快递物流系统

体系结构设计文档

第45组

**组员**

陈之威

刘璇琳

王子安

张诗琦

更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| **修改人员** | **日期** | **变更原因** | **版本号** |
| 陈之威 | 2015-10-18 | 文档框架完成并整合子安的接口规范 | V1.0 |
| 张诗琦 | 2015-10-19 | 完善业务逻辑接口规范和数据接口规范 | V1.0 |
| 陈之威 | 2015-10-20 | 完整文档完成 | V2.0 |
| 陈之威 | 2015-10-26 | 补充词汇表与代码示例截图 | V3.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

TOC \t "heading 1, 1,heading 2, 2,heading 3, 3"

引言 PAGEREF \_Toc \h 3

编制目的 PAGEREF \_Toc1 \h 3

词汇表 PAGEREF \_Toc2 \h 3

参考资料 PAGEREF \_Toc3 \h 3

产品描述 PAGEREF \_Toc4 \h 4

逻辑视角 PAGEREF \_Toc5 \h 4

组合视角 PAGEREF \_Toc6 \h 6

开发包图 PAGEREF \_Toc7 \h 6

运行时进程 PAGEREF \_Toc8 \h 9

物理部署 PAGEREF \_Toc9 \h 9

接口视角 PAGEREF \_Toc10 \h 10

模块的职责 PAGEREF \_Toc11 \h 10

用户界面层的分解 PAGEREF \_Toc12 \h 12

用户界面层模块的职责 PAGEREF \_Toc13 \h 13

用户界面模块的接口规范 PAGEREF \_Toc14 \h 14

用户界面模块设计原理 PAGEREF \_Toc15 \h 14

业务逻辑层的分解 PAGEREF \_Toc16 \h 14

业务逻辑层模块的职责 PAGEREF \_Toc17 \h 14

业务逻辑模块的接口模范 PAGEREF \_Toc18 \h 15

数据层的分解 PAGEREF \_Toc19 \h 29

数据层模块的职责 PAGEREF \_Toc20 \h 29

数据层模块的接口规范 PAGEREF \_Toc21 \h 30

6. 信息视角 PAGEREF \_Toc22 \h 38

6.1 数据持久化对象 PAGEREF \_Toc23 \h 38

6.2 数据格式 PAGEREF \_Toc24 \h 39

# 引言

## 编制目的

本报告详细完成对快递物流行业快递物流管理系统的概要设计，达到指导详细设计和开发的目的，同时实现测试人员及用户的沟通。

本报告面向开发人员、测试人员及最终用户编写，是了解系统的导航。

## 词汇表

|  |  |
| --- | --- |
| 词汇名称 | 词汇含义 |
| \_ui | 表示某展示层 |
| \_bl | 表示某逻辑层 |
| \_data | 表示某数据层 |
| \_service | 表示某层接口服务 |
| \_po | 表示某数据单一持久化对象 |
| \_vo | 表示某数据可视化对象 |

## 参考资料

* IEEE标准
* 《快递物流系统——用例文档》
* 《快递物流系统——软件需求规格说明文档》

# 产品描述

参考快递物流行业快递物流管理系统用例文档和快递物流行业快递物流管理系统软件规格说明中对产品的概括描述。

# 逻辑视角

快递物流行业快递物流管理系统中，选择了分层体系结构的风格，将系统分为3层（展示层、业务逻辑层、数据层）能够很好的示意整个高层抽象。展示层包括GUI页面的实现，业务逻辑层包含业务逻辑处理的实现，数据层负责数据的持久化和访问。分层体系结构的逻辑视角和逻辑设计方案如图1和图二所示。

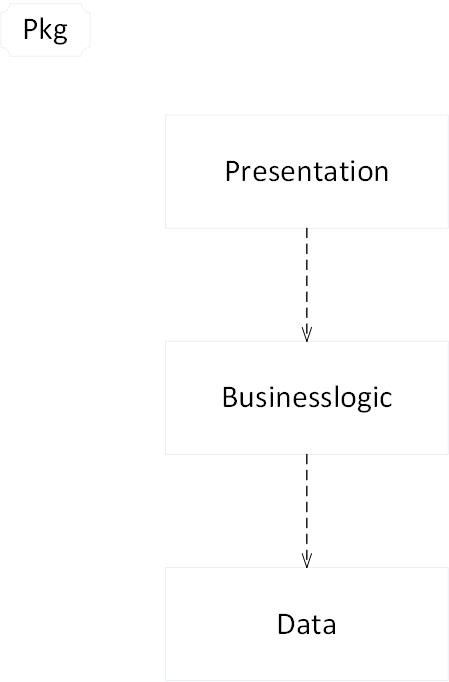


图1参照体系结构风格的包图表达逻辑视角

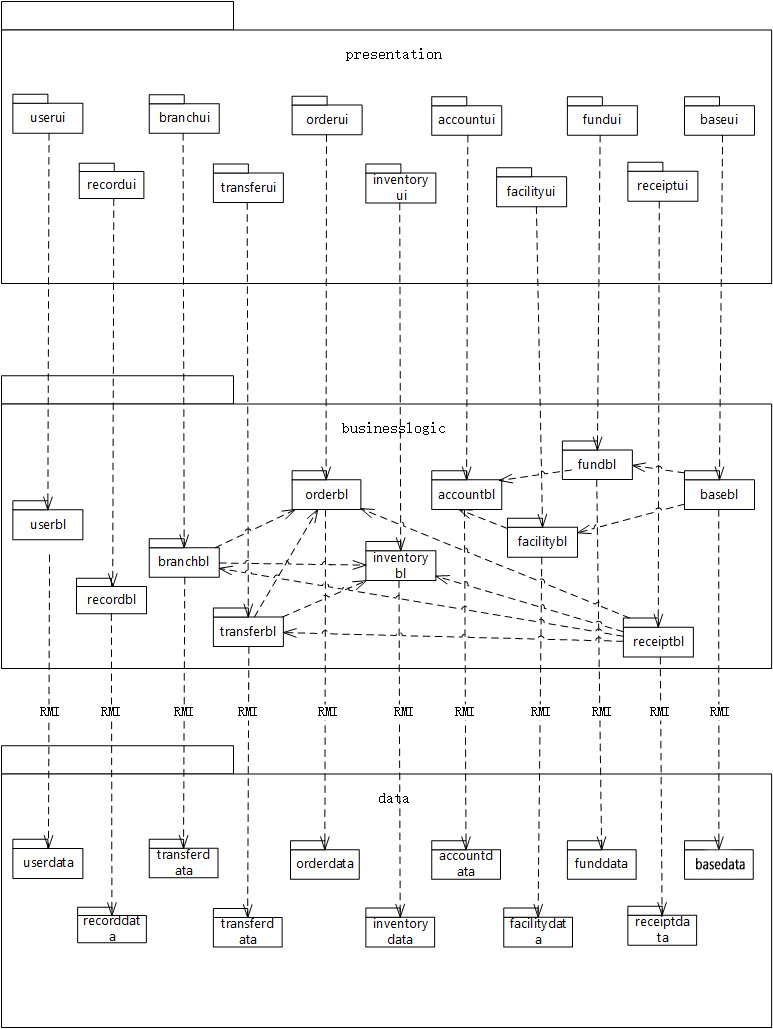


图2软件体系结构逻辑设计方案

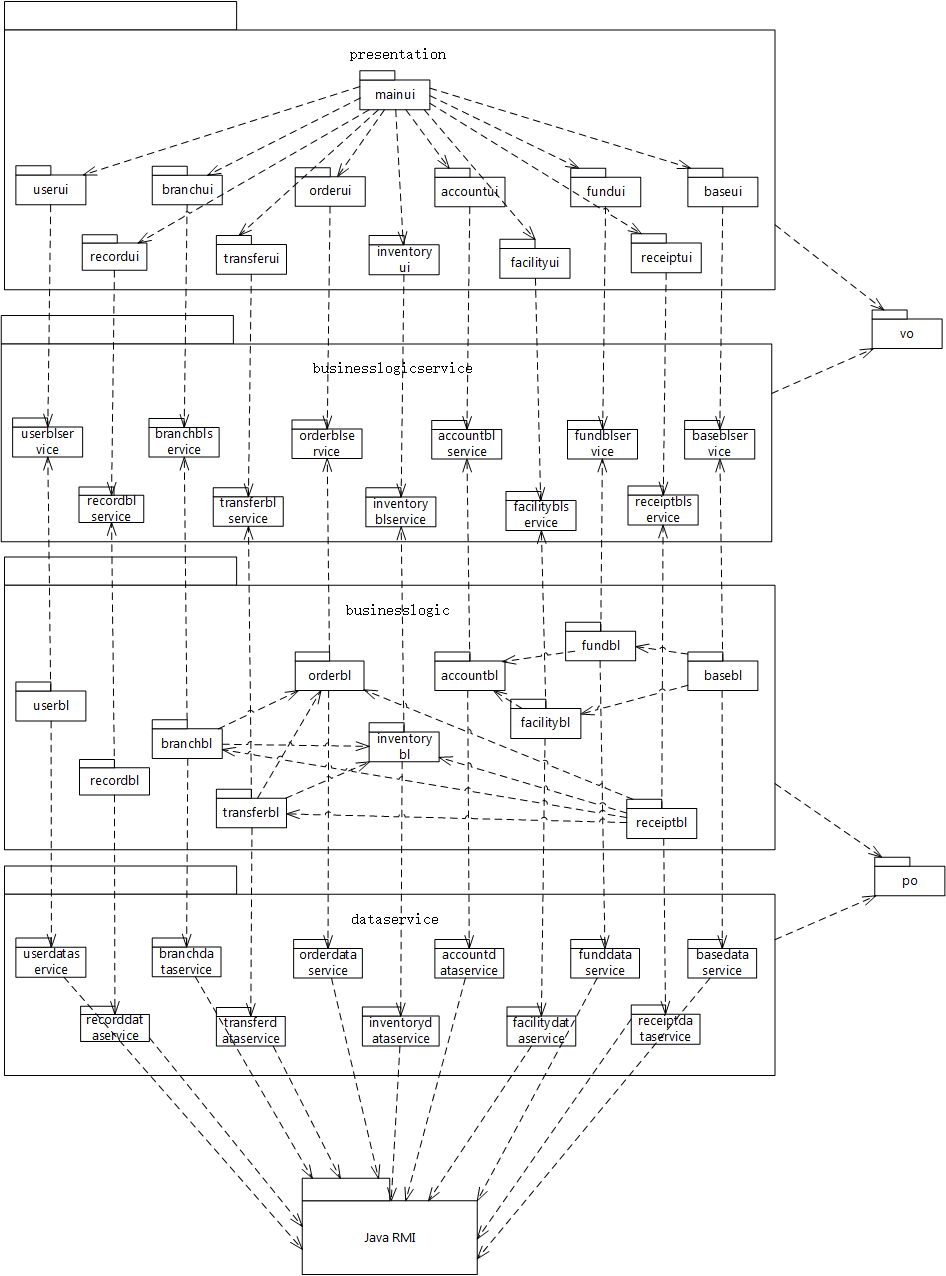
# 组合视角

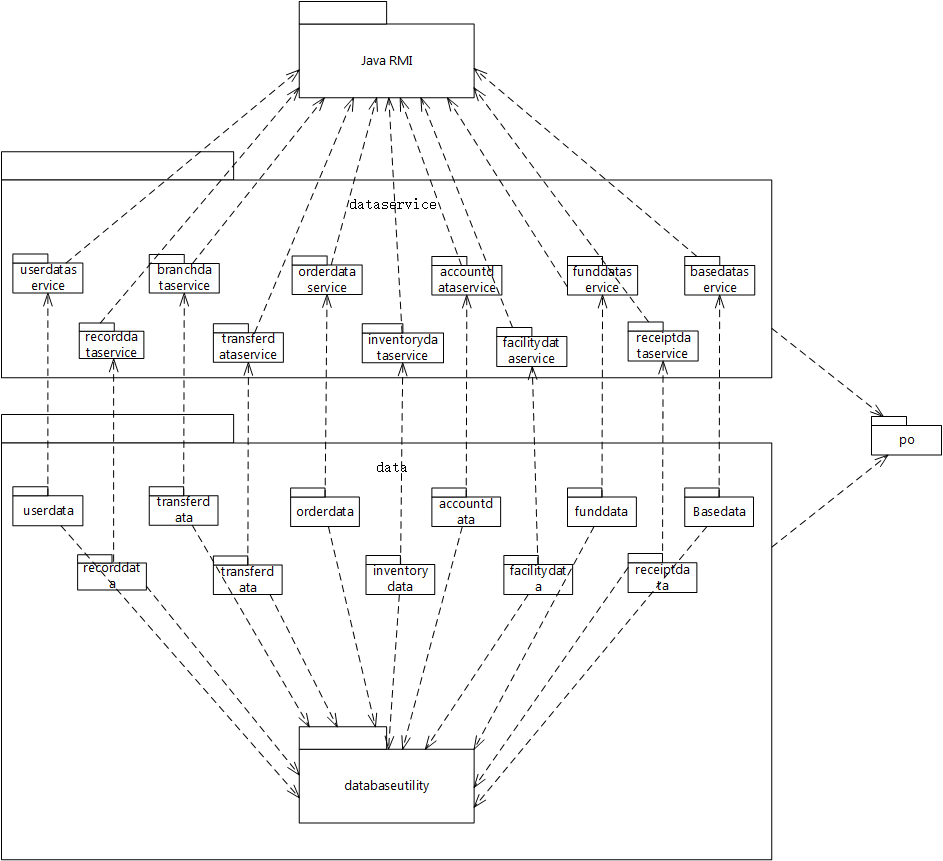
## 开发包图

快递物流行业快递物流管理系统的最终开发包设计如表1所示

**表1快递物流行业快递物流管理系统的最终开发包设计**

|  |  |
| --- | --- |
| **开发（物理包）** | **依赖的其他开发包** |
| mainui | orderui, inventoryui,facilityui,branchui,transferui, fundui, accountui,recordui, baseui, receiptui, userui |
| orderui | orderdataservice, 界面类库包 |
| orderblservice |  |
| orderbl | orderblservice, po |
| orderdataservice | Java RMI, po |
| orderdata | orderdataservice, databaseutility |
| inventoryui | inventoryblservice,界面类库包 |
| inventoryblservice |  |
| inventorybl | inventoryblservice, po |
| inventorydataservice | Java RMI, po |
| inventorydata | inventorydataservice, databaseutility |
| facilityui | facilityblservice,界面类库包 |
| facilityblservice |  |
| facilitybl | facilityblservice, accountbl, po |
| facilitydataservice | Java RMI, po |
| facilitydata | facilitydataservice, databaseutility |
| branchui | branchblservice, 界面类库包 |
| branchblservice |  |
| branchbl | branchblservice, orderbl, inventory, po |
| branchdataservice | Java RMI, po |
| branchdata | branchdataservice, databaseutility |
| transferui | transferblservice, 界面类库包 |
| transferblservice |  |
| transferbl | transferblservice, orderbl, inventory, po |
| transferdataservice | Java RMI, po |
| transferdata | transferdataservice, databaseutility |
| fundui | fundblservice, 界面类库包 |
| fundblservice |  |
| fundbl | fundblservice, accountbl, po |
| funddataservice | Java RMI, po |
| funddata | funddataservice, databaseutility |
| accountui | accountblservice, 界面类库包 |
| accountblservice |  |
| accountbl | accountblservice, fundbl, po |
| accountdataservice | Java RMI, po |
| accountdata | accountdataservice, databaseutility |
| recordui | recordblservice, 界面类库包 |
| recordblservice |  |
| recordbl | recordblservice, fundbl, po |
| recorddataservice | Java RMI, po |
| recorddata | recorddataservice, databaseutility |
| baseui | baseblservice, 界面类库包 |
| baseblservice |  |
| basebl | baseblservice, fundbl, facilitybl, po |
| basedataservice | Java RMI, po |
| basedata | basedataservice, databaseutility |
| receiptui | receiptblservice, 界面类库包 |
| receiptblservice |  |
| receiptbl | receiptblservice, branchbl, transferbl, inventorybl, orderbl, po |
| receiptdataservice | Java RMI, po |
| receiptdata | receiptdataservice, databaseutility |
| userui | userblservice, 界面类库包 |
| userblservice |  |
| userbl | userblservice, po |
| userdataservice | Java RMI, po |
| userdata | userdataservice, databaseutility |
| vo |  |
| po |  |
| utilitybl |  |
| 界面类库包 |  |
| Java RMI |  |
| databaseutility |  |

图 3 客户端开发包图

图 4 服务器端开发包图

## 运行时进程

在快递物流行业快递物流管理系统中，会有多个客户端进程和一个服务器端进程，其进程图如图5所示。结合部署图，客户端进程是在客户端机器上运行，服务器端进程是在服务器端机器上运行。

## 物理部署

灯具照明行业进销存系统中客户端构件是放在客户端机上，服务器端构件是放在服务器端机器上。在客户端节点上，还要部署RMIStub构件。由于Java RMI 构件属于JDK 1.8的一部分。所以，在系统JDK环境已经设置好的情况下，不需要独立部署。部署图如图6所示。

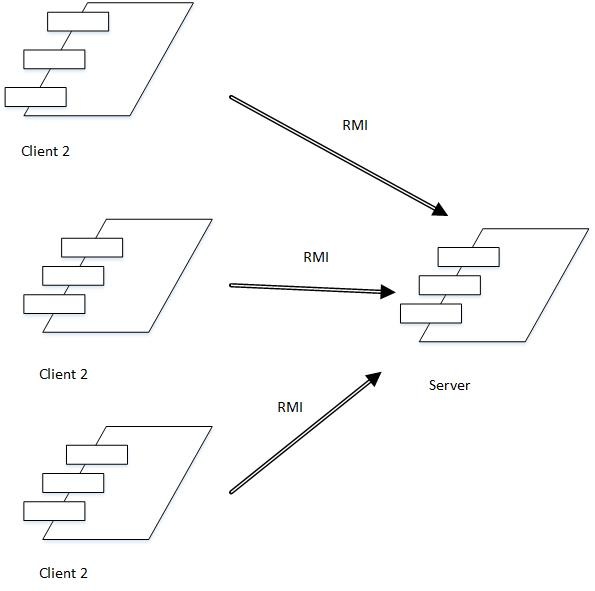


图 5 进程图

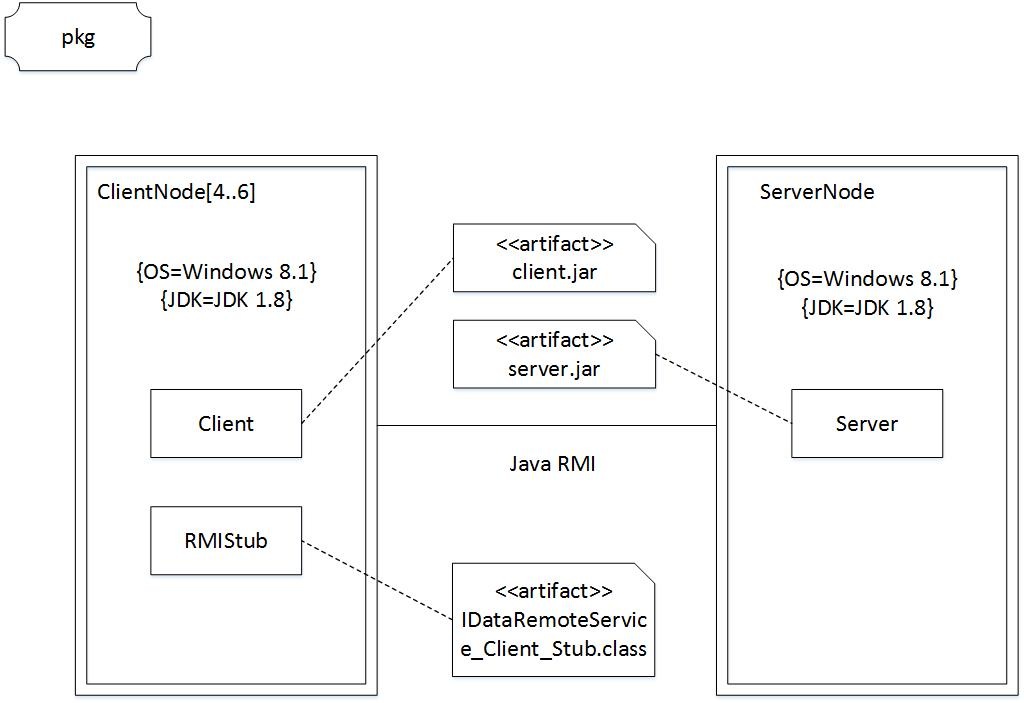


图 6 部署图

# 接口视角

## 模块的职责

客户端模块和服务器模块视图分别如图7和图8所示。客户端各层和服务器端各层的职责分别如表2和表3所示。

启动模块

启动模块

展示层

网络模块

业务逻辑层

数据层

网络模块

图7 客户端模块视图 图8 服务器端模块视图

**表2客户端各层的职责**

|  |  |
| --- | --- |
| **层** | **职责** |
| 启动模块 | 负责初始化网络通信机制，启动用户界面 |
| 用户界面层 | 基于窗口的进销存管理系统客户端用户界面 |
| 业务逻辑层 | 对于用户界面的输入进行响应并进行业务处理逻辑 |
| 客户端网络模块 | 利用java RMI机制查找RMI服务 |

**表3服务器端各层的职责**

|  |  |
| --- | --- |
| **层** | **职责** |
| 启动模块 | 负责初始化网络通信机制，启动用户界面 |
| 数据层 | 负责数据的持久化及数据访问接口 |
| 服务器端网络模块 | 利用Java RMI机制开启RMI服务，注册RMI服务 |

每一层只是使用下方直接接触的层。层与层之间仅仅是通过接口的调用来完成的。层之间调用的接口如表4所示。

**表4层之间调用的接口**

|  |  |  |
| --- | --- | --- |
| **接口** | **服务调用方** | **服务提供方** |
| orderblservice  inventoryblservice  facilityblservice  branchblservice  transferblservice  fundblservice  accountblservice  recordblservice  baseblservice  receiptblservice  userblservice | 客户端展示层 | 客户端业务逻辑层 |
| ordedataservice  inventorydataservice  facilitydataservice  branchdataservice  transferdataservice  funddataservice  accountdataservice  recorddataservice  basedataservice  receiptdataservice  userdataservice | 客户端业务逻辑层 | 服务器端数据层 |

借用车辆信息管理用例来说明层之间的调用，如图9所示。每一层之间都是由上一层依赖了一个接口(需接口)，而下层实现这个接口(供接口)。facilityblservice提供了facility界面所需要的所有业务逻辑功能。facilitydataservice提供了对数据库的增、删、改、查等操作。这样的实现就大大降低了层与层之间的耦合。

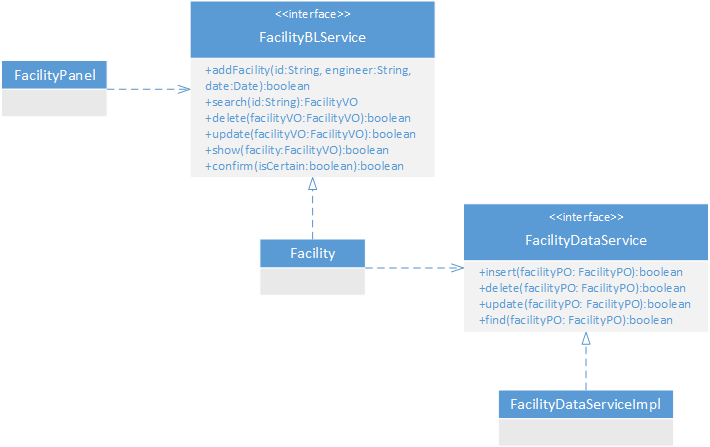
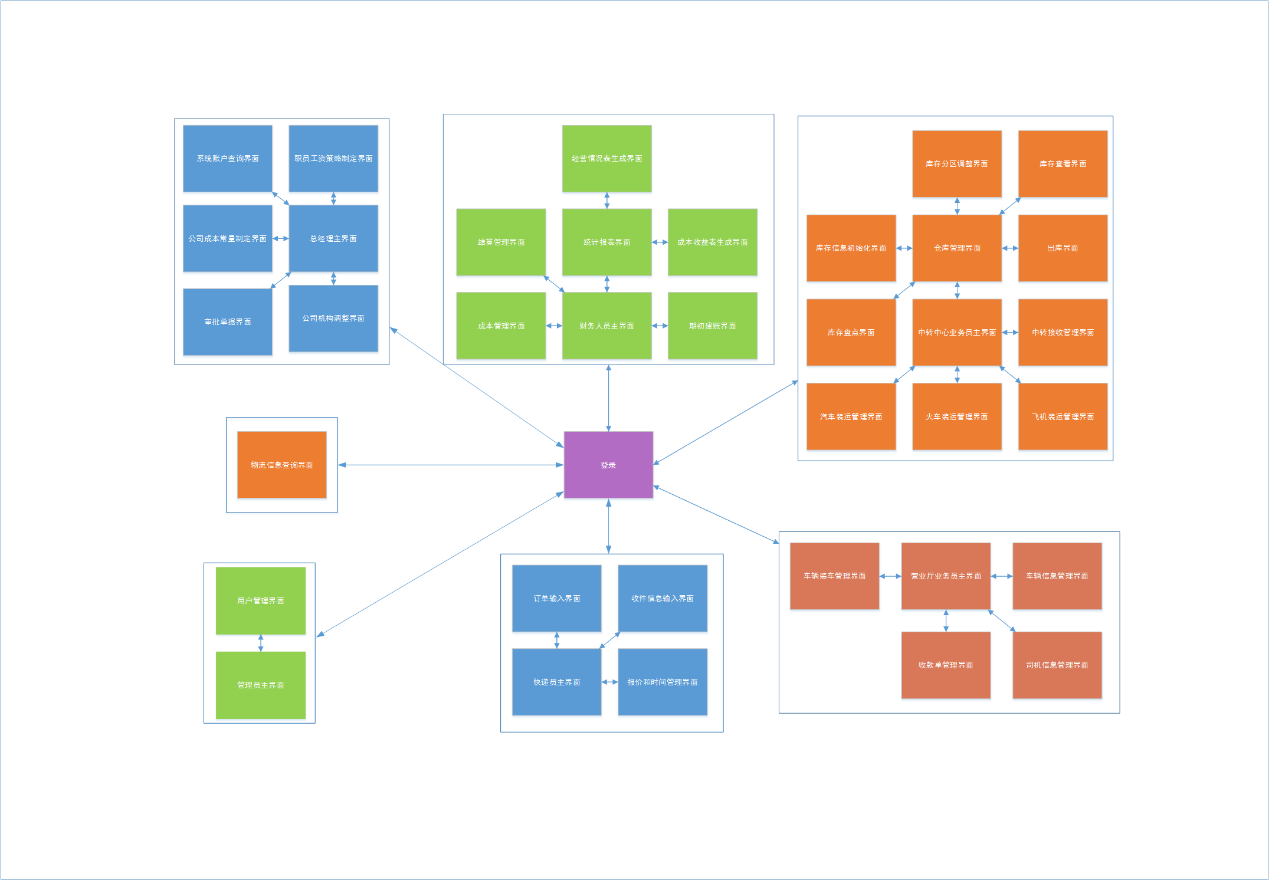


图9 车辆信息管理用例层之间调用的接口

## 用户界面层的分解

系统存在37个用户界面,他们分别为登陆界面、物流信息查询界面、快递员界面组（快递员主界面、订单输入界面、报价和时间管理界面、收件信息输入界面）、营业厅业务员界面组（营业厅业务员主界面、车辆装车管理界面、车辆信息管理界面、司机信息管理界面、收款单管理界面）、中转中心业务员界面组（中转中心业务员主界面、库存盘点界面、中转接收管理界面、汽车装运管理界面、火车装运管理界面、飞机装运管理界面、仓库管理界面、库存信息初始化界面、出库界面、库存分区调整界面、库存查看界面）、财务人员界面组（财务人员主界面、成本管理界面、结算管理界面、期初建账界面、统计报表界面、成本收益表生成界面、经营情况表生成界面）、总经理界面组（总经理主界面、公司机构调整界面、审批单据界面、公司成本常量制定界面、系统账户查询界面、职员工资策略制定界面）、管理员界面组（管理员主界面、用户管理界面）。其用户界面之间的跳转如下图所示：



其中所有用户界面通过同一个类处理，即主用户界面类（MainFrame）



### 用户界面层模块的职责

MainFrame主要用于负责界面的显示和界面之间的跳转工作。

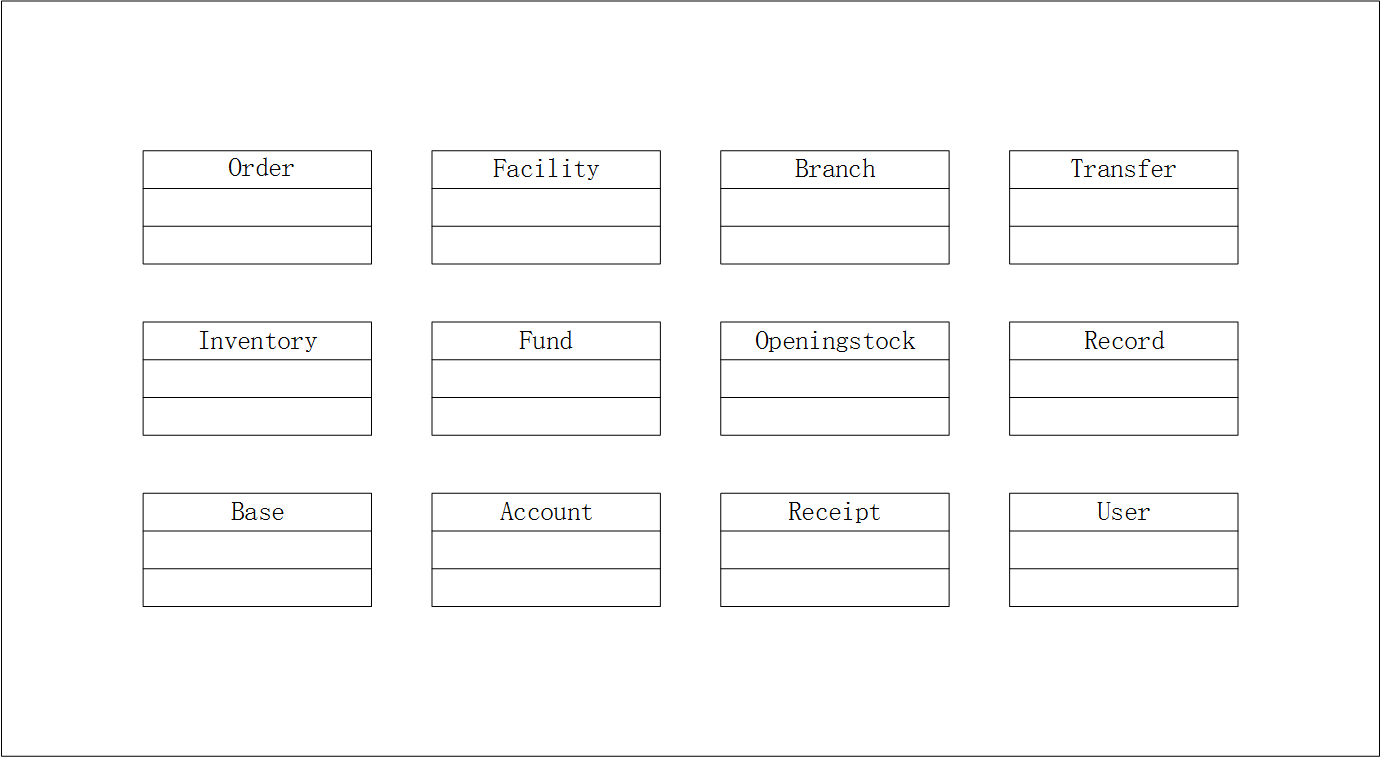
### 用户界面模块的接口规范

|  |  |  |
| --- | --- | --- |
| **MainFrame** | 语法 | init(args:String[]) |
| 前置条件 | 用户触发相应需要显示界面的条件 |
| 后置条件 | 显示Frame并且加载Panel |

### 用户界面模块设计原理

用户界面用java的Swing和AWT库来实现。

## 业务逻辑层的分解



### 业务逻辑层模块的职责

|  |  |
| --- | --- |
| **模块** | **职责** |
| Orderbl | 负责实现订单管理界面所需要的服务 |
| Facilitybl | 负责实现营业厅运输工具管理界面所需要的服务 |
| Branchbl | 负责实现营业厅运输管理界面所需要的服务 |
| Transferbl | 负责实现中转中心运输管理界面所需要的服务 |
| Inventorybl | 负责实现库存管理界面所需要的服务 |
| Fundbl | 负责实现资金管理界面所需要的服务 |
| Openingstockbl | 负责实现建账界面所需要的服务 |
| Recordbl | 负责实现统计报表界面所需要的服务 |
| Basebl | 负责实现公司基础数据管理界面所需要的服务 |
| Accountbl | 负责实现公司职员管理界面所需要的服务 |
| Receiptbl | 负责实现审批单据界面所需要的服务 |
| Userbl | 负责实现系统用户管理所需要的服务 |

### 业务逻辑模块的接口模范

**Orderbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| **Order.addOrder** | 语法 | publicResultMessage addOrder(ArrayList<CommodityPO>commmodities, OrderPO order ); |
| 前置条件 | 待寄货物信息、收件人信息、寄件人信息符合输入规则 |
| 后置条件 | 。 |
| **Order.inquireOrder** | 语法 | Public OrderVO inquireOrder (String orderNumber, String senderName) |
| 前置条件 | 无 |
| 后置条件 | 根据输入的orderNumber查找相应的订单物流信息并返回。 |
| **Order.confirmOrder** | 语法 | public ConfirmState confirmOrder(String orderNumber) |
| 前置条件 | 订单号、收件人姓名符合输入规则 |
| 后置条件 | 查找相应的Order，更新订单的状态为已签收，生成收件单号。 |
| **Order.costAndTime** | 语法 | public ResultMessage costAndTime(OrderVO order) |
| 前置条件 | 被查询的订单已经存在。 |
| 后置条件 | 查找相应的Order，根据输入的订单信息，显示该订单的运费和预估到达时间。 |
| 需要的服务（需接口） | | |
| **服务名** | 服务 | |
| **DatabaseFactory.getOrderDataBase** | 得到Order的服务的引用 | |
| **OrderDateService.find**  **(OrderPO po)** | 查找单一持久化对象 | |
| **OrderDataService.insert**  **(OrderPO po)** | 插入单一持久化对象 | |
| **OrderDataService.update**  **(OrderPO po)** | 更新单一持久化对象 | |

**Facilitybl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Facility.addFacility** | 语法 | public ResultMessage addFacility  (FacilityVO facility) |
| 前置条件 | 车辆信息输入合法 |
| 后置条件 | 添加新的车辆信息，持久化更新涉及的对象的数据 |
| **Facility.deleteFacility** | 语法 | public ResultMessage deleteFacility (FacilityPO facility) |
| 前置条件 | 车辆信息、ID输入合法 |
| 后置条件 | 删除车辆信息，持久化更新涉及的对象的数据 |
| **Facility.modifyFacility** | 语法 | public ResultMessage modifyFacility  (FacilityVO facility) |
| 前置条件 | 车辆信息输入合法 |
| 后置条件 | 更改车辆信息，持久化更新涉及的对象的数据 |
| **Facility.findFacility** | 语法 | public FacilityVO findFacility  (String facilityId) |
| 前置条件 | 车辆信息、ID输入合法 |
| 后置条件 | 查找到该车辆信息 |
| **Facility.addDriver** | 语法 | PublicResultMessage addDriver  (AccountVO account) |
| 前置条件 | 司机信息输入合法 |
| 后置条件 | 添加新的司机信息，持久化更新涉及的对象的数据 |
| **Facility.deleteDriver** | 语法 | public ResultMessage deleteDriver (AccountVO account) |
| 前置条件 | 司机信息、ID输入合法 |
| 后置条件 | 删除司机信息，持久化更新涉及的对象的数据 |
| **Facility.modifyDriver** | 语法 | public ResultMessage modifyDriver  (AccountVO account) |
| 前置条件 | 司机信息输入合法 |
| 后置条件 | 更改司机信息，持久化更新涉及的对象的数据 |
| **Facility.findDriver** | 语法 | public AccountVO findDriver  (intAccountId) |
| 前置条件 | 司机信息、ID输入合法 |
| 后置条件 | 查找到该司机信息 |
| **需要的服务（需接口）** | | |
| **服务名** | 服务 | |
| **FacilityDataService.**  **findFacility (String id)** | 根据ID进行查找单一持久化对象 | |
| **FacilityDataService.**  **findDriver (String id)** | 根据ID进行查找单一持久化对象 | |
| **FacilityDataService.**  **addFacility (FacilityPO po)** | 添加单一化持久对象 | |
| **FacilityDataService.**  **deleteFacility (FacilityPO po o)** | 删除单一化持久对象 | |
| **FacilityDataService.**  **modifyFacility (FacilityPO po)** | 更改单一化持久对象 | |
| **FacilityDataService.**  **getFacilityData()** | 得到Facility数据的服务引用 | |

**Branchbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Branch.getDeliverID** | 语法 | public String getDeliverID() |
| 前置条件 | 得到数据，启动新建一个装车单 |
| 后置条件 | 返回新建的装车单的ID |
| **Branch.getReceiptID** | 语法 | public String getReceiptID() |
| 前置条件 | 得到数据，启动新建一个营业厅到达单 |
| 后置条件 | 返回新建的营业厅到达单的ID |
| **Branch.getExpressID** | 语法 | public String getExpressID() |
| 前置条件 | 得到数据，启动新建一个快递员派件单。 |
| 后置条件 | 返回新建的派件单的ID |
| **Branch.getAllCommidities** | 语法 | public ArrayList<CommodityPO> getAllCommodities () |
| 前置条件 | 得到数据，需要显示等待派送的商品列表 |
| 后置条件 | 返回所有的待送的商品列表 |
| **Branch.getAllOrderNumber** | 语法 | public ArrayList<String> getAllOrderNumber() |
| 前置条件 | 得到数据，需要显示到达商品或者等待派送商品的订单号 |
| 后置条件 | 返回所有的到达商品或者等待派送商品的订单号 |
| **Branch.branchReceipt** | 语法 | public BranchReceiptList branchReceipt(CommodityVO comoditiies,ArrayList<BranchList>branchReceiptList) |
| 前置条件 | 有新的一批货物送达营业厅 |
| 后置条件 | 系统生成并显示货物到达营业厅的到达单。 |
| **Branch.branchDeliver** | 语法 | public BranchDeliveryList branchDeliver(ArrayList<BranchList>branchReceiptList,AccountVO account) |
| 前置条件 | 营业厅生成了新的到达单据且有空闲的快递员 |
| 后置条件 | 系统生成派件单且更新快递物流信息。 |
| **Branch.submit** | 语法 | public ResultMessage submit(BranchList branchList) |
| 前置条件 | 单据已经被制定出 |
| 后置条件 | 单据被提交发送给经理进行审批 |
| **Branch.save** | 语法 | public BranchList save() |
| 前置条件 | 单据已经被制定出 |
| 后置条件 | 单据保存为草稿 |
| **Branch.truckDeliver** | 语法 | public TruckDeliverList truckDeliver(ArrayList< TruckDeliverList > truckDeliverList，FacilityVO facility); |
| 前置条件 | truckDeliverList符合输入规则 |
| 后置条件 | 系统自动生成运费，并生成装车单，更新营业厅车辆信息。 |
| **需要的服务（需接口）** | | |
| **服务名** | 服务 | |
| **DatabaseFactory.getBranchDataBase** | 得到Branch的服务的引用 | |
| **BranchDataService.**  **find(String id)** | 根据查找单一持久化对象 | |
| **BranchDataService.**  **insert(BranchPO po)** | 插入单一持久化对象 | |
| **BranchDataService.**  **delete(BranchPO po)** | 删除单一持久化对象 | |
| **BranchDataService.**  **update(BranchPO po)** | 更新单一持久化对象 | |
| **BranchDataService.init** | 初始化单一持久化对象 | |

**Transferbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Transfer.getReceiptID** | 语法 | public String getReceiptID() |
| 前置条件 | 得到数据，启动新建一个中转中心到达单 |
| 后置条件 | 返回新建的中转中心到达单的ID |
| **Transfer.getTransferID** | 语法 | public String getTransferID() |
| 前置条件 | 得到数据，启动新建一个转运中转单 |
| 后置条件 | 返回新建的转运中转单的ID |
| **Transfer.getAllCommodities** | 语法 | public ArrayList<CommodityVO> getAllCommodities() |
| 前置条件 | 得到数据，需要显示所有的待运货物。 |
| 后置条件 | 返回所有的待运货物列表 |
| **Transfer.planeTransfer** | 语法 | public TransferList planeTransfer(ArrayList<TransferList>planeTransferList, String []orderNumber,OrderVO order,InventoryVO inventory) |
| 前置条件 | planeTransferList的内容输入符合规则 |
| 后置条件 | 系统自动计算此次转运费用，生成新的飞机转运中转单，系统更新中转信息，提出出库申请 |
| **Transfer.truckTransfer** | 语法 | public TransferList truckTransfer(ArrayList<TransferList>truckTransferList, String []orderNumber,OrderVO order,InventoryVO inventory) |
| 前置条件 | truckTransferList的内容输入符合规则 |
| 后置条件 | 系统自动计算此次转运费用，生成新的火车转运中转单，系统更新中转信息，提出出库申请 |
| **Transfer.trainTransfer** | 语法 | public TransferListVO trainTransfer(ArrayList<TransferList>trainTransferList, String []orderNumber,OrderVO order,InventoryVO inventory) |
| 前置条件 | trainTransferList的内容输入符合规则 |
| 后置条件 | 系统自动计算此次转运费用，生成新的火车转运中转单，系统更新中转信息，提出出库申请 |
| **Transfer.submit** | 语法 | public TransferListVO submit() |
| 前置条件 | 单据已经被制定出 |
| 后置条件 | 单据被提交发送给经理进行审批 |
| **Transfer.save** | 语法 | public TransferListVO save() |
| 前置条件 | 单据已经被制定出 |
| 后置条件 | 单据保存为草稿 |
| **Transfer.receiptList** | 语法 | PublicTranferListVO receiptList( ArrayList<CommodityPO>commodities,ArrayList<TransferList>transferListNumber) |
| 前置条件 | 一批货物到达中转中心 |
| 后置条件 | 生成中转中心到达单，提出入库申请。 |
| 需要的服务（需接口） | | |
| **服务名** | 服务 | |
| **DatabaseFactory.getTransferDataBase** | 得到Transfer的服务的引用 | |
| **TransferDataService.**  **find(String id)** | 根据查找单一持久化对象 | |
| **TransferDataService.**  **insert(TransferPO po)** | 插入单一持久化对象 | |
| **TransferDataService.**  **delete(TransferPO po)** | 删除单一持久化对象 | |
| **TransferDataService.**  **update(TransferPO po)** | 更新单一持久化对象 | |
| **TransferDataService.init** | 初始化单一持久化对象 | |

**Inventorybl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Inventory.viewInventory** | 语法 | public InventoryViewVO viewInventory  (Date beginDate, Date endDate) |
| 前置条件 | 日期的输入合法 |
| 后置条件 | 系统显示此时间段内的出入库数量、金额、库存位置 |
| **Inventory.checkRecord** | 语法 | public InventoryCheckVO checkRecord( ) |
| 前置条件 | 已经得到库存单据数据 |
| 后置条件 | 显示截至盘点时间的库存快照 |
| **Inventory.getImportID** | 语法 | public String getImportID () |
| 前置条件 | 开始创建一个入库单 |
| 后置条件 | 得到该入库单的编号 |
| **Inventory.Import** | 语法 | public ResultMessage addCommodities(TransferArrivalListVO receipt) |
| 前置条件 | 选择进行入库的货物 |
| 后置条件 | 添加信息到进货单 |
| **Inventory.buildImport** | 语法 | public InventoryImportReceiptPO buildImport(); |
| 前置条件 | 选择并添加相应的数据（订单托运编号、入库日期、目的地、区号、排号、架号、位号）到进货单中 |
| 后置条件 | 生成进货单PO |
| **Inventory.submitImport** | 语法 | public ResultMessage submitImport(InventoryImportReceiptVO importReceipt); |
| 前置条件 | 单据已经被制定出 |
| 后置条件 | 单据被提交发送给经理进行审批 |
| **Inventory.getExportID** | 语法 | public String getExportID () |
| 前置条件 | 开始创建一个出库单 |
| 后置条件 | 得到该出库单的编号 |
| **Inventory.Export** | 语法 | public ResultMessage minusCommodities(TransferArrivalListVO receipt); |
| 前置条件 | 选择进行出库的货物 |
| 后置条件 | 添加信息到出库单 |
| **Inventory.buildExport** | 语法 | public InventoryExportReceiptPO buildExport(); |
| 前置条件 | 选择并添加相应的数据（订单托运编号、出库日期、目的地、装运形式、中转单编号）到出库单中 |
| 后置条件 | 生成出库单PO |
| **Inventory..submitExport** | 语法 | public ResultMessage submitExport(InventoryExportReceiptVO exportReceipt); |
| 前置条件 | 单据已经被制定出 |
| 后置条件 | 单据被提交发送给经理进行审 |
| **Inventory.getAdjustID** | 语法 | public String getAdjustID () |
| 前置条件 | 该区域库存达到警报值。开始创建一个调整单。 |
| 后置条件 | 得到该调整单的编号 |
| **Inventory.Adjust** | 语法 | public AdjustReceiptPO adjust( ); |
| 前置条件 | 输入被调整货物的订单托运编号、原先区号、排号、架号、位号及调整后的区号、排号、架号、位号 |
| 后置条件 | 添加信息到进货单 |
| **需要的服务（需接口）** | | |
| **服务名** | 服务 | |
| **InventoryDataService.**  **getImportID ()** | 得到入库单ID | |
| **InventoryDataService.getExportID ()** | 得到出库单ID | |
| **InventoryDataService.getAlarmID ()** | 得到调整单ID | |
| **InventoryDataService.getA()** | 得到区号 | |
| **InventoryDataService.getB()** | 得到排号 | |
| **InventoryDataService.getC()** | 得到架号 | |
| **InventoryDataService.getD()** | 得到位号 | |
| **InventoryDataService. getASize()** | 得到区容量 | |
| **InventoryDataService.getBSize()** | 得到排号容量 | |
| **InventoryDataService.getCSize()** | 得到架号容量 | |
| **InventoryDataService.getDSize()** | 得到位号容量 | |
| **InventoryDataService.getANum()** | 得到区此时容量 | |
| **InventoryDataService. getBNum()** | 得到排此时容量 | |
| **InventoryDataService. getCNum()** | 得到架此时容量 | |
| **InventoryDataService.getDNum()** | 得到位此时容量 | |
| **InventoryDataService.showImport()** | 返回所有入库单据的数据 | |
| **InventoryDataService.showExport()** | 返回所有出库单据的数据 | |
| **InventoryDataService.showAdjust()** | 返回所有调整单据的数据 | |
| **InventoryDataService.insertImport** | 增加一条po数据即生成入库单据 | |
| **InventoryDataService.insertExport** | \*增加一条po数据即生成出库单据 | |
| **InventoryDataService.insertAdjust** | 增加一条po数据即生成调整单据 | |

**Fundbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Fund.addFundReceipt** | 语法 | PublicResultMessage addFundReceipt  (String receiptType,ReceiptVO fundReceipt) |
| 前置条件 | 单据类型、内容输入合法 |
| 后置条件 | 添加新的单据，持久化更新涉及的对象的数据 |
| **Fund.deleteFundReceipt** | 语法 | public ResultMessage deleteFundReceipt (  String receiptType,String fundReceiptId) |
| 前置条件 | 单据类型、ID输入合法 |
| 后置条件 | 删除单据，持久化更新涉及的对象的数据 |
| **Fund.modifyFundReceipt** | 语法 | public ResultMessage modifyFundReceipt  (String receiptType, ReceiptVO fundReceipt) |
| 前置条件 | 单据类型、内容输入合法 |
| 后置条件 | 更改单据，持久化更新涉及的对象的数据 |
| **Fund.findFundReceipt** | 语法 | public ResultMessage findFundReceipt  (String receiptType, intfundReceiptId) |
| 前置条件 | 单据类型、ID输入合法 |
| 后置条件 | 查找到该单据 |
| **Fund.addOpeningStock** | 语法 | public ResultMessage addOpeningStock (  OpeningStockVO openingStock) |
| 前置条件 | 期初账内容输入合法 |
| 后置条件 | 添加新的期初账，持久化更新涉及的对象的数据 |
| **Fund.findOpeningStock** | 语法 | public ResultMessage findOpeningStock  (intopeningStockId) |
| 前置条件 | 期初账ID输入合法 |
| 后置条件 | 查找到该期初账 |
| **需要的服务（需接口）** | | |
| **服务名** | 服务 | |
| **FundDataService.**  **findReceipt (int id)** | 根据ID进行查找单一持久化对象 | |
| **FundDataService.**  **findOpeningStock (int id)** | 根据ID进行查找单一持久化对象 | |
| **FundDataService.**  **addReceipt(Receipt PO po)** | 添加单一化持久对象 | |
| **FundDataService.**  **addOpeningStock (OpeningStockPO po)** | 添加单一化持久对象 | |
| **FundDataService.**  **deleteReceipt (Receipt PO po o)** | 删除单一化持久对象 | |
| **FundDataService.**  **modifyReceipt (Receipt PO po)** | 更改单一化持久对象 | |
| **FundDataService.**  **getReceiptData** | 得到Receipt数据的服务引用 | |

**OpeningStockbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **OpeningStock.add** | 语法 | public ResultMessage add(  ArrayList<TransferVO >commodities,  ArrayList<BranchVO > branches,  ArrayList< AccountVO>accounts,  ArrayList< FacilityVO > facilities,  ArrayList< InventoryVO > inventories,  ArrayList< BankVO > banks); |
| 前置条件 | 机构、人员、车辆、库存、银行账户符合输入规范 |
| 后置条件 | 建立期初账目，持久化更新涉及的对象的数据 |
| **OpeningStock.find** | 语法 | public ResultMessage find  (int id) |
| 前置条件 | 期初账目ID输入合法 |
| 后置条件 | 查找到该期初账目 |
| **需要的服务（需接口）** | | |
| **服务名** | 服务 | |
| **OpeningStockDataService.**  **add(OpeningStockPO po)** | 添加单一持久化对象 | |
| **OpeningStockDataService.**  **find(int id)** | 根据ID进行查找单一持久化对象 | |
| **OpeningStockDataService.**  **init** | 初始化持久数据 | |

**Recordbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| **Record.getBusinessProcess** | 语法 | public BussinessProcessVO bussinessProcess(String begin, String end) |
| 前置条件 | 符合输入规则 |
| 后置条件 | 返回符合条件的所有入款单和收款单 |
| **RecordDataService.getBusinessCondition** | 语法 | public BussinessConditionVO bussinessCondition(String end) |
| 前置条件 | 符合输入规则 |
| 后置条件 | 返回收入利润支出数据 |
| **Record.income** | 语法 | public double income(String end) |
| 前置条件 | 时间区间符合输入规则 |
| 后置条件 | 返回计算得到的收入 |
| **Record.earning** | 语法 | public double earning(String end) |
| 前置条件 | 时间区间符合输入规则 |
| 后置条件 | 返回计算得到的利润 |
| **Record.expense** | 语法 | public double expense(String end) |
| 前置条件 | 时间区间符合输入规则 |
| 后置条件 | 返回计算得到的支出 |
| 需要的服务（需接口） | | |
| **服务名** | 服务 | |
| Record**DataService.** getBusinessProcess(String begindate,String enddate) | 返回1. 收款单2.付款单 | |
| Record**DataService.**getBusinessCondition(String enddate);  } | 返回信息： 1. 收入2. 支出。3. 利润：总收入-总支出 | |

**basebl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Base.setBase** | 语法 | public ResultMessage setBase  (BaseVO base) |
| 前置条件 | 城市距离、价格常量符合输入规范 |
| 后置条件 | 修改城市距离、价格常量，持久化更新涉及的对象的数据 |
| **需要的服务（需接口）** | | |
| **服务名** | 服务 | |
| **BaseDataService.**  **set(BasePO po)** | 更改单一化持久对象 | |

**Accountbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Account.show** | 语法 | public ArrayList<AccountVO> show() |
| 前置条件 | 已经创建一个Account领域对象 |
| 后置条件 | 展示所有用户ID |
| **服务名** | 服务 | |
| Account**DataService.**  **init** | 初始化持久数据 | |

**Receiptbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Receipt.getID** | 语法 | public String getID () |
| 前置条件 | 无 |
| 后置条件 | 返回新建的ID |
| **Receipt.addReceipt** | 语法 | PublicResultMessage addReceipt  (String receiptType,ReceiptVO receipt) |
| 前置条件 | 单据类型、内容输入合法 |
| 后置条件 | 添加新的单据，持久化更新涉及的对象的数据 |
| **Receipt.deleteReceipt** | 语法 | public ResultMessage deleteReceipt (  String receiptType,String ReceiptId) |
| 前置条件 | 单据类型、ID输入合法 |
| 后置条件 | 删除单据，持久化更新涉及的对象的数据 |
| **Receipt.modifyReceipt** | 语法 | public ResultMessage modifyReceipt  (String receiptType,ReceiptVO receipt) |
| 前置条件 | 单据类型、内容输入合法 |
| 后置条件 | 更改单据，持久化更新涉及的对象的数据 |
| **Receipt.findReceipt** | 语法 | public ResultMessage findReceipt  (String receiptType, intReceiptId) |
| 前置条件 | 单据类型、ID输入合法 |
| 后置条件 | 查找到该单据 |
| **需要的服务（需接口）** | | |
| **服务名** | 服务 | |
| **ReceiptDataService.**  **getID()** | 得到新建单据的ID | |
| **ReceiptDataService.**  **findReceipt (int id)** | 根据ID进行查找单一持久化对象 | |
| **ReceiptDataService.**  **addReceipt(ReceiptPO po)** | 添加单一化持久对象 | |
| **ReceiptDataService.**  **addOpeningStock (OpeningStockPO po)** | 添加单一化持久对象 | |
| **ReceiptDataService.**  **deleteReceipt (ReceiptPO po )** | 删除单一化持久对象 | |
| **ReceiptDataService.**  **modifyReceipt (ReceiptPO po)** | 更改单一化持久对象 | |
| **ReceiptDataService.**  **getReceiptData** | 得到Receipt数据的服务引用 | |

**Userbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| **User.login** | 语法 | public ResultMessage login(String username, String password); |
| 前置条件 | password符合输入规则 |
| 后置条件 | 查找是否存在相应的User，根据输入的password返回登录验证的结果 |
| **User.addUser** | 语法 | public ResultMessage addUser  (String username, String password, UserIdentity position) |
| 前置条件 | 添加的User不重复 |
| 后置条件 | 添加User |
| **User.deletUser** | 语法 | public ResultMessage deleteUser  (String username) |
| 前置条件 | id符合输入规则 |
| 后置条件 | 查找相应的User，根据输入的id删除该用户 |
| **User.updateUser** | 语法 | public ResultMessage updateUser(String username, String password, UserIdentity newPosition, UserVO user) |
| 前置条件 | 用户名、密码、职位符合输入规则 |
| 后置条件 | 查找相应的User，根据输入的id，更改该用户的信息 |
| 需要的服务（需接口） | | |
| **服务名** | 服务 | |
| **DatabaseFactory.getUserDataBase** | 得到User的服务的引用 | |
| **UserDateService.find**  **(UserPO po)** | 查找单一持久化对象 | |
| **UserDataService.insert**  **(UserPO po)** | 插入单一持久化对象 | |
| **UserDataService.delete**  **(UserPO po)** | 删除单一持久化对象 | |
| **UserDataService.update**  **(UserPO po)** | 更新单一持久化对象 | |

## 数据层的分解

数据层主要给业务逻辑层提供数据访问服务，包括对于持久化数据的增、删、改、查。持久化数据保存采用序列化文件

### 数据层模块的职责

数据层模块职责如下表所示

**数据层模块职责**

|  |  |
| --- | --- |
| **模块** | **职责** |
| OrderDataService | 提供订单数据的增加、查找、修改服务 |
| FacilityDataService | 提供营业厅运输工具数据的增加、查找、修改、删除服务 |
| BranchDataService | 提供营业厅运输事件数据的增加、查找、修改、删除服务 |
| TransferDataService | 提供中转中心运输事件数据的增加、查找、修改、删除服务 |
| InventoryDataService | 提供仓库数据的增加、查找、修改、删除服务 |
| FundDataService | 提供资金数据的增加、查找、修改、删除服务 |
| OpeningstockDataService | 提供期初建账数据集体载入、保存、增加、查看服务 |
| RecordDataService | 提供成本收益表、经营情况表生成服务 |
| BaseDataService | 提供公司基本数据常量的增加、查找、修改、删除服务 |
| AccountDataService | 提供公司员工数据的增加、查找、修改、删除服务 |
| ReceiptDataService | 提供单据数据审批服务 |
| UserDataService | 提供系统用户数据的增加、查找、修改、删除服务 |

### 数据层模块的接口规范

数据层模块的接口规范如下表

**数据层模块的接口规范**

OrderDataService

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| **OrderDataService.insert** | 语法 | public ResultMessage insert(OrderPO po) |
| 前置条件 | 同样的po在数据中不存在 |
| 后置条件 | 增加一条po数据 |
| **OrderDataService.find** | 语法 | public UserPO find(String orderNumber) |
| 前置条件 | 无 |
| 后置条件 | 查找返回相应的结果 |
| **OrderDataService.update** | 语法 | public ResultMessageupdate (OrderPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po |
| **OrderDataService.init** | 语法 | public void init() |
| 前置条件 | 无 |
| 后置条件 | 初始化持久数据 |

FacilityDataService接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **FacilityDataService.add** | 语法 | public ResultMessage add  (FacilityPO po) |
| 前置条件 | 同样的po在数据中不存在 |
| 后置条件 | 增加一条po数据 |
| **FacilityDataService.find** | 语法 | public FacilityPO find(String name) |
| 前置条件 | 无 |
| 后置条件 | 查找返回相应的结果 |
| **FacilityDataService.delete** | 语法 | public ResultMessage delete  (String id) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 删除po |
| **FacilityDataService.modify** | 语法 | public ResultMessage modify  (FacilityPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 修改po |
| **FacilityDataService.init** | 语法 | public void init() |
| 前置条件 | 无 |
| 后置条件 | 初始化持久数据 |
| **FacilityDataService. getID** | 语法 | public void getID () |
| 前置条件 | 无 |
| 后置条件 | 初始化持久数据 |
| **FacilityDataService. show** | 语法 | public ArrayList< FacilityPO >  show () |
| 前置条件 | 无 |
| 后置条件 | 返回所有车辆信息 |

BranchDataService

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| **BranchDataService.insert** | 语法 | public ResultMessageinsert(BranchPO po) |
| 前置条件 | 同样的po在数据中不存在 |
| 后置条件 | 增加一条po数据 |
| **BranchDataService.find** | 语法 | public BranchPO find(String id) |
| 前置条件 | 无 |
| 后置条件 | 查找返回相应的结果 |
| **BranchDataService.update** | 语法 | public ResultMessageupdate (BranchPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po |
| **BranchDataService.init** | 语法 | public void init() |
| 前置条件 | 无 |
| 后置条件 | 初始化持久数据 |

TransferDataService

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| **TransferDataService.insert** | 语法 | public ResultMessageinsert(TransferPO po) |
| 前置条件 | 同样的po在数据中不存在 |
| 后置条件 | 增加一条po数据 |
| **TransferDataService.find** | 语法 | public TransferPO find(String id) |
| 前置条件 | 无 |
| 后置条件 | 查找返回相应的结果 |
| **TransferDataService.update** | 语法 | public ResultMessageupdate (TransferPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po |
| **TransferDataService.init** | 语法 | public void init() |
| 前置条件 | 无 |
| 后置条件 | 初始化持久数据 |

|  |  |  |
| --- | --- | --- |
|  | **后置条件** | **返回所有单据** |

InventoryDataService接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **InventoryDataService.init** | 语法 | public void init() |
| 前置条件 | 无 |
| 后置条件 | 初始化持久数据 |
| **InventoryDataService.**  **getImportID** | 语法 | public String getImportID () |
| 前置条件 | 无 |
| 后置条件 | 得到入库单ID |
| **InventoryDataService.**  **getExportID** | 语法 | public String getExportID () |
| 前置条件 | 无 |
| 后置条件 | 得到出库单ID |
| **InventoryDataService.**  **getAlarmID** | 语法 | public String getAlarmID () |
| 前置条件 | 无 |
| 后置条件 | 得到调整单ID |
| **InventoryDataService.getA()** | 语法 | public String getA() |
| 前置条件 | 无 |
| 后置条件 | 得到区号 |
| **InventoryDataService.getB()** | 语法 | public String getB() |
| 前置条件 | 无 |
| 后置条件 | 得到排号 |
| **InventoryDataService.getC()** | 语法 | public String getC() |
| 前置条件 | 无 |
| 后置条件 | 得到架号 |
| **InventoryDataService.getD()** | 语法 | public String getD() |
| 前置条件 | 无 |
| 后置条件 | 得到位号 |
| **InventoryDataService.getASize()** | 语法 | public String getASize() |
| 前置条件 | 无 |
| 后置条件 | 得到区容量 |
| **InventoryDataService.getBSize()** | 语法 | public String getBSize() |
| 前置条件 | 无 |
| 后置条件 | 得到排号容量 |
| **InventoryDataService.getCSize()** | 语法 | public String getCSize() |
| 前置条件 | 无 |
| 后置条件 | 得到架号容量 |
| **InventoryDataService.getDSize()** | 语法 | public String getDSize() |
| 前置条件 | 无 |
| 后置条件 | 得到位号容量 |
| **InventoryDataService.getANum()** | 语法 | public String getANum() |
| 前置条件 | 无 |
| 后置条件 | 得到区此时容量 |
| **InventoryDataService.getBNum()** | 语法 | public String getBNum() |
| 前置条件 | 无 |
| 后置条件 | 得到排此时容量 |
| **InventoryDataService.getCNum()** | 语法 | public String getCNum() |
| 前置条件 | 无 |
| 后置条件 | 得到架此时容量 |
| **InventoryDataService.getDNum()** | 语法 | public String getDNum() |
| 前置条件 | 无 |
| 后置条件 | 得到位此时容量 |
| **InventoryDataService.showImport()** | 语法 | public ArrayList<InventoryImportReceiptPO> showImport() |
| 前置条件 | 无 |
| 后置条件 | 返回所有入库单据的数据 |
| **InventoryDataService.showExport()** | 语法 | public ArrayList<InventoryExportReceiptPO> showExport() |
| 前置条件 | 无 |
| 后置条件 | 返回所有出库单据的数据 |
| **InventoryDataService.showAdjust()** | 语法 | public ArrayList<AdjustReceiptPO> showAdjust() |
| 前置条件 | 无 |
| 后置条件 | 返回所有调整单据的数据 |
| **InventoryDataService.insertImport(InventoryImportReceiptPO po)** | 语法 | public ResultMessage insertImport(InventoryImportReceiptPO po) |
| 前置条件 | 无 |
| 后置条件 | 增加一条po数据即生成入库单据 |
| **InventoryDataService.insertExport(InventoryExportReceiptPO po)** | 语法 | public ResultMessage insertExport(InventoryExportReceiptPO po) |
| 前置条件 | 无 |
| 后置条件 | 增加一条po数据即生成出库单据 |
| **InventoryDataService.insertAdjust(AdjustReceiptPO po)** | 语法 | public ResultMessage insertAdjust(AdjustReceiptPO po) |
| 前置条件 | 无 |
| 后置条件 | 增加一条po数据即生成调整单据 |

FundDataService接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **FundDataService.add** | 语法 | public ResultMessage add(ReceiptPO po) |
| 前置条件 | 同样的po在数据中不存在 |
| 后置条件 | 增加一条po数据 |
| **FundDataService.find** | 语法 | public ResultMessage find(String id) |
| 前置条件 | 无 |
| 后置条件 | 查找返回相应的结果 |
| **FundDataService.delete** | 语法 | public ResultMessage delete (ReceiptPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 删除po |
| **FundDataService.modify** | 语法 | public ResultMessage update (ReceiptPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po |
| **FundDataService.init** | 语法 | public void init() |
| 前置条件 | 无 |
| 后置条件 | 初始化持久数据 |

OpeningStockDataService接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| **OpeningStockDataService.add** | 语法 | public ResultMessage add(OpeningStockPO po) |
| 前置条件 | 同样的po在数据中不存在 |
| 后置条件 | 增加一条po数据 |
| **OpeningStockDataService.init** | 语法 | public void init() |
| 前置条件 | 无 |
| 后置条件 | 初始化持久数据 |
| **OpeningStockDataService.find** | 语法 | public OpeningStockPOfind(String id) |
| 前置条件 | 无 |
| 后置条件 | 返回OpeningStockPO持久数据 |

RecordDataService接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| **RecordDataService.getBusinessProcess** | 语法 | publicArrayList<BussinessProcessPO> getBusinessProcess(String begindate,String enddate) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 返回1. 收款单2.付款单 |
| **RecordDataService.getBusinessCondition** | 语法 | publicArrayList<BussinessConditionPO> getBusinessCondition(String enddate) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 返回信息： 1. 收入2. 支出。3. 利润：总收入-总支出 |

BaseDataService接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| **BaseDataService.set** | 语法 | public ResultMessage set(BasePO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 修改po数据 |

AccountDataService接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| **AccountDataService.delete** | 语法 | public void init() |
| 前置条件 | 无 |
| 后置条件 | 初始化持久数据 |

ReceiptDataService接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **ReceiptDataService.add** | 语法 | public ResultMessage add  (ReceiptPO po) |
| 前置条件 | 同样的po在数据中不存在 |
| 后置条件 | 增加一条po数据 |
| **ReceiptDataService.find** | 语法 | public ReceiptPO find(String name) |
| 前置条件 | 无 |
| 后置条件 | 查找返回相应的结果 |
| **ReceiptDataService.delete** | 语法 | public ResultMessage delete  (String id) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 删除po |
| **ReceiptDataService.modify** | 语法 | public ResultMessage modify  (ReceiptPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 修改po |
| **ReceiptDataService.init** | 语法 | public void init() |
| 前置条件 | 无 |
| 后置条件 | 初始化持久数据 |
| **ReceiptDataService. getID** | 语法 | public void getID () |
| 前置条件 | 无 |
| 后置条件 | 初始化持久数据 |
| **ReceiptDataService. show** | 语法 | public ArrayList< ReceiptPO >  show () |
| 前置条件 | 无 |

UserDataService接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| **UserDataService.insert** | 语法 | public ResultMessageinsert(UserPO po) |
| 前置条件 | 同样的po在数据中不存在 |
| 后置条件 | 增加一条po数据 |
| **UserDataService.find** | 语法 | public UserPO find(String id) |
| 前置条件 | 无 |
| 后置条件 | 查找返回相应的结果 |
| **UserDataService.delete** | 语法 | public ResultMessagedelete (String username) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 删除po |
| **UserDataService.update** | 语法 | public ResultMessageupdate (UserPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po |
| **UserDataService.init** | 语法 | public void init() |
| 前置条件 | 无 |
| 后置条件 | 初始化持久数据 |

# 6. 信息视角

## 6.1 数据持久化对象

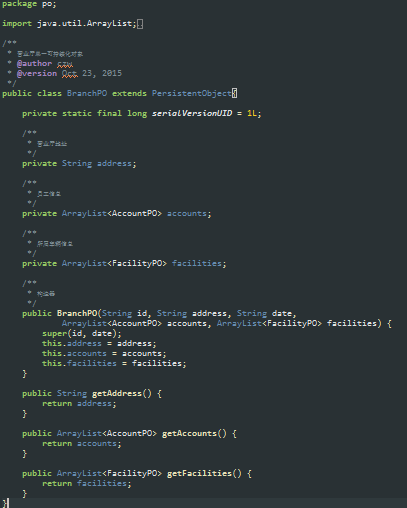
系统的PO类就是对应的相关的实体类，如下所示。

* PersistentObject: 所有PO类的父类，包括时间、编号，时间格式必须为yyyy-mm-dd。
* UserPO: 系统用户的PO类，包括账号、密码、职务权限、操作记录
* AccontPO: 公司职员的PO类，包括员工编号、职务权限、姓名、出生日期、身份证号、手机号、工资情况、任职时间、每日订单完成情况。
* CommodityPO: 货物的PO类，包括货物种类、商品重量、商品体积、到达状态（完整、损坏或丢失）。
* OrderPO: 订单的PO类，包括订单号、收件人（寄件人）姓名、收件人（寄件人）住址、收件人（寄件人）电话、货物集合信息、订单运送途径地点、派件时间、收件时间、费用，费用格式必须是大于等于0、精确到小数点后1为的浮点数，单位是元。
* FacilityPO: 运输工具的PO类，包括编号、监管员编号、运输历史、购入时间，编号格式为营业厅编号+日期(20150921)+五位数字编码(00000)。
* InventoryPO: 仓库的PO类，包括仓库编号、库存总容量、已占用库存数量。
* BranchPO: 营业厅的PO类，包括编号、地址、成立时间、员工信息、所属车辆信息、收款信息，营业厅编号格式为城市编码(000)+营业厅(000)。
* TransferPO: 中转中心的PO类，包括编号、地址、成立时间、员工信息、仓库信息，中转中心编号格式为城市编码(000)+中转中心(0)。
* BankPO: 银行账户的PO类，包括账户名称、账户属性、创建时间、余额。
* ReceiptPO: 所有单据的父类，包括编号、当前状态、创建时间（子类省去这些属性）。

1. LoadingList：装车单，营业厅编号、汽运编号、目的地、车辆代号、监装员、押运员，所有订单订单号
2. BranchArrivalList：营业厅到达单，中转单编号、出发地、货物到达状态、订单订单号
3. DeliveryList：派件单，订单订单号、派送员
4. DebitNote：收款单，（快递员编号）、收款金额
5. TransferArrivalList：中转中心到达单，中转中心编号、出发地、货物到达状态、订单订单号
6. TransferOrder：中转单，航班（火车）（汽车车次）编号、出发地、目的地、监装员、托运订单订单号
7. InventoryExportReceipt：库存入库单，订单编号、目的地、仓库区号、排号、架号、位号
8. InventoryImportReceipt：库存出库单，目的地、装运形式、中转单编号
9. PaymentBill：付款单，付款金额、付款人姓名、付款账号、条目、备注
10. AdjustReceipt：仓库调整单。记录了调整前后的区，排，架位号

* OpeningStockPO:期初账目的PO类，包括机构、人员、车辆、库存、银行账户信息。
* BasePO：常量的PO类，包括城市距离、价格等常量。
* BussinessConditionPO :经营情况po 包括收款单付款单列表
* BussinessProcessPO :成本收益po 包括总收入总支出总利润

持久化营业厅对象BranchPO的定义如下图所示。



## 6.2 数据格式

统一用\*.ser格式，比如order.ser