系统要求

利用策略模式改进汽车销售查询系统

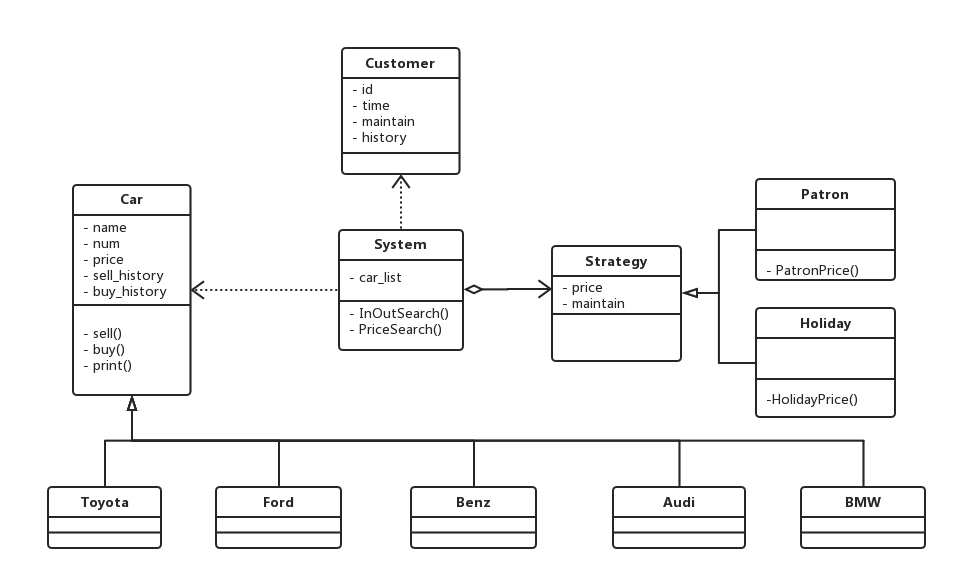
主要实现价格的查询

一般情况下不打折

老客户打8.5折

节日期间打8折，并送保养2次。

UML类图



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

该版本的汽车零售系统相对于v1.0版本增加了三个类，System、Customer以及Strategy。其中System替代了v1.0版本中的Search，对零售系统进行了抽象。

System类中包含一个属性：

1. car\_list：用来存储当前所有的品牌车，类型为对象数组；

System类中包含两个方法：

1. InOutSearch()： 用来实现v1.0版本中的汽车买进卖出的查询功能；
2. PriceSearch(): 用来实现汽车价格查询功能；

Customer类用于表示对顾客的抽象。

Customer类中包含四个属性：

1. id：用来标示顾客；
2. time: 用来标示顾客到达时间；
3. maintain：存储顾客当前的汽车保养次数；
4. history： 用来标示顾客是否为老顾客；

Strategy类为基于策略设计模式的策略类，包含两个子类，分别用来定义系统要求中针对老顾客和节假日两种中定价策略。

Strategy类包含两个属性：

1. price：标示当前品牌汽车的售价；
2. maintain：标示赠送的保养次数；

Patron类包含一个方法：

1. PatronPrice(): 用于对老顾客的定价策略实现；

Holiday类包含一个方法：

（1）HolidayPrice(): 用于对于节假日的定价策略实现；

系统实现

该版本的系统使用C++实现。工程包含九个文件（main.cpp car.cpp car.h customer.cpp customer.h strategy.cpp strategy.h system.cpp system.h）。

其中：

1. main.cpp: 包含简单用户引导界面实现；
2. system.cpp & system.h: 包含简单用户交互界面实现，实现汽车零售查询和价格查询两个功能；其中零售查询功能实现同v1.0；价格查询实现过程如下；
3. strategy.cpp & strategy.h：包含定价策略功能的实现，具体定价策略依照系统要求；
4. customer.cpp & customer.h：包含顾客类定义的实现，类信息见以上类图；
5. car.cpp & car.h：同v1.0版本的定义；

在价格查询实现过程：在用户输入用户ID后显示用户的个人信息以及当前所有品牌车对于该用户的报价，具体定价依据用户信息以及系统要求实现。

需要注意的实现细节如下：

1. 由于系统完成时间有限，用户信息为随机给出，随机判定当前用户是否为老顾客；
2. 用户到店时间定义为系统运行时间，到店时间为某一固定数值（当前系统为3）的整数倍定义为节假日；
3. 在我们的实现过程中两种定价策略可以累加；

系统运行结果

