CHAPTER

5

方法 (Method)

- ✧ 方法
- ✧ 傳值呼叫與參考呼叫
- ✧ 方法間傳遞陣列引數
- ✧ 方法多載
- ✧ 遞迴
- ✧ 認證實例演練

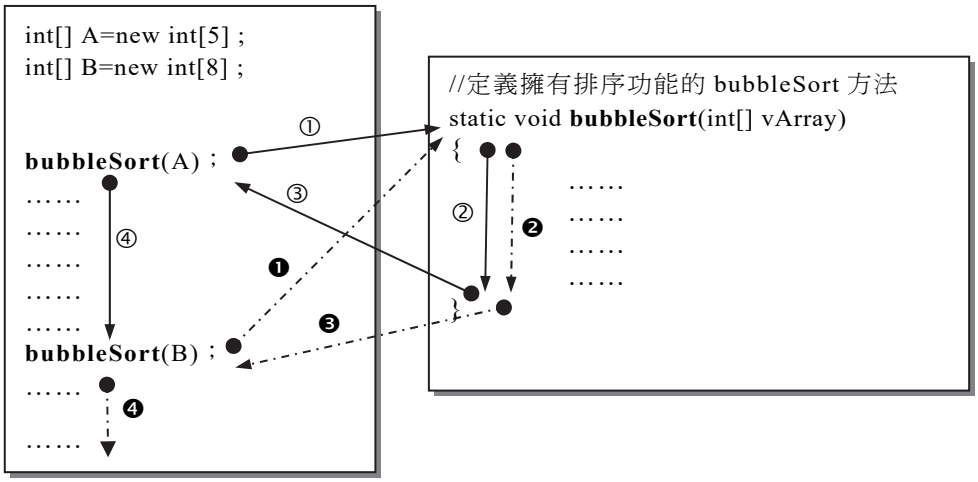
5.1 前言

在撰寫程式的過程中，常會碰到某個敘述區段需要在程式不同地方重複出現。如果每次都要再重寫該敘述區段，撰寫出來的程式將會相當冗長，而且會增加除錯和維護上的困難。所以，程式語言大都會提供一些機制，例如 Java 程式語言可以透過呼叫「方法」來解決這個問題。在 Java 和 C# 程式語言中的「方法」(Method)，在 VB 和 C 程式語言則稱為「函式」或「函數」(Function)，Java 中使用的方法和 C 語言的函式都可重複呼叫，兩者不同處在於方法還可代表屬於該類別的特有行為，而函式則沒此特性。

5.2 方法

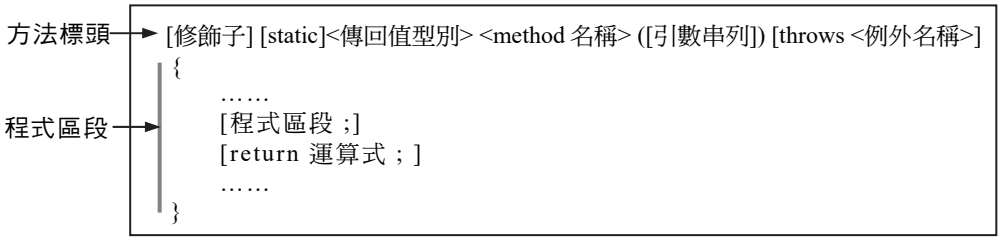
方法(Method)是類別中相當重要的成員之一，到目前為止我們所介紹的都是簡短的程式，只使用到 main() 方法。撰寫較大的程式時，若只用 main()方法來撰寫，不但使得程式不具結構化外，程式除錯的困難度亦會增加。您可以想像要在一個含有數百行，甚至上千行程式碼的 main()方法內，要從中找出錯誤的程式碼，實在是件不容易的事。若程式發生邏輯上的錯誤時，就更難除錯。Java 語言的解決方式，就是將程式中重複的敘述片段，或具有小功能的程式片段獨立出來成為一個方法，自行定義成為「方法」，並給予方法特定的名稱，以方便在程式中重複呼叫。程式中使用方法的好處是，方法在程式中只要編寫一次，卻可在程式中重複呼叫多次，使得程式具結構化外，而且精簡程式的長度使得程式容易維護和除錯。另外，其他的程式有需要用到該方法時，也可套用到該程式上。

假設在下圖中，程式中有兩個整數陣列 A 和 B，和定義一個擁有排序功能的方法(method)，其名稱為「bubbleSort」。當程式執行到 bubbleSort(A); 敘述時(步驟①)，會將 A 陣列傳給 static void bubbleSort( ) {...} 給這個方法執行(步驟②)。當執行完這個方法後，會返回主程式(步驟③)繼續執行接在 bubbleSort(A); 的下一行敘述(步驟④)。接著執行到 bubbleSort(B); 敘述時(步驟❶)，會將 B 陣列傳給 static void bubbleSort() {…}給這個方法執行(步驟❷)。當執行完這個方法時，即會返回主程式(步驟❸)繼續執行接在 bubbleSort(B); 的下一行敘述(步驟❹)。bubbleSort()方法，在程式執行過程中被重複執行了兩次。



5.2.1 如何定義方法

Java 是屬於物件導向的程式語言，不允許方法以單獨的型式存在，必須包含在某個物件之下，成為物件組成的一部分。方法就是這個物件專屬的函式，代表該物件的特殊行為。方法在定義時是由方法標頭(method-head)及程式區段兩部份所構成，其語法如下：



1. **修飾子**：在物件導向程式設計中封裝是物件重要的特性，透過這種機制可以讓物件的資訊適當地隱藏。修飾子(或稱為存取修飾元)的功能就是用來控制該方法，可以指定程式中其他敘述來存取此方法的權限。Java 提供 public、protected、private 與 default 四種存取權限等級，不加存取修飾元時預設為 default。本書第七章中會詳細說明修飾子的存取權限。
2. **Static**：若使用 static 來宣告方法，表示該方法宣告為靜態方法(屬於靜態成員或稱類別成員)。呼叫靜態方法時，不須再使用 new 來建立該類別的物件實體就可以直接使用，例如 main()方法就是屬於靜態方法。

3. **傳回值型別**：方法的傳回值可以是數值、日期、字元、字串、物件…等資料型別。若方法不傳回任何資料，可以將傳回值型別設為 void，例如 main() 方法的傳回值型別即是設為 void。

4. **method 名稱**：方法名稱的命名除了要符合識別字規定外，在 Java 中通常以小寫字母命名， 例如 add；如果是兩個單字以上組合，則第二個單字起的字首要大寫，例如 bubbleSort。

5. **引數串列**：引數串列的個數可以為零、一個或一個以上，每個引數之間必須以「,」逗號來分隔，若省略引數串列表示呼叫此方法不傳入任何值。引數可以是變數、常數、陣列、物件，但不可以是運算式。若呼叫方法時所傳遞的引數是基本資料型別(如：char、int、byte…等)，則引數的傳遞方式是屬於傳值呼叫；若所傳遞的引數是參考資料型別(如：物件型別的資料、但字串例外)，則引數的傳遞方式是屬於參考呼叫。

6. **throws <例外名稱>**：關於 throws 敘述是用來宣告某個方法可能會拋出那些例外，出現的情形依程式應用而定，並非所有的方法都會用到。例如使用 Scanner 類別的 nextInt()方法時，當使用者輸入不合法的字元時，會拋出屬於 Exception 的例外，因此為了讓程式能正常執行，可以在 main 方法後加上「throws Exception」。有關例外處理的部份會在第八章中詳加說明。

7. **程式區段**：方法的主體即是呼叫該方法時所要執行的程式區段，其範圍是由左、右大括號所框住。

8. **return 運算式**：return 敘述是有傳回值時才會使用，「return 運算式;」中的運算式即是執行方法後要傳回的值，該運算式的資料型別應和方法的「傳回值型別」一致。若方法宣告為 void 型態，沒有傳回值，return 敘述就不可以出現。

下例的 Math 類別中建立了 mul 和 div 方法，這兩個方法皆設為 void 表示沒有傳回值，而且沒有使用存取修飾子，因此是屬於 default 預設存取權限等級。

檔名： \ex05\src\ex05\Math.java

```
01 package ex05;
02 class Math {
03   static void mul(int x, int y) {
04       System.out.print(x + " * " + y + " = " + (x * y));
```

```
05  }
06  void div(int x, int y) {
07      System.out.print(x + " / " + y + " = " + (x / y));
08  }
09  }
```

1. 第 3~5 行：用 static void mul(int x, int y)宣告 mul 為靜態方法，其名稱為 mul，void 表示無回傳值，呼叫 mul 方法時，必須給予兩個 int 型別的引數。因為 mul()以 static 宣告為靜態方法，因此在呼叫這個方法時不需要建立 Math 類別的物件實體，即可直接呼叫使用。
2. 第 6~8 行：用 void div(int x, int y)宣告 div 方法，其名稱為 div，void 表示無回傳值，呼叫 div 方法時，必須給予兩個 int 型別引數。因為 div()方法是屬於 math 類別中的成員，因此在呼叫這個方法時必須先使用 new 建立 Math 類別的物件實體，才可以使用 div()方法。

5.2.2 如何呼叫方法

在上一節學會了如何定義方法後，接著本節將介紹如何呼叫方法。

一. 呼叫靜態方法

若類別中的方法使用 static 宣告成靜態方法，可以不用使用 new 建立該類別的物件實體，就可直接呼叫使用。以下兩種寫法都是呼叫方法的敘述，其語法如下：

```
語法 1：[類別名稱.]方法名稱([引數串列])
語法 2：變數 = [類別名稱.]方法名稱([引數串列])
```

1. 若呼叫的靜態方法沒傳回值時，使用語法 1；若呼叫的靜態方法有傳回值則使用語法 2，會將方法的結果指定給等號左邊的變數。
2. 若要呼叫的靜態方法定義在不同類別中，則呼叫該靜態方法時，前面必須加上「類別名稱.」；若呼叫的靜態方法定義在同一類別中，則可以省略「類別名稱.」。

3. 緊接在「呼叫敘述」後面的引數串列稱為「實引數」，緊接在「被呼叫方法」後面的引數串列稱為「虛引數」。呼叫敘述之實引數可以是常數、變數、運算式、陣列、物件；被呼叫方法的虛引數可以是變數、陣列、物件，但不可以是常數或運算式。
4. 呼叫與被呼叫的方法名稱必須相同，而且兩者對應引數的數目和資料型別也必須相同，但是兩者的引數名稱可以不相同。

實作 FileName：\ex05\src\ex05\Static1.java

Static1 類別包含 main()靜態方法和 sub()靜態方法，Static2 類別中只包含 sub()的靜態方法。main()靜態方法內的呼叫敘述 sub(a - 5, 3)呼叫 Static1 類別內的 sub()方法；呼叫敘述 sub(a + 3, 5)呼叫 Static2 類別內的 sub()方法，兩者 sub()方法都將傳入的兩個引數相減，並顯示其差。

程式碼

檔名： \ex05\src\ex05\Static1.java

```
01 package ex05;
02 public class Static1 {
03    static void sub(int x, int y) { // 被呼叫方法主體
04        System.out.print("呼叫 Static1 類別的 sub 方法-->");
05        System.out.println(x + " - " + y + " = " + (x - y));
06    }
07    public static void main(String[] args) {
08        int a = 25;
09        // 呼叫同一類別的 sub 方法
10        sub(a - 5, 3); // 呼叫敘述
11        // 呼叫不同類別的 sub 方法
12        Static2.sub(a + 3, 5); // 呼叫敘述
13    }
14}
15
16 public class Static2 {
17    static void sub(int x, int y) { // 被呼叫方法主體
18        System.out.print("呼叫 Static2 類別的 sub 方法-->");
19        System.out.println(x + " - " + y + " = " + (x - y));
20    }
21 }
```

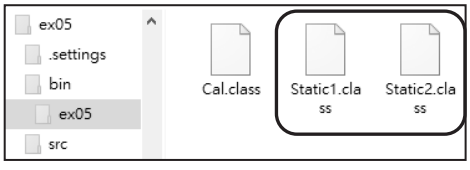
結 果

```
呼叫 Static1 類別的 sub 方法-->20 - 3 = 17
呼叫 Static2 類別的 sub 方法-->28 - 5 = 23
```

說明

1. 第 10 行：執行 sub(a-5, 3) 敘述時，會呼叫同一類別(Static1) 內第 3~6 行 sub()靜態方法。將第一個實引數(a-5)的結果(20)傳遞給第 3 行程式的虛引數 x；第二個實引數 3 傳遞給第二個虛引數 y。接著執行第 4,5 行{…}程式區塊內的敘述，進行兩數相減並顯示結果。執行完畢後，跳回 sub(a-5, 3); 的下一行敘述繼續往下執行。
2. 第 12 行：執行 Static2.sub(a+3, 5) 敘述時，會呼叫不同類別 Static2 當中的 sub() 靜態方法(第 17~20 行)。此方法的功能和 Static1 類別中的 sub()方法相同，當這個方法執行完成即會跳回 Static2.sub(a+3, 5); 的下一行敘述繼續往下執行。
3. 由上可知，要呼叫同一類別內的靜態方法，只要直接撰寫該方法名稱與引數串列即可，第 10 行敘述即呼叫屬於 Static1 類別的 sub()靜態方法。若呼叫不同類別中的靜態方法，則必須撰寫「類別名稱.方法名稱」，第 12 行敘述即是呼叫不屬於 Static1 類別，而是屬於 Static2 不同類別的 sub 靜態方法。

NOTE 在 Java 中的每一個類別都會使用一個 Class 檔，本例 Static1.java 中建立 Static1 與 Static2 兩個類別，編譯後會產生 Static1 和 Static2 兩個 .class 檔。因此執行時，Static1.class和Static2.class檔必須在相同路徑下才能正常執行。



二. 呼叫物件實體的方法

如果要呼叫類別中不是使用 static 宣告的靜態方法，就必須先使用 new 建立該類別的實體－「物件」，此時該物件會擁有該類別所有的方法成員與資料成員，接著就可以使用「物件.方法名稱(引數串列)」方式呼叫該方法。類別是建立物件的藍圖，所以每個使用類別所建立物件都會擁有該類別的方法成員或資料成員，關於類別與物件以及物件導向技術，請參閱第 6 章。

說明

1. 第 12~17 行：建立名稱為 factorial()的靜態方法，此方法擁有計算引數階乘的功能，並將計算後的結果以 return 傳回(第 16 行)，其傳回的資料型別為整數型別資料。
2. 第 6、8 行：分別以引數 6 和 9 呼叫 factorial()方法，並將結果傳回給指定的 fac1 與 fac2 整數變數。
3. 第 7、9 行：分別顯示 fac1 與 fac2 等 factorial()方法計算後的傳回值。

5.3 傳值呼叫與參考呼叫

Java 中有關方法的引數傳遞方式有：傳值呼叫(Call By Value)和參考呼叫(Call By Reference)兩種主要機制，本節將介紹這兩種引數傳遞機制的使用方式。

5.3.1 傳值呼叫

方法中的虛引數如果宣告為基本資料型別，如 char、byte、short、int、long、float、double、boolean 八種型別變數，就表示該方法的引數傳遞方式是採傳值呼叫(Call By Value)。基本資料型別的變數是存放在 Stack 堆疊儲存空間，方法若採傳值呼叫，則呼叫敘述的實引數與被呼叫方法的虛引數是分別佔用不同記憶體，因此，使用傳值呼可以防止變數被方法更改。

實作 FileName：\ex05\src\ex05\CallByVal.java

本例用來驗證傳值呼叫，請觀察傳值呼叫時實引數與虛引數兩者呼叫前後的變化情形。

結 果

```
傳值呼叫前      a = 10   b = 15
傳值呼叫中      x = 15   y = 10
傳值呼叫後      a = 10   b = 15
```

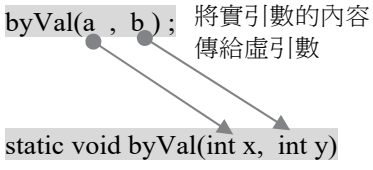
程式碼

檔名： \ex05\src\ex05\CallByVal.java

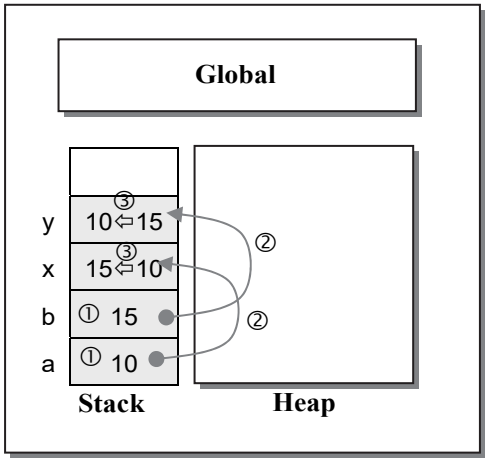
```
01 package ex05;
02 public class CallByVal {
03    public static void main(String[] args) {
04        int a = 10, b = 15;
05        System.out.println(" 傳值呼叫前\t a = " + a + "\t b = " + b );
06        byVal(a, b);
07        System.out.println(" 傳值呼叫後\t a = " + a + "\t b = " + b );
08    }
09    static void byVal(int x, int y) {
10        int t; //以變數 t 作為暫存區，將引數互換
11        t = x;
12        x = y;
13        y = t;
14        System.out.println(" 傳值呼叫中\t x = " + x + "\t y = " + y );
15    }
16 }
```

說明

1. 第 5 行：顯示傳值呼叫前的變數值：a = 10、b = 15。
2. 第 6 行：呼叫 byVal()方法時，將實引數 a 及 b 以傳值方式傳遞給 x 和 y，並執行第 9~15 行程式。



a 的值為 10 傳給 x
b 的值為 15 傳給 y
執行 x = y; y = x; 之後
則 x = 15、y = 10
而 a、b 則仍會保持原值



5.4 方法間傳遞陣列引數

5.4.1 以陣列當引數

若要將整個陣列當做引數傳遞給方法時，只要在被呼叫方法的虛引數或資料型別後加上 [ ] 一對括號，即表示傳來的引數必須是陣列型別。以陣列當引數時是屬於參考呼叫，所以方法對陣列的執行結果會影響原呼叫敘述的陣列值。呼叫方法時實引數只須寫陣列名稱，但後面不可以加上 [ ] 括號。寫法如下：

```
int[] myArray = new int[] {10, 20, 56, 70, 30};
callArray(myArray) ;          // 呼叫方法時，實引數所傳遞的陣列名稱不需加上[]
// 方法內的資料型別後加上[ ]，表示所傳遞引數資料為陣列型別
static void callArray(int[] vArray){
    ……
}
// 上述寫法也可改成在方法內的虛引數後加上[ ]
static void callArray(int vArray[]){
    ……
}
```

實作 FileName：\ex05\src\ex05\Array1.java

試撰寫一個 encode()方法，可以將字元陣列元素值加 1，來達成資料擾碼的方法。和 decode()方法，可以將字元陣列元素值減 1，來達成資料還原的方法。程式先顯示原始字元陣列值，接著顯示擾碼後出陣列元素值，最後再顯示還原後的陣列元素值。

結 果

```
原始字串 -> Java
擾碼後字串 -> Kbwb
還原後字串 -> Java
```

程式碼

檔名： \ex05\src\ex05\Array1.java

```
01 package ex05;
02 public class Array1 {
03     public static void main(String[] args) {
04         char[] str = {'J', 'a', 'v', 'a'};
05         System.out.print("原始字串 -> ");
06         System.out.println(str);
07         encode(str);
08         System.out.print("擾碼後字串 -> ");
09         System.out.println(str);
10         decode(str);
11         System.out.print("還原後字串 -> ");
12         System.out.println(str);
13     }
14
15     static void encode(char[] s) {
16         for(int i = 0; i < s.length; i ++)
17             s[i] += 1;
18     }
19
20     static void decode(char[] s) {
21         for(int i = 0; i < s.length; i ++)
22             s[i] -= 1;;
23     }
24 }
```

說明

1. 第 7 行：呼叫 encode()靜態方法，並將 str 陣列傳入，encode()方法會將陣列內的每個元素值都 +1，以達成擾碼的效果。
2. 第 10 行：呼叫 decode()靜態方法，並將擾碼後 str 陣列傳入，decode()方法會將陣列內的每個元素值都 -1，來還原成原來的字串。
3. 因為以陣列當引數時是屬於參考呼叫，所以經 encode()方法及 decode()方法運算後的陣列元素值會影響原 str 陣列值。將 str 陣列的元素值顯示時，就可以清楚觀察到其中的變化。

5.4.2 取得命令列的資料

main 方法中的虛引數 args 屬於字串陣列型別變數，args 可用來取得由命令列輸入的資料，我們在第一章曾介紹過。若連續輸入資料，資料間必須使用一個空白來加以區隔，由命令列輸入的第一個資料會放在 args[0]，第二個輸入的資料會放在 args[1]，…以此類推。

實作 FileName：\ex05\src\ex05\Score.java

在命令列輸入「java Score」後，再輸入「66 77 88」三個學生成績，注意成績之間要用一個空白字元分開。程式執行時，會顯示所輸入這三位學生的成績。

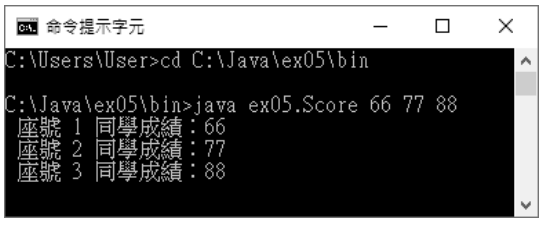
程式碼

檔名： \ex05\src\ex05\Score.java

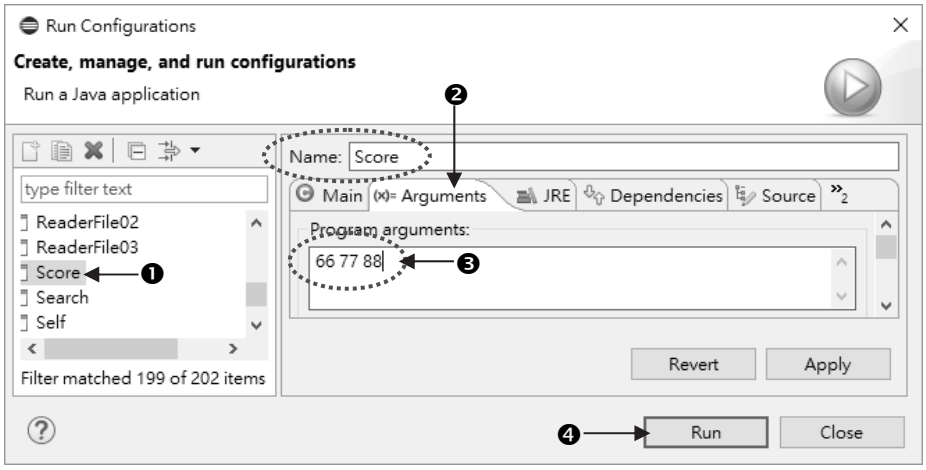
```
01 package ex05;
02 public class Score {
03    public static void main(String[] args) {
04       for (int i = 0; i < args.length; i++)
05           System.out.println(" 座號 " + (i + 1) + " 同學成績：" + args[i]);
06    }
07}
```

說明

1. 在「命令提示字元」下，先執行「cd C:\Java\ex05\bin」指令，切換路徑到 Score.class 所在位置。然後再執行「java ex05.Score 66 77 88」指令(ex05.表 package)，來執行 score 程式並指定資料。輸入的資料會放到 main 方法的 args 字串陣列引數中，所以 args[0] = 66, args[1] = 77, args[2] = 88。



2. 在 Eclipse 整合環境中可不用到命令列中測試，可執行【Run/Run Configurations…】指令，在「(x)=Arguments」標籤頁中輸入引數即可。



結果

```
座號 1 同學成績：66
座號 2 同學成績：77
座號 3 同學成績：88
```

3. 第 4~5 行：使用 for 迴圈逐一由 args[]取得命令列輸入的資料，然後顯示出來。

5.5 方法多載

所謂的「方法多載」(method overloading)或稱為超載、覆載，就是在同一個類別中，允許方法使用相同的名稱，但是後面所接的引數串列必須資料型別不同、個數或順序不同。

```
01  void methed(){}
02  int methed(){}          //不能多載：雖然傳回值不同但引數個數相同
03  void methed(int a){}        //成功多載：比第 1 行的方法多一個引數
04  void methed(int b){}        //不能多載：引數名稱不同，但和第 3 行的型別和個數相同
05  void methed(String s){}    //成功多載：雖然和第 3 行引數個數相同但型別不同
06  void methed(int a ,String s){} //成功多載：引數個數和型別都不同
07  void methed(String s,int a){} //成功多載：和第 6 行引數個數和型別相同但順序不同
```

5.8 認證實例演練

題 目

一. 請問第 5 行執行後顯示結果為何 ？

①5 ②10 ③12 ④17 ⑤24

```
01  public class Test{
02      int x = 12;
03      public void method(int x) {
04          x += x;
05          System.out.println(x);
06      }
07  }
…
30  Test t = new Test();
31  t.method(5);
```

說明

1. 第 31 行：執行 t.method(5);敘述時，會呼叫第 3~6 行的 method 方法，並將引數 5 傳給 x。要特別注意的是 method 方法內的 x 是該方法的區域變數，所以第 2 行的 x(=12)，不會影響到該區域變數。
2. 第 3 行：因為 x = 5 所以 x += x;敘述執行後 x = 10，因此答案為②。

題 目

二. Java 程式碼如下：

```
01  class Test{
02      public String doit(int x, int y) {
03          return "a";
04      }
05
06      public String doit(int …  nums) {
07          return "b";
08      }
09  }
…
30  Test t = new Test();
31  System.out.println(t.doit(2, 3));
```

請問執行結果為何？

① 第 31 行輸出 "a"　　② 第 31 行輸出 "b"
③ 執行時拋出一個例外　　④ 第 6 行編譯錯誤

1. 第 2~4 行和第 6~8 行的兩個 doit()方法是方法多載，兩個方法都合法。
2. 呼叫 t.doit(2,3)多載方法時，以引數完全相等的方法優先，所以會執行第 2~4 行的 doit()方法，會印出 "a" 字元，所以答案是①。
3. 第 6~8 行的 doit()是省略引數的方法，雖然也可能被執行，但仍以不省略且引數完全相等的方法優先。

5.9 習題

一. 選擇題

1. 若要定義靜態方法，必須使用下列何者來宣告？
   ①class　②public　③static　④throws。

2. 下列何者不是使用方法的優點？
   ①使程式具結構化　②使程式容易除錯
   ③可縮短程式碼　④加快程式執行速度。

3. 使用相同名稱來定義多個方法，就稱為：
   ①多載　②覆寫　③遞迴　④靜態方法。

4. 當一個方法再呼叫方法本身，就稱為：
   ①多載　②覆寫　③遞迴　④靜態方法。

5. 如果定義方法時沒有使用修飾子，代表該方法的存取權限等級為何？
   ①default　②private　③protected　④public。

6. 程式執行時被呼叫方法中的引數稱為？
   ①虛引數　②實引數　③參考引數　④靜態引數。

7. 下列敘述何者錯誤？

①使用省略號的方法可以不傳入引數

②靜態方法可以不用建立物件就可以直接使用

③傳值呼叫的實引數和虛引數佔相同記憶體

④方法中的虛引數若為物件表示引數傳遞方式為參考呼叫。

8. 下列敘述何者錯誤？

①方法定義為 int 表示傳回值為整數

②方法以 void 宣告時可以用 return 來傳回值

③方法定義為 void 表示沒有傳回值

④使用 return 來傳回方法的傳回值

9. Java 程式碼如下：

```
01  public class Pass{
02     static public void main(String [] args) {
03          int x = 5;
04          Pass p = new Pass();
05          p.doStuff(x);
06          System.out.print(" main x =  " + x );
07     }
08
09      void doStuff(int x) {
10          System.out.print(" doStuff x =  " + x++);
11      }
12  }
```

請問執行結果為何？

①編譯錯誤　②執行時拋出一個錯誤　③doStuff x =6 main x =6

④doStuff x = 5 main x = 5　⑤doStuff x = 5 main x = 6　⑥doStuff x = 6 main x = 5

10. Java 程式碼如下：

```
01 public class Test{
02    static public void main(String [] args) {
03       go("Hi",1);
04       go("Hi", "World", 2);
05    }
06    static void go(String … y, int x) {
07       System.out.print(y[y.length-1] + " " );
08    }
09 }
```

請問執行結果為何？

①Hi Hi　②Hi World　③World World　④編譯錯誤　⑤執行時拋出例外

二. 程式設計

1. 試設計如下圖四星彩開獎系統，程式執行時會顯示 0~9 重複亂數號碼。

```
本期四星彩開獎號碼如下：
 9 7 1 8
```

2. 試使用靜態遞迴方法算出費氏數列。費氏數列值第 1 及第 2 項皆為 1，第 3 項之後的公式為 $f_n = f_{n-1} + f_{n-2}$。使用者輸入一個正整數 X，計算出費氏數列的第 X 項並輸出。

```
計算費氏數列的第 X 項，請輸入 X= 12
費氏數列的第 12 項= 144
```

3. 使用非靜態遞迴方法算出最大公因數，執行時會顯示最大公因數。

```
計算最大公因數，請輸入第一個數 = 21
計算最大公因數，請輸入第二個數 = 90
21、90 的最大公因數 = 3
```

4. 由命令列輸入多位學生的分數，每個分數要使用一個空白來區隔，當執行程式後，會顯示每一位學生的分數，以及分數的加總。

```
第 1 位同學的分數： 50
第 2 位同學的分數： 88
第 3 位同學的分數： 100
第 4 位同學的分數： 99
第 5 位同學的分數： 62
總分： 399
```