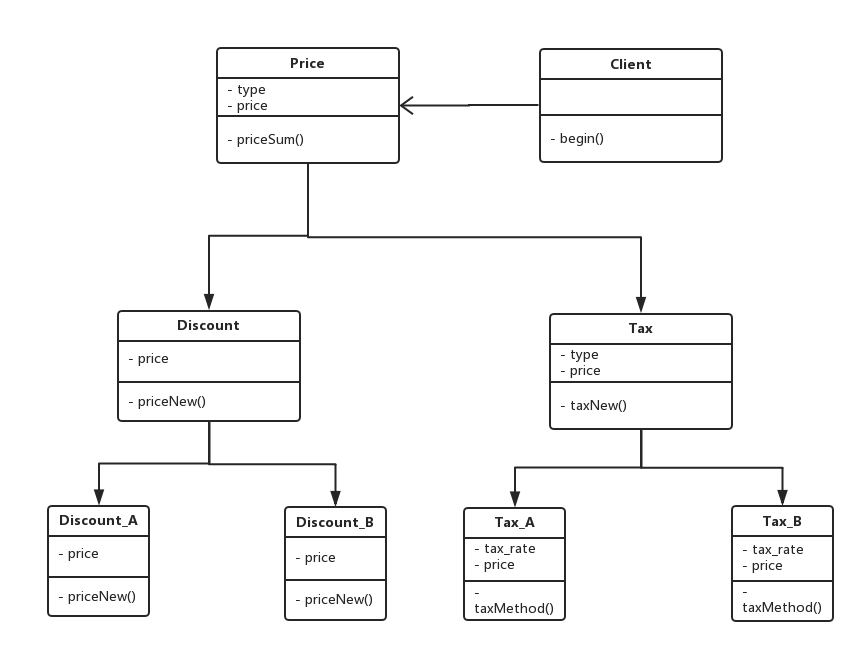
系统要求

3. 某软件企业为影音产品销售公司A开发一套在线销售系统，以提升服务的质量和效率。项目组经过讨论后决定采用面向对象方法开发该系统。在设计建模阶段需要满足以下设计要求：

A公司经常进行促销活动。根据不同的条件（如订单总额、商品数量、产品种类等），公司可以提供百分比折扣或现金减免等多种促销方式供提交订单的用户选择（如订单额度超过10万折扣5%，超过50万折扣8%，超过100万折扣10%， 超过200万减免15万现金）。实现每种促销活动的代码量很大，且会随促销策略不同经常修改。系统设计中需要考虑现有的促销和新的促销，而不用经常地重写控制器类代码。

该在线销售系统需要计算每个订单的税率，不同商品的税率及计算方式会有所区别。所以A公司决定在系统中直接调用不同商品供应商提供的税率计算类，但每个供应商的类提供了不同的调用方法（X供应商税率为5%；Y供应商税率7%；Z供应商税率7.5%）。系统设计中需要考虑如果公司更换了供应商，应该尽可能少地在系统中修改类。

UML类图



类图使用PrcessOn（https://www.processon.com）在线生成工具完成。

程序采用策略模式结合适配器模式进行设计。

主要包含八个类，Client、Price、Discount、Discount\_A、Discount\_B、Tax、Tax\_A、Tax\_B。各类之间关系见上图。

Client类用于对客户的抽象，包含一个方法：

一个方法为：

（1）begin()：引导用户进入在线销售报价界面；

Price类为价格计算界面系统的抽象，包含一个方法，两个属性。

两个属性为：

1. price：记录客户购买的初始总金额；
2. type：记录客户购买商品的种类；

一个方法为：

1. priceSum()：计算商品折扣后以及税后的价格；

Discount类为计算打折后价格功能的抽象，包含一个方法，一个属性。

一个属性为：

（1） price：记录客户的初始总报价；

一个方法为：

（1） priceNew()：依据价格范围采取不同的折扣策略；

Discount\_A类为一种折扣策略的抽象，包含一个方法，一个属性; Discount\_B与其类似。

一个属性为：

1. price： 记录客户的初始总报价；

一个方法为：

（1） priceNew()：一种折扣策略的实现；

Tax类为计算税后价格功能的抽象，包含一个方法，两个属性。

两个属性为：

（1）price：记录客户购买的初始总金额；

（2）type：记录客户购买商品的种类；

一个方法为：

1. taxNew()：规范不同税计算方法的接口，并依据商品类型计算税目；

Tax\_A类为计算税后价格功能的抽象，包含一个方法，两个属性； Tax\_B与其类似。

两个属性为：

（1）price：记录客户购买的初始总金额；

（2）tax\_rate：记录不同的税率；

一个方法为：

（1） taxMethod()：一种税目的计算方法；

系统实现

该版本的系统使用C++实现。工程包含十七个文件（main.cpp client.cpp client.h price.cpp price.h discount.cpp discount.h discount\_a.cpp discount\_a.h discount\_b.cpp discount\_b.h tax.cpp tax.h tax\_a.cpp tax\_a.h tax\_b.cpp tax\_b.h）。

其中：

1. main.cpp: 包含简单用户引导界面实现；
2. client.cpp & client.h: 包含对用户类的实现，进入商品报价计算界面；
3. price.cpp & price.h：包商品报价计算界面类的实现，，
4. discount.cpp & discount.h：包含选择折扣策略功能类的实现，选择一种折扣策略，并良好的应对折扣策略的添加改变；
5. discount\_a.cpp & discount\_a.h：包含折扣策略类的实现，实现一种折扣策略；
6. tax.cpp & tax.h：包含税目确定类的实现，规范不同税目计算方法的接口，选择相应的税目方法计算税目，并应对新的税目计算方法的接入；
7. tax\_a.cpp & tax\_a.h：包含税目计算类的实现，实现一种税目计算方法；

在线销售报价系统实现过程：在用户依据引导界面进入报价页面，用户输入要购买的产品类型以及初始总报价，系统基于报价范围计算相应的折扣策略，并基于产品类型提醒客户输入不同的税率，来接入不同的税率计算方法从而计算税目，最后结合税目以及折扣给出最终报价。

需要注意的实现细节如下：

1. 产品类型默认只有A、B两中，折扣策略、税目计算方法也只有A、B两种类型；
2. 对于折扣策略的设计采取的是策略模式，即对每一种折扣策略进行封装，由discount类选择使用具体的折扣，如果有新折扣策略加入，只需在discount类种添加相关实例即可；
3. 本系统对于折扣策略的选择默认基于报价的范围采取不同的折扣；
4. 对于税目计算的设计采取的是适配器模式，由于每一种税目的计算方法的接口可能有所不同，添加tax这一适配器类来对接口进行统一从而调用相关方法并应对新的计算方法；
5. 本系统对于税目计算的实现，在tax中要求客户输入不同的税率来模拟不同税率方法的接口统一过程，进而基于税率计算税目；

系统运行结果

