

# 浙江大学计算机科学与技术学院

## Java 应用技术课程报告

2020—2021 学年 秋冬学期

题目	简易书籍搜索引擎的设计与开发
学号	3180103772
学生姓名	张溢弛
所在专业	软件工程
所在班级	软件工程 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 .....	6
3.1 爬虫模块基本设计 .....	6
3.1.1 总体框架.....	6
3.1.2 网页的爬取和解析 .....	8
3.1.3 分页的处理.....	9
3.1.4 书籍信息的提取 .....	10
3.1.5 书籍信息的结构化保存.....	13
3.2 搜索模块的基本设计 .....	17
3.2.1 索引的建立.....	17
3.2.2 搜索引擎的运行 .....	20
4.测试与运行 .....	22
4.1 程序测试 .....	22
4.2 程序运行 .....	22
5. 总结.....	25

# 1.引言

本次课程作业开发的是一个基于 Java 的简易 Web 搜索引擎，该搜索引擎分为两个子模块，即爬虫模块和搜索模块，爬虫模块负责爬取“当当网”网站中的书籍数据，保存在文件中，而搜索模块则基于 Lucene 包实现索引的建立和查询，实现简单的离线搜索引擎。

本次实验中使用了较多外置 jar 包，主要有 Jsoup,HtmlUnit 和 Lucene 等。

## 1.1 设计目的

基于 Java 网络爬虫的 Web 搜索引擎的具体功能和设计思路有如下几个关键点：

- 搜索引擎本质上是爬虫模块和搜索模块的结合
- 爬虫模块基于 HtmlUnit 库对当当网网页中的数据信息进行爬取，并使用 Jsoup 对爬取到的 HTML 文件进行解析，提取相关的重要信息，比如书名、作者、出版社、目录、简介、作者介绍等等，并以文本文件的形式保存在本地
- 搜索模块使用导入的 Lucene 包进行索引建立和信息检索，把存储在本地的结构化的书籍信息，转化成便于处理的 document 对象，并建立索引，进行关键字的检索，与用户通过命令行进行交互
- 本次作业中，我们将两个模块进行了解耦，即爬虫模块和搜索模块分别作为两个独立的模块运行，因为网络爬虫需要爬取的数据量比较大，不能在短时间内获取所有的书籍信息，也因为当当网有一定的反爬虫机制存在，因此我们必须控制爬虫的频率和速度，否则就会导致 IP 被封禁一段时间，而老师的要求也是搜索引擎仅在本地数据中进行搜索，不需要在每次运行都去爬取新的内容，因此我将这两个模块拆分成了两部分来写，这也提高了程序的开发和运行效率。
- 具体的设计会在后面进一步阐述

## 1.2 设计说明

本程序采用 Java 程序设计语言，在 IDEA 平台下编辑、编译与调试。具体程序由我一人开发而成。具体工作如表 1 所示：

表 1 各成员分工表

成员名称	完成的主要工作	
	程序设计	课程报告
张溢弛	程序前期的需求分析和整体功能的架构，后期测试和运行	全部由张溢弛一人完成，或许这张表格应该取消
张溢弛	爬虫模块的设计、编写和测试	
张溢弛	搜索模块的设计、编写和测试	

## 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) 使用 HtmlUnit 爬取当当网中的书本信息，并解析动态 JavaScript 中的数据，然后用 Jsoup 来解析爬取到的 HTML 文件，提取关键的信息
- (2) 一定的反反爬虫策略
- (3) 把文档信息结构化地用文本文件的形式存储在本地
- (4) 读取结构化的信息，转化成 Document 对象并建立索引
- (5) 建立索引之后与用户进行交互，来

程序的总体功能如图 1 所示：

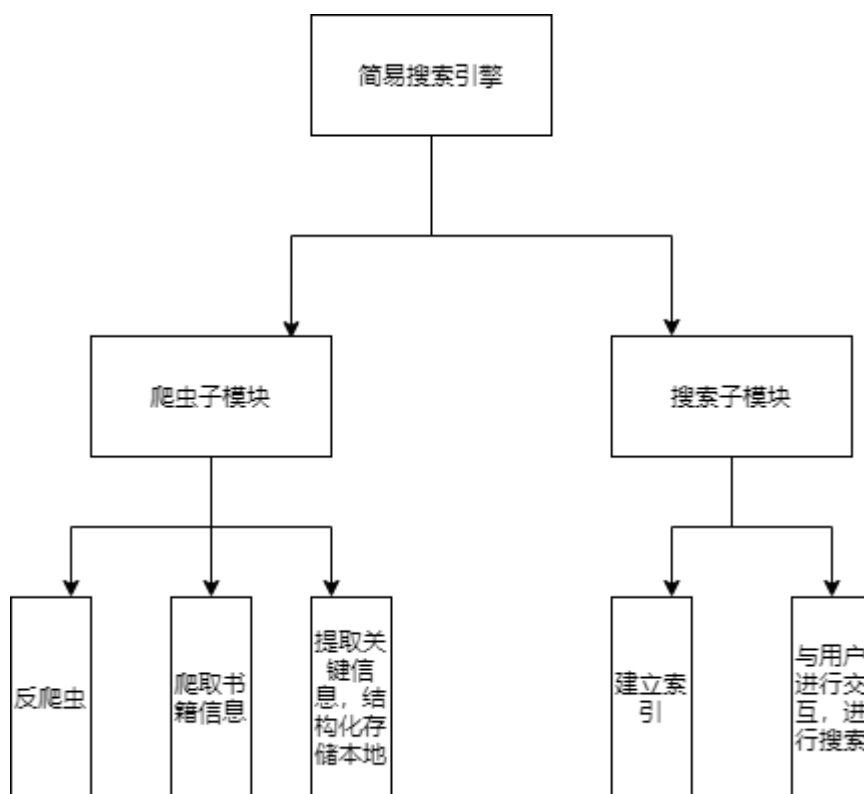


图 1 总体功能图

## 2.2 流程图设计

程序总体流程如图 2 所示：

爬虫模块的总体流程图

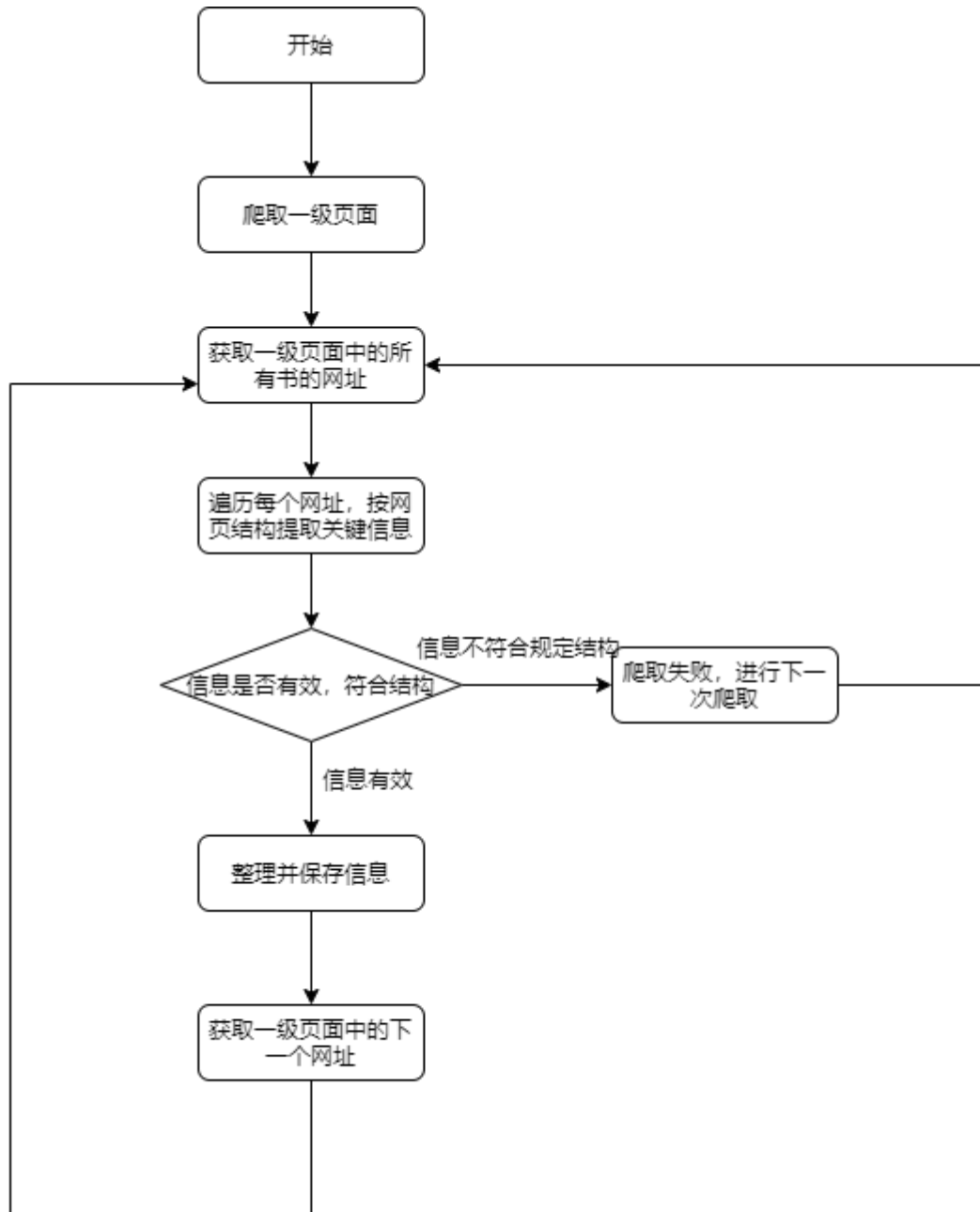
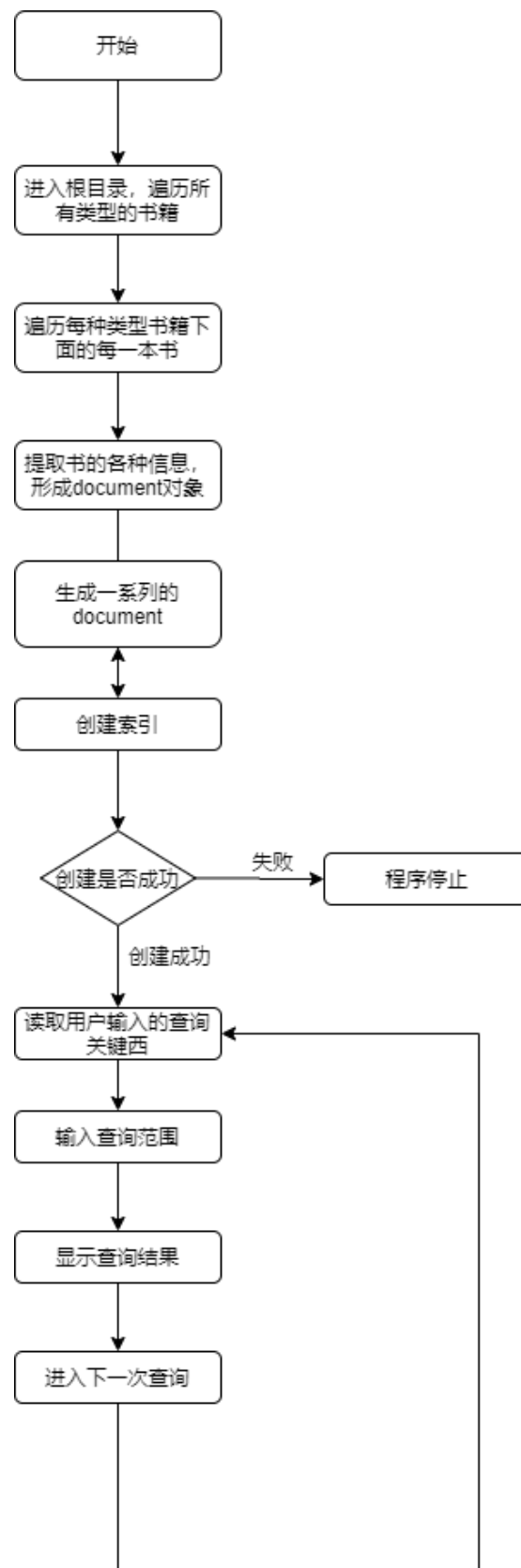


图 2 总体流程图

搜索模块的总体流程如下图所示



## 3.详细设计

### 3.1 爬虫模块基本设计

#### 3.1.1 总体框架

系统的爬虫模块的基本设计逻辑比较清楚明白，我们先来观察一下当当网的网页结构：



我们发现在当当网的搜索引擎中搜索关键字的时候（比如 java），其对应的 URL 就会变成 <http://search.dangdang.com/?key=java&act=input>，其中 key=java 表明 java 是搜索的关键字，可以将其替换成任何的内容，就能生成一个新的 URL，访问这个 URL 就可以获得对应的搜索结果。

而这样的每一个网页，他的书籍展示的页面如下图所示：





我们进入开发者模式可以观察到网页的 HTML 结构如下所示：

```
<a title=" 疯狂Java讲义 (第5版) (含DVD光盘一张)" ddclick="act=normalResult_picture&pos=27858952_0_1_q" class="pic" name="itemlist-picture" dd_name="单品图片" href="http://product.dangdang.com/27858952.html" target="_blank">
  
  <p class="cool_label"></p>
</a>
<p class="name" name="title">
  <a title=" 疯狂Java讲义 (第5版) (含DVD光盘一张) 覆盖Java稳定版本Java11, 渗透Java编程思想, 李刚作品成为海峡两岸读者之选, 本书赠送20+小时视频、源代码、课件、面试题, 提供微信答疑群, 配套学习网站。" href="http://product.dangdang.com/27858952.html" ddclick="act=normalResult_title&pos=27858952_0_1_q" name="itemlist-title" dd_name="单品标题" target="_blank"></a>
</p>
```

我们发现一级搜索页面的每本书的信息被存储在一个<a>标签中，而这个标签的类名叫做 pic，因此我们在爬取这个页面之后，再用 Jsoup 进行解析，提取出所有类名为 pic 的标签，然后再提取出所有的 href 属性，就可以获取当前搜索条件下的获得的书籍的对应页面，比如《疯狂 Java 讲义》对应的页面就是 <http://product.dangdang.com/27858952.html>，相关的代码如下所示：

```
WebCrawler webCrawler = new WebCrawler();
String[] searchTarget = webCrawler.getSearchTarget();
for (String topic: searchTarget) {
    String path = "./data/" + topic;
    File directory = new File(path);
    if (!directory.exists()) {
        try {
            directory.mkdir();
        } catch (Exception exception) {
            System.out.println(exception);
        }
    }
    String url = webCrawler.generateURL(topic);
    System.out.println("开始爬取新的图书主题: " + topic);
}
```

```

Document document = webCrawler.getSingleWebsite(url);
Elements hrefs = document.getElementsByClass("pic");
// 获得所有的子页面连接之后爬取子页面中的所有信息，并记录在文件中
for (int i = 0; i < hrefs.size(); i++) {
    String singleHref =
hrefs.get(i).attributes().get("href");
    webCrawler.getSingleBookInfo(singleHref, topic);
}
}

```

而其中的 `searchTarget` 是预先设定的一些搜索主题，我的这个简易图书搜索引擎主要用于搜索一些计算机相关的书籍，因此我定义了如下几种搜索的关键词：

```

public String[] getSearchTarget() {
    // 一些暂定的搜索目标
    String[] searchTarget = {"Java", "Python", "C++",
        "操作系统", "数据结构", "数据库",
        "计算机组成", "计算机网络", "软件工程", "机器学习"};
    return searchTarget;
}

```

获取这份关键词的列表之后，程序就开始遍历这个关键词列表，每次按照上面所述的规则组成新的 URL，并使用 `HtmlUnit` 和 `Jsoup` 进行爬取和解析这个页面中的可以获取到的所有书籍的 URL，并进行进一步的访问。

### 3.1.2 网页的爬取和解析

我们利用了 `HtmlUnit` 和 `Jsoup` 来进行单个网页的爬取，这一部分主要是调用库中现成的方法和 API，比较简单。`HtmlUnit` 相比于 `Jsoup` 的优势是可以解析出由 JavaScript 动态生成的数据，而 `Jsoup` 没有这一功能（这一点经过实践验证和网络上进行信息查询，都显示 `Jsoup` 不具备解析动态生成的数据的功能，而我们需要爬取的一些重要信息就是由 JavaScript 动态生成的），具体的代码如下：

```

public Document getSingleWebsite(String url) throws
IOException {

    LogFactory.getFactory().setAttribute("org.apache.commons.loggi
ng.Log","org.apache.commons.logging.impl.NoOpLog");

    java.util.logging.Logger.getLogger("com.gargoylesoftware").set
Level(Level.OFF);
}

```

```

java.util.logging.Logger.getLogger("org.apache.http.client").setLevel(Level.OFF);

WebClient webClient = new WebClient(BrowserVersion.CHROME);
webClient.getOptions().setUseInsecureSSL(true);
webClient.setAjaxController(new NicelyResynchronizingAjaxController());
webClient.getOptions().setJavaScriptEnabled(true); //启用 JS 解释器，默认为 true
webClient.getOptions().setCssEnabled(false); //禁用 css 支持

webClient.getOptions().setThrowExceptionOnScriptError(false);
//js 运行错误时，是否抛出异常

HtmlPage page = webClient.getPage(url);
webClient.waitForBackgroundJavaScript(30000);
return Jsoup.parse(page.asXml());
}

```

我们将等待时间设置成了 30000 毫秒，也就是 30s，这样一来虽然爬虫的速度变慢了，但是不容易触发反爬虫机制，我使用一台电脑进行爬取时，获取全部的信息大约使用了 3-4 个小时，

### 3.1.3 分页的处理

当当网的搜索界面是有分页机制的，但是我们细心观察就可以发现分页的 URL 存在一定的规律，比如对于 java 的搜索结果，第一页的 URL 是 <http://search.dangdang.com/?key=java&act=input>，而第二页的 URL 就变成了 [http://search.dangdang.com/?key=java&act=input&page\\_index=2](http://search.dangdang.com/?key=java&act=input&page_index=2)，事实上只是在 URL 中添加了一个 page\_index 的信息，只需要在访问的 URL 中多添加这一信息就可以顺利爬取第 2 页及以后的更多页。

但是我发现当当网的单个页面中内容已经足够多了，因此也就没有进行第二页、第三页的爬取（事实上后面几页开始书籍的信息会重复，因为一样的书太多了）

### 3.1.4 书籍信息的提取

可以从单个页面中提取书籍的信息，包括书名、作者、出版社、出版时间，作者简介、内容简介和目录等等。而这些内容中有一部分是动态生成的，以《疯狂Java讲义》为例

#### 当当自营 疯狂Java讲义（第5版）（含DVD光盘一张）

覆盖Java稳定版本Java11，渗透Java编程思想，李刚作品成为海峡两岸读者之选，本书赠送20+小时视频、源代码、课件、面试题，提供微信答疑群，配套学习网站。 [读更多的书，科技图书，每满100-50>](#)

作者:李刚 出版社:电子工业出版社 出版时间:2019年04月

这些信息是静态的 HTML，而

#### 作者简介

李刚，十余年软件开发从业经验，疯狂软件教育中心教学总监。疯狂Java实训营创始人，疯狂Java体系原创图书作者。广东技术师范学院计算机科学系兼职副教授，CSDN特邀讲师。培训的学生已在腾讯、阿里、华为、IBM、网易、唯品会、电信盈科等名企就职。国内知名高端IT技术图书作家，已出版《疯狂Java讲义》《疯狂Android讲义》《轻量级Java EE企业应用实战》《疯狂前端开发讲义》《疯狂HTML5/CSS3/JavaScript讲义》《疯狂iOS讲义（基础篇）（提高篇）》《疯狂XML讲义》《经典JavaEE企业应用实战》《Struts 2.x权威指南》等著作。其中疯狂Java体系图书均已沉淀多年，赢得极高的市场认同，多次重印，多部著作印刷数量超过10万册，并被多所“985”“211”院校选作教材，部分图书已被翻译成繁体中文版，授权到宝岛台湾。

#### 目 录

第1章 Java语言概述与开发环境 1  
1.1 Java语言的发展简史 2  
1.2 Java程序运行机制 4  
1.2.1 高级语言的运行机制 4  
1.2.2 Java程序的运行机制和JVM 5  
1.3 开发Java的准备 6  
1.3.1 下载和安装Java 11的JDK 6  
不是说JVM是运行Java程序的虚拟机吗？那JRE和JVM的关系是怎样的呢？ 7  
1.3.2 设置PATH环境变量 9  
为什么选择用户变量？用户变量与系统变量有什么区别？ 10  
1.4 第一个Java程序 10  
1.4.1 编辑Java源代码 10  
1.4.2 编译Java程序 11

这些内容则是由 JavaScript 动态生成的，我们进入开发者模式查看网页代码，

```
<div class="messbox_info">
  <span class="t1" id="author" dd_name="作者" ddt-area="002">
    "作者:"
    <a href="http://search.dangdang.com/?key2=%C0%EE%B8%D5&medium=01&category_path=01.00.00.00.00"
      target="_blank" dd_name="作者">李刚</a>
  </span>
  <span class="t1" dd_name="出版社" ddt-area="003">
    "出版社:"
    <a href="http://search.dangdang.com/?key3=%B5%E7%D7%D3%B9%A4%D2%B5%B3%F6%B0%E6%C9%E7&medium=01&category_path=01.00.00.00.00" target="_blank" dd_name="出版社">电子工业出版社</a>
  </span>
  <span class="t1">出版时间:2019年04月&nbsp;  </span>
```

我们可以发现作者、出版社和出版时间这些信息都在 class name 为 messbox\_info 的 div 下面的几个 span 中，因此使用 Jsoup 按照类名来查找类名为 t1 的所有标签就可以获得这三个基本信息。

```

▼<div class="title">
  <span>内容简介</span>
</div>
▼<div class="descrip">
  ▶<span id="content-show" style="display: none;">...</span>
  ▼<span id="content-show-all" style="display: inline;">
    ▼<p> == $0
      "本书是《疯狂Java讲义》的第5版，第5版保持了前4版系统、全面、讲解浅显、细致的特性，全面新增介绍了Java 10、Java 11的新特性。本书深入介绍了Java编程的相关方面，全书内容覆盖了Java的基本语法结构、Java的面向对象特征、Java集合框架体系、Java泛型、异常处理、Java GUI编程、JDBC数据库编程、Java注释、Java的IO流体系、Java多线程编程、Java网络通信编程和Java反射机制。覆盖了java.lang、java.util、java.text、java.io和java.nio、java.sql、java.awt、javax.swing包下绝大部分类和接口。本书重点介绍了Java的模块化系统，还详细介绍了Java 10、Java 11的使用var声明局部变量、在Lambda表达式中使用var声明变量、改进的javac命令、基于嵌套的访问控制、HTTP Client网络编程，以及Java 10、Java 11新增的各种API功能。与前4版类似，本书并不单纯从知识角度来讲解Java，而是从解决问题的角度来介绍Java语言，所以本书中涉及大量实用案例开发：五子棋游戏、梭哈游戏、仿QQ的游戏大厅、MySQL企业管理器、仿EditPlus的文本编辑器、多线程、断点下载工具、Spring框架的IoC容器.....这些案例既能让读者巩固每章的知识，又可以让学生学以致用，激发编程自豪感，进而引爆内心的编程激情。本书光盘里包含书中所有示例的代码和《疯狂Java实战讲义》的所有项目代码，这些项目可以作为本书课后练习的“非标准答案”，如果读者需要获取关于课后习题的解决方法、编程思路，可以登录http://www.crazyit.org站点与笔者及本书庞大的读者群相互交流。本书为所有打算深入掌握Java编程的读者而编写，适合各种层次的Java学习者和工作者阅读，也适合作为大学教育、培训机构的Java教材。但如果只是想简单涉猎Java，则本书过于庞大，不适合阅读。"
    </p>
  </span>
</div>
▼<div id="authorIntroduction" class="section">
  ▼<div class="title">
    <span>作者简介</span>
  </div>
  ▼<div class="descrip">
    <span id="authorIntroduction-all"></span>
    ▼<p> == $0
      "李刚，十余年软件开发从业经验，疯狂软件教育中心教学总监。疯狂Java实训营创始人，疯狂Java体系原创图书作者。广东技术师范学院计算机科学系兼职副教授，CSDN特邀讲师。培训的学生已在腾讯、阿里、华为、IBM、网易、唯品会、电信盈科等名企就职。国内知名高端IT技术图书作家，已出版《疯狂Java讲义》《疯狂Android讲义》《轻量级Java EE企业应用实战》《疯狂前端开发讲义》《疯狂HTML5/CSS3/JavaScript讲义》《疯狂iOS讲义（基础篇）（提高篇）》《疯狂XML讲义》《经典JavaEE企业应用实战》《Struts 2.x权威指南》等著作。其中疯狂Java体系图书均已沉淀多年，赢得极高的市场认同，多次重印，多部著作印刷数量超过10万册，并被多所“985”“211”院校选作教材，部分图书已被翻译成繁体中文版，授权到宝岛台湾。"
    </p>
  </div>
</div>

```

而内容简介、目录、作者简介等信息被存储在一个类名为 `descrip` 的 `div` 标签下面，因此可以先提取出类名为 `descrip` 的类，然后再根据这几项元数据的位置结构特点进行进一步的提取，相关代码如下，我将其抽象为了一个单独的方法，以方便实现代码的复用：

```

public void getSingleBookInfo(String url, String topic) throws IOException {
    Document document = getSingleWebsite(url);
    try {
        Elements elements = document.getElementsByClass("t1");
        //System.out.println(document);
        //System.out.println(elements);
        // 图书的四种最基本的信息
        String title = document.title().replaceAll("/", "")
            .replaceAll(" ", "").replaceAll("<", "")
            .replaceAll(">", "").replaceAll('\\', '\\')
            .replaceAll(" ", "");
    }
}

```

```

        String author = elements.get(0).child(0).html();
        String publisher = elements.get(1).child(0).html();
        String publishTime =
elements.get(2).html().replace("&nbsp;", "");

        Elements dynamicElements =
document.getElementsByClass("descrip");

        // System.out.println(dynamicElements);
        if (dynamicElements.size() >= 4) {
            // 分别获得书本的摘要、内容介绍、作者介绍和具体目录
            String bookAbstract =
getAbstract(dynamicElements.get(1))
                .replaceAll("<.+?>", "").trim();
            String contentIntroduction =
dynamicElements.get(2).text()
                .replaceAll("<.+?>", "").trim();
            String authorIntroduction =
dynamicElements.get(3).text()
                .replaceAll("<.+?>", "").trim();
            String contents = dynamicElements.get(4).text()
                .replaceAll("<.+?>", "\n")
                .replaceAll("\n\n", "\n");
            // 创建一本书的对象
            Book thisBook = new Book(title, author, publisher,
publishTime, bookAbstract,
contentIntroduction,
                authorIntroduction, contents, topic);
            // thisBook.showBookInformation();
            System.out.println("成功爬取了 " + title + " 并开始将
信息写入文件!!");
            thisBook.writeInfoToFile();
        }
    } catch (Exception exception) {
        System.out.println("在爬取主题" + topic + "\n 网址" + url
+ "时失败, 本页面爬取无效!");
    }
}

public String getAbstract(Element element) {
    StringBuilder res = new StringBuilder();
    for (int i = 1; i < element.childrenSize(); i++) {
        if (!element.child(i).html().isEmpty()) {

```

```

res.append(element.child(i).html().replaceAll("&lt;br", "").
    replaceAll("/&gt;", "").replaceAll("p&gt;", "").
    replaceAll("&nbsp;", "").replaceAll("<.+?>", ""));
res.append('\n');
    }
}
return res.toString();
}

```

在处理的过程中我用了一些正则表达式的替换来进行冗余信息的删除，比如多余的空格和换行，以及乱码和标签等等。但是要注意，当当网中的书籍的信息呈现方式并不完全一致，有的时候部分页面会缺少作者简介或者目录或者内容简介等等，这个时候我就设置了异常并进行捕获，发生异常时就说明这本书的页面中缺少关键信息，因此我也将其舍弃了。

### 3.1.5 书籍信息的结构化保存

在获取书籍的信息之后就需要进行结构化的存储，为此，我设计了一个 `Book` 类用来实现结构化的文件信息保存，类中的主要变量有：

```

private final String title;
private final String author;
private final String publisher;
private final String publishTime;
private final String bookAbstract;
private final String contentIntroduction;
private final String authorIntroduction;
private final String contents;
private final String topic;

```

都是上面爬取到的一本书籍的相关信息，我们将这些信息分散到四个 `txt` 文件中，一个文件存储标题、作者、出版商的基本信息，一个存储摘要、一个存储作者介绍、一个存储目录，具体的实现代码如下所示：

```

/**
 * 将一个书本对象的各种信息按照一定的规则存储在对应的目录下面
 */
public void writeInfoToFile() {
    String path = "./data/" + topic + "/" + title;

```



```

File bookDictionary = new File(path);
// 没有的时候创建目录
if (!bookDictionary.exists()) {
    bookDictionary.mkdir();
}
path = path + "/";
// 基本信息单独写一个文件

String basic = path + "基本信息.txt";
File basicInfo = new File(basic);
try {
    if (!basicInfo.exists()) {
        basicInfo.createNewFile();
    }
    FileWriter fileWriter = new FileWriter(basic);
    fileWriter.write(title + '\n');
    fileWriter.write(author + '\n');
    fileWriter.write(publisher + '\n');
    fileWriter.write(publishTime + '\n');
    fileWriter.close();
} catch (IOException e) {
    e.printStackTrace();
}
// 摘要单独写一个文件

```

```

String abstractOfBook = path + "摘要.txt";
File abstractBook = new File(abstractOfBook);
try {
    if (!abstractBook.exists()) {
        abstractBook.createNewFile();
    }
    FileWriter fileWriter = new FileWriter(abstractBook);
    fileWriter.write(this.bookAbstract);
    fileWriter.close();
} catch (IOException e) {
    e.printStackTrace();
}
// 作者简介单独写一个文件

```

```

String authorOfBook = path + "作者简介.txt";
File authorInfo = new File(authorOfBook);
try {
    if (!authorInfo.exists()) {
        authorInfo.createNewFile();
    }
    FileWriter fileWriter = new FileWriter(authorInfo);
    fileWriter.write(this.authorIntroduction);
}

```



```

        fileWriter.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
    // 内容简介单独写一个文件
    String simpleIntroduce = path + "内容简介.txt";
    File introduce = new File(simpleIntroduce);
    try {
        if (!introduce.exists()) {
            introduce.createNewFile();
        }
        FileWriter fileWriter = new FileWriter(introduce);
        fileWriter.write(this.contentIntroduction);
        fileWriter.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
    // 目录单独写一个文件
    String contentOfBook = path + "目录.txt";
    File content = new File(contentOfBook);
    try {
        if (!content.exists()) {
            content.createNewFile();
        }
        FileWriter fileWriter = new FileWriter(content);
        fileWriter.write(this.contents);
        fileWriter.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

因此最后爬虫所得到的数据都存储在源代码同个目录下面的 `data` 文件夹中，结构如下所示：

> Java > JavaHW > HW4 > data >			
名称	修改日期	类型	
📁 C++	2020/12/6 22:42	文件夹	
📁 Java	2020/12/6 22:11	文件夹	
📁 Python	2020/12/6 22:26	文件夹	
📁 操作系统	2020/12/6 23:29	文件夹	
📁 机器学习	2020/12/7 3:04	文件夹	
📁 计算机网络	2020/12/7 1:56	文件夹	
📁 计算机组成	2020/12/7 1:17	文件夹	
📁 软件工程	2020/12/7 2:29	文件夹	
📁 数据结构	2020/12/7 0:10	文件夹	
📁 数据库	2020/12/7 0:43	文件夹	

这是每种类型的书的总目录，每个目录下面有若干本相关的书籍，比如 Java 下面就有

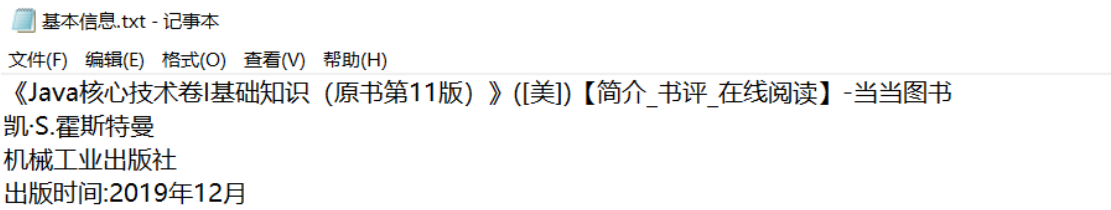
Java > JavaHW > HW4 > data > Java >			
名称	修改日期	类型	
📁 《Java从入门到精通（第5版）》(明日科...	2020/12/6 21:58	文件夹	
📁 《深入理解Java虚拟机：JVM高级特性...	2020/12/6 21:59	文件夹	
📁 《Java核心技术卷I基础知识（原书第11...	2020/12/6 21:59	文件夹	
📁 《HeadFirstJava（中文版）（JAVA经...	2020/12/6 21:59	文件夹	
📁 《零基础学Java（全彩版）》(明日科技(...	2020/12/6 21:59	文件夹	
📁 《疯狂Java讲义（第5版）（含DVD光盘...	2020/12/6 22:00	文件夹	
📁 《Java并发编程实战（第16届Jolt大奖提...	2020/12/6 22:00	文件夹	
📁 《Java精彩编程200例（全彩版）》(明...	2020/12/6 22:00	文件夹	
📁 《Java核心技术卷II高级特性（原书第11...	2020/12/6 22:00	文件夹	
📁 《Java编程思想（第4版）》([美]BruceE...	2020/12/6 22:01	文件夹	
📁 《Java项目开发实战入门（全彩版）》(...	2020/12/6 22:01	文件夹	
📁 《Java并发编程的艺术》(方腾飞)【简介...	2020/12/6 22:01	文件夹	
📁 《EffectiveJava中文版（原书第3版）》...	2020/12/6 22:02	文件夹	
📁 《Java并发编程之美》(霍陆续薛宾田)【...	2020/12/6 22:02	文件夹	
📁 《Java核心技术第11版基础知识+高级特...	2020/12/6 22:02	文件夹	
📁 《Java从入门到精通精粹版》(张玉宏)【...	2020/12/6 22:03	文件夹	
📁 《Offer来了：Java面试核心知识点精讲...	2020/12/6 22:03	文件夹	
📁 《数据结构与算法分析：Java语言描述（...	2020/12/6 22:04	文件夹	
📁 《Java程序性能优化——让你的Java程...	2020/12/6 22:04	文件夹	
📁 《数据结构与算法Java语言描述》([美]Al...	2020/12/6 22:04	文件夹	
📁 《Java设计模式及实践》([印度]卡马尔米...	2020/12/6 22:04	文件夹	
📁 《Java从入门到项目实战（全程视频版）...	2020/12/6 22:05	文件夹	
📁 《Java8入门与实践（微课视频版）》(丁...	2020/12/6 22:05	文件夹	
📁 《Java开发详解（全彩版）》(明日科技(...	2020/12/6 22:05	文件夹	
📁 《Java架构师指南》(王波)【简介_书评_...	2020/12/6 22:06	文件夹	
📁 《亿级流量Java高并发与网络编程实战》...	2020/12/6 22:06	文件夹	
📁 《Java高并发编程详解：深入理解并发核...	2020/12/6 22:06	文件夹	

而每本书的目录下面都有 5 个 txt 文件，如下图所示：

Java > JavaHW > HW4 > data > Java > 《Java核心技术卷I基础知识（原书第11版）》([美])【简介\_书评\_在线阅读】-当当图书

名称	修改日期	类型	大小
基本信息.txt	2020/12/6 21:59	文本文档	1 KB
目录.txt	2020/12/6 21:59	文本文档	14 KB
内容简介.txt	2020/12/6 21:59	文本文档	2 KB
摘要.txt	2020/12/6 21:59	文本文档	2 KB
作者简介.txt	2020/12/6 21:59	文本文档	1 KB

txt 文件中存储的信息如下图所示



至此，爬虫模块的总体架构和具体实现就完成了，我一共爬取了几百本书籍的数据，下面需要建立索引并进行搜索。

## 3.2 搜索模块的基本设计

### 3.2.1 索引的建立

本次作业中我们使用老师提供的 Lucene 库来进行索引的建立，老师已经给出 Lucene 包和相关的测试程序，而我的程序也在此基础上进行编写，利用

Lucene 建立索引需要调用 IKAnalyzer 来对中文分词进行分析，并将需要建立索引的文件一个个加入到创建的 IndexWriter 对象中，不断加入 Document 对象。而 Document 对象需要遍历 data 目录下爬到的所有数据，为每一本书创建一个 document 对象，就可以，建立索引的关键代码如下所示：

```
public void createIndex(String filePath) {
    File f = new File(filePath);
    IndexWriter iwr=null;
    try {
        Directory dir = FSDirectory.open(f);
        Analyzer analyzer = new IKAnalyzer();
        IndexWriterConfig conf=new
IndexWriterConfig(Version.LUCENE_4_10_0,analyzer);
        iwr = new IndexWriter(dir,conf); //建立 IndexWriter。固定
        套路
        // 清空之前建立的所有索引
        try {
            iwr.deleteAll();
        } catch (Exception e) {
            e.printStackTrace();
        }
        // 遍历得到所有的文档，然后进行索引的创建
        LinkedList<Document> docs = getAllDocuments();
        for (Document doc: docs) {
            iwr.addDocument(doc);
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
    try {
        iwr.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

而 getAllDocument 方法会遍历所有爬去到的数据并创建 document 对象，其核心代码如下所示：

```
public LinkedList<Document> getAllDocuments () {
    LinkedList<Document> result = new LinkedList<Document>();
    int id = 0;
```

```

try {
    // 进入爬虫所得数据的根目录
    File root = new File("./data");
    try {
        // 获取每种书的目录并进行遍历
        File[] dictionary = root.listFiles();
        for (File topic: dictionary) {
            // 获取每种书下面的所有单本书籍
            File[] books = topic.listFiles();

            // 遍历一本书所在的目录下面的所有信息
            for (File book: books) {
                // 创建一本书的索引 Document
                Document doc = new Document();
                // 先写入一个编号
                Field no = new IntField("No.", id,
Field.Store.YES);
                id += 1;
                doc.add(no);
                File[] information = book.listFiles();
                // 获取书的每一项信息，并写入这个 doc 中
                for (File info: information) {
                    int temp;
                    StringBuilder thingsInFile = new
StringBuilder();
                    try {
                        Reader reader = new
InputStreamReader(new FileInputStream(info));
                        while ((temp = reader.read()) != -1)
{
                            thingsInFile.append((char) temp);
                        }
                    } catch (Exception exception) {
                        exception.printStackTrace();
                    }
                    String thing =
thingsInFile.toString().trim()
                        .replace('/', '
').replaceAll("\n\n", "\n");
                    // 将爬到书本信息作为 document 中的一项写入
                    Field bookInfo = new
TextField(info.getName().replace(".txt", ""),
                        thing, Field.Store.YES);
                    doc.add(bookInfo);

```

```

        }
        result.add(doc);
    }
}
} catch (NullPointerException exception) {
    exception.printStackTrace();
}

} catch (Exception exception) {
    exception.printStackTrace();
}
return result;
}
}

```

### 3.2.2 搜索引擎的运行

搜索引擎需要和用户进行交互，读取用户希望查找的关键字，然后用户需要选择查找的范围（基本信息、作者简介、内容简介、目录等等），最后系统输出对应的检索结果，而我发现一些书籍的作者简介、目录、摘要等内容往往比较长，如果大量输出也不一定让用户可以看的清楚明白，因此我也设置了用户可以自主选择是否查看详细信息这一个功能，如果选择是就输出，否则就只输出书本的关键信息（书名、作者、出版社），比较方便阅读。

总的来说交互逻辑较为简单，而实现这一交互逻辑的核心代码为：

```

public void search (String filePath) {
    File f = new File(filePath);
    Scanner input = new Scanner(System.in);
    try {
        while (true) {
            IndexSearcher searcher = new
IndexSearcher(DirectoryReader.open(FSDirectory.open(f)));
            System.out.println("输入要查找的关键字，输入 quit 结束
搜索引擎的运行!");
            String userInput, userChoice, needMore;
            userInput = input.next();
            if (userInput.equals("quit")) {
                break;
            } else {
                Analyzer analyzer = new IKAnalyzer();

```

```

        System.out.println("输入信息的检索范围：1.基本信息，2.目录，3.内容简介，4.摘要，5.作者简介");
        int choice = input.nextInt();
        if (choice == 1) {
            userChoice = "基本信息";
        } else if (choice == 2) {
            userChoice = "目录";
        } else if (choice == 3) {
            userChoice = "内容简介";
        } else if (choice == 4) {
            userChoice = "摘要";
        } else {
            userChoice = "作者简介";
        }
        System.out.println("是否显示详细信息？ Y/N");
        needMore = input.next();
        int resNum = 5;
        QueryParser parser = new QueryParser(userChoice, analyzer);

        Query query = parser.parse(userInput);
        TopDocs hits = searcher.search(query, resNum);
        //前面几行代码也是固定套路，使用时直接改 field 和关键词即可

        System.out.println("查询结果： \n");
        for (ScoreDoc res: hits.scoreDocs) {
            Document d = searcher.doc(res.doc);
            System.out.println("-----书籍基本信息-----");

            System.out.println(d.get("基本信息"));
            if (needMore.equals("Y") || needMore.equals("y")) {
                System.out.println(d.get("目录"));
                System.out.println(d.get("内容简介"));
                System.out.println(d.get("摘要"));
                System.out.println(d.get("作者简介"));
            } else {
                System.out.println("目录，内容摘要，作者简介等可以选择查看详细信息进行查看！");
            }
        }
        System.out.println("-----");
    }
}

```

```

    }

    } catch (IOException | ParseException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

```

## 4.测试与运行

### 4.1 程序测试

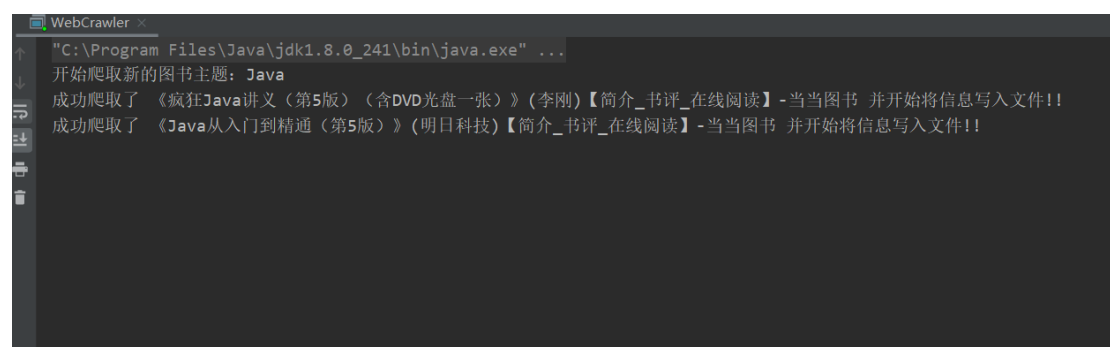
经过了多轮调试和代码优化，目前两个程序都可以正常运行，而在测试的过程中我也发现了一些问题，比如：

- 爬虫程序的运行速度比较慢
- 爬虫程序对特殊字符的兼容性不强，比如出现正反斜杠的时候程序容易抛出异常，根据捕获的异常我认为是因为 Windows 机在创建文件和文件夹的时候对特殊字符有所限制

因此我使用了正则表达式对一些特殊字符进行了替换，解决了这个问题，而爬虫程序运行比较慢的问题依然无法解决

### 4.2 程序运行

- 爬虫模块运行时



```

WebCrawler
"C:\Program Files\Java\jdk1.8.0_241\bin\java.exe" ...
开始爬取新的图书主题: Java
成功爬取了 《疯狂Java讲义（第5版）（含DVD光盘一张）》（李刚）【简介_书评_在线阅读】-当当图书 并开始将信息写入文件!!
成功爬取了 《Java从入门到精通（第5版）》（明日科技）【简介_书评_在线阅读】-当当图书 并开始将信息写入文件!!

```

爬取到的数据以及其格式在之前的报告中已经展示过了，也可以详细查看提交的源代码中的数据集进行查看



« Java » 《Java语言程序设计》(千锋教育高教产品研发部)【简介_书评_在线... 搜索"《Java...				
名称	修改日期	类型	大小	
基本信息.txt	2020/12/6 22:08	文本文档	1 KB	
目录.txt	2020/12/6 22:08	文本文档	6 KB	
内容简介.txt	2020/12/6 22:08	文本文档	1 KB	
摘要.txt	2020/12/6 22:08	文本文档	1 KB	
作者简介.txt	2020/12/6 22:08	文本文档	8 KB	

## ● 搜索引擎模块检索单个关键字时

```
正在创建索引中，请稍等.....
索引建立完成。
输入要查找的关键字，输入quit结束搜索引擎的运行！
Java
输入信息的检索范围：1.基本信息，2.目录，3.内容简介，4.摘要，5.作者简介
是否显示详细信息？Y/N
查询结果：

-----书籍基本信息-----
《Java并发编程实战（第16届Jolt大奖提名图书，Java并发编程必读佳作）》([美]BrianGoetz等)【简介_书评_在线阅读】-当当图书
Brian
机械工业出版社
出版时间:2012年02月
目录，内容摘要，作者简介等可以选择查看详细信息进行查看！

-----书籍基本信息-----
《Java程序性能优化—让你的Java程序更快、更稳定》(葛一鸣)【简介_书评_在线阅读】-当当图书
葛一鸣
清华大学出版社
出版时间:2012年10月
目录，内容摘要，作者简介等可以选择查看详细信息进行查看！

-----书籍基本信息-----
《Java多线程编程核心技术(资深Java专家10年经验总结，全程案例式讲解，首本全面介绍Java多线程编程技术的专著)》(高洪岩)【简介_书评_在线阅读】-当当图书
高洪岩
机械工业出版社
```

## ● 搜索引擎检索单个关键字并显示详细信息

```
输入要查找的关键字，输入quit结束搜索引擎的运行！
数据结构
输入信息的检索范围：1.基本信息，2.目录，3.内容简介，4.摘要，5.作者简介
是否显示详细信息？Y/N
查询结果：

-----书籍基本信息-----
《数据结构与算法实验指导书》(汪沁、邓芳、莫李峰)【简介_书评_在线阅读】-当当图书
汪沁
清华大学出版社
出版时间:2018年08月
前言 数据结构是计算机专业的核心课程,它从长期的程序设计实践中提炼而成,运用于程序设计;更是操作系统、编译原理等计算机核心课程的基础,在计算机专业课程中起着承上启下的作用。 数据结构
本书是数据结构课程的辅助教材,采用C和C++两种语言来描述数据结构,让学生在实验与习题中体会与掌握数据结构,同时培养编程能力和分析能力。主要包括实验与习题两大部分,用于巩固数据结
目录 第1部分实验要求及规范1第2部分面向过程语言实现数据结构3 实验0 复数ADT及其实现3 实验1 线性表(顺序表)4 实验2 线性表(链表)7 实验3 栈12 实验4 队列15 实验5 串与数组20 实验6 树与二叉树24
实验7 图28
第1部分实验要求及规范1第2部分面向过程语言实现数据结构3
实验0 复数ADT及其实现3
实验1 线性表(顺序表)4
实验2 线性表(链表)7
实验3 栈12
```

```
-----书籍基本信息-----
《数据结构（Python版）》(吕云翔、郭颖美、孟文)【简介_书评_在线阅读】-当当图书
吕云翔
清华大学出版社
出版时间:2019年03月
前言 随着近年来计算概念的快速拓展,计算科学已经发展成为一个内涵繁杂的综合性学科,其至少可以划分为计算机工程(CE)、计算机科学(CS)、信息系统(IS)、信息技术(IT)和软件工程(SE)5个
随着近年来计算概念的快速拓展,计算科学已经发展成为一个内涵繁杂的综合性学科,其至少可以划分为计算机工程(CE)、计算机科学(CS)、信息系统(IS)、信息技术(IT)和软件工程(SE)5个
N.Wirth早在20世纪70年代就指出“程序=数据结构+算法”。数据结构主要研究数据在计算机中储存、组织、传递和转换的过程及方法,这些也是构成与支撑算法的基础。近年来,随着面向对象技术的广泛
基于上述情况,本书选择Python作为描述语言。Python语言语法简洁优美,功能强大,有着广泛的应用领域,如互联网、大数据、人工智能等领域。因此,学习Python语言,在未来的学习和工作中,都
在内容的选取与结构安排上,本书通过分类和讲解典型结构使读者对数据结构形成宏观认识,根据内容的侧重,本书分8章,分别为绪论、线性表、栈和队列、串和数组、树结构、图、排序和查找。
第1章介绍数据结构的基本概念、算法描述、算法的时间复杂度和空间复杂度等内容。本章是全书的基础。
第2章主要介绍线性表的基本概念和抽象数据类型的定义,线性表的顺序和链式两种存储方式的标识,以及线性表的基本操作实现和相应应用。
第3章简要介绍栈和队列的基本概念和抽象数据类型定义,栈和队列在顺序存储和链式存储结构下的基本操作和应用。
第4章主要介绍串的基本概念和数据类型定义,串的存储结构、基本操作实现和应用等内容。
第5章主要介绍树和二叉树的基本概念,详细介绍二叉树的性质和存储结构、便利方法的实现及应用、哈夫曼树的概念和构造方法。
第6章主要介绍图的基本概念、抽象数据类型定义、存储结构和遍历方法,还介绍最小生成树的基本概念和方法、最短路径的相关算法、拓扑排序的概念和实现方法。
第7章介绍排序的基本概念,插入排序、交换排序、选择排序、归并排序等多种排序的原理、实现方法及性能分析。
第8章主要介绍查找的基本概念,顺序查找、二分查找等查找的原理、实现方法和性能分析,平衡二叉树、哈希表的概念、结构定义和实现方法。
本书的理论知识的教学安排建议如下表所示。 显示全部信息
本书在选材与编排上,贴近当前普通高等院校“数据结构”课程的现状和发展趋势,内容难易适度,突出实用性和应用性。本书并未面面俱到地介绍各种数据结构,而是通过分类和讲解典型结构,使读者对
目录 第1章绪论 1.1引言 1.1.1学习目的 1.1.2课程内容 1.2基本概念 1.2.1数据与数据结构 目录第1章绪论1.1引言1.1.1学习目的1.1.2课程内容1.2基本概念1.2.1数据与数据结构1.2.2数据类型
```

## ● 多个关键字的检索

```
正在创建索引中,请稍等.....
索引建立完成.
输入要查找的关键字,输入quit结束搜索引擎的运行!
贝叶斯 决策树
输入信息的检索范围: 1.基本信息, 2.目录, 3.内容简介, 4.摘要, 5.作者简介
3
是否显示详细信息? Y/N
N
查询结果:

-----书籍基本信息-----
《机器学习与深度学习算法基础》(贾壮)【简介_书评_在线阅读】-当当图书
贾壮
北京大学出版社
出版时间:2020年08月
目录, 内容摘要, 作者简介等可以选择查看详细信息进行查看!

-----书籍基本信息-----
《机器学习经典算法实践》(肖云鹏卢星宇许明汪浩瀚吴斌刘宴兵著)【简介_书评_
肖云鹏
清华大学出版社
出版时间:2018年07月
目录, 内容摘要, 作者简介等可以选择查看详细信息进行查看!

-----书籍基本信息-----
《机器学习》(赵卫东董亮)【简介_书评_在线阅读】-当当图书
赵卫东
人民邮电出版社

O 4: Run Statistic Terminal
```

## ● 按 quit 退出搜索引擎

```
正在创建索引中，请稍等.....  
索引建立完成。  
输入要查找的关键字，输入quit结束搜索引擎的运行！  
quit  
  
Process finished with exit code 0  
|
```

## 5. 总结

这次的作业难度比较大，对代码能力和设计能力都有比较大的挑战，并且我还第一次尝试了在 Java 中导入并使用外部库的开发模式，快速掌握一个新的库中的重要类和方法的用法也是一项非常重要的技能。

这次作业中我也遇到过很多困难，比如一开始直接使用 jsoup 对当当网中的书籍进行爬取的时候，我发现 jsoup 爬去到的只有静态的 HTML 而缺少了其中一部分需要用 JavaScript 动态生成的数据，查阅资料并询问老师之后，我了解到 jsoup 并没有这一功能，老师也要求我自己想办法解决这个问题，因此我选择了使用 HtmlUnit 这一新的 Java 外置包来帮助我进行爬取并解析动态生成的内容，最终获得了成功。

此外，我在运行爬虫程序的时候发现当当网存在一定的反爬虫机制，如果爬取速度过快就会被检测到并封禁 IP，而我采取的策略就是放慢速度，比较缓慢地爬取网页，最终耗时三个多小时并获得了成功。

建立索引部分因为老师已经提供了对应的库和示例代码因此难度不算特别大，经过摸索之后我就完成了索引模块和用户交互模块的编写。

总的来说这次 Java 的作业收获比较大，也让我了解到了 Java 丰富的功能，时间了课堂中所学到的文件 I/O 和异常检查的相关知识。

## 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] 王鹏. Java Swing 图形界面开发与案例详解[M]. 北京: 清华大学出版社, 2008.
- [4] 丁振凡. Java 语言实验教程[M]. 北京: 北京邮电大学出版社, 2005.
- [5] 郑莉. Java 语言程序设计[M]. 北京: 清华大学出版社, 2006.