

南京大学软件学院

2016 春 软件工程 AnyQuant项目实践

**软件详细设计文档**

**Chicer**

**程翔 陈天歌**

**韩奇 韩梦虞**

**一.引言**

**1.编制目的**

本报告详细描述了对AnyQuant系统的详细设计，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。

**2.历史版本**

|  |  |  |  |
| --- | --- | --- | --- |
| 更改人员 | 日期 | 变更内容 | 版本 |
| 陈天歌 | 2016-4-11 | 完善文档 | V1.0 |

**3.参考资料**

1）IEEE标准

2）《迭代二需求规格说明文档》

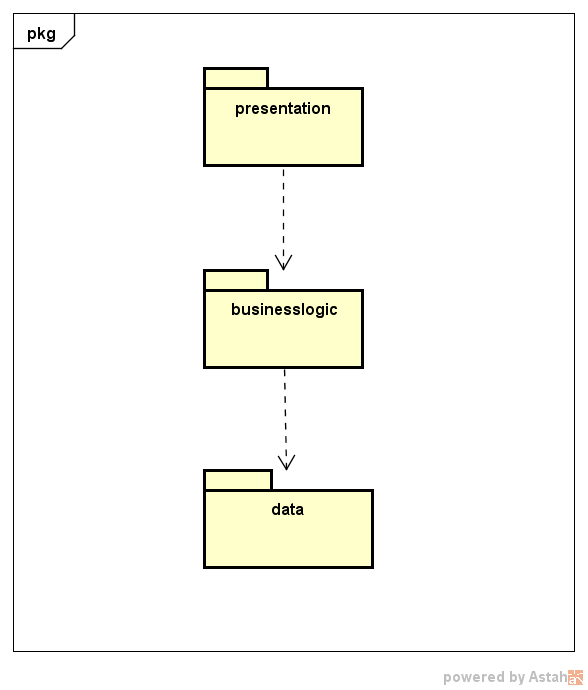
3）《软件工程与计算（卷三）》

**二.产品概述**

参考AnyQuant需求规格说明文档对产品的概括描述。

**三.架构风格**

AnyQuant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下图所示（即其架构风格）。

****

**四．系统设计描述主体**

## 1、逻辑层部分：

1、KLineBLService

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| dayKLineChart | 语法 | KLineData dayKLineChart(String stockName, Calendar start, Calendar end) |
| 前置条件 | 传入股票名称，起始日期和结束日期 |
| 后置条件 | 返回该股票对应时期内的KLineData |
| weekKLineChart | 语法 | KLineData weekKLineChart(String stockName) |
| 前置条件 | 传入指定的股票名称 |
| 后置条件 | 返回该股票周线图对应的KLineData |
| monthKLineChart | 语法 | KLineData monthKLineChart(String stockName) |
| 前置条件 | 传入指定的股票名称 |
| 后置条件 | 返回该股票月线图对应的KLineData |
| getDayAverageLine | 语法 | KLineData getDayAverageLine(String stockName, Calendar start, Calendar end, int aver) |
|  | 前置条件 | 传入股票名，起始日期，结束日期,均线天数 |
|  | 后置条件 | 返回该股票与均线天数对应的日平均线KLineData |
| getWeekAverageLine | 语法 | KLineData getWeekAverageLine(String stockName, int aver) |
|  | 前置条件 | 传入股票名,均线天数 |
|  | 后置条件 | 返回该股票与均线天数对应的周平均线KLineData |
| getMonthAverageLine | 语法 | KLineData getMonthAverageLine(String stockName, int aver) |
|  | 前置条件 | 传入股票名,均线天数 |
|  | 后置条件 | 返回该股票与均线天数对应的月平均线KLineData |
| drawPreview | 语法 | LineChartData drawPreview(String name) |
|  | 前置条件 | 传入股票名 |
|  | 后置条件 | 返回该股票价格波动情况以及涨跌幅、成交量 |
| drawRSI | 语法 | LineChartData drawRSI(String name) |
|  | 前置条件 | 传入股票名 |
|  | 后置条件 | 返回该股票相对强弱指数-三条折线，6、12、24日RSI |
| drawBIAS | 语法 | LineChartData drawBAIS(String name) |
|  | 前置条件 | 传入股票名 |
|  | 后置条件 | 返回该股票乖离率-三条折线，6、12、24日BIAS |
| drawKDJ | 语法 | LineChartData drawKDJ (String name) |
|  | 前置条件 | 传入股票名 |
|  | 后置条件 | 返回该股票随机指标-三条曲线，分别是KDJ的值 |

2、industryBLService

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| getAllIndustries | 语法 | Public List<String> getAllIndustries() |
| 前置条件 | 无 |
| 后置条件 | 返回所有分类行业的名称 |
| getStocksByIndustry | 语法 | Public List<Stock> getStocksByIndustry(String industry) |
| 前置条件 | 传入行业名称 |
| 后置条件 | 返回该行业所有股票信息 |
| getIndustryByName | 语法 | Public String getIndustryByName(String stockName) |
| 前置条件 | 传入股票名称 |
| 后置条件 | 返回该股票所在的行业名 |
| getIndustryByName | 语法 | Public String getIndustryByName(String stockName) |
|  | 前置条件 | 传入股票名称 |
|  | 后置条件 | 返回该股票所在的行业名 |
| getIndustryInfo | 语法 | Public IndustryInfo getIndustryByName(String industry) |
|  | 前置条件 | 传入行业名称 |
|  | 后置条件 | 返回行业信息 |

## 2、数据层部分：

1、IndustryNameDATAService

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **IndustryNameDATAService.getIndustryName** | 语法 | public String getIndustryName(String name) |
| 前置条件 | 传入单只股票代号名 |
| 后置条件 | 返回该股票对应的行业名 |
| **IndustryNameDATAService.getAllIndustries** | 语法 | PublicList<String> getAllIndustries() |
| 前置条件 | 无 |
| 后置条件 | 返回所有股票行业分类的名称 |
| **IndustryNameDATAService.getStockByIndustry** | 语法 | public List<String> getStockByIndustry(String industry) |
| 前置条件 | 传入行业名称 |
| 后置条件 | 返回该行业的所有股票代号 |

2、TurnoverDATAService

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| TurnoverDATAService.getTurnover | 语法 | public Stock getTurnover(String name) |
| 前置条件 | 传入股票代号 |
| 后置条件 | 返回该股票当天成交量和成交金额 |