**实验3 设计模式（单子、工厂、策略、组合、观察者模式）练习**

在上次作业基础上，设计一个贩卖各类书籍的电子商务网站的购物车系统。该系统包括的模块有：

模块1：图书目录的添加模块，实现对书本信息的增加。

书本包含的信息有：

* 书的isbn号、
* 书名、
* 单价、
* 书的类型（书的类型包括：1.非教材类计算机图书、2.教材类图书、3.连环画类图书、4.养生类图书、5.其他 共5种）、

:模块2：定价策略维护模块，实现对定价策略的增加、删除、修改、查询。

定价策略分两种：1）简单策略，包括：百分比折扣策略（每本折扣x%）、绝对值优惠策略（每本优惠x元）。

2）组合策略。在简单策略基础上构成的复杂策略。

简单定价策略包含的信息有：

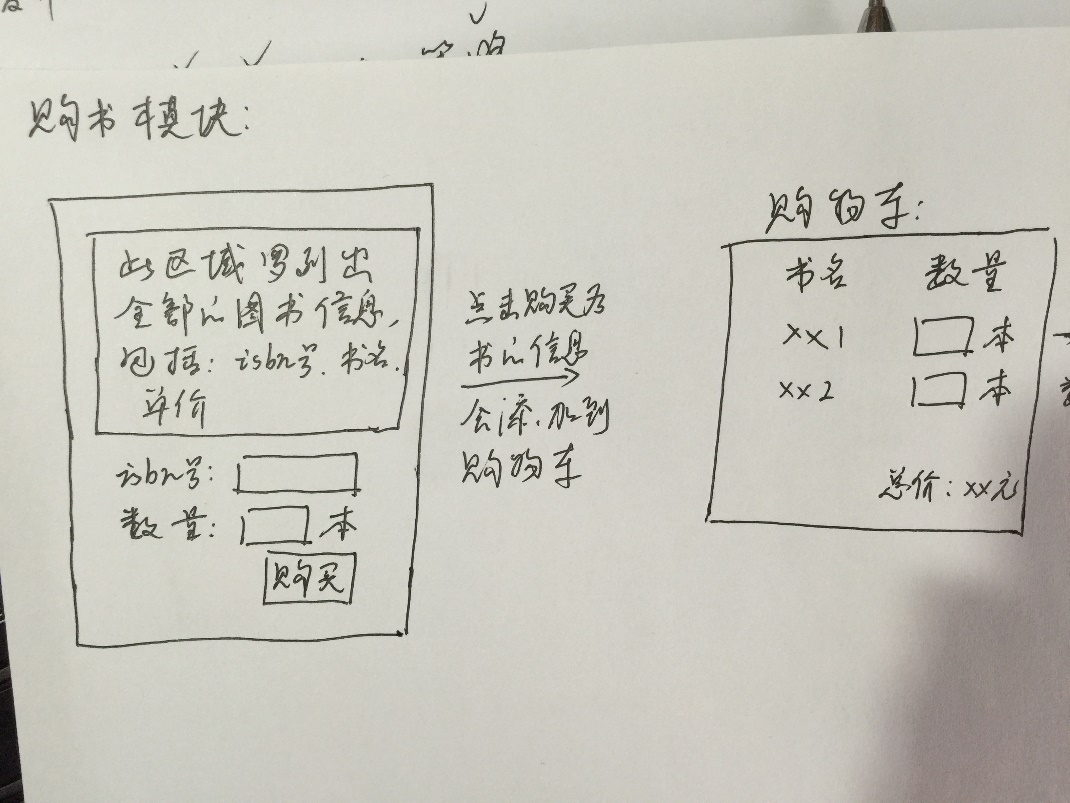
* 定价策略编号、
* 策略名称
* 定价策略类型（1：百分比折扣简单策略，2：绝对值优惠简单策略）
* 折扣百分比/每本优惠额度（单位：元）。

其中如果策略类型是1（百分比折扣策略），则第3项值为折扣百分比；否则为每本优惠额度。

组合定价策略包含的信息有：

* 定价策略编号、
* 策略名称
* 简单策略编号（ 字符串，多个简单策略编号之间用|分割）

模块3：购书模块。实现将书加入购物车，并计算优惠后总价。大概的界面实现如下：



上图左侧界面中，第一次输入要买

的书的isbn号和书的数量，点击购买，右侧的购物车窗口就会出现，显示书的名称、数量和总价（优惠后）。然后，在左侧界面中再次购买书的时候，右侧的购物车窗口就会进行更新，会显示当前全部购买的书以及总价。请使用观察者模式实现购物车窗口的刷新。

正常的购物车，书的数量可以进行修改。此处出于工作量考虑，假设右侧界面中书的数量不可修改。

实现下述场景：

1. 图书目录的维护模块中，录入下述书信息：

|  |  |  |  |
| --- | --- | --- | --- |
| isbn | 书名 | 价格 | 书类型 |
| 978-7-302-2 | 《UML与模式应用》 | 18 | 2 |
| 978-7-312-3 | 《Java与模式》 | 34 | 1 |
| 968-6-302-1 | 《HeadFirst 设计模式》 | 58 | 1 |
| 958-1-302-2 | 《爱丽丝历险记》 | 30 | 3 |
| 900-7-392-2 | 《煲汤大全》 | 20 | 4 |

（图书的类型包括：1.非教材类计算机图书、2.教材类图书、3.连环画类图书、4.养生类图书、5.其他 共5种）

2.定价优惠策略的维护模块中，录入下述策略：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 策略编号 | 策略名称 | 策略类型 | 适用图书类型 | 折扣百分比/每本优惠额度 |
| Discount001 | 绝对值优惠策略1 | 绝对值优惠策略 | 2 | 1 |
| Discount002 | 百分比折扣策略1 | 百分比折扣策略 | 3 | 7 |
| Discount003 | 百分比折扣策略2 | 百分比折扣策略 | 1 | 3 |
| Discount004 | 顾客最优策略1 | 组合策略 | 4 | 由Discount001和Discount003组成 |

定价策略类型（1：百分比折扣策略，2：绝对值优惠策略，3：组合策略）

图书的类型（1.非教材类计算机图书、2.教材类图书、3.连环画类图书、4.养生类图书、5.其他）

关于组合策略Discount004：假设购买了3本养生类图书《煲汤大全》，单价是20元/本，使用简单策略Discount001计算书总价为57元（每本优惠1元），使用简单策略Discount003计算书总价为58.2元（每本优惠0.6元），则最后总价取57元和58.2元中的较小值，即57元。

对于无优惠，如果策略工厂根据图书类型没有找到对应的策略，说明该书无优惠。

1. 购物模块中，实现：

购买了2本教材类图书《UML与模式应用》，单价是18元/本；购买了2本非教材类的计算机图书《Java与模式》，单价是34元/本；购买了1本非教材类的计算机图书《HeadFirst 设计模式》，单价是58元/本；购买了3本连环画类图书《爱丽丝历险记》，单价是30元/本；购买了3本养生类图书《煲汤大全》，单价是20元/本。

每购买一种书，会在购物车中显示当前已经购买的全部书及总价。

主要的类如下，在上次作业的类图基础上进行扩充。

类图说明：（蓝色字体的是本次新增的）

mainUI：主窗口

addBookUI：增加书本窗口

StrategiesUI：策略维护窗口

BuyUI：购买书的窗口

ShoppingCarUI：购物车窗口

Subject：主题接口（新增）

Observer：观察者接口（新增）

Controller：控制器，所有来自UI的请求会发给该控制器对象，由控制器对象再转发给其他领域对象，如Sale。

BookCatalog：书本信息目录，包括多个书本信息。

BookSpecification：书的规格说明，描述一本书的信息。

StrategyCatalog：定价策略目录。（新增）

Sale：代表依次销售。getTotal方法用于返回本次销售的优惠后总价。

SaleLineItem:代表一个销售项目。1次销售会对应多个销售项目。如顾客总共购买了2本《UML与模式应用》，3本《Java与模式》，则对应需要创建两个SaleLineItem对象，1个对应的是2本《UML与模式应用》的购买，1个对应的是3本《Java与模式》的购买。

BookSpecification（原先为ProductSpecification）:书的描述信息，包括书的isbn号、书名、单价和书的类型。书的类型包括：非教材类的计算机图书、教材类图书、连环画类图书、其他共4种。在本例中共有5种书（见后面描述），因此需要创建5个ProductSpecification对象。

PricingStrategyFactory：策略工厂，负责根据书的类型提供对应的策略类，策略类将根据具体的优惠策略计算折扣后1个销售项目（SaleLineItem）的总价。要求：该策略工厂要设计成单子。

IPricingStrategy：策略类的接口，用于向SaleLineItem屏蔽具体的策略类。

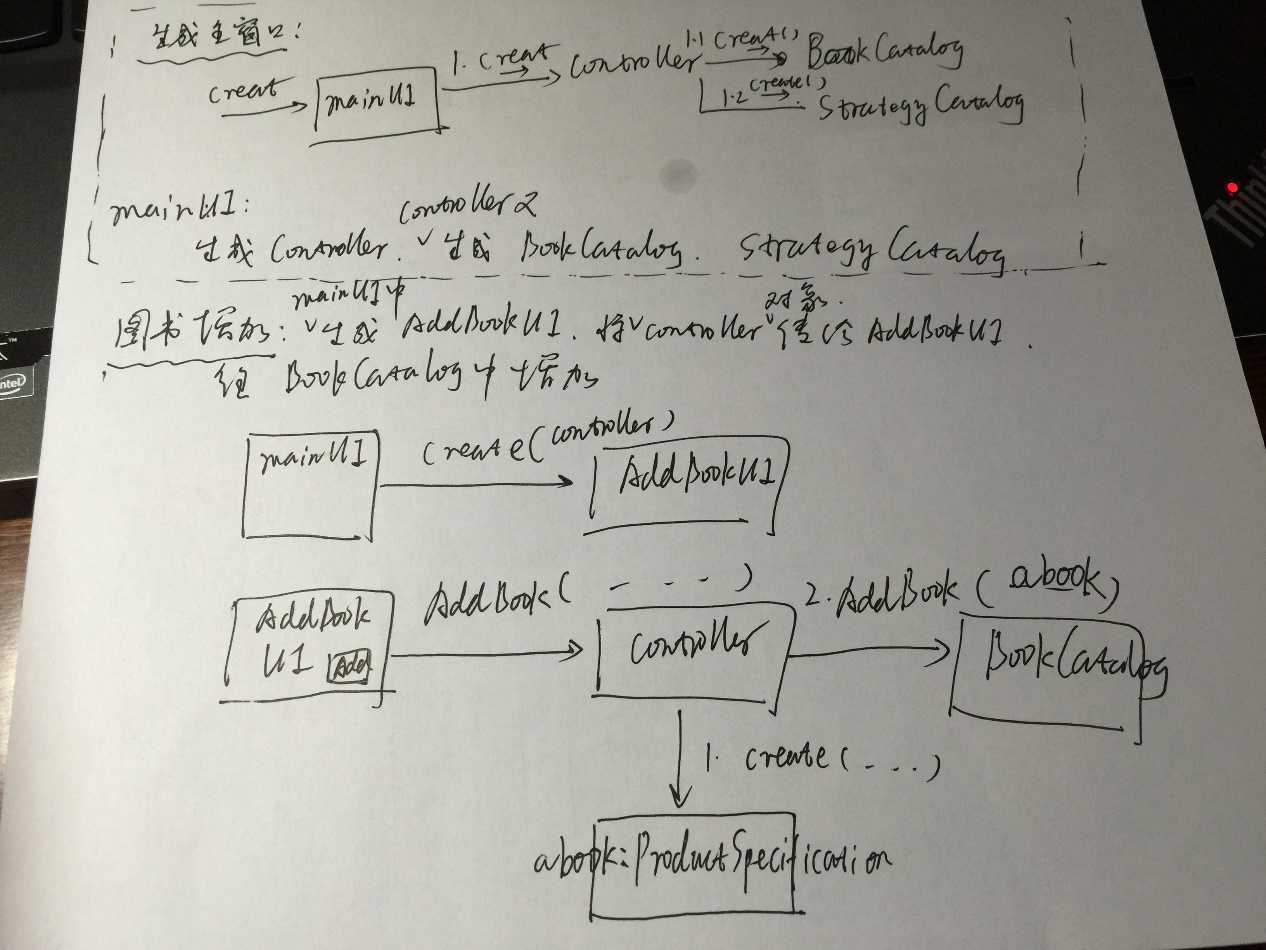
FlatRateStrategy：按每本优惠x元的策略。

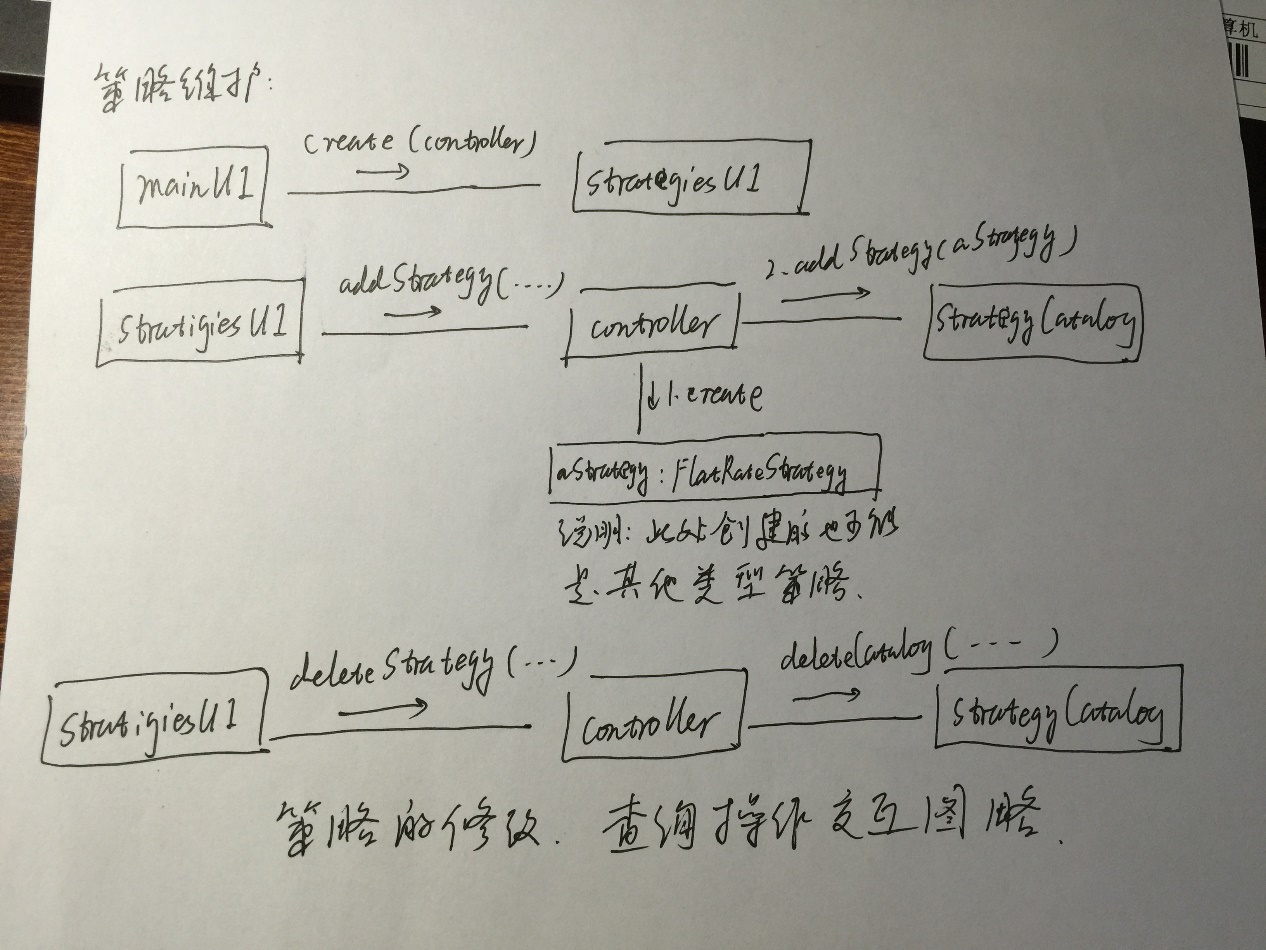
PercentageStrategy：按百分比进行优惠的策略。

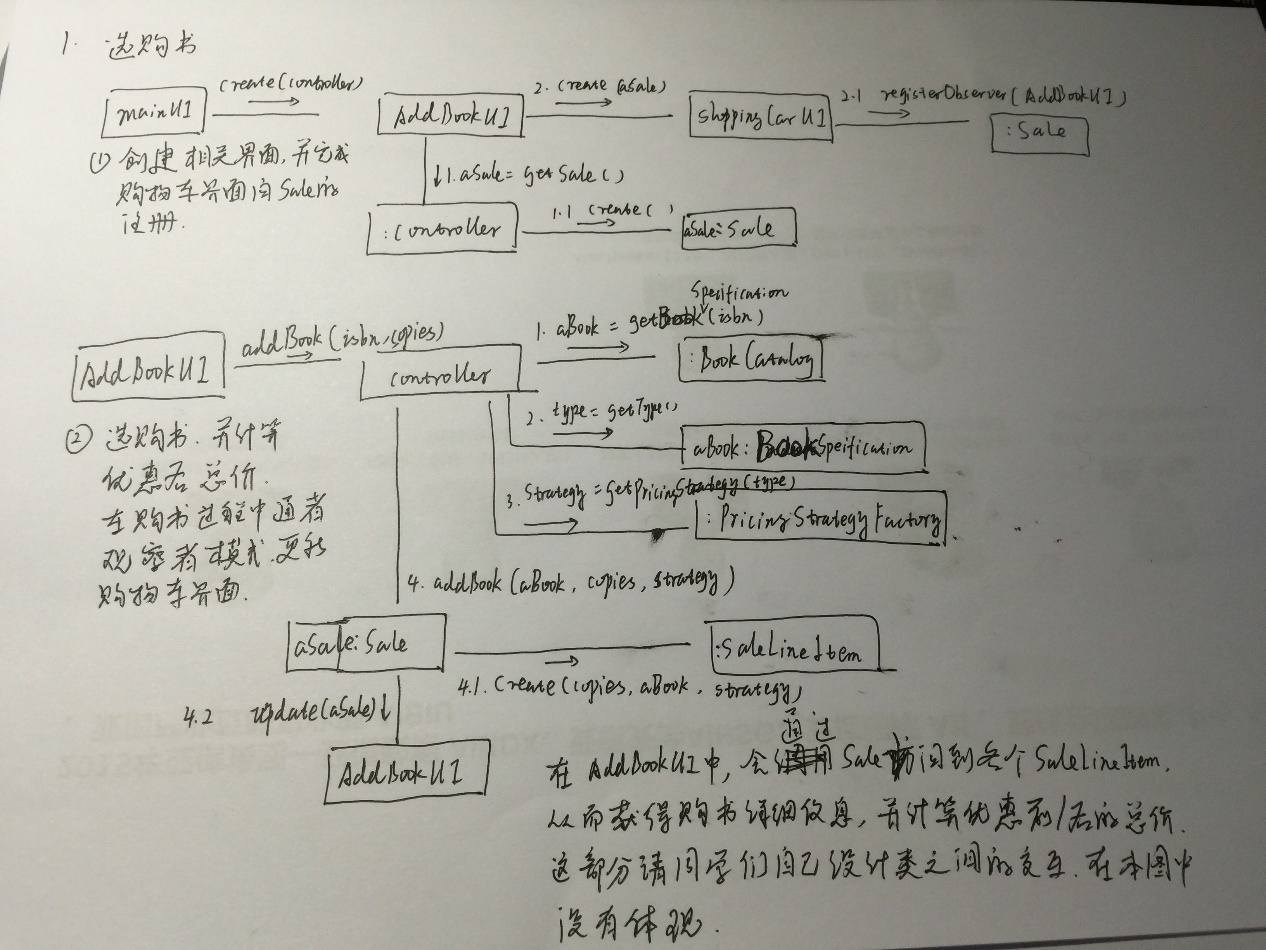
NoDiscountStrategy：不进行优惠的策略。

说明：上面这张类图中的属性和方法不一定全，大家根据需要可以自己补充属性、方法以及新的类。

类之间大概的交互场景是：（下面这三张图均有更新）







上图中：AddBookUI请改为BuyUI

作业提交时间：2016年1月13日

本次作业，除了提交代码，还提交一份文档，里面描述

1. 本次作业的类图（图中可以不给出类的属性和方法，可以手绘再拍成照片，也可以用工具绘制），对于你自己新增的类，要着重标出。
2. 每一个类的属性和方法进行说明。

|  |  |
| --- | --- |
| 类名：XXX | |
| 属性名：属性类型 | 属性说明 |
| 属性1：属性1类型 | 属性1说明 |
|  |  |
| 方法名 | 方法功能说明 |
| 返回值类型 方法1（参数列表） | 方法1说明 |
|  |  |

再次重申包名及文件名命名要求：（由于一些同学不注意命名要求，给助教的作业批改造成了很大的额外工作量。所以请大家严格按照下述要求进行命名，否则助教有可能找不到你的作业从而没有成绩！）

1. 包名的命名格式要求：homework+作业次数，如第二次作业包名应为:homewrok2（均为小写）
2. 一个类一个文件，类名和文件名保持一致，创建类的时候名字就应该是提交的命名。
3. Java文件命名：\_学号（8位）\_姓名\_第几次作业（1位）\_类名.java。eg.\_13211XXX\_wangxiao\_1\_Student.java。

注意：学号一律写成13开头，共8个数字。

1. 提交类文件，不要提交压缩包和工程。

5. 一定要自己运行正确之后再提交。