## Java 设计模式2 文档

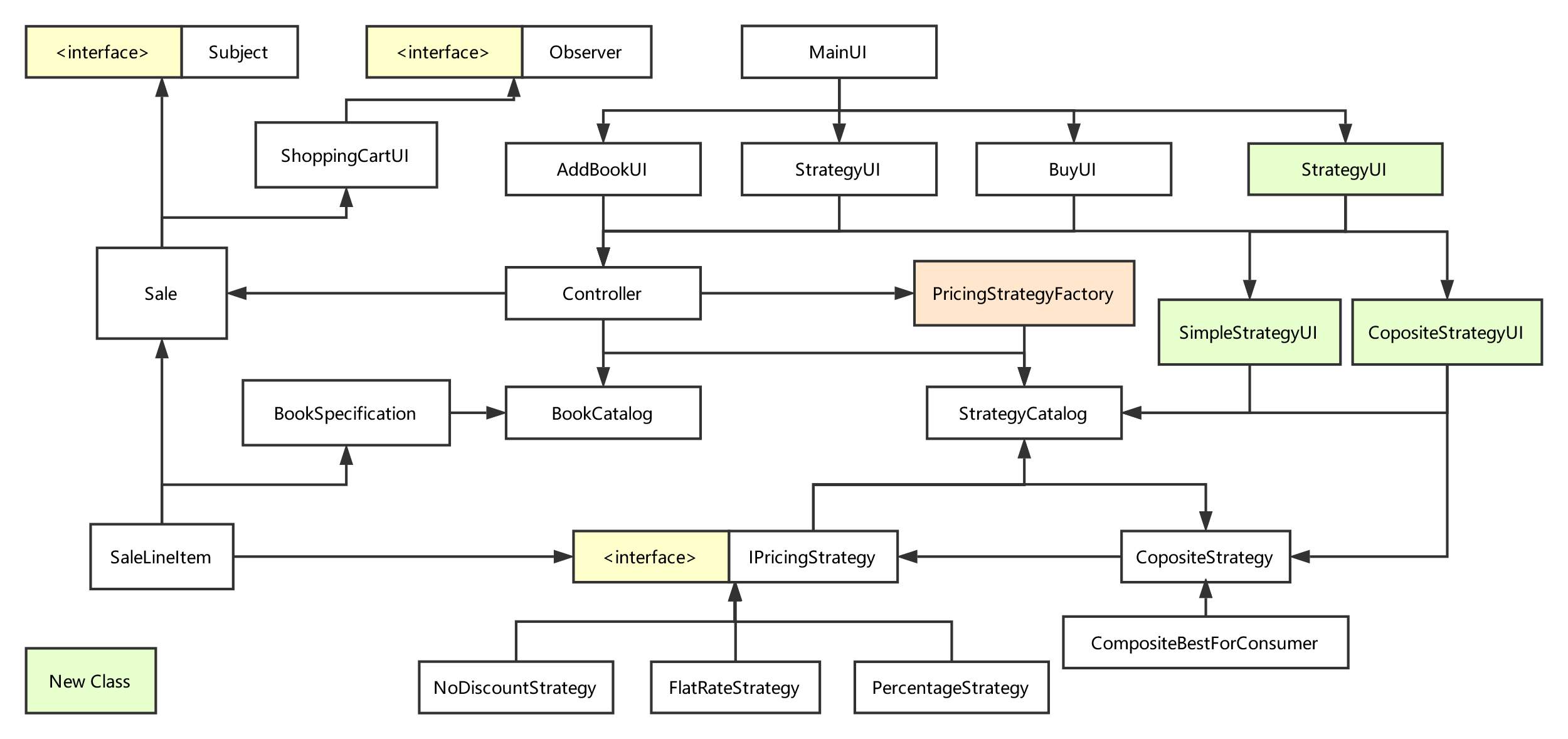
目录

[Java 设计模式2 文档 1](#_Toc502921181)

[1. 类图 1](#_Toc502921182)

[2. 每一个类的属性和方法说明 1](#_Toc502921183)

### 1. 类图



**注**

(1)绿色为新加类，均为界面相关，本身并没有新的功能实现。

(2)黄色为接口类

(3)PricingStrategyFactory设计为单子

### 2. 每一个类的属性和方法说明

|  |  |
| --- | --- |
| 类名：AddbookUI | |
| 属性名：属性类型 | 属性说明 |
| controller: Controller | Controller类对象 |
| isbnTextField: JTextField | ISBN号输入框 |
| nameTextField: JTextField | 书名输入框 |
| integerTextField: JTextField | 价格整数部分输入框 |
| decimalTextField: JTextField | 价格小数部分输入框 |
| typeTextField: JTextField | 种类输入框 |
| Submit: JButton | 提交按钮 |
| isbnLable: JLabel | ISBN提示字 |
| nameLable: JLabel | 书名提示字 |
| priceLable: JLabel | 价格提示字 |
| pointLable: JLabel | 小数点提示字 |
| typeLable: JLabel | 书类型提示字 |
| yuanLable: JLabel | 单位“元”提示字 |
| object: MainUI | 在MainUI对象的基础上建立新的界面 |
| 方法名 | 方法功能说明 |
| AddbookUI(MainUI \_object, Controller \_controller) | 构造函数，建立界面 |
| void actionPerformed(ActionEvent e) | 监听按钮并添加输入验证 |
| boolean isNumeric(String str) | 检测字符串是只含整数 |

|  |  |
| --- | --- |
| 类名：BookCatalog | |
| 属性名：属性类型 | 属性说明 |
| instance: static BookCatalog | 为了实现单例模式 |
| books: ArrayList <BookSpecification> | BookSpecification对象的ArrayList |
| 方法名 | 方法功能说明 |
| BookCatalog() | 构造函数 |
| static BookCatalog getlnstance() | 保证类仅有一实例，提供访问它的全局访问点 |
| void Addbook(BookSpecification a) | 添加书 |
| BookSpecification get(String isbn) | 根据ISBN返回BookSpecification对象 |
| ArrayList getbooks() | 获取私有对象book |

|  |  |
| --- | --- |
| 类名：BookSpecification | |
| 属性名：属性类型 | 属性说明 |
| isbn: String | ISBN号 |
| price: double | 价格 |
| name: String | 书名 |
| type: int | 书的种类 |
| 方法名 | 方法功能说明 |
| BookSpecification(String \_isbn, double \_price, String \_name, int \_type) | 构造函数 |
| String getisbn() | 获取私有变量isbn |
| String gettitle() | 获取私有变量name |
| double getprice() | 获取私有变量price |
| int gettype() | 获取私有变量type |

|  |  |
| --- | --- |
| 类名：BuyUI | |
| 属性名：属性类型 | 属性说明 |
| controller: Controller | Controller类对象 |
| books: ArrayList <BookSpecification> | BookSpecification对象的ArrayList |
| shoppingcart: Shoppingcart | Shoppingcart对象 |
| isbnTextField: JTextField | ISBN号输入框 |
| numberTextField: JTextField | 购买本数输入框 |
| plainLable: JLabel | 输出提示信息 |
| buy: JButton | 购买按钮 |
| isbn1Lable: JLabel | ISBN提示字 |
| isbn2Lable: JLabel | ISBN提示字 |
| nameLable: JLabel | 书名提示字 |
| priceLable: JLabel | 价格提示字 |
| benLable: JLabel | 单位“本”提示字 |
| object: MainUI | 在MainUI对象的基础上建立新的界面 |
| 方法名 | 方法功能说明 |
| BuyUI(MainUI \_object, Controller \_controller) | 构造函数，建立界面 |
| void actionPerformed(ActionEvent e) | 监听按钮并添加输入验证 |
| boolean isNumeric(String str) | 检测字符串是只含整数 |
| 类名：CompositeBestforCustomer | |
| 属性名：属性类型 | 属性说明 |
| 无 | |
| 方法名 | 方法功能说明 |
| CompositeBestforCustomer(String \_strategynumber, ArrayList a, String \_name, int \_type) | 构造函数 |
| int getStrategytype() | 获取价格策略 |
| double getSubTotal(SaleLineItem a) | 得到部分价格 |

|  |  |
| --- | --- |
| 类名：Controller | |
| 属性名：属性类型 | 属性说明 |
| sale: Sale | 私有Sale对象 |
| 方法名 | 方法功能说明 |
| void Addbook(String \_isbn, double \_price, String \_name, int \_type) | 添加书 |
| void AddSimpleStrategy(int strategykind, double value, int kind, String name) | 添加简单策略 |
| void updateSimpleStrategy(String a, double value, int kind,String name) | 更新简单策略 |
| AddCompositeStrategy(ArrayList \_strategies,String \_name,int \_type) | 添加复杂策略(顾客最优) |
| updateCompositeStrategy(String a,ArrayList strategies,String name,int type) | 更新复杂策略(顾客最优) |
| buyBook(String isbn, int copies) | 购买书，调用私有对象Sale |
| Sale getSale() | 获取私有对象sale |

|  |  |
| --- | --- |
| 类名：CopositeStrategy | |
| 属性名：属性类型 | 属性说明 |
| strategies: ArrayList <IPricingStrategy> | 策略接口对象ArrayList |
| strategynumber: String | 策略号 |
| type: int | 书的类型 |
| name: String | 书名 |
| 方法名 | 方法功能说明 |
| CopositeStrategy(String \_strategynumber,ArrayList \_strategies,String \_name, int \_type) | 构造函数 |
| String getname() | 获取私有对象name |
| String getStrategynumber() | 获取私有对象strategynumber |
| int gettype() | 获取私有对象type |
| ArrayList getstrategies() | 获取私有对象strategies |
| void add(IPricingStrategy a) | 添加策略 |
| abstract double getSubTotal(SaleLineItem a) | 得到部分价格 |
| abstract int getStrategytype() | 得到策略类型 |
| void update(ArrayList \_strategies,String \_name,int \_type) | 更新私有变量 |
| boolean isin(IPricingStrategy obj) | 是否存在该策略类型 |

|  |  |
| --- | --- |
| 类名：CopositeStrategyui | |
| 属性名：属性类型 | 属性说明 |
| controller: Controller | Controller类对象 |
| strategykind: int | 策略类型 |
| kind: int | 书的种类 |
| strategynumber: String | 策略号 |
| strategies: HashMap < String, IPricingStrategy > | 策略哈希表，以名称为键值 |
| RecordOps: HashMap < Integer, String > | 操作哈希表，以整数为键值 |
| reset: JButton | 重置按钮 |
| submit: JButton | 确定按钮 |
| strategyLable: JLabel | 选择策略标签 |
| nameTextField: JTextField | 书名输入框 |
| nameLable: JLabel | 策略名称标签 |
| typeLable: JLabel | 适用类型标签 |
| plainLable: JLabel | 提示信息标签 |
| type1: JRadioButton | 非教材类计算机图书单选按钮 |
| type2: JRadioButton | 教材类图书单选按钮 |
| type3: JRadioButton | 连环画类图书单选按钮 |
| type4: JRadioButton | 养生类图书单选按钮 |
| typegroup: ButtonGroup | 按纽ArrayList |
| checkboxes: ArrayList <JCheckBox> | 复选框 |
| object: strategiesui | 在strategiesui对象的基础上建立新的界面 |
| 方法名 | 方法功能说明 |
| CopositeStrategyui(strategiesui \_object,Controller \_controller, int \_strategykind, int \_kind, String \_strategynumber) | 构造函数 |
| void actionPerformed(ActionEvent e) | 按键监听 |

|  |  |
| --- | --- |
| 类名：FlatRateStrategy | |
| 属性名：属性类型 | 属性说明 |
| discountPerbook: double | 每本书降价 |
| type: int | 类型 |
| strategynumber: String | 策略类型 |
| name: String | 书名 |
| 方法名 | 方法功能说明 |
| FlatRateStrategy(String \_strategynumber, double \_discountPerbook, int \_type, String \_name) | 构造函数 |
| void update(double \_discountPerbook, int \_type, String \_name) | 更新信息 |
| int getStrategytype() | 获取策略类型(2) |
| int gettype() | 获取书的类型 |
| String getname() | 获取书名 |
| String getStrategynumber() | 获取策略类型 |
| double getnumber() | 获取数量 |
| double getSubTotal(SaleLineItem a) | 获取部分价格和 |
| boolean isin(IPricingStrategy a) | 是否存在该策略类型 |

|  |  |
| --- | --- |
| 类名：IPricingStrategy | |
| 属性名：属性类型 | 属性说明 |
| 无 | |
| 方法名 | 方法功能说明 |
| double getSubTotal(SaleLineItem a) | 获取部分价格和 |
| public String getStrategynumber() | 获取策略类型 |
| public int gettype() | 获取书的类型 |
| public void update(double a,int b,String \_name) | 更新信息 |
| public double getnumber() | 获取数量 |
| public String getname() | 获取书名 |
| public int getStrategytype() | 获取策略类型 |
| public void update(ArrayList \_strategies,String \_name,int \_type) | 更新信息 |
| public boolean isin(IPricingStrategy a) | 是否存在该策略类型 |

|  |  |
| --- | --- |
| 类名：MainUI | |
| 属性名：属性类型 | 属性说明 |
| welcome: JPanel | 容器 |
| bar: JMenuBar | 菜单 |
| AddbookMenu: JMenu | 菜单选项 |
| Menu: JMenu Strategies | 菜单选项 |
| BuyMenu: JMenu | 菜单选项 |
| AddbookItem: JMenuItem | 菜单二级选项 |
| StrategiesItem: JMenuItem | 菜单二级选项 |
| BuyItem: JMenuItem | 菜单二级选项 |
| Lable1: JLabel | 欢迎语 |
| controller: Controller | Controller类对象 |
| 方法名 | 方法功能说明 |
| MainUI() | 构造函数 |
| actionPerformed(ActionEvent e) | 事件监听 |
| void testbuybook(String \_isbn, int \_copies) | 买书 |

|  |  |
| --- | --- |
| 类名：NoDiscountStrategy | |
| 属性名：属性类型 | 属性说明 |
| strategynumber: String | 策略类型 |
| name: String | 书名 |
| 方法名 | 方法功能说明 |
| void update(double \_discountPerbook, int \_type, String \_name) | 更新信息 |
| int getStrategytype() | 获取策略类型(4) |
| int gettype() | 获取书的类型 |
| String getname() | 获取书名 |
| String getStrategynumber() | 获取策略类型 |
| double getnumber() | 获取数量 |
| double getSubTotal(SaleLineItem a) | 获取部分价格和 |
| boolean isin(IPricingStrategy a) | 是否存在该策略类型 |

|  |  |
| --- | --- |
| 类名：Observer | |
| 属性名：属性类型 | 属性说明 |
| 无 | |
| 方法名 | 方法功能说明 |
| void update(Sale a) | 更新信息 |

|  |  |
| --- | --- |
| 类名：PercentateStrategy | |
| 属性名：属性类型 | 属性说明 |
| discountPercentage: double | 降价百分比 |
| type: int | 类型 |
| strategynumber: String | 策略类型 |
| name: String | 书名 |
| 方法名 | 方法功能说明 |
| void update(double \_discountPerbook, int \_type, String \_name) | 更新信息 |
| int getStrategytype() | 获取策略类型(4) |
| int gettype() | 获取书的类型 |
| String getname() | 获取书名 |
| String getStrategynumber() | 获取策略类型 |
| double getnumber() | 获取数量 |
| double getSubTotal(SaleLineItem a) | 获取部分价格和 |
| boolean isin(IPricingStrategy a) | 是否存在该策略类型 |

|  |  |
| --- | --- |
| 类名：PricingStrategyFactory | |
| 属性名：属性类型 | 属性说明 |
| instance: PricingStrategyFactory | 实现单子模式 |
| catalog: strategyCatalog | Catalog对象 |
| 方法名 | 方法功能说明 |
| PricingStrategyFactory() | 构造函数私有，实现单子模式 |
| static PricingStrategyFactory getlnstance() | 实现单子模式 |
| IPricingStrategy getPricingStrategy(int bookType) | 获取定价策略 |
| void AddStrategy(IPricingStrategy a) | 添加价格策略 |
| void updateStrategy(String a, double value, int kind,String name) | 更新价格策略 |
| IPricingStrategy getStrategy(String strategynumber) | 获取价格策略 |
| HashMap getStrategies() | 获取策略的哈希表 |
| void deleteStrategy(String a) | 删除现有策略 |
| void updateCompositeStrategy(String a,ArrayList strategies,String name,int type) | 更新复杂策略(顾客最优) |
| String assignstrategynumber() | 格式化策略数字输出 |

|  |  |
| --- | --- |
| 类名：Sale | |
| 属性名：属性类型 | 属性说明 |
| items: ArrayList<SaleLineItem> | SaleLineItem对象ArrayList |
| observers: ArrayList<Observer> | Observer对象ArrayList |
| 方法名 | 方法功能说明 |
| public Sale() | 构造函数 |
| public void addBook(BookSpecification abook, int copies, IPricingStrategy strategy) | 添加书 |
| public double getTotal() | 获取价格和 |
| void removeObserver(Observer observer) | 移除Observer对象 |
| ArrayList getitems() | 获取私有对象items |
| void registerObserver(Observer observer) | 添加Observer对象 |
| void notifyObserver() | 更新Observer对象 |

|  |  |
| --- | --- |
| 类名：SaleLineItem | |
| 属性名：属性类型 | 属性说明 |
| copies: int | 买书份数 |
| prodSpec: BookSpecification | 书的信息 |
| strategy: IPricingStrategy | 定价策略 |
| 方法名 | 方法功能说明 |
| SaleLineItem(int \_copies, BookSpecification \_prodSpec, IPricingStrategy \_strategy) | 构造函数 |
| double getoriginalcost() | 获取原价 |
| int getcopies() | 获取私有对象份数 |
| String gettitle() | 获取书名 |
| double getSubTotal() | 计算价格部分和 |
| String getisbn() | 获取书的ISBN号码 |
| void addcopies(int inccopies) | 添加书的份数 |

|  |  |
| --- | --- |
| 类名：SaleTest | |
| 属性名：属性类型 | 属性说明 |
| 无 | |
| 方法名 | 方法功能说明 |
| static void main(String[] args) | 主函数 |

|  |  |
| --- | --- |
| 类名：Shoppingcart | |
| 属性名：属性类型 | 属性说明 |
| shopbook: ArrayList < SaleLineItem > | 买书的对象 |
| nameLable: JLabel | 书名标签 |
| copiesLable: JLabel | 数量标签 |
| plainLable: JLabel | 提示信息标签 |
| totalpriceLable: JLabel | 总价格标签 |
| panel: JPanel | 容器 |
| labels: ArrayList < JLabel > | 标签ArrayList |
| subject: Sale | Sale对象 |
| df: DecimalFormat | 为了格式化输出 |
| 方法名 | 方法功能说明 |
| public Shoppingcart(Sale \_subject) | 构造函数 |
| public void update(Sale \_subject) | 更新信息 |

|  |  |
| --- | --- |
| 类名：simpleStrategyui | |
| 属性名：属性类型 | 属性说明 |
| Controller controller: | Controller对象 |
| strategykind: int | 定价策略 |
| kind: int | 书的种类 |
| strategynumber: String | 定价策略编号 |
| nameTextField: JTextField | 书名输入框 |
| integerTextField: JTextField | 输入框 |
| decimalTextField: JTextField | 输入框 |
| reset: JButton | 重置按钮 |
| submit: JButton | 确定按钮 |
| nameLable: JLabel | 策略名称标签 |
| valueLable: JLabel | 标签 |
| pointLable: JLabel | 小数点标签 |
| typeLable: JLabel | 适用类型标签 |
| type1: JRadioButton | 非教材类计算机图书单选框 |
| zeroLable: JLabel | 0.标签 |
| yuanLable: JLabel | 单位“元”标签 |
| type2: JRadioButton | 教材类图书 |
| type3: JRadioButton | 连环画类图书单选框 |
| type4: JRadioButton | 养生类图书单选框 |
| typegroup: ButtonGroup | 按键组 |
| object: strategiesui | 在strategyui的基础上建立界面 |
| 方法名 | 方法功能说明 |
| simpleStrategyui(strategiesui \_object, Controller \_controller, int \_strategykind, int \_kind, String \_strategynumber) | 构造函数 |
| void actionPerformed(ActionEvent e) | 监听按钮并添加输入验证 |
| boolean isNumeric(String str) | 检测字符串是只含整数 |
| String getdecimal(double b) | 返回价格的格式化字符串 |

|  |  |
| --- | --- |
| 类名：strategiesui | |
| 属性名：属性类型 | 属性说明 |
| controller: Controller | 属性1说明 |
| strategies: HashMap < String, IPricingStrategy > | 策略哈希表 |
| RecordOps: HashMap < Integer, String > | 操作哈希表 |
| scrollPane: JScrollPane | 容器 |
| addPercentateStrategyButton: JButton | 添加百分比折扣策略按钮 |
| addFlatRateStrategyButton: JButton | 添加绝对值优惠策略按钮 |
| addCompositeStrategyButton: JButton | 添加组合策略按钮 |
| delButton: JButton | 删除所选策略按钮 |
| editButton: JButton | 编辑所选策略按钮 |
| optionLable: JLabel | 选项标签 |
| numberLable: JLabel | 策略编号标签 |
| nameLable: JLabel | 策略名称标签 |
| strategytypeLable: JLabel | 策略类型标签 |
| typeLable: JLabel | 适用图书类型标签 |
| valueLable: JLabel | 折扣百分比/每本优惠额度标签 |
| plainLable: JLabel | 提示信息 |
| radiobuttons: ArrayList <JRadioButton> | 按钮ArrayList |
| labels: ArrayList < JLabel > | 标签ArrayList |
| object: MainUI | 在MainUI的基础上建立界面 |
| 方法名 | 方法功能说明 |
| public strategiesui(MainUI \_object,Controller \_controller) | 构造函数 |
| void actionPerformed(ActionEvent e) | 监听按钮并添加输入验证 |
| public void update() | 更新界面 |
| public void paint() | 显示策略界面 |

|  |  |
| --- | --- |
| 类名：strategyCatalog | |
| 属性名：属性类型 | 属性说明 |
| strategies: HashMap < String, IPricingStrategy > | 策略哈希表 |
| 方法名 | 方法功能说明 |
| void AddStrategy(IPricingStrategy a) | 添加策略 |
| void deleteStrategy(String a) | 删除策略 |
| void updateStrategy(String a, double value, int kind, String name) | 更新策略 |
| void updateCompositeStrategy(String a, ArrayList \_strategies, String name, int \_type) | 更新策略(消费者最优) |
| IPricingStrategy getStrategy(String strategynumber) | 通过策略名称返回接口 |
| IPricingStrategy getStrategybytype(int bookType) | 通过书的类型返回接口 |
| HashMap getStrategies() | 获取私有变量策略哈希表 |

|  |  |
| --- | --- |
| 类名：Subject | |
| 属性名：属性类型 | 属性说明 |
| 无 | |
| 方法名 | 方法功能说明 |
| void removeObserver(Observer observer) | 删除Observer对象 |
| void registerObserver(Observer observer) | 添加Observer对象 |
| void notifyObserver() | 发出通知，Observer对象更新 |

所有类的属性和方法介绍完毕。