作为一个简单的购物系统，至少需要具备以下功能（这些功能分布在不同级的菜单中）：

    （1）用户登录功能、用户账号密码修改功能，暂时不提供注册功能；  
    （2）用户成功登录后，需要具备客户信息的管理功能、购物结算功能以及一些抽奖活动等；  
    （3）客户信息管理功能下面又可以分出很多功能，比如说：查询、修改、增加等；  
    （4）购物结算功能下面又可以分出很多功能，比如说：商品选购、付款、账单等；  
    （5）抽奖活动下面又可以设计出多种的抽奖形式，从而进一步划分为许多新的功能模块。  
    （6）在一级菜单中要提供退出系统的功能，在二级菜单中要提供注销登录的功能，其他级菜单都要能够返回上一级菜单。

    上面的这些功能都是一些比较基本的功能，那么如果按照面向流程的思想来设计，就会划分很多功能模块，然后按照流程一步步走就行，但是现在我们采用面向对象的思想来设计，那么应该如何考虑设计框架呢？面向对象的主要思想就是将一些需求抽象为许多类，然后建立这些类之间的联系，通过不同类之间的协同合作，就可以实现所有的功能。所以，现在的主要任务就是如何合理地抽象出这些类，以及这些类要实现什么功能，类之间的联系又是什么？下面通过本次设计的结构对这一过程进行分析。

    （1）StartSMS类：用于系统的启动。我们的系统肯定需要一个启动类，这个类里面包含了main方法，用来启动这个系统，这个类是最顶层的，所以不能牵涉太多底层的细节实现，只需要实现一些顶层的基本流程就行，主要还是要调用底层其他类的一些方法来实现功能。

    （2）Data类：用来存放我们的所有数据信息，本次设计主要存放的是已经预存的一些可供购买的商品信息和已经注册的会员信息。为什么需要这个类呢？大家想一想，在面向对象的设计中，我们的数据比较多，肯定不能零散地到处定义、任意修改，这样会使得系统的聚合程度太低，容易出现很多错误，并且难以进行后期功能扩展和错误修改，所以我们要把用到的一些公有的数据进行归类，然后放在一个类中，并且在该类中提供对这些数据进行操作的方法。

    （3）Menu类：用于显示及处理各级菜单。既然我们设计的是一个购物系统，那么即使再简单，也需要一个基本的菜单，用来和用户进行交互，由于菜单的比较多，并且各级菜单之间层层相连，所以我们需要对菜单进行统一管理，故而出现了菜单类。注意，这里的菜单只是一些顶层的菜单显示和基本的功能调用，具体底层的算法还是需要更加底层的类来实现的。

    （4）Manager类：用于存储用户的账户和密码。既然我们需要用户登录，那么肯定需要一个单独的类来管理用户的账户和密码，从而使系统的独立性更强一些。本次设计的用户只有一个账户和密码，只允许修改账户和密码，但是不允许注册。

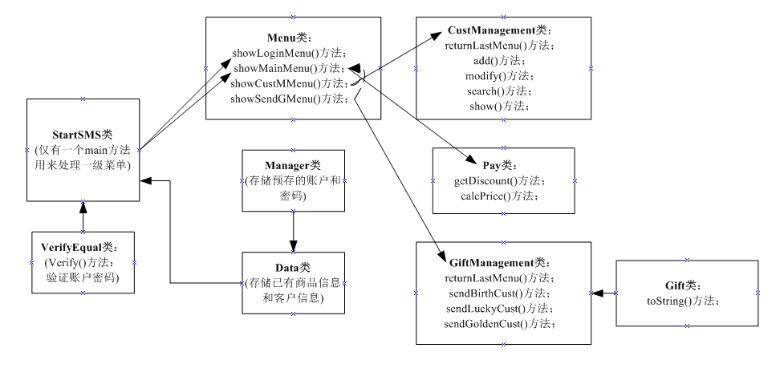
    （5）VerifyEqual类：用于验证登录信息。这个类相当于是把登录这项功能抽象成了一个类，这个实现其实并不是非常必要，但是为了使得系统功能划分更加清晰，所以设计了此类，用来对登录信息和已有的账户和密码进行校验，从而给出校验结果。

    （6）CustManagement类：用于客户信息的管理，该类实现了底层的一些功能，比如说查询、修改、增加等。当我们进入到客户信息管理这个菜单的时候，肯定需要对客户信息进行许多操作，为了方便管理这些操作，并考虑到后续的扩展性，这里把客户信息管理的所有功能都抽象出来，放在此类中，上一级菜单通过调用该类中的方法实现客户信息的管理。

    （7）Pay类：用于处理购物和结算操作。该类和上面的类存在的原理基本一致，当客户选择进行购物的时候，肯定要有很多操作，比如说买什么、多少钱、付款、找零等，这些功能比较零碎，所以我们对其进行集中管理，从而抽象出该类，对购物和结算的菜单选项的底层算法进行实现，上一级菜单通过调用该类的方法实现购物和结算功能，并且可以返回上一级菜单。

    （8）GiftManagement类：用于处理抽奖活动的相关功能。这个类和（6）、（7）中的类存在的理由基本一致，该类对抽奖活动进行了统一管理，上一级菜单只需要通过调用该类的方法就可以实现抽奖的功能。

    （9）Gift类：用于管理礼物。既然设计了抽奖环节，那么肯定需要礼物，那么我们会给出什么样的礼物呢，我们总不能每一样礼物都详细列出来吧，这样十分冗余，也很麻烦，所以我们干脆抽象出一个礼物类，把礼物的一些属性：礼物名称和价格等保存成该类的成员变量，然后就可以很方便的管理该类，需要什么样的礼物就直接新建一个礼物对象，然后对该对象的属性进行修改和管理即可，这样的实现类似于一个接口，但是又和接口完全不一样，功能差不多。



从上图中可以清晰地看出来各类之间的关系，大致关系和流程如下所述：

    （1）StartSMS类是启动类，内含main方法，这个类里面定义了VerifyEqual类和Data类的对象，用来存储数据和验证信息，同时Data类中包含了Manager类，用来存储预存的用户账号信息，然后在main方法中通过一定的逻辑，去调用Menu类中的showLoginMenu()方法，用于处理一级菜单---登录修改流程；

    （2）如果登录成功，就调用Menu类中的showMainMenu()方法，用于处理二级菜单---购物系统的主流程，如果登录失败3次，就直接退出系统；

    （3）在Menu类中的showMainMenu()方法中，通过选择不同的二级菜单选项，从而调用Menu类中的showCustMMenu()方法来处理客户信息管理流程或者调用Menu类中的showSendMenu()方法来处理抽奖活动流程，亦或者调用Pay类中的calcPrice()方法来处理购物结算流程；

    （4）如果选择了二级菜单中的客户信息管理选项，那么就会调用Menu类中的showCustMMenu()方法，这个方法会调用CustManagement类中的各种方法，用以处理客户信息管理的不同操作；

    （5）如果选择了二级菜单中的购物结算选项，那么就会调用Pay类中的calcPrice()方法，从而处理购物结算的流程，注意在Pay类中的getDiscount()方法是用来根据客户会员信息来计算打折率的；

    （6）如果选择了二级菜单中的真情回馈选项，即抽奖活动，那么就会调用Menu类中的showSendMenu()方法，这个方法会调用GiftManagement类中的各种方法，用以处理抽奖活动的不同操作；

    注意到在CustManagement类和GiftManagement类中都有一个returnLastMenu()方法，该方法是用来返回上一级菜单使用的。