

12 - 2 介面 (Inter face)

- 撰寫物件導向程式的第一步, 就是分析出程式中需要哪些類別, 以及類別之間的繼承關係。不過就像我們在現實世界中所看到的, 許多『不同類』的事物, 其間又通常會具有一些相似的行爲。舉例來說, 飛機和小鳥很顯然不會是同性質的類別, 而其飛的方式也不同, 但不可否認它們都具有會飛行的行爲。
- 一些在繼承架構中明顯不同的類別, 它們卻有具有一些相似的行爲 (特性), 造成設計類別時的困擾

介面 (Interface)

- 例如將明顯不同性質的類別 (例如飛機和鳥), 湊成在同一繼承架構下, 使得類別的繼承關係不合常理
- 爲了不打亂原有的繼承架構, 我們可能要選擇分別在各自的類別中定義各自有的行爲。
- 爲了讓這樣的設計, 在實作它們應有的共通行爲時造成遺漏或命名不一致, 在 **Java** 中特別提供了介面(**Interface**) 來描述這個共通的行爲

12-2-1 定義介面

- 介面代表的是一群共通的行為，感覺上和類別好像有些類似之處，但其實兩者具有相當的差異
- 外觀上定義介面也是用大括號來描述此介面的方法，而開頭要改用 **interface** 來表示要定義的是介面：

```
interface 介面名稱 {  
    // 介面中的方法  
}
```

定義介面

- 類別是用以描述實際存在的物件，而介面則僅是用以描述某種行為方式
- 以下是定義介面時要注意的重點：
 - 介面的命名也和類別一樣，通常都是以首字母大寫的方式，使得其在程式中容易被識別。有些人習慣在介面的名稱前加上一個大寫字母‘I’，以特別標示這個名稱是個介面
 - 在介面中只能定義方法的型別 (傳回值) 及參數型別，不可定義方法本體，這些方法預設都會自動成為 **public** 的抽象方法

定義介面

- 由於介面通常代表某種特性, 因此介面的名稱一般都是一個形容詞, 表示可以如何的意思, 例如可用 **Flying** 表示『會飛』的意思。
- 舉例, 計算地價, 要計算地價時, 要算出土地的面積, 而計算面積這件事, 可能是很多類別需要的功能, 所以我們可以定義一個計算面積的介面:

```
interface Surfacing {  
    double area();           // 計算面積的方法  
}
```

定義介面

- 因為每種不同形體，其面積的計算方式也都不同，我們也無法預知有哪些類別需要計算面積，此處 **surfacing** 介面只規定了計算面積的方式名稱為 **area**、沒有參數、但傳回值為 **double** 型別
- 介面中的方法預設都是公開的抽象方法，所以通常 **public**、**abstract** 也都省略不寫

12-2-2 介面的實作

- 定義好介面之後，需要使用該介面的類別，就必須實作該介面，實作介面時，必須在類別名稱之後，使用**implements**保留字，再加上要實作的介面名稱
- 介面中所定義的方法會自動成為抽象方法，因此實作介面時必須完全實作介面中的所有方法

```
interface Surfacing {  
    double area();          // 計算面積的方法  
}  
  
class Circle implements Surfacing {  
    ...  
    public double area() {  
        // 計算圓面積並回傳  
    }  
}
```

介面的實作

```
1 interface Surfacing{
2     double area();
3 }
4 class Shape{
5     protected double x, y;
6     public Shape(double x, double y){
7         this.x = x;
8         this.y = y;
9     }
10    public String toString(){
11        return "圓形圓心: (" + x + ", " + y + ")";
12    }
13 }
14 class Circle extends Shape implements Surfacing{
15     private double r;
16     final static double PI = 3.14159D;
17     public Circle(double x, double y, double r){
18         super(x, y);
19         this.r = r;
20     }
21    public double area(){
22        return PI*r*r;
23    }
24    public String toString(){
25        return "圓心: (" + x + ", " + y + ")、半徑: " + r + "、面積: " + area();
26    }
27 }
28 public class ShapeArea{
29     public static void main(String [] argv){
30         Circle c = new Circle(5, 8, 7);
31         System.out.println(c.toString());
32     }
33 }
```

圓心: (5.0, 8.0)、半徑: 7.0、面積: 153.93791

介面的實作

- 要記住介面所提供的方法都是 **public**, 因此實作介面時也要將之宣告為 **public**, 不可將之設為 **protected** 或 **private**, 如此也會造成編譯錯誤。