CMPSC 112
Introduction to Computer Science II
Spring 2014

Laboratory Assignment Three: A First Step Towards Implementing a Twitter Client

Introduction

Twitter is a commonly used social media network. As we refamiliarize ourselves with the fundamental characteristics of the object-oriented programming paradigm and continue to learn more advanced features of the Vim integrated development environment (IDE), we will take the first steps towards implementing a complete Twitter client. Along with learning how to use, enhance, and extend a Java class, we will also review the steps that you must take to perform console input and output. This laboratory assignment represents the first of a two-part series of assignments.

Accessing and Understanding GatorTweet

To start this laboratory assignment, you should return to the `cs112S2014-share` Git repository and type the command `git pull` in the terminal window. Now, you should have a `lab3/GatorTweet/` directory that you can explore further. Where is the Java source code in this new directory? Why is it stored in the directory where you found it? Now, you should use GVim to study the source code in the `build.xml` file. What do you think is the purpose of this file? For this assignment, you can type the command `:make` in your GVim window and it will compile the `Tweet.java` file and save the bytecode in the correct subdirectories inside of the `bin/` directory.

At this point, you should purposefully insert an error into the `Tweet.java` file. For instance, you can open the file in GVim and then remove one of the semi-colons at the end of a source code line. Now, type `:make` again in GVim. What output do you see on the screen? GVim can provide you with a useful listing of these errors if you type `<,q>` and then navigate around this "Quickfix List". When you want to zoom to a location in your Java program that contains an error, you can find it in the quickfix list and then press enter. After you have returned to the error location, go ahead and fix the problem by adding back the semi-colon and recompiling your program.

Improving the GatorTweet System

You will notice that the `Tweet` class contains two instance variables and several methods. First, you should add proper JavaDoc comments to all of the variables and methods in `Tweet.java`. Once you have finished adding in all of the necessary comments, please notice that the `isValidMessage` method is not correctly implemented. You should enhance this method so that it returns `true` when a candidate message is valid and `false` otherwise. This method should classify a message as valid as long as its length is greater than zero and less than or equal to 140.

Now, you should implement a `GatorTweet` class that performs the following operations in `main`:

1. Constructs an `ArrayList` that will only hold `Tweet` objects for the valid `Tweets`.

2. Constructs an `ArrayList` that will only hold `Strings` for the invalid tweet messages.

3. Uses the `java.util.Scanner` call to read in consecutive lines of text from a file in the `tweets/` subdirectory of the `GatorTweet/` directory, called "Tweets.txt", that contains potential tweets. (It is worth noting that this file can contain both valid and invalid tweets).
4. Checks to see if the current tweet message is a valid one or not. If the message is valid, then it should construct a new `Tweet` and store it inside of the `ArrayList` for the valid instances of the `Tweet` class. If the message is invalid, then it should not construct a `Tweet` and instead store the `String` message in the `ArrayList` for invalid tweet messages.
5. Outputs all of the `Tweet`s stored in the `ArrayList` for the valid tweet messages.
6. Outputs all of the `Tweet`s stored in the `ArrayList` for the invalid tweet messages.

In the next laboratory assignment we will finish implementing a `GatorTweet` program that can interact with Twitter. For instance, we will add features that can authenticate a user with the Twitter servers, display a user's Twitter timeline, and post a `Tweet` to the Twitter site.

Summary of the Required Deliverables

This assignment invites you to submit one printed version of the following deliverables:

1. A description of the meaning and purpose of the provided `build.xml` file.
2. A screenshot demonstrating that you can display and use the quickfix list in GVim.
3. The final version of your `Tweet.java` and `GatorTweet.java` files.
4. The output associated with running `GatorTweet` with a sample "Tweets.txt" file.

Along with turning in a printed version of these deliverables, you should ensure that everything is also available in the repository that is named according to the convention `cs112S2014-<your user name>`. Please note that students in the class are responsible for completing and submitting their own version of this assignment. While it is acceptable for members of this class to have high-level conversations, you should not share source code or full command lines with your classmates. Please see the instructor if you have questions about the policies for this laboratory assignment.