

## 繼承性-繼承性的覆蓋

父類別有 `add` 成員，子類別繼承時也加入 `add` 成員，兩成員的區分如下：

`super.add` 父類別中成員

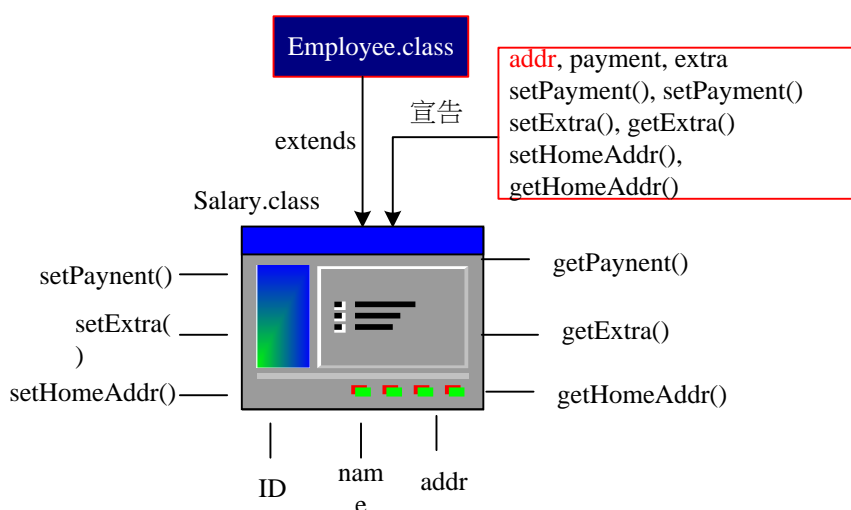
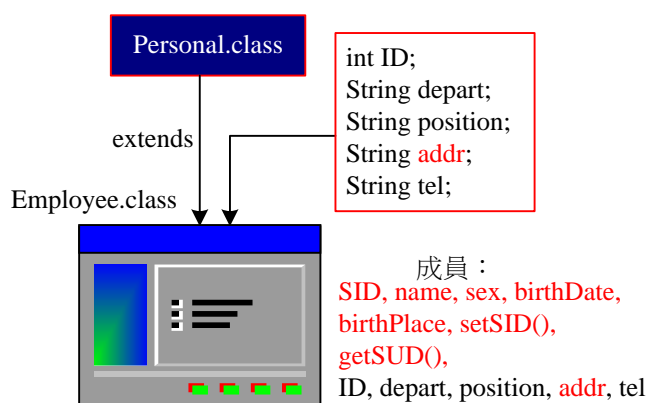
`this.add` 子類別中成員

**覆蓋成員 - 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，如示範。

(B) 製作技巧研討：



Salary 類別產生

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。

### (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     private int payment;   // 底    薪 > 15800
08     private int extra;    // 加班時數 < 45
09
10     /** 宣告物件方法 */
11     int setPayment(int pay) { //設定底薪方法, 1: 正常, 0: 錯誤
12         if (pay < 15800) {
13             System.out.printf("底薪不可低於 15800 元\n");
14             return 0;
15         }
16         else {
17             payment = pay;
18             return 1;
19         }
20     }
21
22     int getPayment() {      // 讀取底薪方法
23         return payment;
24     }
25     int setExtra(int ex) { // 設定加班時數, 1: 正常, 0: 錯誤
26         if (ex > 45) {
27             System.out.printf("加班時數不可超過 45 小時\n");
28             return 0;
29         }
30         else {
31             extra = ex;
32             return 1;
33         }
34     }
35     int getExtra() {       // 取得加班時數方法
36         return extra;
37     }
38     /* 存取父類別變數成員的方法 */
39     void setHomeAddr(String ad) {
40         super.addr = ad;
41     }
42     String getHomeAddr() {
43         return super.addr;
44     }
45 }
```

### (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。(請參閱書本)