Quantour

**软件设计文档**

南京大学软件学院2015级Clam Team

费慧通，冯超，何林洋，梁先伟

2017.03.07

目录

[更新历史 2](#_Toc448516206)

[1.引言 3](#_Toc448516207)

[1.1编制目的 3](#_Toc448516208)

[1.2参考资料 3](#_Toc448516209)

[2.产品概述 3](#_Toc448516210)

[3.逻辑视角 3](#_Toc448516211)

[4.组合视角 5](#_Toc448516212)

[4.1开发包图 5](#_Toc448516213)

[4.2物理部署 6](#_Toc448516214)

[5.接口视角 6](#_Toc448516215)

[5.1模块的职责 6](#_Toc448516216)

[5.1.1用户界面层模块的职责 7](#_Toc448516217)

[5.1.2用户界面模块设计原理 7](#_Toc448516218)

[5.2业务逻辑层的分解 7](#_Toc448516219)

[5.3数据层的分解 1](#_Toc448516220)1

[5.3.1数据层模块的职责 1](#_Toc448516221)1

[5.3.2数据层模块的接口规范 1](#_Toc448516222)1

**更新历史**

|  |  |  |  |
| --- | --- | --- | --- |
| **修改人员** | **修改日期** | **修改原因** | **版本号** |
| 梁先伟 | 2017.03.07 | 初稿 | V0.1 |
| 梁先伟 | 2017.03.10 | 修订版 | V0.2 |
| 梁先伟 | 2017.03.11 | 修订版 | V0.3 |
| 梁先伟 | 2017.03.12 | 修订版 | V0.4 |
| 梁先伟 | 2017.03.13 | 修订版 | V0.5 |
| 梁先伟 | 2017.03.16 | 修订版 | V0.6 |
| 梁先伟 | 2017.03.16 | 正式版 | V1.1 |

# 引言

## 1.1编制目的

本文档详细完成对股票分析软件的详细设计，达到指导后续软件构造的目的，同时实现和测试人员及用户的沟通。

本文档面向开发人员、测试人员及最终用户编写，是了解该软件的导航。

## 1.2参考资料

IEEE标准；

《股票展示分析系统需求规格说明文档》 。

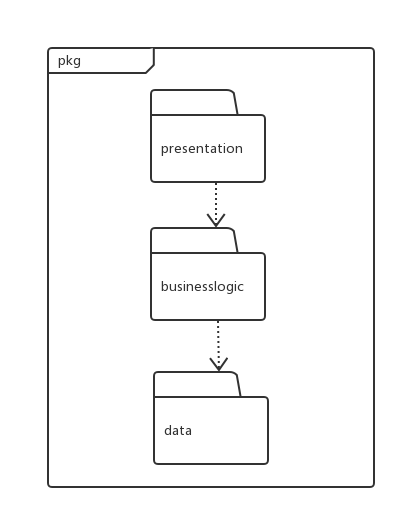
# 产品概述

# 参考《股票展示分析系统需求规格说明文档》中对产品的概括描述。

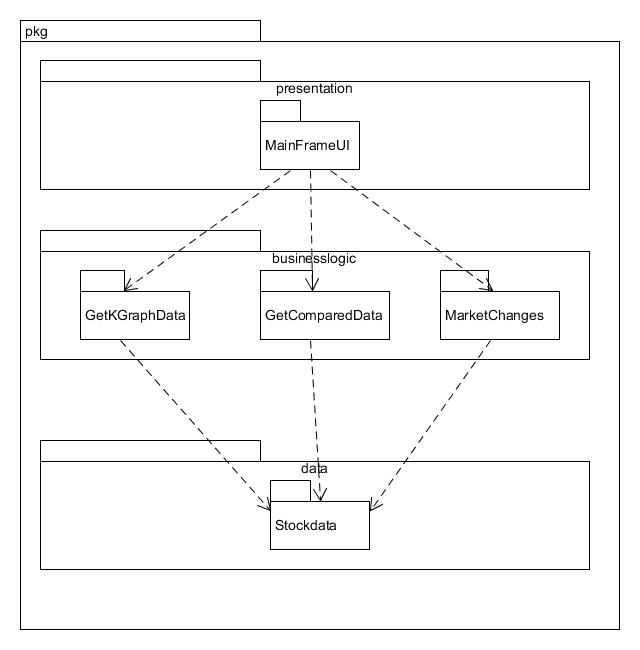
**3.逻辑视角**

股票分析软件中，我们小组选择了分层体系结构风格，将软件系统分为3层（展示层、业务逻辑层、数据层）。

展示层主要负责界面框架的实现，业务逻辑层则是对业务逻辑处理的实现，迭代一中数据层负责从文件中读取数据。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**

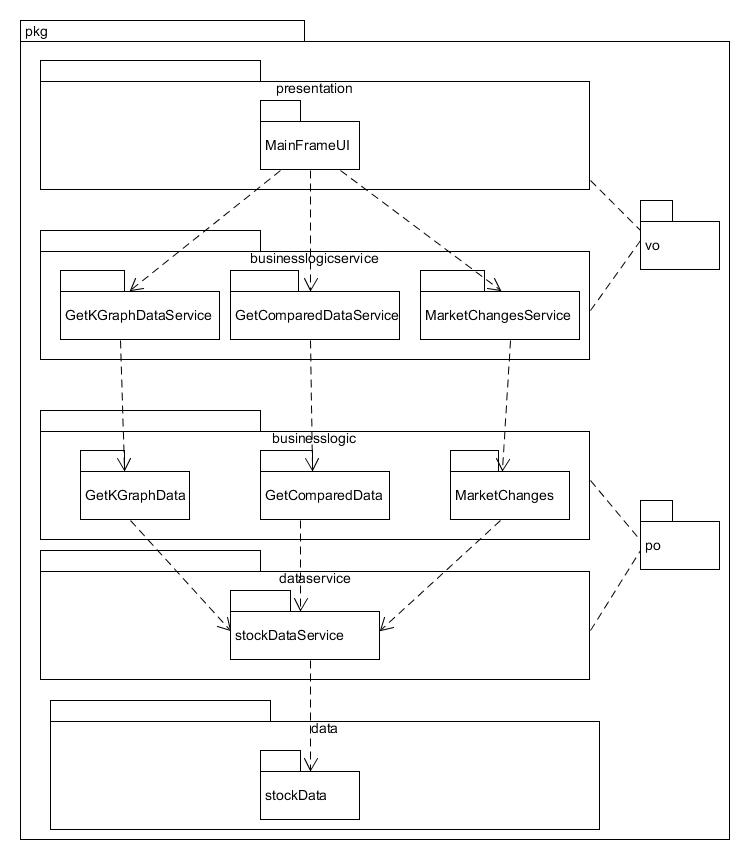
****

**图3-1 软件体系结构逻辑设计方案**

1. **组合视角**

**4.1开发包图**

**图4.1-1 开发包图**



**4.2物理部署**

# 接口视角

## **5.1模块的职责**

**表5.1-1 各层的职责**

|  |  |
| --- | --- |
| **层** | **职责** |
| 用户界面层 | 展示股票信息 |
| 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 |
| 数据层 | 从文件中获取数据并保存，并做简单的数据处理 |

**表5.1-2 层之间调用的接口**

|  |  |  |
| --- | --- | --- |
| **接口** | **服务调用方** | **服务提供方** |
| getKGraphDataService | 客户端展示层 | 客户端业务逻辑层 |
| getComparedDataService |
| MarketChangesService |
| StockDataService | 客户端业务逻辑层 | 客户端数据层 |

### 5.1.1用户界面层模块的职责

如表5.1.1-1所示为用户界面层模块的职责。

**表5.1.1-1 用户界面层模块的职责**

|  |  |
| --- | --- |
| **模块** | **职责** |
| MainFrame | 分为三个部分，分别负责展示个股K线图+均线图、两只股票数据比较和市场行情相关数据。 |

**5.1.2用户界面模块设计原理**

用户界面主要利用利用JavaFX实现。

**5.2业务逻辑层的分解**

**5.2.1 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** |
| getKGraphDataService | 实现K线图和均线图的数据拉取和处理功能 |
| getComparedDataService | 实现两只股票某一时间行情表现差异情况的显示 |
| MarketChangesService | 实现某一日期的股票交易市场行情相关数据的展示 |

**5.2.2 业务逻辑层模块的接口规范**

**5.2.2.1** getKGraphData**模块**

**表5.2.2.1** getKGraphData**模块的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| **提供的服务（供接口）** | | | |
| getKGraphData.getKGraphData | 语法 | | **public** OHLCSeriesCollection getKGraphData(**int** Code, String Begin, String End) |
| 前置条件 | | Data层成功拿到数据 |
| 后置条件 | | 对数据进行K线图所需计算，并对计算所得数据进行存储 |
| getKGraphData.getAverageData | 语法 | | **public** TimeSeriesCollection getAverageData(**int** Code, String Begin, String End) |
| 前置条件 | | Data层成功拿到数据 |
| 后置条件 | | 对数据进行均线图所需计算，并对计算所得数据进行进行存储 |
| getKGraphData.getVolumeData | 语法 | | **public** TimeSeriesCollection getVolumeData(**int** Code, String Begin, String End); |
| 前置条件 | | Data层成功拿到数据 |
| 后置条件 | | 对数据进行成交量柱状图所需计算，并对计算所得数据进行进行存储 |
| getKGraphData.getHighValues | 语法 | | **public** Map<String,Double> getHighValues(String Code, String Begin, String End); |
| 前置条件 | | Data层成功拿到数据 |
| 后置条件 | | 对数据的最大值进行算，并对计算所得数据进行进行存储 |
| getKGraphData.getCodeAndName | 语法 | | **public** ArrayList<stockVO> getCodeAndName(); |
| 前置条件 | | Data层成功拿到数据 |
| 后置条件 | | 读取出文件中所有的股票标号和名称 |
| getKGraphData.GetDateOfSuspension | 语法 | | **public** ArrayList<Date> GetDateOfSuspension(String Code, String Begin, String End) **throws** ParseException |
| 前置条件 | | Data层成功拿到数据 |
| 后置条件 | | 得到回归日期并对计算所得数据进行进行存储 |
| **需要的服务（需接口）** | | | |
| 服务名 | | 服务 | |
| stockData.getStockByCodeAndDate(**int** Code,String Begin,String End) | | 获得指定编号股票的指定日期内的数据 | |
| stockData. getStockByNameAndDate(String Name,String Begin,String End) | | 获得指定名称股票的指定日期内的数据 | |
| stockData.JudgeIfTheLast(**int** Code, String Begin) | | 判断是否到了股票发行的那一天 | |
| stockData.getCodeAndName() | | 读取出文件中所有的股票标号和名称 | |

**5.2.2.2** getComparedData**模块**

**表5.2.2.2** getComparedData**模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| getComparedData.getLowestValue | 语法 | **public** DefaultCategoryDataset getLowestValue(String Name1, String Name2, String Begin, String End); |
| 前置条件 | Data层已经获得所有的数据 |
| 后置条件 | 对用户所选择的股票的最低值数据进行处理，得到数据集 |
| getComparedData.GetLowest | 语法 | **public** String GetLowest(String Name1, String Name2, String Begin, String End); |
| 前置条件 | Data层已经获得所有的数据 |
| 后置条件 | 对用户所选择的股票的最低值数据进行处理，得到数据 |
| getComparedData.getHighestValue | 语法 | **public** DefaultCategoryDataset getHighestValue(String Name1, String Name2, String Begin, String End); |
| 前置条件 | Data层已经获得所有的数据 |
| 后置条件 | 对用户所选择的股票的最高值数据进行处理，得到数据集 |
| getComparedData.GetHighest | 语法 | **public** String GetHighest(String Name1, String Name2, String Begin, String End); |
| 前置条件 | Data层已经获得所有的数据 |
| 后置条件 | 对用户所选择的股票的最高值数据进行处理，得到数据 |
| getComparedData.getRoseAndDropValue | 语法 | **public** DefaultCategoryDataset getRoseAndDropValue(String Name1, String Name2, String Begin, String End); |
| 前置条件 | Data层已经获得所有的数据 |
| 后置条件 | 对用户所选择的股票的涨跌幅数据进行处理，得到数据集 |
| getComparedData.getRoseAndDrop | 语法 | **public** String getRoseAndDrop(String Name1, String Name2, String Begin, String End); |
| 前置条件 | Data层已经获得所有的数据 |
| 后置条件 | 对用户所选择的股票的涨跌幅数据进行处理，得到数据 |
| getComparedData.getCloseValue | 语法 | **public** DefaultCategoryDataset getCloseValue(String Name1, String Name2, String Begin, String End); |
| 前置条件 | Data层已经获得所有的数据 |
| 后置条件 | 对用户所选择的股票的收盘值数据进行处理，得到数据集 |
| getComparedData.getRateOfReturn | 语法 | **public** DefaultCategoryDataset getRateOfReturn(String Name1, String Name2, String Begin, String End); |
| 前置条件 | Data层已经获得所有的数据 |
| 后置条件 | 对用户所选择的股票的对数收益率数据进行处理，得到数据集 |
| getComparedData.getVariance | 语法 | **public** String getVariance(String Name1, String Name2, String Begin, String End); |
| 前置条件 | Data层已经获得所有的数据 |
| 后置条件 | 对用户所选择的股票的对数收益率数据进行方差处理，得到数据集 |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| stockData.getStockByCodeAndDate(**int** Code,String Begin,String End) | 获得指定编号股票的指定日期内的数据 | |
| stockData. getStockByNameAndDate(String Name,String Begin,String End) | 获得指定名称股票的指定日期内的数据 | |
| stockData.getStockByDate(String Begin,String End) | 获得指定日期内的股票的数据 | |
| stockData. getVolumeByDateAndCode(**int** Code,String Begin); | 获得指定日期和股票号的股票的交易量，若该日期不是工作日，返回0；否则返回1号股票的交易量，即非0； | |
| stockData. getVolumeByDateAndName(String Name,String Begin) | 获得指定日期和股票名的股票的交易量，若该日期不是工作日，返回0；否则返回1号股票的交易量，即非0； | |

**5.2.2.3** MarketChanges**模块**

**表5.2.2.3** MarketChanges**模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| MarketChanges.getMarketChanges | 语法 | **public** MarketQuotationVO getMarketChanges(); |
| 前置条件 | Data层已经获得所有的数据 |
| 后置条件 | 对某一日期的股市行情数据进行分析计算 |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| stockData.getStockByDate(String Begin) | 获得指定日期的股票的数据 | |
| stockData. getVolumeByDateAndCode(**int** Code,String Begin); | 获得指定日期和股票号的股票的交易量，若该日期不是工作日，返回0；否则返回1号股票的交易量，即非0； | |
| stockData. getVolumeByDateAndName(String Name,String Begin) | 获得指定日期和股票名的股票的交易量，若该日期不是工作日，返回0；否则返回1号股票的交易量，即非0； | |

**5.3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，包括对数据的增、删、改、查。

**5.3.1数据层模块的职责**

数据层模块的职责如表5.3.1-1所示。

|  |  |
| --- | --- |
| **模块** | **职责** |
| stockData | 负责对文件中的股票数据进行读取 |

**表5.3.1-1 数据层模块的职责**

**5.3.2数据层模块的接口规范**

**5.3.2.1 stockData模块**

**表5.3.2.1 stockData模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| StockData.getStockByCodeAndDate | 语法 | **public** ArrayList<stockPO> getStockByCodeAndDate(**int** Code,String Begin,String End); |
| 前置条件 | 文件中存储着股票所有的数据 |
| 后置条件 | 读取出文件中符合用户所选择编号和起始日期的股票数据 |
| StockData.getStockByNameAndDate | 语法 | **public** ArrayList<stockPO> getStockByNameAndDate(String Name,String Begin,String End); |
| 前置条件 | 文件中存储着股票所有的数据 |
| 后置条件 | 读取出文件中用户所选择名称和起始日期的股票数据 |
| StockData.getStockByDate | 语法 | **public** ArrayList<stockPO> getStockByDate(String Begin) |
| 前置条件 | 文件中存储着股票所有的数据 |
| 后置条件 | 读取出文件中用户所选择起始日期的股票数据 |
| StockData.getVolumeByDateAndCode | 语法 | **public** **int** getVolumeByDateAndCode(**int** Code,String Begin) |
| 前置条件 | 文件中存储着股票所有的数据 |
| 后置条件 | 获得指定日期股票的交易量，若该日期不是工作日，返回0；否则返回1号股票的交易量，即非0； |
| StockData.getVolumeByDateAndName | 语法 | **public** **int** getVolumeByDateAndName(String Name,String Begin) |
| 前置条件 | 文件中存储着股票所有的数据 |
| 后置条件 | 获得指定日期股票的交易量，若该日期不是工作日，返回0；否则返回1号股票的交易量，即非0； |
| StockData.JudgeIfTheLast | 语法 | **public** **int** JudgeIfTheLast(**int** Code, String Begin) |
| 前置条件 | 文件中存储着股票所有的数据 |
| 后置条件 | 根据日期和股票号得到当日交易量，若该日期不是工作日，返回0；否则返回股票1号的交易量，即非0；是股票最开始一天则返回-1； |
| StockData.getCodeAndName | 语法 | **public** ArrayList<stockPO> getCodeAndName() |
| 前置条件 | 文件中存储着股票所有的数据 |
| 后置条件 | 读取出文件中所有的股票标号和名称 |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| Data.getAllStock(StringbeginDate, String endDate, String stockID) | 获得所有年份和交易所的所有股票数据 | |