

实验 7 设计模式 2

实验报告

班级：----- 姓名：----- 学号：-----

本次实验包含了观察者模式、工厂模式、策略模式、组合模式、单子模式等设计模式以及图形化界面、类的继承与接口实现等 Java 知识，综合了本学期学到的 Java 和设计模式的知识，使我们锻炼并初步掌握 Java 面向对象的编程方法以及设计模式中封装变化、多用组合，少用继承和为交互对象之间的松耦合设计而努力等原则。

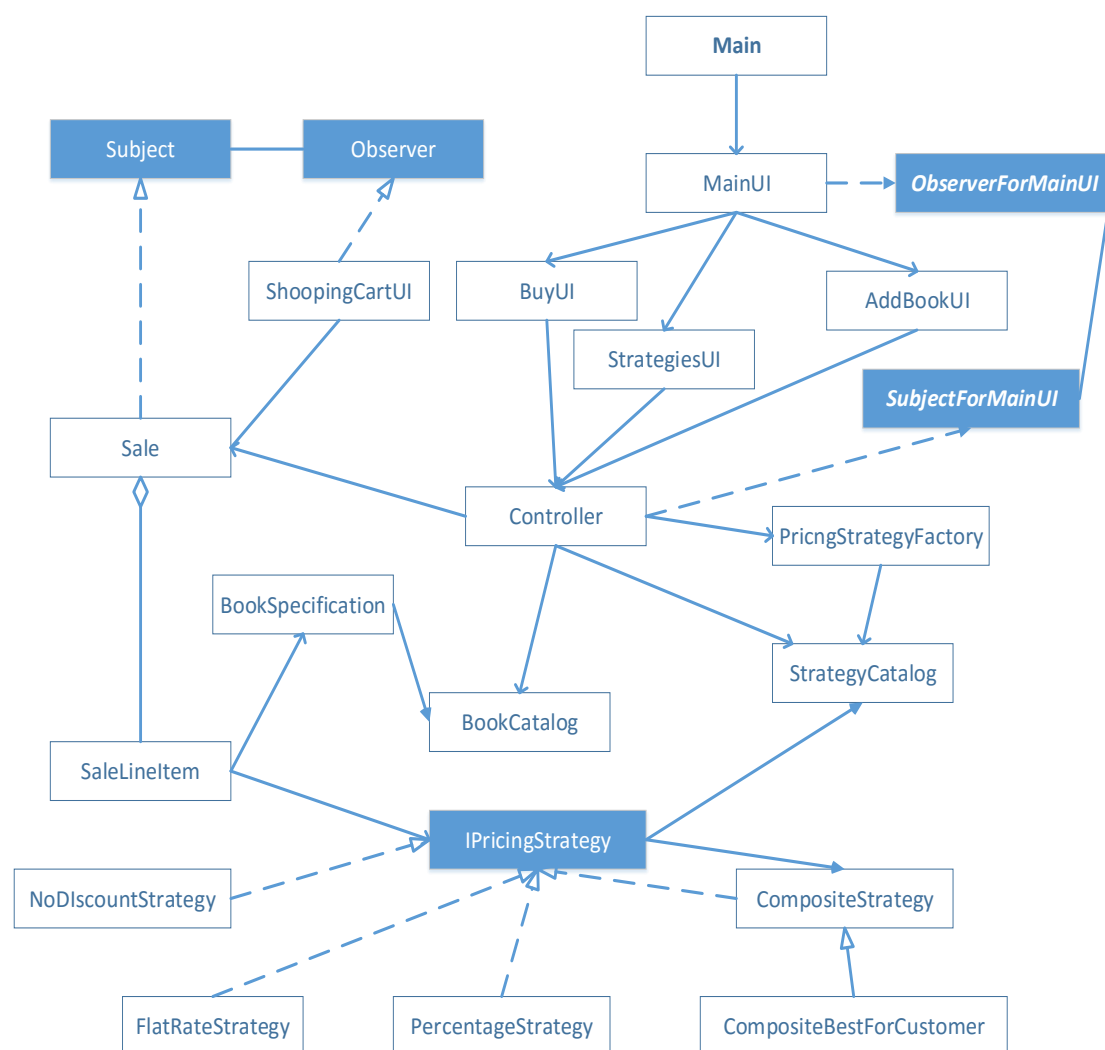


图 设计模式 2 实验的类图

表 设计模式 2 实验中各类的说明

public enum _13211415_---_7_BookType	
NONCOMPUTER	非教材的计算机类
TEXTBOOK	教材类
COMIC	连环画类
HEALTH	养生类
OTHER	其他类

public class _13211415_---_7_Main	
方法名	方法功能说明
public static void main (String[] args)	Main 函数，声明 MainUI 并调用
public class _13211415_---_7_MainUI extends JFrame implements ActionListener, _13211415_---_7_ObserverForMainUI	
属性名	属性说明
controller	声明控制器，并将控制器传给其他 UI，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
shoppingCartUI	购物车界面
class ModelTable extends DefaultTableModel	内部类，用于显示 JTable 中的数据
方法名	方法功能说明
public void actionPerformed(ActionEvent e)	当有事件触发时，按照事件源开始处理相应事件，例如点击“确定”提交购买等等
public void update()	更新 Jtable，在观察者模式中被 Subject 调用
public void initialize()	初始化，将预设的书籍和策略信息填入
public class _13211415_---_7_BuyUI extends JFrame implements ActionListener	
属性名	属性说明
bookSpecification	在主界面选中一本书进行购买，主界面将选中的书籍传入 BuyUI 的 bookSpecification 中
controller	控制器，提供各功能的调用
方法名	方法功能说明
public void actionPerformed(ActionEvent e)	当有事件触发时，按照事件源开始处理相应事件，例如点击“确定”提交购买等等
public class _13211415_---_7_AddBookUI extends JFrame implements ActionListener	
属性名	属性说明
controller	控制器
方法名	方法功能说明
public void actionPerformed(ActionEvent e)	当有事件触发时，按照事件源开始处理相应事件，例如点击“确定”提交购买等等
public class _13211415_---_7_StrategiesUI extends JFrame implements ActionListener	

属性名	属性说明
controller	控制器
addType	当点击“添加”按钮时确定用户想要添加的策略类型
selectedStrategyNumber	在点击“删除”和“修改”时确定用户选中的策略项
strategyType	当点击“修改”时确定用户修改项的策略类型
bookType	确定策略的图书类型
class ModelTable extends DefaultTableModel	内部类，用于显示 JTable 中的数据
方法名	方法功能说明
public BookType int2BookType(int number)	将 int 转成 BookType 枚举类型
public void actionPerformed(ActionEvent e)	当有事件触发时，按照事件源开始处理相应事件，例如点击“确定”提交购买等等
public class _13211415_---_7_ShoppingCartUI extends JFrame implements _13211415_---_7_Observer,ActionListener	
属性名	属性说明
subject	观察者模式中的 subject，用于注册
class ModelTable extends DefaultTableModel	内部类，用于显示 JTable 中的数据
方法名	方法功能说明
public void update (_13211415_---_7_Sale sale)	观察者模式中，subject 调用该函数更新 observer 的界面

public interface _13211415_---_7_Subject	
方法名	方法功能说明
void removeObserver(_13211415_---_7_Observer observer);	移去某个观察者
void registerObserver(_13211415_---_7_Observer observer);	注册某个观察者
void notifyObservers();	当数据变化时，更新观察者的界面
public interface _13211415_---_7_SubjectForMainUI	
方法名	方法功能说明
void removeObserver(_13211415_---_7_ObserverForUI observer);	移去某个观察者
void registerObserver(_13211415_---_7_ObserverForUI observer);	注册某个观察者
void notifyObservers();	当数据变化时，更新观察者的界面
public interface _13211415_---_7_Observer	
方法名	方法功能说明
void update(_13211415_---_7_Sale sale)	受 subject 的调用，更新购物车界面
public interface _13211415_---_7_ObserverForUI	

方法名	方法功能说明
void update()	受 subject 的调用，更新主界面的书籍信息

public class _13211415_---_7_Controller implements _13211415_---_7_SubjectForMainUI	
属性名	属性说明
bookCatalog;	书的信息目录
strategyCatalog	策略目录
pricingStrategyFactory	策略工厂
sale	销售记录
observerForMainUIs	观察者清单
方法名	方法功能说明
void addBook(_13211415_---_7_BookSpecification bookSpecification)	在书目录里添加一本书
boolean addStrategy (int strategyType, int strategyNumber, BookType bookType, String content)	添加一个策略
boolean deleteStrategy(int strategyNumber)	删除一个策略
boolean updateStrategy (int strategyType, int strategyNumber, BookType bookType, String content)	更新一个策略
boolean buyBook(int copies, String isbn)	购买一本书，增加一个销售记录（将会记录当前书籍状态和优惠策略）
int getStrategyType(int strategyNumber)	获得对应策略号的策略类型
13211415---_7_BookSpecification getBookSpecification(String isbn)	获得对应 isbn 号的书籍的信息
ArrayList <_13211415_---_7_IPricingStrategy> findCompositeContainSimple (int strategyNumber)	获取某复合策略里包含的简单策略
13211415---_7_StrategyCatalog getStrategyCatalog()	获取策略目录（用于更新界面）
13211415---_7_Sale getSale()	获取销售额（用于更新界面）
13211415---_7_BookCatalog getBookCatalog()	获取图书目录（用于更新界面）
void removeObserver(_13211415_---_7_ObserverForUI observer);	移去某个观察者（指主界面的书籍信息表格）
void registerObserver(_13211415_---_7_ObserverForUI observer);	注册某个观察者（指主界面的书籍信息表格）
void notifyObservers();	当数据变化时，更新观察者的界面

public class _13211415_---_7_PricingStrategyFactory	
属性名	属性说明

instance	单子模式中的实例
catalog	策略目录
方法名	方法功能说明
public static _13211415_---_7_PricingStrategyFactory getInstance()	返回工厂实例
public _13211415_---_7_IPricingStrategy getPricingStrategy(BookType bookType)	根据不同图书类型返回相应策略
public void setCatalog(_13211415_---_7_StrategyCatalog catalog)	设置策略目录
public interface _13211415_---_7_IPricingStrategy	
方法名	方法功能说明
getSubTotal	计算图书优惠后单价
getStrategyNumber	返回策略号
getStrategyType	返回策略类型
toString	返回策略的字符串形式
13211415---_7_NoDiscountStrategy implements _13211415_---_7_IPricingStrategy	
方法名	方法功能说明
所有方法均继承自接口	
public class _13211415_---_7_PercentageStrategy implements _13211415_---_7_IPricingStrategy	
属性名	属性说明
discountPercentage	百分比折扣 每本书的优惠价格=原价*(1-百分比折扣)
strategyType	策略类型
strategyNumber	策略号
方法名	方法功能说明
public void setStrategyNumber(int strategyNumber)	设置策略号
public void setDiscountPercentage(double discountPercentage)	设置百分比折扣
public class _13211415_---_7_FlatRateStrategy implements _13211415_---_7_IPricingStrategy	
属性名	属性说明
discountPerBook	每本书的直接折扣 书的优惠价=书的原价-直接折扣
strategyType	策略类型
strategyNumber	策略号
方法名	方法功能说明
public void setStrategyNumber(int strategyNumber)	设置策略号
public void set setDiscountPerBook(int discountPerBook)	设置每本书的折扣

public abstract class _13211415_---_7_CompositeStrategy implements _13211415_---_7_IPricingStrategy	
属性名	属性说明
discountPerBook	每本书的折扣
strategyType	策略类型
strategyNumber	策略号
ArrayList <_13211415_---_7_IPricingStrategy> strategies	包含的简单策略
方法名	方法功能说明
public void setStrategyNumber (int strategyNumber)	设置策略号
public void setDiscountPerBook (int discountPerBook)	设置每本书的折扣
void add (_13211415_---_7_IPricingStrategy strategy)	添加一个简单策略
void remove (_13211415_---_7_IPricingStrategy strategy)	移除一个简单策略
public boolean isEmpty()	判断是否包含简单策略
public class _13211415_---_7_CompositeBestForCustomer extends _13211415_---_7_CompositeStrategy	
方法名	方法功能说明
public double getSubTotal(_13211415_--- _7_SaleLineItem saleLineItem)	计算单本书的最优惠价格（继承自接口）
public class _13211415_---_7_StrategyCatalog	
属性名	属性说明
HashMap<BookType, _13211415_--- _7_IPricingStrategy> strategies	策略目录，哈希表类型
方法名	方法功能说明
boolean hasStrategy(int strategyNumber)	是否包含策略号对应的策略
int size()	返回策略目录大小
int getStrategyTypeFromNumber (int strategyNumber)	根据策略号获取对应策略的策略类型
BookType getKeyFromNumber (int strategyNumber)	根据策略号获取对应策略适用的图书类型 （即 Key）
ArrayList <_13211415_---_7_IPricingStrategy> findCompsiteContainSimple (int strategyNumber)	返回含有指定策略号简单策略的所有复合 策略
HashMap<BookType, _13211415_--- _7_IPricingStrategy> getStrategies()	获取策略目录(用于更新 StrategyUI)

public class _13211415_ --- _7_ Sale implements _13211415_ --- _7_ Subject	
属性名	属性说明
private ArrayList <_13211415_ --- _7_ SaleLineItem> items	销售清单里的销售项
private ArrayList <_13211415_ --- _7_ Observer> observers	观察者列表
方法名	方法功能说明
void addSaleLineItem (_13211415_ --- _7_ SaleLineItem item)	添加一个销售项
double getTotal()	计算销售清单（即购物车）里的总价格
public ArrayList<_13211415_ --- _7_ SaleLineItem> getItems	用于更新 ShoppingCartUI
void removeObserver (_13211415_ --- _7_ Observer observer);	移去某个观察者（即购物车界面）
void registerObserver (_13211415_ --- _7_ Observer observer);	注册某个观察者（即购物车界面）
void notifyObservers();	当数据变化时，更新观察者的界面
public class _13211415_ --- _7_ SaleLineItem	
属性名	属性说明
int copies	购买本数
prodSpec	所购买的书的信息
strategy	所购买的书适用的策略信息
方法名	方法功能说明
public double getSubTotal()	计算这个销售项对应的价格

public class _13211415_ --- _7_ BookSpecification	
属性名	属性说明
isbn	ISBN 号
price	价格
title	标题
type	书的类型
方法名	方法功能说明
各私有属性的 get/set 方法	
public class _13211415_ --- _7_ BookCatalog	
属性名	属性说明
ArrayList<_13211415_ 洪 志 龙 _7_ BookSpecification> books	书的目录信息
方法名	方法功能说明
void addBook(_13211415_ --- _7_ BookSpecification bookSpecification)	添加一本书的信息
13211415 - - - _7_ BookSpecification	获取对应 ISDN 号的书的信息

getBookSpecification (String isdn)	
ArrayList<_13211415_ _ _ _ _7_BookSpecification> getBooks ()	返回书的目录信息

public class _13211415_ _ _ _7_BookCatalog (_13211415_ _ _ _7_MainUI、_13211415_ _ _ _7_StrategiesUI 的内部类)	
属性名	属性说明
TableData	表格中的数据，Vector，每次插入数据时需要插入一个 ArrayList
TableTitle	表格的列标题，标明表格每一列的内容
方法名	方法功能说明
public void updateUI()	更新表格内容
public int getColumnCount()	定义表格的列数，重载了原函数
public Object getValueAt(int rowIndex, int columnIndex)	定义表格内容的格式，重载了原函数
public boolean isCellEditable(int rowIndex, int columnIndex)	使表格不可被编辑
此外根据不同需求定义了一些根据列数进行对应操作的函数，例如 public void addBook(_13211415_ _ _ _7_BookSpecification book)（在 _13211415_ _ _ _7_MainUI.java 里），输入一个书的信息，然后将该信息加入 JTable 中，受 updateUI 调用。又例如 public int getSelectedStrategyNumber(int rowIndex)（在 _13211415_ _ _ _7_StrategiesUI.java 里），根据列数返回对应列数的策略的策略号，用于选中策略进行修改、删除的函数中。	