

Python_设计模式：组合模式

```
#coding=utf8
import abc,collections,re,itertools

class Item(object):

    def __init__(self,name,*items,price = 0.0):
        self.children = []
        self.name = name
        self._price = price
        self.children.extend(items)

    def add(self,component):
        self.children.extend(component)

    def remove(self,component):
        if component in self.children:
            self.children.remove(component)
            return True
        else:
            raise ValueError("{} does not have component:{}".format(self.name,component.name))

    def printABC(self,indent=" "):
        if not self.composite:
            print("{}:{}".format(indent,self.name,self.price))
        else:
            print("{}:".format(indent,self.name))
            for child in self:
                child.printABC(indent+" ")

    @property
    def composite(self):
        return bool(self.children)

    @property
    def price(self):
        return sum(item.price for item in self) if self.children else self._price

    def __iter__(self):
        return iter(self.children)

if __name__=="__main__":
    pencil = Item('Pencil',price=5.0)
    box = Item('Box',price=9.0)
    ruler = Item('Ruler',price=2.0)
    eraser = Item('Eraser',price=1.0)
    pencilset = Item('Pencilset',pencil,ruler,eraser)
    BoxedPencilSet = Item('BoxedPencilSet',box,pencilset,pencil)
    BoxedPencilSet.remove(box)
    BoxedPencilSet.printABC()
```