BUFF TREASURE

**详细设计文档**

Warren Buffett

2017-3-9

# 文档更新纪录：

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 作者 | 描述 | 日期 |
| v1.0 | 王帅惟 |  | 2017年03月09日 |

**目录**（按Ctrl+目录有链接）

[文档更新纪录： 1](#_Toc445316314)

[引言： 3](#_Toc445316315)

[词汇表： 4](#_Toc445316316)

[1.产品概述 4](#_Toc445316317)

[2.系统的分层架构 5](#_Toc445316318)

[3.用户界面层的分解 6](#_Toc445316319)

[4.系统的架构设计 7](#_Toc445316320)

[5.系统的类图设计 8](#_Toc445316321)

[6.系统顺序图 9](#_Toc445316322)

[7.数据层的分解 10](#_Toc445316323)

[8.接口规范设计 12](#_Toc445316324)

9.数据模型

# 引言：

|  |  |
| --- | --- |
| 内容和说明 | 项目 |
| 1.编写的目的（说明整个文档所有达到的目标） | 本文档提供股票数据分析系统系统的软件架构概览，采用若干架构视图描述系统的不同方面，以便表示构造系统所需要的重要架构决策。 |
| 2.对象与范围（说明整个文档的内容范围和针对的读者对象） | 本文档的作者是Warren Buffet团队内部的开发和管理人员 |
| 3.参考文献（说明文档中主要的所需引用） | 1.《软件需求规格说明书》；  《软件架构文档模版》，Rational Software Corporation，2002；  2.丁二玉，刘钦.计算与软件工程（卷二）[M]机械工业出版社，2012：134—182  4.Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Micheael Stal.Pattern-Oriented Software Architecture Volume 1: A system of Patterns [M]机械工业出版社,2003  3Duck Plus团队 详细设计文档 |
| 4.名词与术语(说明文档中常用的技术缩略和先相关词条) |  |
|  |  |

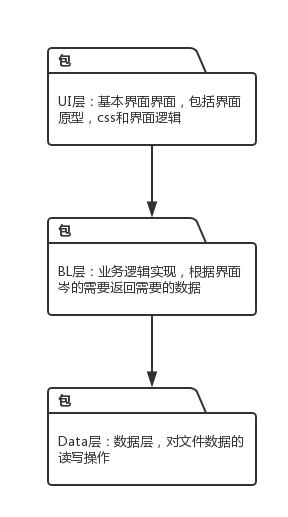
# 词汇表：

|  |  |  |
| --- | --- | --- |
| 词汇名称 | 词汇含义 | 备注 |
| \_ui | 表示某展示层 |  |
| \_bl | 表示某逻辑层 |  |
| \_data | 表示某数据层 |  |
|  |  |  |

# 1.产品概述

Stock Statistic System 是用于统计股票的一些包括开盘价、收盘价、最高价、最低价、成交量、换手率等数据的展示系统，主要由Java语言开发，从API中读取数据，解析数据，并根据合理的逻辑展示在用户界面上，为用户提供一个多方面访问数据的窗口。

# 2.系统的分层架构

图示如下：  


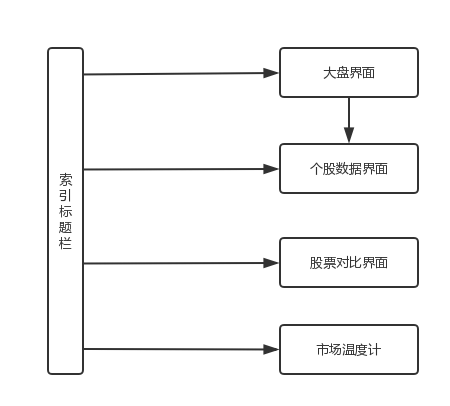
系统划分为一下3个逻辑层次：  
1）UI层：用于前台界面展示和配置的层次  
2）BL层：包含业务控制和逻辑的层次  
3）Data层：定义和储存系统中相关数据的层次

系统可以部署在以下3个物理层次。  
1）访问层：用于用户访问系统的层次。  
2）业务层：部署业务控制和逻辑的层次。  
3）数据层：部署和储存系统中相关数据的层次。

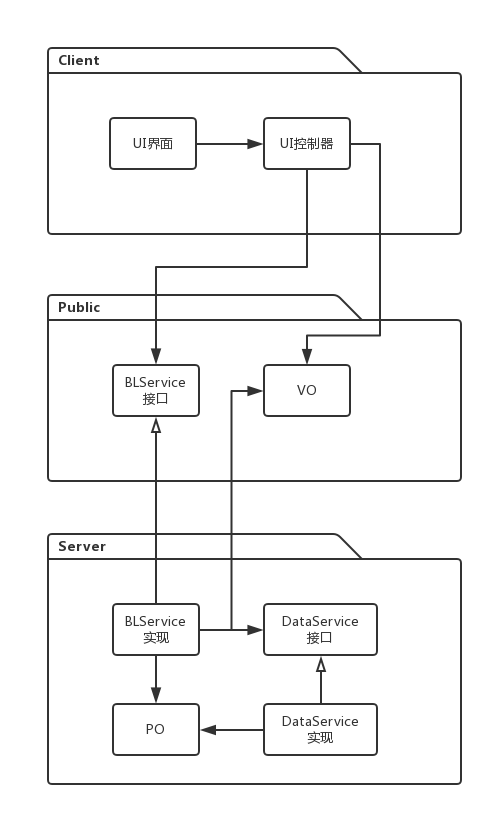
# 3.用户界面层的分解

该系统在迭代一共存在主要界面4个，分别为 大盘界面、个股数据界面、股票对比界面、市场温度计界面。

附加1个索引标题栏

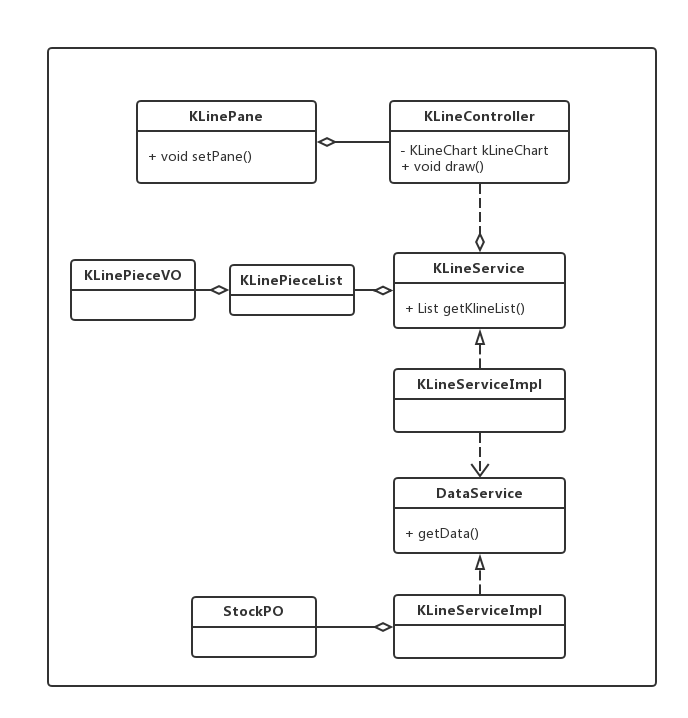


# 4.系统的架构设计

系统架构中分为三个模块：  
1）展示模块，负责处理系统数据的展现和用户的交互。  
3）公有模块，包涵数据层和界面层的接口定义，以及VO的存储。  
4）服务模块，负责业务逻辑的实现，以及数据层的接口和实现。  
5）VO对象，负责封装从本地文件获取的数据并与逻辑、展示层交互。  
6）PO对象，负责与本地文件交互，获取数据。  
  
系统中的模块：  
  


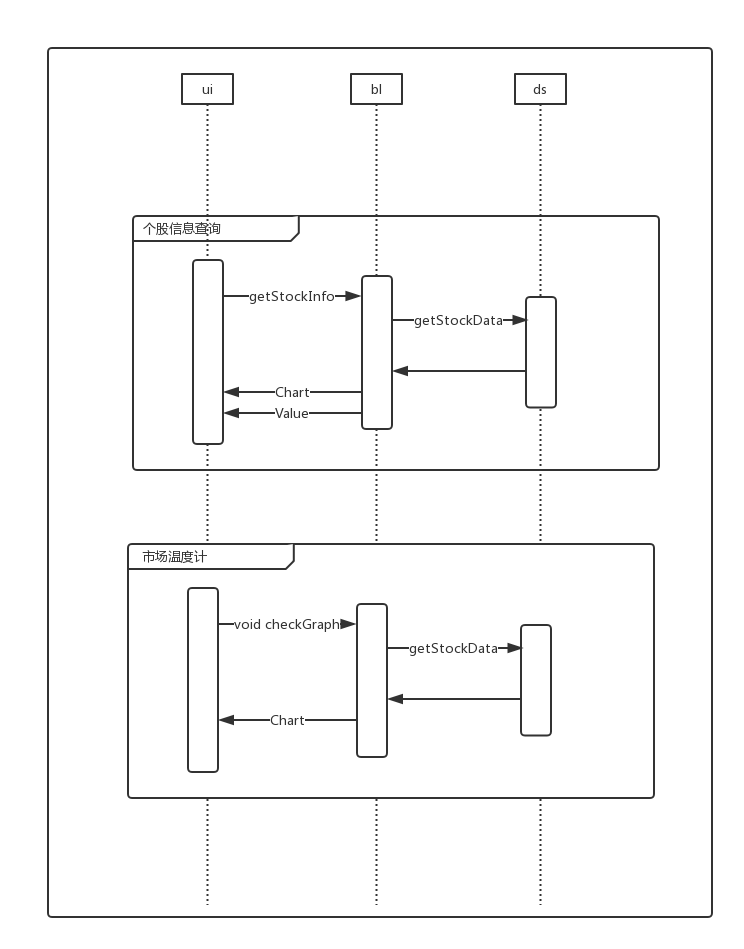
# 5.系统的类图设计

以K线图为例：



# 6.系统顺序图

设计如下：

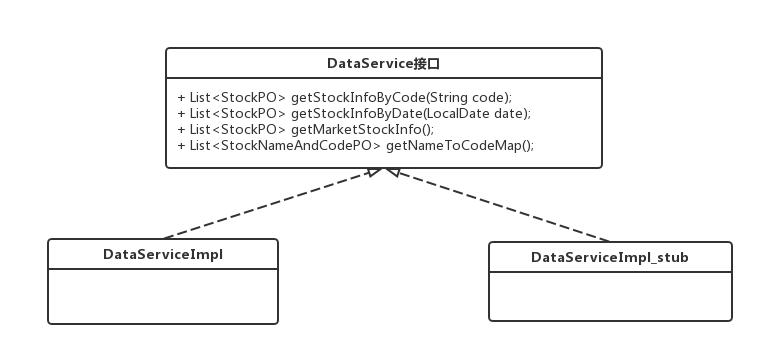


# 7.数据层的分解

数据层主要给业务逻辑层提供数据服务，包括对于持久化数据的增、删、改、查。

（迭代一支提供数据的查询服务）

业务逻辑需要的服务DataService接口提供。数据层模块的具体描述如图



迭代一采用txt文件进行数据存储，所以没有设计数据库以及序列化的存储方式

接口组件设计结构如下表格：

|  |  |  |  |
| --- | --- | --- | --- |
| 接口ID | 连接组件 | 接口信息 | |
| blService | 连接UI与BL  位于BUFFPublic | 语法 | Return(Response)  Factory.create(Request) |
| 前置条件 | 用户的输入正确 |
| 后置条件 | 处理控制组件处理请求并且响应 |
| 不变量 | 用户请求信息 |
| dataservice | 连接BL与DS | 语法 | Return(Response)  Interface(Request) |
| 前置条件 | 文件路径存在 |
| 后置条件 | 对应的DataService执行对应的业务逻辑 |
| 不变量 | 无 |

# 8.接口规范设计

表Comparison. ComparisonService的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| ComparisonService.  init(String mainStockCode, String deputyStockCode, LocalDate beginDate, LocalDate endDate) | 语法 | void init(String mainStockCode, String deputyStockCode, LocalDate beginDate, LocalDate endDate) |
| 前置条件 | 已创建一个ComparisonServiceImpl对象 |
| 后置条件 | 修改ComparisonServiceImpl内部属性的值,并设置日期区间（不用再次访问数据层） |
| ComparisonService. getMainBasisAnalysis() | 语法 | public BasisAnalysisVO getMainBasisAnalysis() |
| 前置条件 | 已创建一个ComparisonServiceImpl对象 |
| 后置条件 | 获取ComparisonServiceImpl领域对象的BasisAnalysisVO |
| ComparisonService. getMainDailyClosingPrice() | 语法 | Public List<DailyClosingPriceVO> getMainDailyClosingPrice() |
| 前置条件 | 已创建一个ComparisonServiceImpl对象 |
| 后置条件 | 获取ComparisonServiceImpl领域对象的BasisAnalysisVO列表, 获得主股每日收盘价信息 |
| ComparisonService. getMainDailyLogReturnAnalysis() | 语法 | public List<DailyLogReturnVO> getMainDailyLogReturnAnalysis() |
| 前置条件 | 已创建一个ComparisonServiceImpl对象 |
| 后置条件 | 调用ComparisonServiceImpl领域对象的DailyLogReturnVO列表，得到主股每日对数收益率 |
| iComparisonService. getMainLogReturnVariance(); | 语法 | public double getMainLogReturnVariance() |
| 前置条件 | 已创建一个ComparisonServiceImpl对象 |
| 后置条件 | 提供主股对数收益率方差 |
| ComparisonService. getDeputyBasisAnalysis() | 语法 | Public BasisAnalysisVO getDeputyBasisAnalysis() |
| 前置条件 | 已创建一个ComparisonServiceImpl对象 |
| 后置条件 | 获得ComparisonServiceImpl领域对象的BasisAnalysisVO对象得到副股总体信息 |
| ComparisonService. getDeputyDailyClosingPrice() | 语法 | Public List<DailyClosingPriceVO> getDeputyDailyClosingPrice() |
| 前置条件 | 已创建一个ComparisonServiceImpl对象 |
| 后置条件 | 获得ComparisonServiceImpl领域对象的BasisAnalysisVO对象列表，得副股每日收盘价信息 |
| ComparisonService. getDeputyDailyLogReturnAnalysis() | 语法 | Public List<DailyLogReturnVO> getDeputyDailyLogReturnAnalysis() |
| 前置条件 | 已创建一个ComparisonServiceImpl对象 |
| 后置条件 | 获得ComparisonServiceImpl领域对象的DailyLogReturnVO对象列表，得到副股每日对数信息 |
| ComparisonService. getDeputyLogReturnVariance() | 语法 | Public double getDeputyLogReturnVariance() |
| 前置条件 | 已创建一个ComparisonServiceImpl对象 |
| 后置条件 | 提供副股对数收益率方差 |
| ComparisonService.getEarliestDate() | 语法 | Public LocalDate getEarliestDate() |
| 前置条件 | 已创建一个ComparisonServiceImpl对象 |
| 后置条件 | 获得分析数据中最早一天的日期 |
| ComparisonService. getLatestDate() | 语法 | Public LocalDate getLatestDate() |
| 前置条件 | 已创建一个ComparisonServiceImpl对象 |
| 后置条件 | 获得分析数据中最晚一天的日期 |
| 需要的服务 | | |
| 服务名 | 服务 | |
| StockDAO. getStockInfoByCode(String code) | 返回StockPO\_列表 | |
| StockDAO. getStockInfoByCode(LocalDate date) | 传输StockPO列表 | |

表market. MarketService的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| MarketService. getMarketDailyKLine(LocalDate beginDate, LocalDate endDate) | 语法 | Public List<KLinePieceVO> getMarketDailyKLine(LocalDate beginDate, LocalDate endDate) |
| 前置条件 | 已创建一个MarketServiceImpl领域对象 |
| 后置条件 | 返回MarketServiceImpl领域对象的KLinePieceVO列表 |
| MarketService. getMarketVol(LocalDate beginDate, LocalDate endDate) | 语法 | Public List<StockVolVO> getMarketVol(LocalDate beginDate, LocalDate endDate) |
| 前置条件 | 已创建一个MarketServiceImpl领域对象 |
| 后置条件 | 返回MarketServiceImpl领域对象的StockVolVO列表 |
| MarketService.getMarketStockDetailVO() | 语法 | public List<MarketStockDetailVO> getMarketStockDetailVO() |
| 前置条件 | 已创建一个MarketServiceImpl领域对象 |
| 后置条件 | 返回MarketServiceImpl领域对象的MarketStockDetailVO列表, 获得所有股票的大盘显示信息 |
| MarketService. getHistoryStockDetailVO(String code) | 语法 | public MarketStockDetailVO getHistoryStockDetailVO(String code) |
| 前置条件 | MarketServiceImpl |
| 后置条件 | 返回MarketServiceImpl领域对象的MarketStockDetailVO, 历史查询中根据股票编号获得股票详细信息 |
| 需要的服务 | | |
| 服务名 | 服务 | |
| StockDAO.getMarketStockInfo() | 历史查询中根据股票编号获得股票详细信息 | |

SingleStock.AllStockService的接口规范

SingleStock. KLineService的接口规范

SingleStock. MALineService的接口规范

SingleStock. StockDetailService的接口规范

SingleStock. VolService的接口规范

放一块了在一个包里 每个都不大

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| AllStockService. getStockVol(String code, LocalDate beginDate, LocalDate endDate) | 语法 | List<StockVolVO> getStockVol(String code, LocalDate beginDate, LocalDate endDate) |
| 前置条件 | 已创建一个AllStockServicImpl对象 |
| 后置条件 | 获取AllStockServicImpl领域对象的StockVolVO数据列表 |
| KLineService. getDailyKLine(String code, LocalDate beginDate, LocalDate endDate) | 语法 | Public List<KLinePieceVO> getDailyKLine(String code, LocalDate beginDate, LocalDate endDate) |
| 前置条件 | 已创建一个KLineServiceImpl对象 |
| 后置条件 | 获取KLineServiceImpl领域对象的KLinePieceVO数据列表 |
| MALineService. getMAInfo(String code, LocalDate beginDate, LocalDate endDate) | 语法 | public List<MAPieceVO> getMAInfo(String code, LocalDate beginDate, LocalDate endDate) |
| 前置条件 | 已创建一个MALineServiceImpl对象 |
| 后置条件 | 获取MALineServiceImpl领域对象的MAPieceVO数据列表 |
| StockDetailService. getStockBriefInfo(String code) | 语法 | public List<StockBriefInfoVO> getStockBriefInfo(String code) |
| 前置条件 | 已创建一个StockDetailServiceImpl对象 |
| 后置条件 | 获取StockDetailServiceImpl领域对象的StockBriefInfoVO数据列表 |
| StockDetailService. getSingleStockDetails(String code, LocalDate date) | 语法 | public StockDetailVO getSingleStockDetails(String code, LocalDate date) |
| 前置条件 | 已创建一个StockDetailVOImpl对象 |
| 后置条件 | 获取StockDetailServiceImpl领域对象的StockDetailVO数据 |
| VolService. getStockVol(String code, LocalDate beginDate, LocalDate endDate) | 语法 | public List<StockVolVO> getStockVol(String code, LocalDate beginDate, LocalDate endDate) |
| 前置条件 | 已创建一个VolServiceImpl对象 |
| 后置条件 | 获取VolServiceImpl领域对象的StockVolVO数据列表 |
| 需要的服务 | | |
| 服务名 | 服务 | |
| StockDAO. getStockInfoByCode(String code) | 返回StockPO\_列表 | |
| StockDAO. getStockInfoByCode(LocalDate date) | 传输StockPO列表 | |
| StockDAO.getMarketStockInfo() | 历史查询中根据股票编号获得股票详细信息 | |

表thermometer. ThermometerService的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| ThermometerService.  getTradingVolume (DateRange dateRange) | 语法 | Public List<StockVolVO> getTradingVolume (DateRange dateRange) |
| 前置条件 | 已创建一个ThermometerServiceImpl对象 |
| 后置条件 | 获取ThermometerServiceImpl领域对象的StockVolVO列表,获得日期范围内的整个股市的每日交易量 |
| ThermometerService. getMainBasisAnalysis() | 语法 | public BasisAnalysisVO getMainBasisAnalysis() |
| 前置条件 | 已创建一个ThermometerServiceImpl对象 |
| 后置条件 | 获取ThermometerServiceImpl领域对象的BasisAnalysisVO |
| ThermometerService. getTradingVolume (DateRange dateRange,String shareID) | 语法 | Public List<StockVolVO> getTradingVolume (DateRange dateRange,String shareID) |
| 前置条件 | 已创建一个ThermometerServiceImpl对象 |
| 后置条件 | 获取ThermometerServiceImpl领域对象的StockVolVO列表获得日期范围内的目标股票的每日交易量 |
| ThermometerService. getLimitUpNum(DateRange dateRange) | 语法 | List<LongPeiceVO> getLimitUpNum (DateRange dateRange) |
| 前置条件 | 已创建一个ThermometerServiceImpl对象 |
| 后置条件 | 获取ThermometerServiceImpl领域对象的LongPeiceVO列表，返回日期内每天的涨停股票数 |
| ThermometerService. getLimitDownNum(DateRange dateRange) | 语法 | public List<LongPeiceVO> getLimitDownNum (DateRange dateRange) |
| 前置条件 | 已创建一个ThermometerServiceImpl对象 |
| 后置条件 | 获取ThermometerServiceImpl领域对象的LongPeiceVO列表，返回日期内每天的跌停股票数 |
| ThermometerService. getRiseOver5Num(DateRange dateRange) | 语法 | Public List<LongPeiceVO> getRiseOver5Num (DateRange dateRange) |
| 前置条件 | 已创建一个ThermometerServiceImpl对象 |
| 后置条件 | 获得ThermometerServiceImpl领域对象的BasisAnalysisVO对象得到副股总体信息, 返回日期内每天的涨幅超过5%的股票数 |
| ThermometerService. getFallOver5Num(DateRange dateRange) | 语法 | Public List<LongPeiceVO> getFallOver5Num (DateRange dateRange) |
| 前置条件 | 已创建一个ThermometerServiceImpl对象 |
| 后置条件 | 获取ThermometerServiceImpl领域对象的LongPeiceVO列表，返回日期内每天的跌幅超过5%的股票数 |
| ThermometerService. g getRiseOver5ThanLastDayNum(DateRange dateRange) | 语法 | Public List<LongPeiceVO> getRiseOver5ThanLastDayNum(DateRange dateRange) |
| 前置条件 | 已创建一个ThermometerServiceImpl对象 |
| 后置条件 | 获取ThermometerServiceImpl领域对象的LongPeiceVO列表，返回日期内每天的 (开盘‐收盘)> 5 % \*上一个交易日收盘价 的股票个数 |
| ThermometerService. getFallOver5ThanLastDayNum(DateRange dateRange) | 语法 | Public List<LongPeiceVO> getFallOver5ThanLastDayNum(DateRange dateRange) |
| 前置条件 | 已创建一个ThermometerServiceImpl对象 |
| 后置条件 | 提获取ThermometerServiceImpl领域对象的LongPeiceVO列表，返回日期内每天的 (开盘‐收盘)<- 5 % \*上一个交易日收盘价 的股票个数 |
| 需要的服务 | | |
| 服务名 | 服务 | |
| StockDAO. getStockInfoByCode(String code) | 返回StockPO\_列表 | |
| StockDAO. getStockInfoByCode(LocalDate date) | 传输StockPO列表 | |

表strategy. StrategyService的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| StrategyService. init(StrategyConditionVO strategyConditionVO, StockPoolConditionVO stockPoolConditionVO, List<StockPickIndexVO> stockPickIndexVOs); | 语法 | void init(StrategyConditionVO strategyConditionVO, StockPoolConditionVO stockPoolConditionVO,  List<StockPickIndexVO> stockPickIndexVOs); |
| 前置条件 | 已创建一个StrategyService对象 |
| 后置条件 | 初始化数据 调用picckledataservice预先 处理部分数据 |
| StrategyService.initMixed(LocalDate beginDate , LocalDate endDate,  StockPoolConditionVO stockPoolConditionVO, List<StockPickIndexVO> stockPickIndexVOs,  TraceBackVO traceBackVO, List<MixedStrategyVO> mixedStrategyVOS); | 语法 | void initMixed(LocalDate beginDate , LocalDate endDate,StockPoolConditionVO stockPoolConditionVO, List<StockPickIndexVO> stockPickIndexVOs,TraceBackVO traceBackVO, List<MixedStrategyVO> mixedStrategyVOS); |
| 前置条件 | 已创建一个StrategyService对象 |
| 后置条件 | 调用pickledataservice 获取混合策略模式下需要谁的数据 |
| StrategyService.calculate(TraceBackVO traceBackVO); | 语法 | Public void calculate(TraceBackVO traceBackVO); |
| 前置条件 | 已创建一个StrategyService对象 已经调用init方法 |
| 后置条件 | 调用pickledataservice 获取单独选中均值或者动量模型的模式下需要谁的数据 |
| StrategyService. getBackDetailVO() | 语法 | BackDetailVO getBackDetailVO() |
| 前置条件 | 已创建一个StrategyService对象,并且实际的pickledata是非空的 |
| 后置条件 | 获取计算过的反馈列表，包含策略评价的一系列信息 |
| StrategyService. getStrategyDayRatePieceVO(); | 语法 | List<DayRatePieceVO> getStrategyDayRatePieceVO(); |
| 前置条件 | 已创建一个StrategyService对象并且实际的pickledata是非空的 |
| 后置条件 | 获得回测累计收益比较图中的策略收益率累积折线图的每个点数据 |
| StrategyService. getBaseDayRatePieceVO(); | 语法 | List<DayRatePieceVO> getBaseDayRatePieceVO(); |
| 前置条件 | 已创建一个StrategyService对象并且实际的pickledata是非空的 |
| 后置条件 | 获得回测累计收益比较图中的基准收益率累积折线图 |
| StrategyService. getBetterTableVOByFormation(int formationPeriod); | 语法 | List<BetterTableVO> getBetterTableVOByFormation(int formationPeriod); |
| 前置条件 | 已创建一个StrategyService对象并且实际的pickledata是非空的 |
| 后置条件 | 通过形成期获得回测的形成期与持有期的TableView |
| StrategyService. g getBetterTableVOByHolding(int holdingPeriod); | 语法 | List<BetterTableVO> getBetterTableVOByHolding(int holdingPeriod); |
| 前置条件 | 已创建一个StrategyService对象并且实际的pickledata是非空的 |
| 后置条件 | 通过持有期获得回测的形成期与持有期的TableView |
| StrategyService. getOverProfitRateByFormation(); | 语法 | List<BetterPieceVO> getOverProfitRateByFormation(); |
| 前置条件 | 已创建一个StrategyService对象并且实际的pickledata是非空的 已经调用过getBetterTableVOByFormation方法 |
| 后置条件 | 通过形成期获得超额收益率图 |
| StrategyService.getOverProfitRateByFormation() | 语法 | List<BetterPieceVO> getOverProfitRateByFormation() |
| 前置条件 | 已创建一个StrategyService对象并且实际的pickledata是非空的 已经调用过getBetterTableVOByFormation方法 |
| 后置条件 | 通过形成期获得超额收益率图 |
| StrategyService. getOverProfitRateByHolding(); | 语法 | List<BetterPieceVO> getOverProfitRateByHolding(); |
| 前置条件 | 已创建一个StrategyService对象并且实际的pickledata是非空的 已经调用过getBetterTableVOByHolding方法 |
| 后置条件 | 通过持有期获得超额收益率图 |
| StrategyService. getWinRateByHolding() | 语法 | List<BetterPieceVO> getWinRateByHolding() |
| 前置条件 | 已创建一个StrategyService对象并且实际的pickledata是非空的 已经调用过getBetterTableVOByHolding方法 |
| 后置条件 | 通过持有期获得策略胜率图 |
| StrategyService. getProfitDistributeBar(); | 语法 | List<Double> getProfitDistributeBar() |
| 前置条件 | 已创建一个StrategyService对象并且实际的pickledata是非空的 |
| 后置条件 | 获得收益分布直方图 |
| StrategyService. getProfitDistributePie() | 语法 | ProfitDistributionPieVO getProfitDistributePie() |
| 前置条件 | 已创建一个StrategyService对象并且实际的pickledata是非空的 |
| 后置条件 | 获得收益分布饼图 |
| StrategyService. getStrategyEstimateResult(); | 语法 | StrategyScoreVO getStrategyEstimateResult(); |
| 前置条件 | 已创建一个StrategyService对象并且实际的pickledata是非空的 |
| 后置条件 | 获得策略的评估结果 |
| StrategyService. getPickleData() | 语法 | List<PickleData> getPickleData() |
| 前置条件 | 已创建一个StrategyService对象并且实际的pickledata是非空的 |
| 后置条件 | 获取股票历史记录 |
|  |  |  |
| 需要的服务 | | |
| 服务名 | 服务 | |
| StockDAO. getStockInfoByCode(String code) | 返回StockPO\_列表 | |
| StockDAO. getStockInfoByCode(LocalDate date) | 传输StockPO列表 | |
| PickStockService. seprateDaysByTrade(LocalDate begin , LocalDate end , int sep); | 获取日期的划分 | |
| StrategyDAO. getStocksInPool | 获取所有符合股票池筛选条件的所有股票，构成策略的基准股票池 | |
| PickStockService .getPickleData | 计算混合策略 返回原始的pickledata 即直接可计算的pickledata | |
| PickStockService .rankAndFilterPickleData | 过滤和排序 pickledata 得到最终的要买的股票的数据 | |
| PickStockService .getNewPickleData | 获取新的pickledata | |

表pick. PickStockService的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| PickStockService.  getTradingVolume (DateRange dateRange) | 语法 | Public List<StockVolVO> getTradingVolume (DateRange dateRange) |
| 前置条件 | 已创建一个PickStockService对象 |
| 后置条件 | 按照正常方式 而不是交易日分割天数 |
| PickStockService. seprateDaysByTrade(LocalDate begin , LocalDate end , int sep); | 语法 | List<PickleData> seprateDaysByTrade(LocalDate begin , LocalDate end , int sep); |
| 前置条件 | 已创建一个PickStockService对象 |
| 后置条件 | 按照易日分割天数 |
| PickStockService. getSingleCodeMAInfo(String code , LocalDate begin , LocalDate end , int days); | 语法 | List<DayMA> getSingleCodeMAInfo(String code , LocalDate begin , LocalDate end , int days); |
| 前置条件 | 已创建一个PickStockService对象 |
| 后置条件 | 获得特定天内的 均线信息 |
| PickStockService. getSingleCodeMOMInfo(String code, LocalDate begin, LocalDate end, int formationPeriod); | 语法 | List<FormationMOM> getSingleCodeMOMInfo(String code, LocalDate begin, LocalDate end, int formationPeriod); |
| 前置条件 | 已创建一个PickStockService对象 |
| 后置条件 | 获得特定天内的股票的收益率信息 |
| PickStockService. getLastVol(String code , LocalDate begin ,LocalDate end); nge) | 语法 | List<LongPeiceVO> getLastVol(String code , LocalDate begin ,LocalDate end); |
| 前置条件 | 已创建一个PickStockService对象 |
| 后置条件 | 获得上一日的交易量 |
| PickStockService.getLastUpRange(String code, LocalDate begin ,LocalDate end); | 语法 | List<UpRangeVO> getLastUpRange(String code, LocalDate begin ,LocalDate end); |
| 前置条件 | 已创建一个PickStockService对象 |
| 后置条件 | 获得上一日的涨幅 |
| PickStockService. getAdj(String code, LocalDate begin ,LocalDate end); | 语法 | List<AdjVO> getAdj(String code, LocalDate begin ,LocalDate end); |
| 前置条件 | 已创建一个PickStockService对象 |
| 后置条件 | 获得上一日的复权平均价 |
| PickStockService. getCirculationMarketValue(String code, LocalDate begin ,LocalDate end); | 语法 | List<CirculationMarketValueVO> getCirculationMarketValue(String code, LocalDate begin ,LocalDate end); |
| 前置条件 | 已创建一个PickStockService对象 |
| 后置条件 | 获得流通市值 |
| PickStockService. getAmplitude(String code, LocalDate begin ,LocalDate end); | 语法 | List<AmplitudeVO> getAmplitude(String code, LocalDate begin ,LocalDate end); |
| 前置条件 | 已创建一个PickStockService对象 |
| 后置条件 | 获得昨日的振幅 |
| 需要的服务 | | |
| 服务名 | 服务 | |
| StockDAO. getStockInfoByCode(String code) | 返回StockPO\_列表 | |
| StockDAO. getStockInfoByCode(LocalDate date) | 传输StockPO列表 | |

迭代二数据层接口设计规范

表strategy. StrategyDAO的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| StrategyDAO.  getStocksInPool(StockPoolConditionPO stockPoolConditionPO); | 语法 | List<String> getStocksInPool(StockPoolConditionPO stockPoolConditionPO); |
| 前置条件 | 已创建一个StrategyDAO对象 |
| 后置条件 | 获取所有符合股票池筛选条件的所有股票，构成策略的基准股票池 |
| StrategyDAO. getPickleData(LocalDate beginDate , LocalDate endDate , StockPoolConditionVO stockPoolConditionVO, List<StockPickIndexVO> stockPickIndexVOs, TraceBackVO traceBackVO, List<MixedStrategyVO> mixedStrategyVOS); | 语法 | List<PickleData> getPickleData(LocalDate beginDate , LocalDate endDate , StockPoolConditionVO stockPoolConditionVO, List<StockPickIndexVO> stockPickIndexVOs, TraceBackVO traceBackVO, List<MixedStrategyVO> mixedStrategyVOS); |
| 前置条件 | 已创建一个StrategyDAO对象 |
| 后置条件 | 计算混合策略 返回原始的pickledata 即直接可计算的pickledata |
| StrategyDAO. rankAndFilterPickleData(List<PickleData> pickleDataList,  List<StockPickIndexVO> stockPickIndexVOs , StrategyType strategyType ,  int holdingNum , double holdingRate, boolean asd); | 语法 | List<PickleData> rankAndFilterPickleData(List<PickleData> pickleDataList,  List<StockPickIndexVO> stockPickIndexVOs , StrategyType strategyType ,  int holdingNum , double holdingRate, boolean asd); |
| 前置条件 | 已创建一个StrategyDAO对象 |
| 后置条件 | 过滤和排序 pickledata 得到最终的要买的股票的数据 |
| StrategyDAO. getNewPickleData(StrategyConditionVO strategyConditionVO, StockPoolConditionVO stockPoolConditionVO, List<StockPickIndexVO> stockPickIndexVOs); | 语法 | List<NewPickleData> getNewPickleData(StrategyConditionVO strategyConditionVO, StockPoolConditionVO stockPoolConditionVO, List<StockPickIndexVO> stockPickIndexVOs); |
| 前置条件 | 已创建一个StrategyDAO对象 |
| 后置条件 | 获取新的pickledata |

# 9.数据模型

**9.1 vo对象规格说明**

DailyClosingPriceVO当日收盘价值对象

BasisAnalysisVO 股票对比信息值对象

DailyLogReturnVO 股票对数值对象

KLinePieceVO K线数值

KLineExtraVO K线额外值

StockVolVO 成交量值对象

VolExtraVO 成交量额外值对象

MAPieceVO 均线数值对象

**9.2 po对象规格说明**

|  |  |  |
| --- | --- | --- |
| **对象名** | **对外接口** | **接口描述** |
| StockPO | Public StockPO() | 初始化数据 |
| public T getxxx()  public void setxxx() | 存取获得数据的属性 |
| StockNameAndCodePo | public StockNameAndCodePo() | 初始化数据 |
| public T getxxx()  public void setxxx() | 存取获得数据的属性 |

StockPO属性:

