

ICT 4361 — Java Programming Exercise 7

Purpose

This exercise familiarizes you with the basic methods of the networking APIs in Java, and simple tasks such as reading the contents of a URL, and writing on a socket. This will also exercise your knowledge of Exceptions.

What To Hand In:

Please hand in a listing for each program requested, formatted in an easy-to-read style.

Ensure your name, and the name of the file is available in a comment at the top of the file.

Also, ensure that you have a sample of the output from the program.

If your program fails to compile, hand in your error listing as your output.

For each question asked, provide one or two sentences summarizing your answer. Please be both complete and succinct.

Problems

For the last problem, you will want to download [hw7.jar](#) This file contains the class files and source for a server process.

1. Create a class called ReadURL which has the following features:

- A private field of type URL which stores a URL
- A constructor which takes a String as a parameter, from which the URL is created
- A constructor which takes a URL as a parameter, from which the URL is assigned
- A constructor which takes a URI as a parameter, assigning to the URL with the toURL() method

Note that there are exceptions which may be created as a result of these assignments, which you should handle in the constructor

- A method called getContent() which returns a String, and performs the following:
 - a. Creates and assigns URLConnection by invoking openConnection() on the URL
 - b. Creates an InputStream from the URLConnection
 - c. Creates a BufferedReader on a new InputStreamReader built from the InputStream
 - d. Reads one line from the BufferedReader and stores it as the return value

- e. Closes the `BufferedReader`
- f. Returns the contents of the line
- A method called `getContentAsURI` which returns a URI, by calling `getContent()` and using the line of text to construct a URI for return.

Note that Exception handling is necessary in these methods as well.

- Create a main method that creates a new `ReadURL` from the URL `http://www.du.edu/~mschwart/ICT4361/hw7.problem1`, gets its content, and then prints the URL, and its content.
 - Run the program
2. Create a new class with just a main method to do the following:
- Creates a new `ReadURL` from the first parameter on the command line, or read from the user
 - Uses that `ReadURL` object to get the contents at that location as a URI
 - Converts that URI to a URL (using the `toURL` method)
 - Creates another `ReadURL` object using that URL
 - Uses the new `ReadURL` object to get the contents at that location as a String
 - Prints the first URL, second URL, and the contents of the second URL
 - Runs the program for the URL `http://www.du.edu/~mschwart/ICT4361/hw7.problem2`
3. Create a new class called `WriteURI` with the following features:
- A private field of type URI
 - A private field of type `Socket`
 - A private field of type `BufferedReader`
 - A private field of type `PrintWriter`
 - A constructor that takes a URI parameter, sets the URI field with it, and calls method `openConnection`
 - A constructor that takes a String parameter, constructs the URI field with it, and calls method `openConnection`
 - A method called `openConnection` that:
 - a. Sets the field of type `Socket` by opening a new `Socket` using the URI field's host and port
 - b. Sets the field of type `BufferedReader` to a new `InputStreamReader` constructed from the `Socket` field's `getInputStream()` method
 - c. Sets the field of type `PrintWriter` to a new `PrintWriter` constructed from the `Socket` field's `getOutputStream()` method, and supplying true for *autoflush*
 - A method called `sendMessage` that takes a String parameter, and uses `println` to send it on the `PrintWriter` object.
 - A method called `receiveMessage` that returns a String which it gets by calling `readLine()` on the `BufferedReader`
 - A main method that creates a `WriteURI` from the URI found in `hw7server.uri`

Run the SimpleServer (provided in the hw7.jar) before doing the last step

- Runs the program, sending your name as the message to sendMessage, and printing the reply from receiveMessage

Notes

1. Exception handling is an important part of this exercise
2. To run the SimpleServer from the command line in Windows, do the following:
 - Open a command prompt window
 - Ensure you can run Java by entering the command `java -version`
 - Run the command line `java -cp hw7.jar hw7.SimpleServer hw7server.uri`
This will make the server run, placing its URI into the file `hw7server.uri`
 - The file `hw7.log` contains the record of connections to the server
3. You may use your ReadURL class to get the server host and port information, or you may examine the file, and then code its contents into your program
4. Note that `hw7.jar` contains the source for the server, as well as the class files, so you may examine this if you wish

Evaluation Criteria:

Criteria Weight

Problem 1 35

Problem 2 30

Problem 3 35