浙江大学计算机学院

Java 程序设计课程报告

2017–2018 学年秋冬学期

题目	基于爬虫的搜索引擎
学号	3150101155
学生姓名	余锦成
所在专业	计算机科学与技术
所在班级	计科 1504

目录

1 引言 **1**

1.1 设计目的 1

1.2 设计说明 1

2 总体设计 **2**

2.1 功能模块设计 2

2.2 流程图设计 3

3 详细设计 **3**

3.1 网址集合类设计 3

3.2 爬虫类设计 5

3.3 搜索引擎类设计 9

3.4 正文提取类设计 11

3.5 线程同步类设计 13

4 测试与运行 **14**

4.1 程序测试 14

4.2 程序运行 14

5 总结 **16**

1 引言

本次开发的是一个爬虫工具，是一个综合性的题目。要求能够对某个网站进行爬虫，然后提取其中的正文和标题，最后作出一个对爬虫结果进行搜索的功能。

1.1 设计目的

爬虫是一种很常见的技术，能够获取某个网站的大量特定信息，可能是用户资料、也可能是文章内容，当然也可能是整个网站的所有文本信息。本次实验使用 Java 语言编写一个基于爬虫搜索引擎，能够搜索网页中的内容。具体功能如下：

(1) 爬取浙江大学计算机学院网站`http://www.cs.zju.edu.cn`，并筛选其中的内容。

(2) 对筛选后的内容进行分词分析，建立倒排索引储存到磁盘上。

(3) 如果磁盘上保存有已经爬取过的内容，可以直接进行搜索（即爬取过之后就无须重新爬取了）。

(4) 对搜索内容进行显示，匹配搜索内容的进行标记，在显示匹配内容附近的部分内容（不进行全文显示），对大量的搜索结果进行分页查看。

(5) 搜索内容必须完全满足用户关键字，结果不会不包含搜索内容的某个字或词。

(6) 分页显示时候，用户可以进行上下翻页，查看当前页码，总共匹配项。

(7) 用户搜索完成之后，可以退出搜索结果显示，进行下一次搜索，或者退出程序。

1.2 设计说明

本程序使用 Java 程序开发，使用的 IDE 为 IntelliJ IDEA，是目前最受 Java 程序员喜欢的 Java IDE。

程序只有没有图形界面，只有命令行界面。搜索出来的链接在普通命令行是无法点击的（在 IDEA 的控制台中可以点击，为 IDEA 的特性）。匹配内容使用的是“【【keywords】】”进行标记。当然，如果作为后端输出到网页中将会有更好的展示效果。

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) 递归爬去网站并且提取标题和正文内容。

(2) 分析分词内容，建立倒排索引储存到磁盘。

(3) 在爬取完成后或者已经爬取过之后，用户可以输入关键字搜索内容。

(4) 搜索内容分页展示，用户可以上下翻页，或者退出展示页面继续搜索。

程序的总体功能如图 1所示。

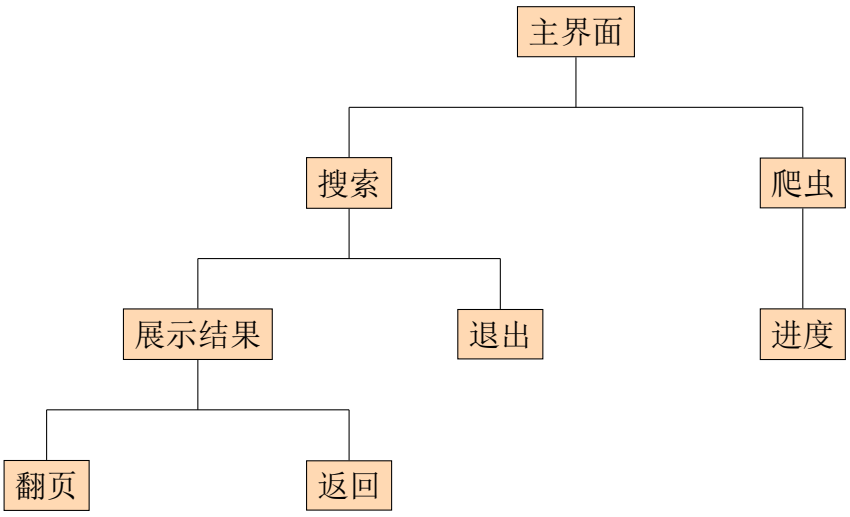

图 1 总体功能图

2.2 流程图设计

程序总体流程图如图 2所示。

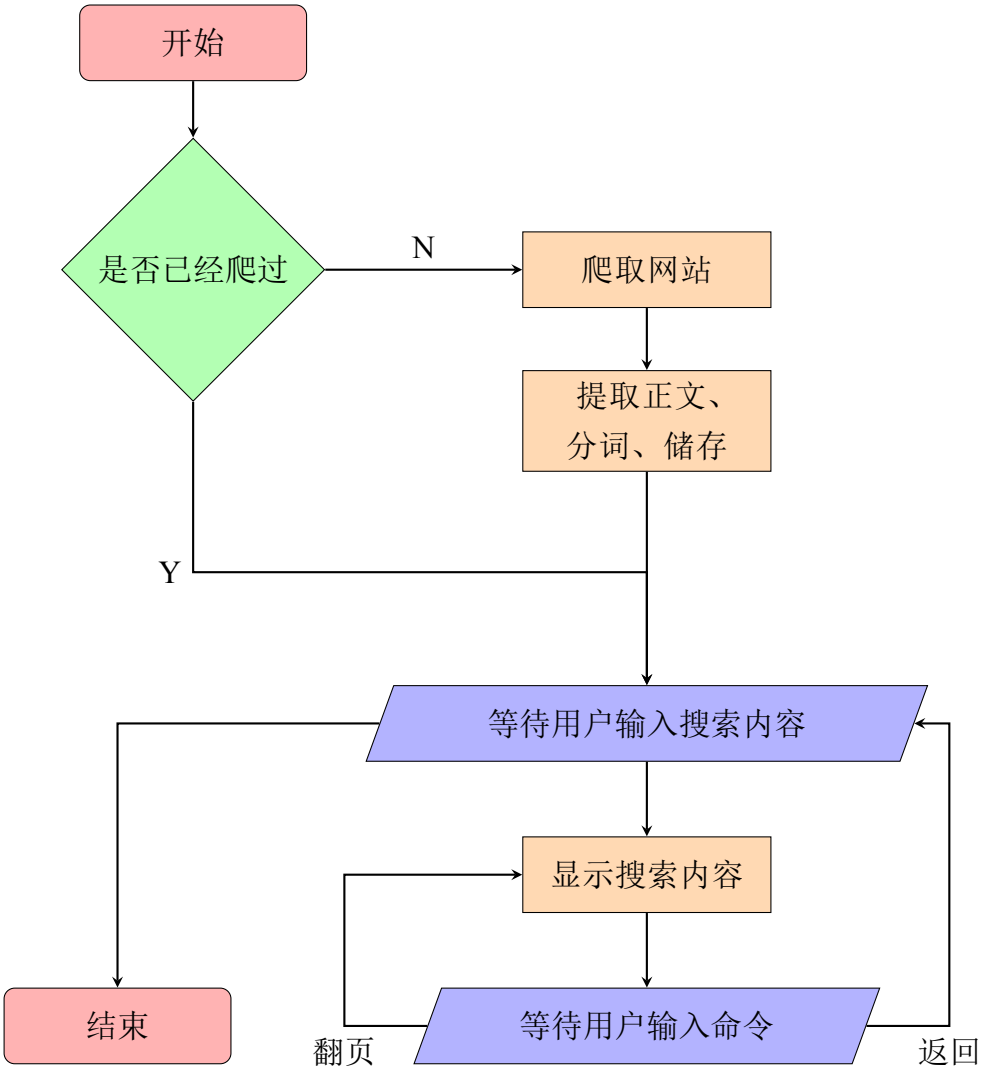

图 2 总体流程图

3 详细设计

3.1 网址集合类设计

网址集合类`class UrlSet`有四个功能：

(1) 记录所有添加的网址。

(2) 记录待爬去的网网址。

(3) 添加需要抓取的网址，添加时候确保证网址没有被添加过。

(4) 获取一个需要抓取的网址，获取之后记录该网址，使用先进先出策略。

UrlSet的 UML 图如图 3所示。

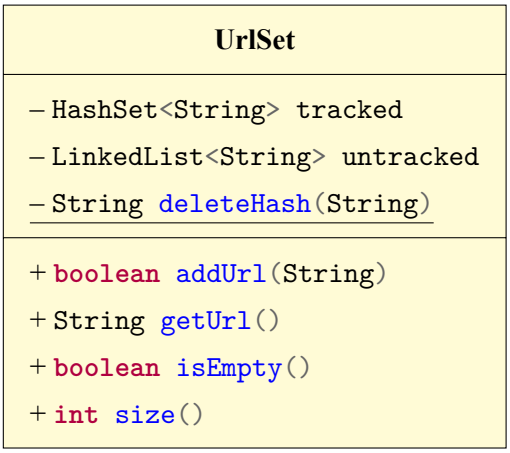

图 3　UrlSet类图

对于该类的简要说明：

(1) HashSet<String> tracked成员变量用于记录已经添加过的网址。

(2) LinkedList<String> untracked用于记录待抓取的网址。

(3) String deleteHash(String)函数用于删除网址中的锚。

(4) boolean addUrl(String)函数用于添加新的网址。

(5) String getUrl()函数用于获得一个没有被抓取的网址。

(6) boolean isEmpty()函数返回待抓取网址列表是否为空。

(7) int size()函数返回待抓取网址列表大小。

3.2 爬虫类设计

类class Crawler是程序中执行爬虫功能的抽象类。其主要功能包括：

(1) 设置搜索引擎和正文提取对象。

(2) 多线程调用虚函数crawl()进行爬取网页内容。

(3) 维护一个UrlSet对象。

(4) 提供接口给子类添加或者获取网址。

这个抽象类设计的目的是可以通过派生使不同类使用不同的引擎爬虫。本实验中使用 Jsoup 引擎抓包，实现了一个class JsoupCrawler类用于爬虫。

该类的 UML 图如图 4所示。

对于该类的简要说明：

(1) int THREAD_SIZE是静态变量，设置爬虫最大并发数（线程数）。

(2) SearchEngine SearchEngine是搜索引擎对象，在添加内容或者搜索时候

(3) String filter正则表达式，用于过滤网址。只有符合该正则表达式的网址才会被抓取。

(4) Extractor extractor正文提取器对象，用在爬到内容之后，进行正文提取，然后再把正文交给搜索引擎对象。

(5) UrlSet urlSet网址集合类，用于储存维护抓取网址。

(6) BlockThreadPoolExecutor executor线程池，用于储存爬虫线程。

(7) Semaphore semaphore线程同步条件变量，用于同步线程。

(8) void checkDependencies()私有函数检查依赖对象（搜索对象和正文提取器对象）是否已经设置。没有设置不能运行。

(9) void getUrl()私有函数用于从网址集合类中拿一个网址。

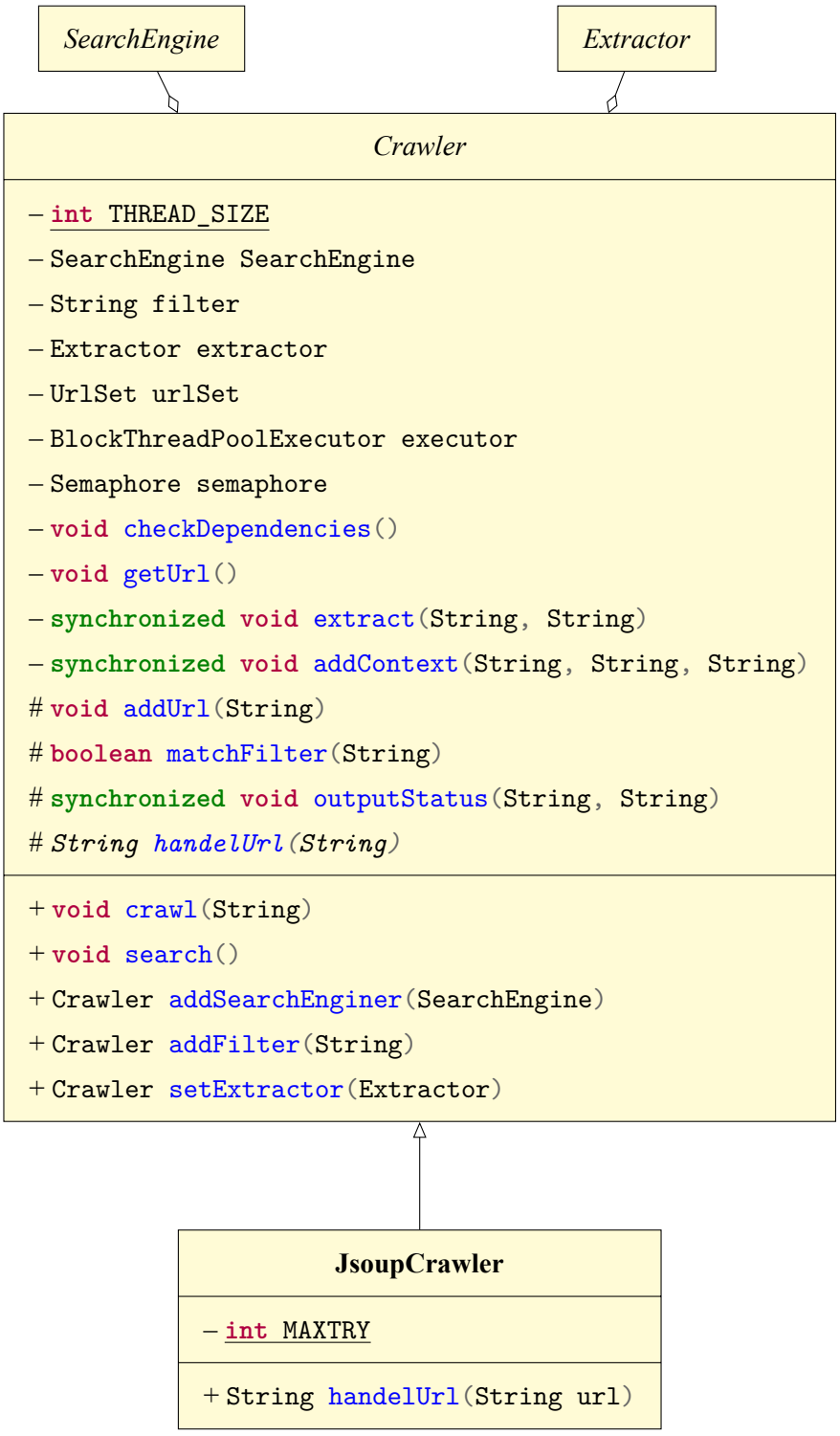

图 4　JsoupCrawler类图

(10) `synchronized void extract(String, String)`保护函数。用于正文提取。提供网址和内容，会调用正文提取器提取网页内容。网址也会提供，可以针对不同网址采取不同行为提取。需要线程安全，因为可能有很多个线程同时调用这个函数。

(11) `synchronized void addContext(String, String, String)`保护函数。把网址、正文和标题交给搜索引擎。同样需要线程安全。

(12) `void addUrl(String)`保护函数。用于给继承类添加网址用。会根据正则过滤网址。

(13) `boolean matchFilter(String)`保护函数，用于判断网址是否符合过滤正则表达式。

(14) `synchronized void outputStatus(String, String)`保护函数,同样线程安全，用于继承类输出抓包情况。

(15) `String handelUrl(String)`虚函数，给继承类实现，需要继承类实现获取网址的网页内容，并且添加网页中的所有超链接网址，最后返回网页内容。

(16) `void crawl(String)`公开函数。调用这个函数之后，对象会开始从参数提供的网址开始递归抓包。

(17) `void search()`公开函数。调用之后可以调用搜索引擎进行内容搜索。

(18) `Crawler addSearchEnginer(SearchEngine)`公开函数，用于添加储存抓包内容的搜索引擎。

(19) `Crawler addFilter(String)`公开函数。用于添加一个网址过滤正则。会替换原来的正则。

(20) `Crawler setExtractor(Extractor)`公开函数。用于设置一个正文提取器。

爬虫类的执行流程图如图 5所示。

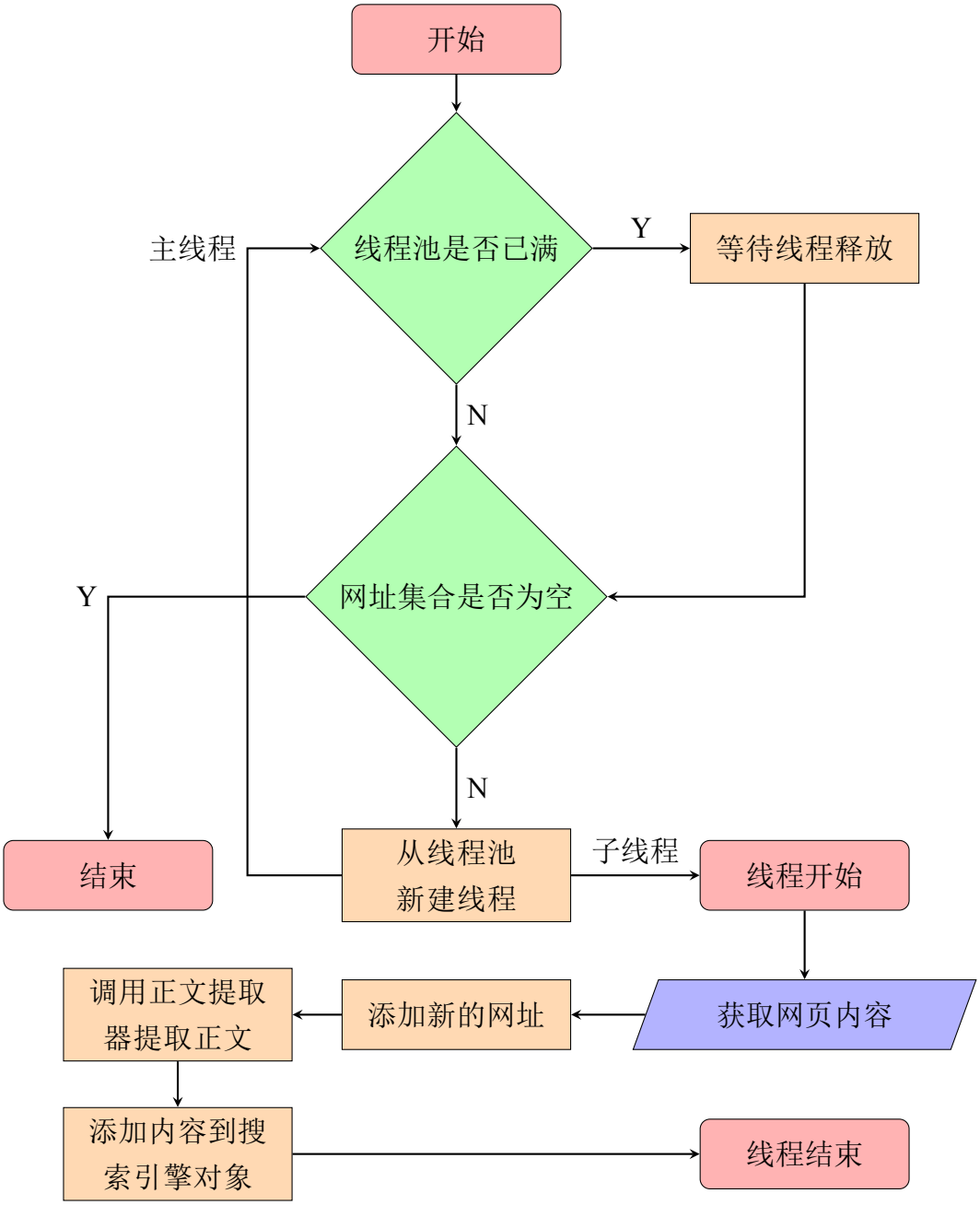

图 5　Crawler类流程图

3.3 搜索引擎类设计

搜索引擎类interface SearchEngine其实是一个接口。该接口实现以下功能：

(1) 添加新内容，并创建倒排索引储存到硬盘。

(2) 判断是否已经有索引。

(3) 与用户进行互交搜索。

本项目中，class LuceneEngine实现了这个接口。并且用户互交中提供翻页查看搜索功能。

其 UML 图如图 6所示。

LuceneEngine类的简要说明：

(1) int FRAGMENT_SIZE静态变量。表示输出结果摘要有多少个字符。

(2) Formatter formatter私有变量。用于最后高亮搜索结果的关键字。

(3) File indexFile私有变量，打开的是存放倒排索引的文件夹。

(4) boolean exits私有变量，用来记录是否已经生成倒排索引。

(5) IndexWriter indexWriter私有变量。用于写入索引。

(6) Analyzer analyzer私有变量。分词器。

(7) void addDocToPage(LuceneSearchPage,IndexSearcher, ScoreDoc, Query) 私有函数，用于把一个搜索结果放到页面中。

(8) void addContext(String, String, String) 添加一个页面内容。

(9) void search() 启动搜索程序。

(10) boolean exists()返回是否存在倒排索引。

LuceneSearchPage类的简要说明：

这个类用于管理搜索结果页面。

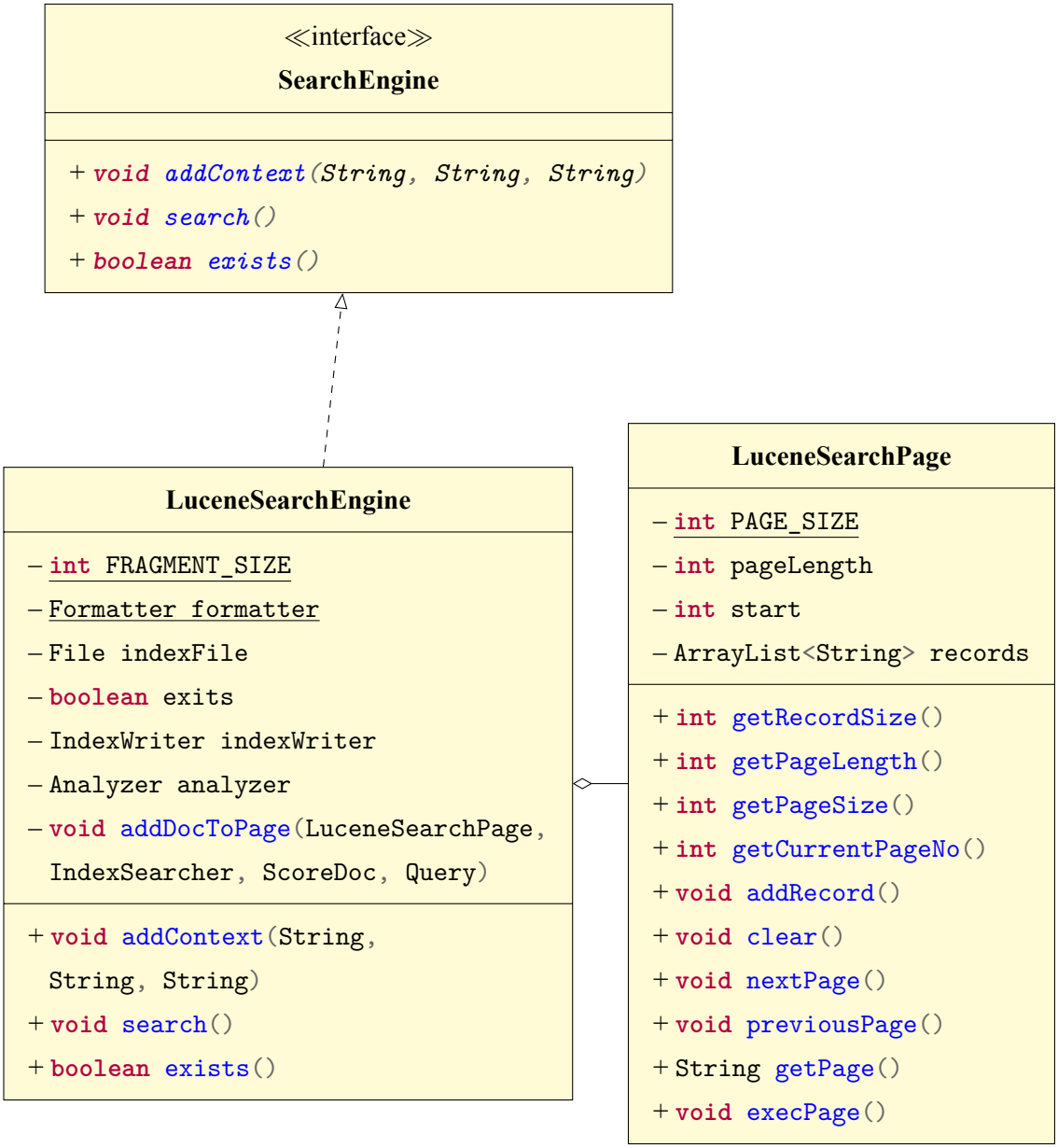

图 6　LuceneEngine类图

(1) int PAGE_SIZE静态私有成员。定义默认一个页面显示多少条记录。

(2) int pageLength私有成员。定义一个页面显示多少条记录。

(3) int start私有成员。当前页面第一条记录的序号。

(4) ArrayList<String> record私有成员。所有记录的列表。

(5) int getRecordSize()获取一共有多少条记录。

(6) int getPageLength()获取一个页面有多少条记录。

(7) int getPageSize()获取一共有多少个页面。

(8) int getCurrentPageNo()获取当前页面编号。

(9) void addRecord()添加一条记录。

(10) void clear()清空所有记录。

(11) void nextPage()跳转到下一个页面（如果存在）。

(12) void previousPage()跳转到前一个页面（如果存在）。

(13) String getPage()获取当前页面的内容。

(14) void execPage()执行页面。可以提供用户翻页的互交。

3.4 正文提取类设计

正文提取类interface Extractor其实也是一个接口。该接口实现一下功能：

(1) 添加新内容，根据网址进行正文提取。

(2) 添加内容后，获得标题。

(3) 添加内容后，获得正文。

实现这个比较接口的是一个匿名类。专门用于提取学院网站的内容。由于不同网页有不同的提取方式，这里不再赘述。提取方式是根据网址使用 CSS 选择器进行正文提取。

该匿名类的 UML 图如图 7所示。

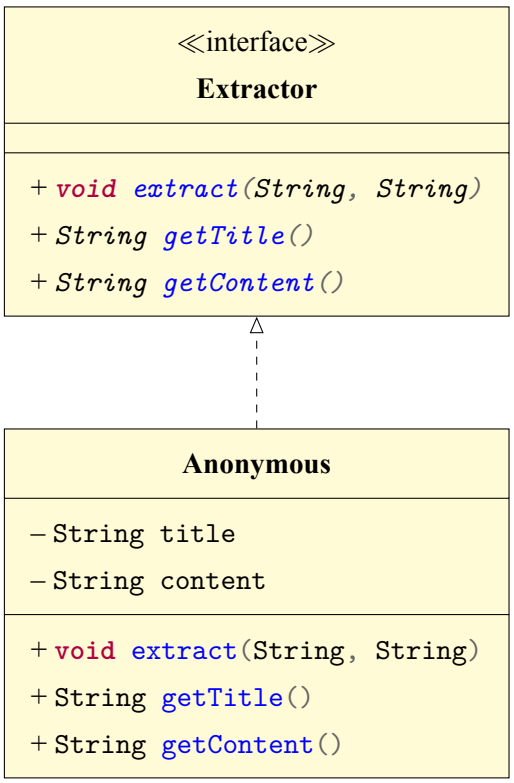

图 7　Extractor类图

以下是该匿名类的简要说明：

(1) `String title`私有变量。储存提取出来的标题。

(2) `String content`私有变量。储存提取出来的正文。

(3) `void extract(String, String)`提取函数。调用时候提供网址和内容。

(4) `String getTitle()`在调用提取函数后，用于获得提取后的标题。

(5) `String getContent()`在调用提取函数后，用于获得提取后的正文。

3.5 线程同步类设计

由于本项目使用了多线程设计，所以还设计一个线程同步的问题。故设计了两个相关类用于线程同步，分别为`class CountLatch`和`class BlockThreadPoolExecutor`。

它们的 UML 图如图 8所示。类CountLatch的简要说明：

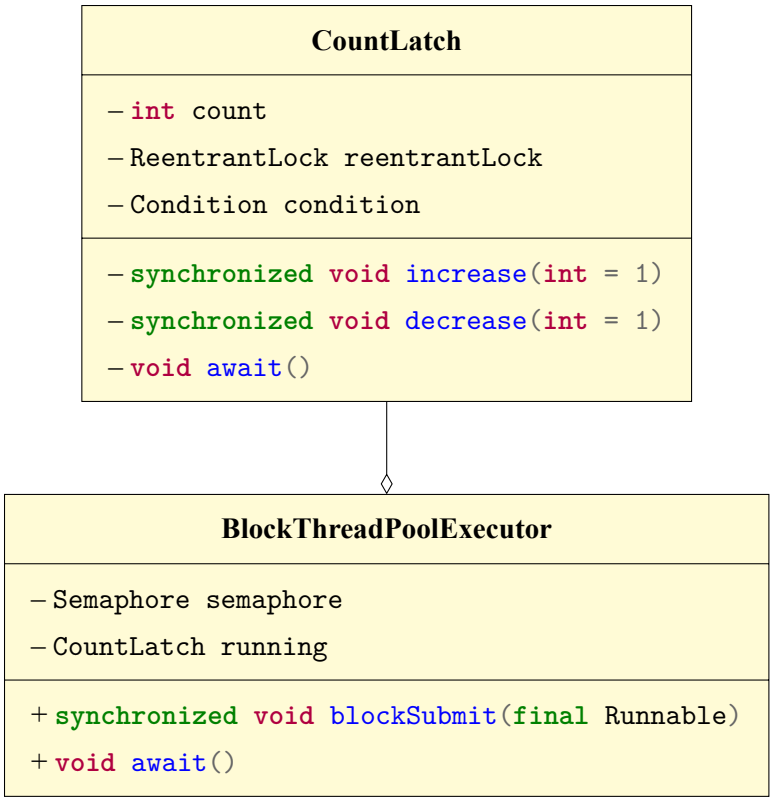

图 8　线程同步相关类类图

该类是用与实现一个可递增递减的阻塞计数器。只有当计数器为 0 时，调用await函数的一个线程才会被唤醒。

(1) `int count`内部计数器。

(2) `ReentrantLock reentrantLock`互斥锁，用于保护条件变量。

(3) `Condition condition`条件变量，用于实现等待和唤醒。

(4) `synchronized void increase(int = 1)`递增计数器，默认递增 1。

(5) `synchronized void decrease(int = 1)`递减计数器，默认递减 1。

(6) `void await()`阻塞等待计数器。

类BlockThreadPoolExecutor的简要说明：

该类用于线程池中：当线程超过制定量之后，会阻塞线程直到线程池有空位之后再重新启动线程。这样可以限制我们爬虫的线程数。

(1) `Semaphore semaphore`同步信号量，用于限制线程数。

(2) `CountLatch running`阻塞计数器，用于等待所有线程完成。

(3) `synchronized void blockSubmit(final Runnable)`阻塞提交线程。当线程池可能超过限制量时，会阻塞调用线程。

(4) `void await()`等待所有线程退出。

4 测试与运行

4.1 程序测试

在程序代码基本完成后，经过不断调试和完善，程序能够正常符合需求运行。

基本功能完善，但是还缺少一些更友好的操作：比如图形界面、限制爬虫深度或者限制条目数、重新爬虫功能、检查索引有效性等。

4.2 程序运行

以下是程序运行的结果：

图 9是第一次打开程序时爬虫的截屏。中括号内的数字为剩余需要抓包的数目。

图 10是第一次搜索的截屏，搜索内容为我的名字“余锦成”。可以看到完全匹配搜索结果靠前，并且命中的关键字有标注。

图 11是翻页搜索截图。这时候可以看到页码增加了，并且内容也发生了改变。

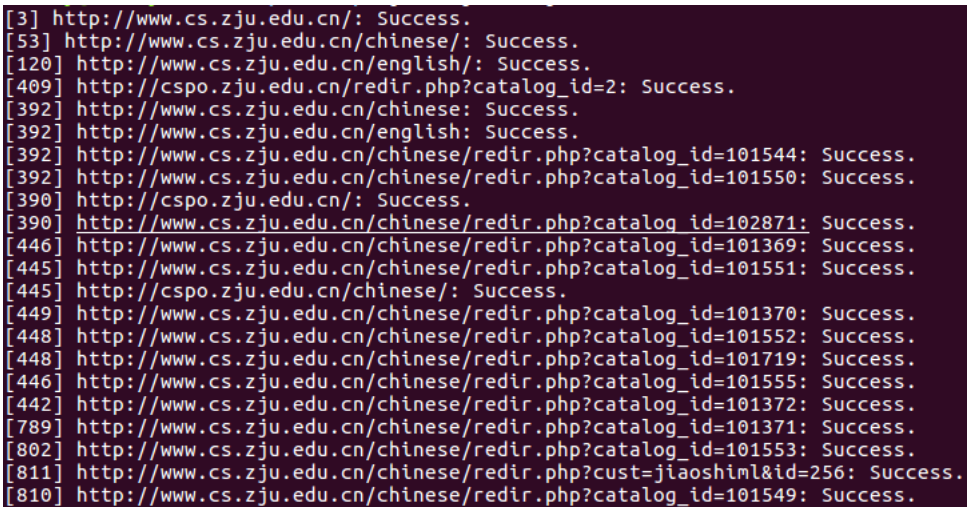
[3] http://www.cs.zju.edu.cn/: Success.
[53] http://www.cs.zju.edu.cn/chinese/: Success.
[120] http://www.cs.zju.edu.cn/english/: Success.
[409] http://cspo.zju.edu.cn/redir.php?catalog_id=2: Success.
[392] http://www.cs.zju.edu.cn/chinese: Success.
[392] http://www.cs.zju.edu.cn/english: Success.
[392] http://www.cs.zju.edu.cn/chinese/redir.php?catalog_id=101544: Success.
[392] http://www.cs.zju.edu.cn/chinese/redir.php?catalog_id=101550: Success.
[390] http://cspo.zju.edu.cn/: Success.
[390] http://www.cs.zju.edu.cn/chinese/redir.php?catalog_id=102871: Success.
[446] http://www.cs.zju.edu.cn/chinese/redir.php?catalog_id=101369: Success.
[445] http://www.cs.zju.edu.cn/chinese/redir.php?catalog_id=101551: Success.
[445] http://cspo.zju.edu.cn/chinese/: Success.
[449] http://www.cs.zju.edu.cn/chinese/redir.php?catalog_id=101370: Success.
[448] http://www.cs.zju.edu.cn/chinese/redir.php?catalog_id=101552: Success.
[448] http://www.cs.zju.edu.cn/chinese/redir.php?catalog_id=101719: Success.
[446] http://www.cs.zju.edu.cn/chinese/redir.php?catalog_id=101555: Success.
[442] http://www.cs.zju.edu.cn/chinese/redir.php?catalog_id=101372: Success.
[789] http://www.cs.zju.edu.cn/chinese/redir.php?catalog_id=101371: Success.
[802] http://www.cs.zju.edu.cn/chinese/redir.php?catalog_id=101553: Success.
[811] http://www.cs.zju.edu.cn/chinese/redir.php?cust=jiaoshiml&id=256: Success.
[810] http://www.cs.zju.edu.cn/chinese/redir.php?catalog_id=101549: Success.

图 9　爬虫

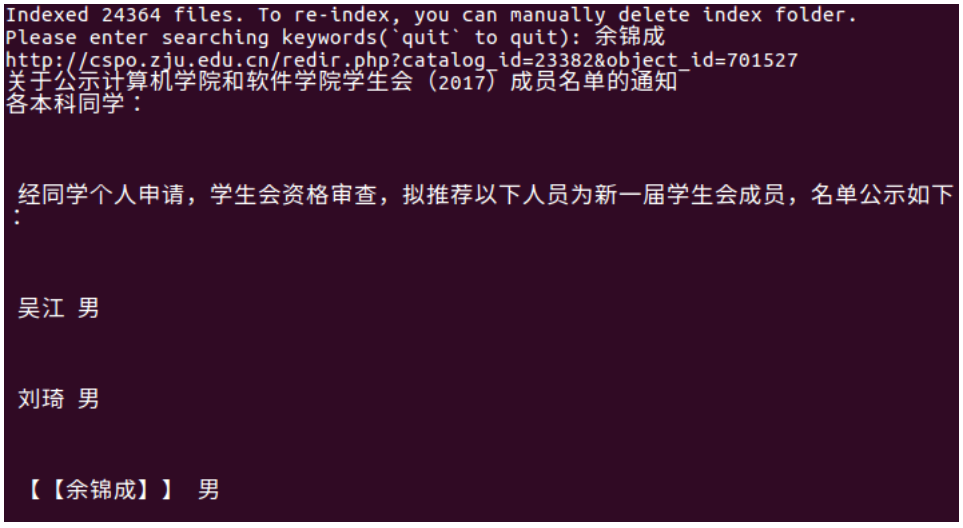
Indexed 24364 files. To re-index, you can manually delete index folder.
Please enter searching keywords(`quit` to quit): 余锦成
http://cspo.zju.edu.cn/redir.php?catalog_id=23382&object_id=701527
关于公示计算机学院和软件学院学生会（2017）成员名单的通知
各本科同学：
经同学个人申请，学生会资格审查，拟推荐以下人员为新一届学生会成员，名单公示如下：
吴江 男
刘琦 男
【【余锦成】】 男

图 10　搜索

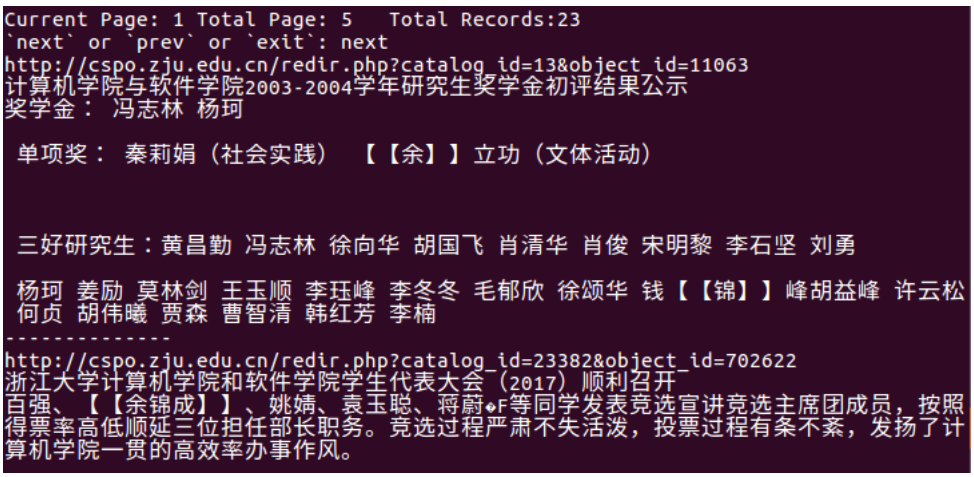

图 11　翻页

图 12是重新搜索截图。我们退出上次搜索结果，然后输入新的关键字，这时候又会出现新的页面的内容。

图 12　重新搜索

5　总结

这次作业综合性比较强。这次实验我们主要学习如何调用第三方库实现我们的项目，并且阅读其文档，并组装不同第三方库实现我们的需求。

除此之外，本次实验还让我们体会到 Java 接口、类设计的技巧，从第三方库的实现方式真切体会，到项目中我们自己实现等。正如本次实验中，我的爬虫类调用的实际是接口，并没有规定具体使用什么第三方库。因此，我的爬虫类在不同环境需求下有复用性。我们可以通过实现另外一个正文提取器可以来抓取不同的网站；可以通过修改搜索引擎类采用其他搜索引擎类。通过子类化爬虫类可以实现不同网络获取方式，比如我们可以改用 socket 来爬虫。

总而言之，本次实验是我对 Java 这门语言有了更深刻的理解，对面向接口编程有了更深刻的实践，为我以后学习和实验打下了基础。