

## OOP七大设计原则-开放封闭原则.ipynb

概述：  
简单地说，本原则的指导思想是，对扩展开放，对修改关闭。我们在做项目的时候，不要指望需求一开始就完全确定，这是不切实际的想法，既然需求一定是会变化的，那么我们就应该把需求变化作为一种常态来处理，将系统设计成相对易扩展、易维护的软件。在工作中的很多时候，程序员与产品经理之间的主要矛盾就是，经常变化的需求与不能变化的代码之间的矛盾。  
  
开闭原则的关键在于抽象化，将经常发生变化的部分封装成接口(Python中没有接口的概念)或抽象类，这样有新的需求或变化时，可以对现有代码进行扩展，以适应新情况；不至于整个推倒重做，只要从项目一开始就遵循开放封闭原则易扩展和易维护都不难做到，所以对于项目开发来说，软件框架本身需要具备灵活的可扩展性。  
  
尽量通过扩展软件实体的行为、而不是通过修改已有的代码来实现变化。  
  
适用场景：  
1)因为需求变化、软件升级或维护时，不去修改原有的代码，而是通过扩展实现热插拔的效果。  
2)将经常发生变化的部分封装为抽象，是实现开闭原则的重要手段。拒绝滥用抽象，只将经常变化的部分进行抽象。  
  
场景1：  
以书店销售书籍为例，原来书店只按照原价销售小说，现在由于经济下滑，书店为了生存开始打折销售：所有50元以上的书8折销售，其他书籍9折销售。

遵循了开放封闭原则的代码实现方式如下：

```
In [2]: #coding:utf8
from abc import ABCMeta, abstractmethod

class Book(object):
    __metaclass__ = ABCMeta
    def __init__(self, name, price, author):
        self.name = name
        self.price = price
        self.author = author

    def getName(self):
        return self.name

    def getAuthor(self):
        return self.author

    @abstractmethod
    def getPrice(self):
        pass

    def get_book_info(self):
        return "Book name is %s Price is %s Author is %s" %(self.getName(), self.getPrice()/ 100.0, self.getAuthor())

class NormalBook(Book):
    def getPrice(self):
        return self.price

#需求变化后需要提供打折处理，根据开闭原则，增加一个子类cheapBook类覆写getPrice()，而不是直接在原NormalBook类上做修改。

class cheapBook(Book):
    def getPrice(self):
        if self.price > 5000:
            return self.price * 0.8
        else:
            return self.price * 0.9

class BookStore():
    def __init__(self, BookObj):
        self.book_list = []
        self.book_list.append(BookObj("西游记", 3000, "吴承恩"))
        self.book_list.append(BookObj("三国演义", 6000, "罗贯中"))
        self.book_list.append(BookObj("红楼梦", 8000, "曹雪芹"))

    def sell(self, book):
        print("Sell info {}".format(book.get_book_info()))

if __name__ == '__main__':
    book_store = BookStore(NormalBook)
    for book in book_store.book_list:
        book_store.sell(book)
    print('#' * 20)
    book_store = BookStore(cheapBook)
    for book in book_store.book_list:
        book_store.sell(book)
```

Sell info Book name is 西游记 Price is 30.0 Author is 吴承恩  
Sell info Book name is 三国演义 Price is 60.0 Author is 罗贯中  
Sell info Book name is 红楼梦 Price is 80.0 Author is 曹雪芹  
#####  
Sell info Book name is 西游记 Price is 27.0 Author is 吴承恩  
Sell info Book name is 三国演义 Price is 48.0 Author is 罗贯中  
Sell info Book name is 红楼梦 Price is 64.0 Author is 曹雪芹

对扩展开放、对修改关闭，并不意味着不作任何修改，需求的变化往往是底层模块的变更倒逼高层模块的逐步解耦，已达到灵活性、可扩展性、可维护性强的目标。  
  
注意事项：  
1.新需求的实现是通过新增加代码来完成的，不是通过修改现有代码完成的。  
2.只对应用程序中频繁变化的部分进行抽象。  
3.将软件实现中易变动的细节抽象成接口或抽象类，当需求变化时可以根据实际要求重新派生一个实现类进行扩展。

In [ ]: