println("Hello Java!");

05 System.out.println("Hello world!");

06 }

07 }

程式碼往內縮，因為這兩行是屬於 main() 函數的內容

執行結果：

Hello Java!

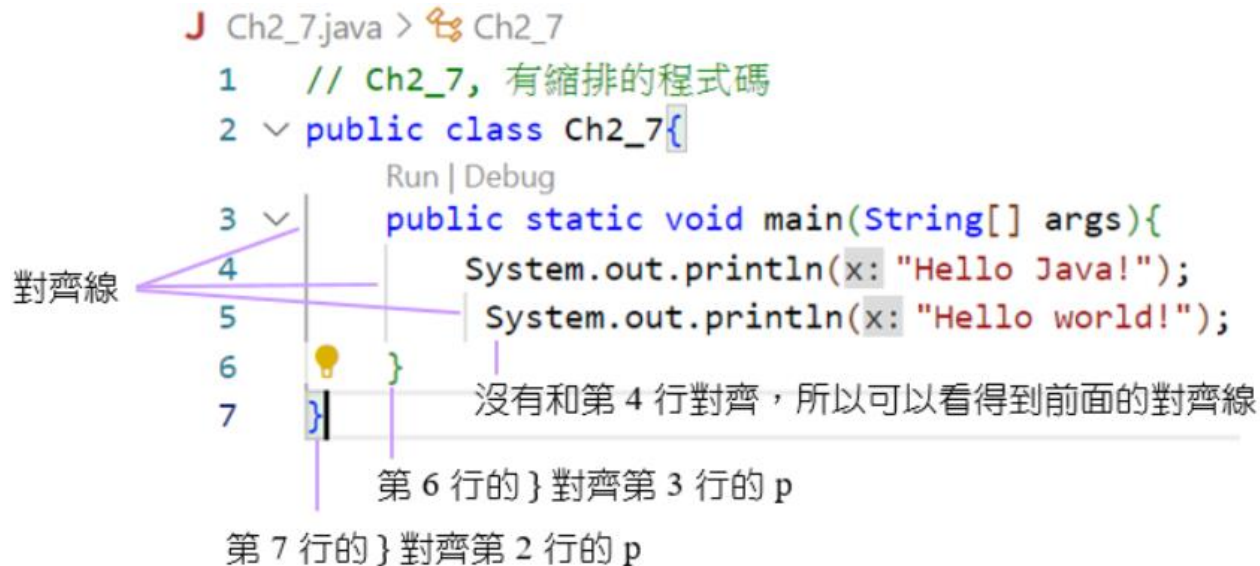
Hello world!

程式碼往內縮，因為整個 main() 是隸屬於 Chp2\_7 類別



# 提高程式的可讀性 (3/4)

- 善用VSCode提供格式化文件的功能





# 提高程式的可讀性 (4/4)

- 將程式碼加上註解，有助於程式的閱讀與偵錯

```
// Ch2_8, examples  
// created by wien hong
```

} 以「//」符號註解

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

} 於「/\*」和「\*/」符號之間的文字均是註解



-The End-