1. Account模块

1.1 类的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| AccountDataController | 负责实现账户数据管理所需要的服务 |
| AccountData | 账户数据管理信息，完成数据库存的管理与存储 |
| AccountManagement | 账户管理领域模型对象，存储账户信息 |
| ReceiptManagement | 收款单领域模型对象，存储收款单信息 |
| PaymentManagement | 付款单领域模型对象，存储付款单信息 |

1.2AccountController接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| AccountDataController.getAccount() | 语法 | Public AccountPO getAccount() |
| 前置条件 | 已创建一个AccountData领域对象 |
| 后置条件 | 调用AccountData领域的getAccount方法 |
| AccountDataController.addAccount(AccountPO po) | 语法 | Public void addAccount(AccountPO po) |
| 前置条件 | 已创建一个AccountData领域对象，输入符合规则 |
| 后置条件 | 调用AccountData领域的addAccount方法 |
| AccountDataController.delAccount(AccountPO po) | 语法 | Public void delAccount(AccountPO po) |
| 前置条件 | 已创建一个AccountData领域对象，输入符合规则，对象中应收应付为0 |
| 后置条件 | 调用AccountData领域的delAccount方法 |
| AccountDataController.updateAccount(AccountPO po1, AccountPO po2) | 语法 | Public void updateAccount(AccountPO po1, AccountPO po2) |
| 前置条件 | 已创建一个AccountData领域对象，输入符合规则 |
| 后置条件 | 调用AccountData领域的updateAccount方法 |
| AccountDataController.findAccount(String name) | 语法 | Public AccountPO findAccount(String name) |
| 前置条件 | 已创建一个AccountData领域对象，输入符合规则 |
| 后置条件 | 调用AccountData领域的findAccount方法 |
| AccountDataController.getAllAccount() | 语法 | Public AccountPO[] getAllAccount() |
| 前置条件 | 已创建一个AccountData领域对象，输入符合规则 |
| 后置条件 | 调用AccountData领域的getAllAccount方法 |
| AccountDataController.addReceipt(ReceiptPO po) | 语法 | Public void addReceipt(ReceiptPO po) |
| 前置条件 | 收款单得到总经理审批通过，已创建一个AccountData领域对象，输入符合规则 |
| 后置条件 | 调用AccountData领域的addReceipt方法 |
| AccountDataController.addPayment(PaymentPO po) | 语法 | Public void addPayment(PaymentPO po) |
| 前置条件 | 付款单得到总经理审批通过，已创建一个AccountData领域对象，输入符合规则 |
| 后置条件 | 调用AccountData领域的addPayment方法 |
| AccountDataController.getAllReceipt() | 语法 | Public ReceiptPO[] getAllReceipt() |
| 前置条件 | 已创建一个AccountData领域对象，输入符合规则 |
| 后置条件 | 调用AccountData领域的getAllReceipt方法 |
| AccountDataController.getAllPayment() | 语法 | Public PaymentPO[] getAllPayment() |
| 前置条件 | 已创建一个AccountData领域对象，输入符合规则 |
| 后置条件 | 调用AccountData领域的getAllPayment方法 |
| AccountDataController.clear | 语法 | Public void clear() |
| 前置条件 | 进行期初建账之前，已创建一个AccountData领域对象，输入符合规则 |
| 后置条件 | 调用AccountData领域的clear方法 |
| 需要的服务 | | |
| AccountData.getAccount() | 获得当前账户的信息 | |
| AccountData.addAccount(AccountPO po) | 添加账户 | |
| AccountData.delAccount(AccountPO po) | 删除账户 | |
| AccountData.updateAccount(AccountPO po1, AccountPO po2) | 更新账户 | |
| AccountData.findAccount(String name) | 根据关键词查找账户 | |
| AccountData.getAllAccount() | 获得所有的账户对象 | |
| AccountData.addReceipt(ReceiptPO po) | 添加收款单 | |
| AccountData.addPayment(PaymentPO po) | 添加付款单 | |
| AccountData.getAllReceipt() | 获得所有的收款单 | |
| AccountData.getAllPayment() | 获得所有的付款单 | |
| AccountData.clear | 清除所有数据 | |

1.3 AccountData接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| AccountData.getAccount() | 语法 | Public AccountPO getAccount() |
| 前置条件 | 无 |
| 后置条件 | 返回当前账户AccountPO对象 |
| AccountData.addAccount(AccountPO po) | 语法 | Public void addAccount(AccountPO po) |
| 前置条件 | 数据库中不存在相同名称的AccountPO对象 |
| 后置条件 | 数据库添加AccountPO对象 |
| AccountData.delAccount(AccountPO po) | 语法 | Public void delAccount(AccountPO po) |
| 前置条件 | 数据库中存在相同名称的AccountPO对象，此对象中应付和应收为0 |
| 后置条件 | 数据库删除AccountPO对象 |
| AccountData.updateAccount(AccountPO po1, AccountPO po2) | 语法 | Public void updateAccount(AccountPO po1, AccountPO po2) |
| 前置条件 | 数据库中存在相同名称的AccountPO对象，输入的两个AccountPO对象的名称相同 |
| 后置条件 | 数据库更新AccountPO对象的信息 |
| AccountData.findAccount(String name) | 语法 | Public AccountPO findAccount(String name) |
| 前置条件 | 无 |
| 后置条件 | 根据名称查找并返回AccountPO对象 |
| AccountData.getAllAccount() | 语法 | Public AccountPO[] getAllAccount() |
| 前置条件 | 无 |
| 后置条件 | 返回一个包含所有AccountPO对象的数组 |
| AccountData.addReceipt(ReceiptPO po) | 语法 | Public void addReceipt(ReceiptPO po) |
| 前置条件 | 收款单得到总经理审批通过 |
| 后置条件 | 数据库添加ReceiptPO对象 |
| AccountData.addPayment(PaymentPO po) | 语法 | Public void addPayment(PaymentPO po) |
| 前置条件 | 付款单得到总经理审批通过 |
| 后置条件 | 数据库添加PaymentPO对象 |
| AccountData.getAllReceipt() | 语法 | Public ReceiptPO[] getAllReceipt() |
| 前置条件 | 无 |
| 后置条件 | 返回一个包含所有ReceiptPO对象的数组 |
| AccountData.getAllPayment() | 语法 | Public PaymentPO[] getAllPayment() |
| 前置条件 | 无 |
| 后置条件 | 返回一个包含所有PaymentPO对象的数组 |
| AccountData.clear | 语法 | Public void clear() |
| 前置条件 | 进行期初建账之前 |
| 后置条件 | 清除所有AccountPO，ReceiptPO，PaymentPO对象 |

2. Commodity模块

2.1 类的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| CommodityDataController | 负责实现库存数据管理所需要的服务 |
| CommodityData | 库存与商品管理信息，完成商品的增删改查，维护商品的属性，管理库存的信息 |
| StockData | 库存领域模型对象，负责管理、修改商品分类及其属性 |
| SortData | 商品分类领域模型对象，负责管理、修改商品及其属性 |
| CommodityData | 商品领域模型对象，负责存储商品的相关信息 |
| CommodityRecord | 库存记录领域模型对象，负责存储库存中商品进出的信息 |

2.2CommodityDataController接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务 | | | |
| CommodityDataController.findGood | 语法 | | Public CommodityPO findGood(String name, String type) |
| 前置条件 | | 已创建一个CommodityData领域对象，输入符合规则 |
| 后置条件 | | 调用CommodityData领域的findGood方法 |
| CommodityDataController.findName(String name) | 语法 | | Public CommodityPO findName(String name) |
| 前置条件 | | 已创建一个CommodityData领域对象，输入符合规则 |
| 后置条件 | | 调用CommodityData领域的findName方法 |
| CommodityDataController.findType(String type) | 语法 | | Public CommodityPO findType(String type) |
| 前置条件 | | 已创建一个CommodityData领域对象，输入符合规则 |
| 后置条件 | | 调用CommodityData领域的findType方法 |
| CommodityDataController.addGood(CommodityPO po1, SortPO po2) | 语法 | | Public void addGood(CommodityPO po1, SortPO po2) |
| 前置条件 | | 已创建一个CommodityData领域对象，输入符合规则 |
| 后置条件 | | 调用CommodityData领域的addGood方法 |
| CommodityDataController.delGood(CommodityPO po) | 语法 | | Public void delGood(CommodityPO po) |
| 前置条件 | | 已创建一个CommodityData领域对象，输入符合规则，商品的库存数量为0 |
| 后置条件 | | 调用CommodityData领域的delGood方法 |
| CommodityDataController.updateGood(CommodityPO po1, CommodityPO po2) | 语法 | | Public void updateGood(CommodityPO po1, CommodityPO po2) |
| 前置条件 | | 已创建一个CommodityData领域对象，输入符合规则，输入的两个CommodityPO的名称与型号相同 |
| 后置条件 | | 调用CommodityData领域的updateGood方法 |
| CommodityDataController.addSort(SortPO po) | 语法 | | Public void addSort(SortPO po) |
| 前置条件 | | 已创建一个CommodityData领域对象，输入符合规则 |
| 后置条件 | | 调用CommodityData领域的addSort方法 |
| CommodityDataController.delSort(SortPO po) | 语法 | | Public void delSort(SortPO po) |
| 前置条件 | | 已创建一个CommodityData领域对象，输入符合规则，此分类下不存在其他SortPO对象或CommodityPO对象 |
| 后置条件 | | 调用CommodityData领域的delSort方法 |
| CommodityDataController.updateSort(SortPO po1, SortPO po2) | 语法 | | Public void updateSort(SortPO po1, SortPO po2) |
| 前置条件 | | 已创建一个CommodityData领域对象，输入符合规则 |
| 后置条件 | | 调用CommodityData领域的updateSort方法 |
| CommodityDataController.addGift(CommodityPO po) | 语法 | | Public void addGift(CommodityPO po) |
| 前置条件 | | 已创建一个CommodityData领域对象，输入符合规则，数量小于库存内相同名称与型号CommodityPO对象的数量 |
| 后置条件 | | 调用CommodityData领域的addGift方法 |
| CommodityDataController.findGift(String name, String type) | 语法 | | Public CommodityPO findGift(String name, String type) |
| 前置条件 | | 已创建一个CommodityData领域对象，输入符合规则 |
| 后置条件 | | 调用CommodityData领域的findGift方法 |
| CommodityDataController.delGift(CommodityPO po) | 语法 | | Public void delGift(CommodityPO po) |
| 前置条件 | | 已创建一个CommodityData领域对象，输入符合规则 |
| 后置条件 | | 调用CommodityData领域的delGift方法 |
| CommodityDataController.getAll() | 语法 | | Public CommodityPO[] getAll() |
| 前置条件 | | 已创建一个CommodityData领域对象，输入符合规则 |
| 后置条件 | | 调用CommodityData领域的getAll方法 |
| CommodityDataController.clear() | 语法 | | Public void clear() |
| 前置条件 | | 进行期初建账之前，已创建一个CommodityData领域对象，输入符合规则 |
| 后置条件 | | 调用CommodityData领域的clear方法 |
| 需要的方法 | | | |
| CommodityData.findGood(String name,String type) | | 根据名称与型号查找商品 | |
| CommodityData.findName(String name) | | 根据名称查找商品 | |
| CommodityData.findType(String type) | | 根据型号查找商品 | |
| CommodityData.addGood(CommodityPO po1, SortPO po2) | | 在分类中添加商品 | |
| CommodityData.delGood(CommodityPO po) | | 删除商品 | |
| CommodityData.updateGood(CommodityPO po1, CommodityPO po2) | | 修改商品信息 | |
| CommodityData.addSort(SortPO po) | | 添加分类 | |
| CommodityData.delSort(SortPO po) | | 删除分类 | |
| CommodityData.updateSort(SortPO po1, SortPO po2) | | 修改分类信息 | |
| CommodityData.addGift(CommodityPO po) | | 添加赠品 | |
| CommodityData.findGift(String name, String type) | | 根据名称与型号查找赠品 | |
| CommodityData.delGift(CommodityPO po) | | 删除赠品 | |
| CommodityData.getAll() | | 获得所有商品 | |
| CommodityData.clear() | | 清除所有数据 | |

2.3CommodityData接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| CommodityData.findGood | 语法 | Public CommodityPO findGood(String name, String type) |
| 前置条件 | 无 |
| 后置条件 | 按名称和型号查找并返回相应的CommodityPO结果 |
| CommodityData.findName(String name) | 语法 | Public CommodityPO findName(String name) |
| 前置条件 | 无 |
| 后置条件 | 按名称查找并返回相应的CommodityPO结果 |
| CommodityData.findType(String type) | 语法 | Public CommodityPO findType(String type) |
| 前置条件 | 无 |
| 后置条件 | 按型号查找并返回相应的CommodityPO结果 |
| CommodityData.addGood(CommodityPO po1, SortPO po2) | 语法 | Public void addGood(CommodityPO po1, SortPO po2) |
| 前置条件 | 数据库中不存在相同的CommodityPO对象 |
| 后置条件 | 在数据库中添加一个CommodityPO对象 |
| CommodityData.delGood(CommodityPO po) | 语法 | Public void delGood(CommodityPO po) |
| 前置条件 | 数据库中存在相同的CommodityPO对象，此商品的库存数量为0 |
| 后置条件 | 数据库删除此CommodityPO对象 |
| CommodityData.updateGood(CommodityPO po1, CommodityPO po2) | 语法 | Public void updateGood(CommodityPO po1, CommodityPO po2) |
| 前置条件 | 数据库中存在相同名称与型号的CommodityPO，输入的两个CommodityPO的名称与型号相同 |
| 后置条件 | 更新CommodityPO信息 |
| CommodityData.addSort(SortPO po) | 语法 | Public void addSort(SortPO po) |
| 前置条件 | 数据库不存在相同名称的SortPO对象 |
| 后置条件 | 数据库添加一个SortPO对象 |
| CommodityData.delSort(SortPO po) | 语法 | Public void delSort(SortPO po) |
| 前置条件 | 数据库中存在相同名称的SortPO对象，此分类下不存在其他SortPO对象或CommodityPO对象 |
| 后置条件 | 数据库删除这个SortPO对象 |
| CommodityData.updateSort(SortPO po1, SortPO po2) | 语法 | Public void updateSort(SortPO po1, SortPO po2) |
| 前置条件 | 数据库存在相同名称的SortPO对象，输入的两个SortPO对象名称一致 |
| 后置条件 | 数据库更新这个SortPO对象信息 |
| CommodityData.addGift(CommodityPO po) | 语法 | Public void addGift(CommodityPO po) |
| 前置条件 | 数据库不存在相同名称与型号的GiftPO，数量小于库存内相同名称与型号CommodityPO对象的数量 |
| 后置条件 | 数据库添加GiftPO对象，库存中相同名称与型号的CommodityPO对象数量减少 |
| CommodityData.findGift(String name, String type) | 语法 | Public CommodityPO findGift(String name, String type) |
| 前置条件 | 无 |
| 后置条件 | 根据名字和型号查找并返回GiftPO |
| CommodityData.delGift(CommodityPO po) | 语法 | Public void delGift(CommodityPO po) |
| 前置条件 | 数据库中存在相同名称与型号的GiftPO对象 |
| 后置条件 | 数据库删除GiftPO对象，库存中相同名称与型号的CommodityPO数量增加 |
| CommodityData.getAll() | 语法 | Public CommodityPO[] getAll() |
| 前置条件 | 无 |
| 后置条件 | 获得所有的CommodityPO对象 |
| CommodityData.clear() | 语法 | Public void clear() |
| 前置条件 | 进行期初建账之前 |
| 后置条件 | 删除所有CommodityPO与SortPO对象 |