

Java编程技能训练课程项目

项目名称：知乎网站数据抓取与分析

组名： 第10小组

组长 ： 吴林锴

组员 : 周天亮

高煜东

狄烨

王勤晓

目录

[一 概述 4](#_Toc408004266)

[1.1系统概述及目标 4](#_Toc408004267)

[1.2系统整体设计 4](#_Toc408004268)

[二 系统设计 5](#_Toc408004269)

[三 程序功能设计 6](#_Toc408004270)

[3.1功能列表 6](#_Toc408004271)

[3.2主要类的设计 7](#_Toc408004272)

[3.2.1 Main类 7](#_Toc408004273)

[3.2.2 Zhihu类 7](#_Toc408004274)

[3.2.3 Spider类 8](#_Toc408004275)

[3.2.4 LoadData类 8](#_Toc408004276)

[3.2.5 FileReaderWriter类 8](#_Toc408004277)

[3.2.6 MyJDBC类 9](#_Toc408004278)

[3.2.7 QuestionTable类 9](#_Toc408004279)

[3.2.8 TextComponentFrame类 10](#_Toc408004280)

[3.2.9 TextComponentTest类 10](#_Toc408004281)

[3.2.10 DataAnalysis类 10](#_Toc408004282)

[3.3分析 11](#_Toc408004283)

[3.3.1 Main类及main函数概述及详细设计 11](#_Toc408004284)

[3.3.2 知乎话题爬取部分概述及Spider类详细设计 11](#_Toc408004285)

[3.3.3相应话题具体问题爬取部分概述及Zhihu类详细设计 13](#_Toc408004286)

[3.3.4爬虫得到的数据写入文件部分概述及FileReaderWriter类简介 14](#_Toc408004287)

[3.3.5 通过MyJDBC和LoadData将相应的数据写入数据库 16](#_Toc408004288)

[3.3.6 preparedStatement和Statement 有什么不一样 17](#_Toc408004289)

[3.3.7分析部分界面概述及TextComponentTest和TextComponentFrame设计概述 18](#_Toc408004290)

[3.3.8具体的分析部分概述及DataAnalysis等类的设计实现 20](#_Toc408004291)

[四 数据库设计 26](#_Toc408004292)

[4.1 Author表 26](#_Toc408004293)

[4.2 Financial 表 26](#_Toc408004294)

[4.3 Learn表 26](#_Toc408004295)

[4.4 Movie表 26](#_Toc408004296)

[4.5 Program表 27](#_Toc408004297)

[4.6 Question表 27](#_Toc408004298)

[4.7 Soccer表 27](#_Toc408004299)

[五 程序运行结果 28](#_Toc408004300)

[5.1 word文档展示 28](#_Toc408004301)

[5.2数据库展示 28](#_Toc408004302)

[5.3查询界面展示 30](#_Toc408004303)

[六 成员分工 34](#_Toc408004304)

[七 Github地址 34](#_Toc408004305)

# 一 概述

**1.1系统概述及目标**

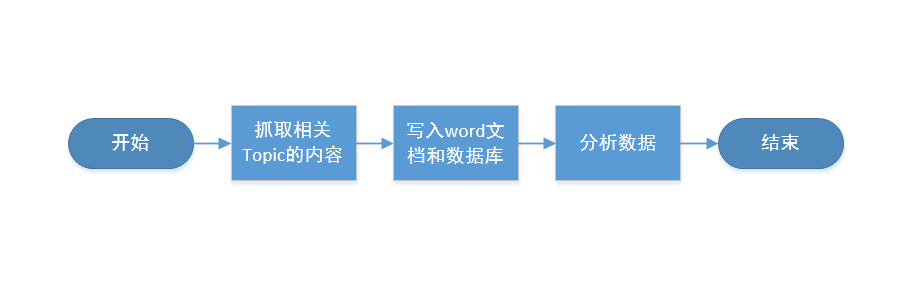
知乎近几年发展迅速，用户数量巨大，各类相关话题讨论传播速度快，范围广，讨论价值大。针对知乎各个类别热点话题的探讨和互相问答的学习形式也成为当下互联网时代的一大热点，同时由于知乎上问答水平具有质量高、话题广，参与度高的特点，所以我们系统爬取了知乎上比较有代表性的5大类话题，来分析其相关的一些属性，并作出成果展示。这具有很重要的理论意义和实用价值。

主要功能： 1、爬取知乎5大类话题精华（例如：学习、电影、足球、金融、编程）

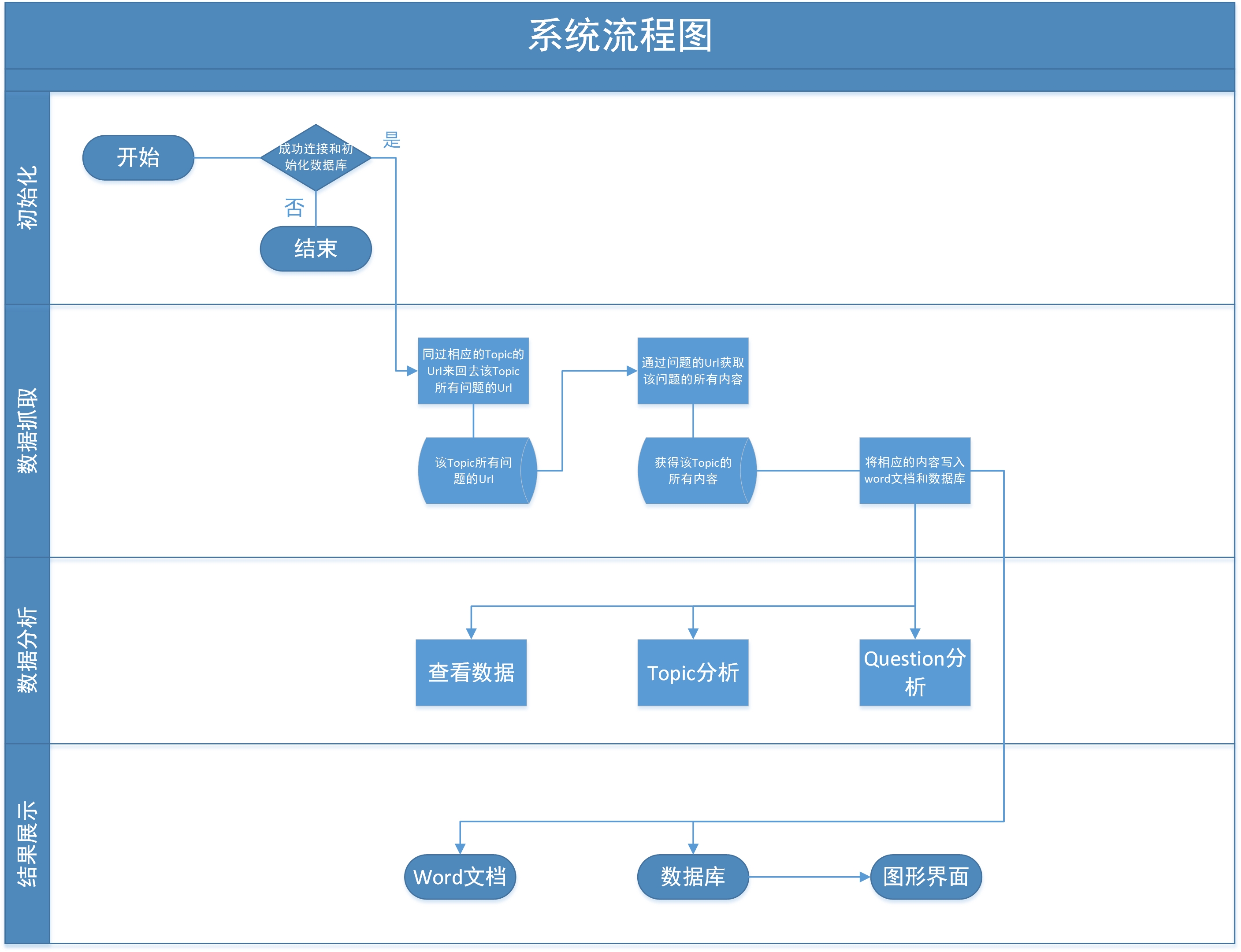
2、获得该话题精华下的具体问题信息、回答内容、及作者信息

3、微博的数据分析（传播速度、传播范围、意见领袖等）

## 1.2系统整体设计



**二 系统设计**



**三 程序功能设计**

**3.1功能列表**



## 3.2主要类的设计

### 3.2.1 Main类

|  |
| --- |
| **Main** |
| +urlTopic : String  +urlMovie : String  +urlLearn : String  +urlFinancial : String  +urlProgram : String  +urlSoccer : String  +urlRecommendations : String |
| +Main( ) : void |

### 3.2.2 Zhihu类

|  |
| --- |
| **Zhihu** |
| +questionID : String  +questionTitle : String  +questionDescription : String  +ZhihuUrl : String  +focusCount : String  +topic : String  +suthorAnswer : JSONObject |
| +Zhihu(String url, String topic ) : void  +getRealUrl(String url) : boolean  +getAllContents(String url) : void  +writeString( ) : String |

### 3.2.3 Spider类

|  |
| --- |
| **Spider** |
| +topic : String  +urlString : String  +content : String  +myZhihu : ArrayList<Zhihu> |
| +Spider(String topic, String url) : void  +run( ) : void  +SendGet(String url) : String  +GetTopics(String content, String topic) : ArrayList<Zhihu> |

### 3.2.4 LoadData类

|  |
| --- |
| **LoadData** |
| +myJDBC : MyJDBC  +myZhihu : ArrayList<Zhihu> |
| + LoadData(ArrayList<Zhihu> myZhihu) : void  + loadAuthors( ) : void  +loadQuestion( ) : void  +LoadTopic( ) : void  +getCounts(String answerContent): String |

### 3.2.5 FileReaderWriter类

|  |
| --- |
| **FileReaderWriter** |
|  |
| +createNewFile(String filePath) : boolean  +writeIntoFile(String content, String filePath, boolean isAppend) : boolean |

### 3.2.6 MyJDBC类

|  |
| --- |
| **MyJDBC** |
| -url : String  -username : String  -password : String  -conn : Connection  -st : Statement  -pStatement : PreparedStatement |
| + MyJDBC( ) : void  +SaveQuestions(String questionID, String questionTitle, String zhihuUrl, String focusCount, String topic, String answerCount) : void  +SaveTopic(String Topic, String questionID, String answercount) : boolean  +SaveAuthors(String authorID, String authorname, String votecount, String answercount, String questionID) : boolean  +getConnection( ) : Connection  +clearDataBase( ) : void |

### 3.2.7 QuestionTable类

|  |
| --- |
| **QuestionTable** |
| +JTableView : JTable  +scrollPane : JscrollPane |
| +QuestionTable ( ) : void  +QuestionTable(String topic) : void |

### 3.2.8 TextComponentFrame类

|  |
| --- |
| **TextComponentFrame** |
| +TEXTAREA\_ROWS : int  +TEXTAREA\_COLUMNS : int |
| +TextComponentFrame( ) : void |

### 3.2.9 TextComponentTest类

|  |
| --- |
| **TextComponentTest** |
|  |
| +main(String[] args) : void |

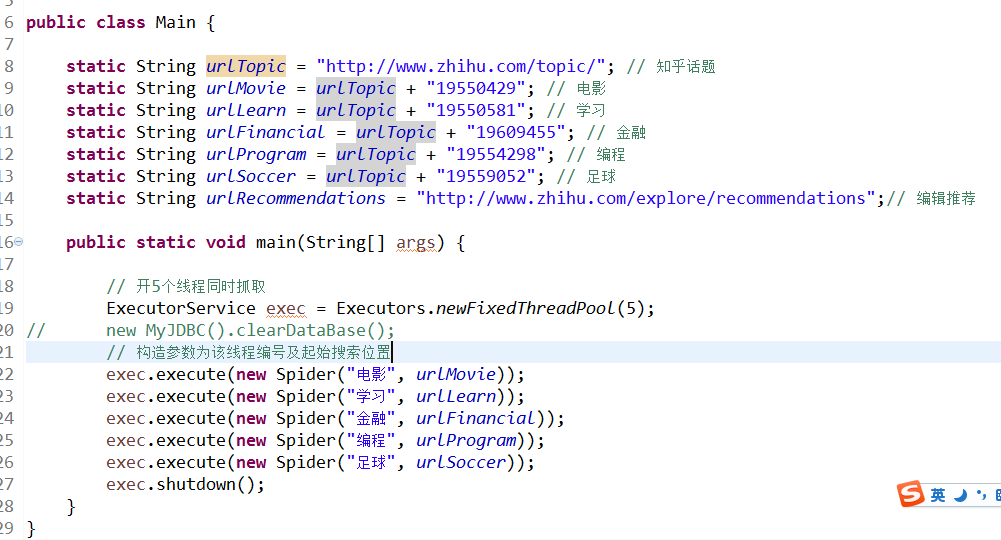
### 3.2.10 DataAnalysis类

|  |
| --- |
| **DataAnalysis** |
| -url : String  -username : String  -password: String  +pStatement: PreparedStatement |
| +getConnection( ) : Connection  +DataAnalysis( ) : void  +searchQueationByID(String questionID) : String  +searchTopic(String topic) : JTable  +getQuestion( ) : JTable  +getResult(String topic) : double[]  +main(String[] args) : void |

## 3.3分析

### 3.3.1 Main类及main函数概述及详细设计

首先我们的项目是一个知乎爬虫工具，我们首先在创建考虑抓取“电影”、“学习”、“编程”等5个主要话题的相应问题，并且我们希望能过同时的进行这几项任务，所以使用了多线程的方法爬取这几个话题，每个线程负责爬取其中某个话题，具体的线程的实现可以参见下面的spider类的run( )方法。Main类的具体情况也可以参见上面的类图部分，下面是Main类的具体的设计实现。



通过上述具体设计的实现可以看出我们爬取数据主要的初始输入就是相应话题的url，针对每个话题都要起一个线程来爬取，这样可以有较高的效率。

### 3.3.2 知乎话题爬取部分概述及Spider类详细设计

我们项目需要根据相应的话题来分析，那么我们首先调用spider类里的SendGet( )方法,该方法的具体实现如下：



这个方法主要是通过get方法像知乎服务器发送报文，并请求服务器将相应url的html主体返回给本地，然后该方法得到的html代码存为一个字符串string类型result，然后将result作为返回值返回。

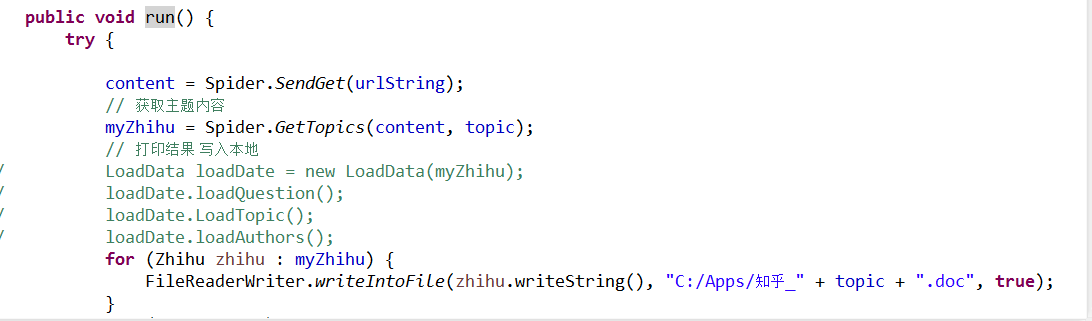
由于我们要获得相应话题的具体的问题，例如“电影”这个话题中可能有几十个具体的问题，我们通过上面提到的SendGet( )方法得到的html，我们该html中的网页链接通过正则表达式的方法匹配出相应的问题的网页url。具体的实现方法在GetTopic( )方法如下:



相应的正则表达式匹配的html标签例如：



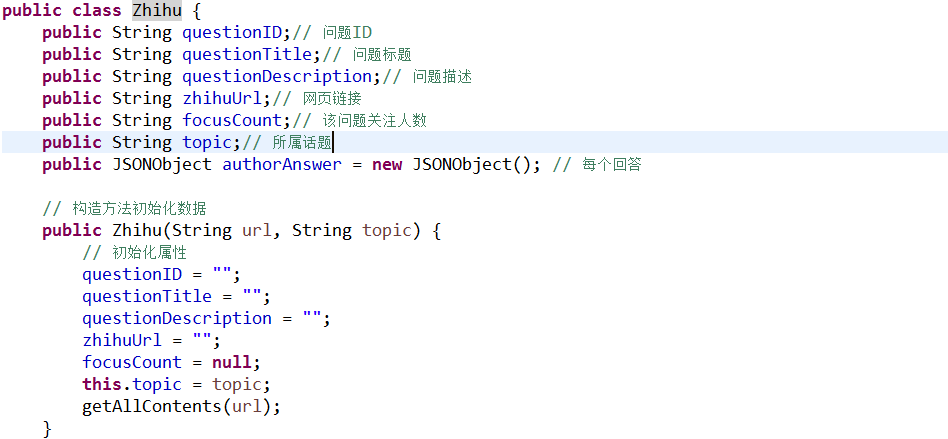
总体而言这个类就的功能就是获得具体的某一个话题例如“电影”话题网页的html，并通过正则匹配获得这个话题的所有问题的url，并为下一步问题信息的提取做准备，该模块的同时是Runnable的一个实现，该类具体的run()部分实现如下：



该方法同时还调用了zhihu类的相应方法，zhihu部分具体如下所示。

### 3.3.3相应话题具体问题爬取部分概述及Zhihu类详细设计

上面两部分提到，我们要爬取具体某一个话题下面的问题，并提取相应问题的信息，其中关于具体问题的类的设计与实现在zhihu类里实现。Zhihu类的数据成员定义及其构造函数的实现如下：

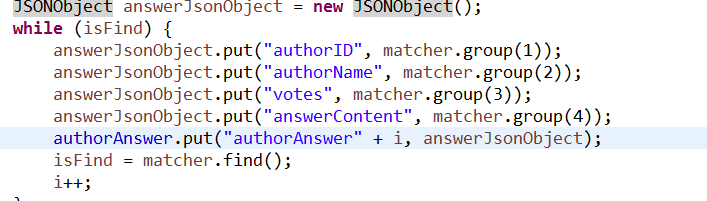


通过上述类的成员变量可以看出，zhihu类是一个具体的知乎问题的类，比如“最近什么电影比较好看”就是一个具体的问题。zhihu类的构造函数主要需要传入的参数有该问题的具体url和该问题的分类归属，比如刚才提到的那个问题的归属就属于“电影”话题。

我们可以看出刚才那个构造函数的最后一部分是调用了getAllContents( )方法。该方法关于正则表达式的匹配部分如下：



可以看出根据相应的正则表达式匹配，该方法分别得到了zhihu这个类问题Id、问题标题，问题描述，问题的关注人数，并通过第三方包中的类JSONObject来存储这些信息，过程如下：

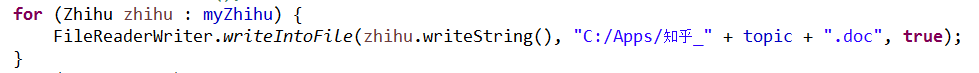


每个话题的每个回答都是用一个JSONObject对象来存储，与此同时这些回答整体又构成了一个JSONObject对象，并且这个对象是属于一个zhihu对象的成员变量。然后这些爬虫的得到的结果将会被写入相应的文件，并被写入到数据库之中，关于写入文件的部分请参见下一部分。

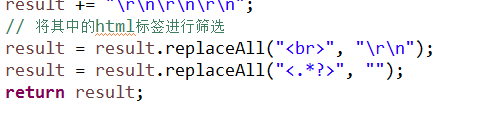
### 3.3.4爬虫得到的数据写入文件部分概述及FileReaderWriter类简介

我们前面几部分提到了爬虫的过程，通过首先爬取相应话题的html，从中提取出相应的问题列表和相应问题的url，在访问相应问题的url得到相应问题的html，从中提取出相应的问题的回答及想关信息，我们最后需要将这些信息写入文件及数据库。写入数据库的相关部分再下一章节数据库设计部分详细描述，关于文件的写入是通过FileReaderWriter这个类来实现的。

首先要创建相应的文件，例如C：/apps/知乎，这一部分是写在spider类中，具体如下：



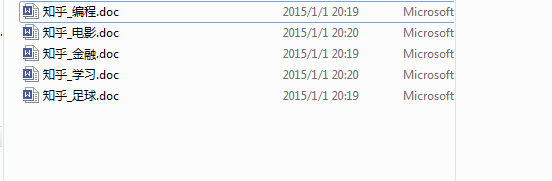
这部分文件写入就是常规java的文件和数据流的操作，比较需要的注意的部分是，上述几个过程中我们爬取得到的数据存在很多html标签那些看起来并不美观，我们通过如下代码：



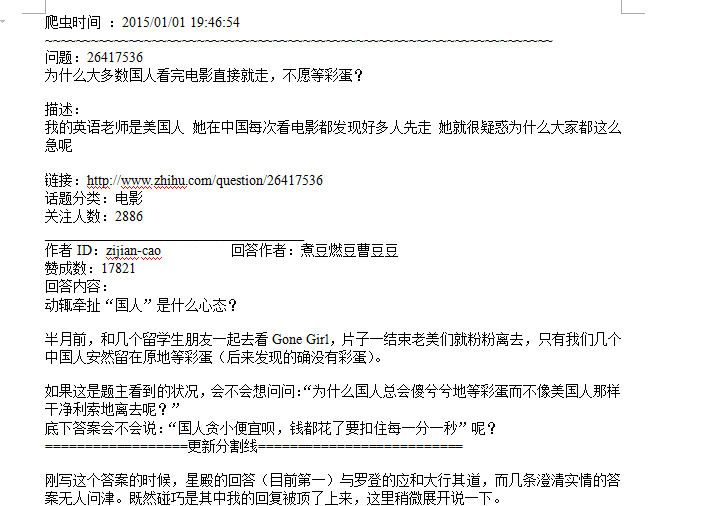
这个代码可以将得到的结果中的html标签替换，例如<br>替换成换行符”\r\n”等。这样就可以使输出的结果更加美观。

最后爬虫得到的结果如下：

首先是几个相应话题的文件，存储格式是doc

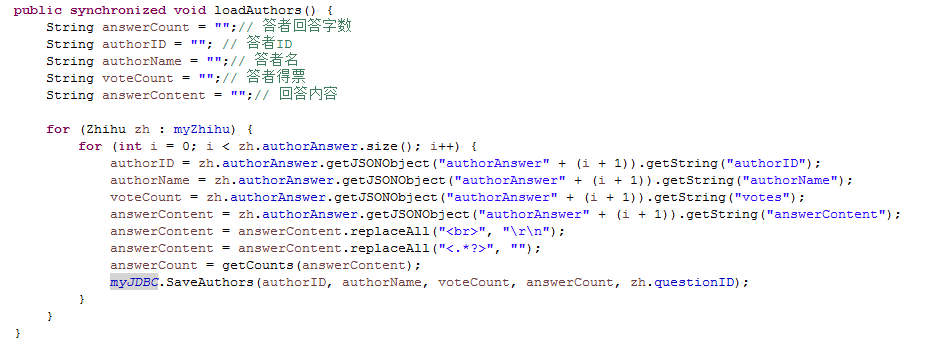


然后以”知乎\_电影“为例，可以看出其中内容格式如下：

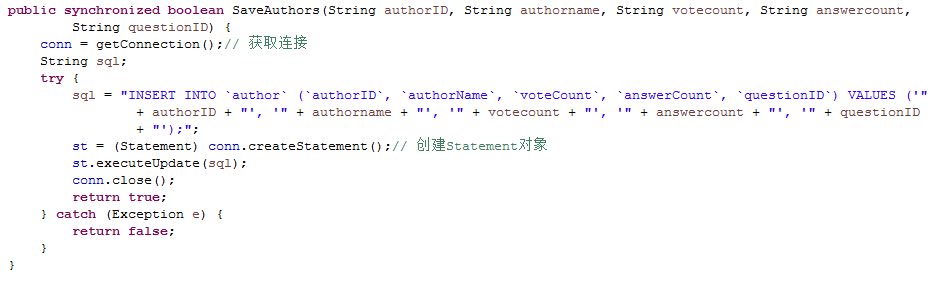


可以看出其中内容详尽，非常有利于分析及使用。

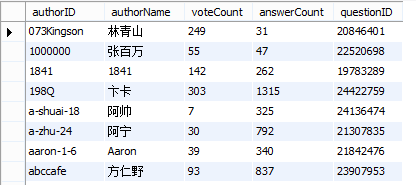
### 3.3.5 通过MyJDBC和LoadData将相应的数据写入数据库



图：LoadData的方法



图：MyJDBC的方法



图：数据库截图

### 3.3.6 preparedStatement和Statement 有什么不一样

一.代码的可读性和可维护性

PreparedStatement尽最大可能提高性能

三．最重要的一点是极大地提高了安全性



图1

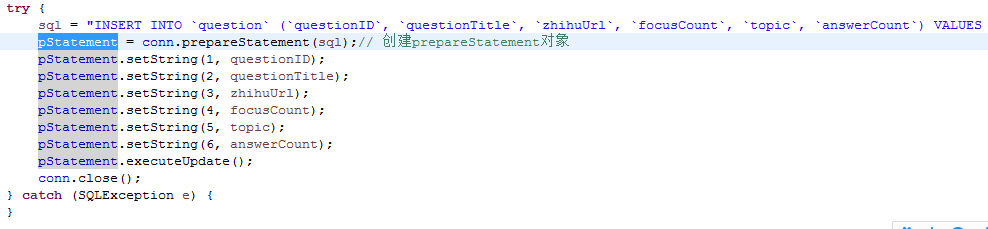


图2：PreparedStatement

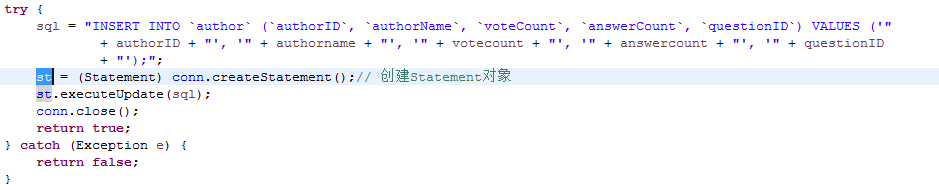
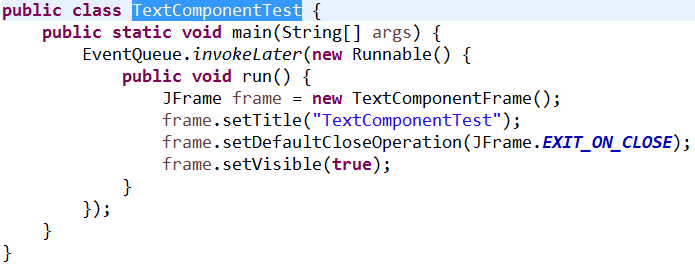


图3：Statement

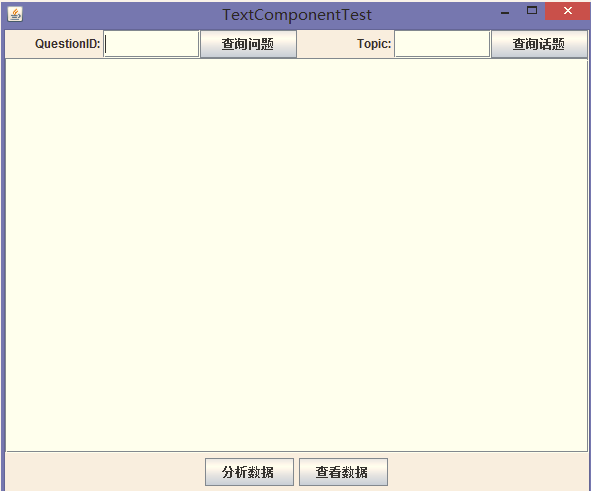
### 3.3.7分析部分界面概述及TextComponentTest和TextComponentFrame设计概述

首先我们的项目分析的展示界面是使用java标准库的awt和swing来实现的，主要界面的绘画的类是TextComponentTest和TextComponentFrame类中实现的。

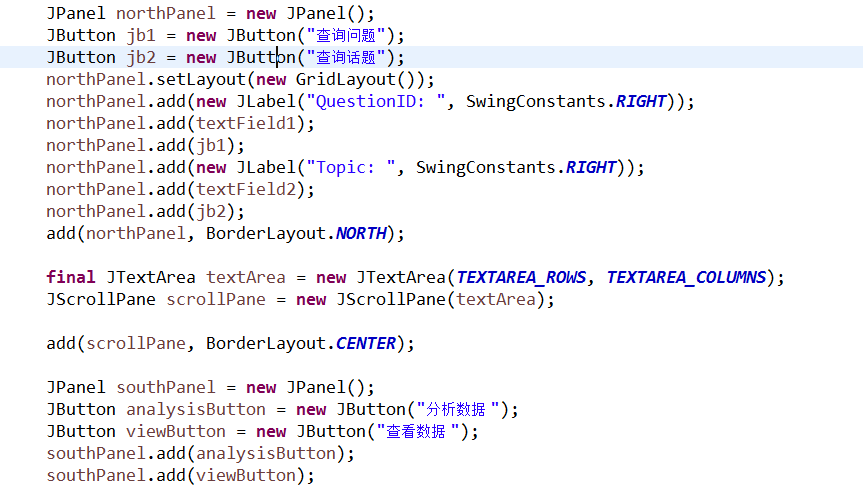
TextComponentTest实现如下：



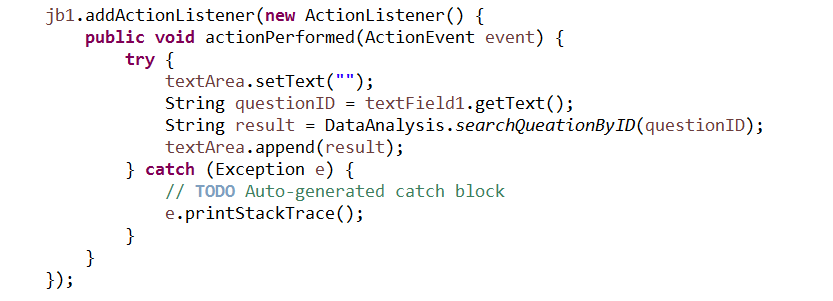
TextComponentFrame类是主要的界面设计类，我们主要的界面如下：



TextComponentFrame类中设定四个按钮一个文本框具体如下:



例如这个jd1这个按钮的监听器绑定如下：



可以看出jd1这个按钮点击时会触发DataAnalysis里面的searchQueationById方法，具体关于DataAnalysis的设计与实现在相应部分详细概述。

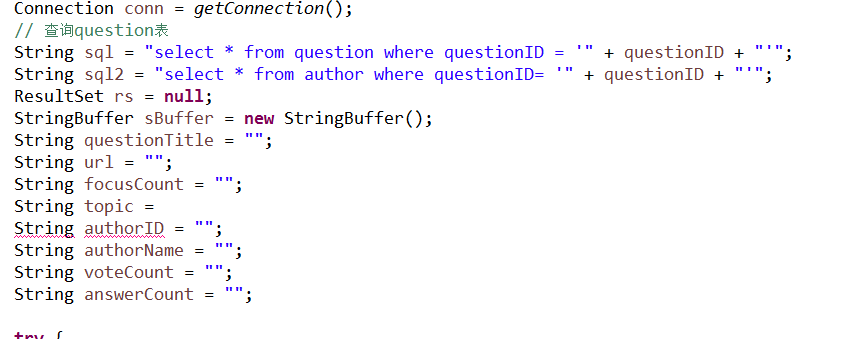
### 3.3.8具体的分析部分概述及DataAnalysis等类的设计实现

上面我们提到我们的分析程序的界面有查询问题、查询话题、分析数据、查看数据等几个按钮，我们点击相应按钮时会被相应的监听器检测到并触发相应的处理函数，具体的处理函数定义在DataAnalysis这个类中。

DataAnalysis当然需要连接数据库，相应部分的代码如下：



其中查询问题按钮的处理函数是在searchQueationByID这个方法中实现的，其代码如下：



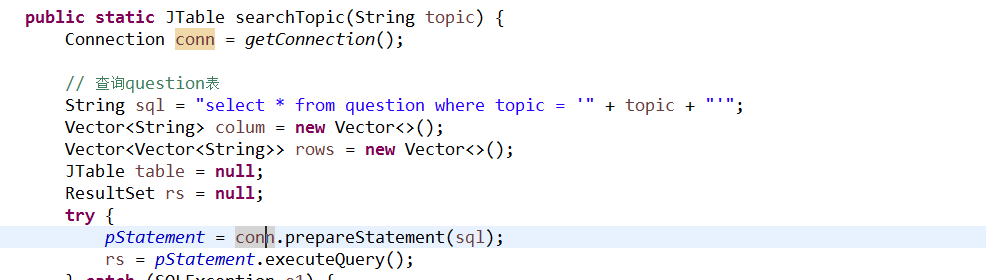
我们可以从中看出该方法从数据库中question表和author表的questionId，并通过它得到1、字符串：questionTitle、topic、url 、关注人数 2、作者名字authorName，点赞数voteCount，字数answerCount等信息，并将它们这些信息存储在sBuffer中并返回，以用于界面textarea中展示，其代码设计如下：



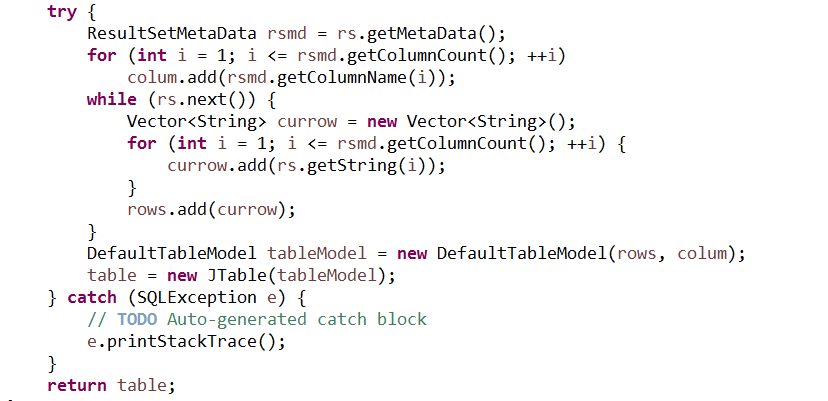
查询问题的显示结果如下：



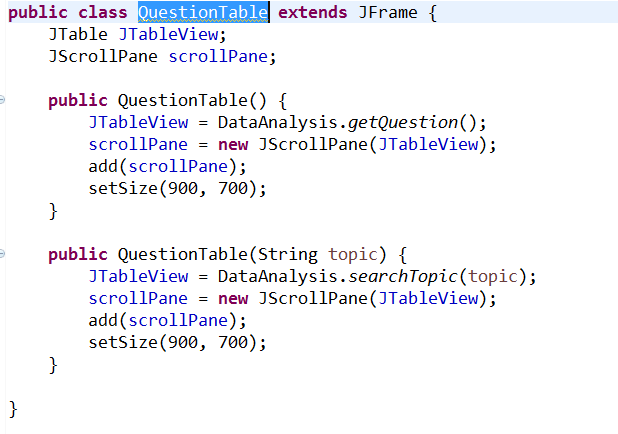
和查询问题不一样的是查看数据按钮和查询话题按钮，这两个部分的实现方法和查询问题略有不同，它们并没有将数据结果显示在textArea里面，而是通过QuestionTable类中的代码jTable描绘出来的，Jtable是标准库中一个用于数据库结果显示的类。具体的过程如下：



以上面的代码就是searchTopic方法，该方法首先从数据库中找到相应话题的问题，把该话题的所有问题的相关属性和记录提取出来存储在“rows”和“colm”里面。然后将这些添加到一个jtable里面代码如下：

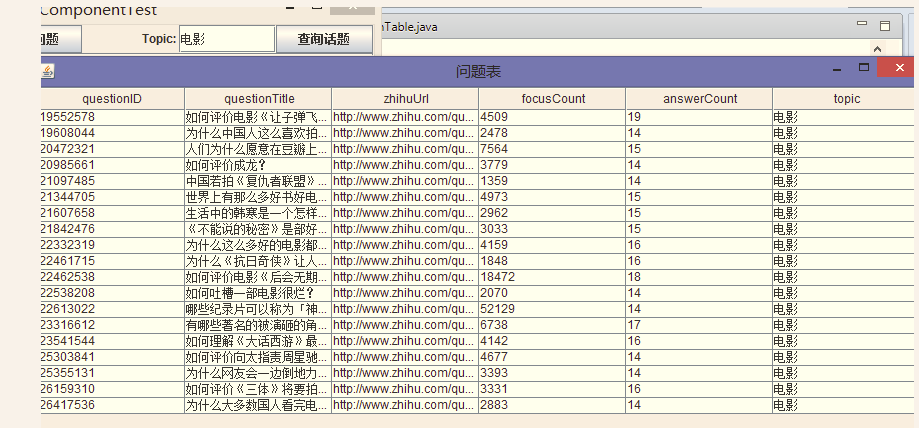
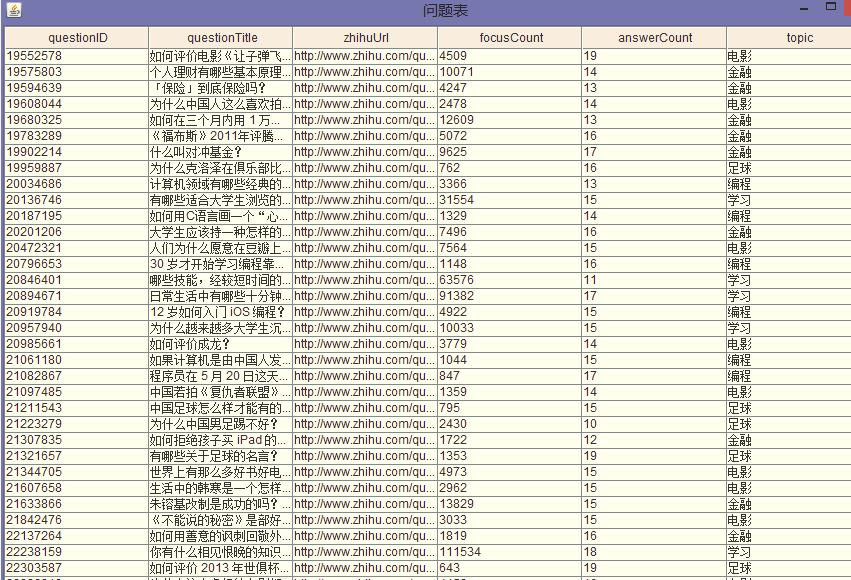


最后返回这个jtable的对象给QuestionTable类中QuestionTable方法来调用。与查询话题按钮类似，查看数据按钮其实就是列出数据库中所有存好的问题，起实现过程和上面如出一辙，整个questiontable类的设计实现如下：



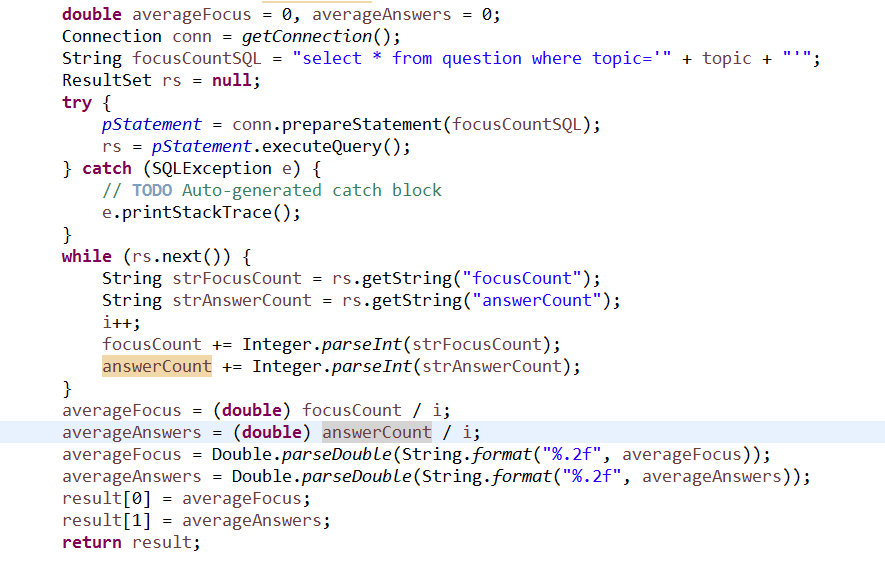
其中不带参数的构造函数是用来处理查看数据按钮的，其中带参数string topic的构造函数是用来处理查询话题按钮的。

查询话题和查看数据按钮的显示效果如下：

上面已经提到了三个按钮的处理函数的设计与实现分别是”查看数据“、”查询问题“、”查询话题“，下面介绍第四个按钮也是”分析数据”相关的处理函数的过程。

当我们点击分析数据时，最终会调用Analysis 类的getResult( )方法，该方法设计实现如下：



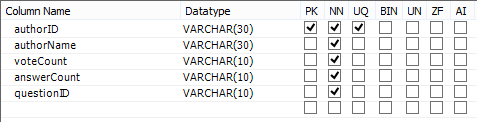
可以看出该方法可以得到各个话题的问题的平均关注人数，平均回答人数等信息并输出到textArea上面。其显示结果如下：



# 四 数据库设计

## 4.1 Author表

作者ID，作者姓名，赞成数，该回答的字数，问题ID



## 4.2 Financial 表

问题ID，该问题的回答个数



## 4.3 Learn表

问题ID，该问题的回答个数



## 4.4 Movie表

问题ID，该问题的回答个数



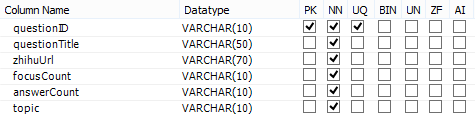
## 4.5 Program表

问题ID，该问题的回答个数



## 4.6 Question表

问题ID，问题标题，问题链接，关注人数，该问题的回答个数，问题主题



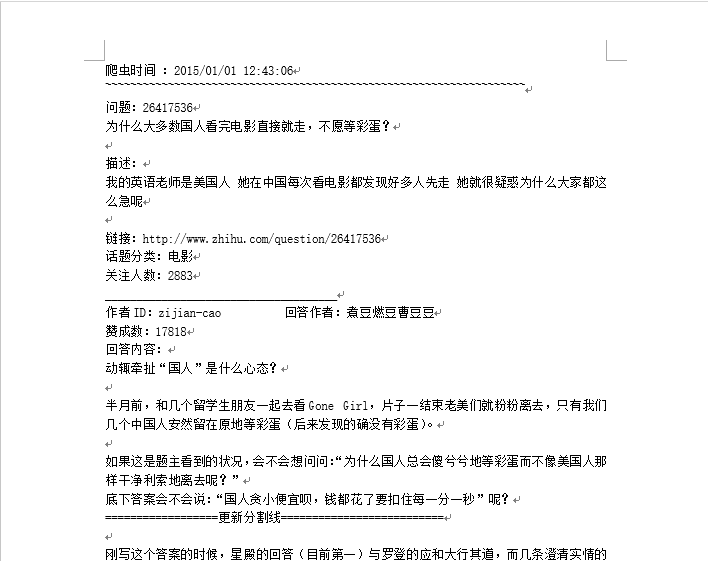
## 4.7 Soccer表

问题ID，该问题的回答个数

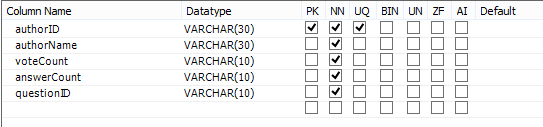


# 五 程序运行结果

## 5.1 word文档展示



## 5.2数据库展示



## 5.3查询界面展示

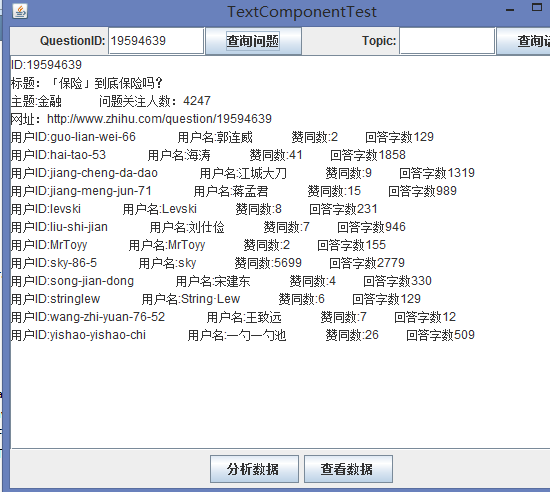
整体的软件分析界面。4大功能，通过输入问题ID查询每个问题详情。输入话题查询所有该话题下的问题。分析数据结果，查看所有问题结果。



点击查询数据按钮，JTable窗口读入数据库数据展示所有爬虫软件取得的问题。包括具体的问题ID,问题标题，问题的网址，问题的关注人数，及该问题的回答人数和所属话题。



通过对爬取问题数据的整合，得到每个话题下相关问题的平均关注人数和每个问题的回答个数。



通过输入每个具体问题的ID，得到每个问题具体的回答详情。有回答的用户人数，得到的赞成数，每个问题回答的字数。



通过输入话题类型，展示所有该话题下的问题。

# 六 成员分工

|  |  |  |
| --- | --- | --- |
| **姓名** | **学号** | **分工（详细填写）** |
| 吴林锴 | 1401210775 | 系统的整体设计框架，爬虫部分的各个代码类的设计和编码工作，参与数据库读取的设计和编码，及数据分析界面的设计和编码。文档的编写工作，工作总量为40%、 |
| 高煜东 | 1401210927 | 参与数据库的设计，及mysql环境的搭建，建库工作，实现数据库的读入和输出操作，及相关函数的设计与编码实现，文档的编写工作。工作总量15% |
| 周天亮 | 1401210895 | 参与数据分析的设计，及分析界面软件的编码工作，及整体软件的测试，文档的编写工作。工作总量15%。 |
| 狄烨 | 1301220961 | 参与爬虫部分的编码，及正则表达式的分析，文件写入本地的代码部分的编程，文档的编写工作。工作总量15%。 |
| 王勤晓 | 1401210749 | 参与数据分析的设计，分析数据调入函数的设计及编码，文档的编写工作，工作总量15% |

# 七 Github地址

Github仓库地址

https://github.com/894939677/adjavaforsspku2014.git

老师用自己的github账户可登陆查看。