4.2.2 CommodityData模块

（1）模块概述

CommodityData模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

CommodityData模块的职责及借款参见软件体系结构描述文档。

（2）整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。Commodity模块对应的接口为展示层与业务逻辑层之间的businesslogicservice.commodityblservice.CommodityblService接口与业务逻辑层与数据层之间的dataservice.commoditydataservice.CommodityblService接口。为了隔离业务逻辑职责和逻辑控制职责，我们增加了CommodityDataController，这样CommodityDataController会将对商品与库存的数据存储委托给CommodityData对象。CommodityPO是作为商品的持久化对象被添加到设计模型中去的。SortPO是作为商品种类的持久化对象被添加到设计模型中去的。而StorkManangement存储库存信息，管理SortManangement对象。SortManangement存储SortPO持久化对象。CommodityManangement存储商品CommodityPO持久化对象。

CommodityData模块的设计如图

CommodityData各个类的职责如表

|  |  |
| --- | --- |
| 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 |

（3）模块内部类的接口规范

CommodityDataController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 | |

CommodityData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 |

（4）数据层的动态模型