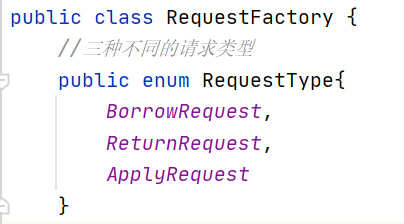
# 简单工厂模式

我们采用简单工厂模式来实现图书管理系统中处理各种请求的功能，包括借书请求、还书请求和增加新书请求。

## 简单工厂模式代码







## 设计思路

首先创建抽象产品类Request，其中包含请求类型字段type，公共方法getType()和抽象方法showRequest()，再创建三个产品类，分别为BorrowRequest、ReturnRequest和ApplyRequest，在具体产品类中各自实现showRequest()方法。创建工厂类RequestFactory，用一个枚举RequestType来存放请求的类型，createRequest()方法生成具体的请求对象，其参数为该请求的类型。要使用该工厂时，先创建一个接口RequestService（包含post()方法，参数为该请求的类型），再用具体类UserRequest实现该接口，在具体使用时新建一个UserRequest对象，便可通过post()方法按不同类型显示对应的请求消息。

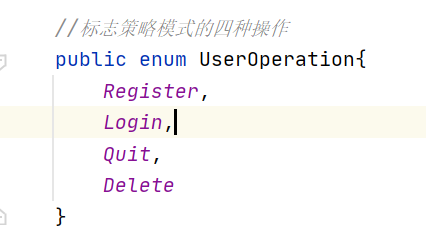
## 采用简单工厂模式的优势

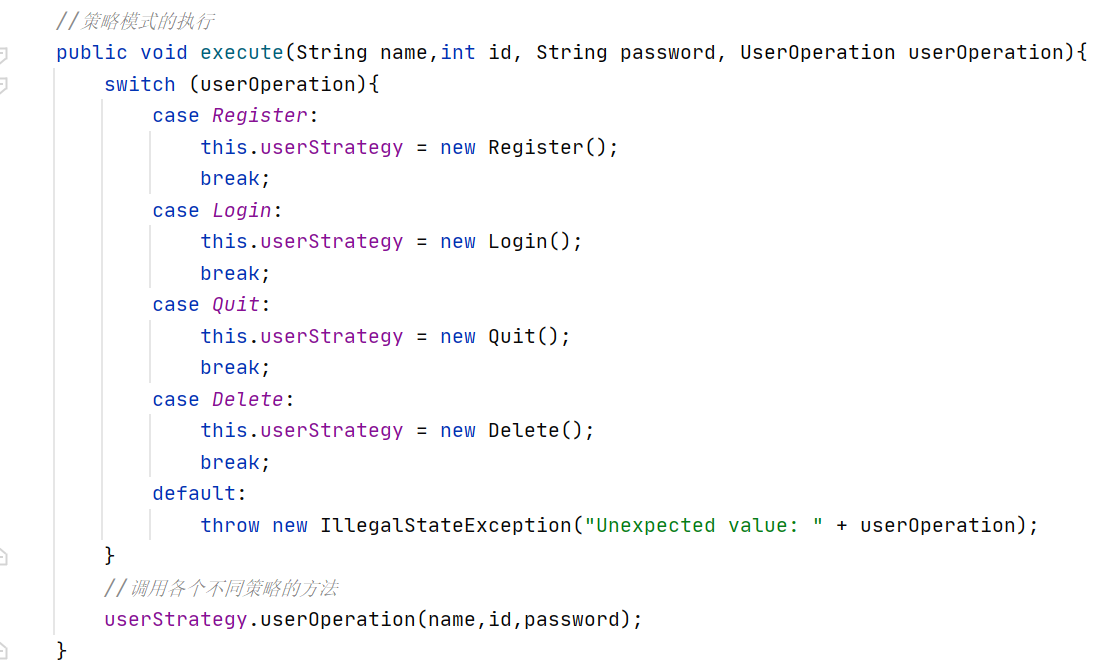
采用简单工厂模式实现请求的处理，客户端无需知道所创建的请求的具体类名，也无需承担创建请求对象的责任，只需指明需要的请求类型，通过工厂类生成请求。简单工厂模式将创建对象与使用对象的任务分割开来，明确了各自的职责和权利，简化了客户端代码。

# 策略模式

使用策略模式实现了账号的注册、登录、退出、删除功能。

## 策略模式代码





## 设计思路

通过枚举UserOperation存放对学生账户的四种操作，在执行时用了execute函数，通过userOperation的传入值判定是哪种策略并生成不同的对象，并用四个对象的不同方法实现不同的功能。

## 采用策略模式的优势

和工厂模式不同的是，策略模式将生成实例的使用策略放在策略类中配置后才提供调用方使用，更看重策略实施的过程，从而无需在实现类中拼接方法。因此，策略模式能够实现不同算法间的切换并具有良好的扩展性。