

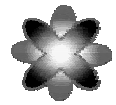
## 第四章 運算式和運算子

### 重點整理

我們都知道電腦可以「處理」很多事，有處理就免不了要進行各種運算。以Java為例，撰寫Java程式的時候，的確牽涉到許多運算的功能，當然Java本身也內建有一些運算子可以運用在這方面。本章主要就是要為您介紹Java當中各種運算子的使用方法。

- ▶ 運算式
- ▶ 運算子
- ▶ 遞增運算子
- ▶ 遞減運算子
- ▶ 指定運算子
- ▶ 位移運算子
- ▶ 運算子的執行優先順序
- ▶ 資料型態的轉換
- ▶ cast運算子





# 進行各種運算 使用變數做為運算



```
class Sample2
```

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

```
{
```

```
    int num1 = 2;
```

```
    int num2 = 3;
```

```
    int sum = num1+num2;
```

多 num1+num2之後，再指定給另一個變數sum

```
    System.out.println("變數num1的值是" + num1 + "。");
```

```
    System.out.println("變數num2的值是" + num2 + "。");
```

```
    System.out.println("num1+num2的值是" + sum + "。");
```

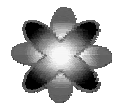
```
    num1 = num1+1;
```

多 num1+1之後，再將結果指定給變數num1

```
    System.out.println("變數num1的值加1後是" + num1 + "。");
```

```
}
```

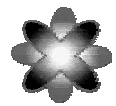
```
}
```



## 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：

|         |            |            |
|---------|------------|------------|
| + (加號)  | (OR)       | < (小於)     |
| - (減法)  | ^ (XOR)    | <= (小於等於)  |
| * (乘法)  | << (左移)    | == (相等)    |
| / (除法)  | >> (右移)    | != (不等於)   |
| % (餘數)  | >>> (右移補零) | ! (邏輯NOT)  |
| + (正號)  | ++ (遞增)    | && (邏輯AND) |
| - (負號)  | -- (遞減)    | (邏輯OR)     |
| ~ (補數)  | > (大於)     | ?: (條件式)   |
| & (AND) | >= (大於等於)  |            |



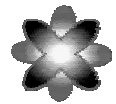
# 遞增運算子與遞減運算子

## ■ 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

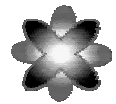
2個減號連續在一起形成的「--」則形成所謂的遞減運算子（decrement operator），遞減運算子會讓變數的值減1。



# 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：

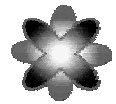
|                     |                           |
|---------------------|---------------------------|
| $+=$ 加法後指定數值給變數     | $\wedge=$ 邏輯XOR後指定數值給變數   |
| $-=$ 減法後指定數值給變數     | $ =$ 邏輯OR後指定數值給變數         |
| $*=$ 乘法後指定數值給變數     | $\ll=$ 位元左移後指定數值給變數       |
| $/=$ 除法後指定數值給變數     | $\gg=$ 位元右移後指定數值給變數       |
| $\%=$ 求餘後指定數值給變數    | $\gg\gg=$ 位元右移並補零再指定數值給變數 |
| $\&=$ 邏輯AND後指定數值給變數 |                           |



# 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。

- 位元左移運算子
- 位元右移運算子



# 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。

---



# ❁ 資料型態的轉換

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：

語法

(資料型態) 運算式

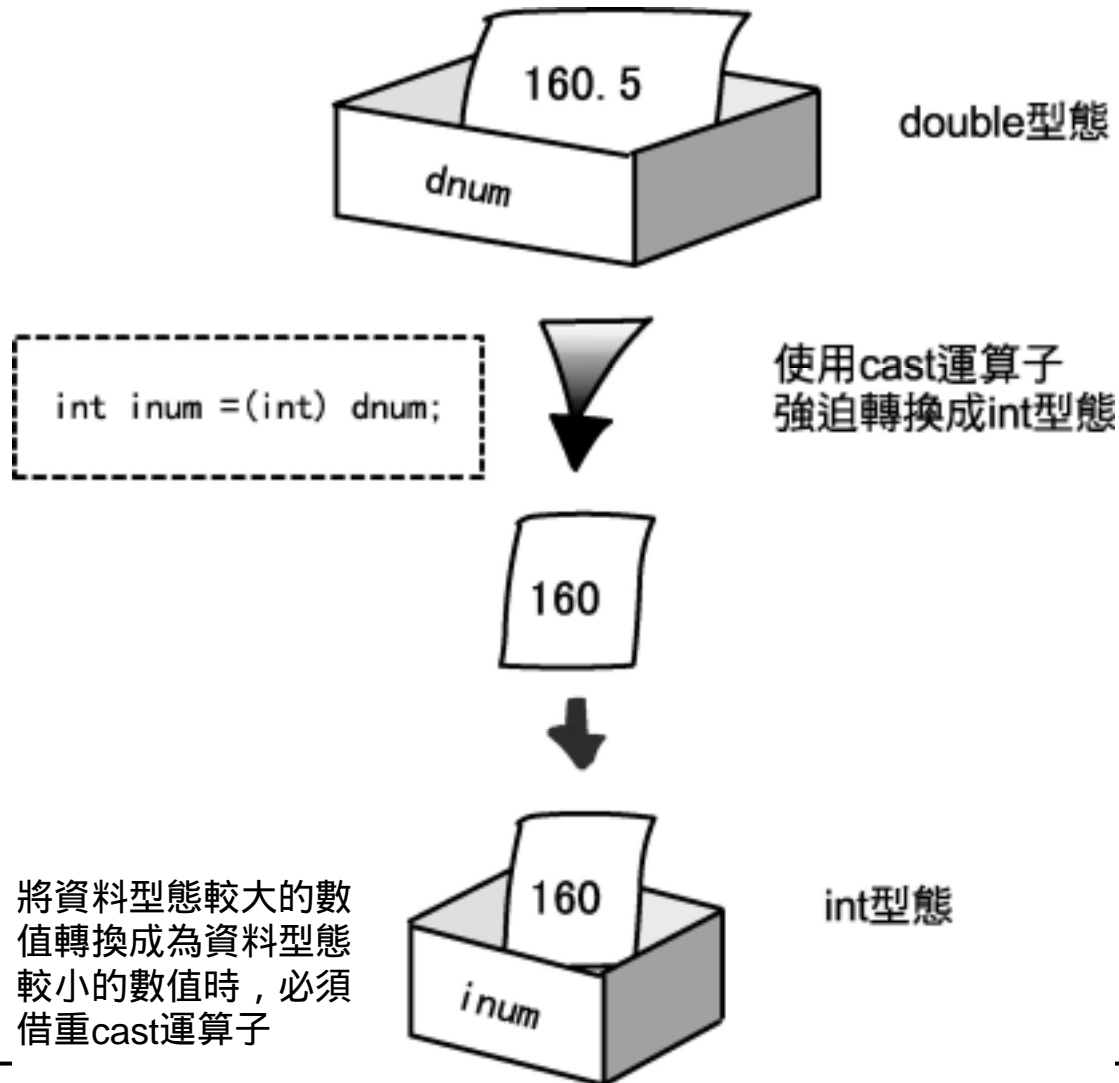
範例

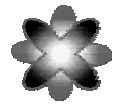
...

```
int inum = (int) dnum;
```

...

括號中指定欲轉換的資料型態





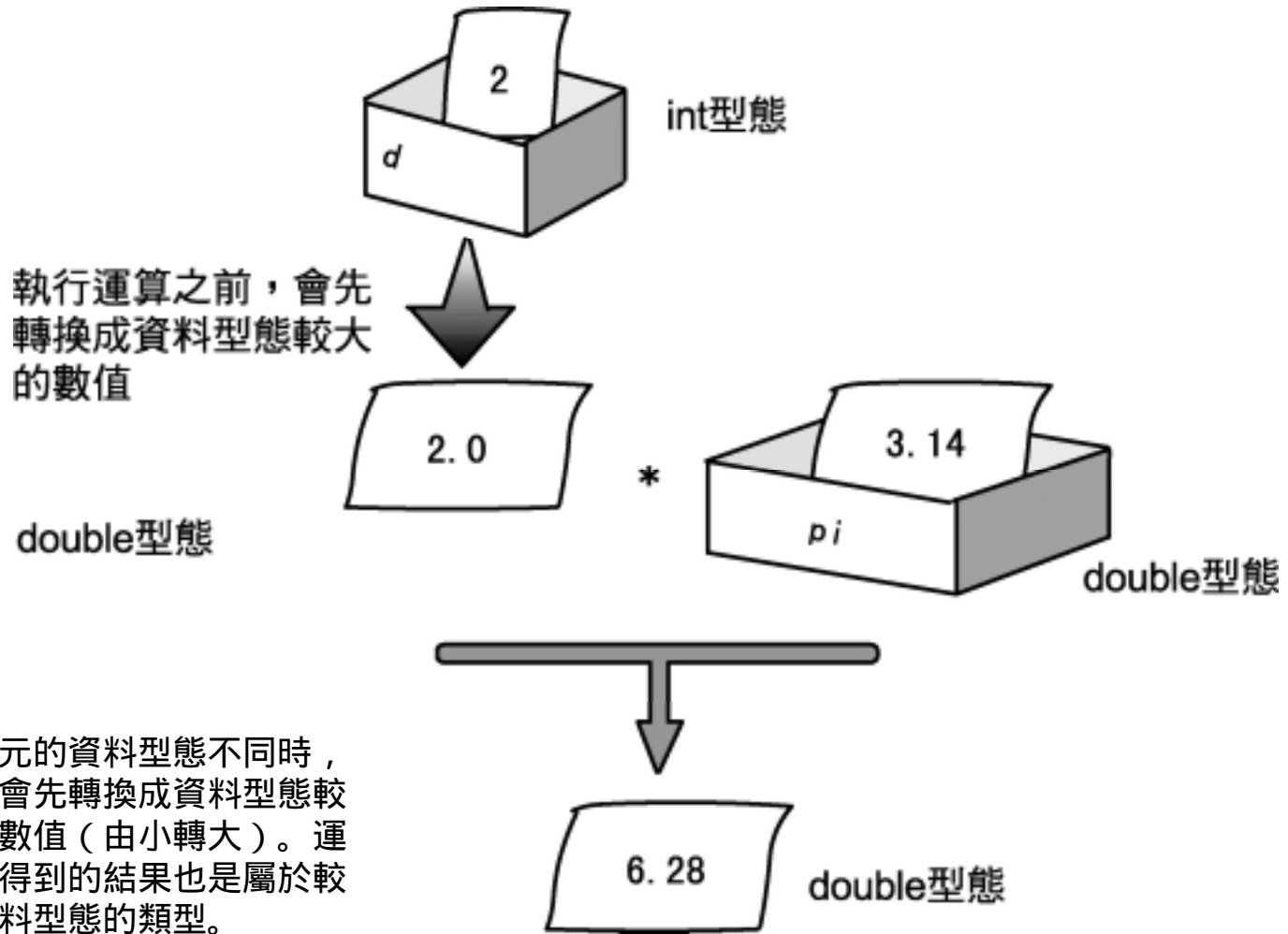
# 不同型態資料的運算

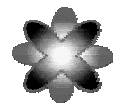


```
class Sample10
{
    public static void main(String args[])
    {
        int d = 2;
        double pi = 3.14;

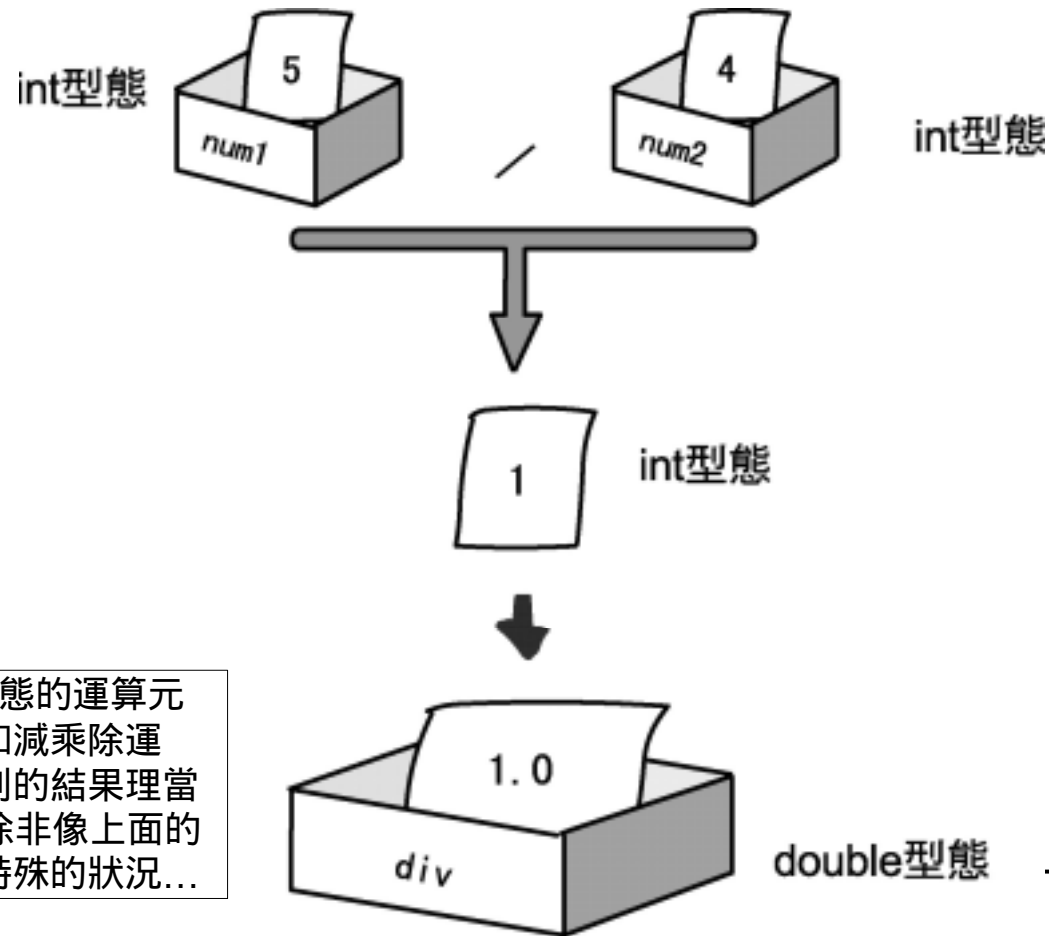
        System.out.println("直徑是" + d + "的圓, ");
        System.out.println("其圓周為" + (d*pi) + "。");
    }
}
```

int型態的變數d會轉換成  
double型態後再進行運算





# 同類型的資料進行運算



2個同屬於int型態的運算元（數值）進行加減乘除運算，理論上得到的結果理當也是int型態。除非像上面的例子一樣，有特殊的狀況...