

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

2020—2021 学年秋冬学期

题目	Web 搜索引擎
学号	3180106071
学生姓名	刘轩铭
所在专业	软件工程
所在班级	软工 1801

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	4
3 详细设计	5
3.1 爬取图书网站的内容.....	5
3.2 对每本图书的各字段建立索引.....	11
3.3 用户通过输入查询关键词进行书籍的查询.....	13
4 测试与运行	14
4.1 程序测试	14
4.2 程序运行	14

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，爬取当当、京东等图书购买网站的网页；
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+htmlUnit 爬取当当网的大量书籍信息。
2. 对网页的内容进行解析，把有用的信息存储到本地文档之中。
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 10 个最相关的文档以及相关的信息。

1.2 设计说明

本程序采用 Java 程序设计语言，使用 Maven 包管理工具，在 IntelliJ IDEA 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，htmlUnit 和 Jsoup 的库文件，熟悉用法
12 月 10 日	完成代码的编写工作，并对代码进行测试，查找 bug。
12 月 12 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. 爬取当当网图书目录下各个分类的链接数据；
2. 在每个分类下，根据指定好的需要爬取的 Page 数量，获取一定数量的图书链接；
3. 对每个图书链接，根据需要的信息进行爬取，具体获得了图书的书名，作者，出版社，分类，价格，图片链接，作者推荐，内容简介，目录等有用信息；
4. 对爬取下来的每个图书的内容建立索引；
5. 用户通过输入查询关键词进行图书的查询，具体可以对作者，出版社，书名，分类四个字段进行选择 and 查询。系统返回与用户查询关键词相关度最高的前 10 个疾病

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：

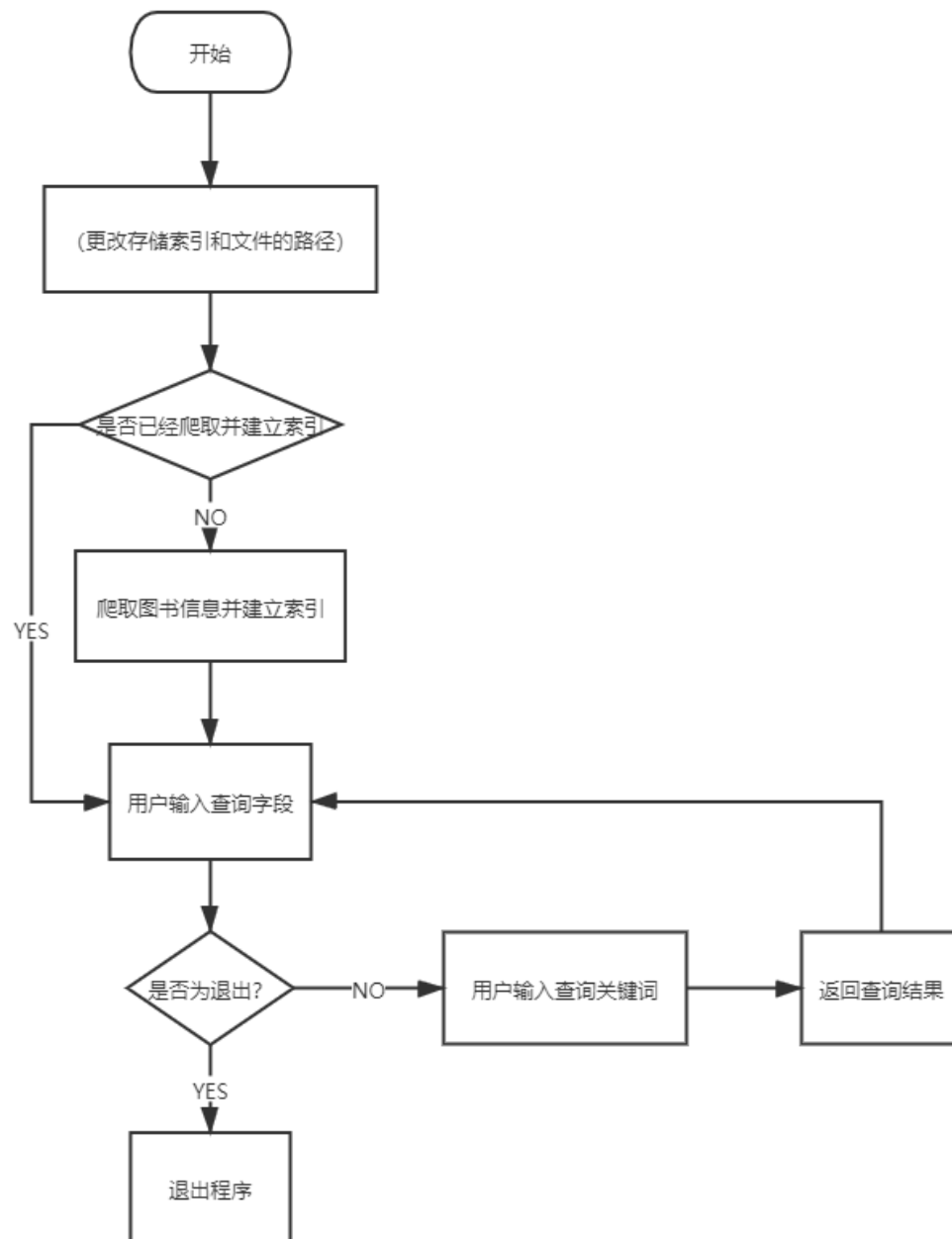


图 2 总体流程图

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 爬取图书网站的内容



图 3 当当分类汇总网页 (<http://category.dangdang.com/>)

打开当当的分类汇总页面，可以看到左侧，每个图书的分类都有一个对应的链接，通过获取每个分类的 href 属性，可以得到每个分类的 url 链接。利用一个循环，可以爬取所有分类的信息。



图 4 当当分类汇总网页部分源代码

接下来，对每个分类下的图书进行爬取。



图 5 当当各个分类下网页 以青春文学为例

(<http://category.dangdang.com/cp01.01.00.00.00.html>)



图 6 当当各个分类下网页部分源代码

可以看到，该页面下面呈现了许多（约 60 本）该分类图书的信息和详细信息链接，以第一本书为例，我们只需要获得该书对应的 url，就可以进入详细信息页面，获取更多的信息。此外，该网页可以翻页，其规律是，如果是第一页，显示一个 BASE_URL (<http://category.dangdang.com>) + URI 定位符；如果不是第一页，则为 BASE_URL (<http://category.dangdang.com>) + pg 页数 + URI 定位符。故我们可以有规律的进行翻页，爬取该分类的所有图书。

接下来对详细信息进行爬取。



图 7 当当图书详细信息-1 (<http://product.dangdang.com/29155128.html>)



图 8 当当图书详细信息-2 (<http://product.dangdang.com/29155128.html>)

可以看到,每本书的信息主要在上面两个部分中进行程序。我们需要的书名,价格,作者,出版社,分类,作者介绍,内容简介,编辑推荐,目录等信息,都可以从上面显式获取。但是需要注意的是,该网页采取 JS 动态加载,且加载速度较慢,如果使用 Jsoup 直接获取,可能会出现无法定位元素的错误,于是我们使用 htmlUnit 获取页面信息,并设置 waitForBackgroundJavaScript(15000),让 WebClient 对象加载 JS 脚本 15 秒钟,然后对资源进行获取。



图 9 当当图书详细信息头部源代码示例

标题,作者,出版社等信息位于头部,它们的获取较为简单。以作者为例,我们只需要使用 Jsoup 定位对应 id (id=“author”),然后获取其中的 text(),即可完成获取。



图 10 当当图书详细信息尾部源代码示例

编辑推荐,作者介绍,目录等信息位于尾部。以作者介绍为例,同样,我们首先定位对应的 id (id=“authorIntroduction”),但是需要注意的是,有一些书籍的介绍中有“显示全部”的按钮,而有一些没有,其对应功能是把 textarea 中的内容加载到 span 标签中来。所以,首先判断该按钮的存在,如果存在,直接获取 textarea 中的内容;如果不存在,获取第一个 span 标签中的内容。

此部分只需要三个函数,主要是对书籍详细信息页面的 url 进行获取。Crawler 类中的 crawlAllCategories 获取所有分类的 url,并调用

crawlEachCategory 对每个分类的书籍详细信息 url 进行获取。然后对于每一本书籍，调用 crawlEachBook，爬取详细信息并保存到文件中。MAX_PAGE 指定了每个分类下需要爬取的页数，为了方便起见，这里设为 1（可以修改）。

由于爬取的内容中存在许多的换行和空格，在保存文件时，我们用字符流将每一本书籍的信息保存到一个 txt 中，其中，各个字段之间使用“@@@”分隔符进行分割，方便之后建立索引时的读取。

UML 图如下：

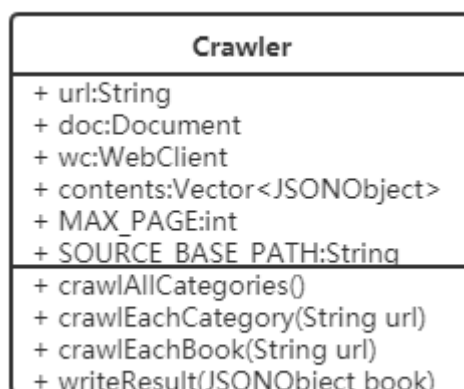


图 11 爬虫 UML 图

以下是 UML 图中有关数据和方法的详细说明：

(1) 成员变量

- a) url 爬虫的起始界面，即当当网的分类页面
- b) doc 爬取的 Jsoup 文档对象
- c) wc htmlUnit 的浏览器对象
- d) contents 存储每本书信息的向量
- e) MAX_PAGE 常量，指定每个分类下爬取书籍的页数
- f) SOURCE_BASE_PATH 存储文件的根路径

(2) 方法

- a) crawlAllCategories() 爬取每个分类的 url，并调用 crawlEachCtegory
- b) crawlEachCategory() 爬取一个分类的书籍，调用 crawlEachBook
- c) crawlEachBook() 爬取一本书的详细信息
- d) writeResult() 将一本书的信息写入文件中

爬取页面的流程图如下：

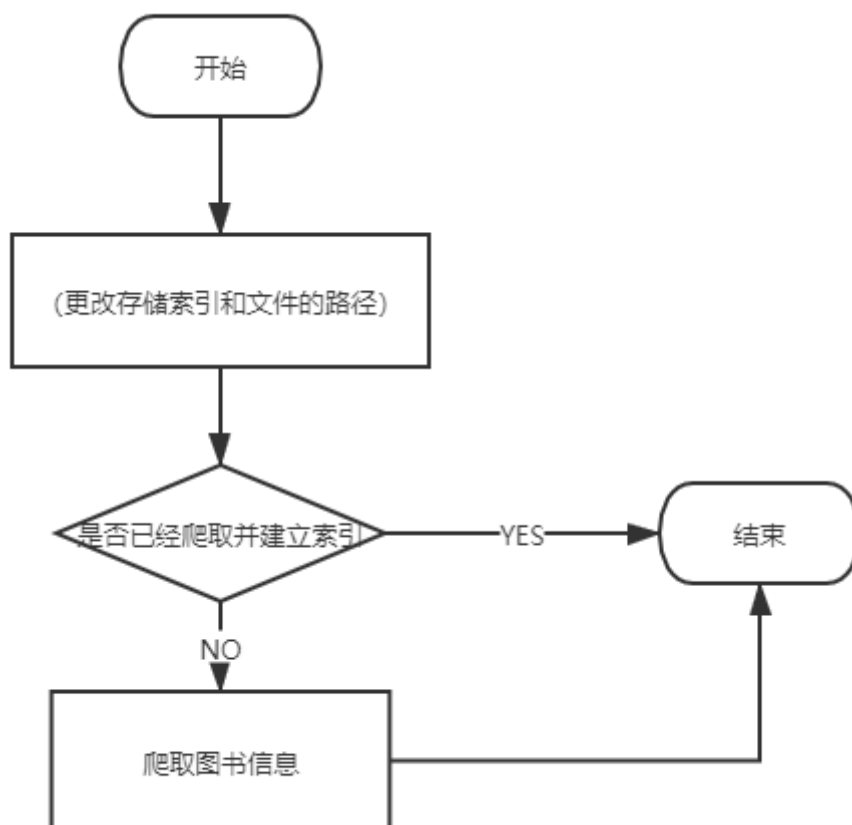


图 12 爬虫设计流程图

3.2 对每个图书的内容建立索引

CreateIndex 通过调用 Lucene 包对每本书的内容建立索引。该类使用时需要获取文档存储的位置。

Lucene 的索引是针对多个 Document 的。每个 Document 相当于一本书的内容，可以包含多个 Field 对象。而 Field 就相当于作者，标题等字段。我把每本书的内容分别放到一个 Document 中，然后添加到索引之中。Document 中存放了多个 Field。其中 textField（可以被用来分词和查询）包含作者，书名，出版社，分类；其他都是 storedField（用于存储，不用来查询）。

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。我们只需要返回指定数量的排名靠前文档，就可以完成查询。

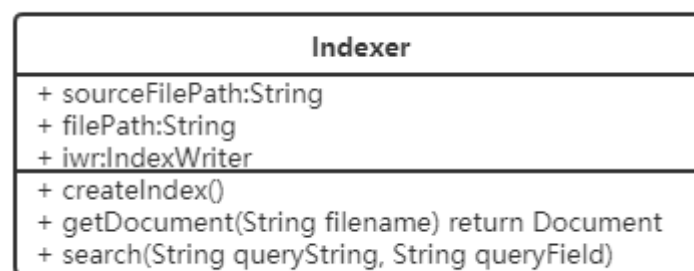


图 13 创建索引 UML 图

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 存储索引的位置。
- b) sourceFilePath 存储文件的位置
- c) iwr 索引写出器

（2）方法

- a) getDocument 读取一个文件，将内容分字段添加到 Field 之中，然后加入到 Document 中，添加到 Lucene 的索引里。

- b) createIndex 是取出文件的内容，对每个 Document，调用 getDocument 方法
- c) search() 在索引中查找用户输入的字符串，并返回查询结果和相关度最高的前 10 个文档的名字。

建立索引的流程图如下：



图 14 建立索引设计流程图

3.3 用户通过输入查询关键词进行图书的查询

在建立好索引之后，主函数提供给用户搜索关键词的方法。

用户输入要查询的字段（在作者，标题，分类和出版社中选择）。如果输入 0 则代表退出。

用户之后输入要查询的关键词，主函数调用 Indexer 类中的 search 方法，将输入的字段和关键词进行查询。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 10 个信息和他们的详细信息，显示在命令行界面上。

用户可以输入其他字符串，再次进行查找，也可以输入 0，退出程序。

4 测试与运行

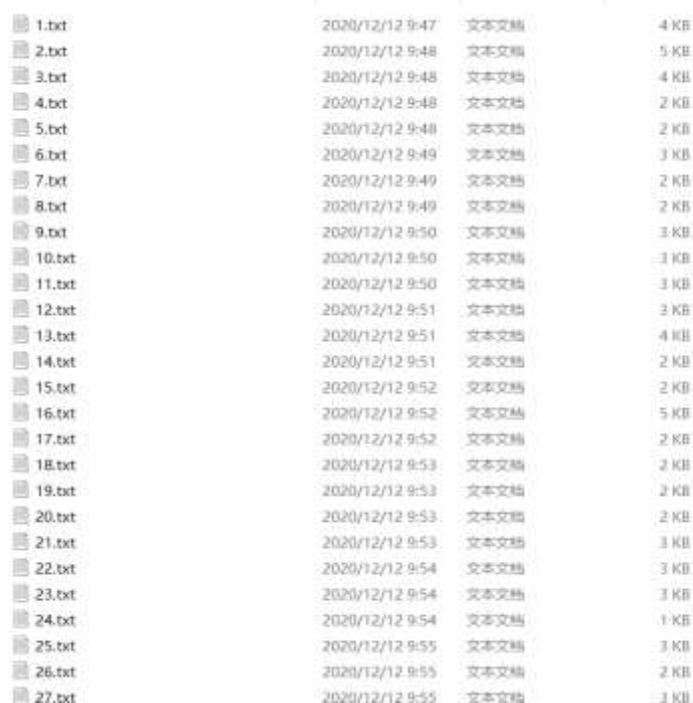
4.1 程序测试

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。

由于当当网设计较为复杂，一些网页可能会出现异常（比如标题和作者信息不存在等），爬虫需要根据这些情况进行特定的修改。所以，后期也可以针对该情况进行进一步完善。

4.2 程序运行

爬虫爬取的书籍信息图所示（目前爬取了 1000 条左右数据）：



1.txt	2020/12/12 9:47	文本文档	4 KB
2.txt	2020/12/12 9:48	文本文档	5 KB
3.txt	2020/12/12 9:48	文本文档	4 KB
4.txt	2020/12/12 9:48	文本文档	2 KB
5.txt	2020/12/12 9:48	文本文档	2 KB
6.txt	2020/12/12 9:49	文本文档	3 KB
7.txt	2020/12/12 9:49	文本文档	2 KB
8.txt	2020/12/12 9:49	文本文档	2 KB
9.txt	2020/12/12 9:50	文本文档	3 KB
10.txt	2020/12/12 9:50	文本文档	3 KB
11.txt	2020/12/12 9:50	文本文档	3 KB
12.txt	2020/12/12 9:51	文本文档	3 KB
13.txt	2020/12/12 9:51	文本文档	4 KB
14.txt	2020/12/12 9:51	文本文档	2 KB
15.txt	2020/12/12 9:52	文本文档	2 KB
16.txt	2020/12/12 9:52	文本文档	5 KB
17.txt	2020/12/12 9:52	文本文档	2 KB
18.txt	2020/12/12 9:53	文本文档	2 KB
19.txt	2020/12/12 9:53	文本文档	2 KB
20.txt	2020/12/12 9:53	文本文档	2 KB
21.txt	2020/12/12 9:53	文本文档	3 KB
22.txt	2020/12/12 9:54	文本文档	3 KB
23.txt	2020/12/12 9:54	文本文档	3 KB
24.txt	2020/12/12 9:54	文本文档	1 KB
25.txt	2020/12/12 9:55	文本文档	3 KB
26.txt	2020/12/12 9:55	文本文档	2 KB
27.txt	2020/12/12 9:55	文本文档	3 KB

图 15 书籍信息文档

图书 > 青春文学 > 悬疑/惊悚/恐怖/玄幻小说
青年作家，其作品被改编成多个国家和地区的影视剧、畅销海外并：
已出版作品：
《嫌疑人的告白（二分之一）》
《我们不一样，年轻又怎样》
《穿越人海拥抱你》
《不夜城迷，谁是第一次当大人》
《你要好好过》
影视剧集：《青春文学》
作者：茅子文 白时光 出版：2020年12月 定价：¥24.90 册数：第一卷 悬疑/惊悚/恐怖/玄幻小说 第二卷 悬疑/惊悚/恐怖/玄幻小说 第三卷 悬疑/惊悚/恐怖/玄幻小说 第四卷 悬疑/惊悚/恐怖/玄幻小说 第五卷 悬疑/惊悚/恐怖/玄幻小说 第六卷 悬疑/惊悚/恐怖/玄幻小说 第七卷 悬疑/惊悚/恐怖/玄幻小说 第八卷 悬疑/惊悚/恐怖/玄幻小说 第九卷 悬疑/惊悚/恐怖/玄幻小说
每个人都是嫌疑人。
每个人都不为人知的另一面。
然而，扑朔迷离的命案，因其中一个嫌疑人的自首而变得扑朔迷离。
这是一场以爱为名的模拟犯罪。
这是一场秘密深处的人性角斗。
抑或是一场旷日持久的炽热告白？
凶手到底是谁。反转到底一步！
经典语录：
☆遇见你之后，我只希望生活有一种可能，那就是和你在一起。
☆人的记忆是有规律可循的，记忆力是一条向下的曲线，无论什么事情，只要你不再去想，都会逐渐被遗忘。
☆当你可能因为我爱你时，我离开你；当你真正受到伤害时，我才步不高。
☆你看，人生真是场无情的戏弄。你大费周章地保护一个人，到头来却发现，是他在保护你后知后觉的自己。☆☆☆☆

图 16 书籍信息内容

建立索引如图所示：







	_0.cfe	2020/12/12 9:42	CFE 文件	1 KB
	_0.cfs	2020/12/12 9:42	CFS 文件	518 KB
	_0.si	2020/12/12 9:42	SI 文件	1 KB
	segments.gen	2020/12/12 9:42	GEN 文件	1 KB
	segments_1	2020/12/12 9:42	文件	1 KB
	write.lock	2020/12/12 9:42	LOCK 文件	0 KB

图 17 索引文件

用户查询如图所示：

1. 书名 2. 作者 3. 出版社 4. 分类 8. 退出
输入需要查找的字段对应序号：
2
输入需要查找的内容：
子文
本次搜索共找到24条数据，显示排名靠前的结果
书名：《嫌疑人的告白（当当专享）》+《保证书》+《创作手札》+《告白卡片》！百万畅销书作家茅子文首部长篇悬疑力作！
作者： 茅子文 白时光 出品
出版社： 江苏凤凰文艺出版社
价格：¥ 24.90
分类：图书 > 青春文学 > 悬疑/惊悚
内容简介：一部连载的热门悬疑小说，书里犯罪的手段竟然在现实中重演。
精于布局的小说家，悬疑作家的富家养女，突然致命的养女闺蜜。
每个人都是嫌疑人。
每个人都不为人知的另一面。
然而，扑朔迷离的命案，因其中一个嫌疑人的自首而变得更扑朔迷离。
这是一场以爱为名的模拟犯罪。
这是一场秘密深处的人性角斗。
抑或是一场旷日持久的炽热告白？
凶手到底是谁。反转到底一步！
经典语录：
☆遇见你之后，我只希望生活有一种可能，那就是和你在一起。
☆人的记忆是有规律可循的，记忆力是一条向下的曲线，无论什么事情，只要你不再去想，都会逐渐被遗忘。
☆当你可能因为我爱你时，我离开你；当你真正受到伤害时，我才步不高。
☆你看，人生真是场无情的戏弄。你大费周章地保护一个人，到头来却发现，是他在保护你后知后觉的自己。
书名：嫌疑人的告白（当当专享）+《嫌疑人的保证书》+《创作手札》+《告白卡片》。百万畅销书作家茅子文首部长篇悬疑力作。）
作者： 茅子文 白时光 出品
出版社： 江苏凤凰文艺出版社

图 18 用户查询界面

用户查看详细信息，如图所示：

5. 总结

此次的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 request, beautifulsoup4, selenium 等库进行过爬取信息的尝试，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。通过阅读网上对于 Lucene 的介绍文章，我从无到有的学会了这个工具的使用方法。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。

通过阅读相关文章，我深刻的明白了该工具的使用方法，明白了其实它的使用都是模板化和固定化的。并没有很大的使用障碍。

在编写过程中，碰到的最大的困扰莫过于在 Maven 工程下，相对路径的基准是哪里？一开始我不是很清楚，进行了大量的尝试，出了一堆的 bug，最后阅读网上的资料后明白了，其基准路径位于 target 目录下生成的 Class 文件那里——也就是 JVM 启动的位置，这让我对于 JVM 运行方法有了更深地理解。

此外，我还学会了使用 Maven 对于 Java 文件进行管理，掌握了如何从网络上获取 jar 文件，以及如何将本地的 jar 文件添加到 Maven 工程中。

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] https://blog.csdn.net/weixin_42633131/article/details/82873731
- [2] <https://jsoup.org/>