快递物流管理系统

软件体系结构设计文档

v1.0 正式版

南京大学软件学院

2015-10-18

更新历史

修改人员	日期	变更原因	版本号
全体成员	2015/10/18	讨论出最终开发包设计	V0.1
全体成员	2015/10/20	初步版, 体系结构初步模型	V0.2 初步版
全体成员	2015/10/23	讨论逻辑层与数据层需要的方法	v0.3
全体成员	2015/10/25	完善接口规范,实现一个简单功能	V0.4
全体成员	2015/10/26	评审后的正式版	V1.0 正式版

目录

1.	引言		
	1.1	编制目的	错误!未定义书签。
	1.2	词汇表	错误!未定义书签。
	1.3	参考资料	错误!未定义书签。
2. j	产品指	苗述	错误!未定义书签。
3.	逻辑	视角	4
4.	组合	视角	
	4.1	开发包图	6
	4.2	运行时进程	9
	4.3	物理部署	10
5.	接口	视角	
	5.1	模块的职责	错误!未定义书签。
	5.2	用户界面层的分解	错误!未定义书签。
		5.2.1 用户界面层模块的职责	13
		5.2.2 用户界面层模块的接口规范	13
		5.2.3 用户界面模块设计原理	13
	5.3	业务逻辑层的分解	
		5.3.1 业务逻辑层模块的职责	错误!未定义书签。
		5.3.2 业务逻辑层模块的接口规范	错误!未定义书签。
	5.4	数据层的分解	22
		5.4.1 数据层模块的职责	
		5.4.2 数据层模块的接口	23
6.	信息	视角	
	6.1	数据持久化对象	29
	6.2	Txt 持久化格式	35

1. 引言

1.1 编制目的

本报告详细完成对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
ELMS	快递物流管理系统	无

1.3 参考资料

- 1) 快递物流管理系统用例文档
- 2) 快递物流管理系统软件需求规格说明文档

2. 产品概述

参考快递物流管理系统用例文档和快递物流管理系统软件需求规格说明文档中对产品的概括描述。

3. 逻辑视角

快递物流管理系统中,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和访问。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。

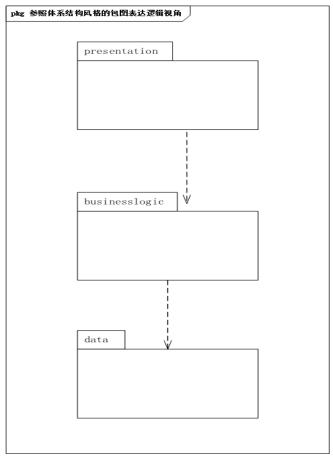


图 3-1 参照体系结构风格的包图表达逻辑视角

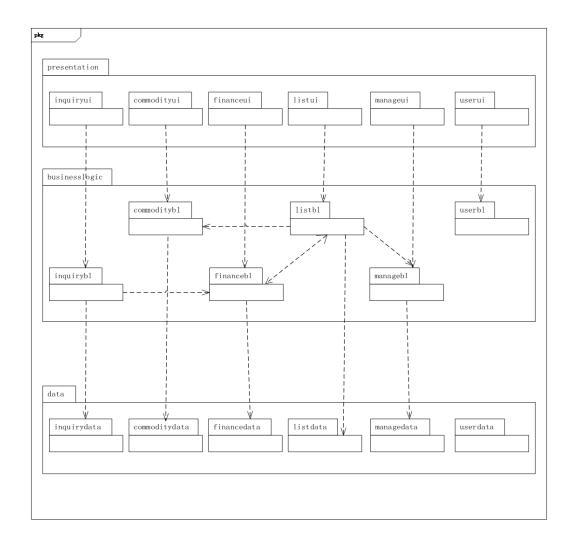


图 3-2 软件体系结构逻辑设计方案

4. 组合视角

4.1 开发包图

快递物流管理系统的最终开发包设计如表 4-1-1 所示

表 4-1-1 快递物流管理系统的最终开发包设计

开发(物理)包	依赖的其他包
mainui	inquiryui, commodityui, financeui, manageui, listui, userui
	, vo
inquiryui	inquiryblservice,界面类库包,vo
inquiryblservice	
inquirybl	inquiryblservice, inquirydataservice, po, financebl
inquirydataservice	JavaRMI, po
inquirydata	JavaRMI, po, databaseutility
commodityui	commodityblservice,界面类库包,vo
commodityblservice	

commoditybl	commodityblservice, commoditydataservice, po
commoditydataservice	JavaRMI, po
commoditydata	JavaRMI, po, databaseutility
financeui	financeblservice,界面类库包,vo
financeblservice	
financebl	financeblservice, financedataservice, po
financedataservice	JavaRMI, po
financedata	JavaRMI, po, databaseutility
manageui	manageblservice,界面类库包,vo
manageblservice	
managebl	manageblservice, managedataservice, po
managedataservice	JavaRMI, po
managedata	JavaRMI, po, databaseutility
listui	listblservice,界面类库包, vo
listblservice	
listbl	listblservice, listdataservice, po, managebl, financebl,
	commoditybl
listdataservice	JavaRMI, po
listdata	JavaRMI, po, databaseutility
userui	userblservice,界面类库包, vo
userblservice	
userbl	userblservice, userdataservice
userdataservice	JavaRMI, po
userdata	RMI, po, databaseutility
VO	
ро	
utilitybl	
界面类库包	
JavaRMI	
databaseutility	JDBC

快递物流管理系统客户端开发包图如图 4-1-1 所示,服务器端开发包图如图 4-1-2 所示。

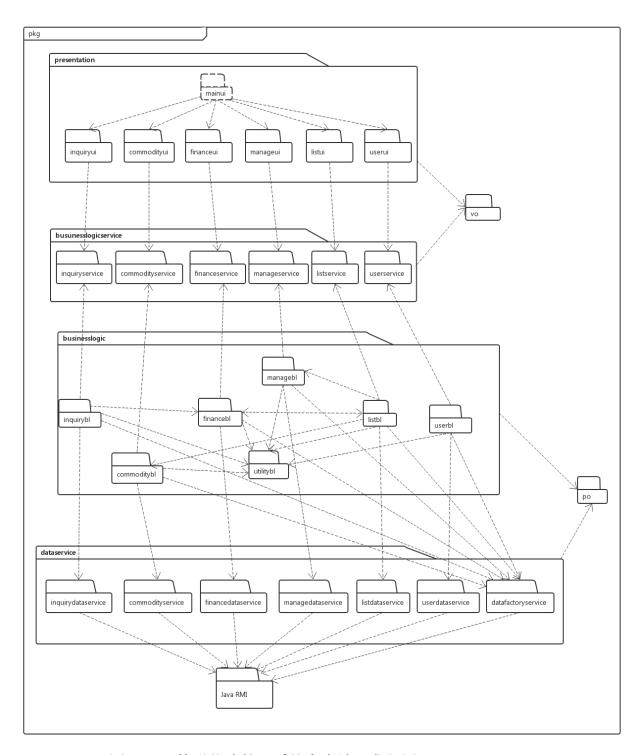


图 4-1-1 快递物流管理系统客户端开发包图

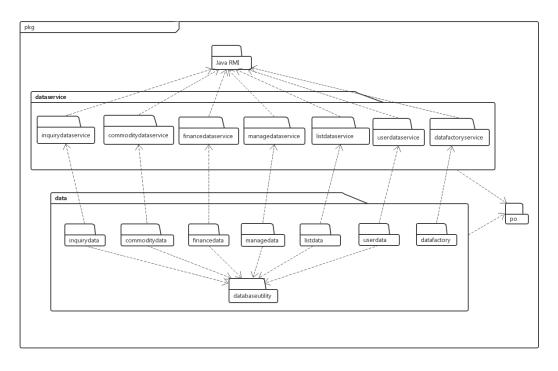


图 4-1-2 快递物流管理系统服务器端开发包图

4.2运行时进程

在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 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。

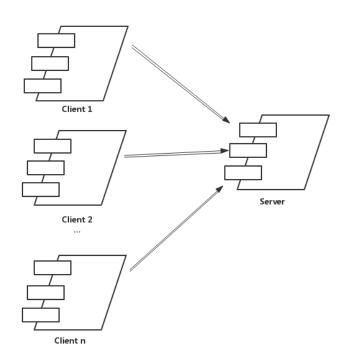


图 4-2-1 进程图

4.3 物理部署

快递物流管理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 1.7 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。

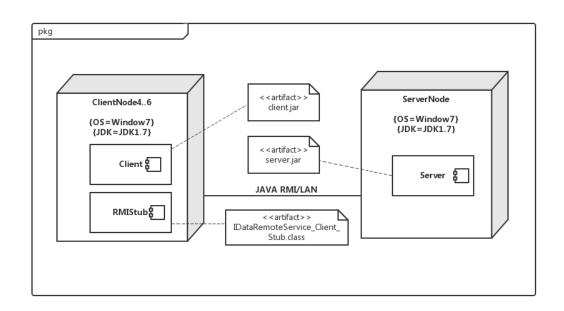


图 4-3-1 部署图

5. 接口视角

5.1 模块的职责

客户端模块和服务器端模块视图分别如图 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入表 5-1-1 和表 5-1-2 所示。

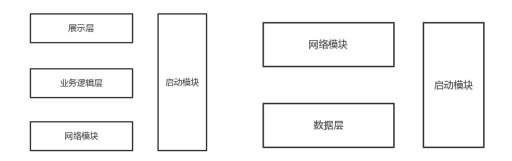


图 5-1-1 客户端模块视图

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	基于窗口的企业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 5-1-2 服务器端各层的职责

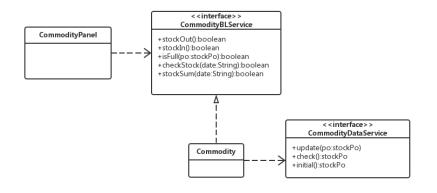
		The state of the s
层		职责
	启动模块	负责初始化网络通信机制,启动服务器
	数据层	负责数据的持久化及数据访问接口
	服务器端网络模块	利用 Java RMI 机制开启 RMI 服务,注册 RMI 服务

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
IquiryBLService		
CommodityBLService		
FinanceBLService	 客户端展示层	 客户端业务逻辑层
ManageBLService	谷广响成小伝 	各广 ^{编业} 分 ² 相层
ListBLService		
LoginBLService		
IquiryDataService		
CommodityDataService		
FinanceDataService		
ManageDataService	客户端业务逻辑层	服务端数据层
ListDataService		
UserDataService		
DatabaseFactory		

借用库存用例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)。CommodityBLService 提供了 Commodity 界面所需要的所有业务逻辑功能。CommodityDataService 提供了对数据库的更新、查看、初始化操作,这样的实现就大大降低了层与层之间的耦合。



5-1-3 库存用例层之间调用的接口

5.2 用户界面层分解

根据需求,系统存在33个用户界面:登录与物流信息查询界面、快递员主界面、营业厅业务员主界面、中转中心业务员主界面、中转中心库存管理人员主界面、财务人员主界面、总经理主界面、管理员主界面、输入订单界面、查询报价和时间界面、输入收件信息界面、车辆装车管理界面、接件与派件界面、记录收款界面、车辆信息管理界面、司机信息管理界面、装运管理界面、中转接收界面、库存管理界面、结算管理界面、成本管理界面、期初建帐界面、统计报表界面、账户管理界面、查看操作日志界面、查看统计分析界面、人员和机构管理界面、审批单据界面、制定薪水策略界面、制定城市距离和价格等常量界面、管理用户的账号和密码和权限界面。

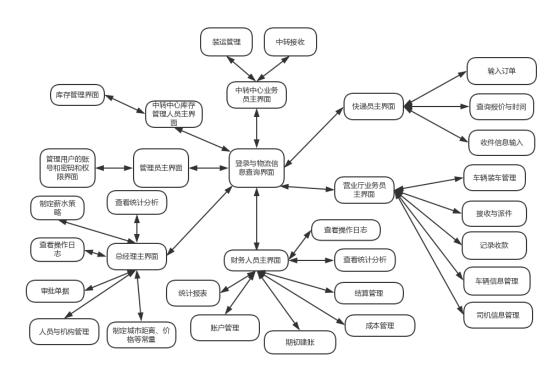


图 5-2-1 用户界面跳转

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。

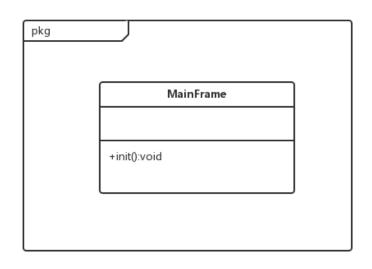


图 5-2-2 用户界面类

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	界面 Frame, 负责界面的显示和界面的跳转		

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	<pre>init(args:String[])</pre>
	前置条件	无
	后置条件	显示 Frame 以及 LoginPanel

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
businesslogicservice.LoginBLS	登录界面的业务逻辑接口
ervice	
businesslogicservice.*BLServi	每个界面都有一个相应的业务逻辑接口
ce	

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。业务逻辑层的设计如图 5-3-1 所示。

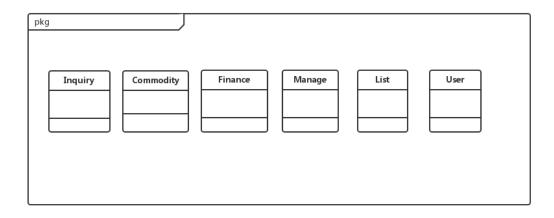


图 5-3-1 业务逻辑层的设计

5.3.1 业务逻辑层模块的职责

业务逻辑层模块的职责如表 5-3-1-1 所示。

表 5-3-1-1 业务逻辑层模块的职责

模块	职责
inquirybl	负责实现查看操作日志与统计分析界面所需要的服务
commoditybl	负责实现库存管理界面所需要的服务
financebl	负责实现成本管理、结算管理、期初建账、统计报表、账户
	管理界面所需要的服务
managohl	负责实现制定薪水策略城市距离界面和车辆司机信息管理
managebl	界面所需要的服务
listbl	负责实现所有单据界面所需要的服务
userbl	负责实现用户账户管理界面和登录界面所需要的服务

5.3.2 业务逻辑层模块的接口规范

业务逻辑层模块的接口规范如表 5-3-2-1~5-3-2-6 所示。

表 5-3-2-1 Inquirybl 模块的接口规范

提供的服务(供接口)			
inquiry.checkOperationLog	public OperationLogVO		
		checkOperationLog()	
	前置条	需要查看操作日志	
	件		
	后置条	显示操作日志	
	件		
inquiry. checkForm	语法	public String	
		checkForm(String type)	

	前置条	需要查看单据
	件	
	后置条	根据输入的类型显示相应类型的
	件	单据
inquiry. checkLogistics	语法	public LogisticsVO
		checkLogistics(String num)
	前置条	需要查看统计分析
	件	
	后置条	根据输入的数值显示收款数据
	件	
需要的	内服务 (需	接口)
服务名		服务
InquiryDataService.getInquiryData()		得到 inquiry 数据的服务的引用
InquiryDataService.		根据输入的字段名查找单一持久
checkLogistics(String num)		化对象
InquiryDataService.		查找单一持久化对象
<pre>checkOperationLog()</pre>		
InquiryDataService. String		根据输入的字段名查找单一持久
checkForm(String type)		化对象

表 5-3-2-2 Commoditybl 模块的接口规范

提供的服务(供接口)				
	语法	public boolean stockOut (StockOutVO		
Commodity stockOut	四亿	vo)		
Commodity.stockOut	前置条件	有货物需要出库		
	后置条件	系统保存出库信息并更新库存		
	语法	public boolean stockIn		
Commodity stockin	后 仏	(StockInVO vo)		
Commodity.stockIn	前置条件	有货物需要入库		
	后置条件	系统保存入库信息并更新库存		
	语法	public boolean isFull(StockPo stockpo)		
Commodity.isFull	前置条件	需要进行入库操作		
	后置条件	返回可以存放货物的地址		
	语法	public boolean checkStock(String		
Commodity.checkStock	1014	startDate,String endDate)		
Commodity.cneckStock	前置条件	需要查看库存的信息		
	后置条件	系统返回这段时间内的库存信息		
	语法	public boolean stockSum		
Commodity.stockSum	归亿	(String startDate,String endDate)		
	前置条件	一天工作完成后需要进行盘点工作		
	后置条件	系统返回一天内的库存信息		
需要的服务(需接口)				
服务名		服务		

DataFactory.getCommodityData()	得到 Commodity 数据的服务的引用
CommodityDataService.update(StockPo	根据名称查找单一或多个 AccountPO
po)	持久化对象
CommodityDataService.check	插入 AccountPO 单一持久化对象
CommodityDataService.initial	删除 AccountPO 单一持久化对象

表 5-3-2-3 Financebl 模块的接口规范

提供的服务(供接口)				
		public boolean gathering(gatheringVO		
Singapor path aging	语法	vo)		
Finance.gathering	前置条件	得到收款信息		
	后置条件	系统保存收款信息		
	语法	public boolean payment(paymentVO		
Finance.payment	1474	vo)		
Finance.payment	前置条件	进行了付款工作		
	后置条件	系统保存付款信息		
	语法	public void generateForm()		
Finance.generateForm	前置条件	需要生成成本收益表		
Timanee.generater orm	后置条件	返回到当前日期为止的成本收益信		
	//1 旦次	息		
	· 语法	public void generateForm(String		
Finance.generateForm		startDate,String endDate)		
	前置条件	需要生成经营情况表		
	后置条件	返回这段时间的经营信息		
	语法	public boolean initial()		
Finance.Initial	前置条件	账的使用周期结束		
	后置条件	系统保存新一套账的信息		
	语法	public boolean		
Finance.addAccount		addAccount(AccountVO vo)		
Finance.addAccount	前置条件	需要添加新的账户		
	后置条件	保存新的账户的信息		
Finance.searchAccount	· 语法	public boolean searchAccount(String		
		name)		
	前置条件	需要查找某个账户		
	后置条件	返回该账户的信息		
	· 语法	public boolean DelAccount(String		
Finance.DelAccount		name)		
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	前置条件	需要删除某个账户		
	后置条件	系统更新账户信息		
Finance. EditAccount	语法	public boolean		
	1	EditAccount(AccountVO vo)		

前置条件	账户信息发生变更	
后置条件	系统保存更新过的账户信息	
需要的服务((需接口)	
服务名	服务	
DataFactory.getFinanceData()	得到 Finance 数据的服务的引用	
FinanceDataService.get()	获取成本收益表	
FinanceDataService. get(String startDate,	 根据字段名进行查找多个持久化对象	
String endDate)	依据于权石进行 旦找多十 持久化对象	
FinanceDataService.addAccount(AccountP	删除 AccountPO 单一持久化对象	
O accountpo)		
FinanceDataService.searchAccount()	显示所有账户的信息	
FinanceDataService.searchAccount(String	 根据字段名进行查找多个持久化对象	
name)	依据于权石近行直找多十诗次化对象	
FinanceDataService.DelAccount(AccountP	 删除单一持久化对象	
O accountpo)	MINTE IT / PLANT &	
FinanceDataService.EditAccount(String	 修改单一持久化对象	
name, AccountPO newAccountPO)	修以中 对八化/1家	

表 5-3-2-4 Managebl 模块的接口规范

提供的服务(供接口)				
	语法	public boole	an	
Managa addDrivan	旧亿	addDriver(DriverVO driverVO)		
Manage.addDriver	前置条件	需要添加新的司机信息		
	后置条件	系统保存司机信息		
	语法	public boole	an	
Managa dalDrivar	后 亿	delDriver(DriverVO drivervO)		
Manage.delDriver	前置条件	司机不在职		
	后置条件	系统更新司机信息		
Manage.checkDriver	; 五、〉十.	public Driver	VO	
	语法 	checkDriver(String driveNumber	r)	
	前置条件	需要查看司机信息		
	后置条件	返回该司机的信息		
	语法	public boole	an	
Managa addVahiala	店伝 	addVehicle(VehicleVO vehicleVO))	
Manage.addVehicle	前置条件	需要增加车辆信息		
	后置条件	系统保存车辆信息		
Manage. delVehicle 前	があり上	public boole	an	
	语法	delVehicle(VehicleVO vehiclevO))	
	前置条件	有车辆不再投入使用		
	后置条件	系统更新车辆信息		
Managa shaakVahiala	4五.沙土	public Vehicle	VO	
Manage.checkVehicle	语法	checkVehicle(String		

	T	1 1 1 1 1		
		vehicleNumber)		
	前置条件	需要查找某辆车的信息		
	后置条件	返回该车辆的信息		
	 语法	public void updateSalary(String		
Manage.updateSalary	四14	position, String Type)		
manage. updatesalary	前置条件	需要制定薪水策略		
	后置条件	系统保存薪水策略		
		public void		
	语法	updateConstant(ConstantVO		
Manage.updateConstant		constantVO)		
	前置条件	需要制定城市距离价格常量		
	后置条件	系统保存城市距离价格常量		
需要的服务。		(需接口)		
服务名		服务		
DataFactory.getManageData()		得到 Finance 数据的服务的引用		
ManageDataService.addDriver(DriverP		 增加单一持久化对象		
0 driverPO)		产目2011年 1万人 KLA13家		
ManageDataService.		删除单一持久化对象		
deleteDriver(DriverPO po)		删除毕 特久化利家		
ManageDataService.checkDriver(Strin		 根据字段名进行查找多个持久化对象		
g driveNumber)		似地 下权石处刊 巨限夕上 时八化剂家		
ManageDataService.addVehicle(Vehicl		增加单一持久化对象		
ePO vehiclePO)				
ManageDataService.deleteVehicle(Veh		删除单一持久化对象		
iclePO po)				
ManageDataService.checkVehicle(Stri		 根据字段名进行查找多个持久化对象		
ng vehicleNumber)		似地于权石进行互认多年特久化对象		
ManageDataService.updateSalary(Stri		 修改单一持久化对象		
ng position, String Type)		沙以牛 村八化州家		
ManageDataService.updateConstant(Co		更新一个 po		
nstantPO constantPO)		史初一个 po 		

表 5-3-2-5 Listbl 模块的接口规范

At a a la la la properties de la constanties de				
提供的服务(供接口)				
	语法	public boolean order(OrderVO		
Liet ender	前置条件	orderVO)		
List. order		需要输入订单信息		
	后置条件	系统保存订单信息		
		public void adresseeInfomation		
List. adresseeInfomation	语法	(AddresseeInformationVO		
		addresseeInformationVO)		

	前置条件	需要输入收件信息
	后置条件	系统保存收件信息
		public void loadingInfo
	语法	(Loading VO loading VO)
List. loadingInfo	前置条件	需要输入装车信息
G	后置条件	系统保存装车信息
		public void receiveInfo
	语法	(ReceiveVO receiveVO)
List. receiveInfo	前置条件	需要输入接收信息
	后置条件	系统保存接收信息
		public void distributeInfo
	语法	(DistributeVO distributeVO)
List. distributeInfo	前置条件	需要输入派件信息
	后置条件	系统保存派件信息
	川旦が川	public void receipt (Receipt VO
	语法	receiptVO)
List. receipt	前置条件	需要输入快递员收款信息
	后置条件	系统保存快递员收款信息
	// 旦 // // // // // // // // // // // //	public void transInfo
	语法	(TransferVO transferVO)
List. transInfo	前置条件	需要输入中转单信息
	后置条件	系统保存中转单信息
	川旦小川	public void transArrive
	语法	(TransferReceiveVO
List. transArrive	ИИ	transferReceiveVO)
Elst. clansmilive	前置条件	需要输入中转中心到达单信息
	后置条件	系统保存中转中心到达单信息
		public void stockOut(StockOutVO
	语法	stockOutVO)
List. stockOut	前置条件	需要输入订单信息
	后置条件	系统保存订单信息
List. stockIn		public void stockIn(StockInVO
	语法	stockInVO)
	前置条件	需要输入入库单信息
	后置条件	系统保存入库单信息
List. gathering		public boolean gathering
	语法	(Gathering VO gathering VO)
	前置条件	需要输入出库单信息
	后置条件	系统保存出库单信息
<u> </u>		public void payment (Payment VO
List. payment	语法	payment (VO)
	前置条件	需要输入收款单信息
	后置条件	系统保存收款单信息
	川旦ホ川	かががける数十百心

Elist. saveList	Г	T	Т		
List. saveList 前置条件 单子输入完毕 后置条件 系统保存单子信息 语法 public ListVO push() 前置条件 单子输入完毕 后置条件 将单据传给总经理 需要的服务(需接口) 服务名 服务 限务 取务 取务 取务 取务 取务 取务 取		海 沙	_	void	saveList(ListPO
Timin	List savolist	ИІД	list)		
語法	List. SaveList		单子输入	完毕	
前置条件 単子输入完毕 后置条件 将单据传给总经理 需要的服务(需接口) 服务名 服务名 服务 限务 限务 日本 日本 日本 日本 日本 日本 日本 日		后置条件	系统保存	单子信息	
店置条件 将単据传给总经理 需要的服务(需接口) 服务名 服务名 服务名 服务 限务 日本 日本 日本 日本 日本 日本 日本 日		语法	public L	istVO p	ush()
R要的服务(需接口) 服务名 DataFactory. getListData() 得到 List 数据的服务的引用 ListDataService. saveOrder (OrderPO orderPO) (R存单一持久化对象 ListDataService. saveAddresseeInfo(A ddresseeInformationPO addresseeInformationPO) ListDataService. saveLoadingInfo (LoadingPO loadingPO) ListDataService. saveReceiveInfo(ReceiptPO receiptPO) ListDataService. saveDistributeInfo(DistributePO distributePO) ListDataService. saveReceipt (ReceiptPO receiptPO) ListDataService. saveReceipt (ReceiptPO receiptPO) ListDataService. saveReceipt (ReceiptPO receiptPO) ListDataService. saveReceipt (ReceiptPO receiptPO) ListDataService. saveReceipt (ReceiptPO transferPO transferPO) ListDataService. saveTransInfo(TransferPO transferPO transferReceivePO) ListDataService. saveTransArrive(TransferReceivePO) ListDataService. saveTransArrive(TransferReceivePO) ListDataService. saveTransArrive(TransferReceivePO) ListDataService. saveTransArrive(TransferReceivePO) ListDataService. saveTransArrive(TransferReceivePO)	List. push	前置条件	单子输入	完毕	
服务名 DataFactory. getListData() ListDataService. saveOrder(OrderPO orderPO) ClistDataService. saveAddresseeInfo(A ddresseeInformationPO addresseeInformationPO) ListDataService. saveLoadingInfo (LoadingPO loadingPO) ListDataService. saveReceiveInfo(ReceiptPO) receiptPO) ListDataService. saveDistributeInfo(DistributePO distributePO) ListDataService. saveReceipt (Receipt PO receiptPO) ListDataService. saveReceipt (Receipt PO receiptPO) ListDataService. saveTransInfo(TransferPO transferPO) ListDataService. saveTransInfo(TransferReceivePO) ListDataService. saveTransArrive(TransferReceivePO)		后置条件	将单据传	给总经理	
服务名 DataFactory. getListData() ListDataService. saveOrder(OrderPO orderPO) ClistDataService. saveAddresseeInfo(A ddresseeInformationPO addresseeInformationPO) ListDataService. saveLoadingInfo (LoadingPO loadingPO) ListDataService. saveReceiveInfo(ReceiptPO) receiptPO) ListDataService. saveDistributeInfo(DistributePO distributePO) ListDataService. saveReceipt (Receipt PO receiptPO) ListDataService. saveReceipt (Receipt PO receiptPO) ListDataService. saveTransInfo(TransferPO transferPO) ListDataService. saveTransInfo(TransferReceivePO) ListDataService. saveTransArrive(TransferReceivePO)	需	要的服务(需接口)		
DataFactory. getListData() ListDataService. saveOrder(OrderPO orderPO) CorderPO) CorderPO) CistDataService. saveAddresseeInfo(A ddresseeInformationPO addresseeInformationPO) CistDataService. saveLoadingInfo (LoadingPO loadingPO) CistDataService. saveReceiveInfo(ReceiptPO) receiptPO) CistDataService. saveDistributeInfo(DistributePO distributePO) CistDataService. saveReceipt(ReceiptPO) receiptPO) CistDataService. saveReceipt(ReceiptPO) receiptPO) CistDataService. saveTransInfo(TransferPO transferPO) CistDataService. saveTransArrive(TransferReceivePO)				服	务
CorderPO			得到 List		
ListDataService. saveAddresseeInfo(A ddresseeInformationPO addresseeInformationPO)	ListDataService.saveOrder(OrderPO	加去	+ 4 /1, -1	Æ1
ddresseeInformationP0 R存单一持久化对象 Receipted Rec	orderPO)		保仔単一持久化对象		
addresseeInformationP0 ListDataService. saveLoadingInfo (LoadingP0 loadingP0) 保存单一持久化对象 ListDataService. saveReceiveInfo(ReceiptP0 receiptP0) 保存单一持久化对象 ListDataService. saveDistributeInfo(DistributeP0 distributeP0) 保存单一持久化对象 ListDataService. saveReceipt (ReceiptP0 receiptP0) 保存单一持久化对象 ListDataService. saveTransInfo(TransferP0 transferP0) 保存单一持久化对象 ListDataService. saveTransArrive(TransferReceiveP0 transferReceiveP0) 保存单一持久化对象 保存单一持久化对象 Ref单一持久化对象 Ref单位的 Ref单	ListDataService.saveAddres	seeInfo(A			
ListDataService. saveLoadingInfo (LoadingPO loadingPO) ListDataService. saveReceiveInfo(Rec eiptPO receiptPO) ListDataService. saveDistributeInfo(DistributePO distributePO) ListDataService. saveReceipt(Receipt PO receiptPO) ListDataService. saveReceipt(Receipt PO receiptPO) ListDataService. saveTransInfo(Trans ferPO transferPO) ListDataService. saveTransArrive(TransferReceivePO) ListDataService. saveTransArrive(TransferReceivePO) ListDataService. saveTransArrive(TransferReceivePO) ListDataService. saveTransArrive(TransferReceivePO)	ddresseeInformationPO		保存单一排	寺久化对	象
(LoadingPO loadingPO) ListDataService. saveReceiveInfo(ReceiptPO)	addresseeInformationPO)				
ListDataService. saveReceiveInfo(ReceiptPO receiptPO) ListDataService. saveDistributeInfo(DistributePO distributePO) ListDataService. saveReceipt (Receipt PO receiptPO) ListDataService. saveTransInfo(TransferPO transferPO) ListDataService. saveTransArrive(TransferReceivePO) ListDataService. saveTransferReceivePO) ListDataService. saveTransArrive(TransferReceivePO) ListDataService. saveStockOut(StockOut (StockOut (S	ListDataService.saveLoadin	gInfo	加去	+ 4 /1, -1	Ä
ListDataService. saveReceiveInfo(ReceiptP0 receiptP0) ListDataService. saveDistributeInfo(DistributeP0 distributeP0) ListDataService. saveReceipt(ReceiptP0 receiptP0) ListDataService. saveTransInfo(TransferP0 transferP0) ListDataService. saveTransArrive(TransferReceiveP0 transferReceiveP0) ListDataService. saveTransArrive(TransferReceiveP0) ListDataService. saveStockOut(StockOutCatalogy)	(LoadingPO loadingPO)		保仔里一持久化对家		
ListDataService. saveDistributeInfo(DistributeP0 distributeP0) ListDataService. saveReceipt (Receipt P0 receiptP0) ListDataService. saveTransInfo(Trans ferP0 transferP0) ListDataService. saveTransArrive(TransferReceiveP0 transferReceiveP0) ListDataService. saveTransArrive(TransferReceiveP0 transferReceiveP0)	ListDataService.saveReceiv	eInfo(Rec	加去	+ 4 /1, -1	Æ1
DistributeP0 distributeP0) ListDataService. saveReceipt (Receipt P0 receiptP0) ListDataService. saveTransInfo(Trans ferP0 transferP0) ListDataService. saveTransArrive(TransferReceiveP0 transferReceiveP0) ListDataService. saveTransArrive(TransferReceiveP0) ListDataService. saveStackOut (StackOut StackOut StackOu	eiptPO receiptPO)		徐仔里一# 	守久化刈	家
ListDataService. saveReceipt (Receipt PO receiptPO) ListDataService. saveTransInfo(Trans ferPO transferPO) ListDataService. saveTransArrive(TransferReceivePO transferReceivePO) ListDataService. saveTransArrive(TransferReceivePO transferReceivePO)	ListDataService.saveDistributeInfo(四方出 4	生 4 71234	名
PO receiptPO) ListDataService. saveTransInfo(Trans ferPO transferPO) ListDataService. saveTransArrive(TransferReceivePO) ListDataService. saveTransArrive(TransferReceivePO) ListDataService. saveStackOut (StackOut (StackOu	DistributePO distributePO)		沐仔毕 ^一 	寸 久化刈	豕
ListDataService.saveTransInfo(Trans ferP0 transferP0)	ListDataService.saveReceipt(Receipt		四方出 4	生 4 71274	名
ferPO transferPO) ListDataService. saveTransArrive(TransferReceivePO) ListDataService. saveTransArrive(TransferReceivePO) R存单一持久化对象 R存单一持久化对象	PO receiptPO)		徐仔里一# 	守久化刈	家
ListDataService.saveTransArrive(TransferReceiveP0) ListDataService.saveTransArrive(TransferReceiveP0) ListDataService.saveTransArrive(TransferReceiveP0)	ListDataService.saveTransInfo(Trans		四方出 +	生 4 7 1 7 4	名
nsferReceiveP0 transferReceiveP0) 保存单一持久化对象	ferPO transferPO)		沐行毕 ^一 	N.11.十一14.7.17.13.	
I ist Data Service serves teakOut (SteekO	ListDataService.saveTransArrive(Tra		促 友畄一持な <i>い</i> 対免		鱼
ListDataService. saveStockOut (StockO 保方畄一 はなかが免	nsferReceivePO transferRec	eivePO)	体付半一村久化刈冢		
	ListDataService.saveStockOut(StockO		但方 说 ,扶力从对免		
utPO stockOutPO)	utPO stockOutPO)		体行中一村久化刈豕		豕
ListDataService. saveStockIn (StockIn / 反友的一样 A / 以对角	ListDataService.saveStockI	n(StockIn			
PO stockInPO)	PO stockInPO)		保存单一持久化对象		
ListDataService. savePayment (Payment / 反友的一块力化对象	ListDataService.savePayment(Payment				
PO paymentPO) 保存单一持久化对象	PO paymentPO)		体行中一行久化剂家		
ListDataService. saveFinance (Gatheri 根皮的一块力化对象	ListDataService.saveFinance(Gatheri		促友弟―特友ル社会		
ngPO gatheringPO)	ngPO gatheringPO)		保存单一持久化对象		

表 5-3-2-6 Userbl 模块的接口规范

提供的服务(供接口)				
	 语法	public boolean login(String		
User.addDriver	ロゼ	userID, String password)		
	前置条件	password 符合输入规范		

	后置条件	查找是否存在相应的 User,根据输入
		的 password 返回一个 Boolean 值
	语法	public boolean add(UserVO
User. add	りな	userVO)
oser. add	前置条件	需要增加一个用户
	后置条件	系统保存用户信息
	 语法	public boolean delete(UserVO
User. delete	归仏	userVO)
oser, defete	前置条件	需要删除某用户账户的信息
	后置条件	系统更新用户账户信息
	语法	public boolean modify(UserVO
Haara madifu	后亿	userVOOld, UserVO userVONew)
User. modify	前置条件	新用户密码符合规则
	后置条件	系统保存用户账户信息
	语法	<pre>public UserVO find(String name)</pre>
User.find	前置条件	用户名名称输入符合规范
	占型 及	系统根据输入的名称查找该用户,并
	后置条件	返回用户信息
需要的服务(需接口)
服务名		服务
DataFactory.getUserData()		得到 User 数据的服务的引用
UserDataService.addUser(UserPO po)		增加单一持久化对象
UserDataService.deleteUser(UserPO		删除单一持久化对象
po)		删除毕一行久化对家
UserDataService.editUser(UserPO po)		修改单一持久化对象
UserDataService.find(String		根据字段名进行查找持久化对象
		根据字段名进行查找持久化对象

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,包括对于持久化数据的增、删、改、查。各业务逻辑需要的服务由各 DataService 接口提供。持久化数据的保存采用 Txt 文件形式进行保存。以 FinanceDataService 为例,数据层模块的描述具体如图 5-4-1 所示。

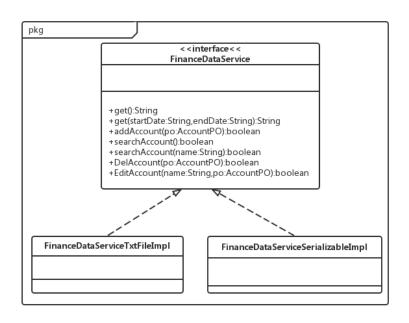


图 5-4-1 FinanceDataService 数据层模块的描述

5.4.1 数据层模块的职责

数据层模块的职责如表 5-4-1-1~5-4-1-6 所示。

表 5-4-1-1 数据层模块(InquiryDataService)的职责

农了中工工 数相应快气(induity DataSci Vice/用)不负	
模块	职责
InquiryDataService	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
InquiryDataServiceTextFileImpl	基于 Txt 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
InquiryDataServiceSerializableFileImpl	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

表 5-4-1-2 数据层模块(CommodityDataService)的职责

模块	职责
CommodityDataService	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
CommodityDataServiceTextFileImpl	基于 Txt 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
CommodityDataServiceSerializableFileImpl	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

表 5-4-1-3 数据层模块(FinanceDataService)的职责

模块	职责
FinanceDataService	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
FinanceDataServiceTextFileImpl	基于 Txt 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
FinanceDataServiceSerializableFileImpl	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

表 5-4-1-4 数据层模块(ManageDataService)的职责

Man = 1 300 H/A DOM (manager and) HV 1/30	
模块	职责
ManageDataService	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
ManageDataServiceTextFileImpl	基于 Txt 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
ManageDataServiceSerializableFileImpl	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

表 5-4-1-5 数据层模块(ListDataService)的职责

次 3 年 1 3	
模块	职责
ListDataService	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
ListDataServiceTextFileImpl	基于 Txt 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
ListDataServiceSerializableFileImpl	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

表 5-4-1-6 数据层模块(UserDataService)的职责

次3~10 效相公庆外(Osci-Dutasci vice/的外页	
模块	职责
UserDataService	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
UserDataServiceTextFileImpl	基于 Txt 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
UserDataServiceSerializableFileImpl	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

数据层模块的接口规范如表 5-4-2-1~5-4-2-6 所示。

5-4-2-1 数据层模块 (InquiryDataService) 的接口规范

提供的服务(供接口)			
InquiryDataService.checkLogistics	语法	public LogisticsPO checkLogistics (String	
		num) throws	
		RemoteException;	
	前置条件	无	
	后置条件	按照 num 进行查找返回相应	
		的 Logistics 的结果	
	语法	Public operationLogPO	
InquiryDataService.checkOperation	ИИ	checkOperation() throws	
		RemoteException;	
	前置条件	无	
	后置条件	按照先后顺序返回	
		operationLogPO	
	3五分十	public String	
InquiryDataService.checkForm	语法	checkForm(String type)	
		throws RemoteException;	
	前置条件	无	
	后置条件	按照 type 进行查找,返回相	
		应的 String	

表 5-4-2-2 数据层模块 (CommodityDataService) 的接口规范

次5年22数循层模式(CommodityDataService)的接口然包		
提供的服务(供接口)		
	语法	public void update(StockPO PO)
CommodityDataService.update		throws RemoteException;
	前置条件	在 txt 文件中存在同样 ID 的 po
	后置条件	更新一个 po
CommodityDataService.check	语法	public StockPO check() throws
		RemoteException;
	前置条件	无
	后置条件	返回库存信息的 PO
CommodityDataService.initial	语法	public StockPO initial() throws
		RemoteException;
	前置条件	无
	后置条件	初始化 txt 文件

表 5-4-2-3 数据层模块 (FinanceDataService) 的接口规范

提供的服务(供接口)

FinanceDataService.get FinanceDataService.get RemoteException; 前置条件 无 E置条件 更新一个 po public String get(String startDate, String endDate) throws RemoteException; 前置条件 在 txt 文件中存在同样 ID 的 po 后置条件 在 txt 文件中删除一个 po 记录 public boolean addAccount(AccountPO accountpo) throws RemoteException; 前置条件 同样 ID 的 po 在 txt 文件中不存在 后置条件 在数据库中增加一个 po 记录 public AccountPO searchAccount() throws RemoteException; 前置条件 无 返回所有的 AccountPO public AccountPO searchAccount() throws RemoteException; 前置条件 无 下inanceDataService. Eak	FinanceDataService.get	语法	public String get() throws
FinanceDataService.get 語法 更新一个 po public String get(String startDate, String endDate) throws RemoteException; 前置条件 在 txt 文件中存在同样 ID 的 po 后置条件 在 txt 文件中開除一个 po 记录 public boolean addAccount(AccountPO accountpo) throws RemoteException; 前置条件 同样 ID 的 po 在 txt 文件中不存在 后置条件 在数据库中增加一个 po 记录 public AccountPO searchAccount() throws RemoteException; 前置条件 无 后置条件 返回所有的 AccountPO public AccountPO searchAccount() throws RemoteException; 前置条件 无 后置条件 返回所有的 AccountPO public AccountPO searchAccount() throws RemoteException; 前置条件 无 fill 系件 表 fill 和 mame 进行查找并返回相应的 AccountPO 结果			RemoteException;
FinanceDataService.get In a public String get(String startDate, String endDate) throws RemoteException; in 置条件 在 txt 文件中存在同样 ID 的 po 后置条件 在 txt 文件中刷除一个 po 记录 public boolean addAccount(AccountPO accountpo) throws RemoteException; in 置条件 同样 ID 的 po 在 txt 文件中不存在 后置条件 在数据库中增加一个 po 记录 public AccountPO searchAccount() throws RemoteException; in 置条件 无 后置条件 返回所有的 AccountPO searchAccount() throws RemoteException; in 置条件 无 后置条件 逐回所有的 AccountPO searchAccount() throws RemoteException; in 置条件 无 后置条件 逐回所有的 AccountPO searchAccount() throws RemoteException; in 置条件 无 后置条件 逐回所有的 AccountPO searchAccountPO searchAccount() throws RemoteException; in 置条件 无 for a public for a publ		前置条件	无
FinanceDataService.get 语法 String endDate) throws RemoteException; 前置条件 在 txt 文件中存在同样 ID 的 po 后置条件 在 txt 文件中删除一个 po 记录 Public boolean addAccount(AccountPO accountpo) throws RemoteException; 前置条件 同样 ID 的 po 在 txt 文件中不存在 后置条件 在数据库中增加一个 po 记录 Public AccountPO searchAccount		后置条件	更新一个 po
RemoteException; 前置条件 在 txt 文件中存在同样 ID 的 po 后置条件 在 txt 文件中删除一个 po 记录 public boolean addAccount(AccountPO accountpo) throws RemoteException; 前置条件 同样 ID 的 po 在 txt 文件中不存在 后置条件 在数据库中增加一个 po 记录 public AccountPO searchAccount() throws RemoteException; 前置条件 无 后置条件 返回所有的 AccountPO public AccountPO searchAccount() throws RemoteException; 前置条件 无 后置条件 返回所有的 AccountPO searchAccount() throws RemoteException; 前置条件 无 后置条件 无 方面,			public String get(String startDate,
FinanceDataService.addAccount FinanceDataService.searchAccount FinanceDa	Finance Data Compiles and	语法	String endDate) throws
FinanceDataService.addAccount	FinanceDataService.get		RemoteException;
FinanceDataService.addAccount -		前置条件	在 txt 文件中存在同样 ID 的 po
FinanceDataService.addAccount 语法		后置条件	在 txt 文件中删除一个 po 记录
### TinanceDataService.addAccount ### throws RemoteException; ### ### ### ### ### ### ###			
## throws RemoteException; ## in a public Account PO search Account PO ## public Account PO ## in a public Account PO ##		语法	addAccount(AccountPO accountpo)
前置条件 后置条件同样 ID 的 po 在 txt 文件中不存在 在数据库中增加一个 po 记录FinanceDataService.searchAccount nt语法public AccountPO searchAccount() throws RemoteException; 前置条件 后置条件 原则的ic AccountPO public AccountPOFinanceDataService. searchAccount语法public AccountPO public Account(String name) throws RemoteException; 前置条件 方 按 name 进行查找并返回相应的 AccountPO 结果	FinanceDataService.addAccount		throws RemoteException;
FinanceDataService.searchAccount后置条件在数据库中增加一个 po 记录public AccountPO searchAccount() throws RemoteException;前置条件 无后置条件 返回所有的 AccountPOpublic AccountPOpublic AccountPOsearchAccount(String name) throws RemoteException;前置条件 无后置条件 按 name 进行查找并返回相应的AccountPO 结果		前置条件	同样 ID 的 po 在 txt 文件中不存在
FinanceDataService.searchAccount nt in a			
FinanceDataService.searchAccount nt int int int int int int i			-
nt前置条件无后置条件返回所有的 AccountPOpublicAccountPO语法searchAccount(String name) throws RemoteException;前置条件无后置条件按 name 进行查找并返回相应的 AccountPO 结果	FinanceDataService.searchAccou	''	'
后置条件 返回所有的 AccountPO public AccountPO 语法 searchAccount(String name) throws RemoteException; 前置条件 无 后置条件 按 name 进行查找并返回相应的 AccountPO 结果	nt	前置条件	• •
FinanceDataService. searchAccount in 置条件 方置条件 方置条件 方置条件 方面条件 方面条件 方面条件 方面条件 方面条件 方面条件 方面条件 方面条件 方面表件 方面表示 方面条件 方面表示 一面表示 一面表示			返回所有的 AccountPO
FinanceDataService. searchAccount 前置条件 后置条件 后置条件 AccountPO 结果			
FinanceDataService. searchAccount name 进行查找并返回相应的 后置条件 按 name 进行查找并返回相应的 AccountPO 结果		语法	searchAccount(String name) throws
searchAccount前置条件无后置条件按 name 进行查找并返回相应的 AccountPO 结果			
后置条件 按 name 进行查找并返回相应的 AccountPO 结果	searchAccount	前置条件	
后直条件 AccountPO 结果			按 name 进行查找并返回相应的
pasic boolean	FinanceDataService.DelAccount		public boolean
语法 DelAccount(AccountPO accountPO)		语法	DelAccount(AccountPO accountPO)
FinanceDataService.DelAccount throws RemoteException;			•
前置条件 在 txt 文件中存在同样 ID 的 po		前置条件	• •
后置条件 删除一个 po			
public boolean EditAccount(String			
语法 name. AccountPO newAccountPO)		语法	1.
FinanceDataService.EditAccount throws RemoteException;	FinanceDataService.EditAccount		,
前置条件 在 txt 文件中存在同样 ID 的 po		前置条件	• •
后置条件 更新一个 po			

表 5-4-2-4 数据层模块 (ManageDataService) 的接口规范

提供的服务(供接口)		
ManageDataService.addDriver	语法	<pre>public boolean addDriver (DriverPO driverPO) throws RemoteException;</pre>
	前置条件	同样 ID 的 po 在 txt 文件中不存在
	后置条	在 txt 文件中增加一个 po 记录

	件	
ManageDataService.addVehic1	语法	<pre>public boolean addVehicle(VehiclePO vehiclePO) throws RemoteException;</pre>
	前置条件	同样 ID 的 po 在 txt 文件中不存在
e	后置条件	在 txt 文件中增加一个 po 记录
ManageDataService.deleteDri	语法	<pre>public boolean deleteDriver(DriverPO driverPO) throws RemoteException;</pre>
	前置条件	在 txt 文件中存在同样 ID 的 po
ver	后置条件	删除一个 po
ManageDataService.checkDriv	语法	<pre>public DriverPO checkDriver (String driverNumber) throws RemoteException;</pre>
	前置条件	无
er	后置条件	根据 driverNumber 进行查询返回相 应的 DriverPO 结果
ManageDataService.deleteVeh	语法	publicbooleandeletaVehicle (VehiclePO)throws
icle	前置条件	RemoteException; 在 txt 文件中存在同样 ID 的 po
	后置条 件	删除一个 po
	语法	<pre>Public void updateSalary(String position, String Type);</pre>
ManageDataService.updateSal ary	前置条 件	在 txt 中存在一个 po 需要编辑
	后置条 件	更新 txt 文件中的 po
M. D. A. C	语法	Public void updateConstant(ConstantPO constantPO);
ManageDataService.updateCon stant	前置条 件	在 txt 文件中存在同样 ID 的 po
	后置条 件	更新 txt 文件中的 po

表 5-4-2-5 数据层模块 (ListDataService) 的接口规范

		stDataService)的接口规范			
提	提供的服务(供接口)				
ListDataService.saveOrder	 语法	public boolean saveOrder (OrderPO			
	1014	orderPO) throws RemoteException;			
	前置条件	无			
	后置条件	在 txt 中增加一个 po 记录,并返回			
		boolean 值			
) T) L	public boolean saveAddresseeInfo			
ListData Carvina savoAddrass	语法	(AddresseeInformationPO			
ListDataService.saveAddress		addresseeInformationPO)throws			
		RemoteException;			
	前置条件	无			
eeInfo	后置条件	在 txt 中增加一个 po 记录,并返回			
		boolean 值			
		Public Boolean			
	语法	saveLoadingInfo(LoadingPO			
ListDataService.saveLoading		loadingPO) throws			
		RemoteException;			
	前置条件	无			
Info	后置条件	在 txt 中增加一个 po 记录,并返回			
	川旦ホ川	boolean 值			
	语法	public boolean saveReceiveInfo			
		(ReceiptPO receiptPO) throws			
ListDataService.saveReceive		RemoteException;			
	前置条件	无			
	后置条件	在 txt 中增加一个 po 记录,并返回			
Info	川旦ホ川	boolean 值			
		bootean in			
		public boolean saveFinance			
	语法	(GatheringPO) gatheringPO) throws			
ListDataService.saveFinance		RemoteException;			
	前置条件	无			
		在 txt 中增加一个 po 记录,并返回			
	后置条件	boolean 值			
		public boolean			
ListDataService.saveDistrib uteInfo	语法	saveDistributeInfo(DistributeInfo			
		rmationPO			
		distributeInformationPO) throws			
		RemoteException;			
		Remotebaception,			
	前置条件	 无			
	川旦ボ汁	<i>/</i> u			

ListDataService. saveReceipt ListDataService. saveTransInfo ListDataService. saveTransInfo ListDataService. saveTransInfo ListDataService. saveTransArrive ListDataService. saveTransArrive ListDataService. saveStockOut ListDataService. saveStockIn ListDataService. saveStockIn ListDataService. saveStockIn ListDataService. saveStockIn ListDataService. savePayment ListDataService. saveRaveRaveCevice		后置条件	在 txt 中增加一个 po 记录,并返回
Bix saveReceipt (ReceiptPO receiptPO); n置条件 无 在 txt 中增加一个 po 记录,并返回 boolean 值 public boolean 多数		75 223(1)	,
ListDataService. saveReceipt 前置条件 元			1 -
E置条件 在 txt 中增加一个 po 记录,并返回 boolean 值 public boolean 第 saveTransInfo (TransferPO transferPO);			
ListDataService. saveTransIn fo boolean 值	ListDataService.saveReceipt	前置条件	1 -
Boolean 個 public boolean saveTransInfo Factor		后置条件	在 txt 中增加一个 po 记录,并返回
語法 saveTransInfo(TransferPO transferPO); 前置条件 无 后置条件 を txt 中増加一个 po 记录,并返回 boolean 值 public Boolean saveTransArrive(TransferReceivePO); 前置条件 无 后置条件 を txt 中増加一个 po 记录,并返回 boolean 值 public Boolean saveTransArrive(TransferReceivePO); 前置条件 无 后置条件 を txt 中増加一个 po 记录,并返回 boolean 值 public boolean 值 public boolean 结 public boolean saveStockOut(StockOutPO); 前置条件 无 后置条件 を txt 中増加一个 po 记录,并返回 boolean 值 public boolean 信 public boolean saveStockIn(StockInPO); 前置条件 无 后置条件 を txt 中増加一个 po 记录,并返回 boolean 值 public boolean 信			boolean 值
transferPO); in 置条件 无 foolean 值 public Boolean 值 public Boolean 值 public Boolean 值 foolean function		语法	public boolean
前置条件 元 在 txt 中増加一个 po 记录,并返回 boolean 值 public Boolean 底 saveTransArrive (TransferReceivePO transferReceivePO); 前置条件 元 在 txt 中増加一个 po 记录,并返回 boolean 值 public boolean 信 和			saveTransInfo(TransferPO
E置条件 在 txt 中増加一个 po 记录,并返回 boolean 值 public Boolean 第本 saveTransArrive(TransferReceivePO transferReceivePO); 前置条件 无	ListDataService.saveTransIn		transferPO);
Data	fo	前置条件	无
ListDataService. saveTransArrive ListDataService. saveStockOut t ListDataService. saveStockIn ListDataService. saveStockIn ListDataService. saveStockIn ListDataService. saveStockIn ListDataService. saveStockIn ListDataService. saveStockIn ListDataService. savePayment ListDataService. saveStockIn ListDataService. saveS		后置条件	在 txt 中增加一个 po 记录,并返回
ListDataService. saveTransArrivesaveTransArrive (TransferReceivePO) ; transferReceivePO) ; 前置条件后置条件在 txt 中增加一个 po 记录,并返回 boolean 值 public boolean saveStockOut (StockOutPO) ; 前置条件ListDataService. saveStockIn在 txt 中增加一个 po 记录,并返回 boolean 值ListDataService. saveStockIn语法在 txt 中增加一个 po 记录,并返回 boolean 值ListDataService. saveStockIn有 txt 中增加一个 po 记录,并返回 boolean 值ListDataService. savePayment语法在 txt 中增加一个 po 记录,并返回 boolean 值ListDataService. savePayment语法在 txt 中增加一个 po 记录,并返回 boolean 值正法市置条件无后置条件在 txt 中增加一个 po 记录,并返回			boolean 值
ListDataService. saveTransAr rivetransferReceiveP0); 前置条件后置条件在 txt 中增加一个 po 记录,并返回 boolean 值ListDataService. saveStockOut t声法public boolean saveStockOut (StockOutPO) stockOutPO); 前置条件ListDataService. saveStockIn在 txt 中增加一个 po 记录,并返回 boolean 值ListDataService. saveStockInpublic boolean saveStockIn(StockInPO stockInPO); 前置条件ListDataService. savePayment在 txt 中增加一个 po 记录,并返回 boolean 值ListDataService. savePayment语法public boolean savePayment(PaymentPO paymentPO); 前置条件ListDataService. savePayment在 txt 中增加一个 po 记录,并返回			public Boolean
rive前置条件 后置条件无 在 txt 中增加一个 po 记录,并返回 boolean 值ListDataService. saveStockOut t语法public boolean saveStockOut (StockOutPO) stockOutPO); 前置条件 后置条件ListDataService. saveStockIn语法在 txt 中增加一个 po 记录,并返回 boolean 值ListDataService. saveStockInin 置条件 后置条件无 在 txt 中增加一个 po 记录,并返回 boolean 值ListDataService. savePayment语法在 txt 中增加一个 po 记录,并返回 boolean 值ListDataService. savePayment语法public boolean savePayment (PaymentPO paymentPO); 前置条件 新定正法市置条件 后置条件在 txt 中增加一个 po 记录,并返回		语法	saveTransArrive(TransferReceivePO
E置条件 在 txt 中增加一个 po 记录,并返回 boolean 值 public boolean 组 public boolean saveStockOut (StockOutPO stockOutPO); 前置条件 无 E置条件 在 txt 中增加一个 po 记录,并返回 boolean 值 public boolean 第 aveStockIn (StockInPO); 前置条件 无 E置条件 在 txt 中增加一个 po 记录,并返回 boolean 值 public boolean saveStockIn (StockInPO stockInPO); 前置条件 无 E置条件 在 txt 中增加一个 po 记录,并返回 boolean 值 public boolean 第 avePayment (PaymentPO paymentPO); 前置条件 无 E置条件 无 在 txt 中增加一个 po 记录,并返回 boolean 有 public bo	ListDataService.saveTransAr		transferReceivePO);
ListDataService. saveStockOut t in its boolean 值 public boolean	rive	前置条件	无
Example Exa		后置条件	在 txt 中增加一个 po 记录,并返回
ListDataService. saveStockOut语法saveStockOut (StockOutPO) stockOutPO); 前置条件据表后置条件在 txt 中增加一个 po 记录,并返回 boolean 值ListDataService. saveStockInpublic boolean saveStockIn(StockInPO stockInPO); 前置条件后置条件在 txt 中增加一个 po 记录,并返回 boolean 值ListDataService. savePayment语法ListDataService. savePayment语法上istDataService. savePayment在 txt 中增加一个 po 记录,并返回 savePayment (PaymentPO paymentPO); 前置条件后置条件在 txt 中增加一个 po 记录,并返回			boolean 值
ListDataService. saveStockOut t in 置条件 E txt 中增加一个 po 记录,并返回 boolean 值 E法 Dublic boolean saveStockIn(StockInPO stockInPO); in 置条件 E txt 中增加一个 po 记录,并返回 boolean saveStockIn(StockInPO stockInPO); in 置条件 E txt 中增加一个 po 记录,并返回 boolean 值 E txt 中增加一个 po 记录,并返回 boolean 值 public boolean 结 public boolean savePayment (PaymentPO paymentPO); in 置条件 E txt 中增加一个 po 记录,并返回 boolean savePayment (PaymentPO paymentPO); in 置条件 E txt 中增加一个 po 记录,并返回 boolean savePayment (PaymentPO paymentPO);		语法	public boolean
t前置条件无后置条件在 txt 中增加一个 po 记录,并返回 boolean 值ListDataService. saveStockIn语法public boolean saveStockIn(StockInPO stockInPO);前置条件无后置条件在 txt 中增加一个 po 记录,并返回 boolean 值ListDataService. savePayment语法public boolean savePayment(PaymentPO paymentPO);前置条件无后置条件在 txt 中增加一个 po 记录,并返回			saveStockOut(StockOutPO
后置条件 在 txt 中増加一个 po 记录,并返回 boolean 值 正式	ListDataService.saveStockOu		stockOutPO);
Final Description Fin	t	前置条件	无
Boolean 但			在 txt 中增加一个 po 记录,并返回
ListDataService. saveStockIn ListDataService. saveStockIn E E E E E E E E E E E E E			boolean 值
SaveStockIn(StockInPO stockInPO); 前置条件 无		语法	public boolean
后置条件 在 txt 中增加一个 po 记录,并返回 boolean 值 ElistDataService. savePayment			<pre>saveStockIn(StockInPO stockInPO);</pre>
Final Content	ListDataService.saveStockIn	前置条件	无
Boolean 但 public boolean savePayment (PaymentPO paymentPO); 正istDataService.savePayment 前置条件 を 在 txt 中増加一个 po 记录,并返回		后置条件	在 txt 中增加一个 po 记录,并返回
ListDataService.savePayment 方置条件 元			boolean 值
SavePayment (PaymentPO); in		语法	public boolean
ListDataService. savePayment 前置条件 无 后置条件 在 txt 中增加一个 po 记录,并返回			savePayment(PaymentPO paymentPO);
后置条件 在 txt 中增加一个 po 记录,并返回	ListDataService.savePayment	前置条件	
后百金件			
			boolean 值

5-4-2-6 数据层模块 (UserDataService) 的接口规范

提供的服务(供接口)				
	语法	<pre>public void addUser(UserPO po)</pre>		
UserDataService.addUser	店伝 	throws RemoteException;		
	前置条件	同样 ID 的 po 在 txt 中不存在		
	后置条件	在 txt 中增加一个 po 记录		

UserDataService.deleteUser	语法	public void deleteUser(UserPO
		po) throws RemoteException;
	前置条件	在 txt 文件中存在相同 ID 的 po
	后置条件	删除一个 po
UserDataService.editUser	语法	public void editUser (UserPO
		po) throws RemoteException;
	前置条件	在 txt 文件中存在相同 ID 的 po
	后置条件	更新一个 po
UserDataService.find	语法	public UserPO find (String
		userInfo) throws
		RemoteException;
	前置条件	无
	后置条件	按 ID 进行查找返回相应的
		UserPO 结果

6. 信息视角

6.1 数据持久化对象

系统的 PO 类是对应的相关的实体类。系统主要的 PO 类如下所示:

- OperationLogPO: 操作日志 PO 类,包含操作的时间,操作人,操作类型。
- OrderPO: 订单信息 PO 类,包含寄件人信息(寄件人姓名、住址、单位、电话、手机),收件人信息(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、长宽高、内件品名),快递类型,包装材料,运费和包装费,总费用,预计到达日期,订单条形码号(快递编号)。
- AddresseeInformationPO: 收件信息 PO 类,包括收件单号,收件人,收件时间。
- LoadingPO: 装车信息 PO 类,包括装车日期,本营业厅编号,汽运编号,出发地,到达地,监装员,押运员,车辆代号,托运单号列表,运费。
- ReceivePO: 接收单 PO 类,包括到达日期、中转单编号、出发地、货物 到达状态(损坏、完整、丢失)。
- DistributePO: 派件单 PO 类,包括到达日期、托运订单条形码号、派送员。
- ReceiptPO: 收款信息 PO 类,包括收款日期、收款金额、收款快递员、对应的所有快递订单条形码号。
- VehiclePO: 车辆信息 PO 类,包括车辆代号、车牌号、购买时间、服役时间。
- DriverPO: 司机信息 PO 类,包括司机编号、姓名、出生日期、身份证号、手机、车辆单位、性别、行驶证期限。
- TransferPO: 中转单 PO 类,包括装运方式,装运日期、本中转中心航运编号(火车为货运编号,汽车为汽运编号)、航班号(火车和汽车为

车次号)、出发地、到达地、货柜号(火车为车厢号)、监装员、押运员(仅汽运有)、本次装箱所有托运单号、运费。

- TransferReceivePO: 中转到达单 PO 类,包括寄件单编号,到达日期, 出发地,货物状态(损坏、完整、丢失),中转中心编号,中转单编号
- StockPO: 库存信息 PO 类,包括快递编号、入库日期、目的地、区号、排号、架号、位号,库存状态
- StockInPO: 入库单 PO 类,包括快递编号、入库日期、目的地、区号、排号、架号、位号
- StockOutPO: 出库单 PO 类,包括快递编号、出库日期、目的地、装运形式(火车、飞机、汽车)、中转单编号
- FinancePO: 收款信息 PO 类,包括收款日期、收款单位、收款人、收款金额、收款地点
- PaymentPO: 付款信息 PO 类,包括付款日期、付款金额、付款人、付款 账号、条目、备注
- InitialPO:期初信息 PO 类,包括机构、人员、车辆、库存、银行账户名称、银行账户余额
- AccountPO: 账户信息 PO 类,包括账户名称和余额
- WorkerPO; 人员机构信息 PO 类,包括姓名,身份证号、职位、所属机构,系统用户名
- ConstantPO: 城市距离、价格等常量 PO 类,包括两个城市,价格,距离
- UserPO;用户信息PO类,包括用户名和密码,权限持久化对象如OrderPO的定义如下图所示
- LogisticsPO: 物流信息 PO 类,包括快递编号,货运状态
- ListPO: 待审批单据 PO 类,一个含有待审批单据对象的数组 持久化对象如 OrderPO 的定义如下所示。

```
package po;
import enumClass.DeliveryType;
import java.io.Serializable;

/**
 * Created by Administrator on 2015/10/23 0023.
 */
public class OrderPO extends ListPO implements Serializable {
   private String senderName;
   private String senderAddress;
   private String senderWorkplace;
   private String senderTelephone;
   private String senderTelephone;
   private String senderPhone;
```

```
private String receiverName;
    private String receiverAddress;
    private String receiverWorkplace;
    private String receiverTelephone;
    private String receiverPhone;
    private int original Num;
    private double weight;
    private double volume:
    private String goods Name;
    private enumClass. DeliveryType DeliveryType;
    private String wrapper;
    private double expenseOfTransport;
    private double expenseOfWrapper;
    private double expense:
    private double date;
    public OrderPO(String senderName, String senderAddress, String
senderWorkplace, String senderTelephone,
                   String senderPhone, String receiverName, String
receiverAddress, String receiverWorkplace,
                   String receiverTelephone, String receiverPhone, int
originalNum, double weight, double volume,
                   String
                            goods_Name,
                                          DeliveryType
                                                         DeliveryType,
String wrapper) {
        this. senderName = senderName;
        this.senderAddress = senderAddress;
        this.senderWorkplace = senderWorkplace;
        this. senderTelephone = senderTelephone;
        this. senderPhone = senderPhone;
        this.receiverName = receiverName:
        this.receiverAddress = receiverAddress;
        this.receiverWorkplace = receiverWorkplace;
        this.receiverTelephone = receiverTelephone;
        this.receiverPhone = receiverPhone;
        this.originalNum = originalNum;
        this.weight = weight;
        this. volume = volume;
        this. goods Name = goods Name;
        this. DeliveryType = DeliveryType;
        this.wrapper = wrapper;
```

```
}
/**
 * @return
 */
public String getSenderName() {
    return senderName;
/**
 * @param senderName
 */
public void setSenderName(String senderName) {
    this.senderName = senderName;
public String getSenderAddress() {
    return senderAddress;
public void setSenderAddress(String senderAddress) {
    this.senderAddress = senderAddress;
public String getSenderWorkplace() {
    return senderWorkplace;
public void setSenderWorkplace(String senderWorkplace) {
    this.senderWorkplace = senderWorkplace;
}
public String getSenderTelephone() {
    return senderTelephone;
public void setSenderTelephone(String senderTelephone) {
    this. senderTelephone = senderTelephone;
public String getSenderPhone() {
```

```
return senderPhone;
public void setSenderPhone(String senderPhone) {
    this. senderPhone = senderPhone;
public String getReceiverName() {
   return receiverName;
public void setReceiverName(String receiverName) {
    this.receiverName = receiverName;
public String getReceiverAddress() {
   return receiverAddress;
public void setReceiverAddress(String receiverAddress) {
    this.receiverAddress = receiverAddress;
public String getReceiverWorkplace() {
   return receiverWorkplace;
public void setReceiverWorkplace(String receiverWorkplace) {
    this.receiverWorkplace = receiverWorkplace;
public String getReceiverTelephone() {
   return receiverTelephone;
public void setReceiverTelephone(String receiverTelephone) {
    this.receiverTelephone = receiverTelephone;
public String getReceiverPhone() {
   return receiverPhone;
public void setReceiverPhone(String receiverPhone) {
```

```
this.receiverPhone = receiverPhone;
}
public int getOriginalNum() {
   return originalNum;
public void setOriginalNum(int originalNum) {
    this.originalNum = originalNum;
public double getWeight() {
   return weight;
public void setWeight(double weight) {
    this.weight = weight;
public double getVolume() {
   return volume;
public void setVolume(double volume) {
    this.volume = volume;
public String getGoods_Name() {
   return goods_Name;
public void setGoods_Name(String goods_Name) {
    this.goods_Name = goods_Name;
}
public DeliveryType getDeliveryType() {
   return DeliveryType;
public void setDeliveryType (DeliveryType DeliveryType) {
    this. DeliveryType = DeliveryType;
public String getWrapper() {
```

```
return wrapper;
}
public void setWrapper(String wrapper) {
    this.wrapper = wrapper;
public double getExpenseOfTransport() {
   return expenseOfTransport;
public void setExpenseOfTransport(double expenseOfTransport) {
    this.expenseOfTransport = expenseOfTransport;
public double getExpenseOfWrapper() {
   return expenseOfWrapper;
public void setExpenseOfWrapper(double expenseOfWrapper) {
    this.expenseOfWrapper = expenseOfWrapper;
}
public double getExpense() {
   return expense;
public void setExpense(double expense) {
    this.expense = expense;
public double getDate() {
   return date;
}
public void setDate(double date) {
   this. date = date;
```

6.2 Txt 持久化格式

各 txt 数据保存格式如下所示:

每行属性之间用英文分号";"分隔,若属性为列表,则列表项间用竖线

"1"分隔,列表项内部用英文逗号","分隔。

OperationLog.txt

操作时间;操作人;操作类型 2015-10-23 20:15;管理员;新建用户

Vehicle.txt

车辆代号;车牌号;购买时间;服役时间; 025-000-001;A 00000;2015-1-2;2015-1-10

Driver.txt

司机编号(城市编号(电话号码区号南京 025)+营业厅编号(000 三位数字)+000 三位数字);姓名;出生日期;身份证号;手机;车辆单位;性别;行驶证期限

025001001;蒙奕锟;1985-10-10;450803198510109178;18795855882;南京营业厅;男;2018-10-10

Stock.txt

快递编号;入库日期;目的地;区号;排号;架号;位号 123456789;2015-1-1;北京;5;5;5;5

Account.txt

账户名:余额

Test;1000.5

Worker.txt

姓名;身份证号;职位;所属机构;系统用户名 蒙奕锟; 450803198510109178;中转中心营业员;南京营业厅;myk123

Constant.txt

城市 1;价格;城市 2;距离 南京;20.7;北京;900

User.txt

用户名;密码;权限 myk123;qwer13;总经理

Salary.txt

人员类型;薪水策略营业厅业务员;按月