

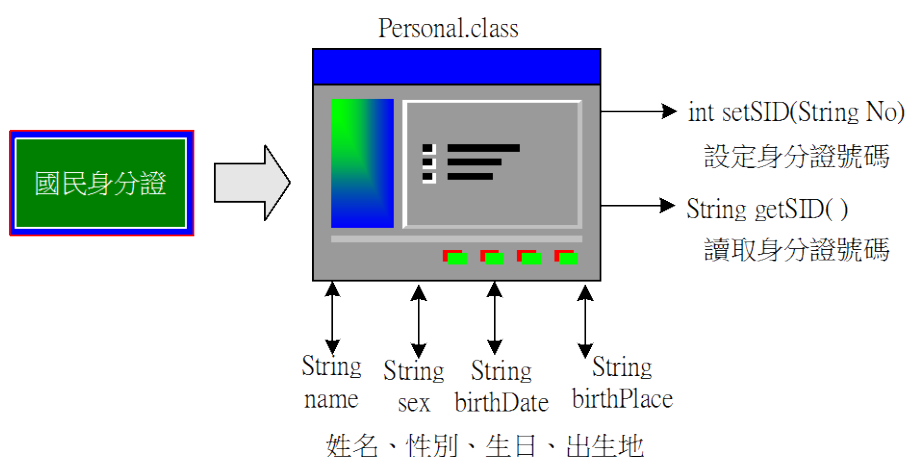
類別繼承範例：建構人事資料

(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，如操作示範。

(B) 製作技巧研討：

- 對外共通屬性：Personal.java



描述『國民』的屬性製作成 Personal 類別，如下：

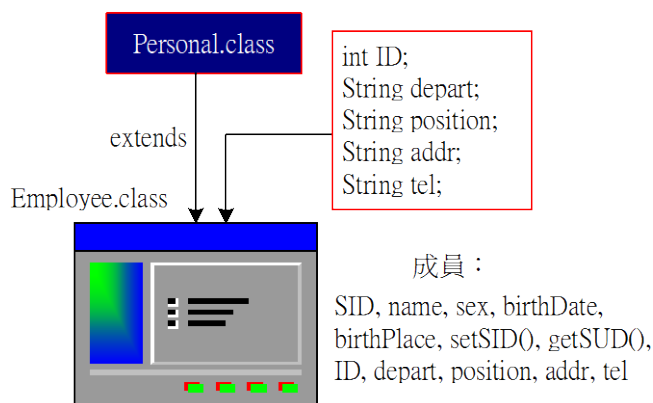
```

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

```

18         return 1;
19     }
20 }
21 String getSID() {           // 讀取身分證號方法
22     return SID;
23 }
24 }
    
```

● 對內相容屬性：Employee.java



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     String depart;   // 所屬部門
08     String position; // 職 位
09     String addr;     // 地 址
10     String tel;      // 電 話
11 }
12 }
    
```

(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     System.out.printf("***** 建立工作員資料 *****\n");
13     System.out.printf("請輸入員工姓名 =>");
14         emp.name = keyin.readLine();
15     System.out.printf("身分證字號 =>");
16         tempS = keyin.readLine();
17         while (emp.setSID(tempS) == 0) {
18             System.out.printf("請重新輸入 =>");
19             tempS = keyin.readLine();
20         }
21     System.out.printf("員工代號 (700 ~ 900) =>");
22         emp.ID = Integer.parseInt(keyin.readLine());
23     System.out.printf("性別 (男/女) =>");
24         emp.sex = keyin.readLine();
25     System.out.print("服務部門 =>");
26         emp.depart = keyin.readLine();
27     System.out.printf("職位 =>");
28         emp.position = keyin.readLine();
29     System.out.printf("地址 =>");
30         emp.addr = keyin.readLine();
31     System.out.printf("電話 (##-#####) =>");
32         emp.tel = keyin.readLine();
33
34
35     System.out.printf("***** 列印員工基本資料 *****\n");
36     System.out.printf("代號 證號 姓名 性別 ");
37     System.out.printf("部門 職位 電話 地址\n");
38     System.out.printf("(%d %s %s %s)",
39         emp.ID, emp.getSID(), emp.name, emp.sex);
40     System.out.printf("\t%s %s %s %s\n",
41         emp.depart, emp.position, emp.tel, emp.addr);
42
43 }
44 }
```