类的继承和组合：

例：程序class\_detail/employee.py

class Employee:  
 def \_\_init\_\_(self, name, salary=0):  
 self.name = name  
 self.salary = salary  
  
 def give\_raise(self, percent):  
 self.salary = self.salary + self.salary \* percent  
  
 def work(self):  
 print(self.name, 'does stuff')  
  
 def \_\_repr\_\_(self):  
 return '[Employee: name={0}, salary={1}]'.format(self.name, self.salary)  
  
  
class Chef(Employee):  
 def \_\_init\_\_(self, name):  
 super().\_\_init\_\_(name, 50000)  
  
 def work(self):  
 print(self.name, "makes food")  
  
  
class Server(Employee):  
 def \_\_init\_\_(self, name):  
 super().\_\_init\_\_(name, 40000)  
  
 def work(self):  
 print(self.name, 'interfaces with customer')  
  
  
class PizzaRobot(Chef):  
 def \_\_init\_\_(self, name):  
 super().\_\_init\_\_(name)  
  
 def work(self):  
 print(self.name, "makes pizza")  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 bob = PizzaRobot('Bob')  
 print(bob)  
 bob.work()  
 bob.give\_raise(0.2)  
 print(bob)  
 print()

输出为：

[Employee: name=Bob, salary=50000]

Bob makes pizza

[Employee: name=Bob, salary=60000.0]

例：程序class\_detail/pizzashop.py

from employee import PizzaRobot, Server  
  
  
class Customer:  
 def \_\_init\_\_(self, name):  
 self.name = name  
  
 def order(self, server):  
 print(self.name, "orders from", server)  
  
 def pay(self, server):  
 print(self.name, "pays for item to", server)  
  
  
class Oven:  
 def bake(self):  
 print('oven bakes')  
  
  
class PizzaShop:  
 def \_\_init\_\_(self):  
 self.server = Server('Pat') # 组合  
 self.chef = PizzaRobot('Bob')  
 self.oven = Oven()  
  
 def order(self, name):  
 customer = Customer(name)  
 customer.order(self.server)  
 self.chef.work()  
 self.oven.bake()  
 customer.pay(self.server)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 scene = PizzaShop()  
 scene.order('Homer')  
 print('...')  
 scene.order('Shaggy')

输出为：

Homer orders from [Employee: name=Pat, salary=40000]

Bob makes pizza

oven bakes

Homer pays for item to [Employee: name=Pat, salary=40000]

...

Shaggy orders from [Employee: name=Pat, salary=40000]

Bob makes pizza

oven bakes

Shaggy pays for item to [Employee: name=Pat, salary=40000]