快递物流系统体系结构

设计文档

编写人：陶子涵

谭琼

徐文杰

徐家逸

编写时间：2015.10.18

目录

[更新记录 3](#_Toc433415569)

[1.引言 4](#_Toc433415570)

[1.1编制目的 4](#_Toc433415571)

[1.2词汇表 4](#_Toc433415572)

[1.3参考资料 4](#_Toc433415573)

[2.产品概述 4](#_Toc433415574)

[3.逻辑视角 5](#_Toc433415575)

[4.组合视角 6](#_Toc433415576)

[4.1开发包图 6](#_Toc433415577)

[4.2运行时进程 9](#_Toc433415578)

[4.3物理部署 9](#_Toc433415579)

[5.接口视角 10](#_Toc433415580)

[5.1模块的职责 10](#_Toc433415581)

[5.2用户界面层的分解 12](#_Toc433415582)

[5.2.1用户界面层模块的职责 14](#_Toc433415583)

[5.2.2用户界面层模块的接口规范 16](#_Toc433415584)

[5.2.3用户界面模块设计原理 23](#_Toc433415585)

[5.3业务逻辑层的分解 23](#_Toc433415586)

[5.4数据层的分解 38](#_Toc433415587)

[5.4.1数据层模块的职责 39](#_Toc433415588)

[5.4.2数据层模块的接口规范 39](#_Toc433415589)

[6.信息视角 53](#_Toc433415590)

[6.1数据持久化对象 53](#_Toc433415591)

[6.1.1持久化对象UserPO 57](#_Toc433415592)

[6.1.2持久化对象carInfoPO 58](#_Toc433415593)

[6.1.3持久化对象ConstPO 58](#_Toc433415594)

[6.1.5持久化对象OrderFormPO 60](#_Toc433415595)

[6.1.6持久化对象OfficeArrivalFormPO 65](#_Toc433415596)

[6.1.7持久化对象CarOfficeFormPO 67](#_Toc433415597)

[6.1.8持久化对象DeliveryFormPO 70](#_Toc433415598)

[6.1.9持久化对象InventoryInfoPO 72](#_Toc433415599)

[6.1.10持久化对象StoreArrivalFormPO 73](#_Toc433415600)

[6.1.11持久化对象TransferFormPO 74](#_Toc433415601)

[6.1.12持久化对象CarInputFormPO 77](#_Toc433415602)

[6.1.13持久化对象StorageOutFormPO 79](#_Toc433415603)

[6.1.14持久化对象StorageInFormPO 80](#_Toc433415604)

[6.1.15持久化对象DriverInfoPO 80](#_Toc433415605)

[6.1.16持久化对象AccountPO 81](#_Toc433415606)

[6.1.17持久化对象BeginningAccountPO 82](#_Toc433415607)

[6.1.18持久化对象CostPayChartPO 83](#_Toc433415608)

[6.1.19持久化对象PaymentFormPO 84](#_Toc433415609)

[6.1.20持久化对象ReceiptFormPO 84](#_Toc433415610)

# 更新记录

|  |  |  |  |
| --- | --- | --- | --- |
| 更新时间 | 更新者 | 更新原因 | 版本号 |
| 2015/10/24 | 谭琼 | 业务逻辑层接口的分解需要大改 | V1.0 |
|  |  |  |  |

# 1.引言

## 1.1编制目的

本报告详细完成对快递物流系统的概要设计，达到指导详细设计和开发目的，同时实现和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而编写，是了解系统的导航。

## 1.2词汇表

|  |  |  |
| --- | --- | --- |
| 词汇名称 | 词汇含义 | 备注 |
| \_ui | 表示某展示层 |  |
| \_bl | 表示某逻辑层 |  |
| \_data | 表示某数据层 |  |

## 1.3参考资料

1.IEEE标准

2.《快递物流系统软件需求规格说明文档v2.1》

3. Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Micheael Stal.Pattern-Oriented Software Architecture Volume 1: A system of Patterns [M]机械工业出版社,2003

4.《快递物流系统用例文档》

# 2.产品概述

参考《快递物流系统用例文档》和《快递物流系统软件需求规格说明文档v2.1》中对产品的概括描述。

# 3.逻辑视角

快递物流系统中，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合访问。分层体系结构的逻辑视角和逻辑设计方案如图3-1和图3-2所示。



图3-1 参照体系结构风格的包图表达逻辑视角

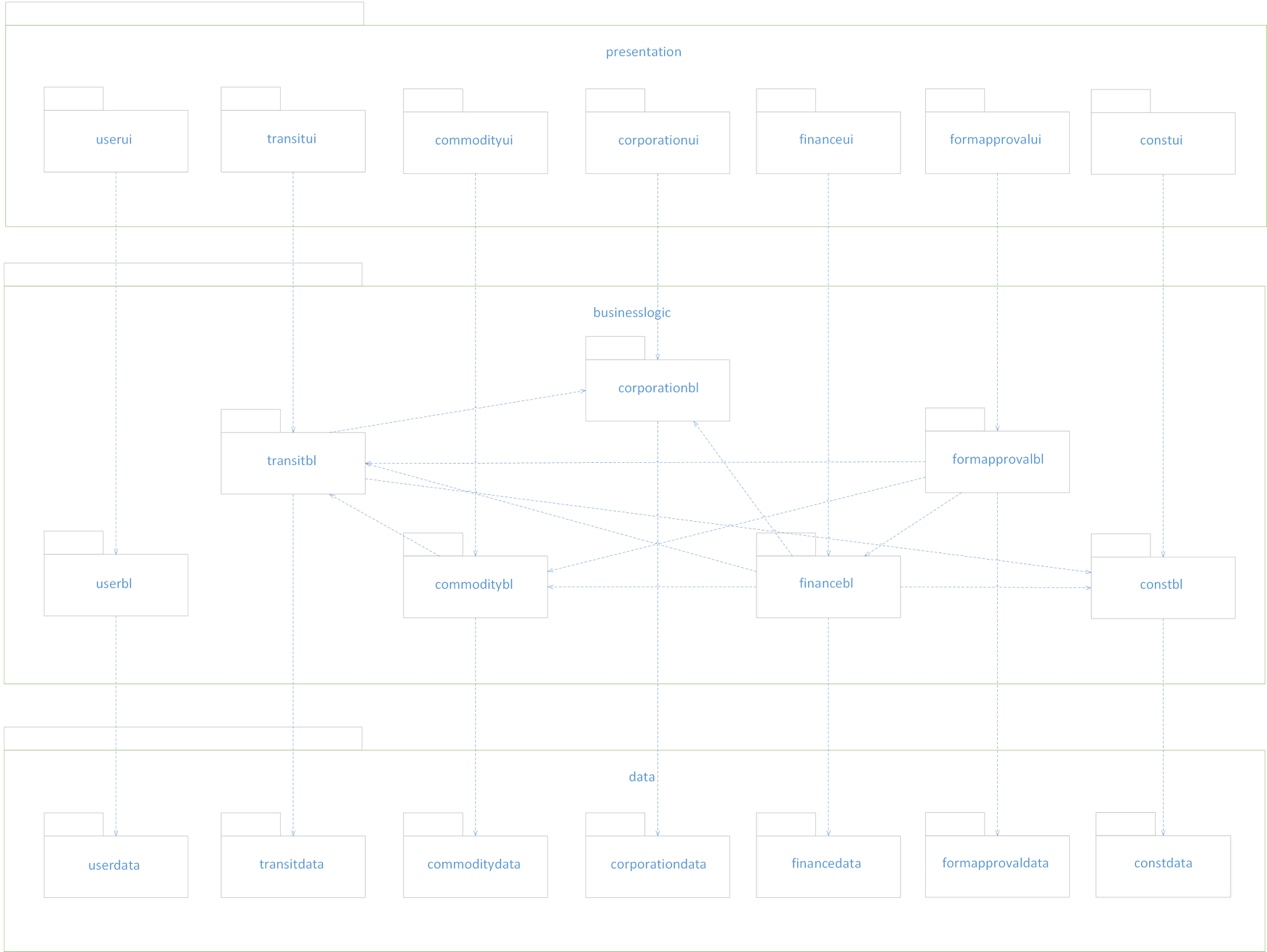


图3-2 软件体系结构逻辑设计方案

# 4.组合视角

## 4.1开发包图

表4.1-1 快递物流系统最终开发包设计

|  |  |
| --- | --- |
| **开发（物理）包** | **依赖的其他开发包** |
| mainui | transitui，inventoryui，financeui，userui，corporationui，constui，formapprovalui，vo |
| transitui | transitblservice，界面类库包，vo |
| transitblservice |  |
| transitbl | transitblservice，transitdataservice，po，corporationbl，constbl |
| transitdataservice | Java RMI，po |
| transitdata | Java RMI，po |
| inventoryui | inventory blservice，界面类库包 |
| inventoryblservice |  |
| inventorybl | inventoryblservice，inventorydataservice，po，transitbl |
| inventorydataservice | Java RMI，po |
| inventorydata | Java RMI，po |
| financeui | financeblservice，界面类库包 |
| financeblservice |  |
| financebl | financeblservice，financedataservice，po，transitbl，corporationbl，inventorybl，constbl |
| financedataservice | Java RMI，po |
| financedata | Java RMI，po |
| userui | userblservice，界面类库包 |
| userblservice |  |
| userbl | UserInterface，UserDataClient，UserPO |
| userdataservice | Java RMI，po |
| userdata | Java RMI，po |
| corporationui | corporationblservice，界面类库包 |
| corporationblservice |  |
| corporationbl | corporationblservice，corporationdataservice，po |
| corporationdataservice | Java RMI，po |
| corporationdata | Java RMI，po |
| constui, | constblservice，界面类库包 |
| constblservice |  |
| constbl | constblservice，constdataservice，po |
| constdataservice | Java RMI，po |
| constdata | Java RMI，po |
| formapprovalui | formapprovalblservice，界面类库包 |
| formapprovalblservice |  |
| formapprovalbl | formapprovalblservice，formapprovaldataservice，po，transitbl，inventorybl，financebl |
| formapprovaldataservice | Java RMI，po |
| formapprovaldata | Java RMI，po |
| vo |  |
| po |  |
| 界面类库包 |  |
| Java RMI |  |

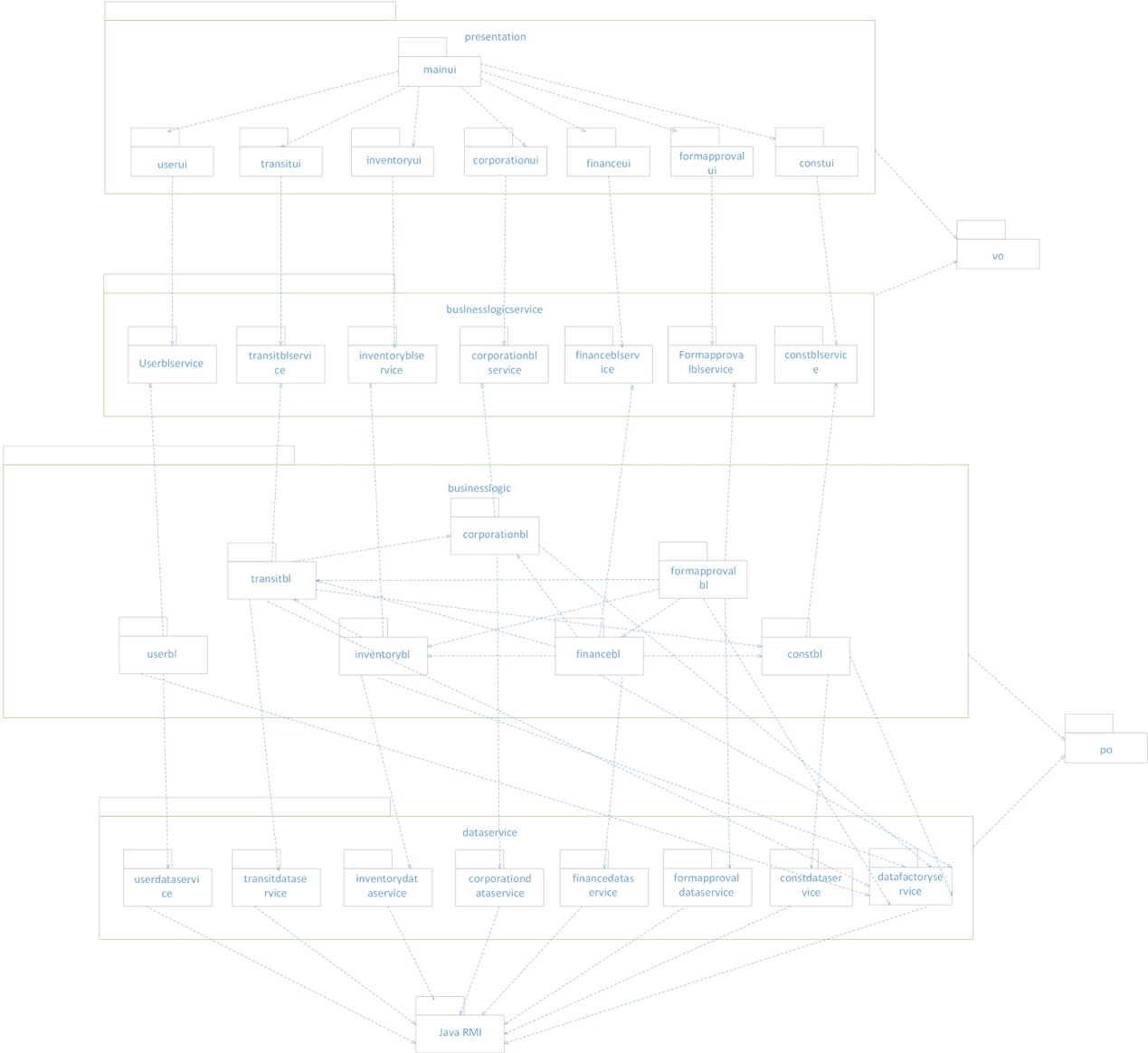


图4.1-1客户端开发包图

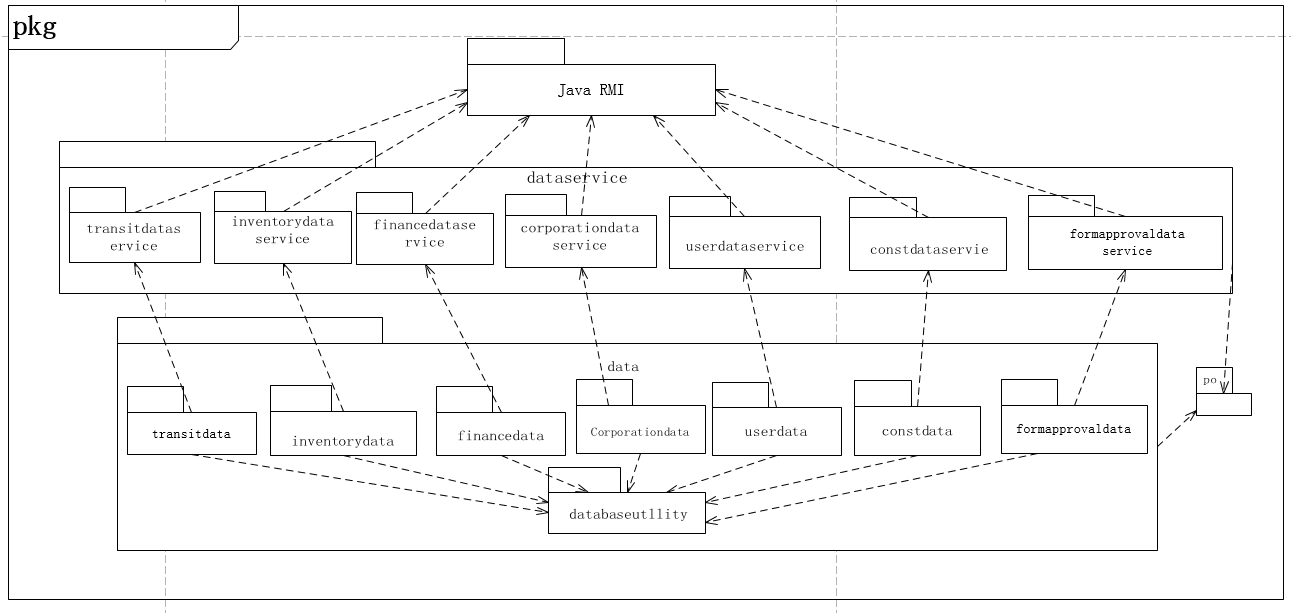


图4.1-2服务器端开发包图

## 4.2运行时进程

在快递物流信息系统中，会有多个客户端进程和一个服务端进程，其进程如图4.2-1所示。结合部署图，客户端进程是在客户端机器上运行，服务器端进程在服务器端机器上运行。

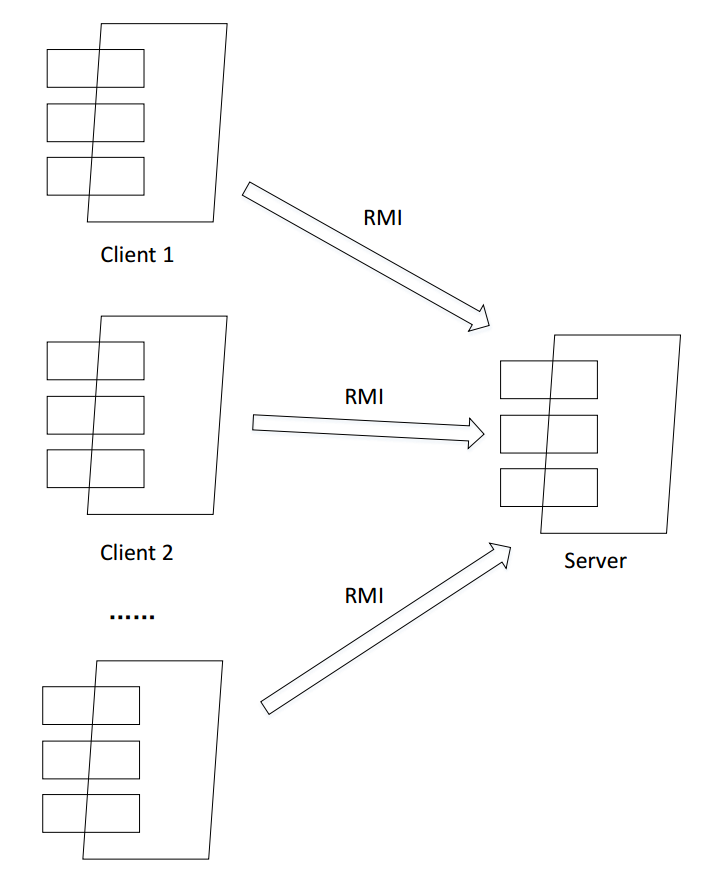


图4.2-1 进程图

## 4.3物理部署

快递物流信息系统中客户端构件是放在客户端机器上，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 1.8.0的一部分。所以在系统JDK环境已经设置好的情况下，不需要再独立部署。部署图如图4.3-1所示。

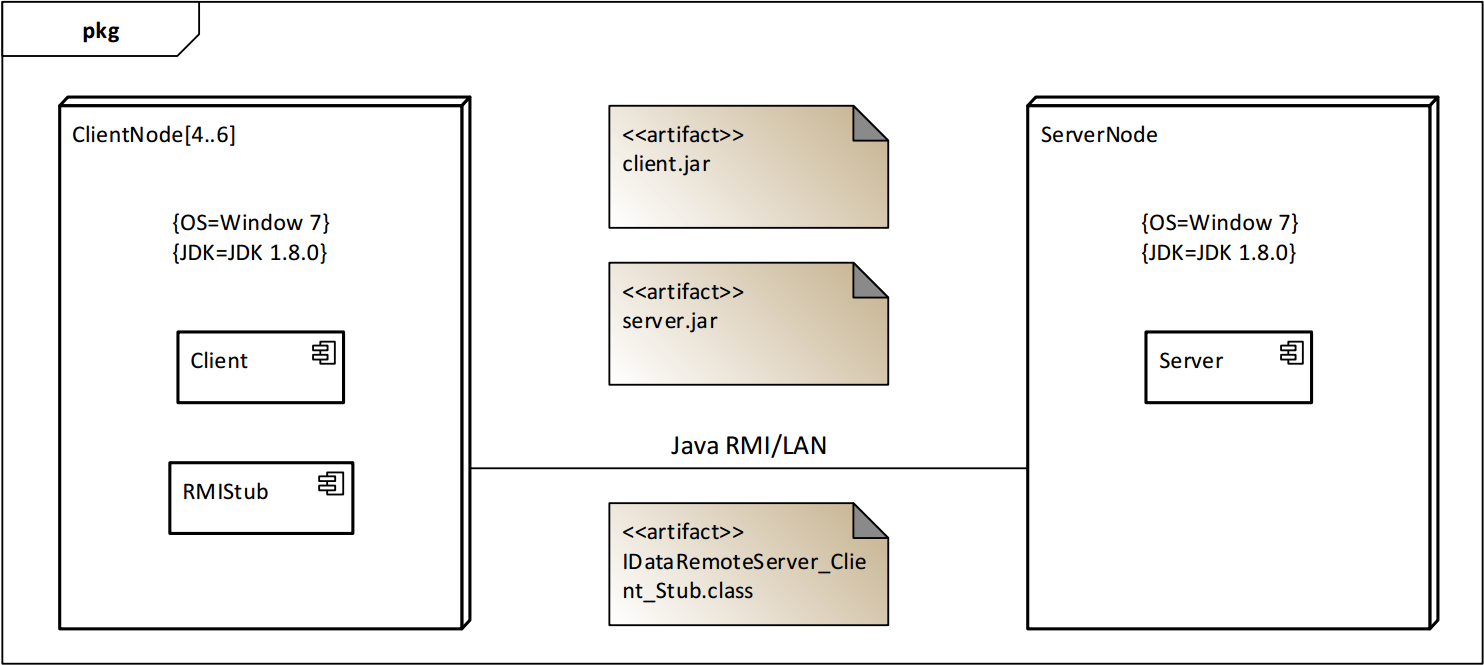


图4.3-1 部署图

# 5.接口视角

## 5.1模块的职责

客户端模块和服务器端模块试图分别如图5.1-1和图5.1-2所示。客户端各层和服务器端各层的职责分别如表5.1-1和表5.1-2所示。

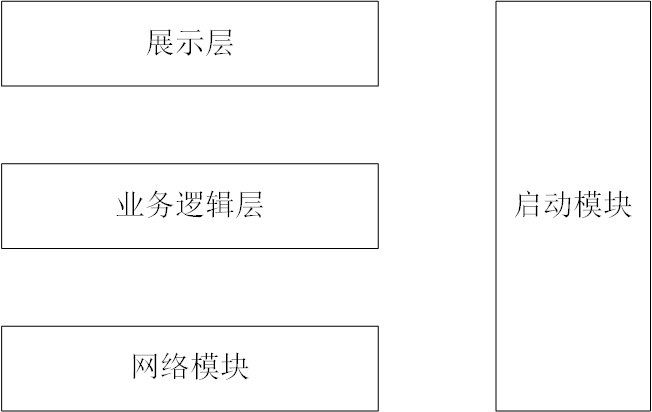


图5.1-1 客户端模块视图



图5.1-2 服务器端模块视图

表5.1-1 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 | 基于窗口的快递物流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5.1-2 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5.1-3所示。

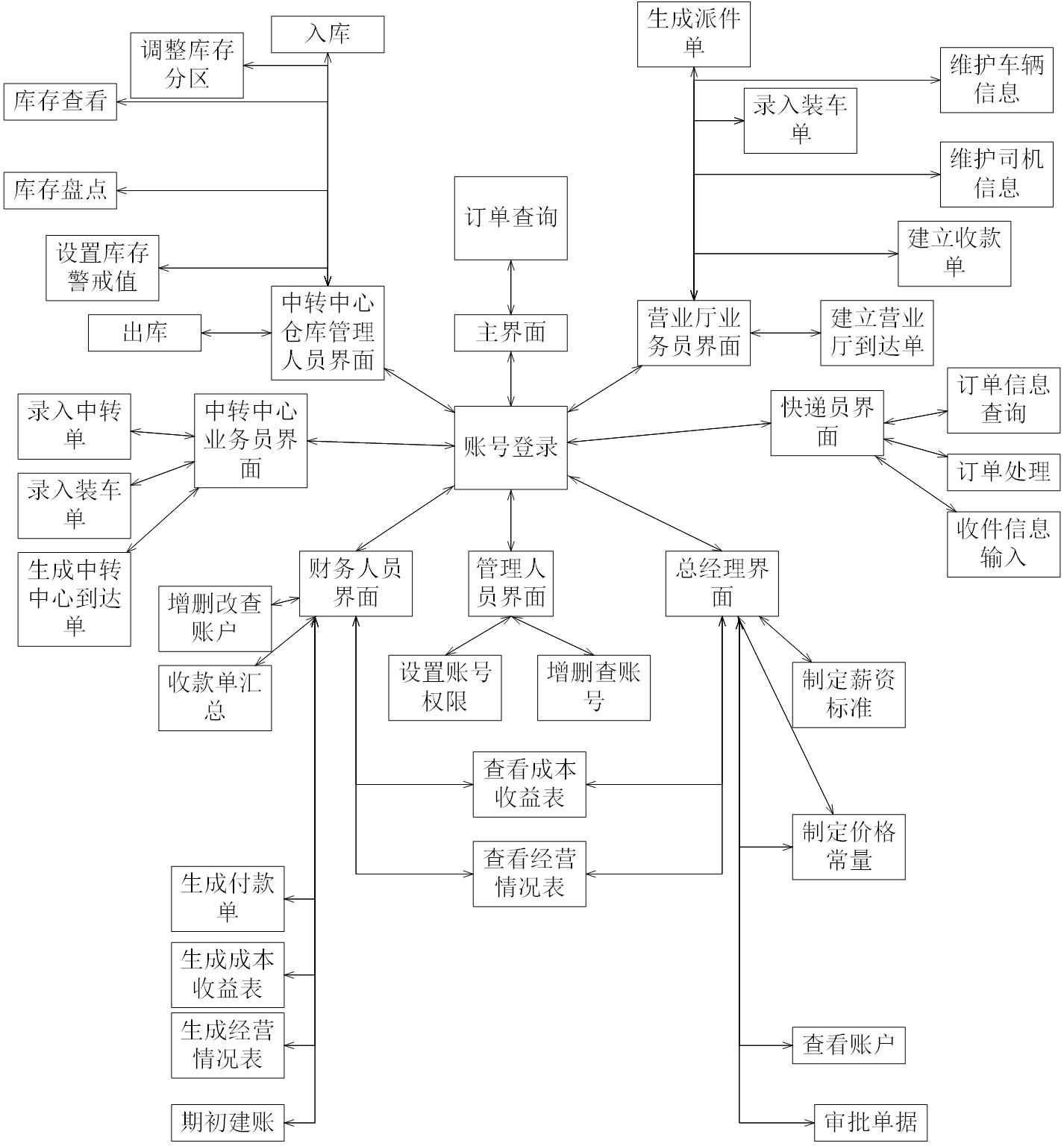
**表5.1-3 层之间调用的接口**

|  |  |  |
| --- | --- | --- |
| **接口** | **服务调用方** | **服务提供方** |
| LoginBLServer  TransitBLServer  CommodityBLServer  CorporationBLServer  FinanceBLServer  FormapprovalBLServer  ConstBLServer | 客户端展示层 | 客户端业务逻辑层 |
| UserDataServer  TransitDataServer  CommodityDataServer  CorporationDataServer  FinanceDataServer  FormapprovalDataServer  ConstDataServer  DatabaseFactory | 客户端业务逻辑层 | 服务器端数据层 |

借用…来说明层之间的调用，如图5.1-3所示。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）。…提供了…界面所需要的所有业务逻辑功能。…提供了对数据库的…等操作。这样的实现就大大降低了层与层之间的耦合。

## 5.2用户界面层的分解

根据需求，系统存在42个用户界面：订单查询界面、主界面、账号登录界面、中转中心仓库管理人员界面、入库界面、调整库存分区界面、库存查看界面、库存盘点界面、设置库存警戒值界面、出库界面、营业厅业务员界面、生成派件单界面、维护车辆信息界面、录入装车单界面、维护司机信息界面、建立收款单界面、建立营业厅到达单界面、中转中心业务员界面、录入中转单界面、录入装车单界面、生成中转中心到达单界面、快递员界面、订单信息查询界面、订单处理界面、收件信息输入界面、管理人员界面、设置账号权限界面、增删查账号界面、财务人员界面、增删改查账户界面、收款单汇总界面、生成付款单界面、生成成本收益表界面、生成经营情况表界面、期初建账、总经理界面、制定薪资标准、制定价格常量、查看账户界面、审批单据界面、查看成本收益表界面、查看经营情况表界面、查看系统日志。界面跳转如图5.2-1所示。



5.2-1 用户界面跳转

服务器端和客户端的用户界面设计接口是一致的，只是具体的页面不一样。用户界面类如图5.2-2所示。（字实在是大不出了助教您凑合着看…下面有详细的不要虚）



5.2-2 用户界面类

### 5.2.1用户界面层模块的职责

如表5.2.1-1所示为用户界面层模块的职责。

表5.2.1-1 用户界面层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| MainFrame | 负责界面的显示和界面的跳转。 |
| FrameEnd | 负责界面的关闭 |
| Main | 系统主界面，负责主界面的监听和信息传递 |
| OutCheckLabel | 外部查询界面，负责输入信息的传递和同步 |
| LoginLabel | 内部登录界面，负责输入信息的传递和同步 |
| ManagerPanel | 总经理板块，负责该板块的监听和信息传递 |
| CourierPanel | 快递员板块，负责该板块的监听和信息传递 |
| AdminPanel | 管理员板块，负责该板块的监听和信息传递 |
| OfficerPanel | 营业厅业务员板块，负责该板块的监听和信息传递 |
| StoragerPanel | 中转中心业务员板块，负责该板块的监听和信息传递 |
| KeeperPanel | 中转中心仓库管理员板块，负责该板块的监听和信息传递 |
| AccountantPanel | 财务人员板块，负责该板块的监听和信息传递 |
| CheckAccountBoa  Rd | 增删改查账户界面，负责该界面的监听和信息传递 |
| ReciptGatherBoar  D | 收款单汇总界面，负责该界面的监听和信息传递 |
| PaymentBoard | 生成付款单界面，负责该界面的监听和信息传递 |
| CostPayFormBoar  D | 成本收益表界面，负责该界面的监听和信息传递 |
| ProfitFormBoard | 经营情况表界面，负责该界面的监听和信息传递 |
| AccountsBoard | 期初建账界面，负责该界面的监听和信息传递 |
| CheckBoard | 查看账户界面，负责该界面的监听和信息传递 |
| CheckProfitBoard | 查看经营情况表界面，负责该界面的监听和信息传递 |
| CheckCostPsyBoa  Rd | 查看成本收益表界面，负责该界面的监听和信息传递 |
| JudgeBoard | 审批单据界面，负责该界面的监听和信息传递 |
| SalaryBoard | 制定薪资标准界面，负责该界面的监听和信息传递 |
| PriceBoard | 制定价格常量界面，负责该界面的监听和信息传递 |
| DeliveryBoard | 生成派件单界面，负责该界面的监听和信息传递 |
| CarOfficeBoard | 维护车辆信息界面，负责该界面的监听和信息传递 |
| CarBoard | 录入装车单界面，负责该界面的监听和信息传递 |
| ChauffeurBoard | 维护司机信息界面，负责该界面的监听和信息传递 |
| ReceiptsBoard | 生成收款单界面，负责该界面的监听和信息传递 |
| OfficeArrivalBoard | 生成营业厅到达单界面，负责该界面的监听和信息传递 |
| StorageInBoard | 入库界面，负责该界面的监听和信息传递 |
| AdjustmentBoard | 调整库存分区界面，负责该界面的监听和信息传递 |
| StorageCheckBoa  Rd | 库存查看界面，负责该界面的监听和信息传递 |
| StorageCountBoa  Rd | 库存盘点界面，负责该界面的监听和信息传递 |
| WarningLimintBo  Ard | 设置库存警戒值界面，负责该界面的监听和信息传递 |
| StorageOutBoard | 出库界面，负责该界面的监听和信息传递 |
| StoreArrivalBoard | 中转中心到达单界面，负责该界面的监听和信息传递 |
| TransferBoard | 中转单界面，负责该界面的监听和信息传递 |
| CarInputBoard | 装车单界面，负责该界面的监听和信息传递 |
| OrderBoard | 订单处理，负责该界面的监听和信息传递 |
| QueryBoard | 订单信息查询界面，负责该界面的监听和信息传递 |
| OrderReceiveBoa  Rd | 收件信息输入界面，负责该界面的监听和信息传递 |
| ManageAccountB  oard | 增删查账号界面，负责该界面的监听和信息传递 |
| SystemBoard | 查看系统日志界面，负责该界面的监听和信息传递 |
| AuthorityBoard | 设置账号权限界面，负责该界面的监听和信息传递 |

### 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如表5.2.2-1所示。

表5.2.2-1 用户界面层模块的接口规范

|  |  |  |
| --- | --- | --- |
| MainFrame | 语法 | init(string args[]) |
| 前置条件 | 收到初始化请求 |
| 后置条件 | 跳转到对应的面板并显示 |

|  |  |  |
| --- | --- | --- |
| FrameEnd | 语法 | end( sting args[]); |
| 前置条件 | 收到关闭某界面的请求 |
| 后置条件 | 关闭相应界面 |

|  |  |  |
| --- | --- | --- |
| Main | 语法 | listen（）；sync（）； |
| 前置条件 | 系统开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值并提交MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| OutCheckLabel | 语法 | listen（）；sync（）； |
| 前置条件 | 被Main请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值并提交MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| LoginLabel | 语法 | listen（）；read（）；sync（）； |
| 前置条件 | 被Main请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值并提交loginbl和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| ManagerPanel | 语法 | listen（）；sync（）； |
| 前置条件 | 由loginbl请求开启，通过通过MainFrame初始化 |
| 后置条件 | 返回有效监听值并提交managerbl和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| CourierPanel | 语法 | listen（）；sync（）； |
| 前置条件 | 由loginbl请求开启，通过通过MainFrame初始化 |
| 后置条件 | 返回有效监听值并提交courierbl和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| AdminPanel | 语法 | listen（）；sync（）； |
| 前置条件 | 由loginbl请求开启，通过通过MainFrame初始化 |
| 后置条件 | 返回有效监听值并提交adminbl和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| OfficerPanel | 语法 | listen（）；sync（） |
| 前置条件 | 由loginbl请求开启，通过通过MainFrame初始化 |
| 后置条件 | 返回有效监听值并提交officerbl和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| StoragerPanel | 语法 | listen（）；sync（） |
| 前置条件 | 由loginbl请求开启，通过通过MainFrame初始化 |
| 后置条件 | 返回有效监听值并提交storagerbl和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| KeeperPanel | 语法 | listen（）；sync（） |
| 前置条件 | 由loginbl请求开启，通过通过MainFrame初始化 |
| 后置条件 | 返回有效监听值并提交keeperbl和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| AccountantPanel | 语法 | listen（）；sync（） |
| 前置条件 | 由loginbl请求开启，通过通过MainFrame初始化 |
| 后置条件 | 返回有效监听值并提交accountant和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| CheckAccountBoa  rd | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由accountantbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| ReciptGatherBoar  d | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由accountantbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| PaymentBoard | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由accountantbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| CostPayFormBoar  d | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由accountantbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| ProfitFormBoard | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由accountantbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| AccountsBoard | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由accountantbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| CheckBoard | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由managerbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| CheckProfitBoard | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由managerbl与accountantbl请求开启，由MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| CheckCostPsyBoa  rd | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由managerbl与accountantbl请求开启，由MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| JudgeBoard | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由managerbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |
| 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由managerbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| SalaryBoard | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由managerbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| DeliveryBoard | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由officerbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| CarOfficeForm | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由officerbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| CarBoard | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由officerbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| ChauffeurBoard | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由officerbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| ReceiptsBoard | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由officerbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| ReceiveBoard | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由officerbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| StorageInBoard | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由keeperbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| AdjustmentBoard | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由keeperbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| StorageCheckBoard | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由keeperbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| StorageCountBoa  rd | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由keeperbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| WarningLimintBo  ard | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由keeperbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| StorageOutBoard | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由keeperbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| StoreArrivalBoard | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由storagerbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| TransferBoard | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由storagerbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| CarInputBoard | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由storagerbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| OrderBoard | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由storagerbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| QueryBoard | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由storagerbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| OrderReceiveBoa  rd | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由storagerbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| AuthorityBoard | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由adminbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| ManageAccountB  oard | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由adminbl请求开启，通过MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

|  |  |  |
| --- | --- | --- |
| SystemBoard | 语法 | listen（）；read（）；sync（） |
| 前置条件 | 由adminbl与managerbl请求开启，用MainFrame初始化 |
| 后置条件 | 返回有效监听值和输入值并提交面板层对应逻辑层和MainFrame或FrameEnd |

用户界面层需要的服务接口如表5.2.2-2所示。

表5.2.2-2 用户界面层模块需要的服务接口

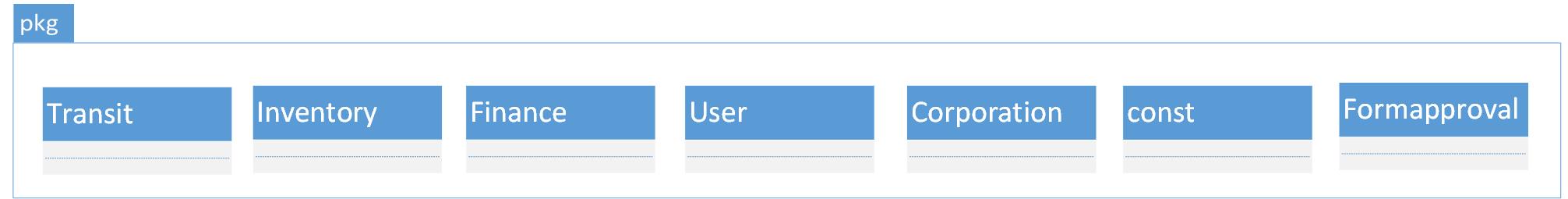
|  |  |
| --- | --- |
| 服务名 | 服务 |
| logicservice.\*BLService | 每个界面都有一个相应的业务逻辑接口（这里不一一赘述） |

### 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

业务逻辑层包括多个针对界面的业务逻辑处理对象。例如，User对象负责处理登录界面的业务逻辑；inventory对象负责库存界面的业务逻辑。业务逻辑层的设计如图：



5.3.1 业务逻辑层模块的职责

业务逻辑层模块的职责如下表所示：

|  |  |
| --- | --- |
| **模块** | **职责** |
| transitbl | 负责实现对应的物流界面所需要的服务 |
| inventorybl | 负责实现对应的库存界面所需要的服务 |
| financebl | 负责实现对应的财务界面所需要的服务 |
| userbl | 负责实现对应的登录界面所需要的服务 |
| corporationbl | 负责实现对应的机构信息管理（司机信息维护、车辆信息维护、增删改查银行账户）界面所需要的服务 |
| constbl | 负责实现对应的常量制定（制定薪资标准、制定价格常量）界面 |
| formapprovalbl | 负责实现对应的各类单据审批的界面所需要的服务 |

5.3.2 业务逻辑层模块的接口规范

#### 5.3.2.1 transitbl模块

表5.3.2-1 transitbl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| transitbl.CarInputFormTransitbl.create | 语法 | public CarInputFormVO CarInputFormcreate () | |
| 前置条件 | 无 | |
| 后置条件 | 增加一个CarInputFormVO，并将单子改为编辑状态 | |
| transitbl.CarInputFormTransitbl.cancel | 语法 | public void CarInputFormcancel (CarInputFormVO vo) | |
| 前置条件 | 单子为编辑状态 | |
| 后置条件 | 删除一个CarInputFormVO | |
| transitbl.CarInputFormTransitbl.save | 语法 | public boolean CarInputFormsave (CarInputFormVO vo) | |
| 前置条件 | 单子为编辑状态 | |
| 后置条件 | 将单子改为草稿状态 | |
| transitbl.CarInputFormTransitbl.submit | 语法 | public boolean CarInputFormsubmit (CarInputFormVO vo) | |
| 前置条件 | 单子为草稿状态 | |
| 后置条件 | 将单子改为提交状态 | |
| transitbl.CarInputFormTransitbl.withdraw | 语法 | public boolean CarInputFormwithdraw (CarInputFormVO vo) | |
| 前置条件 | 单子为提交状态 | |
| 后置条件 | 将单子改为草稿状态 | |
| transitbl.CarOfficeFormTransitbl.create | 语法 | public CarOfficeFormVO CarOfficeFormcreate () | |
| 前置条件 | 无 | |
| 后置条件 | 增加一个CarOfficeFormVO，并将单子改为编辑状态 | |
| transitbl.CarOfficeFormTransitbl.cancel | 语法 | public void CarOfficeFormcancel (CarOfficeFormVO vo) | |
| 前置条件 | 单子为编辑状态 | |
| 后置条件 | 删除一个CarOfficeFormVO | |
| transitbl.CarOfficeFormTransitbl.save | 语法 | public boolean CarOfficeFormsave (CarOfficeFormVO vo) | |
| 前置条件 | 单子为编辑状态 | |
| 后置条件 | 将单子改为草稿状态 | |
| transitbl.CarOfficeFormTransitbl.submit | 语法 | public boolean CarOfficeFormsubmit (CarOfficeFormVO vo) | |
| 前置条件 | 单子为草稿状态 | |
| 后置条件 | 将单子改为提交状态 | |
| transitbl.CarOfficeFormTransitbl.withdraw | 语法 | public boolean CarOfficeFormwithdraw (CarOfficeFormVO vo) | |
| 前置条件 | 单子为提交状态 | |
| 后置条件 | 将单子改为草稿状态 | |
| transitbl.DeliveryFormTransitbl.create | 语法 | public DeliveryFormVO DeliveryFormcreate () | |
| 前置条件 | 无 | |
| 后置条件 | 增加一个DeliveryFormVO，并将单子改为编辑状态 | |
| transitbl.DeliveryFormTransitbl.cancel | 语法 | public void DeliveryFormcancel (DeliveryFormVO vo) | |
| 前置条件 | 单子为编辑状态 | |
| 后置条件 | 删除一个DeliveryFormVO | |
| transitbl.DeliveryFormTransitbl.save | 语法 | public boolean DeliveryFormsave (DeliveryFormVO vo) | |
| 前置条件 | 单子为编辑状态 | |
| 后置条件 | 将单子改为草稿状态 | |
| transitbl.DeliveryFormTransitbl.submit | 语法 | public boolean DeliveryFormsubmit (DeliveryFormVO vo) | |
| 前置条件 | 单子为草稿状态 | |
| 后置条件 | 将单子改为提交状态 | |
| transitbl.DeliveryFormTransitbl.withdraw | 语法 | public boolean DeliveryFormwithdraw (DeliveryFormVO vo) | |
| 前置条件 | 单子为提交状态 | |
| 后置条件 | 将单子改为草稿状态 | |
| transitbl.OfficeArrivalFormTransitbl.create | 语法 | public OfficeArrivalFormVO OfficeArrivalFormcreate () | |
| 前置条件 | 无 | |
| 后置条件 | 增加一个OfficeArrivalFormVO，并将单子改为编辑状态 | |
| transitbl.OfficeArrivalFormTransitbl.cancel | 语法 | public void OfficeArrivalFormcancel (OfficeArrivalFormVO vo) | |
| 前置条件 | 单子为编辑状态 | |
| 后置条件 | 删除一个OfficeArrivalFormVO | |
| transitbl.OfficeArrivalFormTransitbl.save | 语法 | public boolean OfficeArrivalFormsave (OfficeArrivalFormVO vo) | |
| 前置条件 | 单子为编辑状态 | |
| 后置条件 | 将单子改为草稿状态 | |
| transitbl.OfficeArrivalFormTransitbl.submit | 语法 | public boolean OfficeArrivalFormsubmit (OfficeArrivalFormVO vo) | |
| 前置条件 | 单子为草稿状态 | |
| 后置条件 | 将单子改为提交状态 | |
| transitbl.OfficeArrivalFormTransitbl.withdraw | 语法 | public boolean OfficeArrivalFormwithdraw (OfficeArrivalFormVO vo) | |
| 前置条件 | 单子为提交状态 | |
| 后置条件 | 将单子改为草稿状态 | |
| transitbl.OrderFormTransitbl.create | 语法 | public OrderFormVO OrderFormcreate () | |
| 前置条件 | 无 | |
| 后置条件 | 增加一个orderFormVO，并将单子改为编辑状态 | |
| transitbl.OrderFormTransitbl.cancel | 语法 | public void OrderFormcancel (OrderFormVO vo) | |
| 前置条件 | 单子为编辑状态 | |
| 后置条件 | 删除一个OrderFormVO | |
| transitbl.OrderFormTransitbl.save | 语法 | public boolean OrderFormsave (OrderFormVO vo) | |
| 前置条件 | 单子为编辑状态 | |
| 后置条件 | 将单子改为草稿状态 | |
| transitbl.OrderFormTransitbl.submit | 语法 | public boolean OrderFormsubmit (OrderFormVO vo) | |
| 前置条件 | 单子为草稿状态 | |
| 后置条件 | 将单子改为提交状态 | |
| transitbl.OrderFormTransitbl.withdraw | 语法 | public boolean OrderFormwithdraw (OrderFormVO vo) | |
| 前置条件 | 单子为提交状态 | |
| 后置条件 | 将单子改为草稿状态 | |
| transitbl.StoreArrivalFormTransitbl.create | 语法 | public StoreArrivalFormVO StoreArrivalFormcreate () | |
| 前置条件 | 无 | |
| 后置条件 | 增加一个StoreArrivalFormVO，并将单子改为编辑状态 | |
| transitbl.StoreArrivalFormTransitbl.cancel | 语法 | public void StoreArrivalFormcancel (StoreArrivalFormVO vo) | |
| 前置条件 | 单子为编辑状态 | |
| 后置条件 | 删除一个StoreArrivalFormVO | |
| transitbl.StoreArrivalFormTransitbl.save | 语法 | public boolean StoreArrivalFormsave (StoreArrivalFormVO vo) | |
| 前置条件 | 单子为编辑状态 | |
| 后置条件 | 将单子改为草稿状态 | |
| transitbl.StoreArrivalFormTransitbl.submit | 语法 | public boolean StoreArrivalFormsubmit (StoreArrivalFormVO vo) | |
| 前置条件 | 单子为草稿状态 | |
| 后置条件 | 将单子改为提交状态 | |
| transitbl.StoreArrivalFormTransitbl.withdraw | 语法 | public boolean StoreArrivalFormwithdraw (StoreArrivalFormVO vo) | |
| 前置条件 | 单子为提交状态 | |
| 后置条件 | 将单子改为草稿状态 | |
| transitbl.TransferFormTransitbl.create | 语法 | public TransferFormVO TransferFormcreate () | |
| 前置条件 | 无 | |
| 后置条件 | 增加一个TransferFormVO，并将单子改为编辑状态 | |
| transitbl.TransferFormTransitbl.cancel | 语法 | public void TransferFormcancel (TransferFormVO vo) | |
| 前置条件 | 单子为编辑状态 | |
| 后置条件 | 删除一个TransferFormVO | |
| transitbl.TransferFormTransitbl.save | 语法 | public boolean TransferFormsave (TransferFormVO vo) | |
| 前置条件 | 单子为编辑状态 | |
| 后置条件 | 将单子改为草稿状态 | |
| transitbl.TransferFormTransitbl.submit | 语法 | public boolean TransferFormsubmit (TransferFormVO vo) | |
| 前置条件 | 单子为草稿状态 | |
| 后置条件 | 将单子改为提交状态 | |
| transitbl.TransferFormTransitbl.withdraw | 语法 | public boolean TransferFormwithdraw (TransferFormVO vo) | |
| 前置条件 | 单子为提交状态 | |
| 后置条件 | 将单子改为草稿状态 | |
| **需要的服务（需接口）** | | | |
| DatabaseFactory.getTransitDatabase | | | 得到Transit数据库的服务的引用 |
| TransitDataService. OrderFormdata.insert(OrderFormPO po) | | | 插入单一持久化OrderFormPO对象 |
| TransitDataService. OrderFormdata.delete(OrderFormPO po) | | | 删除单一持久化OrderFormPO对象 |
| TransitDataService. OrderFormdata.update(OrderFormPO po) | | | 更新单一持久化OrderFormPO对象 |
| TransitDataService. OrderFormdata.find(long ID) | | | 根据订单号查找单一持久化OrderFormPO对象 |
| TransitDataService. OfficeArrivalFormdata.find(long NO) | | | 根据编号查找单一持久化OfficeArrivalFormPO对象 |
| TransitDataService. OfficeArrivalFormdata.insert(OfficeArrivalFormPO po) | | | 插入单一持久化OfficeArrivalFormPO对象 |
| TransitDataService. OfficeArrivalFormdata.delete(OfficeArrivalFormPO po) | | | 删除单一持久化OfficeArrivalFormPO对象 |
| TransitDataService. OfficeArrivalFormdata.update(OfficeArrivalFormPO po) | | | 更新单一持久化OfficeArrivalFormPO对象 |
| TransitDataService. CarOfficeFormdata.insert(CarOfficeFormPO po) | | | 插入单一持久化CarOfficeFormPO对象 |
| TransitDataService. CarOfficeFormdata.find(long NO) | | | 根据编号查找单一持久化CarOfficeFormPO对象 |
| TransitDataService. CarOfficeFormdata.delete(CarOfficeFormPO po) | | | 删除单一持久化CarOfficeFormPO对象 |
| TransitDataService. CarOfficeFormdata.update(CarOfficeFormPO po) | | | 更新单一持久化CarOfficeFormPO对象 |
| TransitDataService. DeliveryFormdata.find(DeliveryFormPO po) | | | 根据编号查找单一持久化DeliveryFormPO对象 |
| TransitDataService. DeliveryFormdata.insert(DeliveryFormPO po) | | | 插入单一持久化DeliveryFormPO对象 |
| TransitDataService. DeliveryFormdata.delete(DeliveryFormPO po) | | | 删除单一持久化DeliveryFormPO对象 |
| TransitDataService. DeliveryFormdata.update(DeliveryFormPO po) | | | 更新单一持久化DeliveryFormPO对象 |
| TransitDataService. StoreArrivalFormdata.find(long NO) | | | 根据编号查找单一持久化StoreArrivalFormPO对象 |
| TransitDataService. StoreArrivalFormdata.insert(StoreArrivalFormPO po) | | | 插入单一持久化StoreArrivalFormPO对象 |
| TransitDataService. StoreArrivalFormdata.delete(StoreArrivalFormPO po) | | | 删除单一持久化StoreArrivalFormPO对象 |
| TransitDataService. StoreArrivalFormdata.update(StoreArrivalFormPO po) | | | 更新单一持久化StoreArrivalFormPO对象 |
| TransitDataService. DeliveryFormPO. TransferFormdata.find(long NO) | | | 根据编号查找单一持久化TransferFormPO对象 |
| TransitDataService. TransferFormdata.insert(TransferFormPO po) | | | 插入单一持久化TransferFormPO对象 |
| TransitDataService. TransferFormdata.delete(TransferFormPO po) | | | 删除单一持久化TransferFormPO对象 |
| TransitDataService. TransferFormdata.update(TransferFormPO po) | | | 更新单一持久化TransferFormPO对象 |
| TransitDataService. CarInputFormdata.insert(CarInputFormPO po) | | | 插入单一持久化CarInputFormPO对象 |
| TransitDataService. CarInputFormdata.delete(CarInputFormPO po) | | | 删除单一持久化CarInputFormPO对象 |
| TransitDataService. CarInputFormdata.update(CarInputFormPO po) | | | 更新单一持久化CarInputFormPO对象 |
| TransitDataService. CarInputFormdata.find(long NO) | | | 根据编号查找单一持久化CarInputFormPO对象 |

#### 5.3.2.2 inventorybl模块

表5.3.2-2 inventorybl模块的接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| inventorybl.InvetoryInfoInventorybl. invetoryInfoSetWarningLimit | 语法 | public boolean invetoryInfoSetWarningLimit (double limit) |
| 前置条件 | 启动一个设置库存警戒值的任务 |
| 后置条件 | 在一个设置库存警戒值的任务中增加警戒值 |
| inventorybl. InvetoryInfoInventorybl.inventoryAlarm | 语法 | public Zone inventoryAlarm(InventoryVO vo) |
| 前置条件 | 启动一个库存报警的任务 |
| 后置条件 | 关闭一个库存报警任务 |
| inventorybl. InvetoryInfoInventorybl. invetoryAdjustment | 语法 | public boolean invetoryAdjustment (Zone zone) |
| 前置条件 | 库存报警了 |
| 后置条件 | 机动区改为库存报警的区 |
| inventorybl. InvetoryInfoInventorybl.inventoryCheck | 语法 | public InventoryInfoVO storageCheck(String startTime, String endTime) |
| 前置条件 | 启动一个查看库存任务 |
| 后置条件 | 关闭一个库存查看任务 |
| inventorybl. InvetoryInfoInventorybl.inventoryCounting | 语法 | public inventoryInfoVO inventoryCounting(Zone zone) |
| 前置条件 | 要求库存盘点 |
| 后置条件 | 可以导出Excel |

|  |  |  |
| --- | --- | --- |
| inventorybl.StorageInFormInventorybl.create | 语法 | public StorageInFormVO StorageInFormcreate () |
| 前置条件 | 无 |
| 后置条件 | 增加一个StorageInFormVO，并将单子改为编辑状态 |
| inventorybl.StorageInFormInventorybl.cancel | 语法 | public void StorageInFormcancel (StorageInFormVO vo) |
| 前置条件 | 单子为编辑状态 |
| 后置条件 | 删除一个StorageInFormVO |
| inventorybl.StorageInFormInventorybl.save | 语法 | public boolean StorageInFormsave (StorageInFormVO vo) |
| 前置条件 | 单子为编辑状态 |
| 后置条件 | 将单子改为草稿状态 |
| inventorybl.StorageInFormInventorybl.submit | 语法 | public boolean StorageInFormsubmit (StorageInFormVO vo) |
| 前置条件 | 单子为草稿状态 |
| 后置条件 | 将单子改为提交状态 |
| inventorybl.StorageInFormInventorybl.withdraw | 语法 | public boolean StorageInFormwithdraw (StorageInFormVO vo) |
| 前置条件 | 单子为提交状态 |
| 后置条件 | 将单子改为草稿状态 |
| inventorybl.StorageOutFormInventorybl.create | 语法 | public StorageOutFormVO StorageOutFormcreate () |
| 前置条件 | 无 |
| 后置条件 | 增加一个StorageOutFormVO，并将单子改为编辑状态 |
| inventorybl.StorageOutFormInventorybl.cancel | 语法 | public void StorageOutFormcancel (StorageOutFormVO vo) |
| 前置条件 | 单子为编辑状态 |
| 后置条件 | 删除一个StorageOutFormVO |
| inventorybl.StorageOutFormInventorybl.save | 语法 | public boolean StorageOutFormsave (StorageOutFormVO vo) |
| 前置条件 | 单子为编辑状态 |
| 后置条件 | 将单子改为草稿状态 |
| inventorybl.StorageOutFormInventorybl.submit | 语法 | public boolean StorageOutFormsubmit (StorageOutFormVO vo) |
| 前置条件 | 单子为草稿状态 |
| 后置条件 | 将单子改为提交状态 |
| inventorybl.StorageOutFormInventorybl.withdraw | 语法 | public boolean StorageOutFormwithdraw (StorageOutFormVO vo) |
| 前置条件 | 单子为提交状态 |
| 后置条件 | 将单子改为草稿状态 |

|  |  |
| --- | --- |
| **需要的服务（需接口）** | |
| DatabaseFactory.getInventoryDatabase | 得到inventor数据库的服务的引用 |
| inventoryDataService. inventoryInfoInventorydata.delete(inventoryInfoPO po) | 删除单一持久化inventoryInfoPO对象 |
| inventoryDataService. inventoryInfoInventorydata.add(inventoryInfoPO po) | 增加单一持久化inventoryInfoPO对象 |
| inventoryDataService. inventoryInfoInventorydata.update(inventoryInfoPO po) | 更新单一持久化inventoryInfoPO对象 |
| inventoryDataService. inventoryInfoInventorydata.find(long NO) | 查找单一持久化inventoryInfoPO对象 |
| inventoryDataService. inventoryInfodata. StorageOutFormInventorydata.insert(StorageOutFormPO po) | 插入单一持久化StorageOutFormPO对象 |
| inventoryDataService. StorageOutFormInventorydata.delete(StorageOutFormPO po) | 删除单一持久化StorageOutFormPO对象 |
| inventoryDataService. inventoryInfodata. StorageOutFormInventorydata.update(inventoryInfoPO po) | 更新单一持久化inventoryInfoPO对象 |
| inventoryDataService. StorageOutFormInventorydata.find(long NO) | 查找单一持久化StorageOutFormPO对象 |
| inventoryDataService. StorageInFormInventorydata.insert(StorageInFormPO po) | 插入单一持久化StorageInFormPO对象 |
| inventoryDataService. StorageInFormInventorydata.delete(StorageInFormPO po) | 删除单一持久化StorageInFormPO对象 |
| inventoryDataService. StorageInFormInventorydata.update(StorageInFormPO po) | 更新单一持久化StorageInFormPO对象 |
| inventoryDataService. StorageInFormInventorydata.find(long NO) | 根据单据编号查找单一持久化StorageInFormPO对象 |

#### 5.3.2.3 constbl模块

表5.3.2-3 constbl模块的接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| Constbl.ConstInfoConstbl.find | 语法 | public ConstVO find() |
| 前置条件 |  |
| 后置条件 | 返回一个ConstVO |
| Constbl.ConstInfoConstbl.update | 语法 | public ConstVO update() |
| 前置条件 |  |
| 后置条件 | 返回一个ConstVO |
| **需要的服务（需接口）** | | |
| DatabaseFactory.getConstDatabase | | 得到Const数据库的服务的引用 |
| ConstDataService.ConstInfoConstdata.update(constPO po) | | 更新单一持久化constPO对象 |
| ConstDataService. ConstInfoConstdata.find() | | 查找单一持久化constPO对象 |

#### 5.3.2.4 financebl模块

表5.3.2.4 financebl模块的接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| financebl.beginningAccountFinancebl.init | 语法 | public boolean initAccounts (int year, ArrayList< UserPO >,ArraList< carInfoPO > ,ArrayList< DriverInfoPO > ,ArrayList< AccountPO >) |
| 前置条件 | 要求期初建账 |
| 后置条件 | 生成新账目，并保存一份初始账目 |
| financebl.RecieptFormFinancebl.create | 语法 | public RecieptFormVO RecieptFormcreate () |
| 前置条件 | 无 |
| 后置条件 | 增加一个RecieptFormVO，并将单子改为编辑状态 |
| financebl.RecieptFormFinancebl.cancel | 语法 | public void RecieptFormcancel (RecieptFormVO vo) |
| 前置条件 | 单子为编辑状态 |
| 后置条件 | 删除一个RecieptFormVO |
| financebl.RecieptFormFinancebl.save | 语法 | public boolean RecieptFormsave (RecieptFormVO vo) |
| 前置条件 | 单子为编辑状态 |
| 后置条件 | 将单子改为草稿状态 |
| financebl.RecieptFormFinancebl.submit | 语法 | public boolean RecieptFormsubmit (RecieptFormVO vo) |
| 前置条件 | 单子为草稿状态 |
| 后置条件 | 将单子改为提交状态 |
| financebl.RecieptFormFinancebl.withdraw | 语法 | public boolean RecieptFormwithdraw (RecieptFormVO vo) |
| 前置条件 | 单子为提交状态 |
| 后置条件 | 将单子改为草稿状态 |
| financebl.PaymentFormFinancebl.create | 语法 | public PaymentFormVO PaymentFormcreate () |
| 前置条件 | 无 |
| 后置条件 | 增加一个PaymentFormVO，并将单子改为编辑状态 |
| financebl.PaymentFormFinancebl.cancel | 语法 | public void PaymentFormcancel (PaymentFormVO vo) |
| 前置条件 | 单子为编辑状态 |
| 后置条件 | 删除一个PaymentFormVO |
| financebl.PaymentFormFinancebl.save | 语法 | public boolean PaymentFormsave (PaymentFormVO vo) |
| 前置条件 | 单子为编辑状态 |
| 后置条件 | 将单子改为草稿状态 |
| financebl.PaymentFormFinancebl.submit | 语法 | public boolean PaymentFormsubmit (PaymentFormVO vo) |
| 前置条件 | 单子为草稿状态 |
| 后置条件 | 将单子改为提交状态 |
| financebl.PaymentFormFinancebl.withdraw | 语法 | public boolean PaymentFormwithdraw (PaymentFormVO vo) |
| 前置条件 | 单子为提交状态 |
| 后置条件 | 将单子改为草稿状态 |

|  |  |  |
| --- | --- | --- |
| financebl.RecieptFormFinancebl.create | 语法 | public RecieptFormVO RecieptFormcreate () |
| 前置条件 | 无 |
| 后置条件 | 增加一个RecieptFormVO，并将单子改为编辑状态 |
| financebl.RecieptFormFinancebl.cancel | 语法 | public void RecieptFormcancel (RecieptFormVO vo) |
| 前置条件 | 单子为编辑状态 |
| 后置条件 | 删除一个RecieptFormVO |
| financebl.RecieptFormFinancebl.save | 语法 | public boolean RecieptFormsave (RecieptFormVO vo) |
| 前置条件 | 单子为编辑状态 |
| 后置条件 | 将单子改为草稿状态 |
| financebl.RecieptFormFinancebl.submit | 语法 | public boolean RecieptFormsubmit (RecieptFormVO vo) |
| 前置条件 | 单子为草稿状态 |
| 后置条件 | 将单子改为提交状态 |
| financebl.RecieptFormFinancebl.withdraw | 语法 | public boolean RecieptFormwithdraw (RecieptFormVO vo) |
| 前置条件 | 单子为提交状态 |
| 后置条件 | 将单子改为草稿状态 |
| financebl.PaymentFormFinancebl.create | 语法 | public PaymentFormVO PaymentFormcreate () |
| 前置条件 | 无 |
| 后置条件 | 增加一个PaymentFormVO，并将单子改为编辑状态 |
| financebl.PaymentFormFinancebl.cancel | 语法 | public void PaymentFormcancel (PaymentFormVO vo) |
| 前置条件 | 单子为编辑状态 |
| 后置条件 | 删除一个PaymentFormVO |
| financebl.PaymentFormFinancebl.save | 语法 | public boolean PaymentFormsave (PaymentFormVO vo) |
| 前置条件 | 单子为编辑状态 |
| 后置条件 | 将单子改为草稿状态 |
| financebl.PaymentFormFinancebl.submit | 语法 | public boolean PaymentFormsubmit (PaymentFormVO vo) |
| 前置条件 | 单子为草稿状态 |
| 后置条件 | 将单子改为提交状态 |
| financebl.PaymentFormFinancebl.withdraw | 语法 | public boolean PaymentFormwithdraw (PaymentFormVO vo) |
| 前置条件 | 单子为提交状态 |
| 后置条件 | 将单子改为草稿状态 |

|  |  |  |
| --- | --- | --- |
| Financebl.AccountInfoFinancebl.create | 语法 | public AccountInfoVO create() |
| 前置条件 | 无 |
| 后置条件 | 增加一个AccountInfoVO对象 |
| Financebl.AccountInfoFinancebl.add | 语法 | public boolean add(AccountInfoVO) |
| 前置条件 | 不存在相同的VO对象 |
| 后置条件 | 无 |
| Financebl.AccountInfoFinancebl.delete | 语法 | public boolean delete(AccountInfoVO) |
| 前置条件 | 存在相同的VO对象 |
| 后置条件 | 删除一个AccountInfoVO对象 |
| Financebl.AccountInfoFinancebl.find | 语法 | public AccountInfoVO find(long bankAccount) |
| 前置条件 | 输入正确的银行账户名 |
| 后置条件 | 返回一个AccountInfoVO对象 |
| financebl.CostPayChartFinancebl. costPayChartCreat | 语法 | public CostPayChartVO costPayChartCreat() |
| 前置条件 | 无 |
| 后置条件 | 增加一个VO |
| financebl.CostPayChartFinancebl. costPayChartAdd | 语法 | public boolean costPayChartCreat(CostPayChartVO vo) |
| 前置条件 | 不存在相同NO的vo |
| 后置条件 | 无 |
| financebl.CostPayChartFinancebl. costPayChartCheck | 语法 | public CostPayChartVO costPayChartCreat(long NO) |
| 前置条件 | 无 |
| 后置条件 | 返回一个VO |
| financebl.ProfitChartFinancebl. profitChartCreat | 语法 | public ProfitChartVO profitChartCreat () |
| 前置条件 | 无 |
| 后置条件 | 增加一个vo |
| financebl.ProfitChartFinancebl. profitChartAdd | 语法 | public boolean profitChartAdd (ProfitVO vo) |
| 前置条件 | 不存在相同NO的vo |
| 后置条件 | 无 |
| financebl.ProfitChartFinancebl. profitChartCheck | 语法 | public profitChartVO void profitChartCheck (long NO) |
| 前置条件 | 输入正确的NO |
| 后置条件 | 返回一个vo |
| **需要的服务（需接口）** | | |
| DatabaseFactory.getFinanceDatabase | 得到Finance数据库的服务的引用 | |
| FinanceDataService. receiptFormFinancedata.find(long NO) | 查找单一持久化对象 | |
| FinanceDataService. receiptFormFinancedata.insert(receiptFormPO po) | 插入单一持久化对象 | |
| FinanceDataService. receiptFormFinancedata.delete(receiptFormPO po) | 删除单一持久化对象 | |
| FinanceDataService. FinanceDataService. receiptFormFinancedata.update(receiptFormPO po) | 更新单一持久化对象 | |
| FinanceDataService. paymentFormFinancedata.find(long NO) | 查找单一持久化对象 | |
| FinanceDataService. paymentFormFinancedata.insert(paymentFormPO po) | 插入单一持久化对象 | |
| FinanceDataService. paymentFormFinancedata.delete(paymentFormPO po) | 删除单一持久化对象 | |
| FinanceDataService. paymentFormFinancedata.update(paymentFormPO po) | 更新单一持久化对象 | |
| FinanceDatabl. costPayChartFinancedata.find(long NO) | 查找单一持久化对象 | |
| FinanceDataService. costPayChartFinancedata.insert(costPayChartPO po) | 插入单一持久化对象 | |
| FinanceDataService. costPayChartFinancedata.delete(costPayChartPO po) | 删除单一持久化对象 | |
| FinanceDataService. costPayChartFinancedata.update(costPayChartPO po) | 更新单一持久化对象 | |
| FinanceDataService. accountInfoCorparationdata.find (long bankAccount) | 查找单一持久化对象 | |
| FinanceDataService. accountInfoCorparationdata.insert(accountPO po) | 插入单一持久化对象 | |
| FinanceDataService. accountInfoCorparationdata.delete(accountPO po) | 删除单一持久化对象 | |

#### 5.3.2.5 cororationbl模块

|  |
| --- |
| **提供的服务（供接口）** |

|  |  |  |
| --- | --- | --- |
| Corporationbl.CarInfocorporationbl.create | 语法 | public CarInfoVO create() |
| 前置条件 | 无 |
| 后置条件 | 增加一个CarInfoVO对象 |
| Corporationbl.CarInfocorporationbl.add | 语法 | public boolean add(CarInfoVO vo) |
| 前置条件 | 不存在相同的VO对象 |
| 后置条件 | 无 |
| Corporationbl.CarInfocorporationbl.delete | 语法 | public boolean delete(CarInfoVO vo) |
| 前置条件 | 存在相同的VO对象 |
| 后置条件 | 删除一个CarInfoVO对象 |
| Corporationbl.CarInfocorporationbl.find | 语法 | public CarInfoVO find(String carNumber) |
| 前置条件 | 输入正确的银行账户名 |
| 后置条件 | 返回一个CarInfoVO对象 |
| Corporationbl.CarInfocorporationbl.update | 语法 | public CarInfoVO update() |
| 前置条件 |  |
| 后置条件 | 返回一个CarInfoVO |
| Corporationbl.DriverInfocorporationbl.create | 语法 | public DriverInfoVO create() |
| 前置条件 | 无 |
| 后置条件 | 增加一个DriverInfoVO对象 |
| Corporationbl.DriverInfocorporationbl.add | 语法 | public boolean add(DriverInfoVO vo) |
| 前置条件 | 不存在相同的VO对象 |
| 后置条件 | 无 |
| Corporationbl.DriverInfocorporationbl.delete | 语法 | public boolean delete(DriverInfoVO vo) |
| 前置条件 | 存在相同的VO对象 |
| 后置条件 | 删除一个DriverInfoVO对象 |
| Corporationbl.DriverInfocorporationbl.find | 语法 | public DriverInfoVO find(long driverNumber) |
| 前置条件 | 输入正确的银行账户名 |
| 后置条件 | 返回一个DriverInfoVO对象 |
| Corporationbl.DriverInfocorporationbl.update | 语法 | public DriverInfoVO update() |
| 前置条件 |  |
| 后置条件 | 返回一个DriverInfoVO |

|  |  |
| --- | --- |
| **需要的服务（需接口）** | |
| DatabaseFactory.getCoporationDatabase | 得到coporation数据库的服务的引用 |
| CorporationDataService. carInfoCorparationdata.delete(carInfoPO po) | 删除单一持久化对象 |
| CorporationDataService. carInfoCorparationdata.update(carInfoPO po) | 更新单一持久化对象 |
| CorporationDataService. carInfoCorparationdata.insert(carInfoPO po) | 更新单一持久化对象 |
| CorporationDataService. carInfoCorparationdata.find(long carNumber) | 更新单一持久化对象 |
| CorporationDataService. dariverInfoCorparationdata.find(long driverNumber) | 查找单一持久化对象 |
| CorporationDataService. dariverInfoCorparationdata.update(DriverInfoPO po) | 更新单一持久化对象 |
| CorporationDataService. dariverInfoCorparationdata.delete(DriverInfoPO po) | 删除单一持久化对象 |
| CorporationDataService. dariverInfoCorparationdata.update(DriverInfoPO po) | 更新单一持久化对象 |

**表5.2.3.5** cororationbl**模块的接口规范**

#### 5.3.2.6 userbl模块

**表5.2.3.6** userbl**模块的接口规范**

|  |
| --- |
| **提供的服务（供接口）** |

|  |  |  |
| --- | --- | --- |
| Userbl.UserInfoUserbl.create | 语法 | public UserInfoVO create() |
| 前置条件 | 无 |
| 后置条件 | 增加一个UserInfoVO对象 |
| Userbl.UserInfoUserbl.add | 语法 | public boolean add(UserInfoVO vo) |
| 前置条件 | 不存在相同的VO对象 |
| 后置条件 | 无 |
| Userbl.UserInfoUserbl.delete | 语法 | public boolean delete(UserInfoVO vo) |
| 前置条件 | 存在相同的VO对象 |
| 后置条件 | 删除一个UserInfoVO对象 |
| Userbl.UserInfoUserbl.find | 语法 | public UserInfoVO find(long userAccount) |
| 前置条件 | 输入正确的银行账户名 |
| 后置条件 | 返回一个UserInfoVO对象 |
| Userbl.UserInfoUserbl.login | 语法 | public boolean login(long userAccount, 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password返回相应的值 |
| Userbl.UserInfoUserbl.update | 语法 | public UserInfoVO update() |
| 前置条件 | 无 |
| 后置条件 | 返回一个UserInfoVO |

|  |  |  |
| --- | --- | --- |
| **需要的服务（需接口）** | | |
| DatabaseFactory.getUserDatabase | 得到user数据库的服务的引用 | |
| userDataService. UserInfodataService.insert(UserInfoPO po) | 插入单一持久化UserInfoPO对象 |
| TransitDataService. UserInfodataService.delete(UserInfoPO po) | 删除单一持久化UserInfoPO对象 |
| TransitDataService. UserInfodataService.update(UserInfoPO po) | 更新单一持久化UserInfoPO对象 |
| TransitDataService. UserInfodataService.find(long userAccount) | 根据userAccount查找单一持久化UserInfoPO对象 |

#### 5.3.2.7 formapprovalbl模块

**表5.2.3.7 formapproval模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| Formapprovalbl.OderFormbl.judge | 语法 | public boolean judge (OrderFormVO vo) |
| 前置条件 | 启动一个审批寄件单任务 |
| 后置条件 | 更改寄件单状态为已审批，更改寄件单货运状态 |
| Formapprovalbl.OfficeArrivalFormbl.judge | 语法 | public boolean judge (OfficeArrivalFormVO vo) |
| 前置条件 | 启动一个审批营业厅到达单任务 |
| 后置条件 | 更改营业厅到达单状态为已审批，更改寄件单货运状态 |
| Formapprovalbl.CarOfficeFormbl.judge | 语法 | public boolean judge (CarOfficeFormVO vo) |
| 前置条件 | 启动一个审批营业厅装车单任务 |
| 后置条件 | 更改营业厅装车单状态为已审批 |
| Formapprovalbl.DeliveryFormbl.judge | 语法 | public boolean judge (DeliveryFormVO vo) |
| 前置条件 | 启动一个审批派件单任务 |
| 后置条件 | 更派件单状态为已审批，更改寄件单货运状态 |
| Formapprovalbl.StoreArrivalFormbl.judge | 语法 | public boolean judge (StoreArrivalFormVO vo) |
| 前置条件 | 启动一个审批中转中心到达单任务 |
| 后置条件 | 更改中转中心到达单状态为已审批，更改寄件单货运状态 |
| Formapprovalbl.TransferFormbl.judge | 语法 | public boolean judge (TransferFormVO vo) |
| 前置条件 | 启动一个审批中转单任务 |
| 后置条件 | 更改中转单状态为已审批 |
| Formapprovalbl.CarInputFormbl.judge | 语法 | public boolean judge (CarInputFormVO vo) |
| 前置条件 | 启动一个审批中转中心装车单任务 |
| 后置条件 | 更改中转中心装车单状态为已审批 |
| Formapprovalbl.StorageOutForm.judge | 语法 | public boolean judge (StorageOutFormVO vo) |
| 前置条件 | 启动一个审批出库单任务 |
| 后置条件 | 更改出库单状态为已审批 |
| Formapprovalbl.StorageInFormbl.judge | 语法 | public boolean judge (StorageInFormVO vo) |
| 前置条件 | 启动一个审批入库单单任务 |
| 后置条件 | 更改入库单状态为已审批 |
| Formapprovalbl.PaymentFormbl.judge | 语法 | public boolean judge (PaymentFormVO vo) |
| 前置条件 | 启动一个审批付款单任务 |
| 后置条件 | 更改付款单状态为已审批 |
| Formapprovalbl.ReceiptFormbl.judge | 语法 | public boolean judge (ReceiptFormVO vo) |
| 前置条件 | 启动一个审批收款单任务 |
| 后置条件 | 更改收款单状态为已审批，更改寄件单货运状态 |

## 5.4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，包括对持久化数据的增、删、改、查。Transity业务逻辑需要的服务由TransityDataService接口提供。数据层模块的描述具体如图5.4-1所示。

图5.4-1 数据层模块的描述

### 5.4.1数据层模块的职责

数据层模块的职责如表5.4.1-1所示。

表5.4.1-1 数据层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| DataService | 持久化数据库的接口，提供集体载入，集体保存、增、删、改、查服务 |
| DataServiceSerializableImpl | 基于序列化文件的持久化数据库的接口，提供集体载入、集体保存、增、删、改、查服务 |

### 5.4.2数据层模块的接口规范

#### 5.4.2.1 transit模块

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| transitdataservice. OrderFormdata.find | 语法 | public OrderFormPO find(long ID) throws RemoteException; |
| 前置条件 | 输入正确的ID |
| 后置条件 | 按订单号返回相应的OrderFormPO结果 |
| transitdataservice. OrderFormdata.insert | 语法 | public void insert(OrderFormPO po) throws RemoteException; |
| 前置条件 | 同样的po在Mapper中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| transitdataservice. OrderFormdata.delete | 语法 | public void delete(long ID) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 删除一个po |
| transitdataservice. OrderFormdata.update | 语法 | public void update(OrderFormPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 更新一个po |

|  |  |  |
| --- | --- | --- |
| transitdataservice. OfficeArrivalFormdata .find | 语法 | public OfficeArrivalFormPO find(long NO) throws RemoteException; |
| 前置条件 | 输入正确的NO |
| 后置条件 | 按类型查找返回相应的transitPO结果 |
| transitdataservice. OfficeArrivalFormdata .insert | 语法 | public void insert(OfficeArrivalFormPO po) throws RemoteException; |
| 前置条件 | 同样的po在Mapper中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| transitdataservice. OfficeArrivalFormdata .delete | 语法 | public void delete(OfficeArrivalFormPO po) throws RemoteException; |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 删除一个po |
| transitdataservice. OfficeArrivalFormdata.update | 语法 | public void update(OfficeArrivalFormPO po) throws RemoteException; |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 更新一个po |

|  |  |  |
| --- | --- | --- |
| Transitdataservice. CarOfficedata .find | 语法 | public CarOfficePO find(long NO) throws RemoteException; |
| 前置条件 | 输入正确的NO |
| 后置条件 | 按类型查找返回相应的CarOfficePO结果 |
| transitdataservice. CarOfficedata .insert | 语法 | public void insert(CarOfficePO po) throws RemoteException; |
| 前置条件 | 同样的po在Mapper中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| transitdataservice. CarOfficedata .delete | 语法 | public void delete(CarOfficePO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 删除一个po |
| transitdataservice. CarOfficedata .update | 语法 | public void update(CarOfficePO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 更新一个po |
| transitdataservice. DeliveryFormdata .find | 语法 | public DeliveryFormPO find(long NO) throws RemoteException; |
| 前置条件 | 输入正确的NO |
| 后置条件 | 按类型查找返回相应的DeliveryFormPO结果 |
| transitdataservice. DeliveryFormdata .insert | 语法 | public void insert(DeliveryFormPO po) throws RemoteException; |
| 前置条件 | 同样的po在Mapper中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| transitdataservice. DeliveryFormdata .delete | 语法 | public void delete(DeliveryFormPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 删除一个po |
| transitdataservice. DeliveryFormdata .update | 语法 | public void update(DeliveryFormPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 更新一个po |
| transitdataservice. StoreArrivalFormPdata.find | 语法 | public StoreArrivalFormPO find(long NO) throws RemoteException; |
| 前置条件 | 输入正确的NO |
| 后置条件 | 按类型查找返回相应的DeliveryFormPO结果 |
| transitdataservice. StoreArrivalFormdata .insert | 语法 | public void insert(DeliveryFormPO po) throws RemoteException; |
| 前置条件 | 同样的po在Mapper中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| transitdataservice. StoreArrivalFormdata .delete | 语法 | public void delete(long NO) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 删除一个po |
| transitdataservice. StoreArrivalFormdata .update | 语法 | public void update(StoreArrivalFormPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 更新一个po |
| transitdataservice. TransferFormdata .find | 语法 | public TransferFormPO find(long NO) throws RemoteException; |
| 前置条件 | 输入正确的NO |
| 后置条件 | 按类型查找返回相应的TransferFormPO结果 |
| transitdataservice. TransferFormdata. insert | 语法 | public void insert(TransferFormPO po) throws RemoteException; |
| 前置条件 | 同样的po在Mapper中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| transitdataservice. TransferFormdata .delete | 语法 | public void delete(TransferFormPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 删除一个po |
| transitdataservice. TransferFormdata. update | 语法 | public void update(TransferFormPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 更新一个po |
| transitdataservice. CarInputFormdata .find | 语法 | public CarInputFormPO find(TransferFormPO po) throws RemoteException; |
| 前置条件 | 输入正确的NO |
| 后置条件 | 按类型查找返回相应的CarInputFormPO结果 |
| transitdataservice. CarInputFormdata .insert | 语法 | public void insert(CarInputFormPO po) throws RemoteException; |
| 前置条件 | 同样的po在Mapper中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| transitdataservice. CarInputFormdata .delete | 语法 | public void delete(CarInputFormPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 删除一个po |
| transitdataservice. CarInputFormdata.update | 语法 | public void update(CarInputFormPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 更新一个po |

#### 5.4.2.2 inventory模块

|  |  |  |
| --- | --- | --- |
| inventorydataservice. StorageInFormdata .find | 语法 | public StorageInFormdata find(long NO) throws RemoteException; |
| 前置条件 | 输入正确的NO |
| 后置条件 | 按类型查找返回相应的StorageInFormdata结果 |
| inventorydataservice. StorageInFormdata .insert | 语法 | public void insert(StorageInFormdata po) throws RemoteException; |
| 前置条件 | 同样的po在Mapper中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| inventorydataservice. StorageInFormdata .delete | 语法 | public void delete(StorageInFormdata po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 删除一个po |
| inventorydataservice. StorageInFormdata .update | 语法 | public void update(StorageInFormPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 更新一个po |
| inventorydataservice. StorageOutFromdata .find | 语法 | public StorageOutFromPO find(long NO) throws RemoteException; |
| 前置条件 | 输入正确的NO |
| 后置条件 | 按类型查找返回相应的StorageOutFromPO结果 |
| inventorydataservice. StorageOutFromdata .insert | 语法 | public void insert(StorageOutFromPO po) throws RemoteException; |
| 前置条件 | 同样的po在Mapper中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| inventorydataservice. StorageOutFromdata .delete | 语法 | public void delete(StorageOutFromPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 删除一个po |
| inventorydataservice. StorageOutFromdata .update | 语法 | public void update(StorageOutFromPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 更新一个po |
| inventorydataservice. InventoryInfodata  .find | 语法 | public InventoryInfoPO find() throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 按类型查找返回相应的InventoryInfoPO结果 |
| inventorydataservice. InventoryInfodata  .update | 语法 | public void update(InventoryInfoPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 更新一个po |

#### 5.4.2.3 finance模块

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| financedataservice. BeginningAccountdata.  find | 语法 | public BeginningAccountPO find(int year) throws RemoteException; |
| 前置条件 | 输入正确的年份 |
| 后置条件 | 按year查找返回相应的BeginningAccountPO结果 |
| financedataservice. BeginningAccountdata.insert | 语法 | public void insert(BeginningAccountPO po) throws RemoteException; |
| 前置条件 | year一样的po在Mapper中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| financedataservice. BeginningAccountdata.delete | 语法 | public void delete(BeginningAccountPO po) throws RemoteException |
| 前置条件 | 在数据库中存在year值相同的po |
| 后置条件 | 删除一个po |
| financedataservice. BeginningAccountdata.update | 语法 | public void update(BeginningAccountPO po) throws RemoteException |
| 前置条件 | 在数据库中存在year值相同的po |
| 后置条件 | 更新一个po |
| financedataservice. AccountFinancedata  .find | 语法 | public accountPO find(long bankAccount) throws RemoteException; |
| 前置条件 | 输入正确的bankAccount |
| 后置条件 | 按bankAccount返回相应的accountPO结果 |
| financedataservice. AccountFinancedata  .insert | 语法 | public void insert(AccountPO po) throws RemoteException; |
| 前置条件 | 同样的po在Mapper中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| financedataservice. AccountFinancedata  .delete | 语法 | public void delete(AccountPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 删除一个po |
| financedataservice. AccountFinancedata  .update | 语法 | public void update(AccountPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 更新一个po |
| financedataservice. CostPayChartdata  .find | 语法 | public CostPayChartPO find(long NO) throws RemoteException; |
| 前置条件 | 输入正确的NO |
| 后置条件 | 按NO返回相应的CostPayChartPO结果 |
| financedataservice. CostPayChartdata.  insert | 语法 | public void insert(CostPayChartPO po) throws RemoteException; |
| 前置条件 | 同样的po在Mapper中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| financedataservice. CostPayChartdata  .delete | 语法 | public void delete(CostPayChartPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 删除一个po |
| financedataservice. CostPayChartdata  .update | 语法 | public void update(CostPayChartPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 更新一个po |
| financedataservice. PaymentFormdata  .find | 语法 | public PaymentFormPO find(long NO) throws RemoteException; |
| 前置条件 | 输入正确的NO |
| 后置条件 | 按NO返回相应的PaymentFormPO结果 |
| financedataservice. PaymentFormdata  .insert | 语法 | public void insert(PaymentFormPO po) throws RemoteException; |
| 前置条件 | 同样的po在Mapper中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| financedataservice. PaymentFormdata  .delete | 语法 | public void delete(PaymentFormPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 删除一个po |
| financedataservice. PaymentFormdata  .update | 语法 | public void update(PaymentFormPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 更新一个po |
| financedataservice. ReceiptFormdata  .find | 语法 | public ReceiptFormPO find(long BO) throws RemoteException; |
| 前置条件 | 输入正确的NO |
| 后置条件 | 按NO返回相应的ReceiptFormPO结果 |
| financedataservice. ReceiptFormdata  .insert | 语法 | public void insert(ReceiptFormPO po) throws RemoteException; |
| 前置条件 | 同样的po在Mapper中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| financedataservice. ReceiptFormdata  .delete | 语法 | public void delete(ReceiptFormPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 删除一个po |
| financedataservice. ReceiptFormdata  .update | 语法 | public void update(ReceiptFormPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 更新一个po |

#### 5.4.2.4 userdata模块

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| userdataservice.  UserInfodata.find | 语法 | public userPO find(long userAccount) throws RemoteException; |
| 前置条件 | 输入正确的userAccount |
| 后置条件 | 按userAccount查找返回相应的userPO结果 |
| userdataservice. UserInfodata.insert | 语法 | public void insert(UserPO po) throws RemoteException; |
| 前置条件 | 同样userAccount的po在Mapper中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| userdataservice. UserInfodata.delete | 语法 | public void delete(UserPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样userAccount的po |
| 后置条件 | 删除一个po |
| userdataservice. UserInfodata.update | 语法 | public void update(UserPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样userAccount的po |
| 后置条件 | 更新一个po |

#### 5.4.2.5 corporation模块

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| corporationdataservice. CarInfoCorporationdata.find | 语法 | Public CarInfoPO find(String carNumber) throws RemoteException; |
| 前置条件 | 输入正确的carNumber |
| 后置条件 | 返回carInfoPO |
| corporationdataservice. CarInfoCorporationdata.update | 语法 | public void update(CarInfoPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 更新po |
| corporationdataservice. CarInfoCorporationdata. insert | 语法 | public void insert(CarInfoPO po) throws RemoteException; |
| 前置条件 | 同样carNumber的po在Mapper中不存在 |
| 后置条件 | 返回driverInfoPO |
| corporationdataservice. CarInfoCorporationdata.delete | 语法 | public void delete(CarInfoPO po) throws RemoteException; |
| 前置条件 | 在数据库中存在同样carNumber的po |
| 后置条件 | 删除po |
| corporationdataservice. DriverInfoCorporationdata.find | 语法 | public DriverInfoPO find(long driveNumber) throws RemoteException; |
| 前置条件 | 输入正确的driveNumber |
| 后置条件 | 返回DriverInfoPO |
| corporationdataservice. DriverInfoCorporationdata.update | 语法 | public void update(DriverInfoPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 更新一个po |
| corporationdataservice. DriverInfoCorporationdata.insert | 语法 | public void insert(DriverInfoPO po) throws RemoteException; |
| 前置条件 | 同样driveNumber的po在Mapper中不存在 |
| 后置条件 | 返回driverInfoPO |
| corporationdataservice. DriverInfoCorporationdata.update | 语法 | public void update(DriverInfoPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 更新一个po |

#### 5.4.2.6 const模块

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| constdataservice.Constdata.update | 语法 | public void update(ConstPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样的po |
| 后置条件 | 更新ConstPO |
| constDataService.Constdata.find | 语法 | public ConstPO find() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回一个ConstPO |

# 6.信息视角

## 6.1数据持久化对象

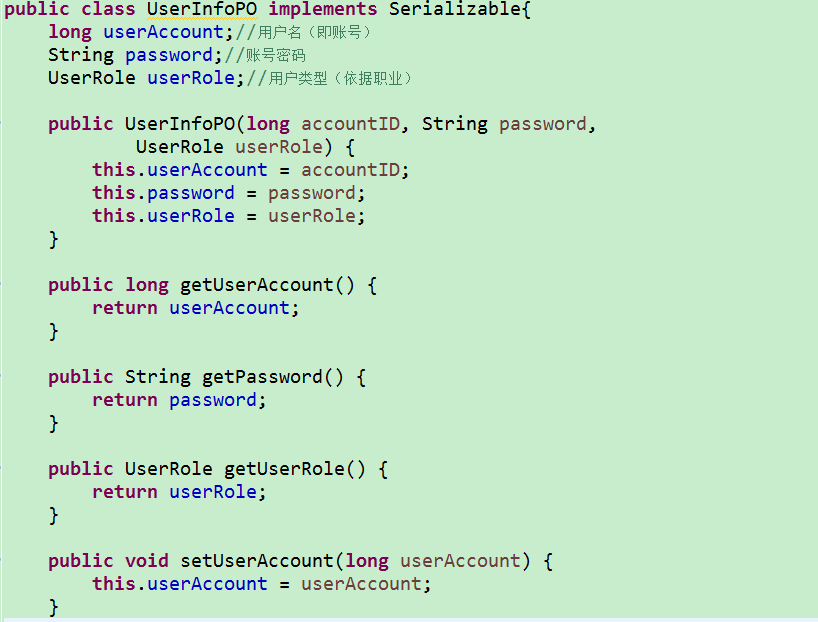
序列化数据保存在.ser文件中。

系统的PO类就是对应的相关的实体类，在此只做简单的介绍，如表6.1-1所示。

表6.1-1 PO类简单介绍

|  |  |
| --- | --- |
| 类名 | 包含的属性 |
| UserPO | 用户名(即账号)；  用户类型；  密码； |
| constPO | 薪资标准信息；  距离价格常量； |
| formapprovalPO | 待审批和已审批单据数量 |
| OrderFormPO | 寄件单：  寄件人姓名、住址、单位、电话、手机；  收件人姓名、住址、单位、电话、手机；  托运货物信息（原件数、实际重量、体积、内件品名）；  快递种类（经济快递、标准快递、特快专递）；  包装费（纸箱5元、木箱10元、快递袋1元）；  费用合计（自动计算，运费+包装费）；  订单条形码号（10位数）；  揽件日期；  货运状态； |
| OfficeArrivalFormPO | 营业厅到达单：  到达日期；  中转单编号；  出发地；  货物到达状态（损坏、完整、丢失）； |
| CarOfficeFormPO | 装车单：  装车日期；  本营业厅编号（025城市编号+000鼓楼营业厅）；  汽运编号（营业厅编号+20150925日期+00000编码）；  到达地（本地中转中心或者其他营业厅）；  车辆代号；  监装员；  押运员；  本次装箱所有订单条形码号；  运费（运费根据出发地和目的地自动生成）； |
| DeliveryFormPO | 派件单：  到达日期；  托运订单条形码号；  派送员； |
| InventoryInfoPO | 库存警戒值；  空仓库状态；  仓库中每个位置的信息； |
| TransferFormPO | 中转单：  装车日期；  本中转中中转单编号；  运输工具；  出发地；  到达地；  货柜号；  监装员；  本次装箱所有托运单号；  运费； |
| StoreArrivalFormPO | 中转中心到达单：  中转中心编号；  到达时间；  中转单编号；  出发地；  货物到达状态（损坏、完整、丢失）； |
| CarInputFormPO | 中转中心装车单  装车日期；  汽运编号（中转中心编号+20150925日期+0000000编码）；  到达地（营业厅）；  车辆代号；  监装员；  押运员；  本次装箱所有订单条形码号；  运费（运费根据出发地和目的地自动生成）； |
| StorageOutFormPO | 出库单：  订单编号；  出库日期；  目的地；  转运形式；  中转单编号或汽运编号； |
| StorageInFormPO | 入库单：  订单号；  入库日期；  目的地；  区号、排号、架号、位号； |
| AccountPO | 账号；  密码；  余额； |
| BeginningAccountPO | 建立年份；  期初建账信息:  当前机构人员账户信息；  车辆信息；  库存信息；  银行账户信息（名称、余额）； |
| ProfitChatPO | 成本收益表：  总收入；  总支出；  总利润（总利润=总收入-总支出）； |
| CostPayChartPO | 经营情况表：  包含选择日期区间内的所有付款单和收款单信息 |
| DriverInfoPO | 司机编号（城市编号+营业厅编号+三位数字）；  姓名；  出生日期；  身份证号；  性别；  手机；  行驶证期限； |
| CarInfoPO | 车辆代号（城市编号+营业厅编号+三位数字）；  车牌号；  发动机号；  底盘号；  购买时间；  服役时间； |
| PaymentFormPO | 付款日期；  付款金额；  付款账号；  条目、备注（其中条目包括以下几个类型：租金、运费、人员工资、奖励）； |
| ReceiptFormPO | 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； |

### 6.1.1持久化对象UserInfoPO



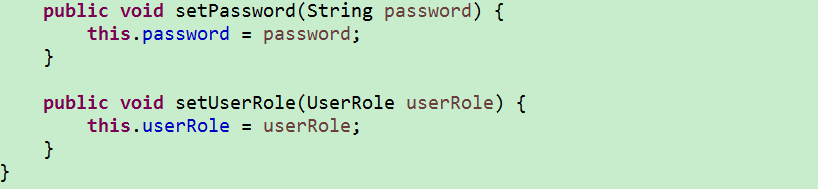


图6.1-1 持久化对象UserPO的定义

### 6.1.2持久化对象CarInfoPO

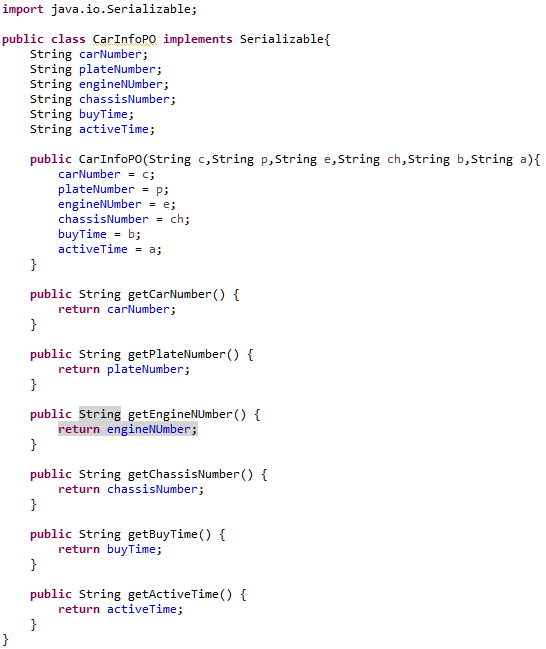
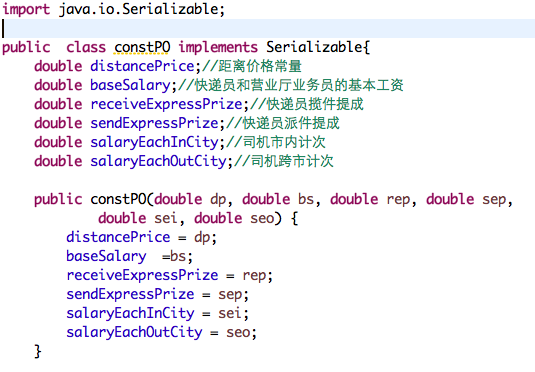
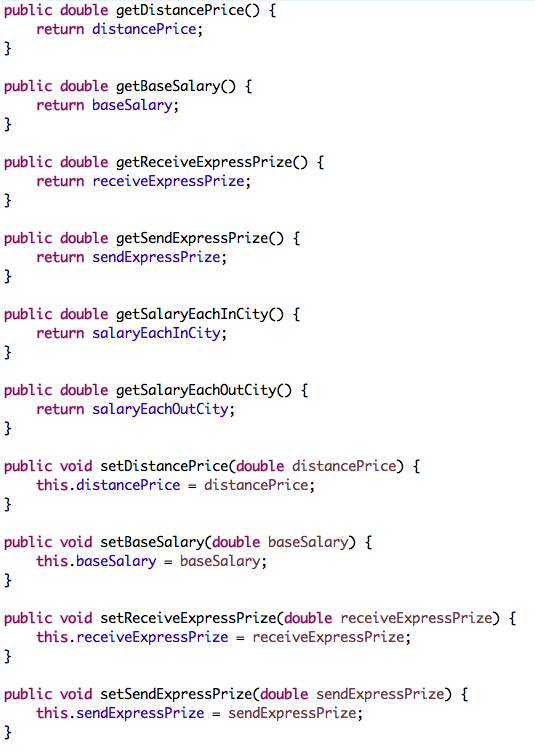


图6.1-21持久化对象carInfoPO的定义

### 6.1.3持久化对象ConstPO





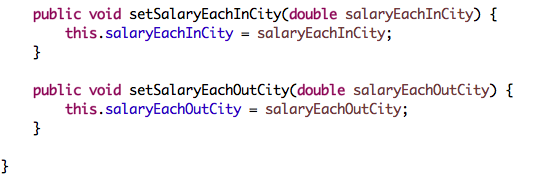


图6.1-3 持久化对象constPO的定义

### 6.1.5持久化对象OrderFormPO











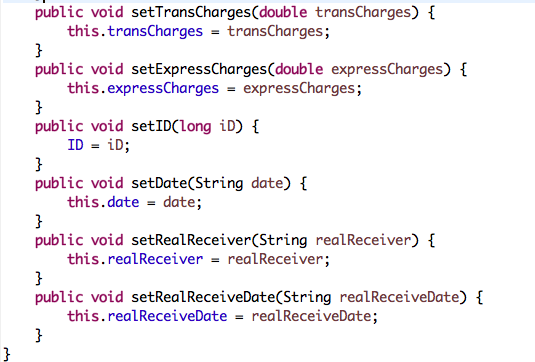


图6.1-5持久化对象OrderFormPO的定义

### 6.1.6持久化对象OfficeArrivalFormPO





图6.1-6持久化对象OfficeArrivalFormPO的定义

### 6.1.7持久化对象CarOfficeFormPO





图6.1-7持久化对象CarOfficePO的定义

### 6.1.8持久化对象DeliveryFormPO

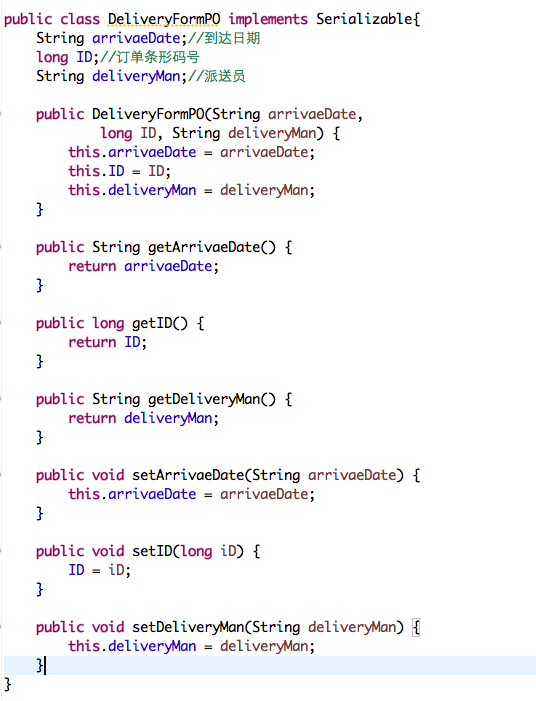


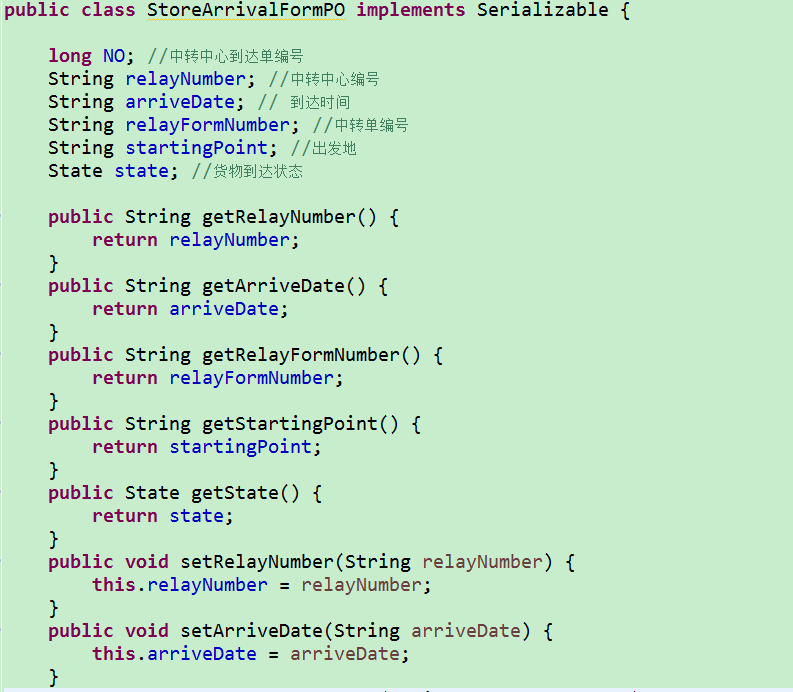
图6.1-8持久化对象DeliveryFormPO的定义

### 6.1.9持久化对象InventoryInfoPO



图6.1-9 持久化对象InventoryInfoPO的定义

### 6.1.10持久化对象StoreArrivalFormPO



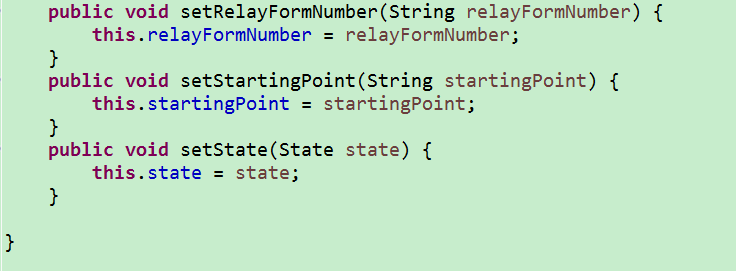
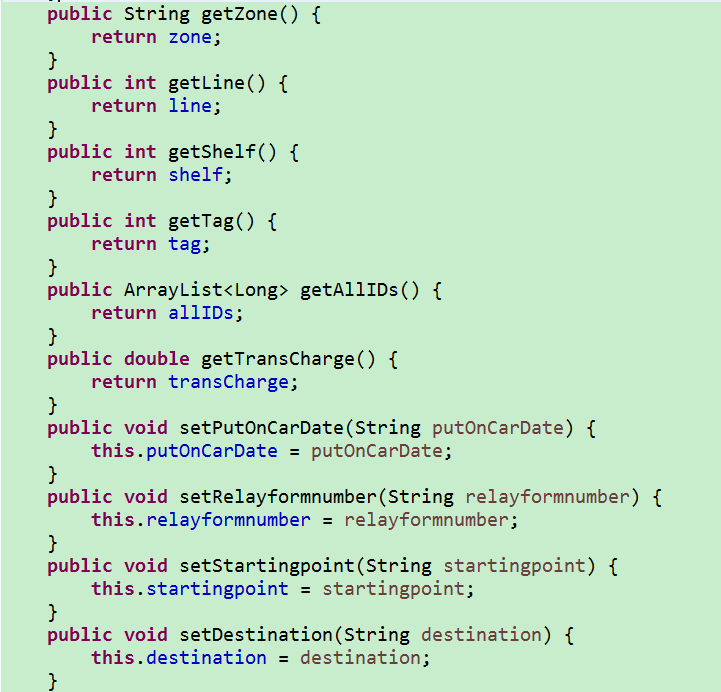


图6.1-10 持久化对象StoreArrivalFormPO的定义

### 6.1.11持久化对象TransferFormPO





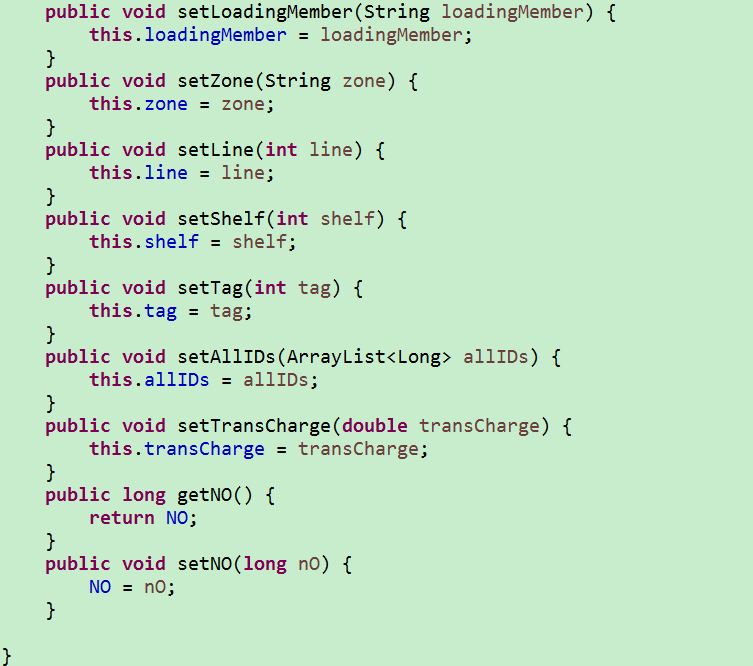
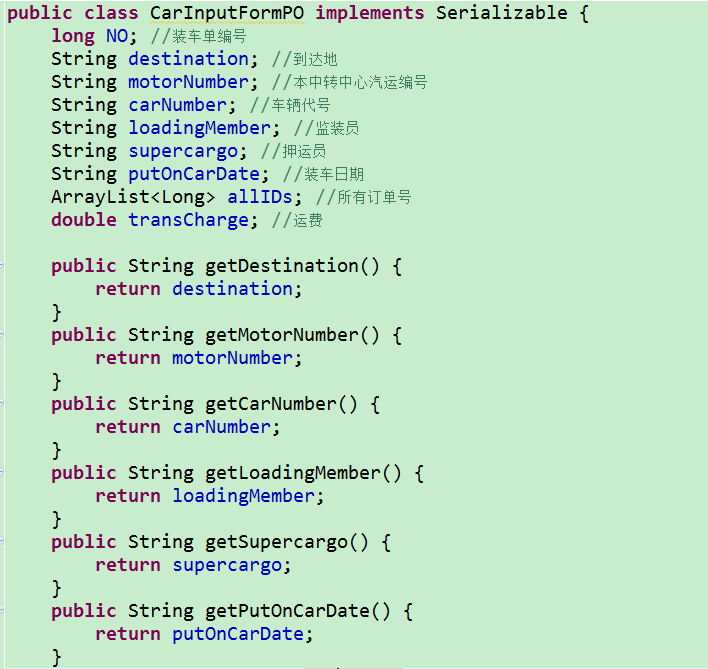
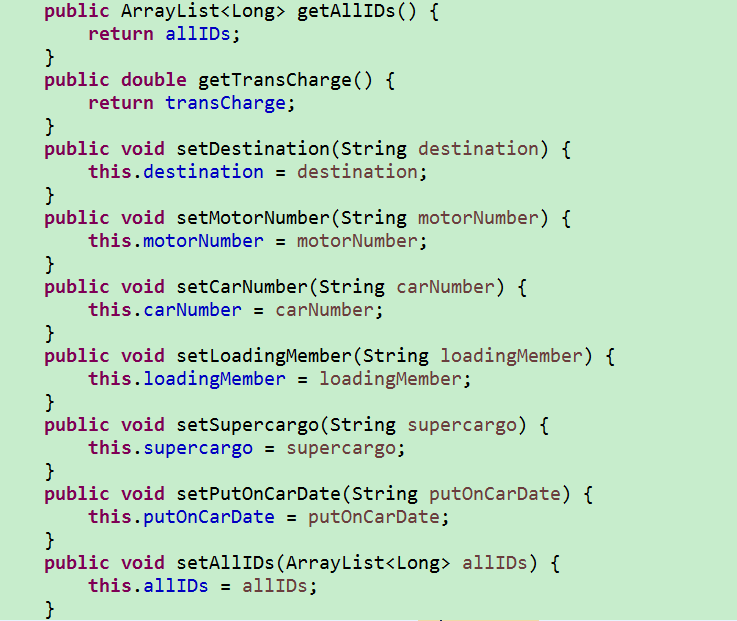


图6.1-11 持久化中转单对象TransferFormPO的定义

### 6.1.12持久化对象CarInputFormPO





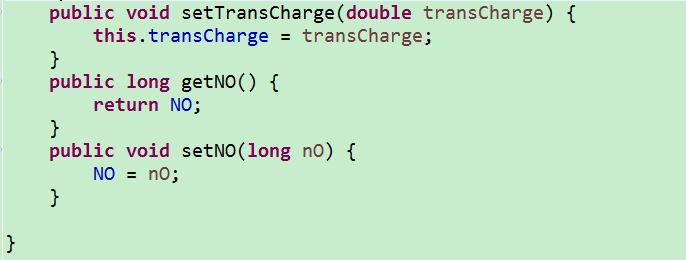


图6.1-12 持久化中转中心装车单对象CarInputFormPO的定义

### 6.1.13持久化对象StorageOutFormPO

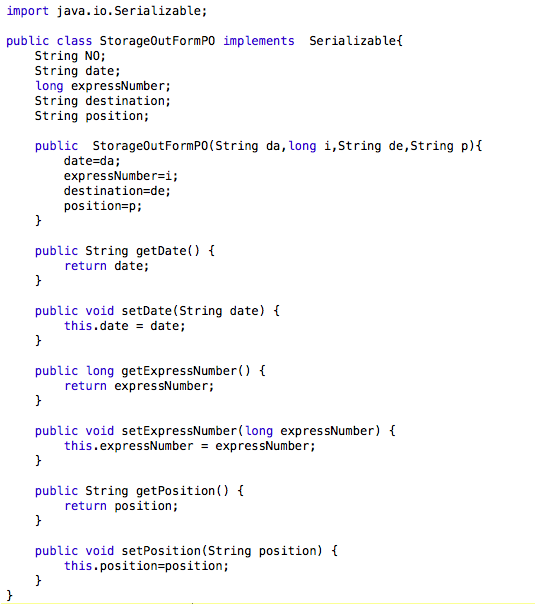


图6.1-13持久化对象StorageOutFormPO的定义

### 6.1.14持久化对象StorageInFormPO

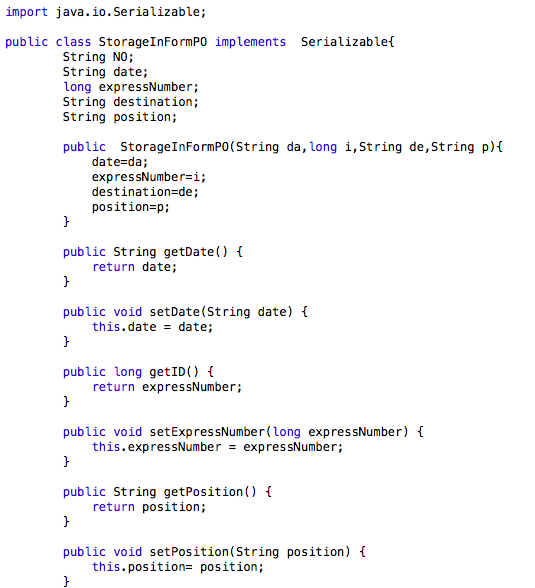
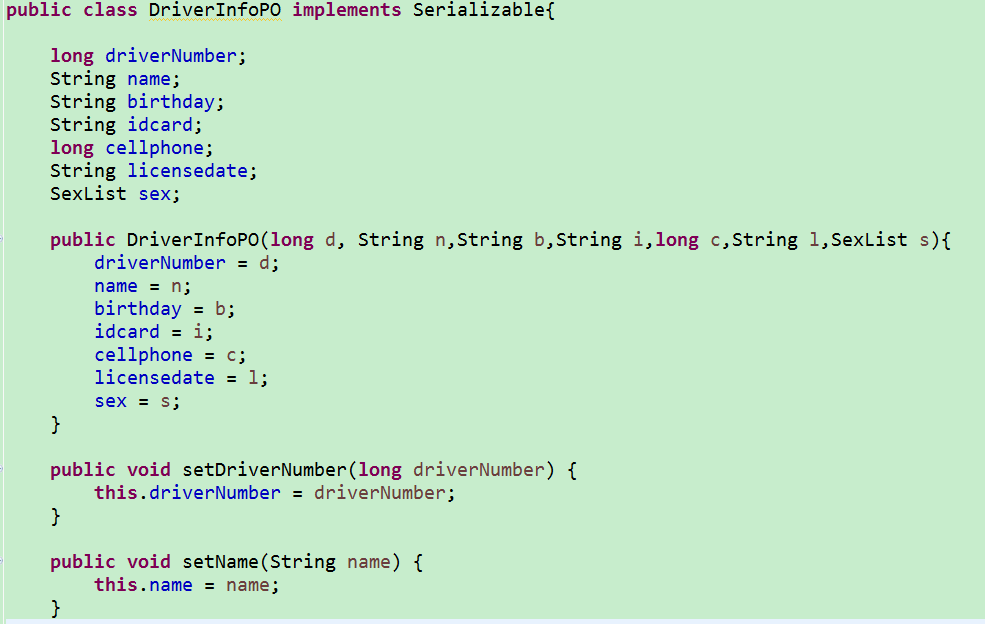
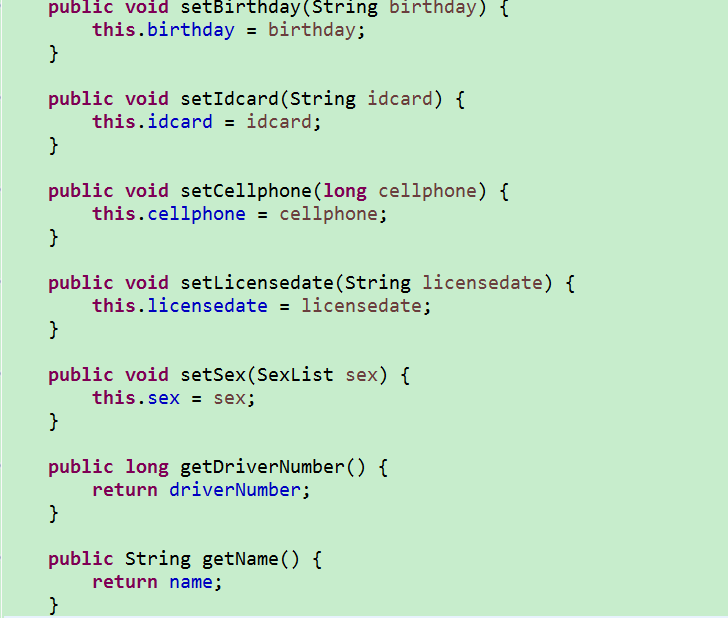


图6.1-14持久化对象StorageInFormPO的定义

### 6.1.15持久化对象DriverInfoPO





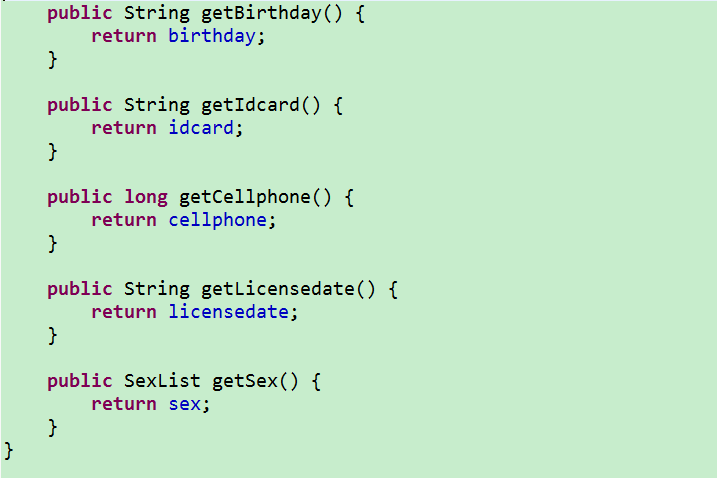


图6.1-15持久化对象DriverInfoPO的定义

### 6.1.16持久化对象AccountInfoPO



图6.1-16持久化对象AccountInfoPO的定义

### 6.1.17持久化对象BeginningAccountPO

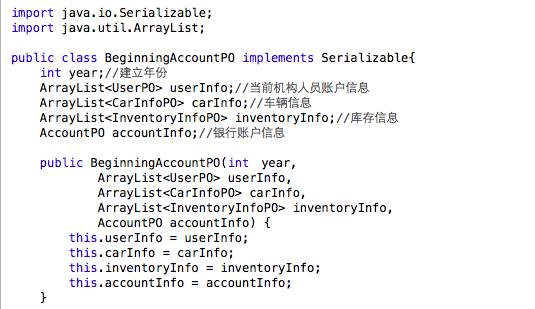




图6.1-17持久化对象BeginningAccountPO的定义

### 6.1.18持久化对象CostPayChartPO

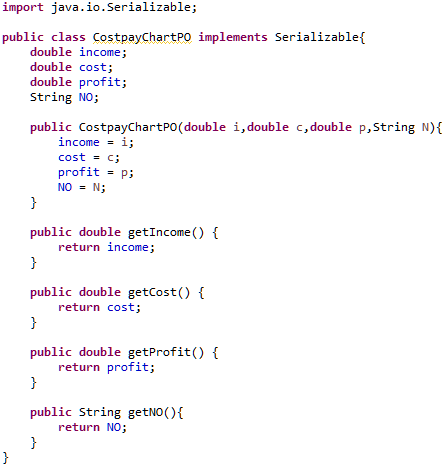
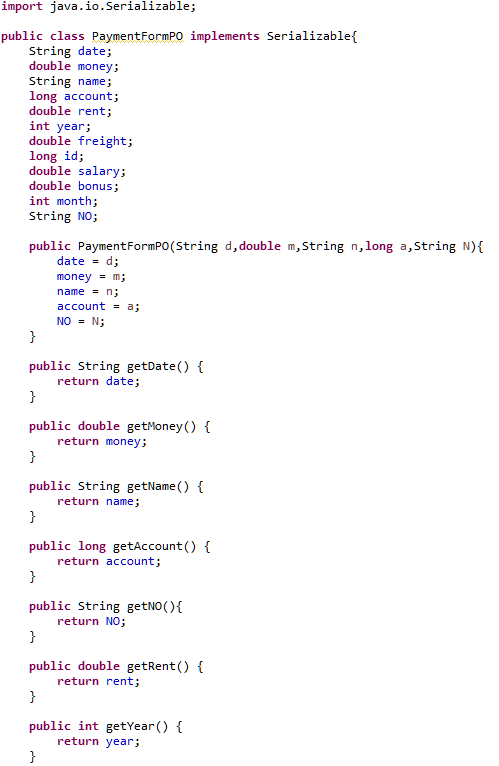


图6.1-18持久化对象CostPayPO的定义

### 6.1.19持久化对象PaymentFormPO



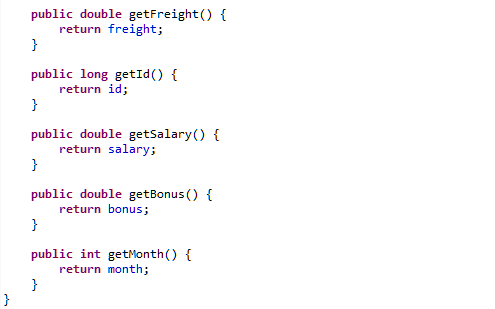
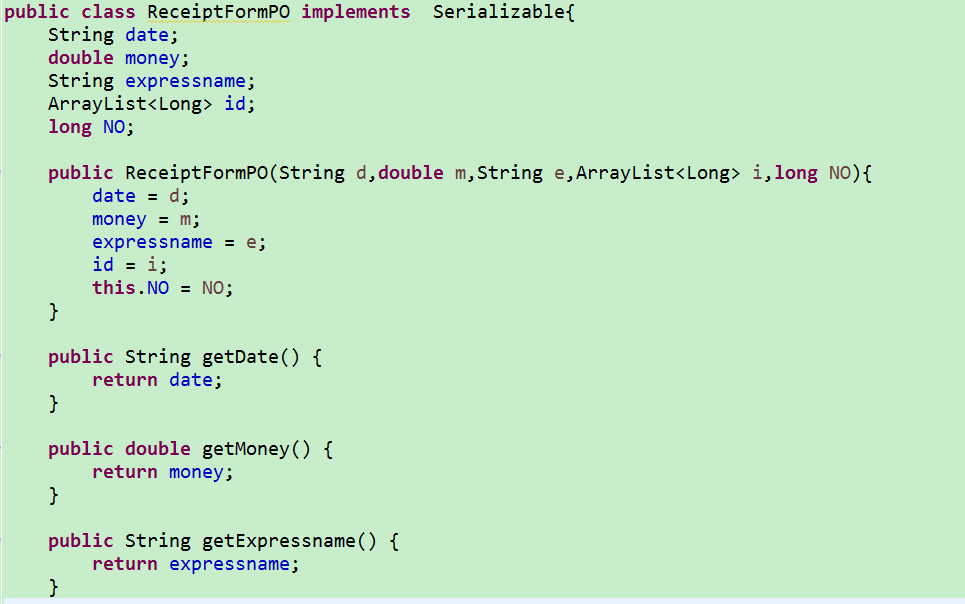


图6.1-19持久化对象PaymentFormPO的定义

### 6.1.20持久化对象ReceiptFormPO



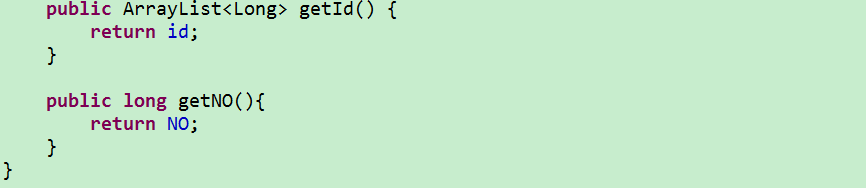
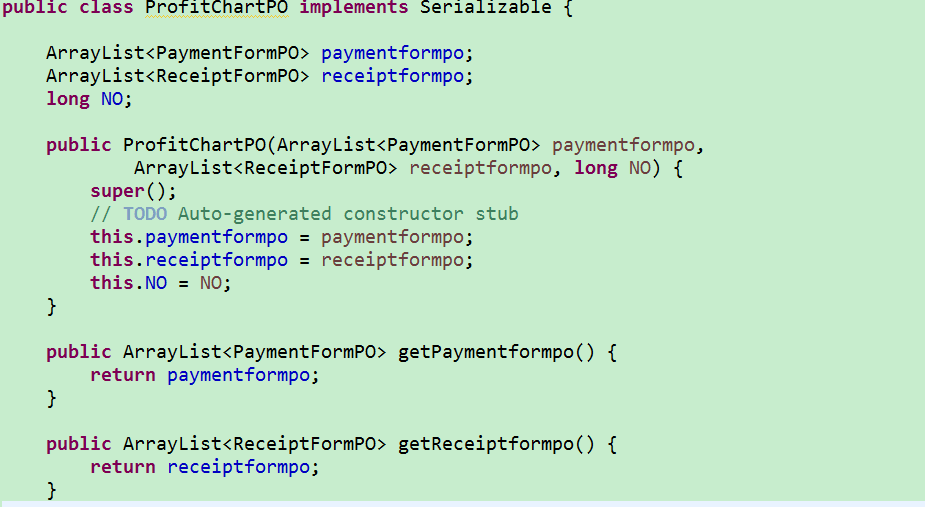


图6.1-20持久化对象ReceiptFormPO的定义

### 6.1.21持久化对象ProfitChartPO



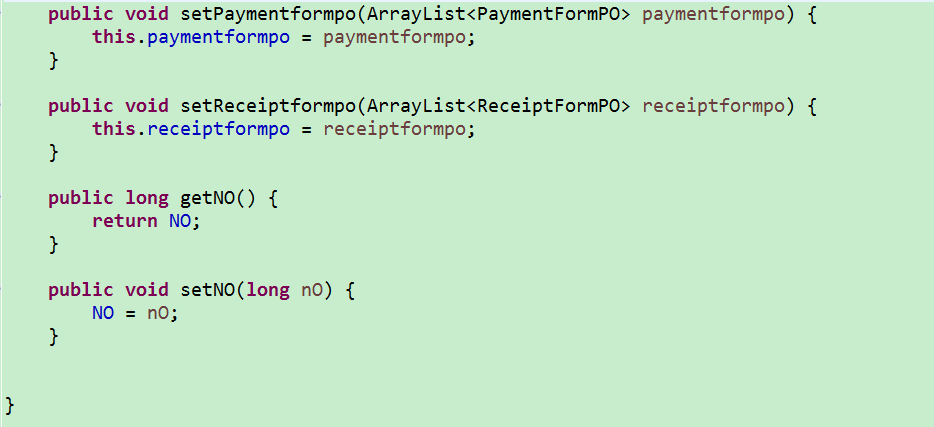


图6.1-21持久化对象ProfitChartPO的定义