**5.接口视角**

**5.1模块的职责**

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

启动模块

展示层

业务逻辑层

网络模块

图7 客户端模块视图

启动模块

网络模块

数据层

图8 服务器端模块视图

**表2 客户端各层的职责**

|  |  |
| --- | --- |
| **层** | **职责** |
| 启动模块 | 负责初始化网络通信机制，启动用户界面 |
| 用户界面层 | 基于窗口的物流公司客户端用户界面 |
| 业务逻辑层 | 对于用户界面的输入进行响应并进行业务处理逻辑 |
| 客户端网络模块 | 利用JavaRMI机制查找RMI服务 |

**表3 服务器端各层的职责**

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

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

**表4 层之间调用的接口**

|  |  |  |
| --- | --- | --- |
| **接口** | **服务调用方** | **服务提供方** |
| informationBLService  sendBLService  judgementBLService  shipmentBLService  receivementBLService  commodityBLService  moneyBLService  reportBLService  staffBLService | 客户端展示层 | 客户端业务逻辑层 |
| informationDataService  sendDataService  judgementDataService  shipmentDataService  receivementDataService  commodityDataService  moneyDataService  reportDataService  staffDataService | 客户端业务逻辑层 | 服务器端数据层 |

以下图9是寄件用例层之间的调用。每一层之间都是由上层依赖了一个接口（需接口），而下层实现这个接口（供接口）。sendblservice提供了send界面所需要的所有业务逻辑功能。senddataservice提供了对数据库的增、删、改、查等操作。这样的实现就大大降低了层与层之间的耦合。

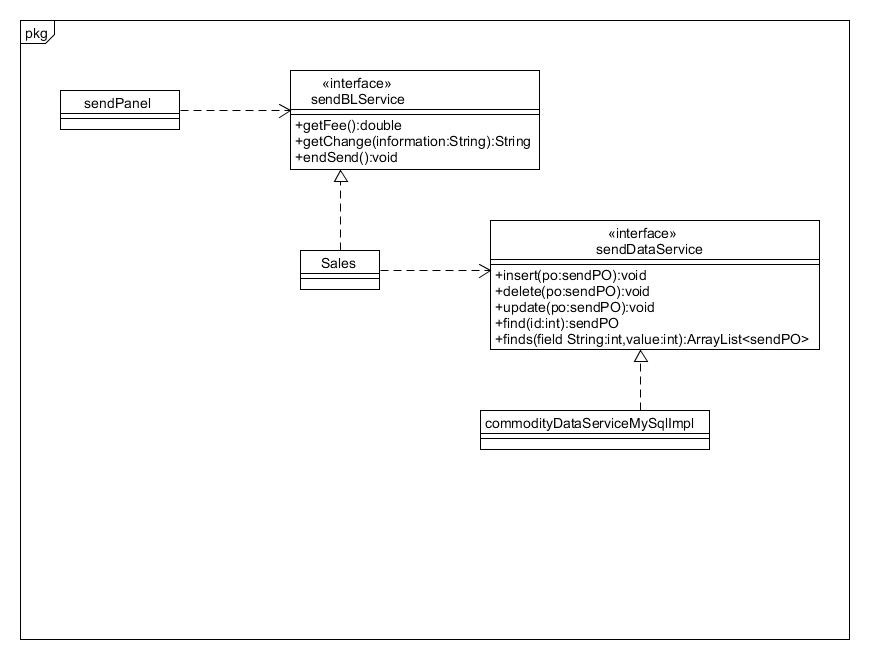


图9 寄件用例层之间调用的接口