1. Account模块

1.1类的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| AccountController | 负责实现账户界面所需要的服务 |
| Account | 收付款以账户管理领域模型对象，拥有一次收款、付款以及账户等信息，可以帮助完成账户界面所需要的服务 |
| ReceiptList | 管理此次操作中的收款的转账列表，完成计算收款总额操作 |
| Receipt | 收款操作中的转账列表的一项 |
| PaymentList | 管理此次操作中的付款的条目清单，完成计算付款总额的操作 |
| Payment | 付款操作中的条目清单的一项 |

1.2 AccountController接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| AccountController.checkAccount() | 语法 | Public ResultMessage checkAccount() |
| 前置条件 | 已创建一个Account领域对象 |
| 后置条件 | 调用Account领域对象的checkAccount方法 |
| AccountController.addAccount | 语法 | Public ResultMessage addAccount(String name, int money) |
| 前置条件 | 已创建一个Account领域对象，输入符合规则 |
| 后置条件 | 调用Account领域对象addAccount的方法 |
| AccountController.delAccount | 语法 | Public ResultMessage delAccount(String name) |
| 前置条件 | 已创建一个Account领域对象，输入符合规则 |
| 后置条件 | 调用Account领域对象delAccount的方法 |
| AccountController.updateAccount | 语法 | Public ResultMessage updateAccount(String name1, String name2) |
| 前置条件 | 已创建一个Account领域对象，输入符合规则 |
| 后置条件 | 调用Account领域对象updateAccount的方法 |
| AccountController.searchAccount | 语法 | Public ResultMessage searchAccount(String word) |
| 前置条件 | 已创建一个Account领域对象 |
| 后置条件 | 调用Account领域对象searchAccount的方法 |
| AccountController.addReceipt | 语法 | Public ResultMessage addReceipt(String customer\_name, String clerk, String account\_name, int money, int all\_money, String ps) |
| 前置条件 | 已创建一个Account领域对象，输入符合规则 |
| 后置条件 | 调用Account领域对象addReceipt的方法 |
| AccountController.addPayment | 语法 | Public ResultMessage addPayment(String customer\_name, String clerk, String account\_name, int money, int all\_money, String ps) |
| 前置条件 | 已创建一个Account领域对象，输入符合规则 |
| 后置条件 | 调用Account领域对象addPayment的方法 |
| 需要的服务 | | |
| Account.checkAccount() | | 获得当前账户信息 |
| Account.addAccount (String name, int money) | | 添加账户 |
| Account.delAccount(String name) | | 删除账户 |
| Account.updateAccount(String name1, String name2) | | 更新账户 |
| Account.searchAccount(String word) | | 根据word查找账户 |
| Account.addReceipt(String customer\_name, String clerk, String account\_name, int money, int all\_money, String ps) | | 添加收款单 |
| Account.addPayment(String customer\_name, String clerk, String account\_name, int money, int all\_money, String ps) | | 添加付款单 |

1.3 Account接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| Account.checkAccount | 语法 | Public ResultMessage checkAccount() |
| 前置条件 | 管理人员为总经理或财务经理 |
| 后置条件 | 返回账户信息 |
| Account.addAccount | 语法 | Public ResultMessage addAccount(String name, int money) |
| 前置条件 | 需要添加账户时，管理人员为总经理或财务经理 |
| 后置条件 | 返回是否添加成功 |
| Account.delAccount | 语法 | Public ResultMessage delAccount(String name) |
| 前置条件 | 需要删除账户时，管理人员为总经理或财务经理 |
| 后置条件 | 返回是否删除账户成功 |
| Account.updateAccount | 语法 | Public ResultMessage updateAccount(String name1, String name2) |
| 前置条件 | 需要更新账户信息时，管理人员为总经理或财务经理 |
| 后置条件 | 完成账户信息修改，返回是否成功修改 |
| Account.searchAccount | 语法 | Public ResultMessage searchAccount(String word) |
| 前置条件 | 操作员是总经理或财务经理 |
| 后置条件 | 返回查找结果 |
| Account.addReceipt | 语法 | Public ResultMessage addReceipt(String customer\_name, String clerk, String account\_name, int money, int all\_money, String ps) |
| 前置条件 | 收到货款时，输入的money合法 |
| 后置条件 | 完成收款单，交由总经理审批 |
| Account.addPayment | 语法 | Public ResultMessage addPayment(String customer\_name, String clerk, String account\_name, int money, int all\_money, String ps) |
| 前置条件 | 完成付款时，输入的money合法 |
| 后置条件 | 完成付款单，交由总经理审批 |
| 需要的服务 | | |
| AccountDataService.getAccount() | | 获得当前账户信息 |
| AccountDataService.addAccount(AccountPO po) | | 添加账户AccountPO对象 |
| AccountDataService.delAccount(AccountPO po) | | 删除账户AccountPO对象 |
| AccountDataService.updateAccount(AccountPO po1, AccountPO po2) | | 更新账户AccountPO对象 |
| AccountDataService.findAccount(String name) | | 根据name查找账户对象 |
| AccountDataService.getAllAccount() | | 获得所有的账户对象 |
| AccountDataService.addReceipt(ReceiptPO po) | | 添加收款单ReceiptPO对象 |
| AccountDataService.addPayment(PaymentPO po) | | 添加付款单PaymentPO对象 |
| AccountDataService.getAllReceipt() | | 获得所有的收款单对象 |
| AccountDataService.getAllPayment() | | 获得所有的付款单对象 |
| AccountDataService.clear | | 清除所有对象 |

2. Commodity模块

2.1 类的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| CommodityController | 负责实现商品界面所需要的服务 |
| Commodity | 库存领域模型对象，拥有库存与商品的信息，可以完成库存管理与商品管理操作 |
| Stork | 临时管理库存中商品与分类，保存最基本的商品与分类信息 |
| Sort | 商品分类，临时保存最基本的商品信息 |
| Good | 商品，保存有商品的名称与型号 |

2.2 CommodityController接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| CommodityController.addCommodity | 语法 | Public ResultMessage addCommodity(String name, String type)  Public ResultMessage addCommodity(String name, String type, int in\_price, int out\_price) |
| 前置条件 | 已创建一个Commodity领域对象，输入符合规则 |
| 后置条件 | 调用Commodity领域对象的addCommodity方法 |
| CommodityController.delCommodity | 语法 | Public ResultMessage delCommodity(String name, String type) |
| 前置条件 | 已创建一个Commodity领域对象，输入符合规则 |
| 后置条件 | 调用Commodity领域对象的delCommodity方法 |
| CommodityController.updateCommodity | 语法 | Public ResultMessage updateCommodity(String name, String type, int in\_price, int out\_price) |
| 前置条件 | 已创建一个Commodity领域对象，输入符合规则 |
| 后置条件 | 调用Commodity领域对象的updateCommodity方法 |
| CommodityController.searchCommodity | 语法 | Public ResultMessage searchCommodity(String word) |
| 前置条件 | 已创建一个Commodity领域对象，输入符合规则 |
| 后置条件 | 调用Commodity领域对象的searchCommodity方法 |
| CommodityController.addSort | 语法 | Public ResultMessage addSort(String name) |
| 前置条件 | 已创建一个Commodity领域对象，输入符合规则 |
| 后置条件 | 调用Commodity领域对象的addSort方法 |
| CommodityController.delSort | 语法 | Public ResultMessage delSort(String name) |
| 前置条件 | 已创建一个Commodity领域对象，输入符合规则 |
| 后置条件 | 调用Commodity领域对象的delSort方法 |
| CommodityController.updateSort | 语法 | Public ResultMessage updateSort(String name1, String name2) |
| 前置条件 | 已创建一个Commodity领域对象，输入符合规则 |
| 后置条件 | 调用Commodity领域对象的updateSort方法 |
| CommodityController.Examine | 语法 | Public ResultMessage examine(String time1, String time2) |
| 前置条件 | 已创建一个Commodity领域对象，输入符合规则 |
| 后置条件 | 调用Commodity领域对象的Examine方法 |
| CommodityController.Inventory | 语法 | Public ResultMessage inventory() |
| 前置条件 | 已创建一个Commodity领域对象 |
| 后置条件 | 调用Commodity领域对象的Inventory方法 |
| CommodityController.addGift | 语法 | Public ResultMessage addGift(String name, String type) |
| 前置条件 | 已创建一个Commodity领域对象，输入符合规则 |
| 后置条件 | 调用Commodity领域对象的addGift方法 |
| CommodityController.delGift | 语法 | Public ResultMessage delGift(String name, String type) |
| 前置条件 | 已创建一个Commodity领域对象，输入符合规则 |
| 后置条件 | 调用Commodity领域对象的delGift方法 |
| CommodityController.patch | 语法 | Public ResultMessage patch(String name, String type, int number) |
| 前置条件 | 已创建一个Commodity领域对象，输入符合规则 |
| 后置条件 | 调用Commodity领域对象的patch方法 |
| CommodityController.warn | 语法 | Public ResultMessage warn(String name, String type, int number) |
| 前置条件 | 已创建一个Commodity领域对象，输入符合规则 |
| 后置条件 | 调用Commodity领域对象的warn方法 |
| 需要的服务 | | |
| 服务名 | 服务 | |
| Commodity.addCommodity(String name, String type, int in\_price, int out\_price) | 在分类中添加商品 | |
| Commodity.delCommodity(String name, String type) | 删除商品 | |
| Commodity.updateCommodity(String name, String type, int in\_price, int out\_price) | 更新商品信息 | |
| Commodity.searchCommodity(String word) | 根据关键词搜索商品 | |
| Commodity.addSort(String name) | 添加分类 | |
| Commodity.delSort(String name) | 删除分类 | |
| Commodity.updateSort(String name1, String name2) | 更新分类 | |
| Commodity. examine(String time1, String time2) | 查看库存 | |
| Commodity. inventory() | 盘点库存 | |
| Commodity.addGift(String name, String type) | 添加赠品对象 | |
| Commodity.delGift(String name, String type) | 删除赠品对象 | |
| Commodity.patch(String name, String type, int number) | 库存中商品报溢或报损 | |
| Commodity。warn(String name, String type, int number) | 设置商品的警戒线 | |

2.3 Commodity接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| Commodity.addCommodity | 语法 | Public ResultMessage addCommodity(String name, String type)  Public ResultMessage addCommodity(String name, String type, int in\_price, int out\_price) |
| 前置条件 | 输入的in\_price和out\_price为正整数 |
| 后置条件 | 添加商品，返回是否成功 |
| Commodity.delCommodity | 语法 | Public ResultMessage delCommodity(String name, String type) |
| 前置条件 | 无 |
| 后置条件 | 删除商品，返回是否成功 |
| Commodity.updateCommodity | 语法 | Public ResultMessage updateCommodity(String name, String type, int in\_price, int out\_price) |
| 前置条件 | 输入的in\_price和out\_price为正整数 |
| 后置条件 | 更新商品，返回是否更新成功 |
| Commodity.searchCommodity | 语法 | Public ResultMessage searchCommodity(String word) |
| 前置条件 | 无 |
| 后置条件 | 返回搜索结果 |
| Commodity.addSort | 语法 | Public ResultMessage addSort(String name) |
| 前置条件 | 无 |
| 后置条件 | 添加分类，返回是否添加成功 |
| Commodity.delSort | 语法 | Public ResultMessage delSort(String name) |
| 前置条件 | 无 |
| 后置条件 | 删除分类，返回是否删除成功 |
| Commodity.updateSort | 语法 | Public ResultMessage updateSort(String name1, String name2) |
| 前置条件 | 无 |
| 后置条件 | 修改分类，返回是否修改成功 |
| Commodity.Examine | 语法 | Public ResultMessage examine(String time1, String time2) |
| 前置条件 | 输入的时间格式正确 |
| 后置条件 | 返回查看结果 |
| Commodity.Inventory | 语法 | Public ResultMessage inventory() |
| 前置条件 | 无 |
| 后置条件 | 返回盘点结果 |
| Commodity.addGift | 语法 | Public ResultMessage addGift(String name, String type) |
| 前置条件 | 无 |
| 后置条件 | 增加赠品，返回是否成功 |
| Commodity.delGift | 语法 | Public ResultMessage delGift(String name, String type) |
| 前置条件 | 无 |
| 后置条件 | 删除赠品，返回是否成功 |
| Commodity.patch | 语法 | Public ResultMessage patch(String name, String type, int number) |
| 前置条件 | 输入的number为数字 |
| 后置条件 | 生成报溢报损单，返回是否成功 |
| Commodity.warn | 语法 | Public ResultMessage warn(String name, String type, int number) |
| 前置条件 | 输入的number为数字 |
| 后置条件 | 返回是否设置成功 |
| 需要的服务 | | |
| 服务名 | 服务 | |
| CommodityDataService.findGood(String name, String type) | 根据name和type查找商品对象 | |
| CommodityDataService.findName(String name) | 根据name查找商品对象 | |
| CommodityDataService.findType(String type) | 根据type查找商品对象 | |
| CommodityDataService.addGood(CommodityPO po1, SortPO po2) | 在分类中添加商品对象 | |
| CommodityDataService.delGood(CommodityPO po) | 删除商品对象 | |
| CommodityDataService.updateGood(CommodityPO po1, CommodityPO po2) | 更新商品对象信息 | |
| CommodityDataService.addSort(SortPO po) | 添加分类对象 | |
| CommodityDataService.delSort(SortPO po) | 删除分类对象 | |
| CommodityDataService.updateSort(SortPO po1, SortPO po2) | 更新分类对象 | |
| CommodityDataService.addGift(CommodityPO po) | 添加赠品对象 | |
| CommodityDataService.findGift(String name, String type) | 查找赠品对象 | |
| CommodityDataService.delGift(CommodityPO po) | 删除赠品对象 | |
| CommodityDataService.getAll() | 获得所有商品对象 | |
| CommodityDataService.clear() | 清空所有商品与分类对象 | |

3. Financial模块

3.1 类的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| FinancialController | 负责实现财务表单界面所需要的服务 |
| Financial | 财务账单领域模型对象，拥有经过加工后的账单信息，可以帮助完成财务表单界面所需要的服务 |

3.2 FinancialController接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| FinancialController.saleList | 语法 | Public ResultMessage saleList(String time1, String time2, String good\_name, String good\_type, String customer\_name, String clerk, int warehouse) |
| 前置条件 | 已创建一个Financial领域对象，输入符合规则 |
| 后置条件 | 调用Financial领域对象的saleList方法 |
| FinancialController.allBill | 语法 | Public ResultMessage allBill(String time1, String time2, String note\_type, String customer\_name, String clerk, int warehouse) |
| 前置条件 | 已创建一个Financial领域对象，输入符合规则 |
| 后置条件 | 调用Financial领域对象的allBill方法 |
| FinancialController.operatingCondition | 语法 | Public ResultMessage operatingCondition(String time1, String time2) |
| 前置条件 | 已创建一个Financial领域对象，输入符合规则 |
| 后置条件 | 调用Financial领域对象的operatingCondition方法 |
| 需要的服务 | | |
| Financial.saleList(String time1, String time2, String good\_name, String good\_type, String customer\_name, String clerk, int warehouse) | 根据条件查询销售明细 | |
| Financial.allBill(String time1, String time2, String note\_type, String customer\_name, String clerk, int warehouse) | 根据条件查询经营历程 | |
| Financial.operatingCondition(String time1, String time2) | 根据条件查询经营情况 | |

3.3 Financial接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| Financial.saleList | 语法 | Public ResultMessage saleList(String time1, String time2, String good\_name, String good\_type, String customer\_name, String clerk, int warehouse) |
| 前置条件 | 操作员是经过验证的财务人员 |
| 后置条件 | 查询销售明细，返回操作是否成功 |
| Financial.allBill | 语法 | Public ResultMessage allBill(String time1, String time2, String note\_type, String customer\_name, String clerk, int warehouse) |
| 前置条件 | 操作员是经过验证的财务人员 |
| 后置条件 | 查询单据，返回操作是否成功 |
| Financial.operatingCondition | 语法 | Public ResultMessage operatingCondition(String time1, String time2) |
| 前置条件 | 操作员是经过验证的财务人员 |
| 后置条件 | 查看经营情况，返回操作是否成功 |
| 需要的服务 | | |
| FinancialDataService.getSaleList() | 获得所有销售明细对象 | |
| FinancialDataService.getAllBill() | 获得所有经营历程对象 | |
| FinancialDataService.getOperatingCondition() | 获得所有经营情况对象 | |
| FinancialDataService.addSaleList(SaleListPO po) | 添加销售明细的单据对象 | |
| FinancialDataService.addAllBill(AllBillPO po) | 添加经营历程的单据对象 | |
| FinancialDataService.addOperatingCondition(OperatingConditionPO po) | 添加经营情况的单据对象 | |
| FinancialDataService.clear() | 清空所有对象 | |