

# 浙江大学大学计算机科学与技术学院

Java 程序设计课程报告

2017—2018 学年秋冬学期

题目	Web 搜索引擎
学号	3160104875
学生姓名	杨樾人
所在专业	软件工程
所在班级	软工 1602

目录

- 1 引言 ..... 1
  - 1.1 设计目的..... 1
  - 1.2 设计说明..... 2
- 2 总体设计..... 3
  - 2.1 功能模块设计..... 3
  - 2.2 流程图设计..... 3
- 3 详细设计..... 5
  - 3.1 爬取丁香园各个疾病的内容..... 5
  - 3.2 对每个疾病的内容建立索引..... 7
  - 3.3 用户通过输入查询关键词进行疾病的查询..... 9
- 4 测试与运行..... 11
  - 4.1 程序测试..... 11
  - 4.2 程序运行..... 11

# 1 引言

本次是用 Java 开发一个 Web 搜索引擎，需要学习 Web 爬虫、解析网页内容、对内容建立索引和查询等知识。通过本次作业，我可以对 Java 语言中的各项功能有更好的理解和使用，通过具体的程序来加深对 Java 语言的掌握，提高自己的编程水平，为以后的工作打下一定的基础。

## 1.1 设计目的

1. 写一个 Web 爬虫，爬取问答类或课程类网站的网页；
2. 解析网页内容，对内容进行结构化，并存储到文件中；
3. 为内容建立索引；
4. 通过命令行进行内容检索，并展示内容列表

Web 搜索引擎是非常有实际意义的一个工程，不仅需要爬取网上的信息，还需要对这些信息建立索引，便于查询，本次我完成的搜索引擎功能如下：

1. 使用 Jsoup 爬取丁香园的所有的疾病信息链接，约有一千一百个。
2. 对每个疾病信息链接再次进行爬取，对网页的内容进行解析，把有用的信息下载到本地文档之中。
3. 使用 Lucene 对每个文档建立索引，并添加查询功能。
4. 用户通过在命令行输入字符串进行内容检索，程序会将检索到含有关键词的文档返回，并按相关度返回前 20 个最相关的文档。
5. 用户可以通过输入文档前面的序号查看对应疾病的内容。

## 1.2 设计说明

本程序采用 Java 程序设计语言，在 Eclipse 平台下编辑、编译与调试。具体程序由我个人开发而成。工作时间轴如表 1 所示：

表 1 工作时间轴表

时间	完成的主要工作
12 月 1 日	整个程序前期的需求分析和整体功能的架构 阅读 Lucene 和 Jsoup 的库文件，熟悉用法
12 月 7 日	完成代码的编写工作，并对代码进行测试，查找 bug。
12 月 10 日	完成报告的撰写工作

## 2 总体设计

### 2.1 功能模块设计

本程序需实现的主要功能有：

1. 爬取丁香园疾病的链接数据集
2. 爬取每个疾病链接的网页内容
3. 对爬取下来的每个疾病的内容建立索引
4. 用户通过输入查询关键词进行疾病的查询
5. 系统返回与用户查询关键词相关度最高的前 20 个疾病
6. 用户通过输入疾病前的代号，查看疾病的详细信息。

程序的总体功能如图 1 所示：

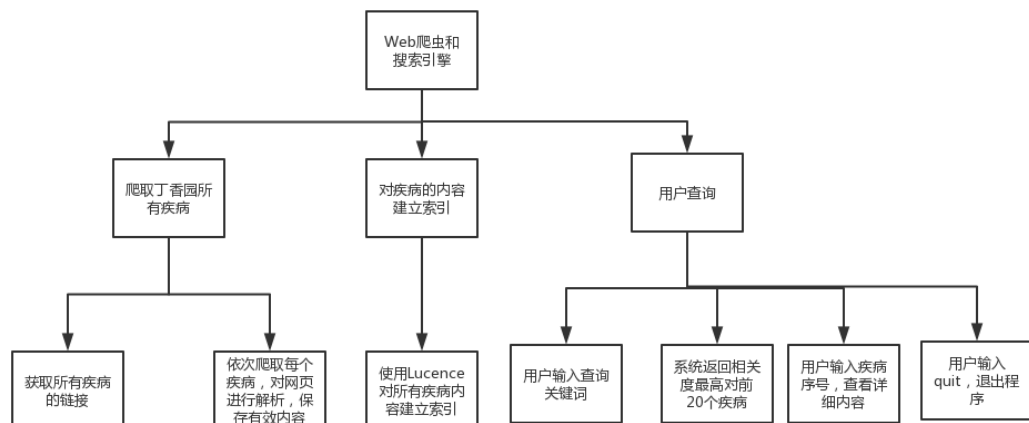


图 1 总体功能图

### 2.2 流程图设计

程序总体流程如图 2 所示：

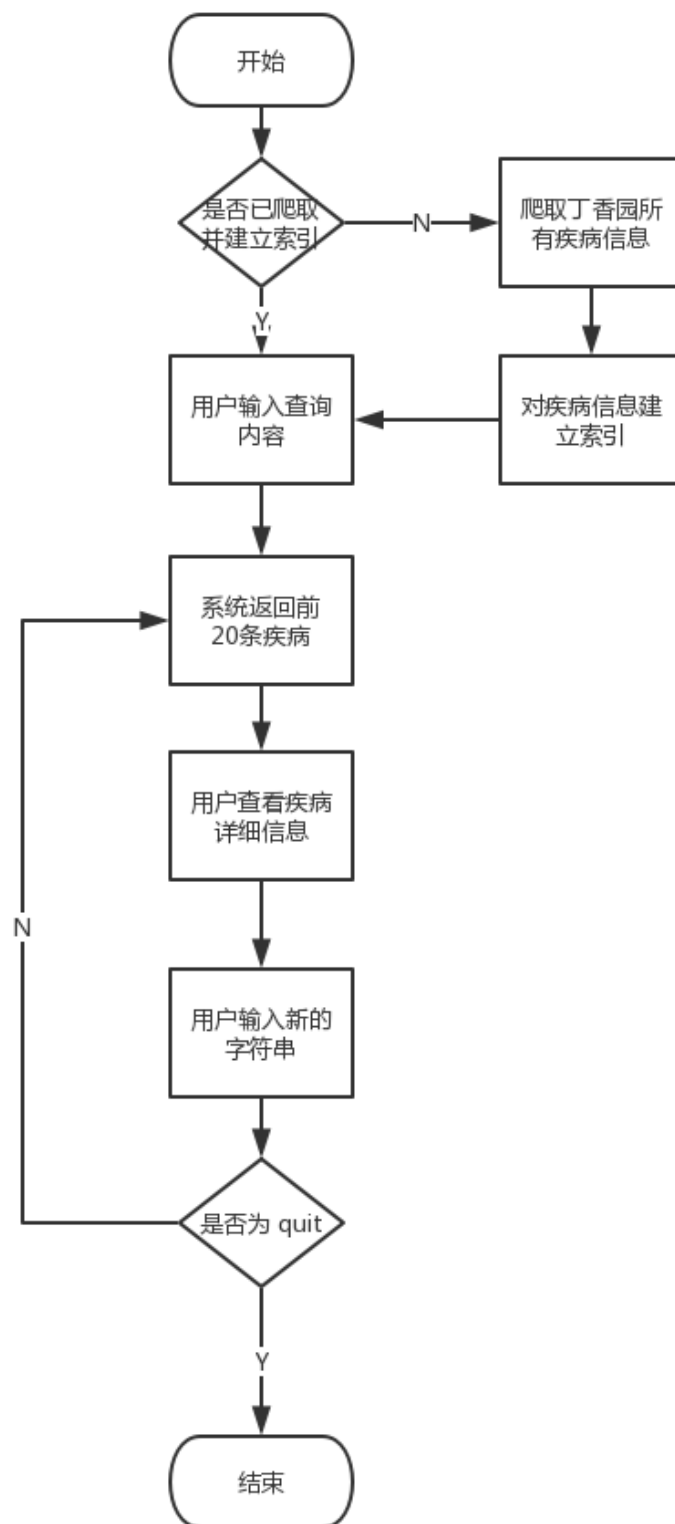


图 2 总体流程图

### 3 详细设计

本搜索引擎主要包含三部分，爬取网站内容，建立索引，用户查询，以下将做详细介绍。

#### 3.1 爬取丁香园各个疾病的内容

打开丁香园疾病的汇总页面，每个疾病都有一个对应的链接，每种疾病的页面内容相同，有 6 个 id，从 1-6，分别对应简介、症状、病因、诊断、治疗、生活。每个 id 中，包含 disease-detail-content-title 和 disease-detail-content-box 两部分，disease-detail-content-content 是对 title 的详细展开介绍。

```
▼<div id="0"> == 50
  ▶<div class="disease-detail-content-title">_</div>
  ▶<div class="disease-detail-content-box">_</div>
</div>
▼<div id="1">
  ▶<div class="disease-detail-content-title">_</div>
  ▶<div class="disease-detail-content-box">_</div>
</div>
▼<div id="2">
  ▶<div class="disease-detail-content-title">_</div>
  ▶<div class="disease-detail-content-box">_</div>
</div>
▼<div id="3">
  ▶<div class="disease-detail-content-title">_</div>
  ▶<div class="disease-detail-content-box">_</div>
</div>
▼<div id="4">
  ▶<div class="disease-detail-content-title">_</div>
  ▶<div class="disease-detail-content-box">_</div>
</div>
▼<div id="5">
  ▶<div class="disease-detail-content-title">_</div>
  ▶<div class="disease-detail-content-box">_</div>
</div>
▶<div id="6">_</div>
```

图 3 丁香园疾病网页

```
▼<div id="0"> == 50
  ▼<div class="disease-detail-content-title">
    ::before
    "简介"
  </div>
  ▼<div class="disease-detail-content-box">
    ▼<div class="disease-detail-content-content">
      <h2>艾森门格综合征是什么? </h2>
      ▶<p>_</p>
      ▶<ul>_</ul>
      <h2>艾森门格综合征可以治愈吗? </h2>
      ▶<p>_</p>
      <h2>艾森门格综合征会影响寿命吗? </h2>
      <p>病人的生存时间是受疾病影响的，没有控制的艾森门格综合征病人，往往死于心力衰竭，预期平均寿命 37 岁，也有病人存活至 60 岁。</p>
      <h2>艾森门格综合征常见吗? </h2>
      <p>艾森门格综合征往往在先天性心脏病患者进入成人期后才会出现症状。随着先天性心脏病手术治疗的发展，艾森门格综合征已经逐渐在减少了，但仍不容忽视。</p>
    </div>
  </div>
</div>
```

图 4 丁香园疾病网页详细信息

此部分只需要两个函数，主要是对疾病详细页面中有效内容的筛选。Spider 类中的 allUrl 存储页面中所有的疾病链接，currentUrl 是当前要爬取的页面，documentName 是存放的文件名。调用 Spider 类中的 getAllUrl 函数，返回所有的疾病链接，对疾病链接这个 String 数组进行遍历，对每个疾病链接调用 spiderUrl 函数，对该页面内容进行爬取，并存储到文档中。

UML 图如下：

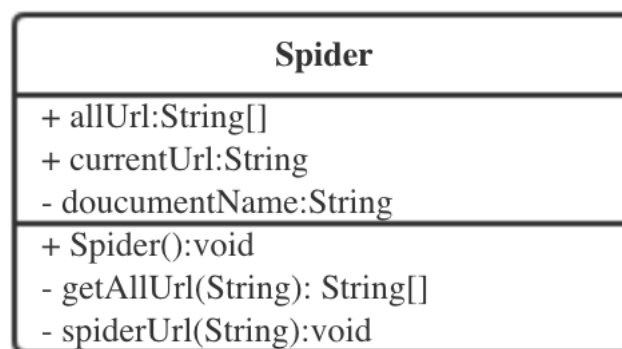


图 5 爬虫 UML 图

以下是 UML 图中有关数据和方法的详细说明：

(1) 成员变量

- a) allUrl 存储页面中所有的疾病链接，
- b) currentUrl 是当前要爬取的页面，
- c) documentName 是存放的文件名

(2) 方法

- a) Spider() 方法是此类的入口函数，供主函数调用
- b) getAllUrl(String) 是爬取丁香园疾病主页面的所有链接，并返回。
- c) spiderUrl(String) 是爬取该页面的主要内容，并对页面进行解析，格式化保存有价值的内容。



爬取页面的流程图如下：

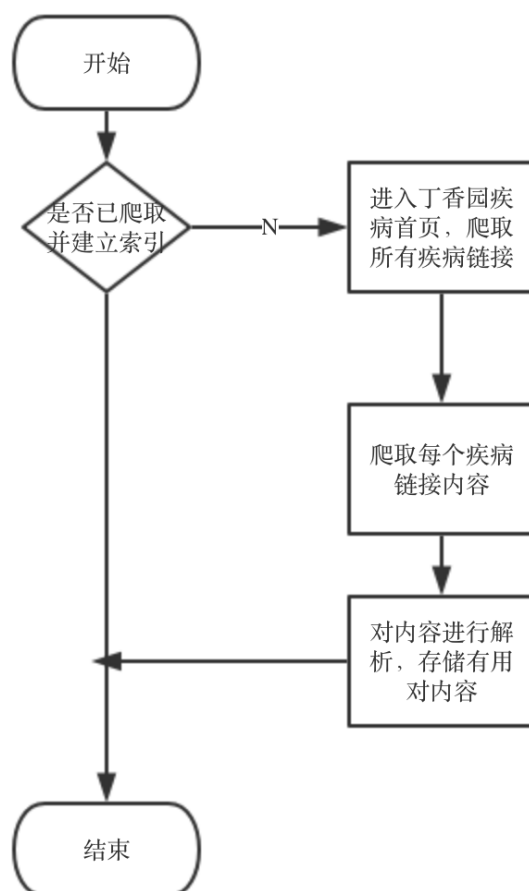


图 6 爬虫设计流程图

### 3.2 对每个疾病的内容建立索引

CreateIndex 通过调用 Lucene 包对每种疾病中的内容建立索引。该类使用时需要获取文档存储的位置。

Lucene 的索引包含多个 Document, 每个 Document 可以包含多个 Field 对象。我把每个疾病的内容分别放到一个 Document 中, 然后添加到索引之中。Document 中存放了两个 Field, 一个是疾病的名字, 另一个是疾病的详细信息。

在 Query 时, 只需要调用 Lucene 的 Query 函数, 可以得到 Hits, 里面包含了多个 doc, 并按相关度对文档进行了排序。

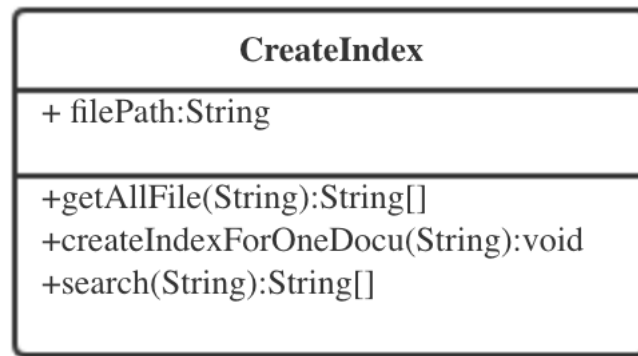


图 7 创建索引 UML 图

以下是 UML 图中有关数据和方法的详细说明：

(1) 成员变量

a) filePath 存储文档的位置。

(2) 方法

b) getAllFile(String) 获取文件夹内的所有疾病文件，返回文件名的字符串数组。

c) createIndexForOneDocu(String) 是取出该文件的内容，并添加到 Field 之中，然后加入到 Document 中，添加到 Lucene 的索引里。

d) search(String) 在索引中查找用户输入的字符串，并返回相关度最高的前 20 个文档的名字。

建立索引的流程图如下：



图 8 建立索引设计流程图

### 3.3 用户通过输入查询关键词进行疾病的查询

在建立好索引之后，主函数提供给用户搜索关键词的方法。

用户输入要查询的关键词，Query 类调用 CreateIndex 类中的 search 方法对用户输入的关键词进行查找，返回包好关键字的相关度最高的前 20 个信息，显示在命令上，用户可以输入序号，查看该疾病的详细信息。

用户可以输入其他字符串，再次进行查找，也可以输入 quit，退出程序。

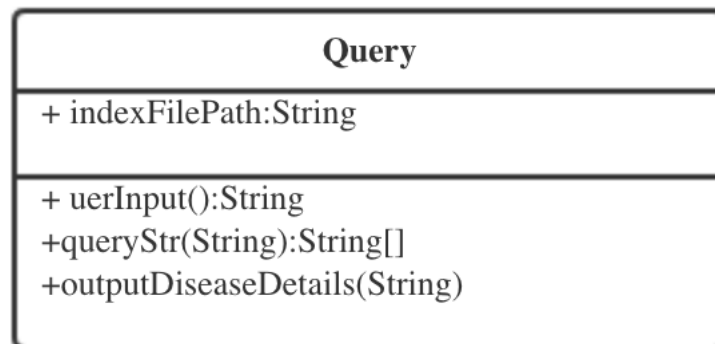


图 9 查询 UML 图

以下是 UML 图中有关数据和方法的详细说明：

(1) 成员变量

a) indexFilePath 存放了 Lucene 建立的索引的位置。

(2) 方法

b) userInput() 方法是读入用户的输入，返回字符串。

c) queryString(String) 是在索引中查找该关键词所在的文档，调用 CreateIndex 类中的 search 方法，并返回前 20 个相关度最高的文档名。

d) outputDiseaseDetails(String) 是输出该文档的详细内容，供用户查看。

用户查询的流程图如下：

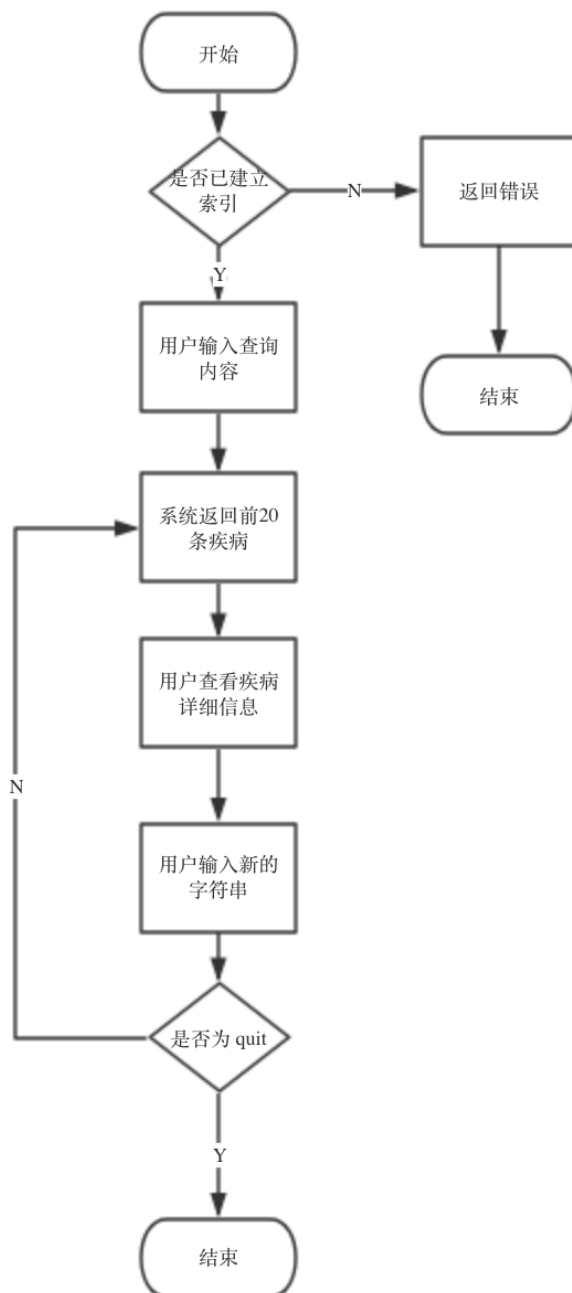


图 10 查询设计流程图

## 4 测试与运行

### 4.1 程序测试

在程序代码基本完成后，经过不断的调试与修改，最后测试本次所设计的Web爬虫和搜索引擎能够正常运行，没有出现明显的错误和漏洞，但是在一些细节方面仍然需要完善，比如用户交互方面，可以进行优化。总的来说本次设计在功能上已经基本达到要求，其他细节方面有待以后完善。

### 4.2 程序运行

爬虫爬取的疾病信息图所示：



图 11 疾病文档



图 12 疾病信息内容

建立索引如图所示：

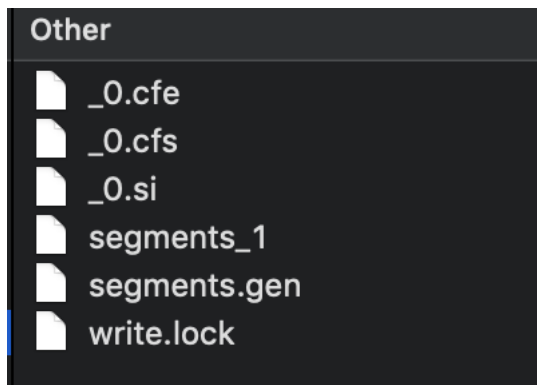


图 13 索引文件

用户查询如图所示：

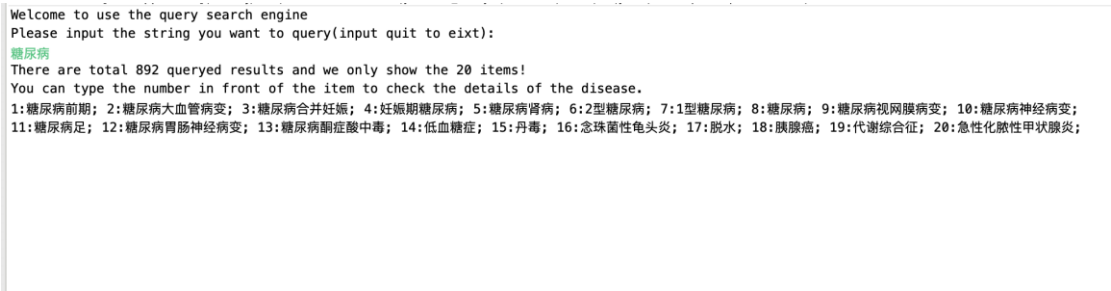


图 14 用户查询界面

用户查看详细信息，如图所示：

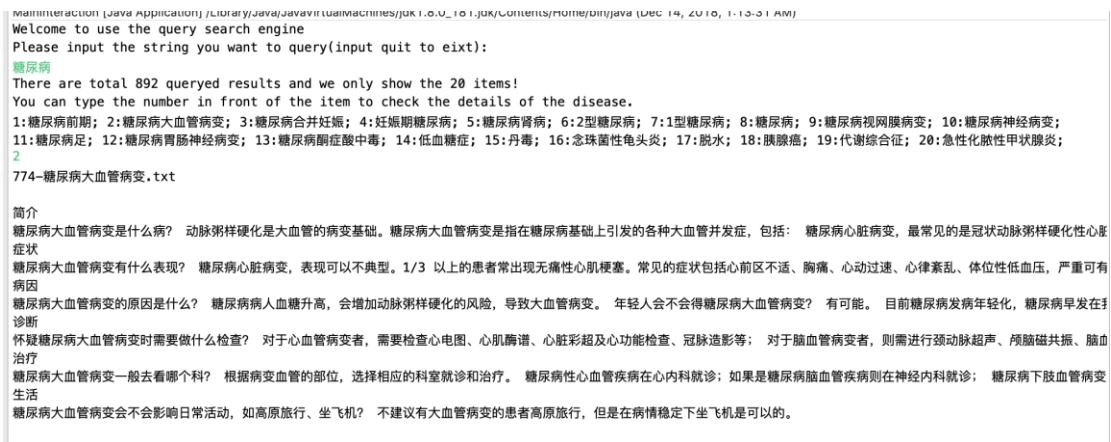


图 15 查看疾病详细信息

用户持续查询如图所示：

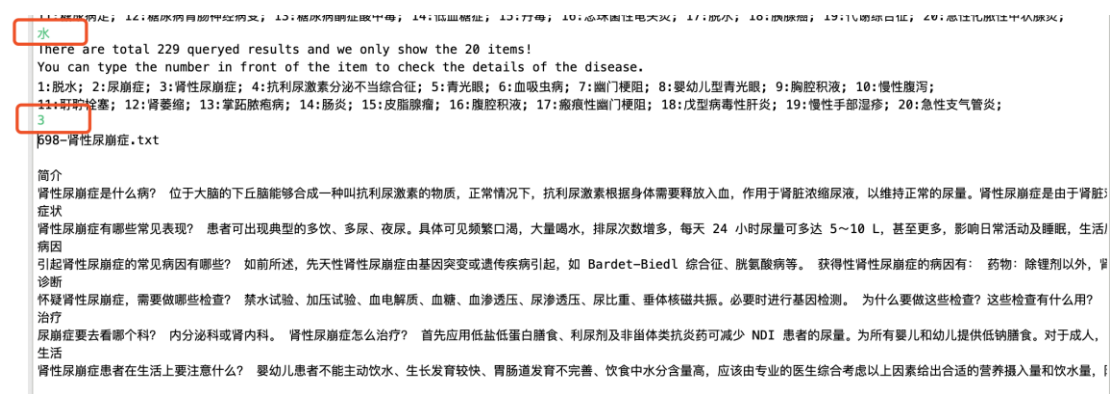


图 16 用户持续查询

用户退出查询如图所示：

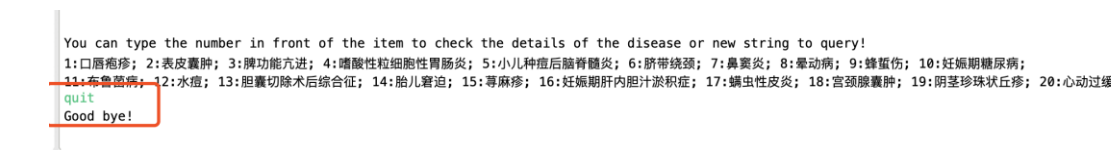


图 17 用户退出查询

## 5. 总结

此次的 Web 编程是我第一次使用 Java 的库进行爬取网站和建立搜索引擎，第一感觉是 Java 的库太方便了。我之前也有用 python 的 beautifulsoup4 进行过爬取百度百科并建立搜索引擎的尝试，以为已经是比较方便的了。此次使用 Java 进行爬虫和建立索引，体会到了 Java 的方便之处。

使用 Jsoup 对网页内容进行爬取并没有遇到太大的困难，我比较感兴趣对是 lucene。引用一段网上对 lucene 重要概念的介绍：

*Document*：索引包含多个 *Document*。而每个 *Document* 则包含多个 *Field* 对象。  
*Document* 可以从数据库表里取出的一堆数据，可以是一个文件，也可以是一个网页等。  
注意，它不等同于文件系统中的文件。

*Field*：一个 *Field* 有一个名称，它对应 *Document* 的一部分数据，表示文档的内容或者文档的元数据（与下文中提到的资源元数据不是一个概念）。一个 *Field* 对象有两个重要属性：*Store*（可以有 YES, NO, COMPACT 三种取值）和 *Index*（可以有 TOKENIZED, UN\_TOKENIZED, NO, NO\_NORMS 四种取值）

*Query*：抽象了搜索时使用的语句。

*IndexSearcher*：提供 *Query* 对象给它，它利用已有的索引进行搜索并返回搜索结果。

*Hits*：一个容器，包含了指向一部分搜索结果的指针。

我一开始按照老师给的样例进行了代码编写，觉得 lucene 建立索引的流程如下：将输入的数据源统一为字符串，然后从数据源提取数据，创建合适的 *Field* 添加到对应数据源的 *Document* 对象之中。我只使用了 *TextField*，因为我在把文档中的数据放到 *StringField* 时会无法建立索引，我还没有查到为什么，有可能是超过长度了。

当从 *IndexSearcher* 对象得到搜索结果 (*Hits*) 之后，可以直接从中获取需要的值，再格式化予以输出。但是为了交互的简洁性和美观，我选择了列出前 20 个相关度最高的疾病并对他们编号，用户可以输入编号进行更加详细的查看，我认为我是给用户做了一个接口，这个接口提供了便捷性，正如 lucene 包一样简洁。

通过该次作业，我对 Java 调用库有了更深入对认识，理解了程序构造的一



般原理和基本实现方法。能够把课堂上学的知识通过自己设计的程序表示出来，加深了对理论知识的理解。在写代码调试的过程中，对 **Java** 的特性也有了更加深入的理解。

## 参考文献

- [1] 耿祥义. Java 大学实用教程[M]. 北京: 清华大学出版社, 2009.
- [2] 耿祥义. Java 课程设计[M]. 北京: 清华大学出版社, 2008.
- [3] <https://lucene.apache.org/core/>
- [4] <https://jsoup.org/>