

Module VI

方法的宣告與使用

1. 方法的建立與呼叫
2. 傳遞參數與回傳值
3. 方法覆載 (Overloading)

方法建立與呼叫 (1/2)

- 定義：
 - 可重覆使用的程式碼片段
- 範例：
 - 如 `public static void main(String[] args) {...}`
main方法為程式的進入點
- 宣告：
 - `<modifier>* <return_type> <name> (<parameter>*) {...}`
修飾存取方式 回傳值型別 方法名稱 (傳入參數) {...}

方法建立與呼叫 (2/2)

- 方法的呼叫

- 呼叫不同類別的方法：

- 如同物件參考變數是利用「.」運算子來操作物件的值
 - 此處同樣是利用「.」運算子來呼叫不同類別中的方法
 - 例：`myPen.showInfo()`;

- 呼叫相同類別中的方法：

- 同一類別中，可從calling method直接叫用worker method
 - 例：

```
public void showInfo() {  
    p("Brand is : " + brand);  
    p("Price is : " + price);  
}  
public void p(String s) {  
    System.out.println(s);  
}
```

傳遞參數與回傳值

- worker method

- 可能需要有參數傳進來
- 如有回傳值，則不再是void，
且需加return

```
public class Calculator {  
    public int sum(int x, int y) {  
        return x + y;  
    }  
}
```

- calling method

- 傳遞worker method要的參數型別與值
- 如有回傳值，則可以用適當的型別去接

```
public class Calculator {  
    public static void main(String[] args) {  
        Calculator myCal = new Calculator();  
        int sum = myCal.sum(1, 2);  
        System.out.println(sum); //3  
    }  
}
```

課堂練習

- 請新建一個Student.java，並定義一個屬性為score，型別為int，另定義兩個方法分別為play()與study()，功能如下：
 - public void play(int hours)
每玩樂一個小時，score就會減1
 - public void study(int hours)
每讀書一個小時，score就會加1

方法覆載 (1/4)

- 基本觀念
 - **Overloading**讓我們可以用統一的方法名稱來呼叫相同功能的方法
 - 程式設計者為**功能類似的方法**提供**相同的方法名稱**時，Java會自動依據參數的數目及不同的資料型別，自動呼叫對應的方法，如：
 - `public void println(int i)`
 - `public void println(float f)`
 - `public void println(String s)`
- 注意事項
 - 必須注意的是，Overloading的方法無法根據回傳值型態的不同而區別，如以下為**重覆宣告**的錯誤：
 - `void method(int i)`
 - `int method(int i) //重覆宣告`
 - `String method(int i) //重覆宣告`

方法覆載 (2/4)

```
1 public class TestOverloading{
2
3     //圓形面積
4     public static double areaMeasure(double radius){
5         return radius* radius*3.14;
6     }
7     //長方形面積
8     public static double areaMeasure(double height, double width){
9         return height*width;
10    }
11    //梯形面積
12    public static double areaMeasure(double upper, double bottom, double height){
13        return (upper+bottom)*height/2;
14    }
15
16    public static void main(String args[]) {
17        double i = areaMeasure(3.0);
18        double j = areaMeasure(3.0,4.0);
19        double k = areaMeasure(3.0,4.0,5.0);
20        System.out.println("圓形面積=" +i);
21        System.out.println("長方形面積=" +j);
22        System.out.println("梯形面積=" +k);
23    }
24 }
```

方法覆載 (3/4)

- 方法覆載(Overloading)與Java API
 - 許多在Java API中的方法都是Overloading的
 - 譬如System.out.println(...)，使用上非常方便就是因為Overloading的關係
 - 註：println(...)是PrintStream類別的方法

方法覆載 (4/4)

- 方法覆載(Overloading)與Java API

void	println() Terminates the current line by writing the line separator string.
void	println(boolean x) Prints a boolean and then terminate the line.
void	println(char x) Prints a character and then terminate the line.
void	println(char[] x) Prints an array of characters and then terminate the line.
void	println(double x) Prints a double and then terminate the line.
void	println(float x) Prints a float and then terminate the line.
void	println(int x) Prints an integer and then terminate the line.
void	println(long x) Prints a long and then terminate the line.
void	println(Object x) Prints an Object and then terminate the line.
void	println(String x) Prints a String and then terminate the line.

章節整理

- 方法建立後，可使用「.» 運算子來呼叫不同類別的方法
- 瞭解方法的傳遞參數與回傳值觀念
- 運用方法得宜，可以讓程式設計師減少許多程式碼的撰寫
- 方法覆載(Overloading)藉由傳遞參數的數量與資料型別的不同而對應至合適的方法，這讓我們在撰寫程式碼的命名上顯得輕鬆許多
- 需注意方法重覆宣告的問題