**股票查询分析系统**

Quantra Stock Analysis System

项目设计文档

(迭代二)

南京大学软件学院 Quadra团队

李珺 陈锐 梁家铭 罗宇堃

2017年3月22日

**目录**

[更新历史 3](#_Toc477573437)

[1. 引言 4](#_Toc477573438)

[1.1 编制目的 4](#_Toc477573439)

[1.2 参考资料 4](#_Toc477573440)

[2. 产品概述 4](#_Toc477573441)

[3. 逻辑视角 5](#_Toc477573442)

[4. 组合视角 6](#_Toc477573443)

[5. 接口视角 7](#_Toc477573444)

[5.1 模块的职责 7](#_Toc477573445)

[5.2 用户界面层的分解 7](#_Toc477573446)

[5.3 控制层的分解 7](#_Toc477573447)

[5.4 数据层的分解 8](#_Toc477573448)

[6. 信息视角 9](#_Toc477573449)

[6.1 数据对象 9](#_Toc477573450)

# 更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| 修改人员 | 修改日期 | 修改原因 | 版本号 |
| 全体成员 | 2017/3/20 | 项目启动 | V1.0 |
| 陈锐 | 2017/4/7 | 更新接口定义以符合最新系统 | V1.1 |

# 引言

## 编制目的

本文档描述对股票查询分析系统（Quantra）的概要设计，达到指导详细设计和开放的目的，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而编写，是了解系统的导航。

## 参考资料

* + 1. 《股票查询分析系统(迭代一)需求规格说明文档》
    2. 《软件工程与计算（卷二）》
    3. 《软件工程与计算（卷三）》
    4. IEEE标准

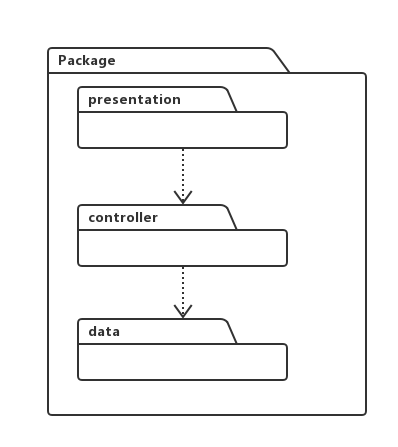
# 产品概述

随着中国经济的发展，许多大公司在证券交易所上市交易获得更多资金促进公司发展，而其他人民也试图通过买卖股票来获取利润。然而股市有风险，入市需谨慎，股民经常会遇到信息短缺，股市行情难以分析等问题。

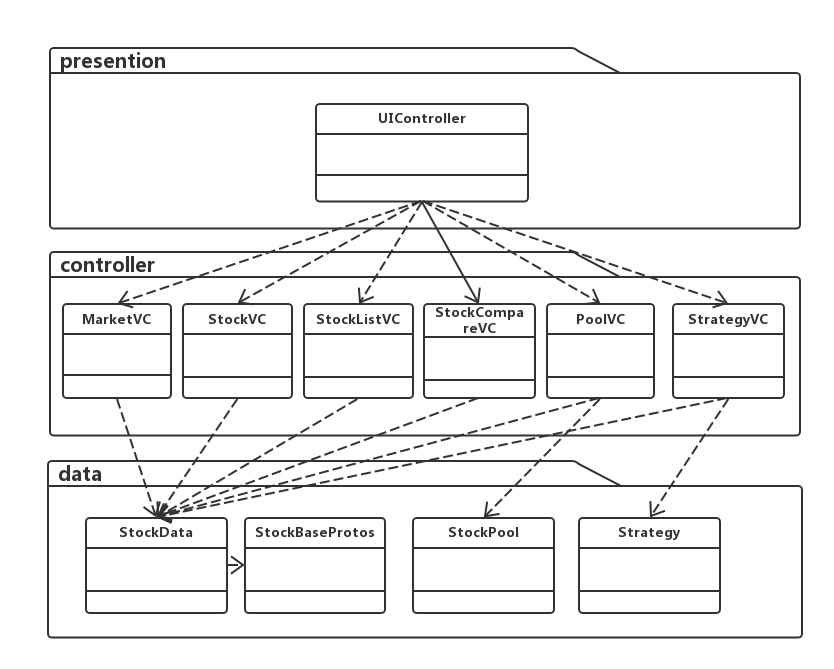
为了解决上述问题，方便股民在股市中及时掌握信息，决定开发一个股票查询分析系统，来帮助股民能够做出合理的判断。

# 逻辑视角

股票查询分析系统（Quantra）选择了分层体系结构风格，将系统分为3层（界面层、控制层、数据层）能够很好地示意整个高层抽象。界面层包含GUI的实现，控制层包含界面控制的处理，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下图所示。



# 组合视角

股票查询分析系统（Quantra）的包图如下：

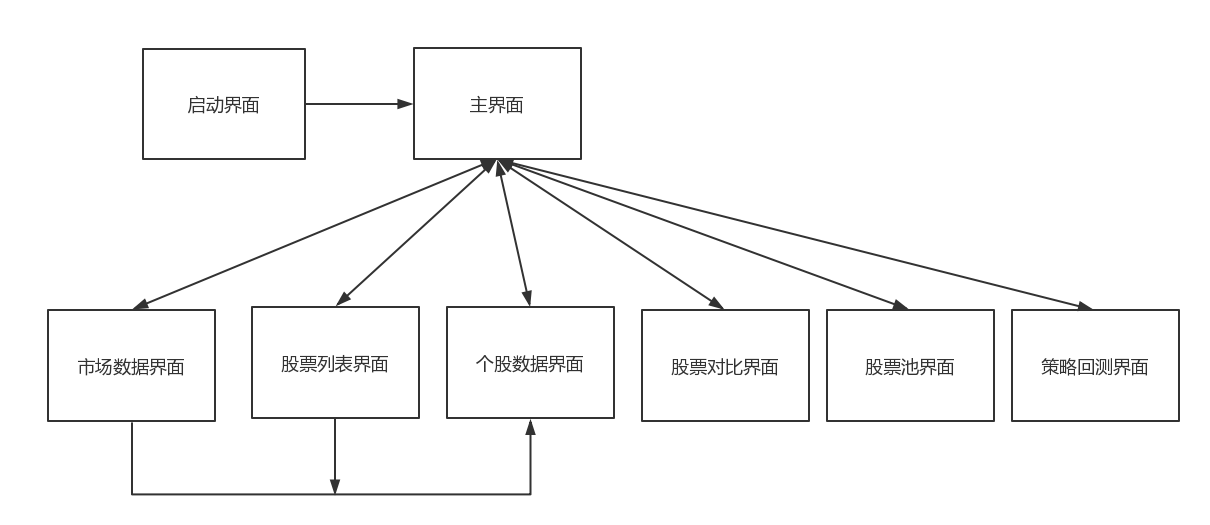
# 接口视角

## 模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| 启动模块 | 负责初始化软件，启动用户界面。 |
| 界面层 | 基于窗口的互联网酒店股票查询分析系统（Quantra）用户界面。 |
| 控制层 | 对于用户界面的输入响应和界面控制处理逻辑。 |
| 数据层 | 负责数据的持久化及数据访问接口。 |

## 用户界面层的分解

根据需求，系统存在6个用户界面：启动界面、主界面、市场数据界面、股票列表界面、个股数据界面、股票对比界面。用户界面利用Java的JavaFX库来实现，跳转风格为委托式，界面跳转如下图所示。



## 控制层的分解

控制层包括多个针对界面的逻辑处理对象。控制层模块的职责如下表所示。

|  |  |
| --- | --- |
| 模块 | 职责 |
| SplashVC | 负责实现启动界面所需要的服务 |
| UIContainer | 负责实现主界面所需要的服务，以及界面的跳转 |
| MarketVC | 负责实现市场数据界面所需要的服务 |
| StockListVC | 负责实现股票列表界面所需要的服务 |
| StockVC | 负责实现个股数据界面所需要的服务 |
| StockCompareVC | 负责实现股票对比界面所需要的服务 |
| PoolVC | 负责实现股票池界面所需要的服务 |
| StrategyVC | 负责实现股票回测界面所需要的服务 |

## 数据层的分解

数据层主要给控制层提供数据访问服务，包括对于结构化数据的查询，以及将非结构化数据转换为结构化数据。数据层模块的职责如下表所示。

|  |  |
| --- | --- |
| 模块 | 职责 |
| StockData | 提供股票数据的访问服务，以及将非结构化数据转换为结构化数据。 |
| StockPoolData | 提供股票池数据的访问服务 |
| StrategyData | 提供回测策略的访问服务 |

**StockData模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **StockData.** **loadProtobuf** | **语法** | void loadJSON(Label status) |
| **前置条件** | 磁盘中存在结构化的数据 |
| **后置条件** | 从磁盘加载结构化的数据 |
| **StockData.**  **loadCSV** | **语法** | void loadCSV(Label status) |
| **前置条件** | 磁盘中存在CSV格式数据 |
| **后置条件** | 将CSV格式数据转换为结构化数据 |
| **StockData.**  **getPtrList** | **语法** | List<StockInfoPtr> getPtrList() |
| **前置条件** | 数据文件中存在股票信息 |
| **后置条件** | 返回分别指向所有股票信息的指针 |
| **StockData.**  **getIndex** | **语法** | List<StockInfoPtr> getIndex() |
| **前置条件** | 数据文件中存在股票信息 |
| **后置条件** | 返回股票的索引指针 |
| **StockData.**  **getByDate** | **语法** | List<StockInfoPtr> getByDate(String date) |
| **前置条件** | 数据文件中存在指定日期的股票信息 |
| **后置条件** | 返回分别指向指定日期所有股票信息的指针 |
| **StockData.**  **getByCode** | **语法** | List<StockInfoPtr> getByCode(int code) |
| **前置条件** | 数据文件中存在指定代码的股票信息 |
| **后置条件** | 返回分别指向指定代码所有股票信息的指针 |

**StockPoolData模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **StockPoolData.**  **getPoolMap** | **语法** | Map<String, CustomPool> getPoolMap() |
| **前置条件** | 数据文件中存在指定代码的股票池信息 |
| **后置条件** | 返回所有股票池信息 |
| **StockPoolData.**  **addPool** | **语法** | void addPool(CustomPool pool) |
| **前置条件** | 请求添加股票池信息 |
| **后置条件** | 将股票池信息导入数据文件 |
| **StockPoolData.**  **removePool** | **语法** | Void removePool(CustomPool pool) |
| **前置条件** | 请求删除股票池信息 |
| **后置条件** | 将对应信息从数据文件删除 |
| **StackPoolData.**  **loadFromFile** | **语法** | void loadFromFile() |
| **前置条件** | 磁盘中存在结构化的数据 |
| **后置条件** | 从磁盘加载结构化的数据 |
| **StackPoolData.**  **SaveFromFile** | **语法** | void saveToFile() |
| **前置条件** | 有数据文件存在 |
| **后置条件** | 将数据文件保存到磁盘 |

**StrategyData模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **StrategyData.**  **getStrategyList** | **语法** | List<AbstractStrategy> getStrategyList() |
| **前置条件** | 数据文件中存在指定代码的策略信息 |
| **后置条件** | 返回所有策略信息 |
| **StrategyData.**  **addStrategy** | **语法** | void addStrategy (AbstractStrategy Strategy) |
| **前置条件** | 请求添加策略信息 |
| **后置条件** | 将策略信息导入数据文件 |
| **StrategyData.**  **removeStrategy** | **语法** | Void removeStrategy (AbstractStrategy Strategy) |
| **前置条件** | 请求删除策略信息 |
| **后置条件** | 将对应信息从数据文件删除 |
| **StrategyData.**  **loadFromFile** | **语法** | void loadFromFile() |
| **前置条件** | 磁盘中存在结构化的数据 |
| **后置条件** | 从磁盘加载结构化的数据 |
| **StrategyData.**  **SaveFromFile** | **语法** | void saveToFile() |
| **前置条件** | 有数据文件存在 |
| **后置条件** | 将数据文件保存到磁盘 |

# 信息视角

## 数据对象

StockInfo类包含了股票信息，属性如下表所示。

|  |  |
| --- | --- |
| 类名 | 包含的属性 |
| StockInfo | 股票代码、股票名称、拼音首字母、日期、开盘价、收盘价、最高价、最低价、复权收盘价、交易量、所在市场 |