**深 圳 大 学 实 验 报 告**

|  |
| --- |
| **课程名称 互联网编程**  **项目名称 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 计算机科学与技术**  **指导教师 周宇**  **报 告 人 梁润宇 学号 2021220003**  **实验时间 2023.05.25**  **提交时间 2023.05.25** |

**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. 编写域名解析程序（30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。

1. 编写下载功能程序（40分）：

使用URL类，编写一个可重用的下载程序模块，完成HTML和图像文件的下载。即用户输入一个URL，程序会将URL所指向的网络资源下载到本地，并保存到指定磁盘位置。

1. 编写断点续传下载功能程序（30分）：

选择网络上一个较大的文件（如10MB以上），并继续完善2中的文件下载程序，使其具有“断点续传”功能，改写成为一个可重用的具“断点续传”功能的可重用程序模块。

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。**

思路：

首先，创建一个Scanner对象来读取用户的输入。

然后，进入一个死循环while (true)，表示程序会一直执行循环中的代码，直到遇到跳出循环的条件。

在每次循环开始时，程序会提示用户输入域名，并通过scanner.nextLine()方法读取用户输入的域名，并将其赋值给domainName变量。

接着，通过条件判断，检查用户输入的域名是否等于"exit"。如果用户输入的是"exit"，那么使用break语句跳出循环，程序退出。这样，用户有了退出的选项。如果用户输入的不是"exit"，那么将执行resolveDomain()方法来解析用户输入的域名。

在resolveDomain()方法中，使用InetAddress类的getByName()方法来获取域名对应的InetAddress对象，即IP地址。然后，通过调用getHostAddress()方法，将IP地址以字符串的形式返回。

解析成功后，会打印出解析后的IP地址。如果解析过程中发生了UnknownHostException异常，表示无法解析域名，会捕获该异常并打印出错误信息。

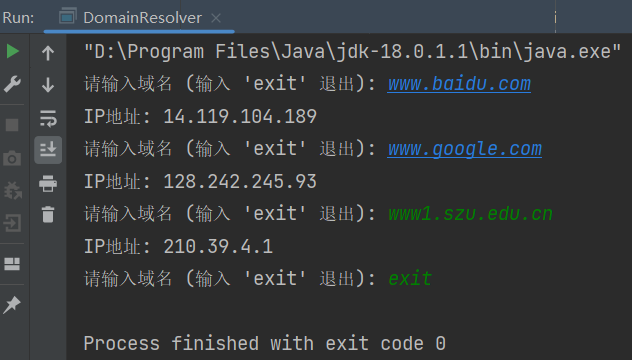
循环会继续下一轮迭代，程序会再次提示用户输入域名，重复执行上述逻辑。

当用户输入"exit"后，循环会被中断，程序退出，关闭Scanner对象，释放资源。

代码：



运行结果截图：

****

此域名解析程序运行顺利，成功将用户输入的域名解析为IP地址

**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。**

**思路：**

当用户输入一个 URL 后，首先下载该 URL 对应的 HTML 文件，并将其保存到指定的目录。这一步通过调用 `downloadFile` 方法实现。

使用 Jsoup 解析下载的 HTML 文件。使用 `Jsoup.parse()` 方法将 HTML 文件转换为 Jsoup 的 `Document` 对象。已知HTML文件包含两类引用，img 标签和 css 内引用的背景图片。

使用 `doc.select("img")` 选择器来获取所有的 `img` 标签元素，并使用 `Elements` 对象来存储这些元素。创建了一个 `AtomicInteger` 类型的计数器 `counter`，用于生成图片的递增序号。

然后，遍历 `images` 中的每个 `img` 标签元素，并获取其 `src` 属性值。这个属性值可能是相对路径或绝对路径。使用 `absUrl("src")` 方法来获取其绝对路径。

调用 `downloadImage` 方法来下载图片。在这个方法中，使用 `URL` 和 `HttpURLConnection` 类来建立与图片 URL 的连接，并下载图片数据。使用 `counter` 来生成递增序号，并将其格式化到图片的文件名中。

在解析 HTML 文件的过程中，还获取了所有的 CSS 文件链接（使用选择器 `link[rel=stylesheet]`），并将它们存储在 `cssLinks` 中。

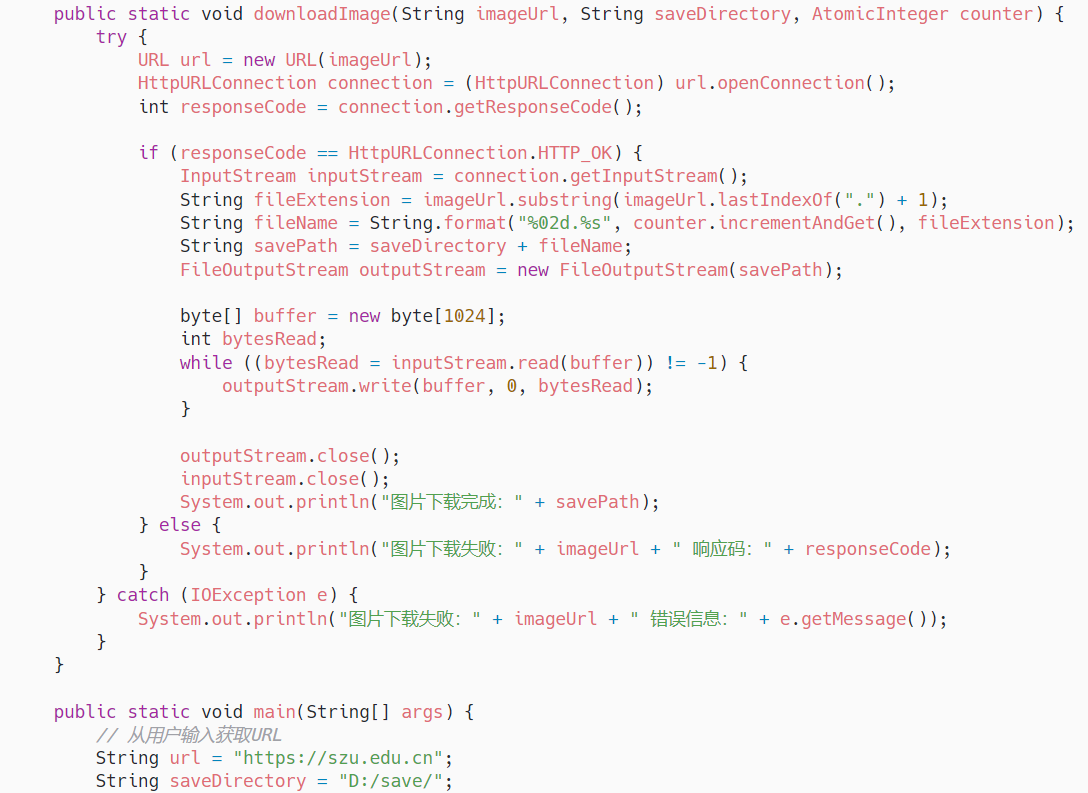
接下来，遍历 `cssLinks` 中的每个链接，并使用 `Jsoup.connect()` 方法连接到 CSS 文件。然后，使用 `cssDoc` 来解析 CSS 文件，并使用选择器 `[style\*=background-image]` 获取所有具有背景图片的元素。

在这个过程中，遍历了每个具有背景图片的元素，并从其 `style` 属性中提取出图片的 URL。然后，调用 `downloadImage` 方法来下载这些背景图片，同样使用 `counter` 来生成递增序号，并将其格式化到图片的文件名中。

这样，就完成了对 HTML 文件中 `img` 标签和 CSS 文件中背景图片的下载。图片将按照递增的序号命名并保存到指定的目录中。

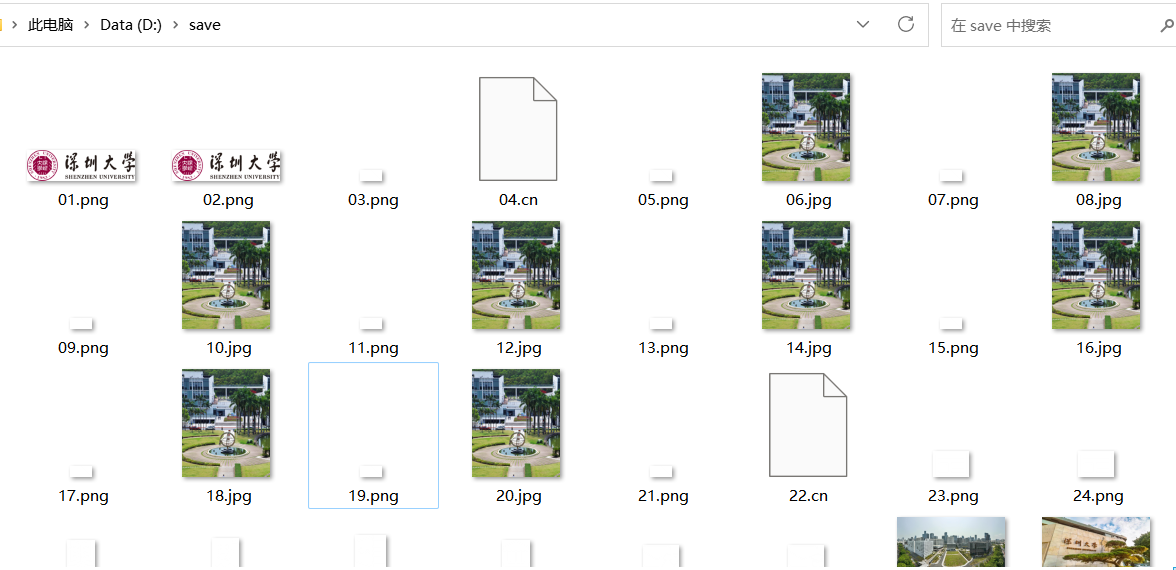
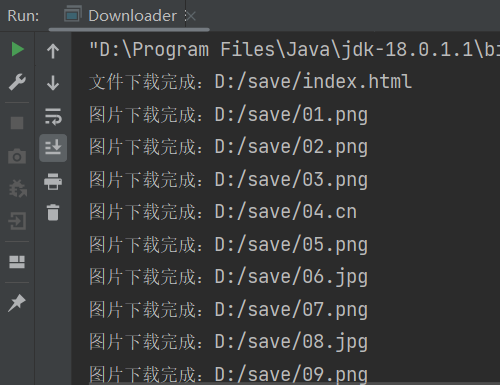
**代码截图：**







**运行结果截图：**



**3.按内容要求3的程序源码及运行结果，简述思路或实验过程。**

思路：

用户提供文件的 URL 和保存文件的目录作为输入参数。

检查本地目录是否存在要下载的文件，如果文件已存在，设置请求头的 Range 属性，告诉服务器从已下载的文件字节末尾开始继续下载。如果文件不存在，则从文件的起始位置开始下载。

发送 HTTP 请求，并获取服务器的响应状态码。检查响应状态码。如果状态码为 206 Partial Content，则表示服务器支持断点续传，可以开始下载文件。

获取响应头中的 Content-Length 字段，确定要下载的文件的总大小。创建本地文件对象，并准备写入下载的文件内容。如果文件已存在，将输出文件流定位到已下载部分的末尾，以便继续写入文件。

获取输入流，读取服务器返回的数据，并写入到文件中。更新已下载的文件大小，以便在下次下载时设置正确的 Range 属性。关闭输入流和输出流，完成文件下载。

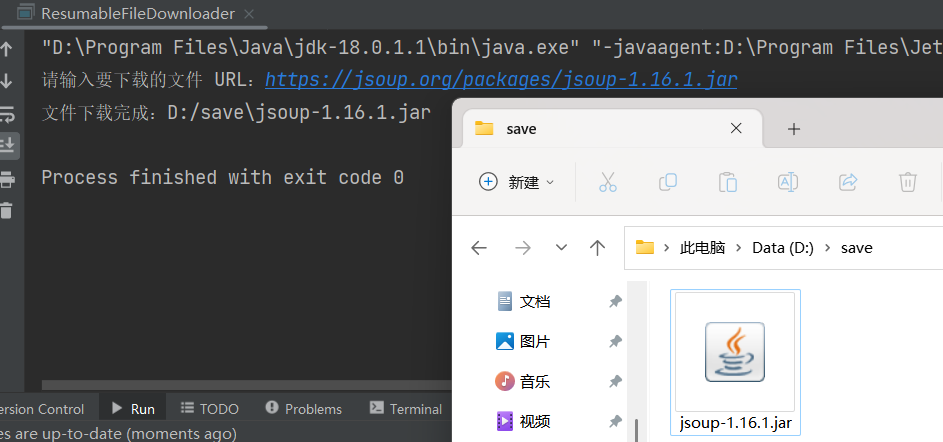
实现**断点续传**的关键在于通过设置请求头的 Range 属性，服务器会返回指定范围的文件内容，从上次下载中断的位置继续下载。下载过程中，不断更新已下载的文件大小，以便在下次下载时设置正确的 Range 属性。

代码截图：





运行结果截图：



成功下载文件。

与此同时，我还试验了<https://download-cdn.jetbrains.com/idea/ideaIC-2023.1.2.exe> 的下载，这个文件有600多MB。下载到20MB时，我退出程序。重新打开程序，可以在文件管理器内看到文件的大小先是保持20MB不变，然后增大（开始从服务器下载新内容），实现了断点续传。由于动态过程不好表示，没有截图。

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）

本次实验共有3个任务。

下载程序模块：这个程序利用 Java 的 URL 类实现了从指定 URL 下载 HTML 和图像文件的功能。它使用 HttpURLConnection 发起 HTTP 请求，并通过输入流将文件内容写入本地文件。这个程序对网络请求和文件处理有了更深入的理解。

解析 HTML 并下载其中的图片：这个程序使用了 Jsoup 库，通过解析 HTML 文件并提取其中的 img 标签，实现了下载其中图片的功能。它展示了如何使用 Jsoup 解析 HTML，并通过提取 URL 并下载图片文件来完成任务。这个程序让我了解了如何使用第三方库来处理 HTML 内容。

具有断点续传功能的文件下载程序：这个程序的关键在于通过设置请求头的 Range 属性，从指定的字节位置继续下载文件。我深入理解了断点续传的概念和实现原理。

这些作业帮助我加深了对 Java 网络编程和文件处理的理解，并提高了我的编程技能。我学会了如何利用 Java 提供的类和库来处理网络请求、HTML 解析以及文件下载等任务。此外，这些作业也让我更加熟悉了网络传输中的一些概念，如请求头、状态码和断点续传等。

|  |
| --- |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。