

# Java Homework3

---

蔡佳伟 3220104519

## 一、引用

---

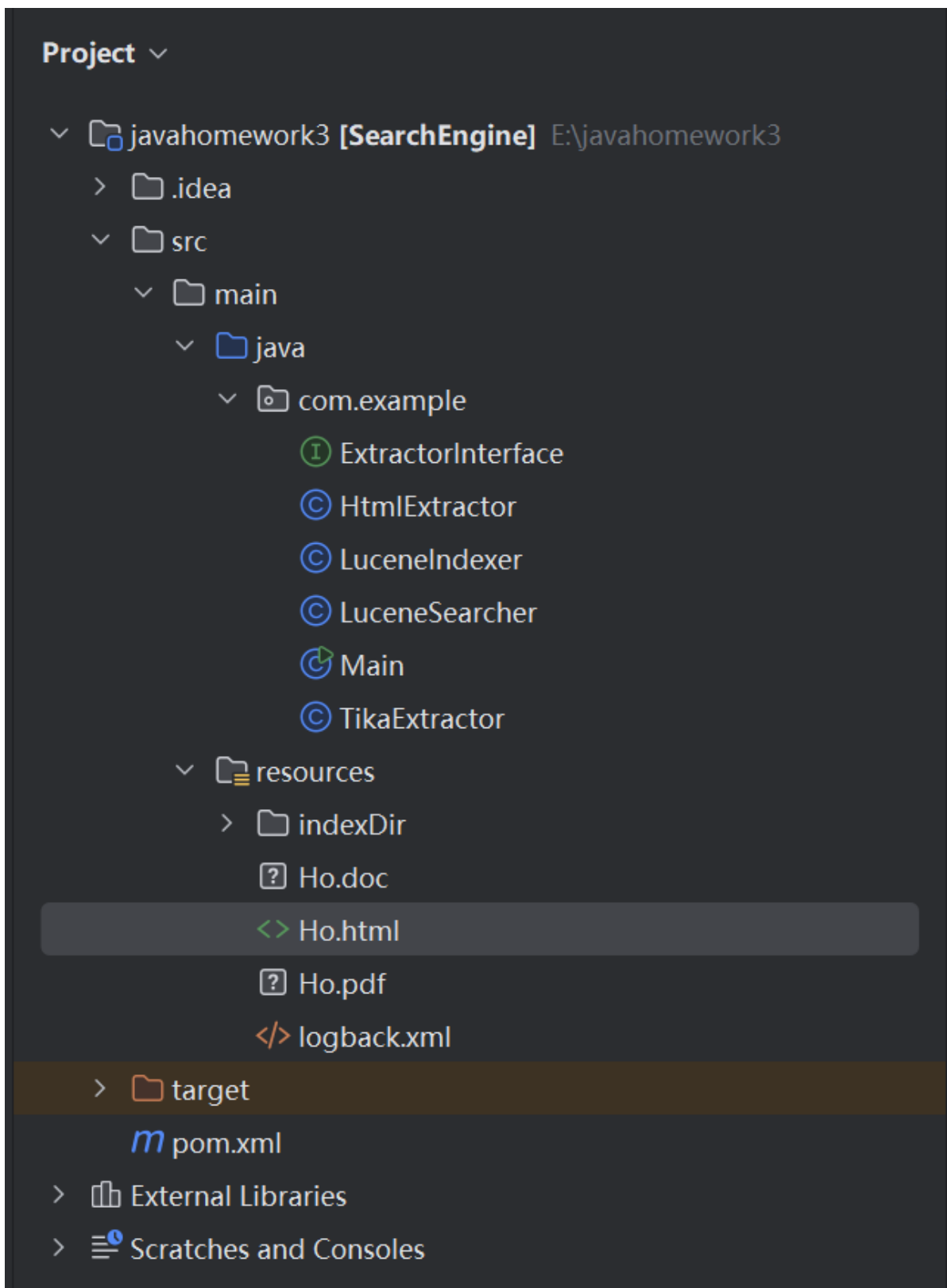
本次作业我采用了作业要求中推荐的Jsoup, apache tika, lucene。其中Jsoup是一个Java库, 用于解析、操作和清理HTML。它可以从URL、文件或字符串中加载HTML, 然后使用其非常方便的DOM、CSS和类似 jQuery的方法来提取和操作数据。Jsoup特别适合处理网页数据抓取、解析HTML文档中的信息(如标题、段落、链接等)。Apache Tika 是一个开源的、跨平台的库, 用于检测、提取和解析各种类型文件的元数据。它支持多种文件格式, 包括文档、图片、音频和视频。Tika是一个底层库, 经常用于搜索引擎、内容管理系统、数据分析任务等领域, 无缝地集成到其他应用或服务中以增强对文件内容处理的能力。Apache LuceneTM 是一个完全用 Java 编写的高性能、功能齐全的文本搜索引擎库。该技术几乎适用于任何需要全文搜索的应用程序, 尤其是跨平台应用程序。

## 二、总体设计

---

本次作业我采用了Maven, 这是一个广泛使用的构建自动化工具。

采用Maven的原因主要是对于Apache Lucene等外部库, Maven会自动下载添加到项目中, 只需要在pom.xml中声明一个依赖, Maven会自动下载需要的库并确保版本兼容性。项目的文件结构组成大概是这样:



## 三、详细设计

### Main.java

是整个程序的入口，在这里用户启动整个索引和搜索过程。我在Main.java中初始化LuceneIndexer和LuceneSearcher，并通过TikaExtractor或HtmlExtractor提取文档内容并创建索引，还负责管理用户的输入，并调用LuceneSearcher进行搜索，搜索我设计的是直接在文件中更改，如果需要输入也可以。

### LuceneIndexer.java

这个文件负责将文件内容（文本）索引到Lucene索引中，采用CreateIndex来添加索引，close来关闭。索引包括文件路径和文件内容，方便后续进行全文搜索。

## LuceneSearcher.java

这个文件负责根据我输入的查询词，搜索已经建立的Lucene索引，使用IndexSearcher来执行查询，并返回匹配的文档路径和内容。search方法解析查询字符串，并在索引中搜索，返回匹配结果。

## ExtractorInterface.java

这个文件是一个接口，定义了提取文件内容的方法，支持不同格式的文件提取器。里面的实现类如HtmlExtractor和TikaExtractor将实现该接口来提取不同格式文件的文本内容。

## HtmlExtractor.java

这个文件实现了ExtractorInterface的一个具体类，专门用于提取HTML文件的内容，使用Jsoup库。

## TikaExtractor.java

这个文件也实现了ExtractorInterface的一个具体类，使用Apache Tika来提取各种文件，自动识别文件类型并使用相应的解析器提取文本内容。

Main.java调用 TikaExtractor或 HtmlExtractor 提取文件内容。

Main.java调用LuceneIndexer创建索引。

Main.java调用 LuceneSearcher进行查询。

## 四、测试与运行

先配置好Maven环境后（我是运用的IDEA），运行Main.java

```
1 package com.example;
2
3 import java.io.File;
4
5 public class Main {
6
7     public static void main(String[] args) throws Exception {
8         String filePath = "E:/javahomework3/src/main/resources/Ho.pdf";
9         String indexDir = "E:/javahomework3/src/main/resources/indexDir";
10
11         File file = new File(filePath);
12
13         // 检查文件是否存在，使用标准输出
14         if (!file.exists()) {
15             System.err.println("File does not exist: " + filePath); // 错误信息输出到控制台
16             return;
17         }
18     }
19 }
```

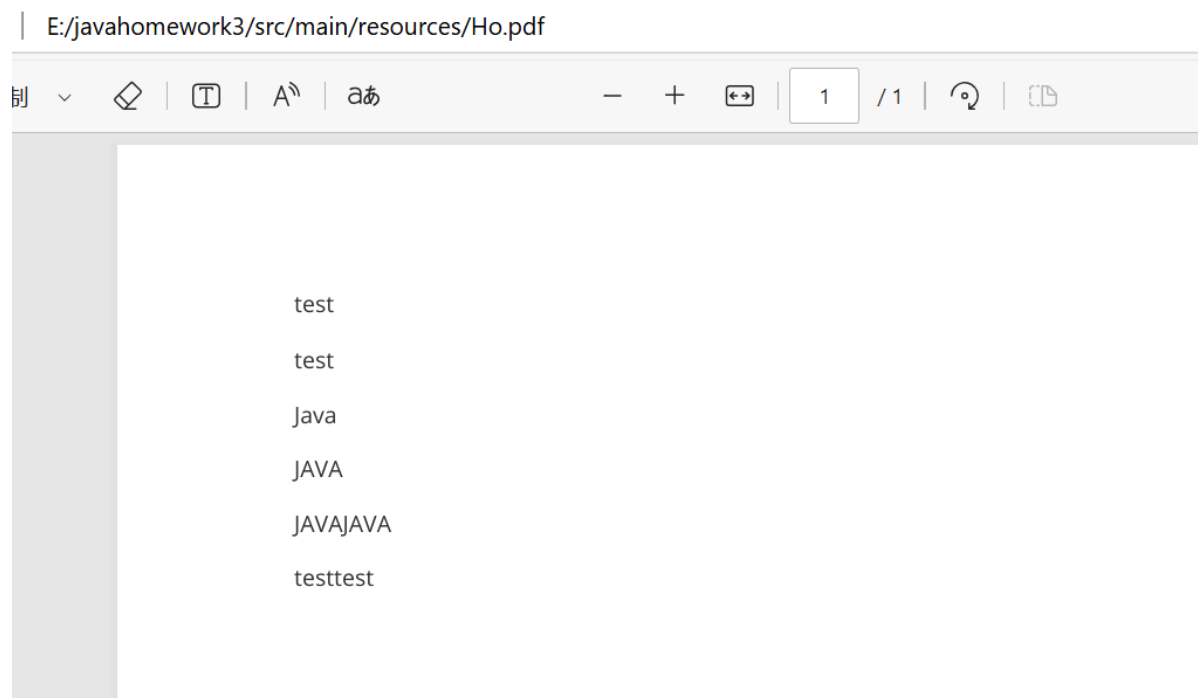
```
LuceneSearcher searcher = new LuceneSearcher(indexDir);
searcher.search(queryStr: "JAVA", maxHits: 10);

// 搜索完成后，输出日志
System.out.println("Search operation completed.");
}
}
```

先搜索的是pdf文档

Document 11:  
Path: E:/javahomework3/src/main/resources/Ho.pdf  
Content:  
test  
  
test  
  
Java  
  
JAVA  
  
JAVAJAVA  
  
testtest

可以看到输出了索引



正确输出了三个查找到的'JAVA'

```
Performing search for query
Searching for: JAVA
Found 3 result(s).
File: E:/javahomework3/src/main/resources/Ho.pdf
Score: 0.08345711
File: E:/javahomework3/src/main/resources/Ho.pdf
Score: 0.08345711
File: E:/javahomework3/src/main/resources/Ho.pdf
Score: 0.08345711
Search operation completed.
```

换成搜索.doc格式文件后也输出了索引

```
└─┬─┘
    Test ↵
    testJAVA ↵
    JAVA ↵
    Java ↵
    java ↵
    Javajava ↵
    JAVA ↵
    ↵
    ↵
```

```
Document 21:
Path: E:/javahomework3/src/main/resources/Ho.doc
Content: Test
testJAVA
JAVA
Java
java

Javajava

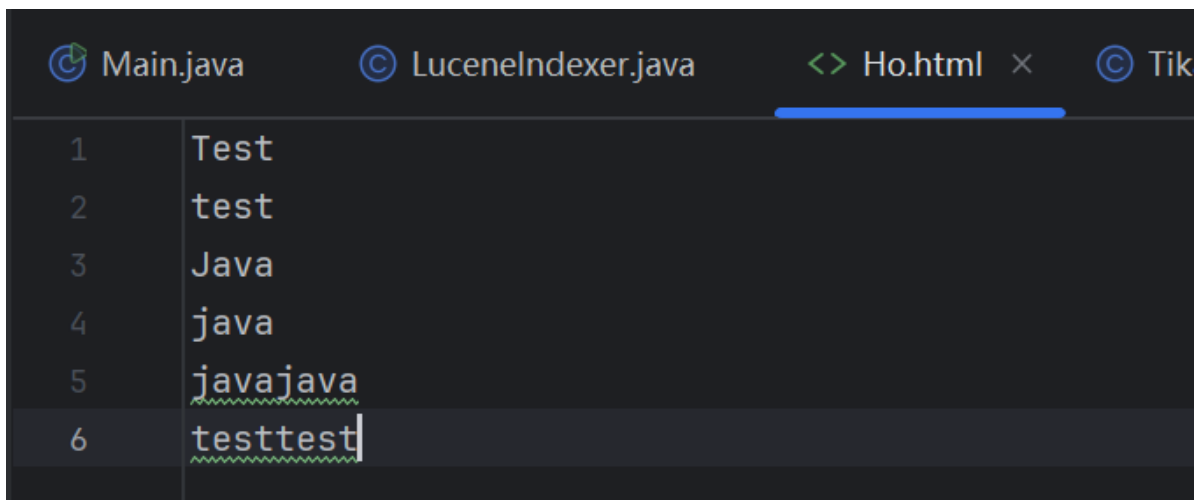
JAVA
```

也找到了'JAVA'

```
File: E:/javahomework3/src/main/resources/Ho.doc
Score: 0.056257147
File: E:/javahomework3/src/main/resources/Ho.doc
Score: 0.056257147
File: E:/javahomework3/src/main/resources/Ho.doc
Score: 0.056257147
```

换成.html文件后:

索引如下:



The screenshot shows an IDE with three tabs: 'Main.java', 'LuceneIndexer.java', and 'Ho.html'. The 'Ho.html' tab is active and shows a list of words: 'Test', 'test', 'Java', 'java', 'javajava', and 'testtest'. Below the code editor, a terminal window displays the output of a search operation, showing the path to the file and the words found.

```
-----
Document 26:
Path: E:/javahomework3/src/main/resources/Ho.html
Content: Test
test
Java
java
javajava
testtest
```

同样正确找到

```
File: E:/javahomework3/src/main/resources/Ho.html
Score: 0.029469693
Search operation completed.
```

说明搜索引擎运行正确无误。

## 五、总结

这次开发过程困难重重，首先我完全不知道该如何下手，其次在我了解到使用Maven后，就开始尝试文件的撰写，经过网上的查询和自己的学习，初步完成了文件的组织和具体内容设计。

在初步完成后，配置环境问题也很大。我先采用了Vscode，但不知为何一直无法正常使用，一直提示我文件系统的问题，我还把所有Path库都改成了File相关，但是还是不行。网上也没有找到有用的解决方法。

然后我换成了IDEA，不知道为什么就不是这个问题了，变成提示日志文件有问题，因为Apache库不能没有日志，我配置了日志文件，然后又提示tika版本有问题，本来是2.6.0，一直无法正常运行，在网上查询后，尝试降成了1.24，使用install更新了版本库，然后可以正常运行了。

但是目测这总算实现了一个能用的搜索引擎，由于时间紧张，可能文档略有草率，求求鲁老师和助教gg手下留情tut。

数据在src/main/resources目录下，就是测试用的各个格式的文档。