

组合和派生

1

组合和派生

——组合

什么是组合

- 类被定义后，目标就是要把它当成一个模块来使用，并把这些对象嵌入到你的代码中去
- 组合就是让不同的类混合并加入到其它类中来增加功能和代码重用性
- 可以在一个大点的类中创建其它类的实例，实现一些其它属性和方法来增强对原来的类对象

组合应用

- 两个类明显不同
- 一个类是另一个类的组件

```
class Manufacture:
    def __init__(self, phone, email):
        self.phone = phone
        self.email = email

class BearToy:
    def __init__(self, size, color, phone, email):
        self.size = size
        self.color = color
        self.vendor = Manufacture(phone, email)
```

2

组合和派生

——子类和派生

创建子类

- 当类之间有显著的不同，并且较小的类是较大的类所需要的组件时组合表现得很好；但当设计“相同的类但有一些不同的功能”时，派生就是一个更加合理的选择了
- OOP 的更强大方面之一是能够使用一个已经定义好的类，扩展它或者对其进行修改，而不会影响系统中使用现存类的其它代码片段
- OOD（面向对象设计）允许类特征在子孙类或子类中进行继承

创建子类（续1）

- 创建子类只需要在圆括号中写明从哪个父类继承即可

```
class BearToy:
    def __init__(self, size, color):
        self.size = size
        self.color = color
    ... ..

class NewBearToy:
    pass
```

继承

- 继承描述了基类的属性如何“遗传”给派生类
- 子类可以继承它的基类的任何属性，不管是数据属性还是方法

```
class BearToy:
    def __init__(self, size, color):
        self.size = size
        self.color = color
    ... ..

class NewBearToy:
    pass

if __name__ == '__main__':
    tidy = NewBearToy('small', 'orange')
    tidy.speak()
```


通过继承覆盖方法

- 如果子类中有和父类同名的方法，父类方法将被覆盖
- 如果需要访问父类的方法，则要调用一个未绑定的父类方法，明确给出子类的实例

```
class BearToy:
    def __init__(self, size, color, phone, email):
        self.size = size
        self.color = color
        self.vendor = Manufacture(phone, email)
    ... ..

class NewBearToy(BearToy):
    def __init__(self, size, color, phone, email, date):
        super(NewBearToy, self).__init__(size, color, phone, email)
        self.date = date
```

多重继承

- python允许多重继承，即一个类可以是多个父类的子类，子类可以拥有所有父类的属性

```
>>> class A:
...     def foo(self):
...         print('foo method')
>>> class B:
...     def bar(self):
...         print('bar method')
>>> class C(A, B):
...     pass
>>> c = C()
>>> c.foo()
foo method
>>> c.bar()
bar method
```



更多精彩...



<http://bj.linux.tedu.cn/>
企业QQ：86198501