

MultiMedia Systems Laboratory



CHAPTER 1

Java的第一步(ch1)

Java的基本知識(ch2)

Java的變數(ch3)

Java的運算式和運算子(ch4)





本節介紹

- Java的特點
- Java程式的設計與執行
- 常見的Java開發平台
- 認識開發環境-Eclipse

關鍵詞彙

- ◆ 多執行緒
- ◆ 垃圾收集
- ◆ 例外處理
- ◆ 編譯器
- ◆ 直譯器
- ◆機器語言

1.1 Java的第一步



■ Java的特點(1/2)

✓ 嬌小且完美的語言

• Java的語法簡潔、勻稱,容易撰寫且易於維護

✓ 具物件導向的功能

- Java具有物件導向的特點
- 它是全新的語言,因而可以將物件導向的特性發揮到極致

√ 完全支援網際網路

- Java所撰寫的程式很容易地在Browser裡呈現
- 可藉此讓所有能夠上網的電腦都能執行Java程式

✓ 通用的語言

• 凡是C/C++所能做到的事, Java也都能做到

1.1 Java的第一步



■ Java的特點(2/2)

✓ 跨平台的語言

• 可在不修改程式碼的情況下,便能在不同的作業系統執行

✓ 具有豐富的函數庫

- Java語言背後有龐大的函數庫支撐
- 這些函數庫包含了繪圖函數庫、圖形使用者介面函數庫 (graphical user interface) 、網路設計函數庫...等等

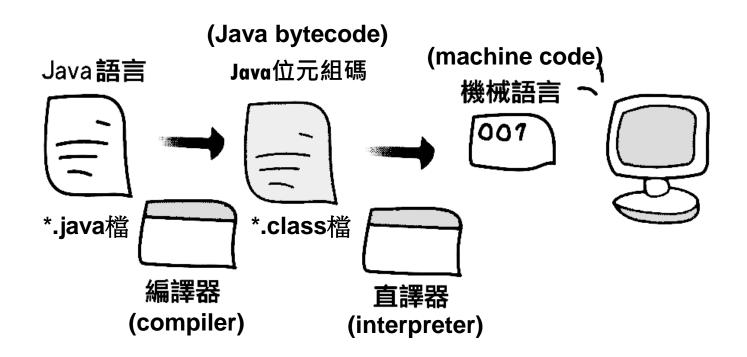
√ 特殊的處理機制

- 「多執行緒」機制可在同一時間執行不同的程序
- 「垃圾收集」機制可將無用的變數所佔用之記憶體釋放
- 「例外處理」機制可依情況拋出例外,使得程式不會因此中斷執行



■ Java程式的設計與執行

Java可以透過編譯器 (compiler) 及直譯器 (interpreter),將寫好的 Java程式翻譯成機器語言 (machine code),並特過這種機器語言來讓電腦進行實際的處理。







■ 常見的Java開發平台:

- ✓ Eclipse(本課程使用)
- ✓ NetBeans
- ✓ JDeveloper
- ✓ IntelliJ IDEA
- ✓ Jbuilder













☑ 認識開發環境-Eclipse 編譯與執行程式 Java - chapter1_1/src/chapter1_1.java - Eclipse File Edit Refactor Source Navigate Search Project Run Window Help Quick Access 🖺 🕍 Java EE 🐉 Java 🕸 Debug 🖺 Package Explorer 🗯 📄 🥞 🦫 class chapter1 1 b 📂 chapter1_1 chapter10 1 public static void main(String[] args) chapter10_2 chapter10_3 System.out.println("歐巡使用Java!"); 專案目錄 chapter10_9 程式輸入區塊 chapter11_1 b septer11_2 chapter11_3 chapter11 4 chapter11_5 chapter11_6 chapter11_7 chapter11_8 chapter11_9 ⊜ chapter12_1 chapter12 2 chapter12_3 chapter12_4 ⊜ chapter13_1 chapter13_2 chapter13_3 b ab chapter13_4 執行結果 🕎 Console 🛭 🔝 Problems @ Javadoc 📵 Declaration 🔡 Outline 🐚 LogCat 💖 Debug 👙 Call Hierarchy chapter13_6 <terminated > chapter1_1 [Java Application] C:\Program Files (x86)\Java\jre7\bin\javaw.exe (2013/6/12 下午8:42:49) schapter13_7 chapter14_1 chapter14_10 chapter14 2 ⊜ chapter14_3 b 📂 chapter14 4 ⊜ chapter14_6 chapter14_7







本節介紹

- 輸出執行結果到螢幕上
- 檢視程式的執行流程
- main()方法
- 以敘述句為單位執行程式
- 註解 (comment)
- 探討class (類別)
- 何謂常數 (literal)
- 跳脫字元 (escape sequence)
- 數值常數

關鍵詞彙

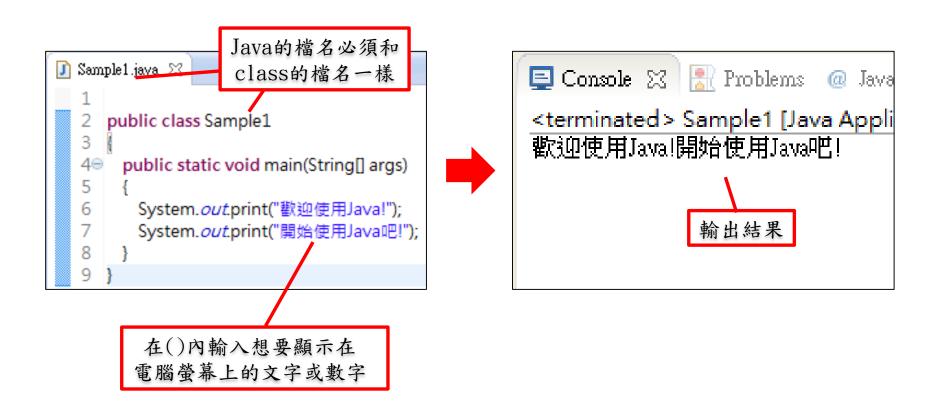
- ◆ 換行效果
- **♦** main
- ◆ 註解
- ◆ 類別名稱
- ◆ 字元常數
- ◆ 字串常數
- ◆ 整數常數
- ◆ 浮點數常數
- ◆ 跳脫字元





■輸出執行結果到螢幕上(1/2)

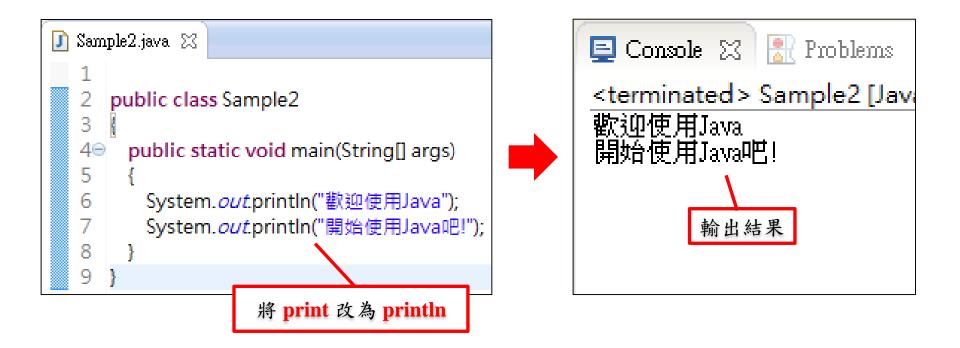
如果要將執行結果的文字和數值都「輸出」到電腦螢幕時,程式要怎麼寫?





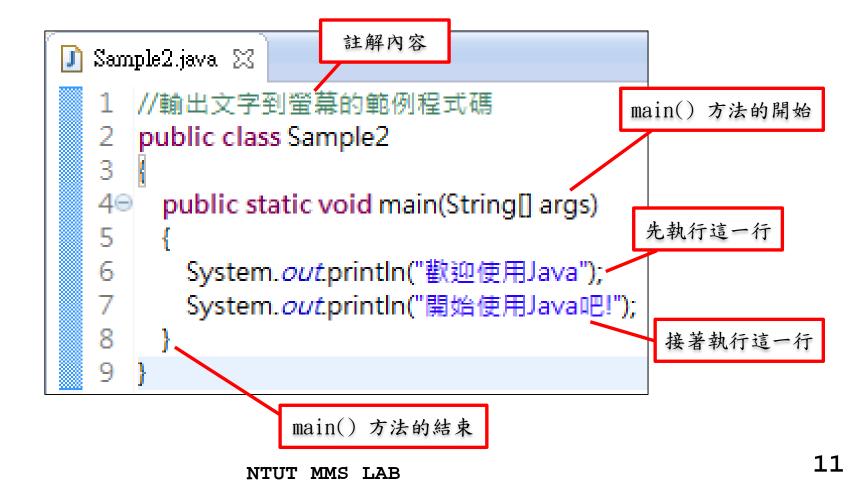
■輸出執行結果到螢幕上(2/2)

如果要將執行結果的文字和數值都「輸出」到電腦螢幕時,並且希望輸出 結果產生換行效果,程式要怎麼寫?



☑ 檢視程式的執行流程

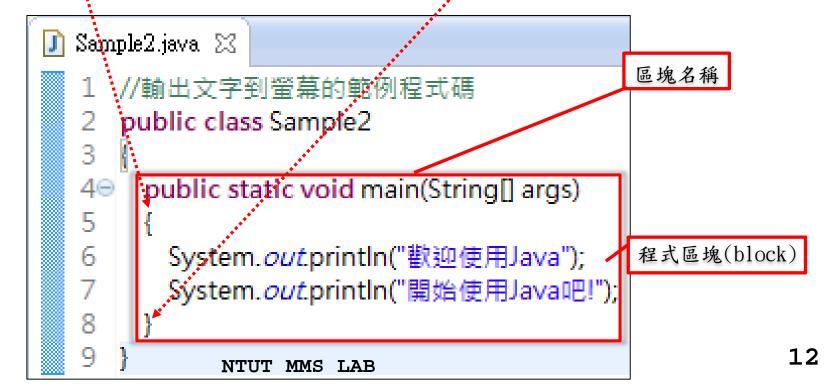
仔細檢視程式碼是下達什麼樣的指令給電腦,讓執行結果輸出到電腦螢幕上。





■ main()方法

- ✓ 程式都是以 main() 做為程式主體開端並且逐一往下執行,在 main()這一行的大括號({)之後,隔數行又出現相對的結束大括號(})。





🧧 以敘述句為單位執行程式

執行敘述句時會根據程式的先後順序,從最前面開始一行接著一行執行。 敘述句尾端要加上分號「;」。

```
4回 public static void main(String[] args)
5 {
6 System.out.println("歡迎來到Java!");
7 System.out.println("開始學Java吧!");
8 }
```

MMS Lab

1.2 Java 的基本知識

些 註解 (comment)

程式註解有二種方法:

- ✓ 『//』:後面的文字與程式的執行結果無關,您可以在//後面輸入程式註 解或是任何您想要輸入的文字。
- ✓ 『/**/』: 所有包圍在/**/的文字都算是註解的一部份。所以只要利用/* */將註解文字包圍起來,即使這些註解文字跨越2行以上也沒有問題。



響探討class (類別)

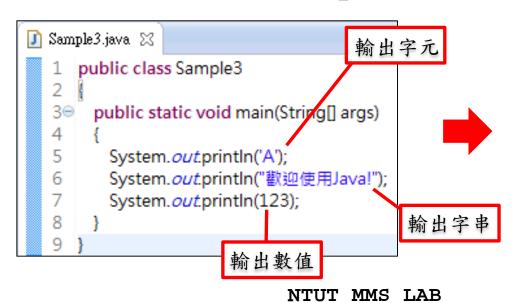
撰寫Java程式時都會在程式的一開始加上「class」字樣,做為程式區塊的 起始。Java程式當中,最少必須有1個class(類別),連接在class後面的名稱 稱為「類別名稱」(class name),類別名稱可以自由命名。

MMS Lab

1.2 Java 的基本知識

■何謂常數 (literal)

- ✓ 『字元常數』:字元是由單一字母所構成,如 'A',只有1個字母,所以字元的外圍必須用單引號('')圍起來。
- ✓ 『字串常數』:字串是由2個或2個以上的字元連接在一起所構成的, 所以字串的外圍必須用雙引號("")圍起來。
- ✓ 『整數常數、浮點數常數』







■ 跳脫字元 (escape sequence)(1/2)

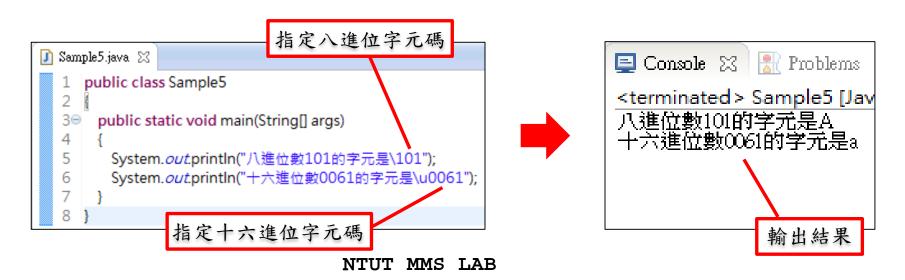
跳脫字元(escape sequence),是指有些單一字元無法表示的特殊文字,這些特殊文字的前面必須加上「\」符號,與原來的字元組合後產生新的意義。

跳脫字元	說明
\b	Backspace
\t	tab
\n	換行
\f	換頁
\r	return
` '	6
\ "	66
\\	\
\000	8進制000字元碼
\uhhhh	16進制hhhh字元碼

17



■ 跳脫字元 (escape sequence)(2/2)





■ 數值常數

如果要將常數仔細劃分的話,又可分為下列幾種資料形態:

- ✓ 『整數 (integer literal)』(Ex: 1、3、100)
- ✓ 『浮點數 (floating-point literal)』(Ex: 2.1、3.14、5.0)

數值並不需要像字元或字串一樣,需要以『"』或『'』框起來,除了可以用一般方式標示整數常值之外,還有很多種不同的寫法。例如可以使用八進位(0x)與十六進位(0x)來表示數值。







本節介紹

- 關於變數
- 變數的資料型態
- 識別字
- 宣告變數
- 使用變數
- 從鍵盤輸入數字與字串

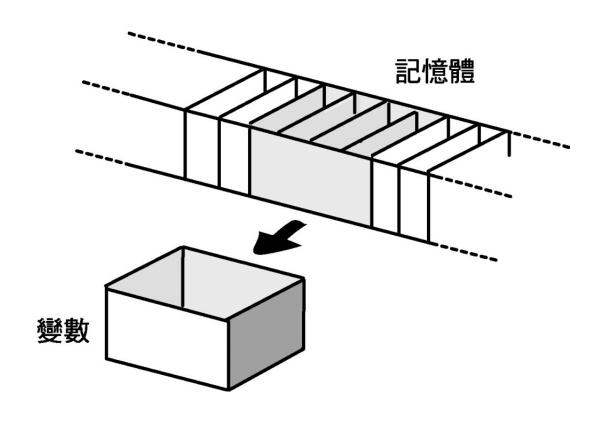
關鍵詞彙

- ◆ 變數
- ◆ 資料型態
- ◆ 識別字



■ 關於變數

所謂的變數,指的就是**儲存在記憶體內的數值或資料**,這種儲存在記憶體內的現象與人類用紙筆記下特定資料的用意相似,都是為了必要時的取用。





■ 變數的資料型態

變數可以用來「存放」特定的文字或數字,這些文字或數字有下列幾種不同的資料型態(data type),下列是Java的變數可以使用的基本資料型態:

名稱	資料的範圍
boolean	布林值,true或false
char	2個byte的文字資料,表示範圍從\u0000~\uffff
byte	1個byte的整數,表示範圍從-128~127
short	2個byte的整數,表示範圍從-32768~32767
int	4個byte的整數,表示範圍從-2147483648~2147483647
long	8個byte的整數,表示範圍從- 9223372036854775808~9223372036854775807
float	4個byte的單精度浮點數
double	8個byte的雙倍精度浮點數

Java字元以 Unicode做編碼



≝ 識別字

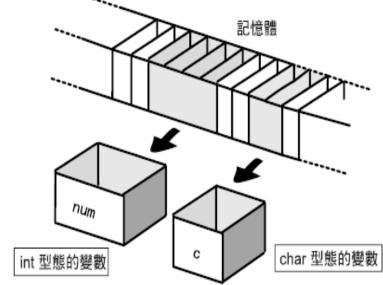
- ✓ 在Java程式中如果要使用某變數,必須先針對該變數進行下列2項設定工作 :
 - 指定變數的『名稱』:您可以使用「識別字」做為變數名稱,例如num就是一個 合法的變數名稱。
 - 指定變數的『型態』:使用變數之前,一定要先確定變數的資料型態,資料型態 確定之後,該變數才能夠「記住」該類資料型態內容。
- ✓ 建立新的識別字時仍然要遵照下列的規則:
 - 一般而言,我們會使用**英文字母、阿拉伯數字、底線連接、\$等符號**做為識別字 名稱(例如: car123, hello_world, \$_\$)。
 - 識別字並沒有長度的限制。
 - Java既有的關鍵字(keyword)不能做為識別字名稱(例如最常見的2個關鍵字main和class)。
 - 識別字名稱不能用阿拉伯數字做為開頭(例如: 123car)。
 - 英文字母的大小寫有區別(例如: Abc 不等於 ABC)。



■ 宣告變數

決定好變數的名稱和資料型態之後,必須在做變數的宣告,才能順利使用 變數。以下是變數宣告的方式:

```
語法 變數的資料型態 識別字;
```

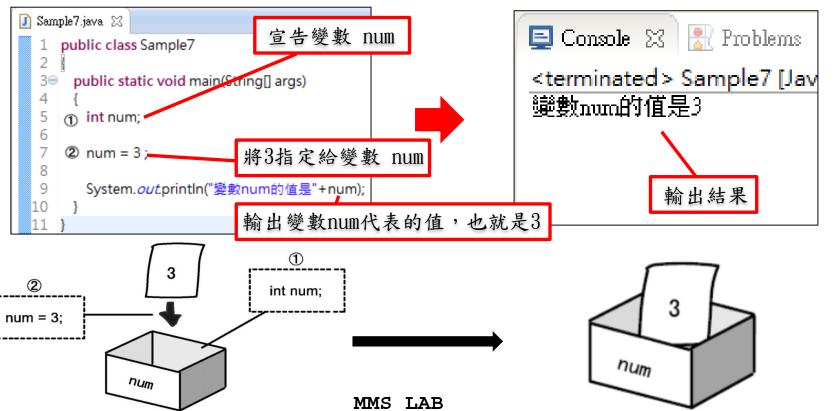


```
3 public static void main(String[] args)
4 {
5 int num; //int 型態的變數num
6 char c; // char 型態的變數 c
7 double db, dd; // double 型態的變數 db \ dd
8 }
```



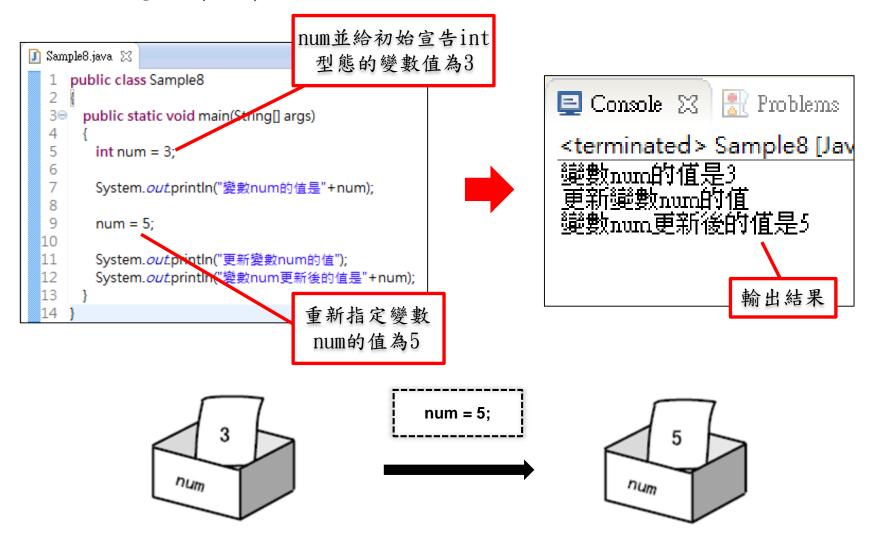
■ 使用變數(1/3)

- ✓指定「數值或其他資料」給某變數
 - 變數一經宣告之後,就可以指定 (assignment) 特定的內容給變數, 也就是讓變數真正「記住」您所指定的內容。實際上指定內容給變 數時,必須使用「=」。



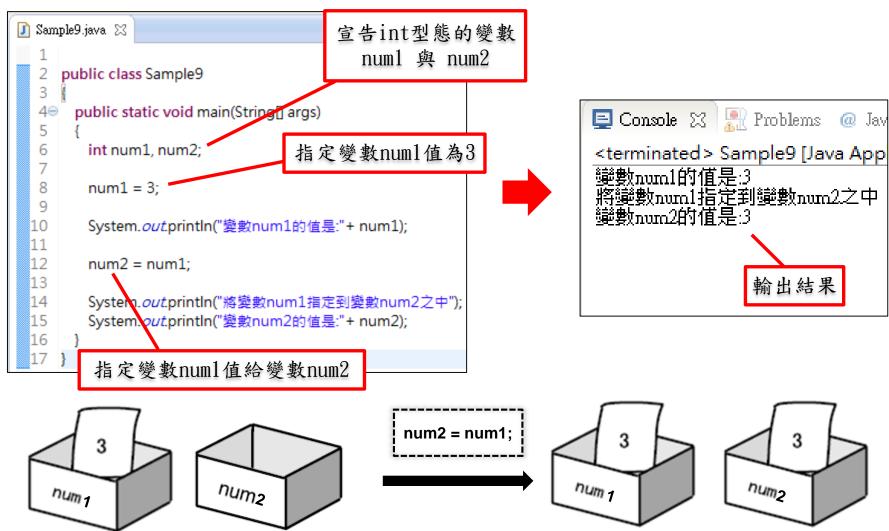


■ 使用變數(2/3)





■ 使用變數(3/3)



MMS LAB

NTUT



₫ 從鍵盤輸入數字與字串



的strl轉換成int型別

```
🚺 Sample10.java 💢
                                                 建立BufferedReader
  1⊕ import java.io.BufferedReader;□
                                                類別並指定給物件brl
     public class Sample10
  6
      public static void main(String[] args) throws IOException
        System.out.println("請輸入一個垄數");
 10
 11
        BufferedReader br1 =
 12
           new BufferedReader(new InputStreamReader(System.in));
 13
 14
        String str1 = br1.readLine();
 15
 16
        int num = Integer.parseInt(str1);
 17
 18
        System.out.println("您輸入的數字是:"+ num);
                                             使用BufferedReader類別的
 19
                                             readLine實體方法回傳使用
 20
        System.out.println("請輸入字串");
                                              者在按下Enter鍵之前的所
 21
 22
                                             有字元並放到strl
        BufferedReader br2 =
 23
           new BufferedReader(new InputStreamReader(System.in));
 24
 25
        String str2 = br2.readLine();
 26
 27
        System.out.println("剛剛輸入的字串是:"+ str2);
                                               使用Interger類別內的類別
 28
                                               方法parseInt將string型別
 29
```

小知識

class(類別)用來定義物件導向程式設計中的物件(object)的格式, 像是這裡提到的BufferedReader、 InputStreamReader、Integer等都是一種class(類別)。

Console ☆ March Problems @ Javadoc <a hr

輸出結果

小知識

要使用類別時通常會使用new運算子來建立實體物件,之後我們便可以透過物件來使用實體方法。如左邊brl物件的readLine()方法就是一個實體方法。Integer類別的parseInt()方法沒有經過new建立就可以使用,此種方法稱為類別方法。







本節介紹

- 運算式與運算子
- 進行各種運算
- 運算子的種類
- 遞增運算子與遞減運算子
- 位移運算子
- 運算子的執行優先順序
- 資料型態的轉換

關鍵詞彙

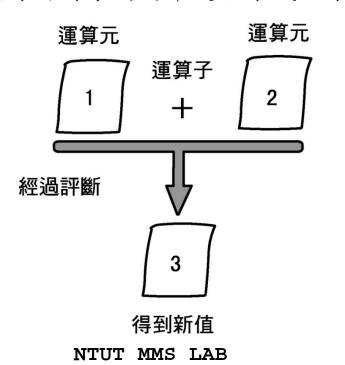
- ◆ 運算子
- ◆ 運算元
- ◆ 前置&後置
- ◆ 位移運算
- ◆ cast運算子

MMS Lab

1.4 運算式與運算子

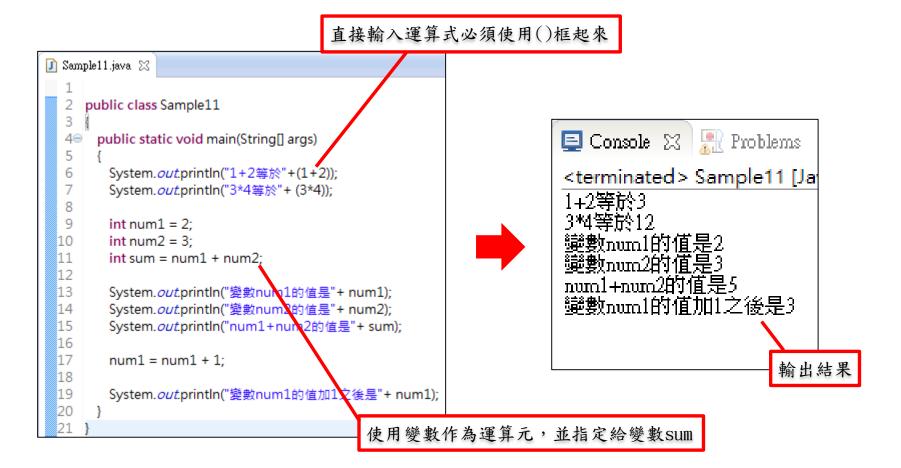
■ 運算式與運算子

- ✓ Java運算式都是由下列2種元素所構成:
 - 運算子(operator)
 - 負責演算的加減乘除等符號。
 - 運算元(operand)
 - 計算執行的對象(通常是數字或文字)。





■ 進行各種運算





■ 運算子的種類

Java語言中,除了加號(+)這個運算子之外,其他尚有很多運算子。主要的運算子種類有下列幾種:

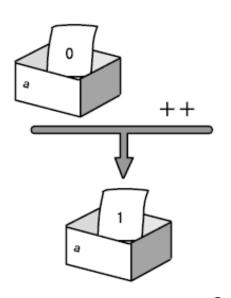
運算子	說明	運算子	說明
+	加法或自串串接時使用	++	遞增
_	減法		遞減
*	乘法	>	大於
/	除法	>=	大於等於
%	餘數	<	小於
+	正號	<=	小於等於
-	負號	==	相等
~	補數	!=	不等於
&	AND位元運算	instanceof	資料型態比較
	OR位元運算	!	邏輯NOT
^	XOR位元運算	&&	邏輯AND
<<	位元左移		邏輯OR
>>	位元右移	?:	(條件式) =
>>>	位元右移並補零	new	建立物件使用

MMS Lab

1.4 運算式與運算子

■ 遞增運算子與遞減運算子(1/2)

- ✓遞增運算子
 - 遞增運算子 (increment operator) ,顧名思義就是會讓變數的值加1。
- ✓遞減運算子
 - 2個減號連續在一起形成的「--」則形成所謂的遞減運算子(decrement operator),遞減運算子會讓變數的值減1。
- √遞增(或遞減)運算子的「前置」與「後置」







₫ 遞增運算子與遞減運算子(2/2)

```
🚺 Sample12.java 🖂
     public class Sample12
                                                    進行各種運算
  3
      public static void main(String[] args)
  5
        int num1 = 10:
        int num2 = 5;
                                                                         📃 Console 💢 🦳 Problems

    Javadoc

                                                                         <terminated > Sample12 [Java Application]
        System.out.println("num1和num2的各種運算:");
        System.out.println("num1+num2等於"+(num1+num2));
                                                                         num1和num2的各種運算:
 11
        System.out.println("num1-num2等於"+(num1-num2));
                                                                         num1+num2等於15
 12
        System.out.println("num1*num2等於"+(num1*num2));
                                                                         num1-num2等於5
 13
        System.out.println("num1/num2等於"+(num1/num2));
                                                                         num1*num2等於50
 14
15
        System.out.println("num1%num2等於"+(num1%num2));
                                                                         num1/num2等於2
                                                                         num1%num2等於0
因為是在指定值之後才遞增,所以b的值為0
因為是在遞增之後才指定值,所以c的值為2
 16
        int a = 0:
 17
        int b = 0:
 18
        int c = 0:
                             前置與後置運算子
 19
 20
        b = a + +:
 21
22
23
24
        c = ++a;
                                                                                               輸出結果
        System.out.println("因為是在指定值之後才遞增,所以b的值為"+b);
        System.out.println("因為是在遞增之後才指定值,所以c的值為"+c);
 25
 26
```



■指定運算子

指定運算子主要是用來將資料指定給變數。總之,指定運算子的功用是將 右邊的值(不管文字或數字),指定給左邊的變數,指定運算子有下列幾種

指定運算子	說明
+=	加法後指定數值給變數
-=	減法後指定數值給變數
*=	乘法後指定數值給變數
/=	除法後指定數值給變數
%=	求餘後指定數值給變數
& =	邏輯AND後指定數值給變數
^=	邏輯XOR後指定數值給變數
=	邏輯OR後指定數值給變數
<<=	位元左移後指定數值給變數
>>=	位元右移後指定數值給變數
>>>=	位元右移並補零再指定數值給變數



☑ 位移運算子

所謂的「**位移運算**」,是將10進制的數值轉換成2進制數值之後,根據指定的位移數目,在2進制數值的位數中向左或向右移動(位移),以求取新值。

5<<2 的運算過程如下:

5 0000000000000101

<< 2

20 000000000010100

原本的數字向左移動2個格,右邊適 時補上2個0

5 0000000000000101

>> 2

1 00000000000000000<u>1</u>

原本的數字向右移動2個格,左邊 適時補上2個0。



■ 運算子的執行優先順序

當在Java程式中同時出現了二種不同的運算子,您必須區分式子當中運算子的執行優先順序,才能產生正確的執行結果。

✓當運算子的優先順序相同時

• 同一行Java程式碼當中,如果所有的運算子優先順序相同時,應該遵循「由左至右」的原則。

✓改變運算子的執行順序

程式碼當中的「括號」,其實就是改變運算子優 先順序的方法,用括號先框起來才會優先計算。

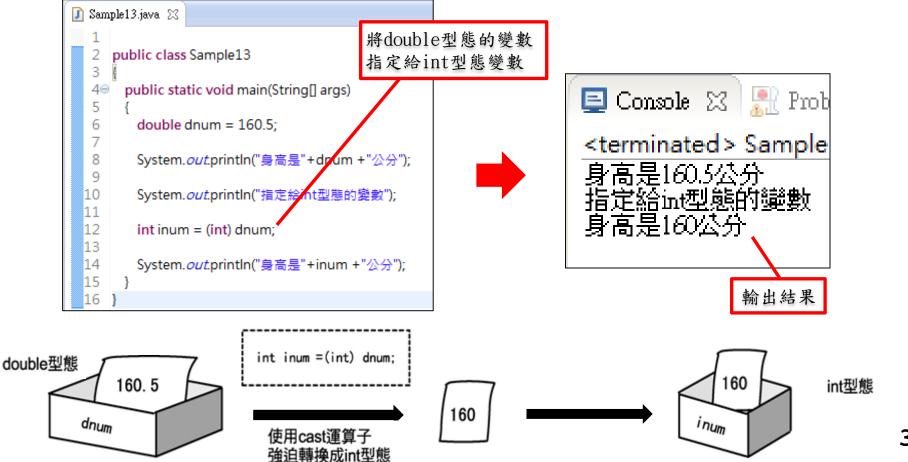
EX: 3+6*2=15;

加括號後(3+6)*2=18



■ 資料型態的轉換(1/2)

要將資料型態較大的數值指定給資料型態較小的變數,須「清楚的明示即 將轉換的資料型態」,必須透過所謂的「cast運算子」來完成。使用的方式如 下所示:





■ 資料型態的轉換(2/2)

```
🚺 Sample14.java 🖂
                            int 型態的變數 d 會轉成
    public class Sample14
                            double 型態來進行運算
  3
                                                          📃 Console 💢 🤙 Problems
      public static void main(String[] args)
  6
       int d = 2:
                                                          <terminated > Sample14 [Ja
       double pi = 3.14;
                                                           直徑是2公分的圓
                                                           其圓周為6.28公分
  9
       System.out.println("直徑是"+d+"公分的圓");
 10
       System.out.println("其圓周為"+(d*pi)+"公分");
                                                           5/4等於1.0
 11
                                                           54等於1.25
 12
       int num1 = 5:
 13
       int num2 = 4;
 14
                                                                          輸出結果
 15
       double div1 = num1/num2:
 16
       double div2 = (double)num1/(double)num2;
 17
 18
       System.out_println("5/4等於"+div1);
                                                                                           int型態
                                                         int型態
 19
       System.outprintln("5/4等於"+div2);
 20
                                                                     使用cast運算子,強制
 21
                                                                      轉換成double型態
                                                                                          double型態
                                                      double型態
                                                                                   4.0
                                                                 5.0
   但在兩個運算元都是int 型態時,
   其運算結果就會是int型態
                                                                         1.25
                                                                                 double型態
```

MMS Lab

Homework

- 繳交期限:10/15 (四)晚上11:59前
- · 繳交格式: 101360352_黃彥岳.zip
- · 繳交內容:將投影片的P9, P10, P16, P18, P25, P26, P27, P28, P31, P34, P38, P39程式重新寫過
 - Homework\Upload\Upload-HW-CH1
 - 繳交內容: 心得報告和程式專案
 - 需要上傳到Github
- 10/16要考試, 在紙本上直接寫程式碼