CIS 2168: Assignment #4

Due on Friday, September 26, 2014

11:59pm

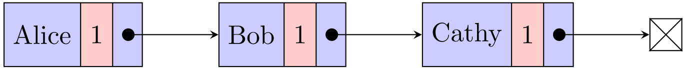
Anwar Mamat

Problem 1

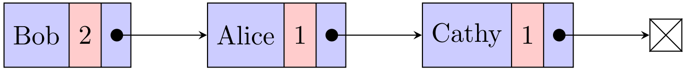
Word Frequency (100%) We have already studied linked lists. In this assignment, we use linked list to count the words from a given text file or URL and print the words by their frequnct order. The iterable class ”Frequency” uses a “Node” to represent each word and its frequency. The Node class is given here:

```
private class Node {
    private E key;
    private int count;
    private Node next;
    Node(E item){
      key = item;
      count = 1;
      next = null;
    }
}
```

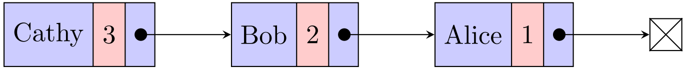
Frequency class can insert words into a linked list. When a new word is inserted, it adds the word at the end of the linked list. If the word exists in the list, we simply increment the value of ”count” for the word. We also keep the words in the linked list by their frequency order. Therefore, when the frequency of a word has changed, we have to move the word to correct position in the linked list. For example: if we insert three words Alice, Bob, Cathy, we will have



If we insert another word, Bob, then we will have



If we insert the word, Cathy twice, then we will have



The Frequency class is partially implemented. **Your job is to implement the “insert” method in the Frequency class. “insert” adds new words at the end of the list, and increment the count for existing word, and move the word to the correct position in the list. DO NOT try to sort the list. When “count” is changed, remove the node from the list, and add it to the correct position in the list.** The main method is provided for you. The jar file “stdlib-package.jar” includes many useful utility classes. The ”In” class in the “stdlib-package.jar” can read strings from a text file or a URL. Do not forget to add “stdlib-package.jar” to your project.

Code examples

To read a webpage at the given URL, and print the content (the html file) of the webpage:

```
In in = new In(urlName);
while (!in.isEmpty()) {
String s = in.readLine();
System.out.println(s);
freq.insertWords(s);
}
```

To read the text file and print the content.

```
In in = new In(fileName);
while (!in.isEmpty()) {
String s = in.readLine();
System.out.println(s);
```

Problem 1 continued on next page. . .

```
freq.insertWords(s);
}
```

The InsertWords method of the Frequency class takes a sentence, split the sentence into words, and calls ”insert” method of the Frequency class to insert each single word into the Linked List. The InsetWords method is shown here:

```
public void insertWords(String str){
  String delims = "[ .,?!’\"()}{;/<>&=#-:\\ _]+";
  String[] words = str.split(delims);
  for(String s: words){
  insert((E)s);
}
```

Testing

The main method iterates through linked list of Frequency class and prints MAX number of words from the list. “inputSourceName” is the text file name with full path or the URL for the webpage. We provided text files to test. You can create your test files too. You can also test with URLs. **pay attention to the corner cases, where words are inserted at the start of the linked list, at the end of the linked list. Create small text files, test your code with small files. Test with bigger files such as “War and Peace” when you feel confident with your program.**

```
MAX=10;
inputSourceName="einstein.txt";
output:

readLine() from einstein.txt
----------------------------------------------------------------------------


 Elapsed time:0.122


the  45
is   44
of   43
to   34
a    30
and  28
in   17
be   15
not  15
that 13
```

```
inputSourceName="test1.txt"
output:

readLine() from test1.txt
----------------------------------------------------------------------------
 Elapsed time:0.043

alice     2
bob  1
```

```
cathy     1
```

```
inputSourceName = "war_peace.txt";
output:

readLine() from war_peace.txt
----------------------------------------------------------------------------

 Elapsed time:13.35

the  31784
and  21142
to   16439
of   14821
a    10084
in   8442
he   8131
that 7803
his  7678
was  7339
```

```
String inputSourceName = "http://cis.temple.edu";
readLine() from http://cis.temple.edu
----------------------------------------------------------------------------

 Elapsed time:0.742

a    266
li   184
strong    136
href 115
style     100
normal    96
margin    89
and  74
px   70
html 69
```

```
String inputSourceName = "http://cis.temple.edu/˜anwar/teaching.html";
readLine() from http://cis.temple.edu/~anwar/teaching.html
----------------------------------------------------------------------------

 Elapsed time:0.208

CIS  38
p    32
a    30
html 17
href 17
h    14
Fall 13
Spring    12
css  6
```

```
Society   4
```

What will you learn from this assignment

1. Reading text files and html files from URLs.
2. Implementing linked lists
3. Iterable and Iterator
4. Inserting, deleting linked list nodes
5. comparing objects

Grading

Homework is 100 points. 80 will reflect functionality and correctness. 20 points on your program will reflect your programming style, documentation. If you code does not compile, you will not receive any credit.

Commenting and Documenting Code

Code must be properly commented. The main idea is that the grader should be able to understand your code easily, not have to tear his or her hair out wondering what some statement is doing. The first time you have to deal with poorly commented code (if you haven't already), you will understand how annoying it is. In particular, the top of each code file should contain your name, the course and assignment numbers, and a brief summary of what's in the file. Line-by-line comments should be included as necessary to make the code easy to read. A clear coding style, together with informative variable and function names, will reduce the number of comments required. Obscure code or cryptic function names will cause loss of points (for bad style) and also require more extensive comments.

What to submit

A single zip file called Assignment4_firstname_lastname.zip, where firstname is your first name, and lastname is your last name. In this zip file, put:

1. Java source
2. A README file with:
   - Instructions to compile and run of your code (include a description of command line options).
   - If your solution is not perfect, explain what parts you did and what part you did not do.
   - List of files submitted
   - All your data and results (in plain text files).
   - Anything else you want TA know
3. Submit this zip file to Blackboard

Credit where credit is due

"stdlib-package.jar", and "War and Peace" are downloaded from "http://algs4.cs.princeton.edu/home/".