Programming Project 2 (Due 10-6-2016 at 11:55 PM)

Write a JAVA program to solve the following problem. Your program should properly compile and run. Your code MUST follow the documentation style used in your textbook. You need to upload into Moodle the following:

- From this point on, each of your projects is expected to be composed of multiple source code files. Thus to organize these files, all of them have to be part of a single package. Your package name MUST match the directory containing your code file. Finally, you need to export your package as a .JAR file then upload this single file into Moodle. If you are not familiar with how to export your code into a .JAR file, please check the following link. http://help.eclipse.org/neon/index.jsp?topic=%2Forg.eclipse.jdt.doc.user%2Ftasks%2Ftasks-33.htm

Very Important Note: Do NOT upload any other files into Moodle; just ONE file with the extension ".jar" that is it.

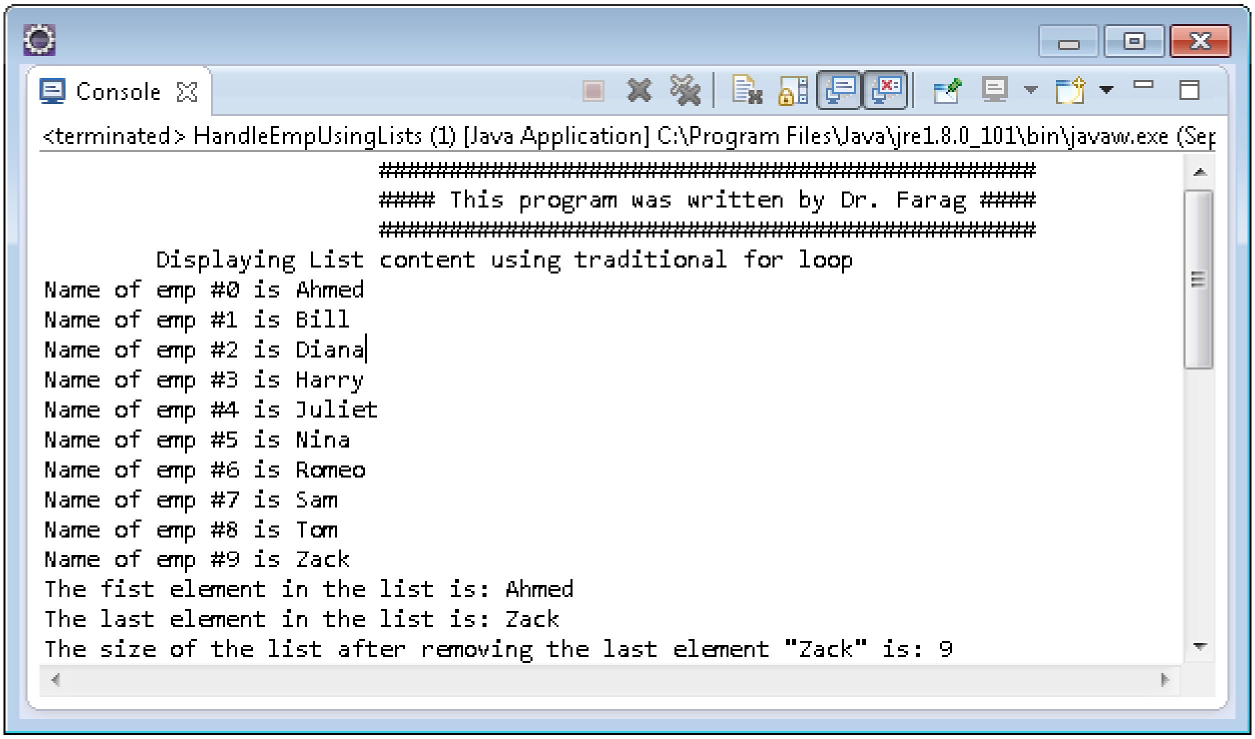
Problem Statement:

In this project, you are asked to write a Java application that handle a list of employees that are employed by a certain company. In your code, you need to demonstrate the ability to effectively use the Java Collection Classes, particularly the LinkedList one and the ability to implement its functionalities (if needed). We will assume that each employee is only represented by her/his first name (just for simplicity and to focus on the use of the data structures themselves). The names of all employees in this small company are stored in a given data file called **empNames.txt** which is posted on Moodle next to this project statement. Your program must do the following task:

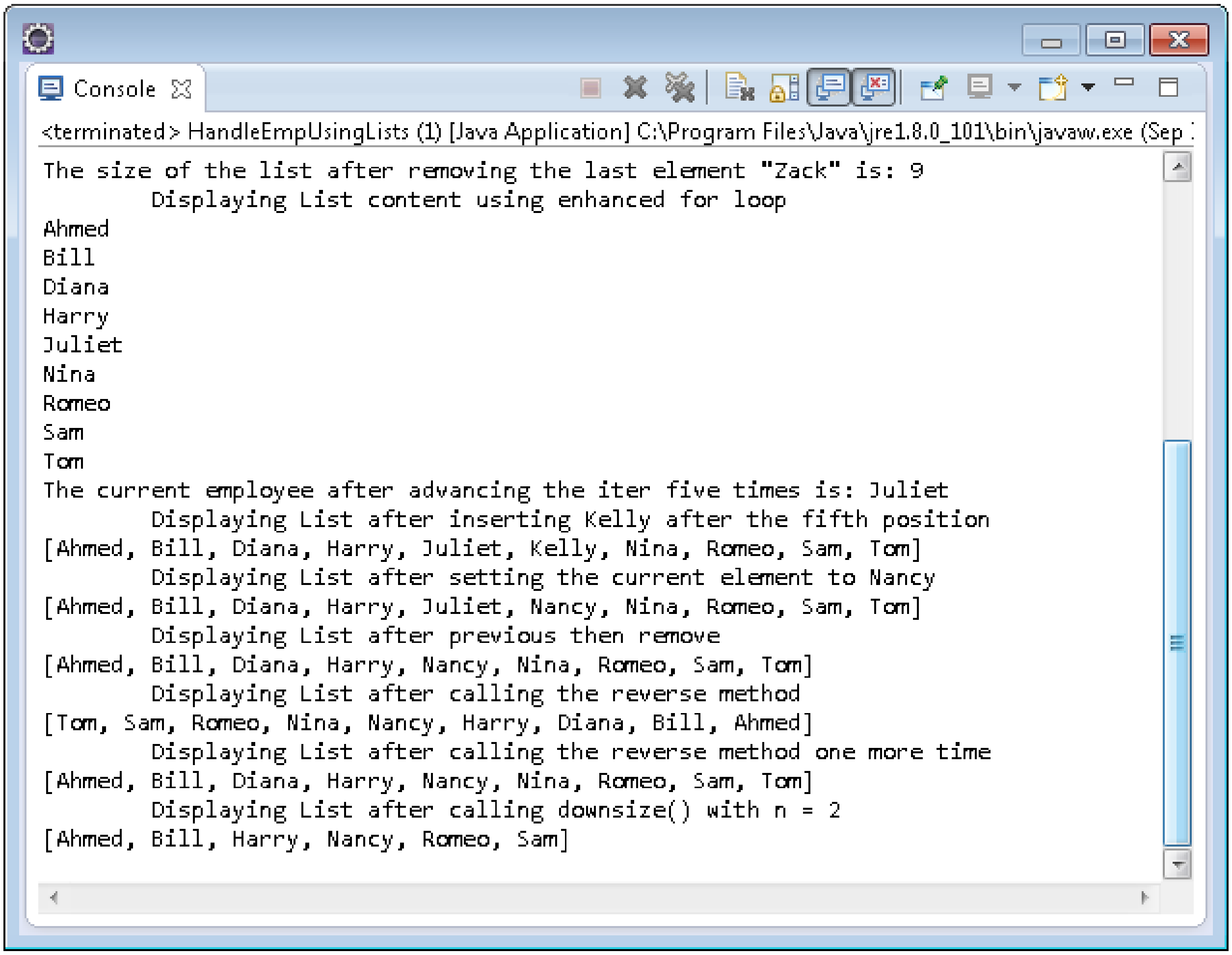
1. Create a LinkedList to be used in storing and handling employees' data.
2. Open the given file and read employee data into the LinkedList. In doing so, your code needs to maintain the order of employee data in the input file.
3. Verify the content of the employee data (just read) by displaying them on the monitor using a traditional for loop along with appropriate output messages.
4. Invoke various methods of the LinkedList class to retrieve then display the first element and the last element of the list. Display appropriate messages on the monitor.
5. Invoke various methods of the LinkedList class to remove the last element then display the size of the list after the removal operation. Display appropriate messages on the monitor.
6. Display the current list contents using the enhanced for loop along with appropriate output messages.
7. Create a ListIterator object, iter, to be used in accessing the LinkedList contents.

8. Advance the iterator for five positions then display the current list item (using iterator method and NOT list method).
9. Insert a new employee called “Kelly” at the current iterator position then display the current list contents using the compact form to be used from this point on for any list content display. Hint, use the implicit toString().
10. Execute the iter.previous() method followed by using the iter.set() method to set the current element to Nancy. Then, display the current list contents.
11. Execute the iter.previous() method followed by the remove() method then display the current list contents.
12. **In a separate file**, create a helper class called HelpLists. This class need to have the code for two methods described below:
    a. public void reverse(LinkedList<String> strings). This method receives a LinkedList as its parameter then reverses the entries in a linked list.
    b. public void downsize(LinkedList<String> employeeNames, int n). This method removes every nth employee from a linked list.
13. In your main class, create an object of type HelpLists class then use it in the following.
14. Invoke the reverse method to reverse the order of the list contents. Display the current list contents.
15. Invoke the reverse method one more time then display the current list contents. Verify that the list is now back to its normal order.
16. Invoke the downsize() method in order to remove all employees at the nth positions where n is 2 in this case. Display the current list contents

Sample output of your code need to look like the following snapshots.



```
<terminated> HandleEmpUsingLists (1) [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (Sep
                        ###############################################
                        #### This program was written by Dr. Farag ####
                        ###############################################
        Displaying List content using traditional for loop
Name of emp #0 is Ahmed
Name of emp #1 is Bill
Name of emp #2 is Diana
Name of emp #3 is Harry
Name of emp #4 is Juliet
Name of emp #5 is Nina
Name of emp #6 is Romeo
Name of emp #7 is Sam
Name of emp #8 is Tom
Name of emp #9 is Zack
The fist element in the list is: Ahmed
The last element in the list is: Zack
The size of the list after removing the last element "Zack" is: 9
```



```
<terminated> HandleEmpUsingLists (1) [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (Sep
The size of the list after removing the last element "Zack" is: 9
        Displaying List content using enhanced for loop
Ahmed
Bill
Diana
Harry
Juliet
Nina
Romeo
Sam
Tom
The current employee after advancing the iter five times is: Juliet
        Displaying List after inserting Kelly after the fifth position
[Ahmed, Bill, Diana, Harry, Juliet, Kelly, Nina, Romeo, Sam, Tom]
        Displaying List after setting the current element to Nancy
[Ahmed, Bill, Diana, Harry, Juliet, Nancy, Nina, Romeo, Sam, Tom]
        Displaying List after previous then remove
[Ahmed, Bill, Diana, Harry, Nancy, Nina, Romeo, Sam, Tom]
        Displaying List after calling the reverse method
[Tom, Sam, Romeo, Nina, Nancy, Harry, Diana, Bill, Ahmed]
        Displaying List after calling the reverse method one more time
[Ahmed, Bill, Diana, Harry, Nancy, Nina, Romeo, Sam, Tom]
        Displaying List after calling downsize() with n = 2
[Ahmed, Bill, Harry, Nancy, Romeo, Sam]
```

(End of Project)