白盒测试

—SE2018—G04简易查



**小组：G04**

**骆佳俊 徐双铅 吕迪**

**修订表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **生成版本** | **修订人** | **修订章节与内容** | **修订日期** |
| 1 | 0.5 | 徐双铅 | 编写内容 | 2018.6.2 |
| 2 | 0.6 | 徐双铅 | 参数预警pad修改 | 2018.6.3 |
| 3 | 0.7 | 徐双铅 | 添加匹配模块pad | 2018.6.4 |
| 4 | 1 | 骆佳俊 | 白盒测试结果 | 2018.06.25 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**目录**

[1. 引言 4](#_Toc26423)

[1.1 编写目的 4](#_Toc11078)

[1.2 背景 4](#_Toc22814)

[1.3 定义 4](#_Toc4344)

[1.4 参考资料 4](#_Toc22265)

[2. 被测试模块列表 5](#_Toc26428)

[3. 模块逻辑结构 5](#_Toc20176)

[3.1输入模块 5](#_Toc20843)

[3.1.1自选股添加输入 5](#_Toc22086)

[3.1.2参数预警输入 5](#_Toc14240)

[3.1.3 模块功能定义 7](#_Toc16900)

[3.2输出模块 7](#_Toc23793)

[3.2.1股票信息输出 7](#_Toc19855)

[3.2.2 模块功能定义 7](#_Toc10351)

[3.3匹配模块 8](#_Toc32467)

[3.2.2 模块功能定义 8](#_Toc16141)

[4. 测试数据 8](#_Toc2781)

[4.1输入模块 8](#_Toc15843)

[4.1.1自选股添加输入 8](#_Toc28064)

[4.1.2参数预警输入 10](#_Toc8041)

[4.2输出模块 12](#_Toc13504)

[4.2.1股票信息输出 12](#_Toc27544)

[4.3匹配模块 15](#_Toc15785)

# 引言

伴随着资金的增长和对富有的向往，越来越多的人开始加入炒股的行列。股市的行情是瞬息多变的，掌握的实时的资料是十分必要的，我们做一个可以查看股价，并提供监控提醒。方便股民在工作空余时间实时查看股价，在股民事务繁忙情况下提供监控实时提醒。股民根据信息作出判断。

## 编写目的

设计测试方案是测试阶段的关键技术问题。白盒测试技术中包含逻辑覆盖和控制结构测试这两种设计测试数据的典型技术。白盒测试用例报告是使用白盒测试技术来设计测试数据和预计的输出结构形成测试用例，在项目完成编码以后使用测试用例来发现软件中的错误。

## 背景

1. 待开发软件的名称为“简易查”
2. 本项目提出者和开发者均为SE2018春-G04小组（组长骆佳俊，组员吕迪，徐双铅），目标人群设定为炒股初入门者还有特殊用户杨老师。
3. 相互来往关系：本软件通过调用阿里云的实时股票API和历史股票API，已K线图形式展现给使用者，并筛选出必要数据进行比较得出结果。

## 1.3 定义

API：Application Programming Interface,应用程序编程接口

## 1.4 参考资料

1. 张海藩,牟永敏.软件工程导论（第六版）[M].北京：清华大学出版社, 2013:162-171
2. 白盒测试用例模板 ：百度文库 （2018.6.2）

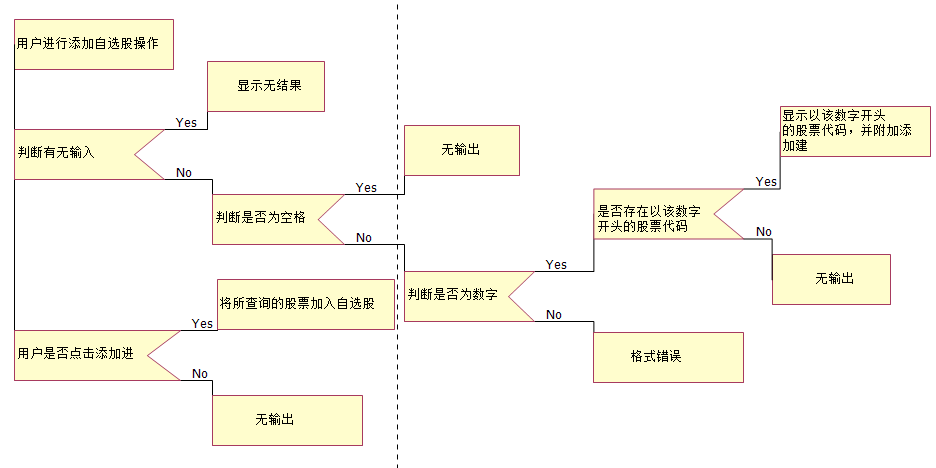
# 被测试模块列表

|  |  |  |
| --- | --- | --- |
| 编号 | 模块名称 | 优先程度 |
| F1 | 输入模块 | 1 |
| F2 | 输出模块 | 2 |

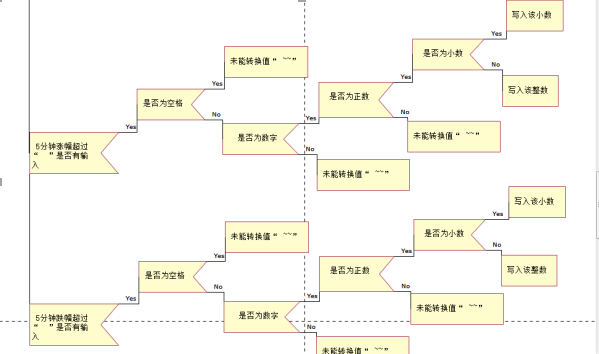
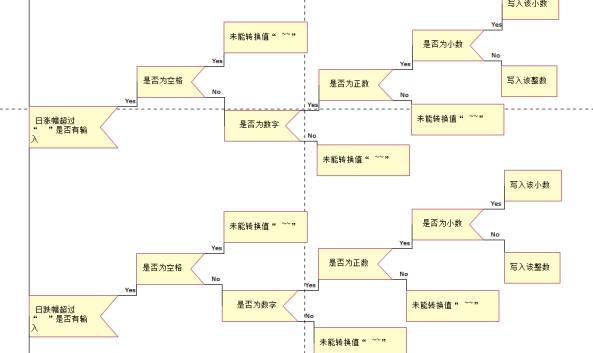
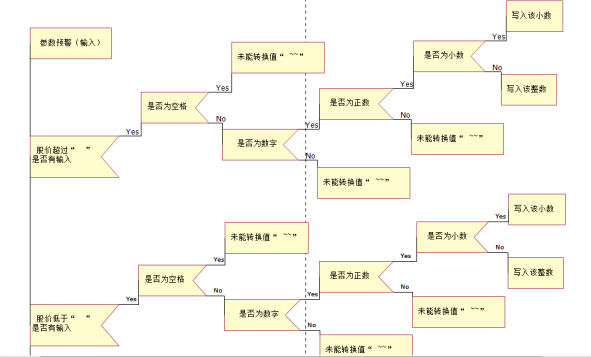
# 模块逻辑结构

## 3.1输入模块

### 3.1.1自选股添加输入



### 3.1.2参数预警输入

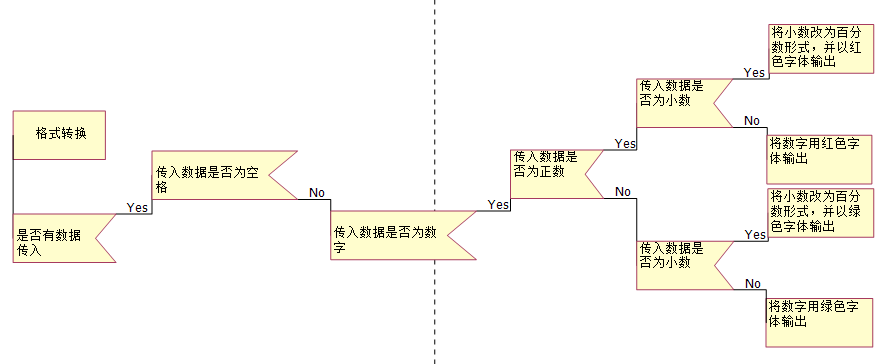


### 3.1.3 模块功能定义

用户进行键盘输入，输入股票代码以查询该股，输入预警所需的参数来进行预警。

## 3.2输出模块

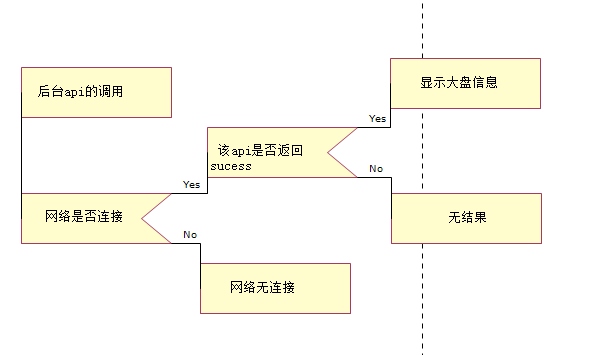
### 3.2.1股票信息输出



### 3.2.2 模块功能定义

把将要进行的数据进行格式变化，满足股票的输出标准。如小数化为百分数，上涨字体用红色字体，下跌用绿色字体

## 3.3匹配模块



### 3.2.2 模块功能定义

后台调用api，用于判断是否成功调用了api，是否成功获取了。若遇到无网络，则无法调用api。又或者对方服务器故障，无法传递股票数据。

# 测试数据

## 4.1输入模块

### 4.1.1自选股添加输入

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编码 | 语句 | 输入 | 输出 | 测试类型 |
| 1 | If（读入的是否为空或者为空格）then  If（输入的是否为数字类型）then  If（用该股票代码api调用返回是否成功）then  If（用户点击添加键）then  …end if…end if…end if | （无输入） | 无结果 | 条件覆盖 |
| 2 | If（读入的是否为空或者为空格）then  If（输入的是否为数字类型）then  If（用该股票代码api调用返回是否成功）then  If（用户点击添加键）then  …end if…end if…end if | （输入空格） | 无结果 | 条件覆盖 |
| 3 | If（读入的是否为空或者为空格）then  If（输入的是否为数字类型）then  If（用该股票代码api调用返回是否成功）then  If（用户点击添加键）then  …end if…end if…end if | xxzzdd | 无结果 | 条件覆盖 |
| 4 | If（读入的是否为空或者为空格）then  If（输入的是否为数字类型）then  If（用该股票代码api调用返回是否成功）then  If（用户点击添加键）then  …end if…end if…end if | 12121211111（输入不存在的股票代码） | 无结果 | 条件覆盖 |
| 5 | If（读入的是否为空或者为空格）then  If（输入的是否为数字类型）then  If（用该股票代码api调用返回是否成功）then  If（用户点击添加键）then  …end if…end if…end if | 60131（股票代码）不点添加键 | 601311骆驼股份  601313江南嘉捷  601318 中国平安 | 条件覆盖 |
| 6 | If（读入的是否为空或者为空格）then  If（输入的是否为数字类型）then  If（用该股票代码api调用返回是否成功）then  If（用户点击添加键）then  …end if…end if…end if | 60131（股票代码）点击中国平安 | 中国平安信息添加到界面 | 条件覆盖 |

### 4.1.2参数预警输入

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编码 | 语句 | 输入 | 输出 | 测试类型 |
| 1 | If（读入的是否为空或者为空格）then  If（输入的是否为数字类型）then  If（符合逻辑）then  …end if…end if…end if | （无输入） | 无结果 | 条件覆盖 |
| 2 | If（读入的是否为空或者为空格）then  If（输入的是否为数字类型）then  If（符合逻辑）then  …end if…end if…end if | （输入空格） | 未能转换值 | 条件覆盖 |
| 3 | If（读入的是否为空或者为空格）then  If（输入的是否为数字类型）then  If（符合逻辑）then  …end if…end if…end if | Xxzzddaa333 | 未能转换值“Xxzzddaa333” | 条件覆盖 |
| 4 | If（读入的是否为空或者为空格）then  If（输入的是否为数字类型）then  If（符合逻辑）then  …end if…end if…end if | 3（输入的值>0) | 3 | 条件覆盖 |
| 5 | If（读入的是否为空或者为空格）then  If（输入的是否为数字类型）then  If（符合逻辑）then  …end if…end if…end if | 3.1111 | 3.1111 | 条件覆盖 |
| 6 | If（读入的是否为空或者为空格）then  If（输入的是否为数字类型）then  If（符合逻辑）then  …end if…end if…end if | -1(目标为负数） | 未能转换值”-1” | 条件覆盖 |
| 7 | If（读入的是否为空或者为空格）then  If（输入的是否为数字类型）then  If（符合逻辑）then  …end if…end if…end if | aaaa(目标为字母） | 未能转换值”aaaa” | 条件覆盖 |

## 4.2输出模块

### 4.2.1股票信息输出

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编码 | 语句 | 输入 | 输出 | 测试类型 |
| 1 | If（读入的是否为空或者为空格）then  If（输入的是否为数字类型）then  If（目标类型为Brush）then  If（目标类型为String）then  …end if…end if…end if | （无输入） | 无结果 | 条件覆盖 |
| 2 | If（读入的是否为空或者为空格）then  If（输入的是否为数字类型）then  If（目标类型为Brush）then  If（目标类型为String）then  …end if…end if…end if | （输入空格） | 无结果 | 条件覆盖 |
| 3 | If（读入的是否为空或者为空格）then  If（输入的是否为数字类型）then  If（目标类型为Brush）then  If（目标类型为String）then  …end if…end if…end if | xxzzdd | 无结果 | 条件覆盖 |
| 4 | If（读入的是否为空或者为空格）then  If（输入的是否为数字类型）then  If（目标类型为Brush）then  If（目标类型为String）then  …end if…end if…end if | 3（输入的值>0) | 颜色转为红色 | 条件覆盖 |
| 5 | If（读入的是否为空或者为空格）then  If（输入的是否为数字类型）then  If（目标类型为Brush）then  If（目标类型为String）then  …end if…end if…end if | -1(输入的值<0） | 颜色转为绿色 | 条件覆盖 |
| 6 | If（读入的是否为空或者为空格）then  If（输入的是否为数字类型）then  If（目标类型为Brush）then  If（目标类型为String）then  …end if…end if…end if | 0.001(目标输入为小数） | 0.1% | 条件覆盖 |
| namespace SimpleTracer.Converters  {  /// <summary>  /// 股票涨跌值的XAML值转换器  /// </summary>  class StockChangeConverter : IValueConverter  {  public object Convert(object value, Type targetType, object parameter, CultureInfo culture)  {  if (IsNumeric(value))  {  var v = Val(value);  //转换涨跌值到颜色  if (targetType == typeof(Brush) || targetType.IsSubclassOf(typeof(Brush)))  {  return v > 0 ? Brushes.Red  : v < 0 ? Brushes.LimeGreen  : Brushes.Black;  }  //格式化涨跌值  else if (targetType == typeof(string))  {  return (v > 0 ? ("+" + value) : value ) +  (parameter?.ToString() == "%" ? "%" : "");  }  }  throw new InvalidCastException($"无法执行从 '{value.GetType().FullName}' 到 '{targetType.FullName}' 的转换。");  }  public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture)  {  throw new NotImplementedException();  }  }  } | | | | |

## 4.3匹配模块

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编码 | 语句 | 输入 | 输出 | 测试类型 |
| 1 | If(网络是否连接)then  If（api调用是否成功）then  end if……end if…… | （不联网API输入正确） | 网络未连接 | 条件覆盖 |
| 2 | If(网络是否连接)then  If（api调用是否成功）then  end if……end if…… | （不联网输入错误API） | 网络未连接 | 条件覆盖 |
| 3 | If(网络是否连接)then  If（api调用是否成功）then  end if……end if…… | （联网API输入错误） | 无结果 | 条件覆盖 |
| 4 | If(网络是否连接)then  If（api调用是否成功）then  end if……end if…… | （联网API输入正确） | 该大盘实时信息 | 条件覆盖 |

