

任务9-2 面向对象三大特征

封装

```
class Person:  
    def __init__(self, name, age):  
        self.name = name  
        self.age = age  
  
laowang = Person("老王", 30)  
laowang.age = 300  
print(laowang.age)
```

人的年龄可以随便
设置，显然不可行。

封装

为了保护类里面的属性，可以采用如下方式解决：

1. 把属性定义为私有属性，即在属性名的前面加上两个下划线；
2. 添加用于设置或获取属性值的两个方法供外界调用。

单继承

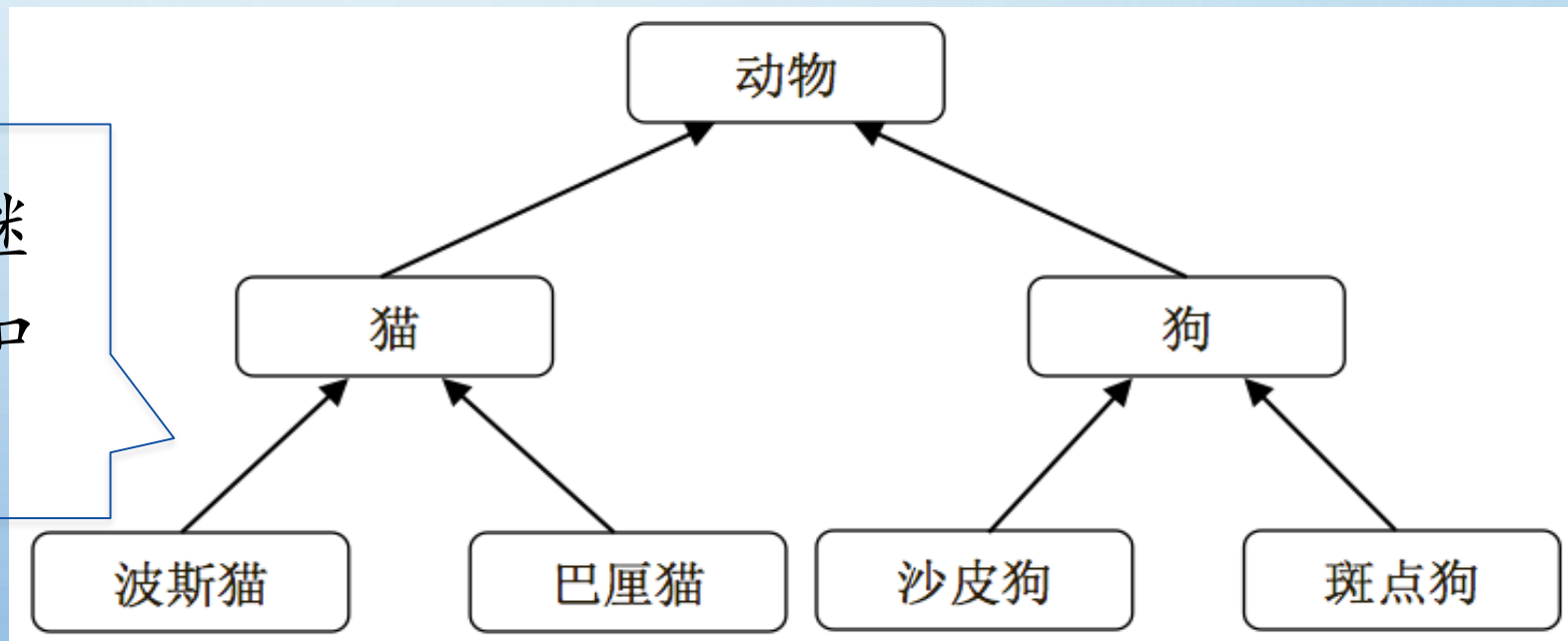


- 在现实生活中，继承一般指的是子女继承父辈的财产。
- 在程序中，继承描述的是事物之间的所属关系。

单继承

类的继承是指在一个现有类的基础上构建一个新的类，构建出来的新类被称作子类。

波斯猫和巴厘猫都继承自猫，而沙皮狗和斑点狗都继承狗。



单继承

Python程序中，继承使用如下语法格式标注：

```
class 子类名(父类名):
```

假设有一个类为A，A派生出来了子类B，示例如下：

```
class B(A):  
class A(object):
```

默认是继承自object的

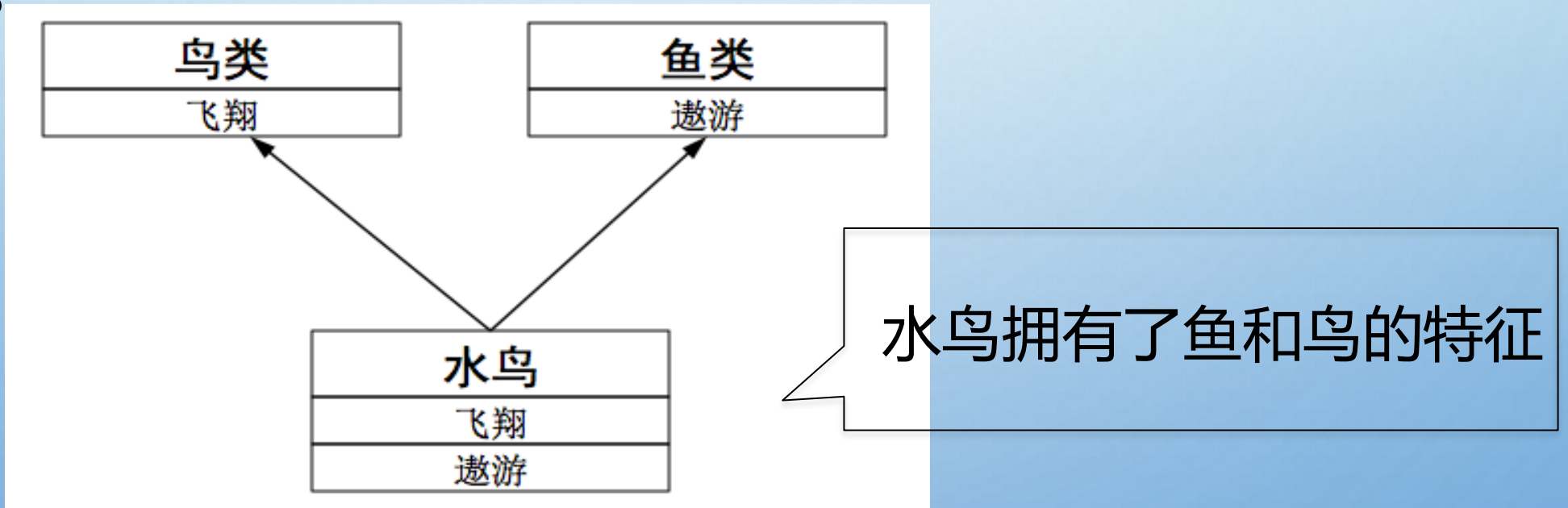
多继承

现实生活中，一个派生类往往会有多个基类。比如沙发床是沙发和床的功能的组合，这都是**多重继承**的体现。



多继承

Python支持多继承，多继承就是子类拥有多个父类，并且具有它们共同的特征，即子类继承了父类的方法和属性。



多继承

多继承可以看做是单继承的扩展，语法格式如下：

```
class 子类名(父类1, 父类2...):
```

如果子类继承的多个父类间是平行的关系，子类先继承的哪个类就会调用哪个类的方法。

重写父类方法和调用父类方法

- 在继承关系中，子类会自动拥有父类定义的方法，但是有时子类想要按照自己的方式实现方法，即对父类中继承来的方法进行重写，使得子类中的方法覆盖掉跟父类同名的方法。
- 需要注意的是，在子类中重写的方法要和父类被重写的方法具有相同的方法名和参数列表。

多态

不同国家的人，打招呼的方式是不同的。



多态

```
class A(object):  
    def test(self):  
        print("--A--test")
```

A类

```
class B(A):  
    def test(self):  
        print("--B--test")
```

B类

```
def func(temp):  
    temp.test()
```

```
a = A()  
b = B()  
func(a)  
func(b)
```

a、b的对象
两次调用func函数
结果不一样

多态

分析过程:

