

## 第九章 繼承性與多形性運用

### 9-1 類別的繼承性

#### 9-1-1 何謂繼承性

製作一個類別之後，再由此類別產生另一個新的類別，新類別除了可以繼承原類別的變數及方法成員之外，還可以增加其他成員（變數或方法），這就是類別的『繼承性』。

類別產生物件之後是一個獨立的程式執行實體，這是實現軟體 IC 的基本概念。然而，結合多個類別成為另一功能更豐富軟體 IC，則需利用類別的繼承性來達成。也就是說，吾人可利用類別的繼承性將若干個類別整合成另一個功能較強的類別，再利用他產生的物件，此為物件導向程式設計的基本精神。另一方面，再透過 Java 可攜性的功能，任何系統發展出來的軟體 IC，都能輕易地移植到其他系統上執行。如此一來，將會有更多人願意發展軟體 IC 來銷售，整個軟體工業將因此而改觀。由此可見，類別繼承性在物件導向程式設計佔有舉足輕重的地位，本章將利用範例引導讀者，來認識繼承性的特性與運用。

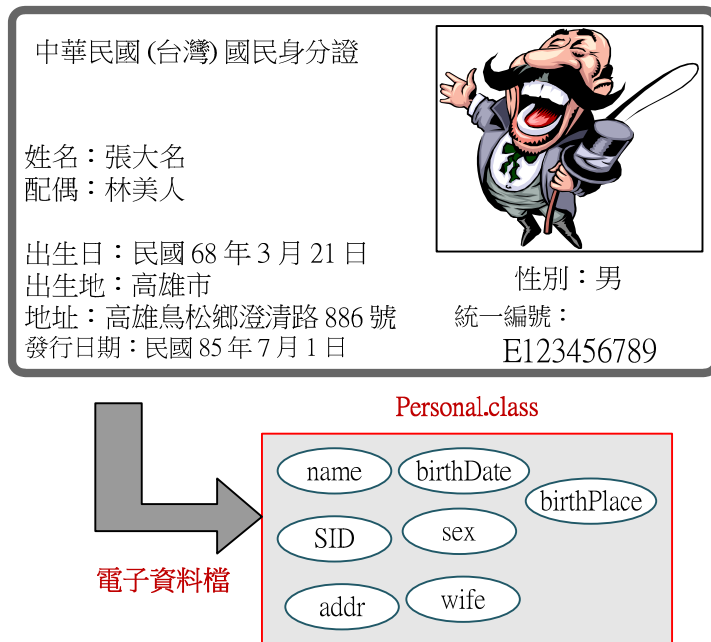
#### 9-1-2 繼承性的運用

我們再回來探討建構資訊系統的基本目的，即是欲將現有存在或虛擬不存在的種種現象，轉換成電子資料並且可由資訊系統來處理。如此不但可以將真實交易、處理、流通等等行為，轉換成電子化處理，達到提高時效以及增加交易行為的可靠度（資訊系統沒有情感也不會有詐欺行為）。

關鍵在於如何將真實環境數位化，傳統語言利用靜態的結構變數描述種種現象，但所描述的每一種實體都是獨立的，很難建立實體之間共同連帶關係。任何一套電子資訊系統必須描述許多實體，在真實環境中這些實體之間有許多關聯，如果資訊環境中每種實體都各自獨立，當系統有任何變動或需擴充系統功能時，將非常的困難。

利用物件導向的概念描述則不然，產生一個物件後，可利用它再擴充成另外一個物件，即是新物件不再重新描述，只不過將舊的物件增加某些屬性或功能，保留舊物件的特性而產生新的物件。自然而然，新物件與舊物件就會保有許多連帶關係。從物件的功能性來談，是整合物件延伸另一個新的物件；從技術面來實現，即是整合多個類別擴充成另一個類別，這就是類別的繼承性。

我們用『人』的描述方法說明物件導向的設計模式。以台灣為範例，政府會針對每位國民給予適當的識別，方法是發給每位國民一張『國民身分證』，無論國人到任何地方洽公或需要表明身分時，只要『秀』出身分證大致上都能通行。在資料系統上，同樣需要紀錄每位國民的資料與身分識別，最簡單的方法即是將『國民身分證』資料轉換成電子檔，並允許利用資料系統儲存與處理，如圖 9-1 所示。

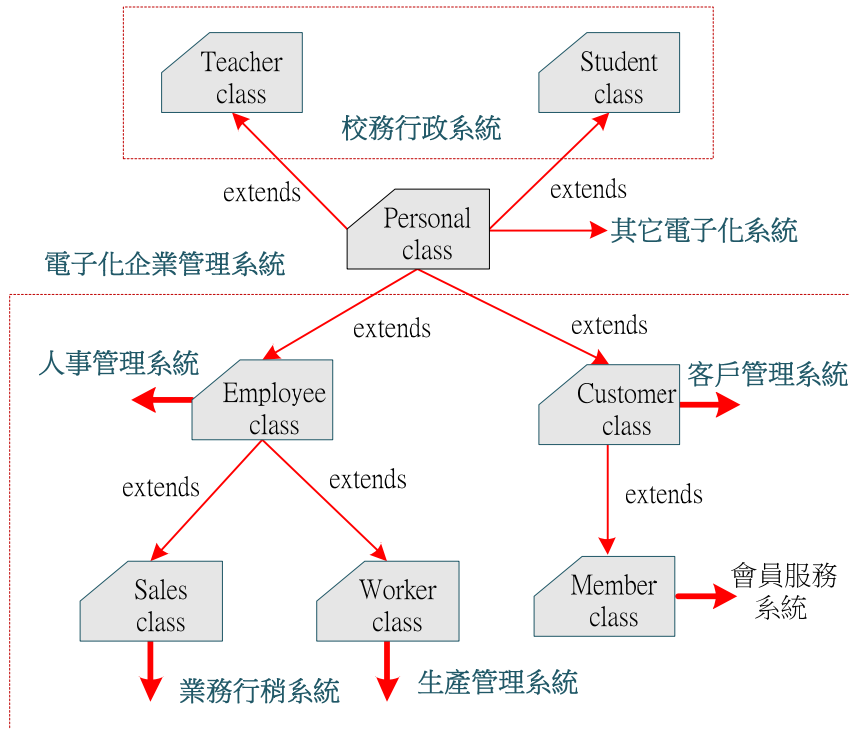


**圖 9-1 電子資料表示個人身分**

圖 9-1 係利用物件導向觀念來描述每位國民的身份資料，暫且不要去理會他的屬性是否能滿足資訊系統需求，反正官方認定的識別資料就是這樣。吾人利用 `Personal.class` 類別描述國人的基本資料，至於其他資訊系統所需的個人資料都由此延伸而來。如圖 9-2 所示，吾人可利用 `Personal` 類別擴充成學生 `Student`、`Employee`、`Customer` 等類別資料，當擴充成其他類別時，除了可以保持原類別屬性外，還可增加其他屬性來滿足所欲開發的資訊系統所需（如校務行政系統）。

### 9-1-3 繼承性的運用範例

以『展鵬網路行銷公司』為範例，他期望建立人事資訊系統，則可以引用 `Personal` 類別，再加某些公司所需的屬性，則得到 `Employee` 類別。`Employee` 類別是由 `Personal` 類別擴充(extends)而來，也得到 `Personal` 原來所擁有的屬性，則稱為 `Employee` 繼承了 `Personal` 所擁有的，這就是『繼承性』（如圖 9-2 所示）。



**圖 9-2 電子化系統的資料關聯**

物件導向的繼承性不同於『田橋』分財產，分給大兒子，小兒子就沒有，而是複製下來的意思。Employee 繼承了 Personal 的所有屬性，同樣的，Customer 也可以再繼承它的所有屬性，Student、Teacher 類別亦可。一般公司行號的人事系統大多會紀錄較詳細的員工資料，其他系統大多由此系統內擷取出適用的資料即可，因此可由 Employee 擴充(也許是縮小)出 Sales 與 Worker 類別，分別提供給業務系統與生產管理系統使用。由此可見，整個公司的員工資料都是來自 Employee 類別，不但可以節省開發時間、提高效率，也能使公司內資料統一。

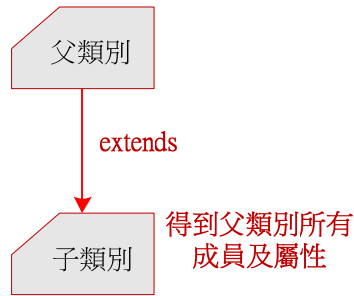
### 9-1-4 類別繼承的語法

繼承類別的語法如下：(如圖 9-3 所示)

```

class new_class extends old_class {
    class extends body
    .....
    .....
}
  
```

其中 class 與 extends 為 Java 的關鍵字。new\_class 首先導入 old\_class 的內容，並增加了 class extends body 所描述的部分。如此一來，new\_class 不但具有 old\_class 的特性(變數及方法)，也擴充了自己的變數與方法 (extends body 部分)。



**圖 9-3 父類別與子類別**

類別繼承性的重點說明如下：

- (1) 子類別繼承了父類別所有變數與方法成員，其屬性（如 public、private 或 protected）也如同父類別一樣。
- (2) 子類別除了繼承父類別之外，也可自行宣告所需的變數與方法成員。
- (3) 如果子類別所宣告的變數或方法與父類別相同名稱的話，則表示新的成員已覆蓋舊的成員（原父類別所擁有）。
- (4) 在成員有被覆蓋的情況下（如覆蓋成員的名稱為 name），也可利用 super 或 this 來區分父類別成員（super.name）或子類別成員（this.name）。
- (5) 子類別可以繼承兩個以上父類別，稱為多重繼承，必須採用 implement 關鍵字。

## 9-2 簡單的繼承關係

### 9-2-1 範例研討：建構人事資料

#### (A) 程式功能：Ex9\_1.java、Personal.java、Employee.java

『展鵬網路行銷公司』期望建立一套較完整的人事資訊系統，對每位員工包含有：身分證字號、姓名、性別、員工代號、服務部門、職位、地址、電話等資料。希望此系統所登錄資料能與外部其他系統結合，對內也可以讓其他資訊系統引用。規劃員工資料內容後，請製作一個簡單的輸入/輸出程式，測試是否可行，期望操作介面如下：

```

G:\Examples\chap12\Ex9_1>java Ex9_1
**** 建立員工基本資料 ****

請輸入員工姓名 =>張有志

身分證字號 =>M111234567
  
```

```

員工代號 (700 ~ 900) =>720
性別 (男/女) =>男
服務部門 =>製造課
職位 =>技術員
地址 =>高雄市三民區
電話 (##-#####) =>07-43123

**** 列印員工基本資料 ****

代號  證號  姓名  性別  部門  職位  電話  地址
(720 M111234567 張有志 男) 製造課  技術員  07-43123  高雄市三民區

```

## (B) 製作技巧研討：

該公司要求所建立的人事資訊系統，儘量可以與外界或內部其他資訊系統結合，即是人事資料的屬性儘量能相容於其他系統。因此，必須由對內與對外兩方面來思考，以下分別說明之。

### (1) 對外共通屬性：Personal.java

在台灣辨識國人之間大多採用哪些屬性，將這些屬性加入人事資訊系統內，與外界溝通則可透過它們。國人資料所採用的屬性有：姓名、身分證字號、性別、出生地、出生日，除了身分證字號外，在台灣內要找到兩個人其他資料都是相同的，應該很困難才對。於是將描述『國人』的屬性製作成 Personal 類別，如下：

```

01 // Personal.java
02
03 class Personal {
04     private String SID;           // 身分證號, 需 10 個字元
05
06     String name;                 // 姓名
07
08     String sex;                  // 性別 (男或女)
09
10     String birthDate;           // 出生日期 (年/月/日)
11
12     String bithPlace;           // 出生地
13
14     int setSID(String No) {      // 設定身分證號方法
15         if (No.length() != 10) {
16             System.out.printf("需 10 個字元, 請重新輸入 !!");
17             return 0;
18         }

```

```

19         else {
20             SID = No;
21             return 1;
22         }
23     }
24     String getSID() {           // 讀取身分證號碼方法
        return SID;
    }
}
    
```

上述範例中，Personal 類別包含 5 個變數成員，其中 SID (身分證號) 為隱藏性變數，利用 2 個方法成員 (setSID() 與 getSID()) 作為該變數的輸入與輸出埠口。Personal 類別制定完成之後，不僅可以由它延伸出管理公司內的人事系統，也可以製作出公司管理非本司員工的客戶管理系統，甚至其他應用管理系統。Personal 類別可存取成員有：(如圖 9-4 所示)

- (a) int setSID(String No)：設定身分證號碼。引數 No 格式正確則設定身分證號碼，並回傳 1 表示執行正常；否則回傳 0，表示身分證號碼格式不對。
- (b) String getSID()：讀取身分證號碼。
- (c) String name：儲存姓名之變數 (字串格式)。
- (d) String sex：儲存性別之變數。
- (e) String birthDate：儲存出生日之變數。
- (f) String birthPlace：儲存出生地之變數。

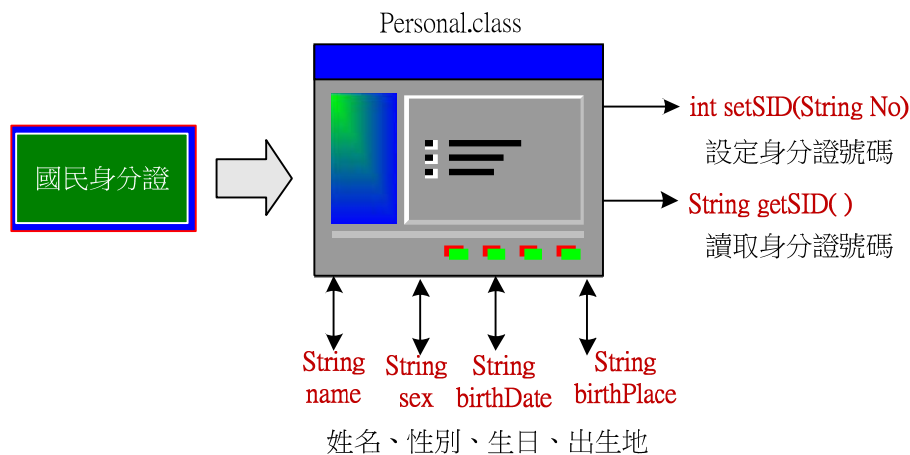


圖 9-4 人事資料基本類別

## (2) 對內相容屬性：Employee.java

人事系統內所登錄的員工資料，大多會其他系統所引用；規範員工資料屬性時，也必須考慮

到其他系統的相容性如何。首先我們增加了員工代號屬性，當作員工在公司內的唯一識別。如果還是利用身份證字號識別，它屬於字串型態，搜尋比對較不方便。另外，再依展鵬公司的環境，增列了：服務部門、職位、電話與地址等 4 個屬性，又引用對外共通屬性：身份證號、姓名與性別等 3 個屬性。合計共 8 個屬性（如圖 9-5 所示）。依此，我們建構了 Employee 類別，它繼承來自 Personal 屬性，程式範例如下：

```

01 //Employee.java
02
03 /* Personal.class 需於同目錄下 */
04 class Employee extends Personal {
05
06     int ID;                // 員工代號
07
08     String depart;        // 所屬部門
09
10     String position;     // 職 位
11
12     String addr;         // 地 址
13
14     String tel;          // 電 話
15
16 }

```

其實上述範例，Employee 類別繼承 Personal 屬性的所有成員，又增列 4 個成員，合計共有：SID、name、sex、birthDate、birthPlace（以上父類別 Personal 具有），ID、depart、position、addr 與 tel（以上由 Employee 宣告產生），合計 10 個變數成員，以及 setSID() 與 getSID() 兩個方法成員。但依照系統的需求，吾人也許會捨棄 birthDate 與 birthPlace 兩個變數成員不用（也許有一天會拿來用）。

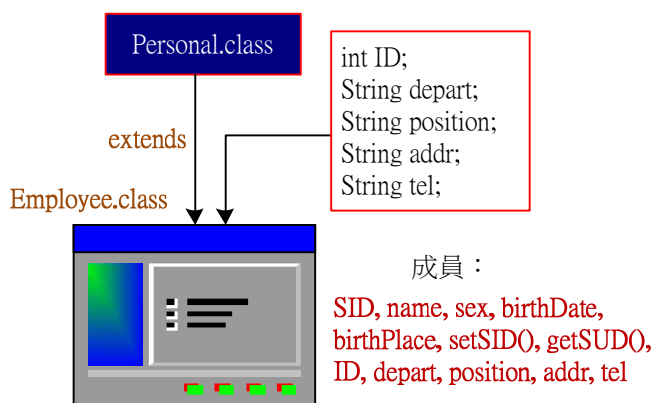


圖 9-5 Employee 類別產生

### (C) 程式範例：Ex9-1.java

完成後 Employee 類別的規劃，吾人則可以編制一個簡單程式來測試結果如何，程式範例如

下：

```
01 //Ex9_1.java
02
03 /* Employee.class 需於同目錄下 */
04 import java.io.*;
05 public class Ex9_1 {
06     public static void main(String args[]) throws IOException {
07         BufferedReader keyin = new BufferedReader(new
08             InputStreamReader(System.in));
09         Employee emp = new Employee();
10         String tempS;
11
12
13         System.out.printf("***** 建立工作員資料 *****\n");
14         System.out.printf("請輸入員工姓名 =>");
15             emp.name = keyin.readLine();
16         System.out.printf("身分證字號 =>");
17             tempS = keyin.readLine();
18             while (emp.setSID(tempS) == 0) {
19                 System.out.printf("請重新輸入 =>");
20                 tempS = keyin.readLine();
21             }
22
23         System.out.printf("員工代號 (700 ~ 900) =>");
24             emp.ID = Integer.parseInt(keyin.readLine());
25         System.out.printf("性別 (男/女) =>");
26             emp.sex = keyin.readLine();
27         System.out.print("服務部門 =>");
28             emp.depart = keyin.readLine();
29         System.out.printf("職位 =>");
30             emp.position = keyin.readLine();
31         System.out.printf("地址 =>");
32             emp.addr = keyin.readLine();
33         System.out.printf("電話 (##-#####) =>");
34             emp.tel = keyin.readLine();
35
36         System.out.printf("***** 列印員工基本資料 *****\n");
37         System.out.printf("代號 證號 姓名 性別 ");
38         System.out.printf("部門 職位 電話 地址\n");
39         System.out.printf("(%d %s %s %s)",
40             emp.ID, emp.getSID(), emp.name, emp.sex);
41         System.out.printf("\t%s %s %s %s\n",
```



```
        emp.depart, emp.position, emp.tel, emp.addr);
    }
}
```

製作完成上述 3 個原始程式之後，即可將他編譯成中界檔 ( bytecode )，但需由 Personal.java 開始，接著是 Employee.java，最後才編譯 Ex9\_1.java，執行範例如下：

```
G:\Examples\chap12\Ex9_1>javac Personal.java

G:\Examples\chap12\Ex9_1>javac Employee.java

G:\Examples\chap12\Ex9_1>javac Ex9_1.java

G:\Examples\chap12\Ex9_1>dir/b/w
Employee.java
Personal.java
Ex9_1.java
Personal.class
Employee.class
Ex9_1.class
```

## 9-2-2 自我挑戰：建構客戶資料

### ( A ) 程式功能：PM9\_1.java

展鵬行銷公司希望建立一套客戶管理系統，針對每位客戶登錄：身份證字號、姓名、性別、生日、購買商品、購買金額、電話與地址等 8 種資料。請規劃後編寫一簡單程式測試是否可行，期望操作介面如下：

```
G:\Examples\chap12\PM9_1>java PM9_1
**** 建立客戶基本資料 ****
請輸入客戶姓名 =>劉靜儀
身分證字號 =>A123412345
性別 (男/女) =>女
生日(年/月/日) =>71/3/12
購買商品 =>化妝水
購買金額 =>5000
地址 =>台北市中山北路
電話 (##-#####) =>02-3451234
```

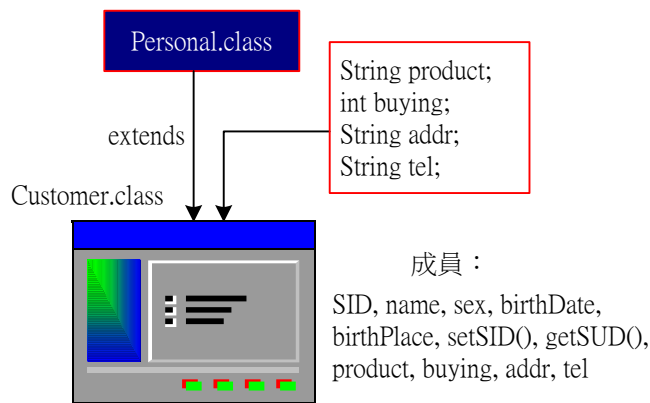
```
**** 列印客戶基本資料 ****
身份證號 姓名 性別 生日 商品 金額 電話 地址
(A123412345 劉靜儀 女 71/3/12) 化妝水 5000 02-3451234 台北市中山北路
```

**( B ) 製作技巧提示：**

如同範例 Ex9-1 一樣，基本資料還是繼承自 Personal 類別較為妥當，編制 Customer 類別的原始程式如下：

```
01 //Customer.java
02
03 /* Personal.class 需於同目錄下 */
04 class Customer extends Personal {
05
06     String product;           // 購買產品
07
08     int buying;               // 購買金額
09
10     String addr;             // 地 址
11
11     String tel;              // 電 話
    }
}
```

由 Customer 類別產生了 product、buying、addr 與 tel 等 4 個屬性，再繼承自 Personal 類別取出 SID、name、sex 與 birthDate 等 4 個屬性 ( 如圖 9-6 所示 )，即可滿足展鵬公司的要求。



**圖 9-6 Customer 類別產生**

**( C ) 程式範例：**

吾人可利用 Customer 類別宣告客戶物件，再依序輸入資料與輸出客戶所有資料，請參考 Ex9\_1.java 範例製作本範例的主程式 ( PM9-1.java )，本書不再贅言。

## 9-3 覆蓋繼承成員

### 9-3-1 繼承性的覆蓋

子類別成員的名稱與父類別成員相同時，則表示覆蓋了父類別成員，如果需要取用原父類別成員則利用 `super` 與 `this` 區分。譬如父類別有 `addr` 變數成員，子類別除了繼承父類別的 `addr` 成員之外，也自行再宣告一個 `addr` 變數成員（兩者名稱相同）。基本上，子類別下的 `addr` 會覆蓋了父類別的 `addr`，但事實上還是存在兩個不同變數（記憶體位址不同），吾人可利用 `super.addr`（父類別）與 `this.addr`（子類別或 `addr` 即可）來分辨。

### 9-3-2 範例研討：員工薪資資料建檔

#### (A) 程式功能：Ex9\_2.java

展鵬行銷公司期望建立一套員工薪資管理系統，目前思考該系統針對某位員工資料為：員工姓名、員工代號、底薪、加班時數、戶籍地址與通訊地址等 5 種屬性。其中依勞基法規定底薪不可以低於 15800 元，每月加班時數不可超過 45 小時。登錄資料時如有違背規定，系統會重複要求重新輸入。設計完成資料屬性後，請編寫一簡單程式，測試系統是否可做資料輸入與輸出，期望操作介面如下：

```
G:\Examples\chap12\Ex9_2>java Ex9_2
**** 建立員工薪資資料 ****
請輸入員工姓名 =>劉美女
員工代號 (700 ~ 900) =>740
戶籍地址 =>高雄市三民區
通訊地址 =>台北市內湖區
底薪 =>50000
加班時數 =>30

**** 列印員工薪資資料 ****
員工代號 姓名 底薪 加班 戶籍地址 通訊地址
(740 劉美女 ) 50000 30 高雄市三民區 台北市內湖區
```

#### (B) 製作技巧研討：

公司裡大多希望各系統之間的資料能夠相容，對於爾後資料搬移會比較容易。之前我們幫展鵬公司設計了人事管理系統 ( Ex9\_1 範例 )，也制定了公司內人事資料的標準格式 ( Employee 類別 )。因此，希望員工薪資資料也是由人事資料延伸下來。Employee 類別包含有：員工代號 ( ID )、身分證號 ( SID )、姓名 ( name )、性別 ( sex )、服務部門 ( depart )、職位 ( position )、電話 ( tel ) 與地址 ( addr )。然而依照薪資管理系統，需要每位員工的：員工代號 ( ID )、姓名 ( name )、底薪 ( payment )、加班時數 ( extra )、通訊地址 ( addr ) 與戶籍地址 ( addr )；其中員工代號、姓名與戶籍地址可以繼承自 Employee 類別，其他屬性可能需要自行宣告產生 ( Salary 類別，如圖 9-7 所示 )。但有兩個地址名稱相同，一個是父類別的 addr 代表戶籍地址，另一者是子類別的 addr 代表通訊地址，因此必須利用 super.addr 與 addr 作之間的區別。依照上述的概念，我們製作了宣告員工薪資資料的 Salary 類別，再編寫主程式 ( Ex9\_2.java ) 引用該類別，測試其結果如何。

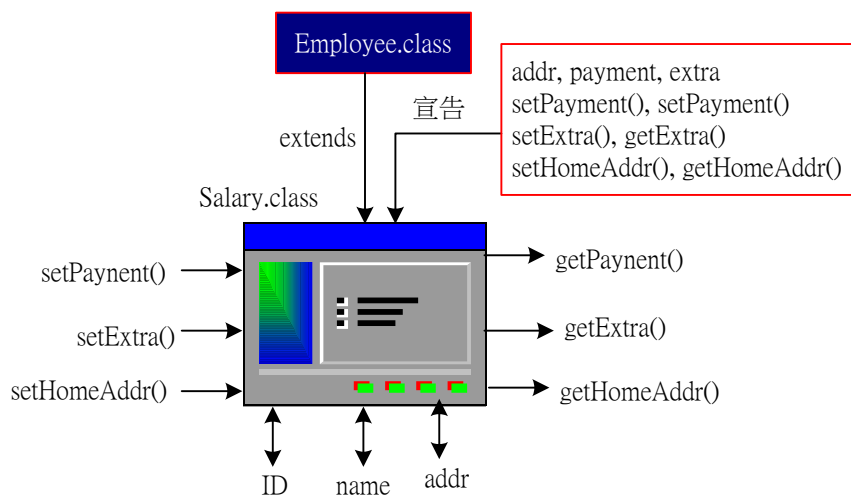


圖 9-7 Salary 類別產生

### ( C ) 程式範例：Salary.java

依照上述的概念，我們製作了員工薪資類別 Salary，也繼承了員工基本資料類別 Employee，原始程式如下：

```

01 //Salary.java
02
03 /* Employee.class 需於同目錄下 */
04 class Salary extends Employee {
05
06     String addr;           // 通訊地址
07
08     private int payment;   // 底    薪 > 15800
09
10     private int extra;    // 加班時數 < 45
  
```

```
11
12     /*** 宣告物件方法 ***/
13
14     int setPayment(int pay) {    //設定底薪方法, 1: 正常, 0: 錯誤
15         if (pay < 15800) {
16             System.out.printf("底薪不可低於 15800 元\n");
17             return 0;
18         }
19         else {
20             payment = pay;
21             return 1;
22         }
23     }
24
25     int getPayment() {        // 讀取底薪方法
26         return payment;
27     }
28
29     int setExtra(int ex) {    // 設定加班時數, 1: 正常, 0: 錯誤
30         if (ex > 45) {
31             System.out.printf("加班時數不可超過 45 小時\n");
32             return 0;
33         }
34         else {
35             extra = ex;
36             return 1;
37         }
38     }
39
40     int getExtra() {        // 取得加班時數方法
41         return extra;
42     }
43     /* 存取父類別變數成員的方法 */
44     void setHomeAddr(String ad) {
45         super.addr = ad;
46     }
47     String getHomeAddr() {
48         return super.addr;
49     }
50 }
```

Salary 類別包含下列成員：( 僅擷取本範例所需部份 )

- (1) 變數成員：ID、name、super.addr ( 以上繼承自 Employee )，以及 payment、Extra 與 addr ( 通訊地址，以上是自行宣告 )。
- (2) 方法成員：setPayment()、getPayment、setExtra()、getExtra、setHomeAddr()、以及

getHomeAddr()，最後兩個方法是設定與取得父類別的 addr 屬性（戶籍地址）

### (D) 程式範例：Ex9\_2.java

設定完成 Salary 類別之後，吾人製作一個簡單程式（Ex9\_2.java），功能是輸入員工薪資資料，再印出輸入的結果如何；其中儲存員工資料的物件，即是由 Salary 類別生產生。

```
01 //Ex9_2.java
02
03 /* Salary.class 需於同目錄下 */
04 import java.io.*;
05 public class Ex9_2 {
06     public static void main(String args[]) throws IOException {
07         BufferedReader keyin = new BufferedReader(new
08             InputStreamReader(System.in));
09         Salary sal = new Salary();
10         String tempS;
11         int value;
12         double m;
13         System.out.printf("***** 建立員工薪資資料 *****\n");
14
15
16         System.out.printf("請輸入員工姓名 =>");
17         sal.name = keyin.readLine();
18         System.out.printf("員工代號 (700 ~ 900) =>");
19         sal.ID = Integer.parseInt(keyin.readLine());
20         System.out.printf("戶籍地址 =>");
21         tempS = keyin.readLine();
22         sal.setHomeAddr(tempS);
23         System.out.printf("通訊地址 =>");
24         sal.addr = keyin.readLine();
25         System.out.printf("底薪 =>");
26         value = Integer.parseInt(keyin.readLine());
27         while (sal.setPayment(value) == 0) {
28             System.out.printf("請重新輸入 =>");
29             value = Integer.parseInt(keyin.readLine());
30         }
31         System.out.printf("加班時數 =>");
32         value = Integer.parseInt(keyin.readLine());
33         while (sal.setExtra(value) == 0) {
34             System.out.printf("請重新輸入 =>");
35             value = Integer.parseInt(keyin.readLine());
36         }
37     }
38 }
39
```

```
40         }
41
42         System.out.printf("\n**** 列印員工薪資資料 ****\n");
43
44         System.out.printf("員工代號 姓名 底薪 加班");
45
46         System.out.printf("\t 戶籍地址\t 通訊地址\n");
47         System.out.printf("(%d %s) %d %d",
48                             sal.ID, sal.name, sal.getPayment(), sal.getExtra());
49         System.out.printf("\t%s %s\n", sal.getHomeAddr(), sal.addr);
50     }
51 }
```

### 9-3-3 自我挑戰：建立客戶會員資料

#### (A) 程式功能：PM9\_2.java

『展鵬網路行銷公司』為了增進與客戶之間的關係，而制定了會員制度；首先思考以下策略：

- (1) 客戶累積購買金額超過 3 萬元，可選擇加入會員，一般會員 (C 級) 購買商品打九折。
- (2) 新進會員有效期為 1 年，如當年累積金額超過 3 萬元者，會員有效期延長 1 年。
- (3) 如果一般會員 (C 級) 當年購買累積金額超過 3 萬元後，自動升級為白金會員 (B 級)，購物打八折。
- (4) 如果白金會員當年購物超過 6 萬元後，自動升級為鑽石會員 (A 級)，購物打七折。

為了客戶隱私權問題，系統不可任意查閱客戶身份證字號與出生日期，但有時候公司還是需要知道客戶身份證號碼。經過多方的考慮，客服部目前規劃會員的基本資料為：身份證號碼 (SID)、會員號碼 (MID)、姓名 (name)、累積金額 (buying)、有效日期 (validDate)、地址 (addr) 與電話 (tel)。其中身份證號碼需要特定程序才可以查閱或修改，以保持客戶的隱私權。另外，會員號碼由一個英文字母連結 5 個數字，英文字 A、B、C 代表會員級數 (鑽石、白金或一般)，5 個數字表示會員號碼 (不變的)。則可能會員代號是 A00001 ~ C10000，設計這樣的目的是會員身份可能隨購買金額改變 (上升或下降)。期望操作介面如下：

```
G:\Examples\chap12\PM9_2>java PM9_2
**** 建立會員基本資料 ****
請輸入客戶姓名 =>張智慧
身分證字號 =>K121834561
```

```

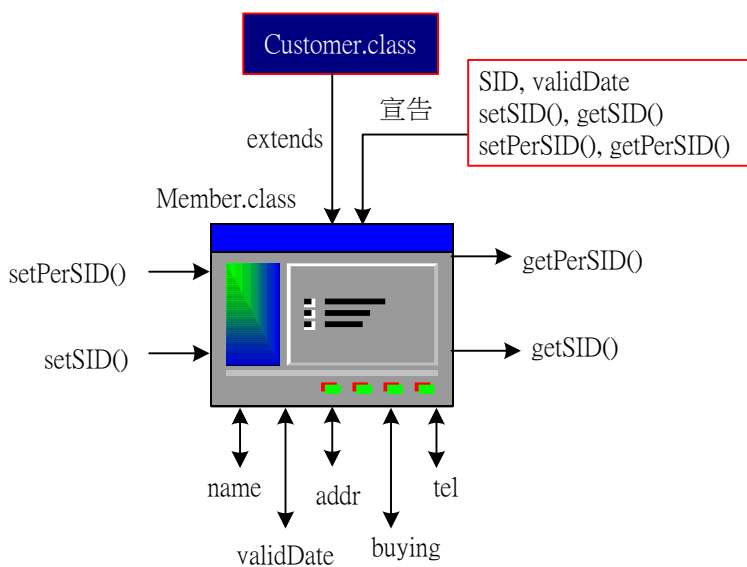
購買金額 =>70023
請輸入會員代號 (10000 ~ 99999) =>34021
有效日期 (年/月/日) =>2009/3/12
地址 =>台中市中港路
電話 (##-#####) =>04-2349871

**** 列印會員基本資料 ****

身分證號    代號    姓名    有效日期    累積金額    電話    地址
(K121834561 B34021 張智慧 2009/3/12)    70023    04-2349871    台中市中港路
    
```

**(B) 製作技巧提示：**

依照公司規定，客戶需累積購買金額後，才可以成為會員，享有折扣優惠，由此可見，會員資料大多來自客戶資料，設計會員類別 ( Member ) 時，也需繼承自客戶類別 ( Customer )，請參閱範例 PM9\_1 )。Customer 類別具有：身份證號 ( SID )、姓名 ( name )、性別 ( sex )、生日 ( birthDate )、購買商品 ( product )、金額 ( buying )、地址 ( addr ) 與電話 ( tel )。如 Member 類別繼承自 Employee 類別，可引用：姓名 ( name )、身份證號 ( SID )、購買金額 ( buying )、電話 ( tel ) 與地址 ( addr )，再自行宣告會員代號 ( SID ) 與有效日期 ( validDate ) 即可。但身份證號與會員代號變數名稱相同，吾人以 super.SID 與 SID 區分 ( 如圖 9-8 所示 )。



**圖 9-8 Member 類別產生**

另外，系統要求會員代號是由會員類別 ( A、B、C )，與 5 個數字的會員號碼組成 ( 10000 ~ 99999 )，又會員類別是由累積購買金額判定。因此，輸入會員資料時，僅輸入會員號碼與累積金額，



系統會自動產生會員代號 ( A10000 ~ C10000 )。

吾人依照上述概念，建立 Member 類別，編寫一主程式引用該類別，再輸入會員資料看系統是否可行。

### ( C ) 程式範例提示：Member.java

Member 類別產生的過程是 Personal => Customer => Member，因此 Member 類別裡可引用原 Personal 與 Customer 所宣告的任何成員 ( 變數或方法 )。

```
01 //Member.java
02
03 /* Customer.class 需於同目錄下 */
04
05 class Member extends Customer {
06
07     private String SID;           // 會員代號
08     String validDate;           // 有效日期
09
10
11     /*** 宣告物件方法 ***/
12
13     int setSID(String M_SID) { //設定會員代號方法, 1: 正常, 0: 錯誤
14         String level = "";
15         int flag;
16         if (buying < 30000) {
17             System.out.printf("請先輸入購買金額或\n");
18             System.out.printf("購買金額需超過 3 萬元\n");
19             flag=0;
20         }
21     }
22     else if (buying <60000 ) {
23         level = "C";
24         flag = 1;
25     }
26     else if (buying <90000 ) {
27         level = "B";
28         flag=1;
29     }
30     else {
31         level = "A";
32         flag = 1;
33     }
34     if (flag == 0)
35         return 0;           // 執行失敗
36 }
```

```
37         SID = level.concat(M_SID);    // 字串連結方法
38
39         return 1;                      // 執行成功
40     }
41     String getSID() {                  // 讀取會員代號方法
42         return SID;
43     }
44
45     int setPerSID(String S_SID) {      // 設定身分證號方法
46         int k = super.setSID(S_SID);   // 呼叫父類別方法
47         return k;
48     }
49
50     String getPerSID() {               // 讀取身分證號方法
51         String S_SID = super.getSID(); // 呼叫父類別方法
52         return S_SID;
53     }
54 }
```

其中，`super.setSID()` (第 40~43 行) 呼叫父類別 (Personal 類別) 方法，功能是設定 SID 變數成員 (身分證號) 內容，執行正常則傳回 1；否則傳回 0。然而，Member 類別可直接引用成員有：

- (1) 變數成員：姓名 (name)、有效日期 (validDate)、購買金額 (buying)、電話 (tel) 與地址 (addr)。
- (2) 方法成員：設定身分證號 (int setPerSID(String))、讀取身分證號 (String getPerSID())、設定會員代號 (int setSID(String)) 與讀取會員代號 (String getSID)。

### (D) 程式範例提示：PM9\_2.java

吾人可利用 Member 類別宣告會員物件，再依序輸入資料與輸出會員所有資料，請參考 Ex9\_2.java 範例製作本範例的主程式 (PM9-2.java)，本書不再贅言。

## 9-4 類別的多形性運用

### 9-4-1 何謂多形性

繼承性裡有一個重要的特點，子類別與父類別所宣告的成員 (變數或方法) 名稱相同時，兩者還是各自獨立的，並沒有真正被覆蓋掉。

如果同一類別內宣告相同名稱的成員(變數或方法)，那又是如何？這也是物件導向另一個特點。基本上，同一類別內變數名稱(或方法名稱)是不可以被重複宣告使用的，但對於相同名稱但資料型態不同者，是被允許的。因為名稱相同但資料型不同的變數之間是各自獨立的。另一方面，名稱相同但輸入/輸出引數不同的方法成員之間也是各自獨立的。許多實際應用環境需要這種特性，譬如，加法運算器大多允許各種資料型態的運算，如果不同資料型態需引用不同的加法器，可能會產生很困擾。但話說回來，電腦在處理各種運算時，的確需考慮是何種資料型態，才可以做運算工作，這又是如何？其實很簡單，吾人製作了許多不同資料型態的加法器，但都採用相同名稱；當系統操作時，依照所欲處理的資料型態再引用哪一個加法器。簡單的說，同一種變數(或方法)名稱，可引用多種資料型態的處理(各自不同的處理器)，這就是類別的『多形性』。

## 9-4-2 程式範例：多功能加法器

### (A) 程式範例：Ex9\_4.java

請製作一個加法器功能是，允許：整數 + 整數 = 整數；整數 + 浮點數 = 浮點數；浮點數 + 整數 = 浮點數；浮點數 + 浮點數 = 浮點數；等 4 種運算處理。當呼叫該加法器時，他會依照所欲處理資料的型態，決定處理方式。請製作完成之後，再編寫一主程式呼叫該工具，觀察是否執行正常；期望操作介面如下：

```
G:\Examples\chap12\Ex9_3>javac Arith.java
G:\Examples\chap12\Ex9_3>javac Ex9_3.java
G:\Examples\chap12\Ex9_3>java Ex9_3
整數 + 整數 = 整數
    Arith.Add(5, 5) = 10
浮點數 + 整數 = 浮點數
    Arith.Add(2.9, 5) = 7.90
整數 + 浮點數 = 浮點數
    Arith.Add(5, 2.9) = 7.90
浮點數 + 浮點數 = 浮點數
    Arith.Add(2.9, 2.9) = 5.80
```

### (B) 製作技巧研討：Arith.java

吾人利用 Arith 類別製作了多功能加法器(靜態方法)，程式範例如下：(如圖 9-9 所示)

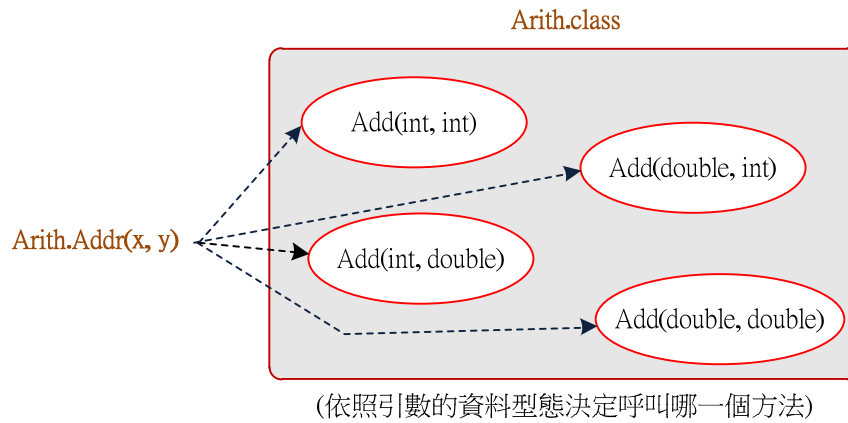


圖 9-9 Arith 類別架構

```

01 //Arith.java
02
03 class Arith {
04     static int Add(int x, int y) {
05         System.out.printf("整數 + 整數 = 整數\n");
06         return (x+y);
07     }
08     static double Add(double x, int y) {
09         System.out.printf("浮點數 + 整數 = 浮點數\n");
10         return (x+y);
11     }
12     static double Add(int x, double y) {
13         System.out.printf("整數 + 浮點數 = 浮點數\n");
14         return (x+y);
15     }
16     static double Add(double x, double y) {
17         System.out.printf("浮點數 + 浮點數 = 浮點數\n");
18         return (x+y);
19     }
20 }

```

**(C) 程式範例：Ex9\_3.java**

Ex9\_3.java 程式裡引用 Arith.Add() 類別方法，它會依照不同的輸入資料型態，選擇執行哪一個函數，程式範例如下：

```

01 //Ex9_3.java
02
03 /* Arith.class 需於同目錄下 */
04
05 public class Ex9_3 {

```

```
06 public static void main(String args[]) {
07     int a=5;
08     double b=2.9;
09
10     System.out.printf("\tArith.Add(5, 5) = %d\n", Arith.Add(a, a));
11     System.out.printf("\tArith.Add(2.9, 5) = %.2f\n", Arith.Add(b, a));
12     System.out.printf("\tArith.Add(5, 2.9) = %.2f\n", Arith.Add(a, b));
13     System.out.printf("\tArith.Add(2.9, 2.9) = %.2f\n", Arith.Add(b, b));
14 }
15 }
```

### 9-4-3 自我挑戰：多功能大小比較器

#### (A) 程式功能：PM9\_3

請製作一套多功能比較器，功能是允許輸入 2 ~ 5 個數字 (浮點數運算)，再輸出其中最大值與最小值。請先製作一個比較器 (Max() 或 Min()) 類別，可隨輸入參數數量，決定執行哪一個方法。主程式 (PM9\_3.java) 再引用該方法，測試執行結果是否正常。期望操作介面如下：

```
G:\Examples\chap12\PM9_3>java PM9_3
請輸入兩個數 (如: 34 45.2) =>42.7 98.6
最大值 = 98.60, 最小值 = 42.70

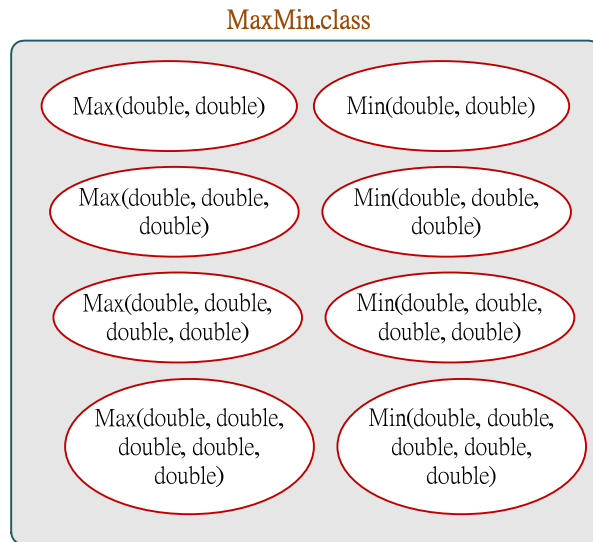
請輸入 3 個數 =>23.8 78 12.6
最大值 = 78.00, 最小值 = 12.60

請輸入 4 個數 =>44.6 31 67 25.9
最大值 = 67.00, 最小值 = 25.90

請輸入 5 個數 =>78.9 45.2 72.6 41 73.5
最大值 = 78.90, 最小值 = 41.00
```

#### (B) 製作技巧提示：MaxMin.java

吾人可製作一個 MaxMin 類別，包含 Max() 與 Min() 類別方法，部分程式範例如下：(如圖 9-10 所示)



(依照引數的個數決定呼叫哪一個方法)

**圖 9-10 MaxMin 類別架構**

```

01 //MaxMin.java
02
03 class MaxMin {
04     /* 尋找最大值方法 */
05     static double Max(double num1, double num2) {
06         if (num1 > num2)
07             return num1;
08         else
09             return num2;
10     }
11     static double Max(double num1, double num2, double num3) {
12         double num = Max(num1, num2);
13         if (num > num3)
14             return num;
15         else
16             return num3;
17     }
18     static double Max(double num1, double num2, double num3,
19                       double num4) {
20         double num = Max(num1, num2, num3);
21         .....
22         .....
23     }
24     static double Max(double num1, double num2, double num3,
25                       double num4, double num5) {
26         .....
27         .....
28     }
29 }
30

```

```

31     /* 尋找最小值方法 */
32     static double Min(double num1, double num2) {
33         if (num1 < num2)
34             return num1;
35         else
36             return num2;
37     }
38     static double Min(double num1, double num2, double num3) {
39         double num = Min(num1, num2);
40         if (num < num3)
41             return num;
42         else
43             return num3;
44     }
45     ..... ( 請自行編寫 )
46     .....
    }

```

- (1) 第 12 行：『double num = Max(num1, num2);』。呼叫本類別的 Max( double double) 方法。
- (2) 第 20 行：『double num = Max(num1, num2, num3);』 呼叫本類別內的其他方法。

### ( C ) 製作技巧提示：PM9\_3.java

完成 MaxMin 類別之後，即可製作一個主程式，測試其功能如何。重點提示如下：

```

01 //PM9_3.java
02
03 /* MaxMin.class 需於同一目錄下 */
04
05 import java.util.Scanner;
06 public class PM9_3 {
07     public static void main(String args[]) {
08         Scanner keyin = new Scanner(System.in);
09         double num1, num2, num3, num4, num5;
10         double max, min;
11         System.out.printf("請輸入兩個數 (如: 34 45.2) =>");
12         num1 = keyin.nextDouble();
13         num2 = keyin.nextDouble();
14         max = MaxMin.Max(num1, num2);
15         min = MaxMin.Min(num1, num2);
16         System.out.printf("最大值 = %.2f, 最小值 = %.2f\n", max, min);
17
18
19         System.out.printf("\n 請輸入 3 個數 =>");
20

```

```
21      .....
22      .....
23      max = MaxMin.Max(num1, num2, num3);
24      min = MaxMin.Min(num1, num2, num3);
25      .....
26      .....
27      max = MaxMin.Max(num1, num2, num3, num4);
28      min = MaxMin.Min(num1, num2, num3, num4);
29      .....
30      .....
31      }
    }
```

MaxMin.java 與 PM9\_3.java 需於同目錄下，並經編譯 ( javac ) 得到 MaxMin.class 與 PM9\_3.class 兩中介碼檔案。