**深 圳 大 学 实 验 报 告**

**课程名称： 互联网编程**

**实验项目名称：实验3 域名IP转换及应用URL类定位和获取数据编程**

**学院： 计算机与软件学院**

**专业： 软件工程**

**指导教师： 毛斐巧**

**报告人：杨皓翔 学号：2023150139 班级： 软工02**

**实验时间： 第8-11周（周二）**

**实验报告提交时间： 2025年5月12日**

**教务处制**

一、实验目的与内容：

**目的：**

1.掌握网络、文件IO流相关编程技术及能力。

2.了解域名系统（DNS）原理，掌握InetAddress类及Inet4Address/Inet6Address的使用方法。

3.掌握NetworkInterface类的使用方法。

4.掌握URL类的使用，在不考虑所使用协议的细节情况下，会用java编写程序在网络上定位和获取数据。

**内容要求：**

1. 编程解析域名（20分）：

编写一个可重用的域名解析程序模块，使之能够将用户输入的域名解析为IP地址，能将用户输入的IP地址，反向解析为对应的主机名或域名。

1. 编程下载一个web页面（20分）：

编写程序，用户输入一个web页面的URL网址，程序会将该URL网址所指向的web页面内容下载到本地显示，并将网页内容以网页文件格式保存到指定磁盘位置。

1. 编程分解URL（20分）：

编写程序，用户输入一个URL网址，程序能够分析该URL，显示输出该URL的各组成部分。

1. 编程完成一次当当网购物网站的购物搜索（30分）：

（说明：如该购物搜索链接失效，可自行更换成别的购物搜索网页链接（如京东、淘宝等），只要实现类似的编程完成该购物搜索网页的自动购物搜索过程。）

当当购物网站主页是http://www.dangdang.com/，如下图1所示，该主页有一个用于搜索商品的表单，在浏览器浏览打开主页情景下，用户可以在搜索关键字输入框中输入商品关键字，点击“搜索”按钮，会返回搜索结果页面。比如输入“洗衣机”关键字，点击“搜索”按钮，会返回如图2所示搜索结果页面。



图1商品搜索页面



图2“洗衣机”商品搜索结果页面

假如没有浏览器辅助浏览，编写程序，程序接收用户输入的一个商品搜索关键词（例如“空调”），程序能够自动打开当当购物网站主页，提交搜索申请，启动搜索，得到搜索结果页面，将搜索结果页面源码显示输出出来，并以productResult.html文件保存在磁盘上。

注意：

1. 实验报告中需要有实验结果的截屏图像。

二、实验过程和代码与结果

**1. 给出满足内容要求1的程序源码及运行结果，简述思路或实验过程。**

1.1 创建目录demo1，在demo1下创建文件myDNS.java用于编写域名解析程序模块，以及一个test.java文件用于测试模块功能。

1.2 编写域名解析程序模块

该模块包含两个实例方法nameToIP和IPtoName，两个方法的实现基本一致，除了提示语和返回的内容有所不同而已。

大致思路如下：

首先，让用户输入需要解析的域名或IP。

然后，根据用户输入的域名或IP获取一个InetAddress实例对象。

最后，根据需要调用InetAddress的两个不同实例方法获取IP或域名，返回出去。

**程序源码：**

*//可重用的域名解析程序模块*package demo1;  
import java.net.InetAddress;  
import java.net.UnknownHostException;  
import java.util.Scanner;  
  
public class myDNS {  
 Scanner sc = new Scanner(System.*in*);  
 *//将用户输入的域名解析为IP地址* public String nameToIP() throws UnknownHostException {  
 *//接收用户输入的域名* System.*out*.print("请输入您要解析的域名：");  
 String hostName = sc.next();  
 *//根据域名获取address* InetAddress address = InetAddress.*getByName*(hostName);  
 *//返回address的IP* return address.getHostAddress();  
 }  
 *//将用户输入的IP地址，反向解析为对应的主机名或域名* public String IPtoName() throws UnknownHostException {  
 *//接收用户输入的IP* System.*out*.print("请输入您要解析的IP：");  
 String IP = sc.next();  
 *//根据IP获取address* InetAddress address = InetAddress.*getByName*(IP);  
 *//返回address的域名* return address.getHostName();  
 }  
}

1.3 编写test测试类

**程序源码：**

package demo1;  
import java.net.UnknownHostException;  
  
public class test {  
 public static void main(String[] args) throws UnknownHostException {  
 *//创建DNS解析服务对象* myDNS dns = new myDNS();  
 *//调用域名解析为IP的服务* String IP = dns.nameToIP();  
 *//打印解析出的IP* System.*out*.println("该域名对应的IP地址为：" + IP);  
 *//调用IP解析为域名的服务* String name = dns.IPtoName();  
 *///打印出解析出的域名* System.*out*.println("该IP地址对应的域名为：" + name);  
 }  
}

1.4 运行结果：

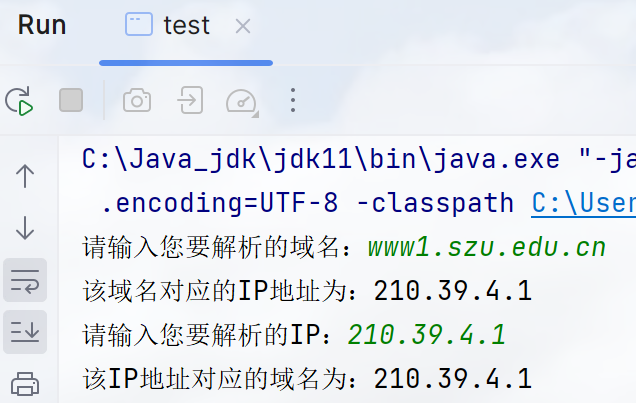
****

图1 要求1运行结果图

由图可见，该域名解析程序模块确实能将域名解析为IP地址，可当将IP地址反解析为域名时，却不能达到目的，这是一个正常现象。

原因：许多网站服务器的 IP 地址都有对应的主机名或域名，但是并非所有的 IP 地址都设置了反向解析（PTR 记录），所以无法通过 IP 地址直接获取到域名。此外，有些网站可能会配置防火墙或安全设置，禁止从外部查询到服务器的详细信息，包括域名。

**2. 给出满足内容要求2的程序源码及运行结果，简述思路或实验过程。**

2.1 让用户输入一个url

2.2 根据用户输入的url创建URL对象并打开url连接

2.3 通过url连接中的流对象读取网页内容

2.4 将网页内容打印在控制台，并保存到指定文件中

2.5 程序源码：

*//功能：下载一个web页面*package demo2;  
import java.io.BufferedReader;  
import java.io.FileWriter;  
import java.io.IOException;  
import java.io.InputStreamReader;  
import java.net.MalformedURLException;  
import java.net.URL;  
import java.net.URLConnection;  
import java.util.Scanner;  
  
public class loadWeb {  
 public static void main(String[] args) throws IOException {  
 Scanner sc = new Scanner(System.*in*);  
 *//用户输入一个url网址* System.*out*.print("请输入您要下载的Web页面的url：");  
 String urlString = sc.next();  
 *//创建 URL 对象* URL url = new URL(urlString);  
 *//打开url连接* URLConnection connection = url.openConnection();  
 *//获取url连接的字节流，创建缓冲流对象* BufferedReader reader = new BufferedReader(new InputStreamReader(connection.getInputStream()));  
 *//读取网页内容* StringBuilder content = new StringBuilder();  
 String line;  
 while ((line = reader.readLine()) != null) {  
 content.append(line);  
 content.append("\n");  
 }  
 reader.close();  
 *// 在控制台显示网页内容* System.*out*.println("网页内容：");  
 System.*out*.println(content);  
 *// 将网页内容保存为文件* String filePath = "src/demo2/webpage.html/";*//文件路径* FileWriter fileWriter = new FileWriter(filePath);  
 fileWriter.write(content.toString());  
 fileWriter.close();  
 System.*out*.println("网页已保存至 " + filePath);  
 }  
}

2.6 运行结果：

图片包含 日历

描述已自动生成

图2 控制台打印内容

文本

描述已自动生成

图3 下载的网页文件内容

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图4 浏览器打开此网页内容效果

**3. 给出满足内容要求3的程序源码及运行结果，简述思路或实验过程。**

3.1 创建Scanner输入对象

3.2 用户输入想要分析的url

3.3 创建URL对象

3.4 利用URL实例对象的实例方法分解url各部分信息，并打印在控制台

3.5 程序源码：

*//功能：分析URL的各组成部分*package demo3;  
import java.io.IOException;  
import java.net.MalformedURLException;  
import java.net.URL;  
import java.util.Scanner;  
public class parseURL {  
 public static void main(String[] args) throws IOException {  
 Scanner sc = new Scanner(System.*in*);  
 *// 获取用户输入的 URL* System.*out*.print("请输入您要分析的url：");  
 String urlString = sc.next();  
 *// 创建 URL 对象* URL url = new URL(urlString);  
 *// 输出各组成部分* System.*out*.println("协议Protocol: " + url.getProtocol());  
 System.*out*.println("域名Authority: " + url.getAuthority());  
 System.*out*.println("主机Host: " + url.getHost());  
 System.*out*.println("端口Port: " + url.getPort());  
 System.*out*.println("默认端口号: " + url.getDefaultPort());  
 System.*out*.println("路径Path: " + url.getPath());  
 System.*out*.println("内容Content: " + url.getContent());  
 }  
}

3.6 运行结果：

图形用户界面, 文本, 聊天或短信

描述已自动生成

图5 分解url各部分内容

**4. 给出满足内容要求4的程序源码及运行结果，简述思路或实验过程。**

4.1 实验前，使用当当网搜索模块后发现，当我们输入搜索关键词“雅思”后，url后面会拼接上“？key = 雅思”，如下图所示：



图6 搜索关键词后的实际url请求

于是，我们可以得知，搜索什么内容，就在“https://search.dangdang.com/”后面拼接上“？key = 关键词”即可。

4.2 实验过程：

1. 用户输入商品关键词
2. 根据关键词构建搜索url
3. 建立url连接
4. 利用url连接通道中的输入流获取网页源码
5. 将网页源码内容输出至控制台
6. 将源码保存到文件productResult.html中

4.3 程序源码：

package demo4;  
import java.io.\*;  
import java.net.URL;  
import java.net.URLConnection;  
import java.util.Scanner;  
public class DangDangSearch {  
 public static void main(String[] args) throws IOException {  
 *// 用户输入商品搜索关键词* Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("请输入您要搜索的内容：");  
 String keyword = scanner.nextLine();  
 *// 构建搜索 URL* String searchUrl = "http://search.dangdang.com/?key=" + keyword;  
 *// 建立url连接 并 获取HTML源代码* String htmlContent = *getHtmlContent*(searchUrl);  
 keyword = URLEncoder.encode(keyword,"gbk"); // 防止乱码

*// 输出HTML源代码到控制台* System.*out*.println(htmlContent);  
 *// 将HTML源代码保存到文件productResult.html中  
 saveHtmlToFile*(htmlContent, "src/demo4/productResult.html");  
 }  
 private static String getHtmlContent(String url) throws IOException {  
 *// 建立连接* URLConnection connection = new URL(url).openConnection();  
 *//获取url连接的字节流，创建缓冲流对象* BufferedReader reader = new BufferedReader(new InputStreamReader(connection.getInputStream()));  
 *//获取 HTML 源代码* StringBuilder content = new StringBuilder();  
 String line;  
 while ((line = reader.readLine()) != null) {  
 content.append(line);  
 content.append("\n");  
 }  
 reader.close();  
 return content.toString();  
 }  
 private static void saveHtmlToFile(String htmlContent, String fileName) throws IOException {  
 BufferedWriter writer = new BufferedWriter(new FileWriter(fileName));  
 writer.write(htmlContent);  
 System.*out*.println("搜索结果已保存到 " + fileName);  
 }  
}

关键细节：（需要对搜索关键词进行编码，否则会导致获取的网页乱码）

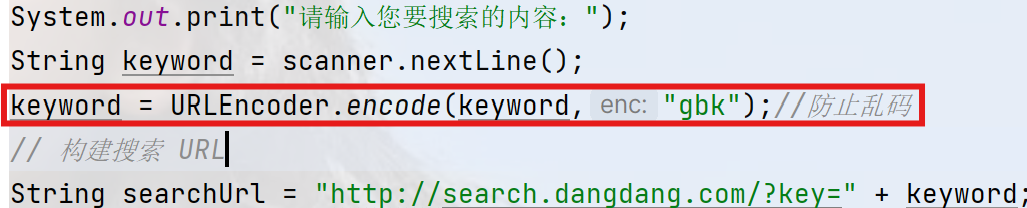


图7 指定搜索关键词的编码格式

4.4 运行结果：



图8 部分控制台打印内容

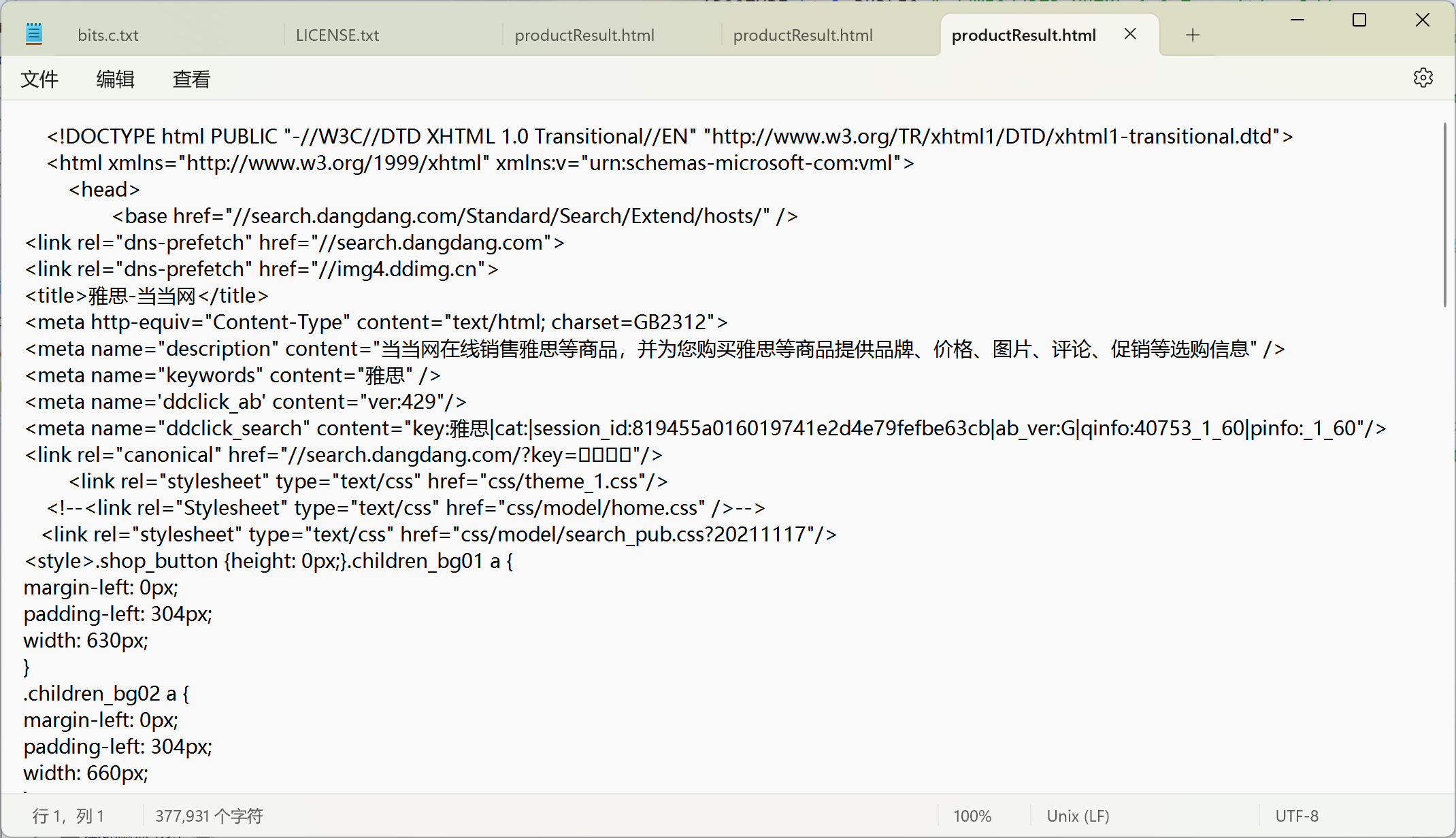


图9 保存的网页文件内容

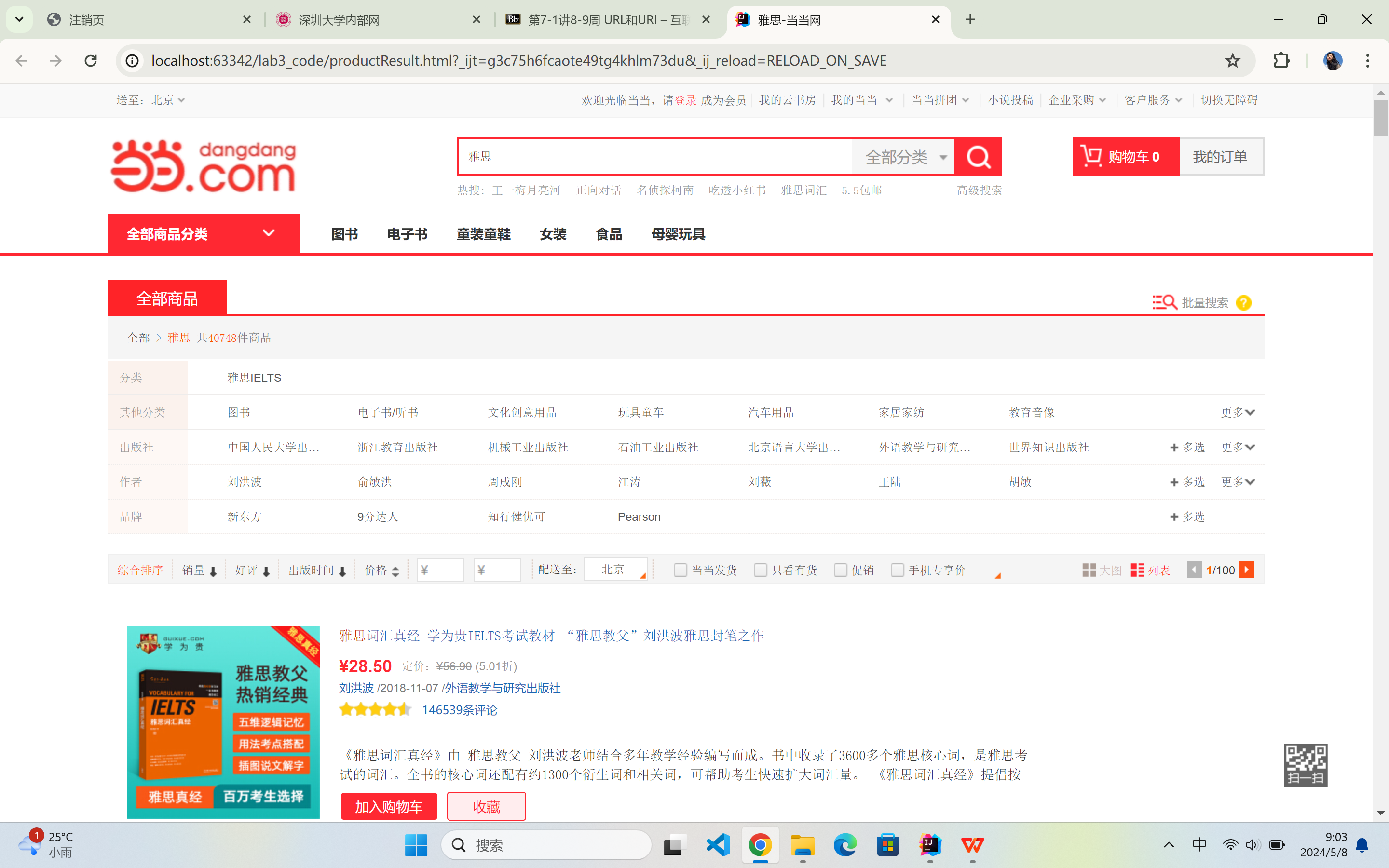


图10 获取的网页在浏览器中的概览

三、实验总结

（此处写你的过程，比如遇到的错误，以及解决方法，你的所想、所得）

**问题1：**使用IP地址创建InetAddress实例对象时，无法获取对象的域名（主机名）信息

**原因：**许多网站服务器的 IP 地址都有对应的主机名或域名，但是并非所有的 IP 地址都设置了反向解析（PTR 记录），所以无法通过 IP 地址直接获取到域名。此外，有些网站可能会配置防火墙或安全设置，禁止从外部查询到服务器的详细信息，包括域名。

**问题2：**在要求3分析某url网址时，使用URL类的getPort（）方法获取不到端口号

**原因：**url.getPort() 方法返回的是 URL 中指定的端口号。如果 URL 中没有明确指定端口号，则返回 -1。在解析 URL 的过程中，如果端口号为空，则会使用默认端口号，比如 HTTP 协议默认端口号为 80，HTTPS 协议默认端口号为 443。因此，如果 URL 中没有指定端口号，url.getPort() 返回 -1 是正常的。

**问题3：**将当当网搜索页面的网页源码保存到本地文件时，出现了中文乱码现象

**原因：**当当网的网页编码格式是“GB2312”，而IDEA、输入输出流的默认编码方式是“utf-8”，编码和解码方式不同会导致乱码。

**解决方案：**在输入“搜索关键词”之后，需要指定关键词string的编码方式为gbk，并且在创建字符流对象时指定编码方式为“GB2312”，这样一来，控制台的中文乱码问题便得到了基本的解决。

四、成绩评定

指导教师批阅意见：

成绩评定：

指导教师签字：毛斐巧

年 月 日