# 股票分析系统体系结构设计文档

学 院：南京大学软件学院

成 员：常德隆 武秀峰 曹鸿荣 陈进

完成日期：2017 年3月5日

[股票分析系统体系结构设计文档 1](#_Toc23339)

[1、引言 3](#_Toc5676)

[1.1 编制目的 3](#_Toc32186)

[1.2 词汇表 3](#_Toc1679)

[1.3 参考资料 3](#_Toc20996)

[2、产品描述 3](#_Toc4214)

[3、逻辑视角 3](#_Toc15005)

[4、组合视角 5](#_Toc29667)

[4.1 开发包图 5](#_Toc10320)

[4.2 运行时进程 9](#_Toc5116)

[4.3 物理部署 10](#_Toc30988)

[5、接口视角 10](#_Toc18640)

[5.1 模块的职责 10](#_Toc14105)

[5.2 用户界面层的分解 13](#_Toc1072)

[5.3 业务逻辑层的分解 15](#_Toc773)

[5.4 数据层的分解 21](#_Toc2670)

[6、信息视角 23](#_Toc1514)

[6.1 数据持久化对象 23](#_Toc17923)

[6.2 Txt持久化格式 25](#_Toc30462)

[6.3数据库表 25](#_Toc13672)

## 1、引言

### 1.1 编制目的

在已经提供的 05 年 2 月到 14 年 4 月 791 只股票 1048575 条 A 股信息数据以及未来可能添加新的数据的基础上，通过对数据的分析处理，实现使用量化交易算法的股票回溯和演算系统，用以深入分析股票行情，模拟和比较量化模型。

### 1.2 词汇表

无

### 1.3 参考资料

无

## 2、产品描述

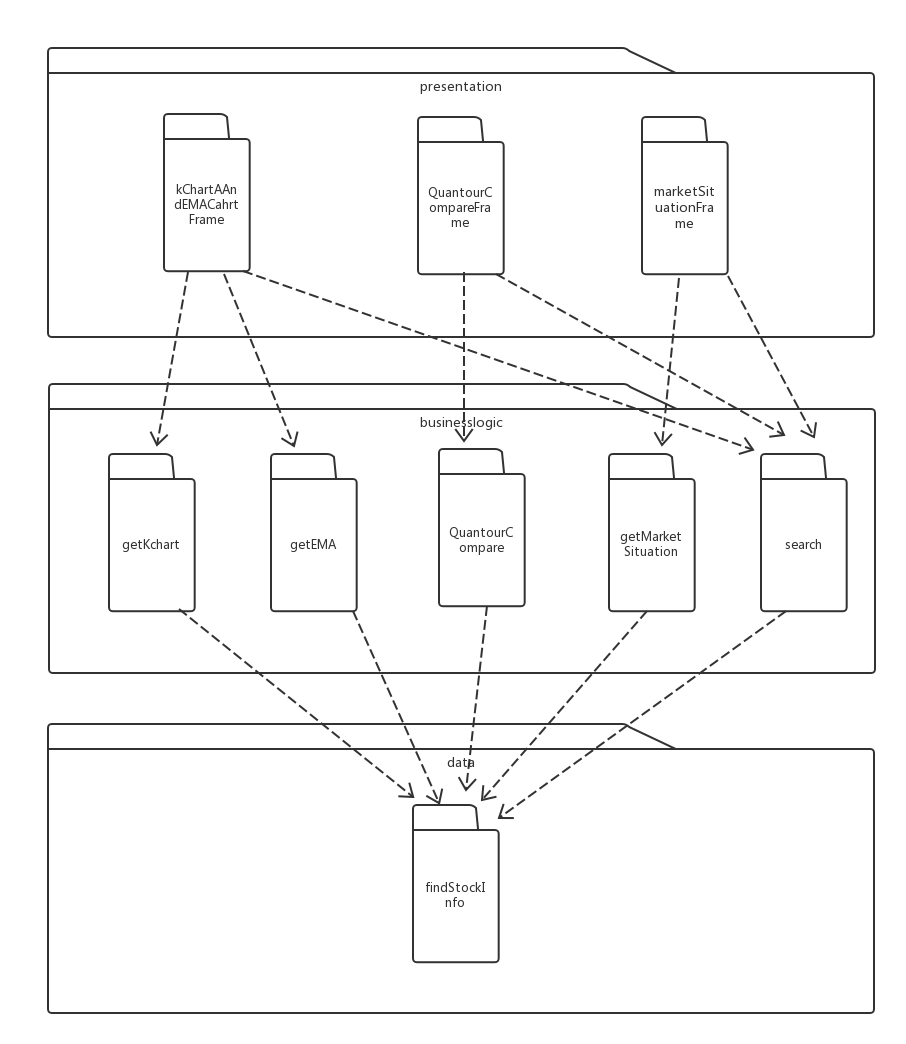
参考股票分析系统用例文档和股票分析系统软件需求规格说明中对产品的概括描述。

## 3、逻辑视角

股票分析系统中，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1和图2所示。



图一 参照体系结构风格的包图表达逻辑视角



图二 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 股票分析系统的最终开发包设计

|  |  |
| --- | --- |
| 开发（物理）包 | 依赖的其他开发包 |
| mainFrame | kChartAndEMAChartFrame,QuantourCompareFrame,marketSituationFrame,VO |
| kChartAndEMAChartFrame | kChartAndEMAChartFrameBlSer,界面类库包,VO |
| quantourCompareFrame | QuantourCompareFrameblSer,界面类库包,VO |
| marketSituationFrame | marketSituationFrameBlSer,界面类库包,VO |
| getkChartBlSer |  |
| getEMABlSer |  |
| quantourCompareblSer |  |
| getMarketSituationBlSer |  |
| searchBlSerl |  |
| getkChart | getChartBlSer,findStockInfo,po |
| getEMA | getEMABlSer,findStockInfo,po |
| quantourCompare | QuantourCompare BlSer,findStockInfo,po |
| getMarketSituation | getMarketSituation BlSer,findStockInfo,po |
| search | searchBlSer,findStockInfo,po |
| findStockInfoDataSer | Java RMI,po |
| findStockDataInfo | Java RMI,po,databaseutility |
| VO |  |
| PO |  |
| utilitybl |  |
| databaseutility | java.JDBC |
| 界面类库包 |  |
| JAVA RMI |  |

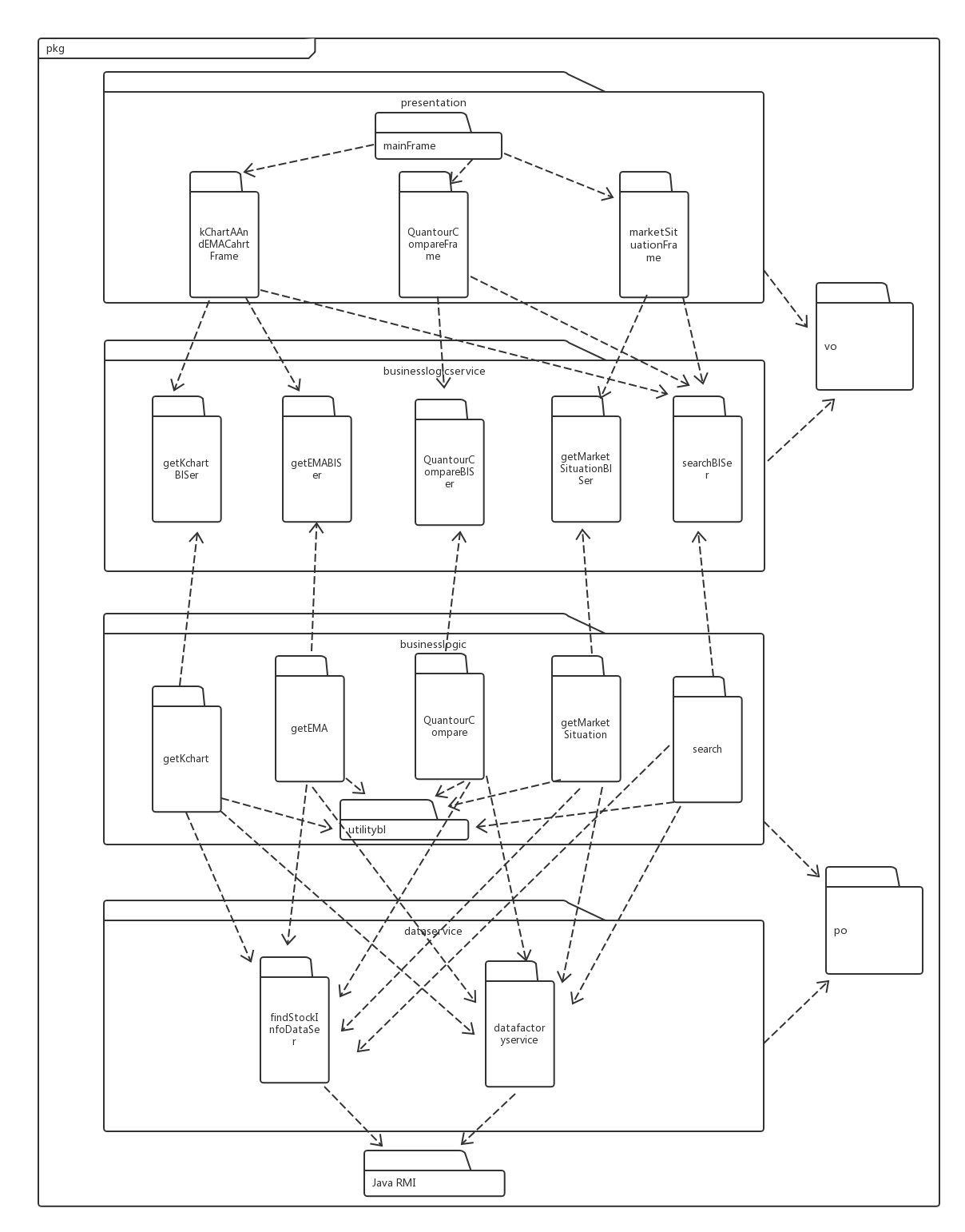


图4.1-1 股票分析系统客户端开发包图

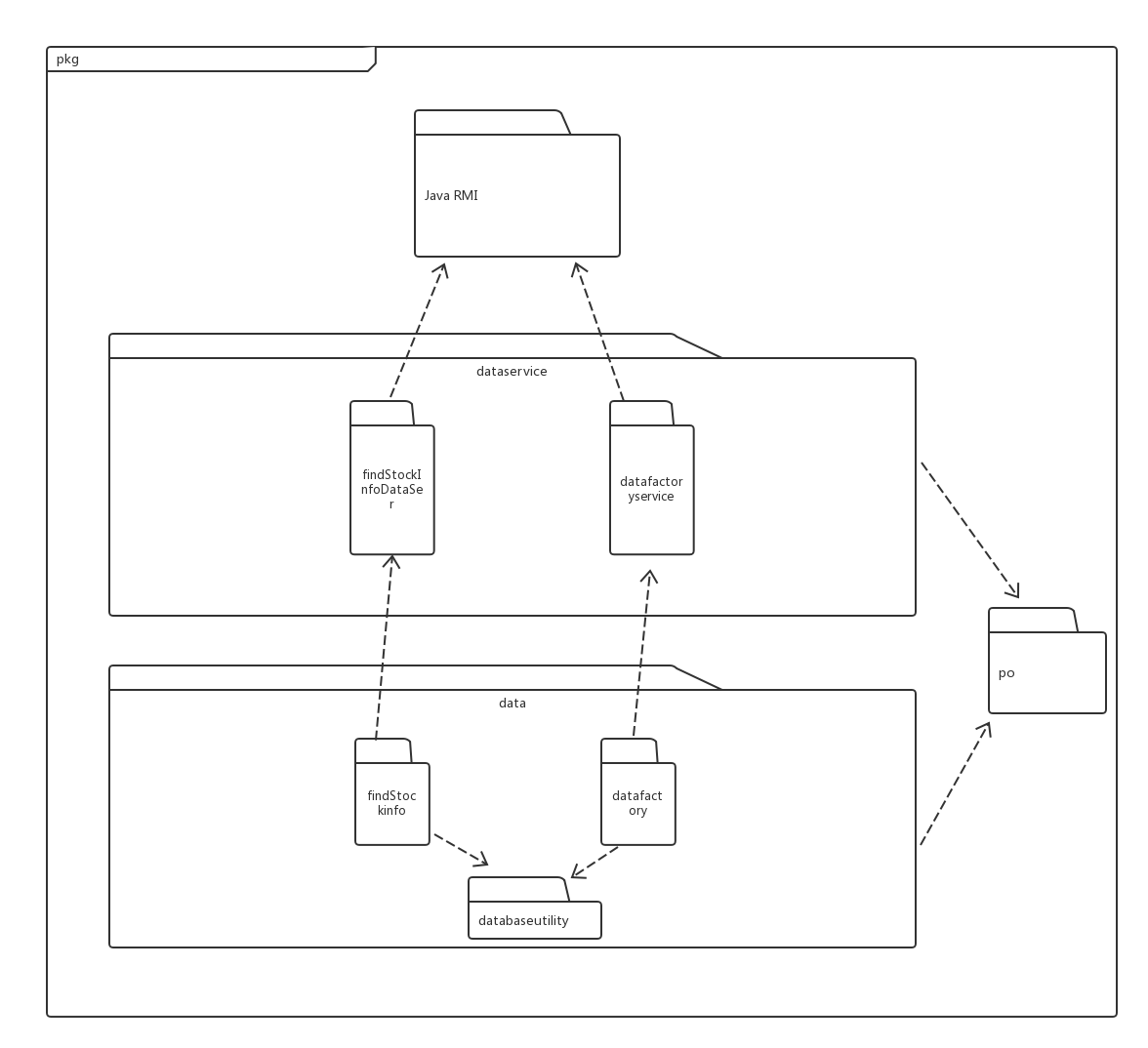


图4.1-2 股票分析系统服务器端开发包图

### 4.2 运行时进程

在股票分析系统中，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所示。结合部署图，客户端进程实在客户端机器上运行，服务器端进程是在服务器端机器上运行。

图4.2 进程图

### 4.3 物理部署

股票分析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JavaRMI构件属于JDK6.0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所示。



4.3 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 | 基于窗口的股票分析系统客户端用户界面。 |
| 业务逻辑层 | 对于用户界面的输入响应和业务处理逻辑。 |
| 客户端网络模块 | 利用Java RMI机制查找RMI服务,检测网络连接状态，进行断线重连 |

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。

|  |  |  |
| --- | --- | --- |
| 接口 | 服务调用方 | 服务提供方 |
| GetKchartBlSer  GetEMABlSer  QuantourCompareBlSer  GetMarketSituationBlSer  SearchBlSer | 客户端展示层 | 客户端业务逻辑层 |
| FindStockInfoDataSer | 客户端业务逻辑层 | 服务器端数据层 |

借用用户管理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）。GetKChartBlSer提供了KChartAndEMAChartFrame界面所需要的所有业务逻辑功能FindStockInfoDataSer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。

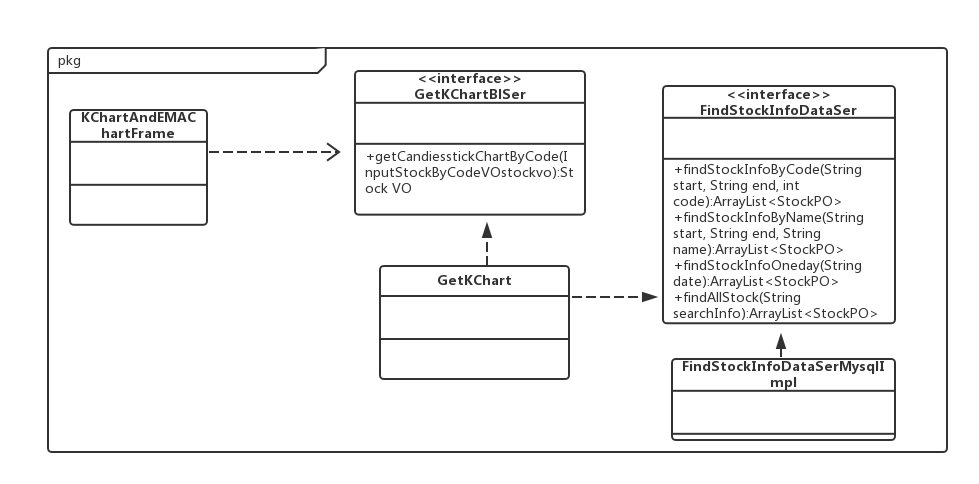


图5.1-3 画k线用例层之间调用的接口

### 5.2 用户界面层的分解

根据需求，系统存在4个用户界面：主界面、K线均线界面、股票比较界面、股票市场界面。

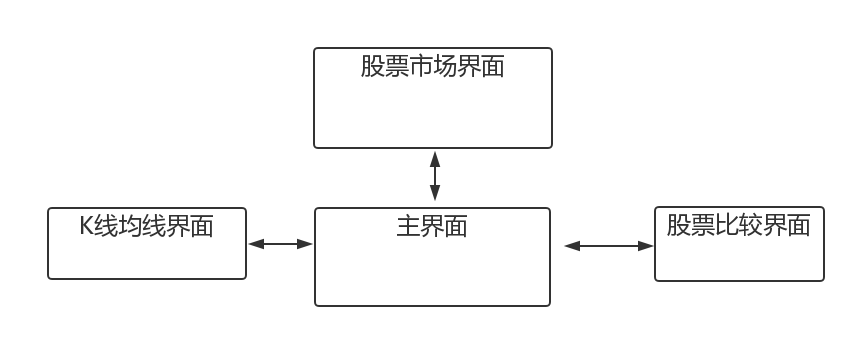


图5.2 用户界面跳转

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。



#### 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。

|  |  |
| --- | --- |
| 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 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示主Frame |
| KChartAndEMAChartFrame | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示K线均线界面 |
| QuantourCompareFrame | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示股票比较界面 |
| MarketSituationFrame | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示股票市场界面 |

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

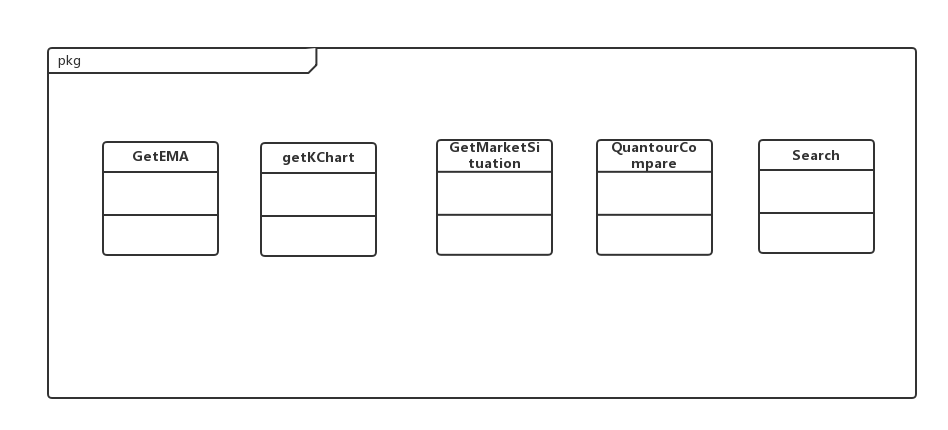
|  |  |
| --- | --- |
| 服务名 | 服务 |
| businesslogicservice. GetEMAChartBlSer | 处理与均线相关接口 |
| businesslogicservice. GetKChartBlSer | 处理K线相关接口 |
| businesslogicservice. GetMarketSituationBlSer | 处理股票比较相关的接口 |
| businesslogicservice. quantourCompareBlSer | 处理股票市场相关接口 |
| businesslogicservice. SearchBlSer.java | 处理搜索相关接口 |

#### 5.2.3 用户界面模块设计原理

用户界面利用Java的Swing和AWT库来实现。

### 5.3 业务逻辑层的分解

务逻辑层包括多个针对界面的业务逻辑处理对象。例如，QuantourCompare对象负责处理股票比较界面的业务逻辑。业务逻辑层的设计如图5.3-1所示。



#### 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 |
| getEMA | 负责均线相关的服务 |
| getKChart | 负责K线相关的服务 |
| getMarketSituation | 负责股票比较相关的服务 |
| quantourCompare | 负责股票市场相关的服务 |
| search | 负责搜索相关的服务 |

#### 5.3.2 业务逻辑层模块的接口规范

表5.3.2-1 expressorder\_bl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| GetEMA. getEMAChartByCode | 语法 | public StockVO getEMAChartByCode(InputStockByCodeVO stockvo); | |
| 前置条件 | 已知股票编码 | |
| 后置条件 | 返回股票均线 | |
| GetEMA. getEMAChartByName | 语法 | public StockVO getEMAChartByName(InputStockByNameVO stockvo); | |
| 前置条件 | 已知股票名字 | |
| 后置条件 | 返回股票均线 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| FindStockInfo. findStockInfoByCode | | | 查找股票 |
| FindStockInfo. findStockInfoByName | | | 查找股票 |
| FindStockInfo. findStockInfoOneday | | | 查找股票 |
| FindStockInfo. findAllStock | | | 查找股票 |

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| GetKChart .getKChartByCode | 语法 | public StockVO getCandiesstickChartByCode(InputStockByCodeVO stockvo)； | |
| 前置条件 | 已知股票编码 | |
| 后置条件 | 返回股票K线 | |
| GetKChart . getCandiesstickChartByName | 语法 | public StockVO getCandiesstickChartByName(InputStockByNameVO stockvo)； | |
| 前置条件 | 已知股票名字 | |
| 后置条件 | 返回股票K线 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| FindStockInfo. findStockInfoByCode | | | 查找股票 |
| FindStockInfo. findStockInfoByName | | | 查找股票 |
| FindStockInfo. findStockInfoOneday | | | 查找股票 |
| FindStockInfo. findAllStock | | | 查找股票 |

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| GetMarketSituation. getMarketSituation | 语法 | public MarketSituationVO getMarketSituation(String date) | |
| 前置条件 | 已知日期 | |
| 后置条件 | 返回股票市场情况 | |
| GetMarketSituation. getIncreaseList | 语法 | public ArrayList<StockVO> getIncreaseList(String date)； | |
| 前置条件 | 已知日期 | |
| 后置条件 | 返回股票市场涨停数 | |
| GetMarketSituation. getDecreaseList | 语法 | public ArrayList<StockVO> getDecreaseList(String date) | |
| 前置条件 | 已知日期 | |
| 后置条件 | 返回股票市场跌停数 | |
| GetMarketSituation. getVolumeList | 语法 | public ArrayList<StockVO> getVolumeList(String date) | |
| 前置条件 | 已知日期 | |
| 后置条件 | 返回股票市场交易量 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| FindStockInfo. findStockInfoByCode | | | 查找股票 |
| FindStockInfo. findStockInfoByName | | | 查找股票 |
| FindStockInfo. findStockInfoOneday | | | 查找股票 |
| FindStockInfo. findAllStock | | | 查找股票 |

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| QuantourCompare. quantourTableCompareByCode | 语法 | public ArrayList<StockCompareTotalVO> quantourTableCompareByCode(ArrayList<InputStockByCodeVO> stockvo)； | |
| 前置条件 | 已知两个股票编码 | |
| 后置条件 | 返回两股票比较情况 | |
| QuantourCompare. quantourChartCompareByCode | 语法 | publicArrayList<ArrayList<StockCompareEverydayVO>>quantourChartCompareByCode (ArrayList<InputStockByCodeVO> stockvo)； | |
| 前置条件 | 已知两个股票编码 | |
| 后置条件 | 返回两股票比较情况 | |
| QuantourCompare. quantourTableCompareByName | 语法 | public ArrayList<StockCompareTotalVO> quantourTableCompareByName(ArrayList<InputStockByNameVO> stockvo)； | |
| 前置条件 | 已知两个股票名字 | |
| 后置条件 | 返回两股票比较情况 | |
| QuantourCompare. quantourChartCompareByName | 语法 | public ArrayList<StockCompareTotalVO> quantourTableCompareByName(ArrayList<InputStockByNameVO> stockvo) | |
| 前置条件 | 已知两个股票名字 | |
| 后置条件 | 返回两股票比较情况 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| FindStockInfo. findStockInfoByCode | | | 查找股票 |
| FindStockInfo. findStockInfoByName | | | 查找股票 |
| FindStockInfo. findStockInfoOneday | | | 查找股票 |
| FindStockInfo. findAllStock | | | 查找股票 |

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Search. findAllStock | 语法 | public ArrayList<StockVO> findAllStock(String searchInfo); | |
| 前置条件 | 已知股票部分信息 | |
| 后置条件 | 返回股票全部信息 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| FindStockInfo. findStockInfoByCode | | | 查找股票 |
| FindStockInfo. findStockInfoByName | | | 查找股票 |
| FindStockInfo. findStockInfoOneday | | | 查找股票 |
| FindStockInfo. findAllStock | | | 查找股票 |

### 5.4 数据层的分解

数据层主要给业务逻辑层提供数据防伪服务，包括对于持久化数据的增、删、改、查。getEMA业务逻辑需要的服务由findStockInfoDataSer接口提供。由于持久化数据的保存可能存在多种形式：Txt文件、序列化文件、数据库等，所示抽象了数据服务。数据层模块的具体描述如图5.4所示。

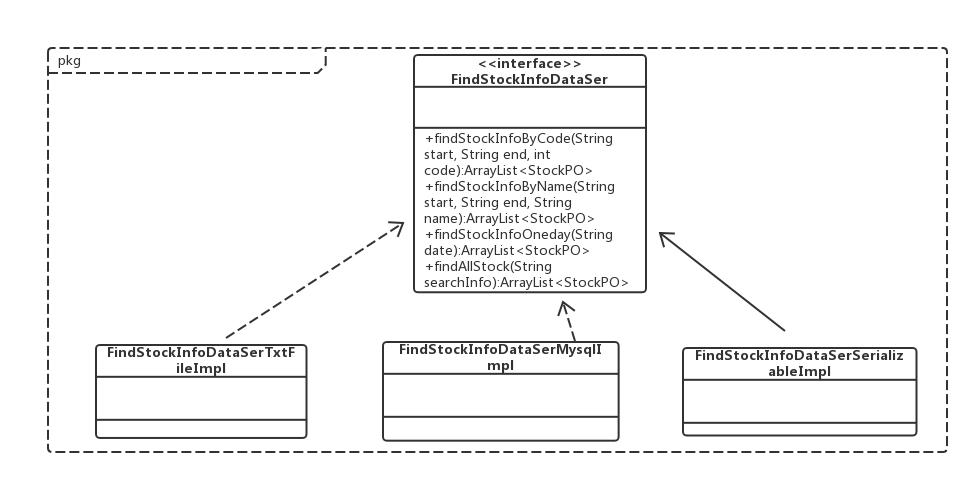


图5.4 数据层模块的描述

#### 5.4.1 数据层模块的职责

数据层模块的职责如表5.4.1所示。

表5.4.1 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。 |
| DataServiceTxtImpl | 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 |
| DataServiceMySqlImpl | 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 |
| DataServiceDatabaseImpl | 基于数据库的持久化数据的接口，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

表5.4.2-1数据层findStockInfo模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| FindStockInfo. findStockInfoByCode | 语法 | public ArrayList<StockPO> findStockInfoByCode(String start, String end, int code); |
| 前置条件 | 无 |
| 后置条件 | 返回相应的StockPO |
| FindStockInfo. findStockInfoByName | 语法 | public ArrayList<StockPO> findStockInfoByName(String start, String end, String name); |
| 前置条件 | 无 |
| 后置条件 | 返回相应的StockPO |
| FindStockInfo. findStockInfoOneday | 语法 | public ArrayList<StockPO> findStockInfoOneday(String date) |
| 前置条件 | 无 |
| 后置条件 | 返回相应的StockPO |
| FindStockInfo. findAllStock | 语法 | public ArrayList<StockPO> findAllStock(String searchInfo) |
| 前置条件 | 无 |
| 后置条件 | 返回相应的StockPO |

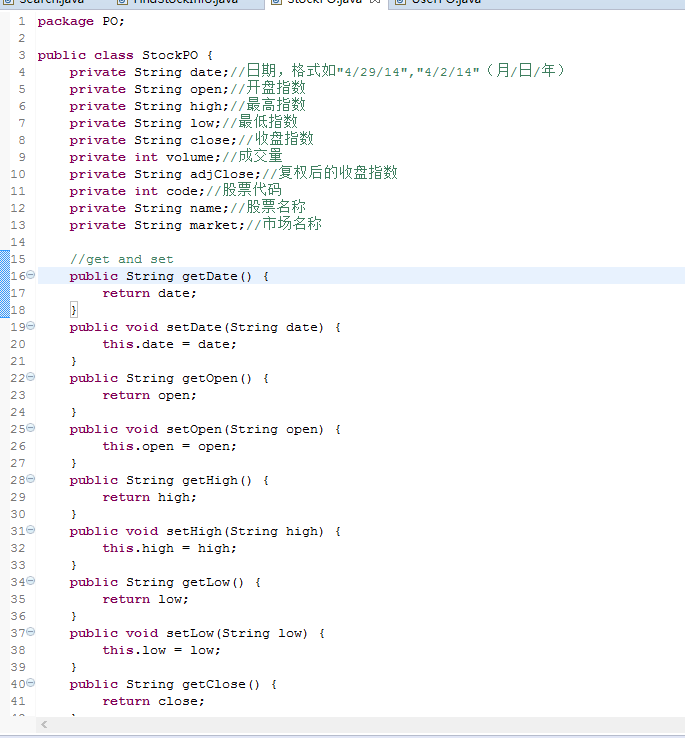
## 6、信息视角

### 6.1 数据持久化对象

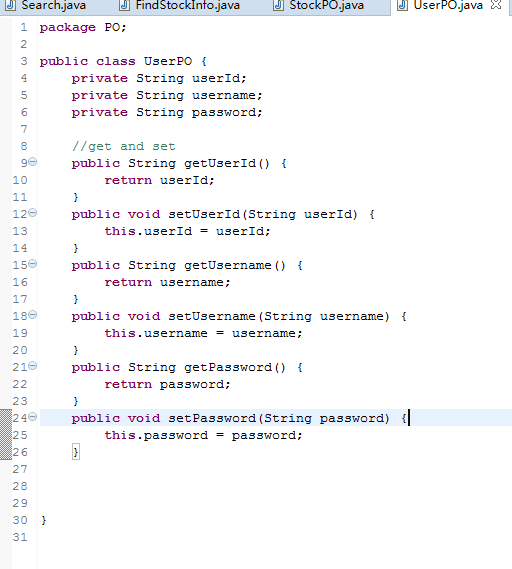
StockPO类包含股票的开盘指数、最高指数、最低指数等属性。

UserPO类包括用户的id、姓名、密码。

持久化用户对象如StockPO的定义如图: 更多定义见原型代码



持久化用户对象如UserPO的定义如图: 更多定义见原型代码



### 6.2 Txt持久化格式

无

### 6.3数据库表

无