CSci 1933 Lab 3

September 29, 2015

Note For the purpose of this lab, you cannot use anything from java.util (i.e. no collections)

1. Introduction

The purpose of this lab is to have you interact with a data file, and begin using loops and arrays in order to process the data that you will be reading.

2. What you need to do: In general

In order to ensure that everything will be setup properly:

- Create a new Project in IntelliJ called “Lab3Project”.
- Inside the src directory of your project, create a new class called Lab3.java and create the public static void main(String args[]) method in it.
- Also be sure to import:
    - TwitterReader.java into your src directory.
    - tweets.dat into your project directory.

3. What you need to do: More Specifically

- For this lab, imagine back to when you were considering which University to attend. How did you gather information about what it would be like to live in the area where the University was located?
- Suppose you're going through that process again, but this time, you know that Twitter has its social network data publicly accessible. Many people tweet about all sorts of things, and you're interested to see what people are saying about the University of Minnesota in order to get a feel for what people are saying, as well as how tech savvy people at the University of Minnesota are (yes, for the purposes of this lab, assume that using Twitter indicates how tech savvy people are).
- For the purposes of the lab, we've simplified the need to read directly from Twitter by providing you a data file of tweets that contain the words "Twin Cities" or "gophers" (make sure you match the cases of the letters).
- We've also provided you a java class (TwitterReader.java) that allows you to read from the data file, without needing to worry about the logistics of doing the actual reading.

- You've heard about some functionality that Twitter has called Twitter Lists which allow for Twitter users to be grouped together, and apparently someone else has had a similar thought to you. They've pulled together a Twitter List of people in MN, who seem to tweet about things relevant to the University of Minneosta. The TwitterReader java class provides a mechanism to get the usernames of the people on this Twitter List, called getMNTwitterList().
- You've decided that the best way for you to get a sense of what people are saying about the University of Minnesota is to pick the first tweet by each of the people in the Twitter List, as well as some overall information about each user: how frequently everyone in the data file tweets about the gophers and the Twin Cities, as well as how many of the overall tweets are generated by the users on this Twitter List.
- More specifically, in order to complete this lab, you should:
    - ✓ Count how many tweets out of the entire data file, contain the phrase "Twin Cities" (case sensitive)
    - ✓ Count how many tweets out of the entire data file contain the word "gopher" (case sensitive)
    - ✓ Calculate the percentage of tweets produced by all the users on the Twitter List out of the total number of tweets in the data file.
- Although your numbers and percentages will be different, sample output for this lab will look like:

```
Twin Cities: 53
Gopher: 18

Cargill - 4%
nine2fives - 2%
TeacherJobMN - 2%
UMMorris - 6%
Jessy__Fox - 8%
TimPostMPR - 4%
```

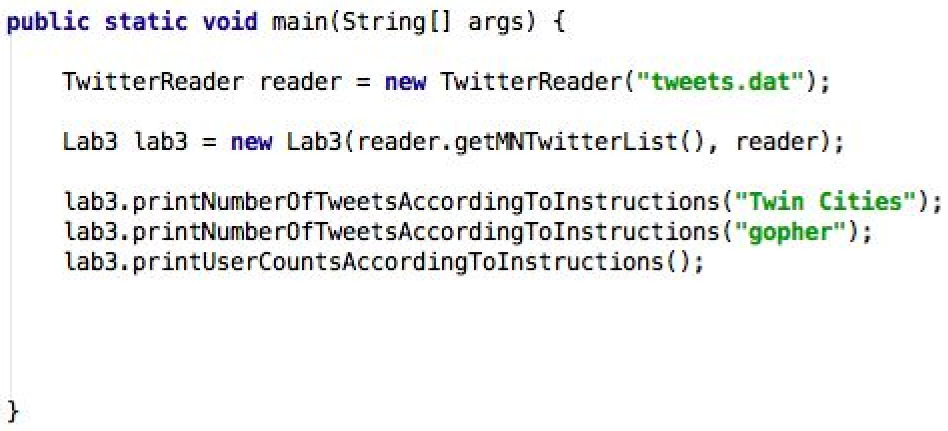
4. Creating new objects

- In order to do anything with most classes, you must instantiate them; that is, create a new object from them. In Java, you do this with the new keyword. For example, to create a new object of class Color, you could do the following:

```
Color c = new Color();
```

- The Color() part is just like a method call; you can put arguments in the parentheses. It calls a method called a constructor which does exactly as it sounds, it constructs the object. As an object is created, the constructor method is called, and it will handle any arguments passed, as well as setup any initial state that needs to occur.

- The constructor for the TwitterReader class requires an argument — a string — which is the file name it will read from.
- We'll do this lab in an object oriented way. To achieve this, you need to create a class "User" that is, create another .java file by the name "User.java". The User class should have two fields:
    - ○ username
    - ○ tweetCount
- The main method of "Lab3.java", which we created earlier, should look exactly like this:



```
public static void main(String[] args) {

    TwitterReader reader = new TwitterReader("tweets.dat");

    Lab3 lab3 = new Lab3(reader.getMNTwitterList(), reader);

    lab3.printNumberOfTweetsAccordingToInstructions("Twin Cities");
    lab3.printNumberOfTweetsAccordingToInstructions("gopher");
    lab3.printUserCountsAccordingToInstructions();



}
```

Lab3.class will have two fields:

- o An array of user objects
- o The TweetReader object it should use

It'll also have a constructor that:

- o Initializes the user array and populates it with user objects
- o Initializes the user counts
    - ➢ Hint: make an initializeUserCounts() private method that is called after you initialize the user array.

5. Using the TwitterReader

- The TwitterReader class class allows you to read from a data file, one line (or record) at a time. It has a notion of a "current record", and allows you to get data from the current record. When it is first created, it has no current record.
- The advance() method of TwitterReader advances the current record by 1 record. The first call to advance() makes the first record the current record. It also returns a boolean value — true if it found another record, or false if it reached the end of the file.
- The getTwitterID() method gets the Twitter ID (or username), as a String, from the current record. The getTweet() method returns the tweet text for that particular tweet, but does

not include the username. Neither of these methods can be called before advance() is called at least once, and neither of them can be called after advance() has returned false.

Note If you violate the rules regarding calling advance() and getting record information, your program will terminate with a RuntimeException stating that the tweet reader is not in a valid state.

- Thus, the basic structure of code to process the records in the file looks like this (in pseudocode):

```
while advance() returns true
    process a record
```

- TwitterReader.java also contains a method reset():
  - Tells it to go to the beginning of its file so you can use advance() again

6. Outputting data

- Java provides an object, `System.out`, which lets you print data to the standard output stream (which goes to your console in Eclipse, or to the terminal from which you ran the program if you're using a command line). There are a couple methods of interest here:

  `print`

  This method prints a string or other data object.

  `println`

  This method takes a string (or other data, such as an integer) and prints it, followed by an end-of-line character.

  `format`

  This method takes a string followed by several other parameters and prints the string, after inserting the other parameters in place of certain character sequences in the initial string. It is used for nicely formatting output. See Using format (penultimate section of this document) for more information.

- Look at the `PrintStream` class in the Java API reference for information on more options for outputting data.