



# 程序设计实习

郭炜 微博 <http://weibo.com/guoweiofpku>

<http://blog.sina.com.cn/u/3266490431>

刘家瑛 微博 <http://weibo.com/pkuliujiaying>



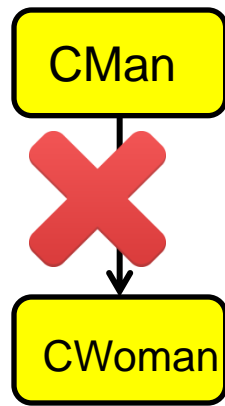
# 继承关系和复合关系 (P231)

# 类之间的两种关系

- 继承：“是”关系。
  - 基类 A，B 是基类 A 的派生类。
  - 逻辑上要求：“一个 B 对象也是一个 A 对象”。
- 复合：“有”关系。
  - 类 C 中“有”成员变量 k，k 是类 D 的对象，则 C 和 D 是复合关系
  - 一般逻辑上要求：“D 对象是 C 对象的固有属性或组成部分”。

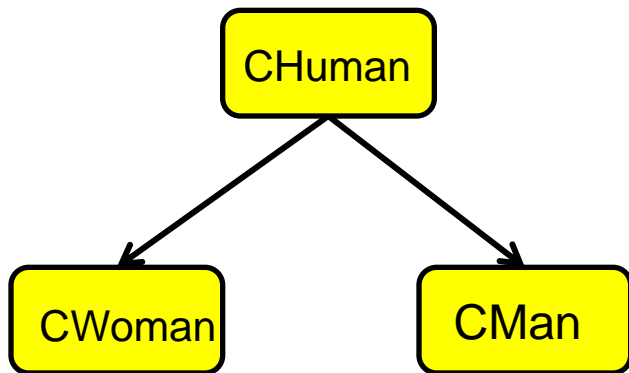
# 继承关系的使用

- 写了一个 CMan 类代表男人
- 后来又发现需要一个CWoman类来代表女人
- CWoman类和CMan类有共同之处
- 就让CWoman类从CMan类派生而来，是否合适？
- 是**不合理的**！因为“一个女人也是一个男人”从逻辑上不成立！



# 继承关系的使用

好的做法是概括男人和女人共同特点，  
写一个 CHuman类，代表“人”，然后CMan和CWoman都从CHuman派生。



# 复合关系的使用

- 几何形体程序中，需要写“点”类，也需要写“圆”类

```
class CPoint
{
    double x,y;
};
```



```
class CCircle:public CPoint
{
    double r;
};
```

# 复合关系的使用

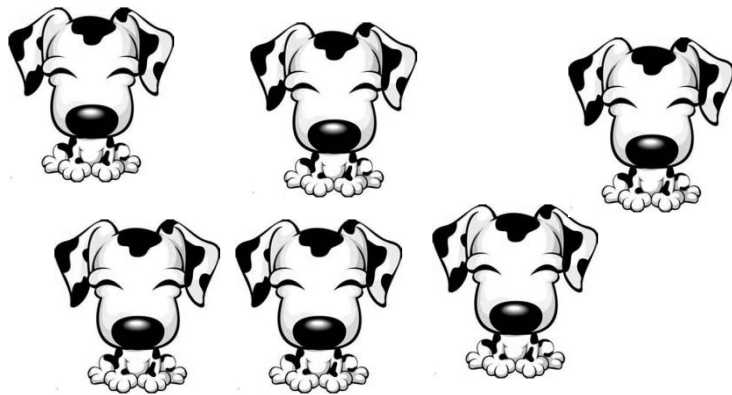
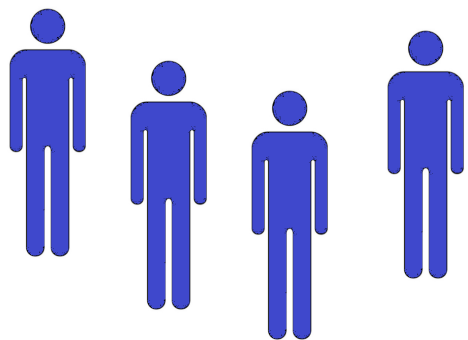
- 几何形体程序中，需要写“点”类，也需要写“圆”类，两者的关系就是复合关系 ---- 每一个“圆”对象里都包含(有)一个“点”对象，这个“点”对象就是圆心

```
class CPoint
{
    double x,y;
    friend class CCircle;
    //便于CCircle类操作其圆心
};
```

```
class CCircle
{
    double r;
    CPoint center;
};
```

# 复合关系的使用

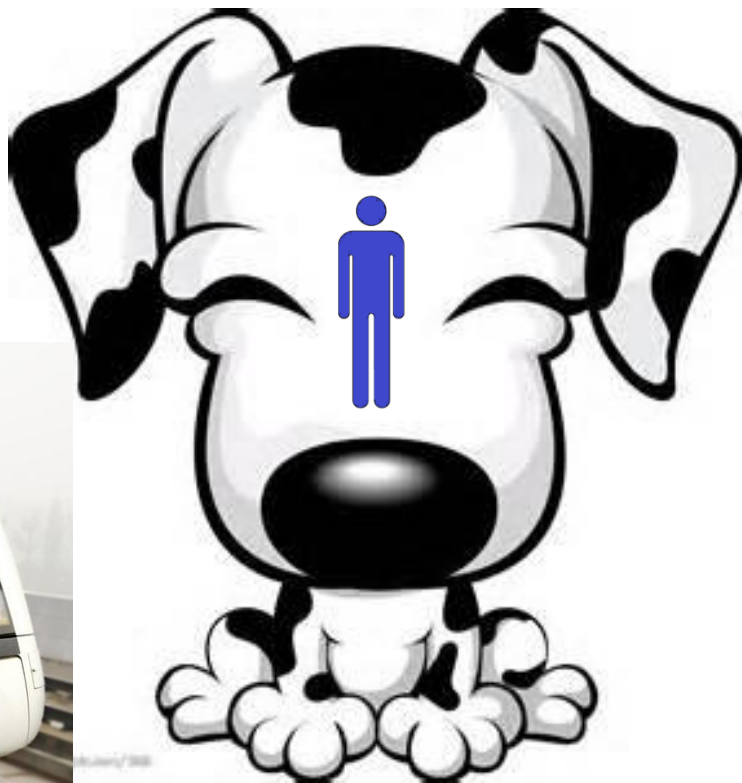
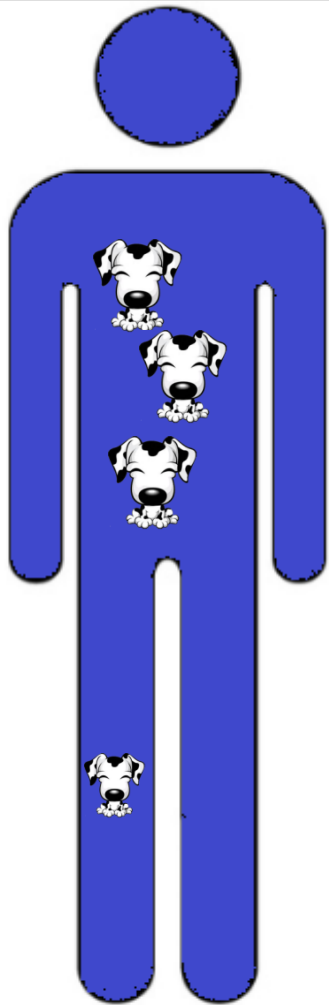
- 如果要写一个小区养狗管理程序，  
需要写一个“**业主**”类，还需要写一个“**狗**”类。
- 而狗是有“主人”的，主人当然是业主(假定狗只有一个主人，但一个业主可以有最多10条狗)

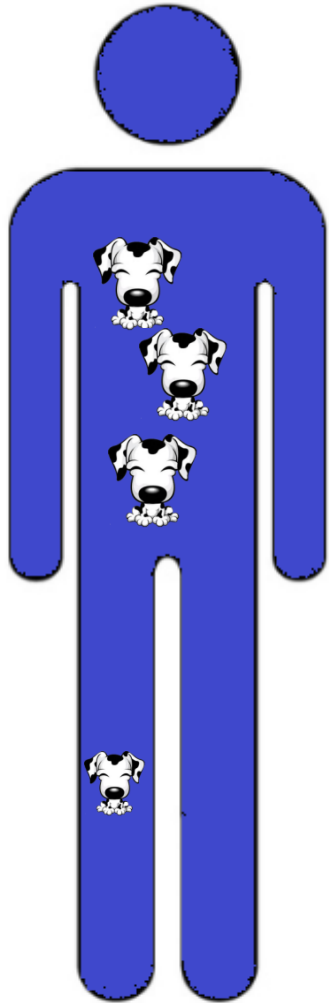




# 复合关系的使用

```
class CDog;  
class CMaster  
{  
    CDog dogs[10];  
};  
class CDog  
{  
    CMaster m;  
};
```





# 复合关系的使用

```
class CDog;  
class CMaster  
{  
    CDog dogs[10];  
};  
class CDog  
{  
    CMaster m;  
};
```



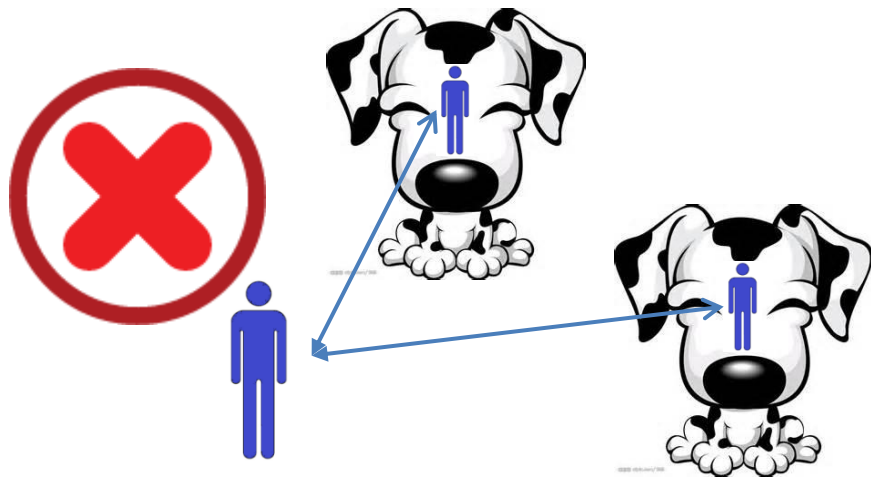
# 复合关系的使用

➤ 另一种写法:

为“狗”类设一个“业主”类的成员对象;

为“业主”类设一个“狗”类的对象指针数组。

```
class CDog;  
class CMaster {  
    CDog * dogs[10];  
};  
class CDog {  
    CMaster m;  
};
```



# 复合关系的使用

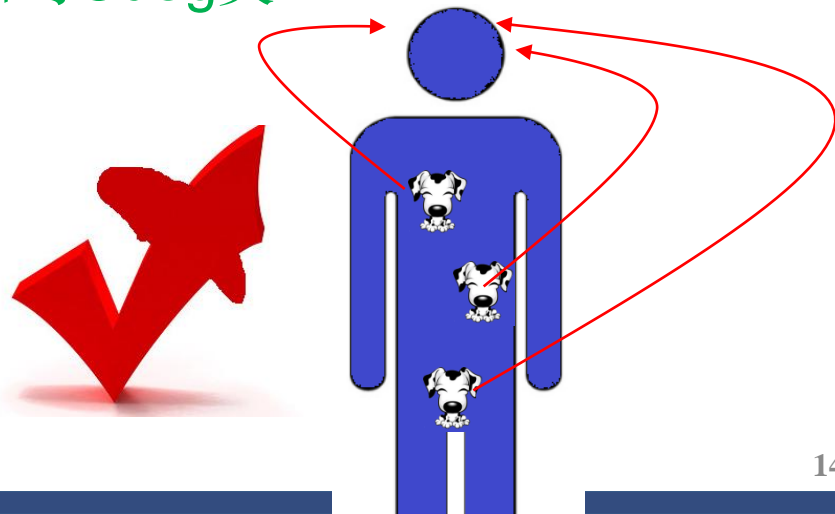
➤ 凑合的写法：

为“狗”类设一个“业主”类的对象指针；

为“业主”类设一个“狗”类的对象数组。

class CMaster; //CMaster必须提前声明，不能先  
//写CMaster类后写Cdog类

```
class CDog {  
    CMaster * pm;  
};  
class CMaster {  
    CDog dogs[10];  
};
```



# 复合关系的使用

➤ 正确的写法:

为“狗”类设一个“业主”类的对象指针;

为“业主”类设一个“狗”类的对象指针数组。

```
class CMaster; //CMaster必须提前声明, 不能先  
               //写CMaster类后写Cdog类
```

```
class CDog {  
    CMaster * pm;  
};  
class CMaster {  
    CDog * dogs[10];  
};
```

