Homework 3

- 搜索引擎 (Search Engine)
  - 目标：开发一个搜索引擎，能够处理多种格式的文档（如PDF、Word、HTML等），并允许用户根据自己的需求进行搜索。

Homework 3

- 结合Jsoup, apache tika, lucene等来搭建搜索引擎
  - JSOUP：Java HTML Parser来抽取信息 (如标题、网站内容等，相同的网站同一个模板)
  - Apache Tika库：用于从多种格式的文档中提取文本内容。
  - Lucene库：构建高效、可扩展的文档索引。

Homework 3

- Jsoup是一个Java库，用于解析、操作和清理HTML。它可以从URL、文件或字符串中加载HTML，然后使用其非常方便的DOM、CSS和类似jQuery的方法来提取和操作数据。Jsoup特别适合处理网页数据抓取、解析HTML文档中的信息（如标题、段落、链接等）。

```
File input = new File("/tmp/input.html");
Document doc = Jsoup.parse(input, "UTF-8", "http://example.com/");

Elements links = doc.select("a[href]"); // a with href
Elements pngs = doc.select("img[src$=.png]");
  // img with src ending .png

Element masthead = doc.select("div.masthead").first();
  // div with class=masthead

Elements resultLinks = doc.select("h3.r > a"); // direct a after h3
```

Homework 3

- Apache Tika 是一个开源的、跨平台的库，用于检测、提取和解析各种类型文件的元数据。它支持多种文件格式，包括文档、图片、音频和视频。
- Tika是一个底层库，经常用于搜索引擎、内容管理系统、数据分析任务等领域，无缝地集成到其他应用或服务中以增强对文件内容处理的能力。

```
public class LuceneIndexer {
    private final Tika tika;

    private final IndexWriter writer;

    public LuceneIndexer(Tika tika, IndexWriter writer) {
        this.tika = tika;
        this.writer = writer;
    }

    public void indexDocument(File file) throws Exception {
        Document document = new Document();
        document.add(new Field("filename", file.getName(), Store.YES, Index.ANALYZED));
        document.add(new Field("fulltext", tika.parseToString(file), Store.NO, Index.ANALYZED));
        writer.addDocument(document);
    }
}
```

Homework 3

Apache Lucene™ is a high-performance, full-featured text search engine library written entirely in Java. It is a technology suitable for nearly any application that requires full-text search, especially cross-platform.

Lucene Query Types

• Single Term VS. Multi-Term	"+name: camel + type: animal"
• Wildcard Queries	"text:wonder*"
• Fuzzy Queries	"room~0.8"
• Range Queries	"date:[25/5/2000 To *]"
• Grouped Queries	"text: animal AND small"
• Proximity Queries	"hamlet macbeth"~10
• Boosted Queries	"hamlet^5.0 AND macbeth"

http://lucene.apache.org/core/

Homework 3

API Sample I (Indexing)

```
private IndexWriter writer;
 public Indexer(String indexDir) throws IOException {
       Directory dir = FSDirectory.open(new File(indexDir));
       writer = new IndexWriter(dir, new StandardAnalyzer(Version.LUCENE_CURRENT),  true,
                IndexWriter.MaxFieldLength.UNLIMITED);
 }

 public void close() throws IOException {
       writer.close();
 }

 public void index(String dataDir, FileFilter filter) throws Exception {
      File[] files = new File(dataDir).listFiles();
      for (File f: files) {
            Document doc = new Document();
            doc.add(new Field("contents", new FileReader(f)));
            doc.add(new Field("filename", f.getName(), Field.Store.YES, Field.Index.NOT_ANALYZED));
            writer.addDocument(doc);
       }
}
```

Homework 3

API Sample II (Searching)

```
public void search(String indexDir, String q) throws IOException, ParseException {
        Directory dir = FSDirectory.open(new File(indexDir));
        IndexSearcher is = new IndexSearcher(dir, true);

        QueryParser parser = new QueryParser("contents",
                                  new  StandardAnalyzer(Version.LUCENE_CURRENT));
        Query query = parser.parse(q);
        TopDocs hits = is.search(query, 10);
        System.err.println("Found " + hits.totalHits + " document(s)");

        for (int i=0; i<hits.scoreDocs.length; i++) {
           ScoreDoc scoreDoc = hits.scoreDocs[i];
           Document doc = is.doc(scoreDoc.doc);
           System.out.println(doc.get("filename"));
        }

        is.close();
}
```

Homework 3

API Sample III (Deleting)

Via **IndexReader**

void **deleteDocument(int docNum)**
Deletes the document numbered docNum

int **deleteDocuments(Term term)**
Deletes all documents that have a given term indexed.

Via **IndexWriter**

void **deleteAll()**
Delete all documents in the index.

void **deleteDocuments(Query query)**
Deletes the document(s) matching the provided query.

void **deleteDocuments(Query[] queries)**
Deletes the document(s) matching any of the provided queries.

void **deleteDocuments(Term term)**
Deletes the document(s) containing term.

void **deleteDocuments(Term[] terms)**
Deletes the document(s) containing any of the terms.

Homework 3

- 代码要求：
  - 遵守编程规范，如命名、注释等规范
  - 遵守面向对象的设计原则
  - 考虑异常处理等应用

Homework 3

- 文档要求：
  - 按附件格式样例，至少包括：引用、总体设计、详细设计、测试与运行、总结
  - 包括：数据格式说明

Homework 3

- 作业包括：java文件 + 文档 + 数据
- 作业打包上传到ftp homework/homework3下
- 文件：学号_姓名_homework3.rar