体系结构文档

物流信息管理系统

更新历史：

|  |  |  |  |
| --- | --- | --- | --- |
| 修改人员 | 日期 | 修改原因 | 版本号 |
| 倪辰皓 | 2015/10/26 | 完成文档 | V1.0 |
| 全体人员 | 2015/11/6 | 修改了模块之间的业务依赖关系 | V1.1 |
| 倪辰皓 | 2015/11/15 | 更新了开发包图 | V1.2 |

目录

[体系结构文档 1](#_Toc433654926)

[1.引言 3](#_Toc433654927)

[1.1编制目的 3](#_Toc433654928)

[2.产品概述 3](#_Toc433654929)

[3.逻辑视角 3](#_Toc433654930)

[4.组合视角 4](#_Toc433654931)

[4.1开发包图 4](#_Toc433654932)

[4.2运行时进程 7](#_Toc433654933)

[4.3物理部署 7](#_Toc433654934)

[5.接口视角 8](#_Toc433654935)

[5.1模块的职责 8](#_Toc433654936)

[5.2 用户界面层的分解 10](#_Toc433654937)

[5.2.1 用户界面层模块的职责 10](#_Toc433654938)

[5.2.2 用户界面层模块的接口规范 12](#_Toc433654939)

[5.2.3 用户界面模块设计原理 12](#_Toc433654940)

[5.3业务逻辑层的分解 12](#_Toc433654941)

[5.3.1业务逻辑层模块的职责 12](#_Toc433654942)

[5.3.2业务逻辑模块的接口规范 12](#_Toc433654943)

[5.4数据层的分解 18](#_Toc433654944)

[5.4.1数据层模块的职责 18](#_Toc433654945)

[5.4.2数据层模块的接口规范 18](#_Toc433654946)

[6.信息视角 21](#_Toc433654947)

[6.1数据持久化对象 21](#_Toc433654948)

[6.2文件格式 22](#_Toc433654949)

# 1.引言

## 1.1编制目的

本报告详细完成对物流信息管理系统的概要设计，达到指导详细设计和开发的目的，同

时实现和开发人员及用户的沟通。

本报告面向开发人员、测试人员及最终用户而编写，是了解系统的导航。

# 2.产品概述

参考物流信息管理系统用例文档和物流信息管理系统软件需求规格说明中对产品的概

括描述。

# 3.逻辑视角

物流信息管理系统中，选择了分层体系结构风格，将系统分为3层（展示层、业务逻辑

层、数据层）能够很好地示意整个高层抽象。展示层包含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访问。

物流信息管理系统的主要功能有：物流信息查询、订单输入、报价和时间管理、收件信息输入、车辆装车管理、接收与派件、收款单填写、车辆信息管理、司机信息管理、飞机装运管理、火车装运管理、汽车装运管理、中转接收、出库、入库、库区调整、库存信息初始化、库存盘点、结算管理、财务支出、成本收益表查询、经营情况表查询、账户管理、人员与机构管理、审判单据、用户管理

可以合并为：

物流信息查询--信息查询

订单输入、报价和时间管理--寄件

审判单据--单据审判

车辆装车管理、飞机装运管理、火车装运管理、汽车装运管理、中转接收--装运

收件信息输入、接收与派件--收件

出库、入库、库区调整、库存信息初始化、库存盘点--库存管理

账户管理、收款单填写、结算管理、财务支出--财务

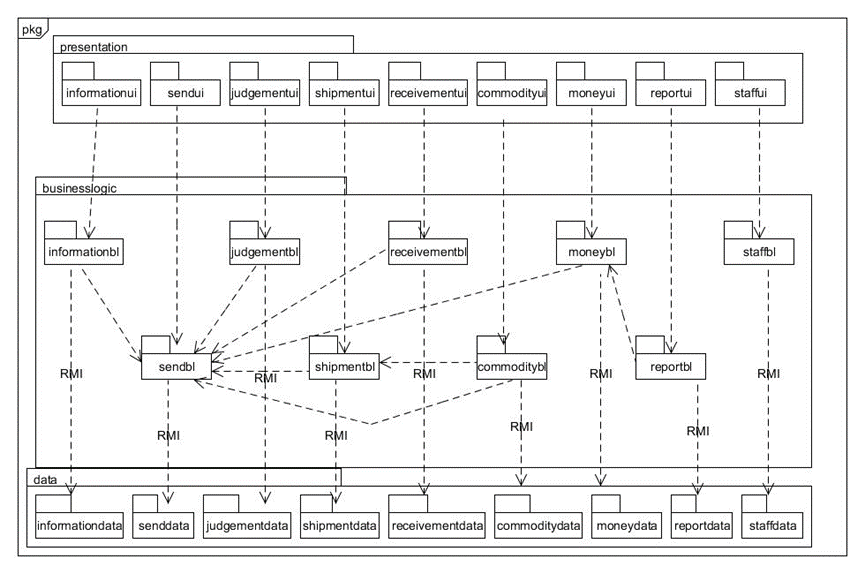
成本收益表查询、经营情况表查询—报表查询

人员与机构管理、车辆信息管理、司机信息管理、用户管理--人员管理

**物流信息管理系统概要功能设计**

|  |  |
| --- | --- |
| **功能** | **对应逻辑包** |
| 信息查询 | informationui，informationbl，informationdata  sendbl,senddata  shipmentbl,shipmentdata  receivementbl, receivementdata |
| 寄件 | sendui,sendbl,senddata |
| 单据审判 | judgementui, judgementbl  sendbl,senddata  shipmentbl, shipmentdata  receivementbl, receivementdata  moneybl,moneydata |
| 装运 | shipmentui,shipmentbl,shipmentdata |
| 收件 | receivementui, receivementbl, receivementdata |
| 库存管理 | commodityui, commoditybl, commoditydata |
| 财务 | moneyui,moneybl,moneydata  sendbl,senddata |
| 报表查询 | reportui,reportbl,reportdata  moneybl,moneydata |
| 人员管理 | staffui,staffbl,staffdata |

**软件体系结构设计方案**



# 4.组合视角

## 4.1开发包图

物流系统的最终开发包设计如表1所示

表1物流系统的最终开发包设计

|  |  |
| --- | --- |
| 开发（物理包） | 依赖的其他开发包 |
| mainui | informationui,sendui,judgementui,shipmentui,receivementui,commodityui,moneyui,reportui,staffui,vo |
| informationui | informationblservice,界面库类包 |
| informationblservice |  |
| informationbl | informationblservice,sendbl,po, informationdataservice |
| informationdataservice | Java RMI,po |
| informationdata | informationdataservice, databaseutility |
| sendui | sendblservice,界面类库包 |
| sendblservice |  |
| sendbl | sendblservice,po, senddataservice |
| senddataservice | Java RMI,po |
| senddata | senddataservice,databaseutility |
| judgementui | judgementblservice,界面类库包 |
| judgementblservice |  |
| judgementbl | judgementblservice, sendbl ,po, judgementdataservice |
| judgementdataservice | Java RMI,po |
| judgementdata | judgementdataservice, databaseutility |
| shipmentui | shipmentblservice,界面类库包 |
| shipmentblservice |  |
| shipmentbl | shipmentblservice, sendbl ,po, shipmentdataservice |
| shipmentdataservice | Java RMI,po |
| shipmentdata | shipmentdataservice, databaseutility |
| receivementui | receivementblservice,界面类库包 |
| receivementblservice |  |
| receivementbl | receivementblservice，sendbl ,po, receivementdata |
| receivementdataservice | Java RMI,po |
| receivementdata | receivementdataservice, databaseutility |
| commodityui | commodityblservice,界面类库包 |
| commodityblservice |  |
| commoditybl | commodityblservice,sendbl ,shipmentbl,po,commoditydataservice |
| commoditydataservice | Java RMI,po |
| commoditydata | commoditydataservice, databaseutility |
| moneyui | moneyblservice,界面类库包 |
| moneyblservice |  |
| moneybl | moneyblservice,sendbl ,po, moneydataservice |
| moneydataservice | Java RMI,po |
| moneydata | moneydataservice, databaseutility |
| reportui | reportblservice,界面类库包 |
| reportblservice |  |
| reportbl | reportblservice，moneybl,po, reportdataservice |
| reportdataservice | Java RMI,po |
| reportdata | reportdataservice, databaseutility |
| staffui | staffblservice,界面类库包 |
| staffblservice |  |
| staffbl | staffblservice,po, staffdataservice |
| staffdataservice | Java RMI,po |
| staffdata | staffdataservice, databaseutility |
| vo |  |
| po |  |
| utilitybl |  |
| 界面类库包 |  |
| Java RMI |  |
| databaseutility |  |

物流系统客户端开发包图如图3所示，服务器端开发包图如图4所示。

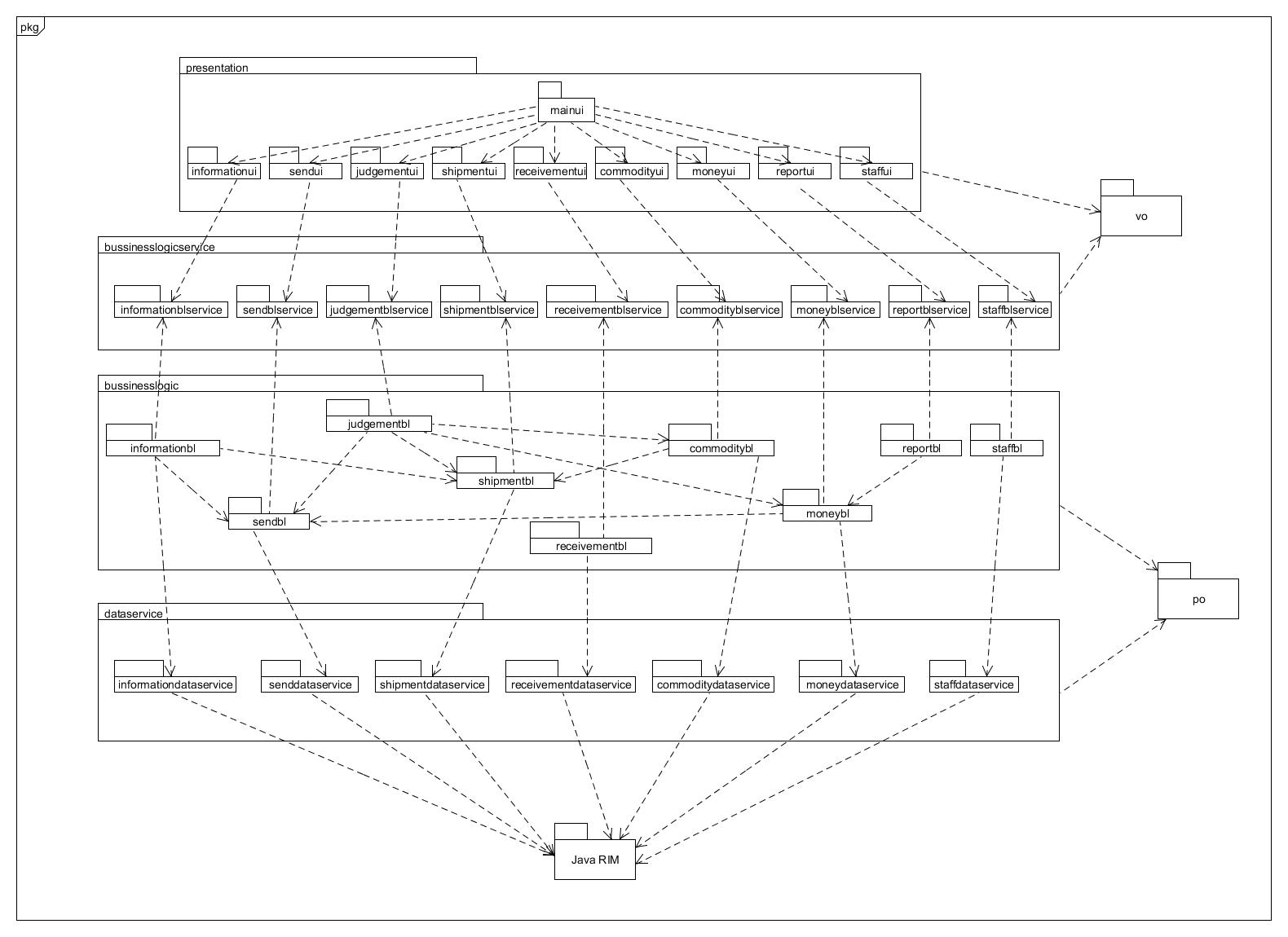


图3 快递系统客户端开发包图

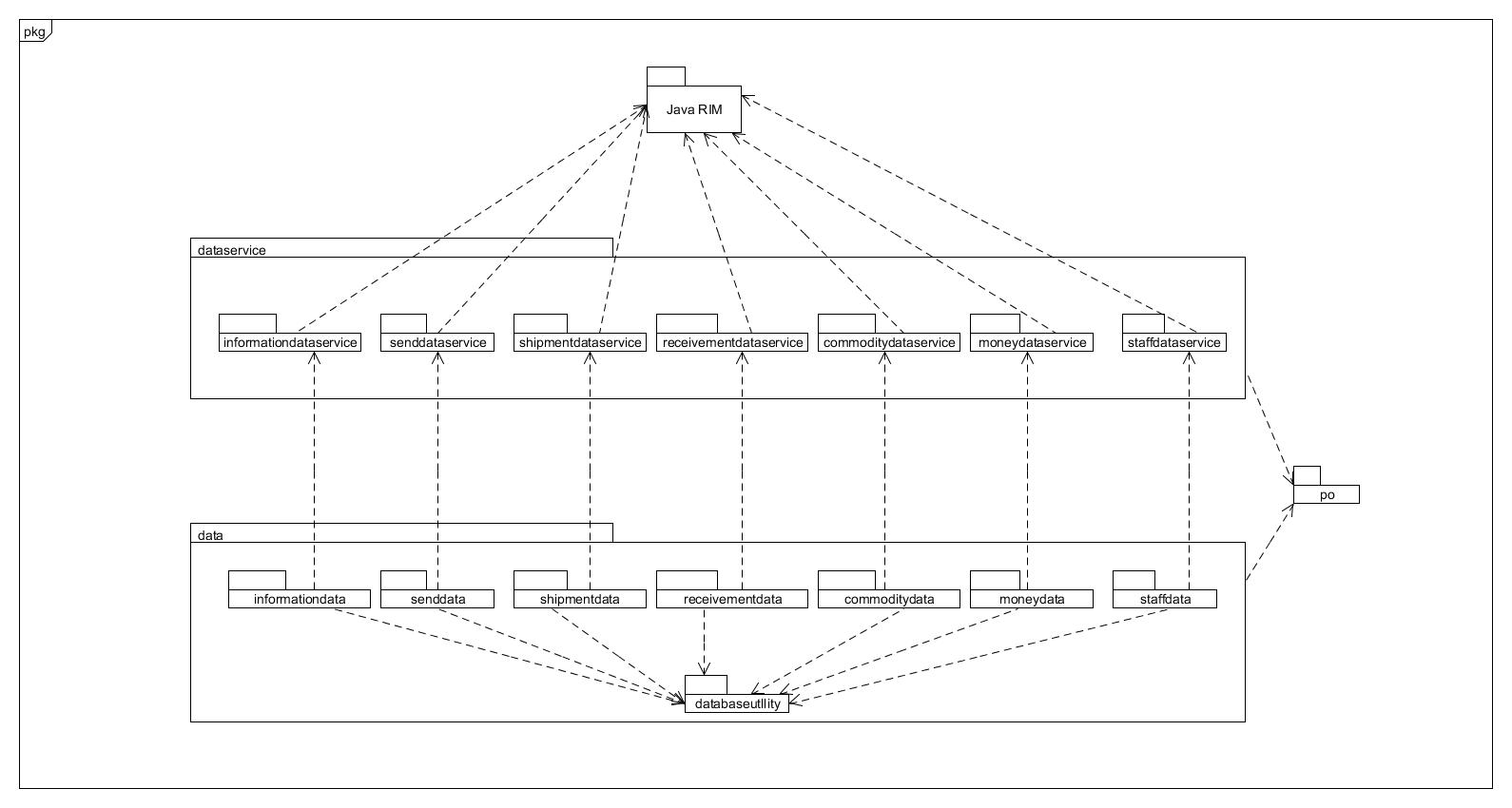


图4 物流系统服务器端开发包图

## 4.2运行时进程

在快递系统中，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。

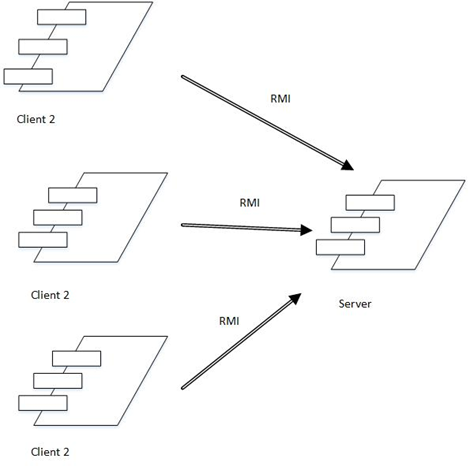


图5 进程图

## 4.3物理部署

物流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。

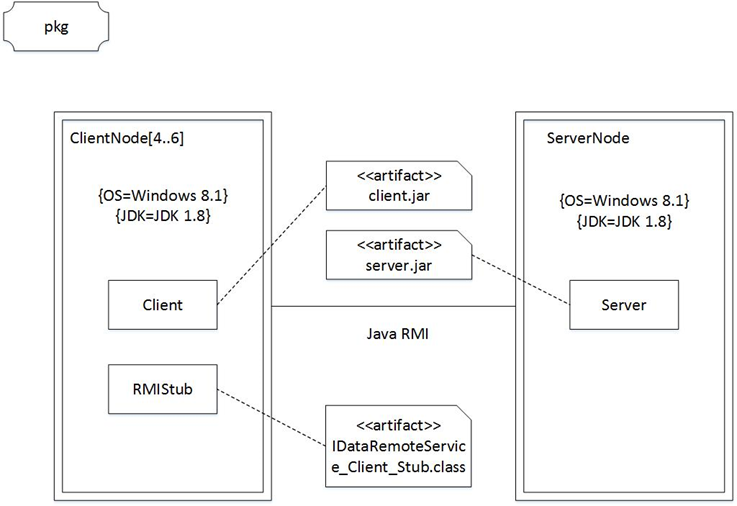


图6 部署图

# 5.接口视角

## 5.1模块的职责

客户端模块和服务器端模块视图分别如图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

展示层

业务逻辑层

网络模块

图7 客户端模块视图

启动模块

网络模块

数据层

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 | 基于窗口的物流公司客户端用户界面 |
| 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JavaRMI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JavaRMI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** |
| informationBLService  sendBLService  judgementBLService  shipmentBLService  receivementBLService  commodityBLService  moneyBLService  reportBLService  staffBLService | 客户端展示层 | 客户端业务逻辑层 |
| informationDataService  sendDataService  judgementDataService  shipmentDataService  receivementDataService  commodityDataService  moneyDataService  reportDataService  staffDataService | 客户端业务逻辑层 | 服务器端数据层 |

以下图9是寄件用例层之间的调用。每一层之间都是由上层依赖了一个接口（需接口），而下层实现这个接口（供接口）。sendblservice提供了send界面所需要的所有业务逻辑功能。senddataservice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。

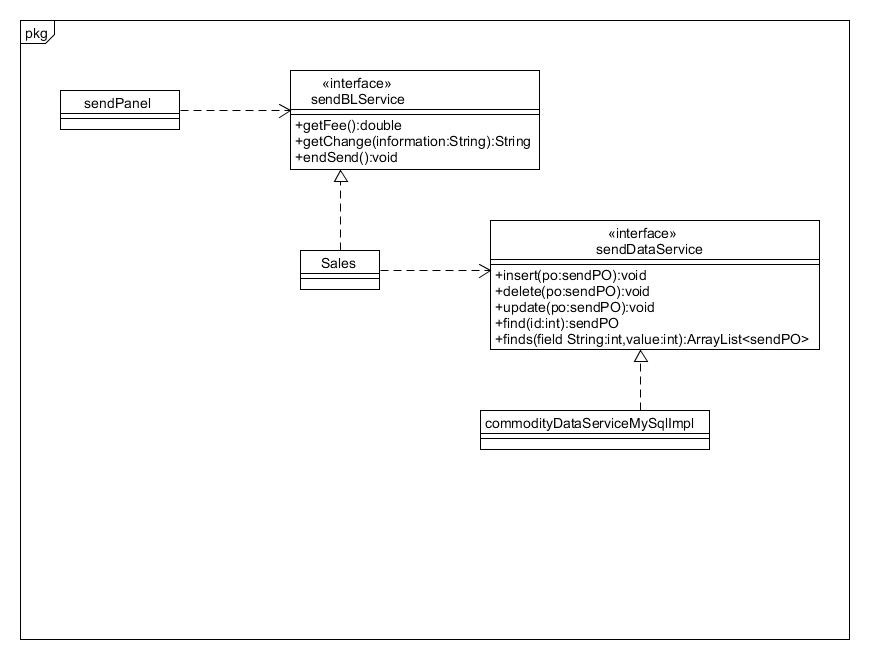


图9 寄件用例层之间调用的接口

## 5.2 用户界面层的分解

根据需求，系统存在26个用户界面：登录界面、寄件人查询界面、物流信息界面、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、营业厅业务员主界面、装车信息管理界面、车辆信息管理界面、司机信息管理界面、中转中心业务员主界面、货物装运信息管理界面、中转中心仓库管理员主界面、仓库信息管理界面、库存信息初始化界面、财务人员主界面、结算信息管理界面、统计报表界面、账户信息管理界面、总经理主界面、人员和机构信息管理界面、审判单据界面、查看统计分析界面、管理员主界面、调整用户界面。界面跳转如图10所示。

所有的用户界面通过同一个主用户界面类（MainFrame)处理，如图11。

### 5.2.1 用户界面层模块的职责

**用户界面层模块的职责**

|  |  |
| --- | --- |
| 模块 | 职责 |
| MainFrame | 界面Frame，负责界面的显示和界面的跳转 |

调整用户界面

管理员主界面

查看统计分析界面

单据审判界面

人员和机构信息管理界面

总经理主界面

账户信息管理界面

统计报表界 面

结算信息管理界面

财务人员主界面

库存信息初始化界面

仓库信息管理界面

中转中心仓库管理员主界面

货物装运信息界面

中转中心业务员主界面

司机信息管理界面

车辆信息管理界面

装车信息管理界面

营业厅业务员主界面

收件信息输入界面

报价和时间管理界面

订单输入

界面

物流信息界面

快递员主界面

寄件人查询

界面

登录界面

图10 界面跳转

pkg

MainFrame

+int():void

图11 用户界面类

### 5.2.2 用户界面层模块的接口规范

**用户界面层模块的接口规范**

|  |  |  |
| --- | --- | --- |
|  | 语法 | int(args:String[]) |
| **MainFrame** | 前置条件 | 无 |
|  | 后置条件 | 显示Frame以及LoginPanel |

**用户界面层模块需要的服务接口**

|  |  |
| --- | --- |
| **服务名** | **服务** |
| businesslogicservice.LoginBLService | 登录界面的业务逻辑接口 |
| Businesslogicservice.\*BLService | 每个界面都有一个相应的业务逻辑接口 |

### 5.2.3 用户界面模块设计原理

用户界面利用Java的Swing和AWT库来实现。

## 5.3业务逻辑层的分解

### 5.3.1业务逻辑层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| imformationbl | 负责实现信息查询界面需要的服务 |
| sendbl | 负责实现寄件界面需要的服务 |
| judgementbl | 负责实现单据审判界面需要的服务 |
| shipmentbl | 负责实现装运界面需要的服务 |
| receivementbl | 负责实现收件界面 |
| commoditybl | 负责实现库存管理界面需要的服务 |
| moneybl | 负责实现财务管理界面需要的服务 |
| reportbl | 负责实现报表查询界面需要的服务 |
| staffbl | 负责实现人员管理界面需要的服务 |

### 5.3.2业务逻辑模块的接口规范

**commoditybl模块的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的接口（供接口） | | | |
|  | 语法 | | Public ResultMessage addcommo(String id,String type,int numOfPosition) |
| commodity.addcommo | 前置条件 | | 货物编号、货物类型、库区位置输入合法 |
|  | 后置条件 | | 将货物添加到仓库，持久化更新涉及的对象的信息 |
|  | 语法 | | Public ResultMessage deletecommo(String id) |
| commodity.deletecommo | 前置条件 | | 货物编号输入合法，且货物信息没被操作过 |
|  | 后置条件 | | 将货物从仓库中删除，持久化更新涉及的对象的信息 |
|  | 语法 | | Public ResultMessage updatecommo(String id,String type,int numOfPosition) |
| commodity.updatecommo | 前置条件 | | 货物编号、货物类型、库区位置输入合法 |
|  | 后置条件 | | 库存更新，持久化更新涉及的对象的信息 |
|  | 语法 | | Public ResultMessage findcommo(String id) |
| commodity.findcommo | 前置条件 | | 货物编号输入合法 |
|  | 后置条件 | | 查找到该货物的信息 |
| Commodity.getList | 语法 | | Public Resultmessage getList() |
| 前置条件 | | 无 |
| 后置条件 | | 系统清空未审批单据空间 |
| 需要的接口（需接口） | | | |
| 服务名 | | 服务 | |
| Commoditydataservice.insert(CommodityPO po) | | 添加单一持久化对象 | |
| Commoditydataservice.delete(CommodityPO po) | | 删除单一持久化对象 | |
| Commoditydataservice.update（CommodityPO po) | | 更新单一持久化对象 | |
| Commoditydataservice.find(String id) | | 根据编号查找相应持久化对象 | |
| Commoditydataservice.getCommoditydata | | 得到commodity数据的服务引用 | |

**Imformationbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Imformationbl.inquireTransMesg | 语法 | public resultmessage inquireTransMesg(String id) |
| 前置条件 | 寄件人被识别和授权 |
| 后置条件 | 无 |
| Imformationbl.refreshMesg | 语法 | Public resultmessage refreshMesg(String id,String position,String time) |
| 前置条件 | 货物接收成功 |
| 后置条件 | 系统储存物流信息 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| Imformationdata.getTransMesg | 返回单一持久化对象，得到物流信息 | |
| Sendbl.inquireSendMesg | 返回单一持久化对象，得到快递信息 | |

**moneybl模块的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供服务） | | | |
|  | 语法 | | Public String getPayid() |
| Money.getPayid | 前置条件 | | 系统产生付款记录 |
|  | 后置条件 | | 返回新建的付款单id |
|  | 语法 | | Public String getExpenseid() |
| Money.getExpenseid | 前置条件 | | 系统产生收款记录 |
|  | 后置条件 | | 返回新建的收款单id |
|  | 语法 | | Public MoneyVO find(String id) |
| Money.find | 前置条件 | | 财务人员输入合法的账单ID |
|  | 后置条件 | | 返回根据ID找到的单据 |
|  | 语法 | | Public ResultMessage getSumMessage() |
| Money.getSumMoney | 前置条件 | | 已得到所有收款和付款数据 |
|  | 后置条件 | | 得到总金额 |
|  | 语法 | | Public MoneyVO submit() |
| Money.submit | 前置条件 | | 单据已被制定出 |
|  | 后置条件 | | 将单据提交审批 |
|  | 语法 | | Public MoneyVO save() |
| Money.save | 前置条件 | | 单据已被制定出 |
|  | 后置条件 | | 单据被保存在payment和expense文件中 |
|  | 语法 | | Public ResultMessage getMember(String name，double bill) |
| Money.getMember | 前置条件 | | 账户已在系统中注册账号 |
|  | 后置条件 | | 得到账户信息（名称，交易金额） |
| 需要的服务（需服务） | | | |
| 服务名 | | 服务 | |
| Moneydataservice.insert(MoneyPO po) | | 添加单一持久化对象 | |
| Moneydataservice.delete(String id) | | 删除单一持久化对象 | |
| Moneydataservice.update(MoneyPO po) | | 更新单一持久化对象 | |
| Moneydataservice.find(String id) | | 根据编号查找相应持久化对象 | |
| Moneydataservice.init() | | 初始化单一持久化对象 | |
| Moneydataservice.insertMember（String name) | | 添加单一持久化账户对象 | |
| Moneydataservice.deleteMember（String name) | | 删除单一持久化账户对象 | |
| Moneydataservice.findMember（String name) | | 查找单一持久化账户对象 | |

**receivementbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
|  | 提供的服务（供接口） |  |
|  | 语法 | Public ResultMessage addCommodity(String id) |
| Receivement.addcommodity | 前置条件 | 启动一个收件任务 |
|  | 后置条件 | 在一次收件任务中，增加货物信息 |
|  | 语法 | Public ResultMessage update(String id) |
| Receivement.update | 前置条件 | 已输入收件货物信息 |
|  | 后置条件 | 系统持久化更新收件信息 |
|  | 语法 | Public ResultMessage delete(String id) |
| Receivement.delete | 前置条件 | 存在已经输入的Po |
|  | 后置条件 | 将选中的po删除 |
|  | 需要的服务（需接口） |  |
| 服务名 |  | 服务 |
| Receivementdataservice.insert(ReceivementPO po) | 插入单一持久化对象 |  |

**reportbl模块的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供服务） | | | |
| Report.getPaymentSheet | 语法 | | Public String getPaymentSheet() |
| 前置条件 | | 无 |
| 后置条件 | | 返回总收入、总支出、总利润、截至时间 |
| Report.getBussinessSheet | 语法 | | Public SheetVO[] getBussinessSheet (String startDate, String endDate) |
| 前置条件 | | startDate,endDate符合输入规范 |
| 后置条件 | | 返回所有的收款单、付款单 |
| Report.export | 语法 | | Public void export() |
| 前置条件 | | 已经正确显示成本收益表或经营情况表 |
| 后置条件 | | 在系统目录下导出当前显示的表单的excel表格 |
| Report.getSumReport | 语法 | | Public ResultMessage getSumMessage() |
| 前置条件 | | 已得到所有收款和付款数据 |
| 后置条件 | | 得到总金额 |
| Report.exit | 语法 | | Public void exit() |
| 前置条件 | | 无 |
| 后置条件 | | 结束表单查询回合，退出当前界面，返回上一层界面 |
| 需要的服务（需服务） | | | |
| 服务名 | | 服务 | |
| moneydataservice.getPayid | | 得到收款单ID | |
| moneydataservie.getExpenseid | | 得到付款单ID | |
| moneyblservice.find | | 根据付款单或收款单ID得到单据 | |

**Sendbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的接口（供接口） | | |
| Sendbl.inquireSendMesg | 语法 | Public resultmessage inquireSendMesg(String id) |
| 前置条件 | 使用者有查看订单信息的权限 |
| 后置条件 | 无 |
| Sendbl.input | 语法 | Public resultmessage input(String nameOfSender,String addressOfSender,String workplaceOfSender,String phoneOfSender,String mobileOfSender, String nameOfReceiver,String addressOfReceiver,String workplaceOfReceiver,String phoneOfReceiver,String mobileOfReceiver,int numOfItem,double weight,String nameOfItemInside,double[][][] size,double moneyForCover,String speedLevel,String kindOfSend) |
| 前置条件 | 使用者有输入订单的权限 |
| 后置条件 | 系统申请存储订单信息 |
| 需要的接口（需接口） | | |
| 服务名 | 服务 | |
| Senddata.getSendMesg | 返回单一持久化对象（物流订单信息） | |
| Senddata.getInput | 写入单一持久化对象（订单信息） | |
| Imformationbl.refreshMesg | 写入单一持久对象（物流信息） | |

**staffbl模块的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供服务） | | | |
| staff.addAccount | 语法 | | Public void addAccount(String name,String department); |
| 前置条件 | | 启动一个人员机构管理回合 |
| 后置条件 | | 分配一个初始的用户账号，并将该员工与账号绑定，持久化该对象 |
| staff.delateAccount | 语法 | | Public void delateAccount(String userID,String department); |
| 前置条件 | | 启动一个人员机构管理回合 |
| 后置条件 | | 删除该用户信息，持续化更新持有该对象的数据 |
| staff.adjustAccount | 语法 | | Public void adjustAccount(String userID,String department); |
| 前置条件 | | 启动一个人员机构管理回合 |
| 后置条件 | | 修改该用户的部门信息，持续化更新该对象的数据 |
| staff.setSalaryModel | 语法 | | Public void setSalaryModel(String userID,SalaryModel salarymodel); |
| 前置条件 | | 输入的ID存在一个员工与之对应 |
| 后置条件 | | 更改该ID所对应的员工的薪水策略，持续化更新该对象的数据 |
| staff.setUserInformation | 语法 | | Public void setUserInformation(String userID,String[] tags,String[] conents); |
| 前置条件 | | 输入的用户ID有效，输入的标签项存在 |
| 后置条件 | | 更改用户的信息或权限，持续化更新该对象的数据 |
| staff.addVehicle | 语法 | | Public VehiclePO addVehicle(String vehicleCode,String engineID,String vehicleID,String chassisID,Sting boughtTime,String inTime,image vehicle); |
| 前置条件 | | 启动一个车辆信息管理回合，输入符合规范 |
| 后置条件 | | 新增一个车辆对象，持续化更新该对象的数据，返回一个VO对象 |
| staff.delateVehicle | 语法 | | Public void delateVehicle (String vehicleCode); |
| 前置条件 | | 启动一个车辆信息管理回合，该车辆代号存在 |
| 后置条件 | | 删除该车辆信息，持续化更新持有该对象的数据 |
| staff.adjustVehicle | 语法 | | Public VehicleVO adjustVehicle(String vehicleCode,String engineID,String vehicleID,String chassisID,Sting boughtTime,String inTime,image vehicle); |
| 前置条件 | | 启动一个车辆信息管理回合 |
| 后置条件 | | 修改该车辆的信息，持续化更新该对象的数据，返回一个VO对象 |
| staff.addDriver | 语法 | | Public DriverVO addDriver(String driverCode,String name,String vehicleID,String birthday,Sting IDnumber,String phoneNumber,String department,String sex,String validDate); |
| 前置条件 | | 启动一个司机信息管理回合，输入符合规范 |
| 后置条件 | | 新增一个司机对象，持续化更新该对象的数据，返回一个VO对象 |
| staff.delateDriver | 语法 | | Public void delateAccount(String driverCode); |
| 前置条件 | | 启动一个司机信息管理回合，该司机代号存在 |
| 后置条件 | | 删除该司机信息，持续化更新持有该对象的数据 |
| staff.adjustDriver | 语法 | | Public DriverVO adjustAccount(String vehicleCode,String engineID,String vehicleID,String chassisID,Sting boughtTime,String inTime,image vehicle); |
| 前置条件 | | 启动一个司机信息管理回合 |
| 后置条件 | | 修改该司机的信息，持续化更新该对象的数据，返回一个VO对象 |
| staff.find | 语法 | | Public UserVO find(String userID); |
| 前置条件 | | 输入的ID有效 |
| 后置条件 | | 返回该ID对应的用户 |
| 需要的服务（需服务） | | | |
| 服务名 | | 服务 | |
| staffdataservice.insert(UserPO po) | | 添加单一持久化对象 | |
| staffdataservice.insert(DriverPO po) | | 添加单一持久化对象 | |
| staffdataservice.insert(VehiclePO po) | | 添加单一持久化对象 | |
| staffdataservice.delete(String id) | | 删除单一持久化对象 | |
| staffdataservice.update(PO po) | | 更新单一持久化对象 | |
| staffdataservice.find(String id) | | 根据编号查找相应持久化对象 | |
| staffdataservice.init() | | 初始化单一持久化对象 | |

## 5.4数据层的分解

### 5.4.1数据层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| Imformationdata | 提供物流信息查询服务 |
| Senddata | 提供快递信息，报价和时间管理相关服务 |
| Judgementdata | 提供单据查看，审判服务 |
| Shipmentdata | 提供装车管理，装运管理和中转接收的数据存储和查看服务 |
| Receivementdata | 提供收件信息存储，接收与派件信息存储与查询的服务 |
| Commoditydata | 提供库存信息的存储，库存信息变更的存储，库存信息的查询等服务 |
| Moneydata | 提供收款单信息的存储查询，结算管理，财务支出等服务 |
| Staffdata | 提供人员信息、账户信息、机构信息、车辆信息、司机信息、用户管理的增删改查服务 |

### 5.4.2数据层模块的接口规范

**commoditydataservice模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
|  | 语法 | Public ResultMessage insert(CommodityPO po) |
| commoditydataservice.insert | 前置条件 | 同样的po在数据中不存在 |
|  | 后置条件 | 插入一条po数据 |
|  | 语法 | Public ResultMessage delete(CommodityPO po) |
| commoditydataservice.delete | 前置条件 | 同样的po在数据中存在 |
|  | 后置条件 | 删除一条po数据 |
|  | 语法 | Public ResultMessage update(CommodityPO po) |
| commoditydataservice.updata | 前置条件 | 同样的po在数据中存在 |
|  | 后置条件 | 更新Po数据 |
|  | 语法 | Public ResultMessage find(String id) |
| commoditydataservice.find | 前置条件 | 无 |
|  | 后置条件 | 返回查找相应的结果 |
|  | 语法 | Public void init() |
| Commoditydataservice.init | 前置条件 | 无 |
|  | 后置条件 | 初始化持久数据 |

**Imformationdata的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的接口（供接口） | | |
| Imformationdata.getTransMesg | 语法 | Public resultmessage getTransMesg(String id) |
| 前置条件 | 快递信息已经返回 |
| 后置条件 | 无 |
| Imformationdata.refreshTransMesg | 语法 | Public resultmessage refreshTransMesg(String id,String position,String time) |
| 前置条件 | 货物接收成功 |
| 后置条件 | 系统保存货物物流信息 |

**moneydataservicce模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
|  | 语法 | Public ResultMessage insert(MoneyPO po) |
| moneydataservice.insert | 前置条件 | 同样的po在数据中不存在 |
|  | 后置条件 | 增加一条po记录 |
|  | 语法 | Public ResultMessage delete(String id) |
| moneydataservice.delete | 前置条件 | 同样的po在数据中存在 |
|  | 后置条件 | 删除一条po数据 |
|  | 语法 | Public ResultMessage update(MoneyPO po) |
| moneydataservice.update | 前置条件 | 同样的po在数据中存在 |
|  | 后置条件 | 更新一条po数据 |
|  | 语法 | Public MoneyPO find(String id) |
| moneydataservice.find | 前置条件 | 无 |
|  | 后置条件 | 查找返回相应的结果 |
|  | 语法 | Public void init() |
| moneydataservice.init | 前置条件 | 无 |
|  | 后置条件 | 初始化持久数据 |
|  | 语法 | Public String getPayid() |
| moneydataservice.getPayid | 前置条件 | 无 |
|  | 后置条件 | 返回付款单id |
|  | 语法 | Public String getExpenseid() |
| moneydataservie.getExpenseid | 前置条件 | 无 |
|  | 后置条件 | 返回收款单id |
|  | 语法 | Public String getBillid() |
| moneydataservice.getBillid | 前置条件 | 无 |
|  | 后置条件 | 返回成本收益单id |
|  | 语法 | Public String getMember(String name) |
| moneydataservice.getMember | 前置条件 | 账户已在系统中建立账号 |
|  | 后置条件 | 返回账户的账号名称 |
|  | 语法 | Public String getMemberBill(double bill) |
| moneydataservice.getMember | 前置条件 | 账户已在系统中进行交易 |
| Bill | 后置条件 | 返回查看账户的交易记录（交易金额） |

**receivementdataservice 模块的接口规范**

|  |  |  |
| --- | --- | --- |
|  | 提供的服务 | （供接口） |
| receivementdataserv | 语法 | Public ResultMessage find(String id) |
| ice.find | 前置条件 | 无 |
|  | 后置条件 | 查找返回相应的结果 |
| reseivementdataservic | 语法 | Public ResultMessage storage(String id) |
| e.storage | 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 |
| receivementdataservic | 语法 | Public ResultMessage delete(String id) |
| e.delete | 前置条件 | 同样的po在数据中存在 |
|  | 后置条件 | 删除po |
| receivementdataservice.update | 语法 | Public ResultMessage update(ReceivementPO po) |
|  | 前置条件 | 同样的po在数据中存在 |
|  | 后置条件 | 更新po |
| receivementdataservic | 语法 | Public void init() |
| e.init | 前置条件 | 无 |
|  | 后置条件 | 初始化持久数据 |
| receivementdataservic | 语法 | Public void insert(ReceivementPO po) |
| e.insert | 前置条件 | 同样id的po在数据中不存在 |
|  | 后置条件 | 在数据库中增加一条po记录 |

**Senddata模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的接口（供接口） | | |
| Senddata.getSendMesg | 语法 | Public resultmessage getSendMesg(String id) |
| 前置条件 | 无 |
| 后置条件 | 无 |
| Senddata.getInput | 语法 | Public resultmessage getInput(OrderBillVO) |
| 前置条件 | 使用者有写入的权限 |
| 后置条件 | 系统保存写入的订单信息 |

**staffdataservice 模块的接口规范**

|  |  |  |
| --- | --- | --- |
|  | 提供的服务 | （供接口） |
| staffdataservice.insert | 语法 | Public void insert(UserPO po); |
| 前置条件 | 无 |
| 后置条件 | 在记录用户信息的数据中增加新的数据 |
| staffdataservice.insert | 语法 | Public void insert(DriverPO po) |
| 前置条件 | 无 |
| 后置条件 | 在记录司机信息的数据中增加新的数据 |
| staffdataservice.insert | 语法 | Public void insert(VehiclePO po) |
| 前置条件 | 无 |
| 后置条件 | 在记录车辆信息的数据中增加新的数据 |
| staffdataservice.delete | 语法 | Public void delete(String id) |
| 前置条件 | 数据中存在该ID标识的PO |
| 后置条件 | 从数据中删除该PO |
| staffdataservice.update | 语法 | Public void update(PO po) |
| 前置条件 | 系统中存在该PO |
| 后置条件 | 更新PO |
| staffdataservice.find(String id) | 语法 | Public PO find(String id) |
| 前置条件 | 系统中记录有该ID |
| 后置条件 | 返回PO对象 |
| staffdataservice.init | 语法 | public void init(); |
| 前置条件 | 无 |
| 后置条件 | 初始化数据 |

# 6.信息视角

## 6.1数据持久化对象

系统的PO类就是对应的实体类，如下所示。

TransMesgPO类包含订单编号、订单列表、货运状态

OrderBillPO是保存及更新订单数据的类，包含订单编号、订单列表、订单参数、是否审批通过、收件信息（收件编号、收件人、收件日期）

ReceiveBillPO是保存接受与派件信息的类，包含货物到达信息（到达日期、中转单编号、出发地、货物到达状态（损坏、完整、丢失））、派件信息（到达日期、托运订单条形码号、派送员）

ShipmentBillPO是保存装运信息的类，包含装车单信息、中转单信息、中转中心到达单信息

CommodityPO类包含货物信息（快递编号、入库日期、目的地、区号、排号、架号、位号）

AccountBillPO类包含收款单信息（收款日期、收款金额、收款快递员、对应的所有快递订单条形码号）、付款单信息（付款日期、付款金额、付款人、付款账号、条目（租金（按年收）运费（按次计算）人员工资（按月统计）奖励（一次性））、成本收益表信息（收款单、付款单、总利润）、经营情况表信息（收款单、付款单、起始日期）

StaffBillPO类包含车辆信息、司机信息、人员与机构信息账户信息

UserBillPO类包含用户授权信息

## 6.2文件格式

统一用\*.ser格式，比如orderbill.ser