面向对象程序设计（Java）

实验报告

实验六 浏览器网络程序设计实验

|  |  |
| --- | --- |
| 学院名称 | 计算机科学与技术学院 |
| 专业班级 | 信安1802 |
| 学生姓名 | 侯天铭 |
| 学号 | 5120182377 |
| 指导教师 | 刘涌 |

2019年 4月

**一、实验目的**

练习URLConnection网络访问方法，理解TCP/IP应用层网络程序的设计思想与实现方法。

**二、实验内容**

1、仿照IE浏览器，设计一简单的网站访问程序，实现如下功能：

（1）具有类似IE的简单界面，使用者能输入URL地址，获得并显示输入URL的HTML源代码；

（2）将获得的HTML源代码保存到指定文件中。

（3）从获得的HTML源代码中，选取并输入一个图片(.jpg,.gif,.png等类型)链接，从WWW服务器获取该文件并下载保存到一个文件中。

**三、概要设计**

1. **需求分析过程**

**用户对界面的需求：**本身对于一个浏览器而言，整体的框架也好，内部的各种功能按键也好，用户不仅对其的需求多，而且要求高。但是本次实验并不是要设计一个具有优美的界面和代码解析功能的浏览器，只需实现：能输入url，能进行“获取源代码”和“获取图片”的操作，以及显示源代码的功能即可。从这个角度出发，结合IE界面的特性，首先在界面顶部设置一个横向文本框用以输入url，框后分别有两个按钮，一个是“获取源代码”，一个是“获取图片”，然后往下是一个大的空白的界面（即浏览器用以显示解析完成html代码后所呈现出的结果的界面），在此就用来显示请求url对应页面的源代码。

**用户对使用过程的需求：**由于功能简单，所以对于使用过程的需求也不复杂：用户输入一个页面的url，点击“获取源代码”，下方空白区域将显示出对应源代码，并自动将该代码保存在默认位置（已设置好）。当然，删除输入框中当前url，输入另一个url后，再次点击“获取源代码”，首先要将当前内容（及上次请求url的源码）清空，然后再显示出此时请求页面的源代码；用户输入一个图片对应的url，点击“获取图片”，则自动将该图片保存在默认位置。

1. **总体设计方案**
2. **界面设计**

按照上述用户对界面的需求，我们需要提供一个输入框输入url，以及两个按钮，分别完成“获取源代码”和“获取图片”的功能，还有一个显示框用以显示源代码。要实现这样的界面，首先需要两个容器，一个panel，一个scrollpane。Panel内部首先放置一个textField，用以输入url，然后在此panel的右方再放入一个panel，并在新增panel内加入两个按钮。最后在scrollpane中放入textArea用以显示源代码（源代码很长，故需要滚动条）。

1. **事件设计**

本程序设计的事件只有两个：“获取源代码”和“获取图片”。

**获取源代码：**该事件完成的功能有二，一是将用户输入的url对应的页面源代码显示在textArea内；二是自动将该源代码保存到已设置好的默认位置。只不过在第二次及之后点击“获取源代码”时，首先要将前一次的源码清空，再显示本次源码。

**获取图片：**用户输入一个图片的url，点击“获取图片”，则将该图片保存在事先设置好的位置。

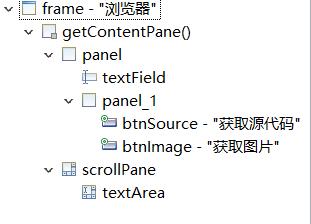
**四、详细设计**

**1、类的设计**

该程序只有一个类，即利用WindowBuilder创建的类。在Design界面中把相应的容器、组件填充完毕，设计好图形界面即可。唯一一点需要手动修改代码的地方，就是将textArea定义成属性，扩大作用域，才能在事件中对其进行修改。接着完成相应事件的设计即可（后续）。

**2、界面设计**

窗体内首先是两个容器，一个panel，一个scrollPane。panel内部首先放置一个textField用以输入url，然后再放置另一个容器panel\_1，接着在panel\_1内部放置两个Button按钮“获取源代码”“获取图片”。最后在scrollPane容器中放入textArea。层次结构如下：



1. **事件设计**
2. **获取源代码：**

首先明确该事件的功能：一是用户输入url，点击“获取源代码”，textArea中显示出用户输入的url对应页面的源代码（输入之前先清空内容）；二是将当前显示的源代码保存到指定位置。那么其对应的工作流程为：首先获取用户输入的url，作为参数传入URL类的构造方法来创建URL对象，并创建输入流，建立udp连接。然后创建File对象，指定默认路径，并创建FileWriter对象。接着先清空textArea中的内容，然后从输入流源读取源代码（字符串），显示在textArea中。最后再调用FileWriter对象fw的write()方法将textArea.getText()写入到指定文件中。

1. **获取图片：**

该事件的功能为：用户输入一个图片的链接，点击“获取图片”，该图片就会被保存在指定位置。那么首先，还是需要获取用户输入的url，然后建立udp连接，创建输入流，也要创建File对象，只不过这里因为是要读图片，所以不采用上面的字符流FileWriter，而是使用FileOutputStream字节流来读取。所以创建FileOutputStream对象，然后创建一个字节数组，大小定为1024，即允许一次性读1kb。使用read(byte[] b)方法，读取数据并返回读取字节长度，然后使用write(byte[] b,int off,int len)方法写入数据。接着再次读取、写入，直至读取到文件末尾，返回-1，停止循环，实现图片的保存。

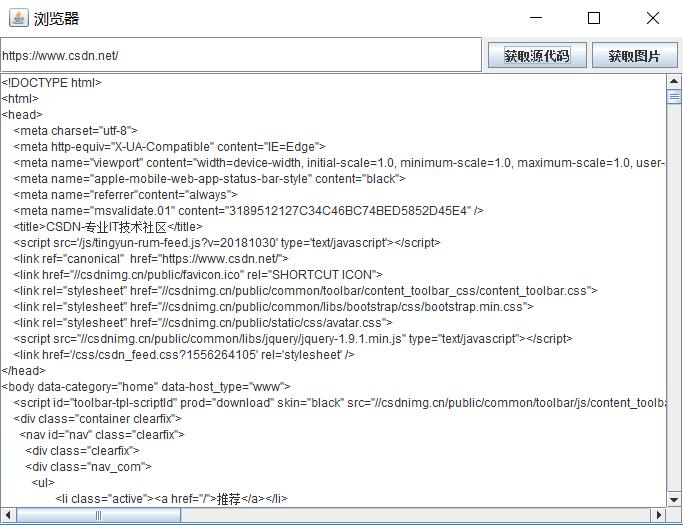
**五、实验结果**

**1、“获取源代码”测试：**

本次测试选取了三个网站：

1. **CSDN：**

程序界面：



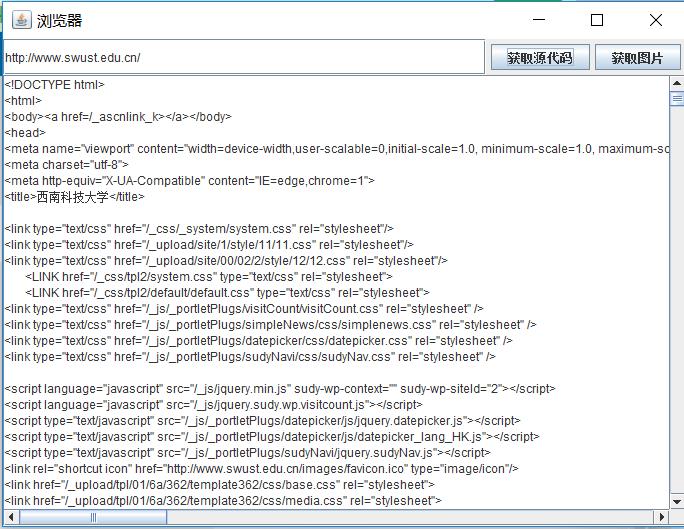
保存文件:





1. **学校官网：**

程序界面：



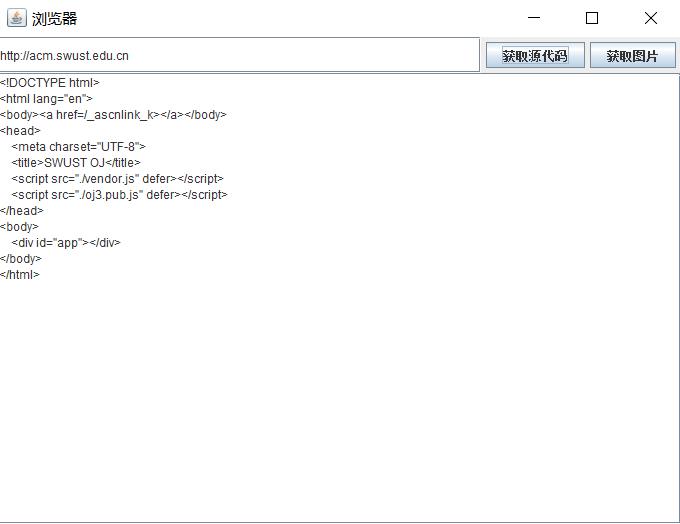
保存文件：



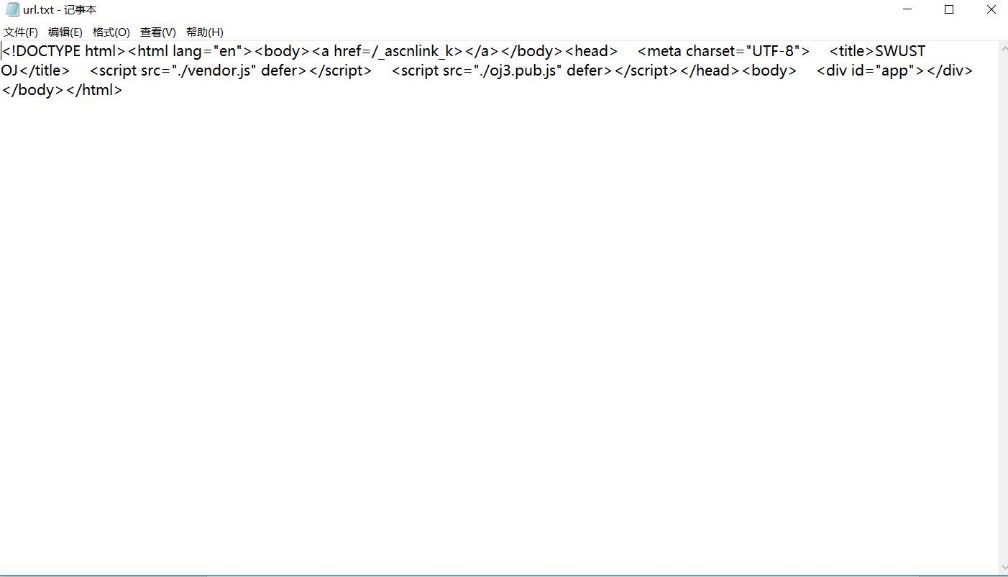


1. **学院oj：**

程序界面：



保存文件：  

测试说明，该程序能正确获取相应url对应页面的源码，并将其保存在指定位置。

**2、“获取图片”测试：**

**（1） 学校官网：**

图片url：

<http://www.swust.edu.cn/_upload/tpl/01/6a/362/template362/images/logo_1.png>

结果



打开后：



**（2） CSDN：**

图片url: <https://csdnimg.cn/feed/20190428/3975b51c2113d06c714040862ba0d773.jpg>

照样保存了下来：





测试说明，该程序能将某服务器上的指定图片保存下来。

**六、讨论与分析**

**对课程知识点的理解：**事实上，本次实验主要涉及到的内容仍然是I\O流的问题，当初在做IO实验的时候，我对此的理解并不是特别深入，但是在这次实验中确实加深了我对其的认识，对于字节输入\输出流，字符输入\输出流分别应该在什么时候使用有了更清楚的认识。主要表现在：在课堂上，老师带领我们写出的程序在读取源代码的时候会出现中文乱码的情况，在当时我确实是没有头绪，不知道如何进行改进。下来以后，查了查资料，才明白了这个简单的道理：课上使用的是InputStream字节输入流，也就是按照字节读取，但一个汉字并不只是一个字节，因此使用字节输入流读取会出现乱码不奇怪。于是按照网上的思路，我最开始是把InputStream转换成了InputStreamReader字符输入流，但是发现这个流并没有readLine()之类的读取字符串的方法，于是又联想到缓冲流。由于其包装的必须是Reader对象，所以还是得把IbputStream转换为InputStreamReader，然后使用BufferedReader包装一下InputStreamReader得到BufferedReader对象，使用其readLine()方法进行读取，最后发现解决了中文乱码问题。

**对设计过程的理解：**对于设计过程，界面外观自不用说，按照浏览器的外观做就行。事件的话关键还是得理解其工作流程，比如说，用户是先输入url再点击“获取源代码”，那么程序就应先获取用户输入的url，然后建立udp连接，创建输入流，读取数据，显示到界面上。或是在保存图片时，因为涉及到文件操作，所以创建File对象，然后读取、写入。所以在设计程序的时候，很重要的是首先明确目的，然后思考为了达到这个目的需要干些什么，比如说需要创建什么对象，需要使用什么方法之类。

**实验结果的分析：**从上述测试步骤可以看出，该程序还是基本达到了实验要求，能够实现获取源代码，保存源代码以及保存图片的功能。

**需要引起注意的问题：**就功能上说，该程序已经达到了实验要求，但是还有一点需要注意，就是每次保存的源代码路径和图片路径都是相同的，也就意味着每一次的保存都会覆盖上一次保存的内容。如果要避免这种情况，那么一是在每次保存的时候都重新创建一个文件对象，指向不同的路径，显然特别麻烦；另一种方法就是使用弹窗，由用户选择想要保存的路径，这种方式显然更易实现且方便。

**七、附录：关键代码**

**（1）“获取源代码”事件：**

String s = textField.getText();

URL url = **new** URL(s);

InputStream is = url.openStream();//建立udp连接

InputStreamReader isr = **new** InputStreamReader(is); //转换输入流

BufferedReader bur = **new** BufferedReader(isr); //采用缓冲流包装

File f = **new** File("D:\\url.txt");//指定源代码保存路径

FileWriter fw = **new** FileWriter(f);

textArea.setText("");

String s1 = bur.readLine();

**while**(s1 != **null**) { //显示源码

textArea.append(s1+"\n");

s1 = bur.readLine();

}

String s2 = textArea.getText();

fw.write(s2); //输入源码

**（2）“获取图片”事件：**

String s = textField.getText();

URL url = **new** URL(s);

InputStream is = url.openStream();

DataInputStream dis = **new** DataInputStream(is);

File imageFile = **new** File("D:\\logo-5.png");//指定图片保存路径

FileOutputStream fos = **new** FileOutputStream(imageFile); //创建输出流

**byte**[] b = **new** **byte**[1024];

**int** num = dis.read(b); //读取数据

**while**(num != -1) {

fos.write(b, 0, num); //写入数据

num = dis.read(b);

}