SNU CSE: Data Structures **Due date**: 11 pm Oct 15 2018

Assignment 1

Lecturer: Srinivasa Rao Satti

Number 1: Linked List implementation (**60P**)

We want to implement an interactive programme that uses Linked list data structure in order to store data. This programme has an interactive menu which asks which linked list operation you want to execute. The menu should be like the list below:

A Add String //Add the string to the specific index
C Check if Empty //use isEmpty() function
E Search for String
L List Strings // use toString() function in the format { str1 str2 str3}
O List Current Size //use size() function
Q Quit
R Remove String //use the remove() function
T Reverse //reverse the whole list the
U Remove Duplicates
? Display Help

Note:
a) the name of the main class is myAssignment
b) your project should contain both myLinkedList which implements LinkedList interface
c) You should handle both uppercase and lowercase character in the command prompt. That is, if somebody types 'a' you should consider it identical to what is done by typing 'A'.
d) Non existence character like "b" should be handled: ask "`What action would you like to perform?`"again .
e) If the action you want to be done is actually occurred , ask "`What action would you like to perform?`" again until the termination of the programme.
d) Exception in the context is not mean java exception.

Add string: add the arbitrary string to the arbitrary index of the list which starts from 0 index.(8p)
Exception: You can not add value to the list which the previous node does not exist, it means if you want to add string with index 10, the linked list should have index 9. You should handle the exception and print out - "`The last index is ..`" and ask the question again - `Please enter an index to add`.

Check if Empty: use isEmpty() function and if the list is empty print out ("`The list is empty.\n` ") and if the list contains at least one element print out ("`The list contains some element(s).\n`").(6p)

Search for String: the program asks the index and should returns the string value of the element in index. You should take the index you want to search and the result should be something like: (`string at the given index is str1`)(6p)
Exception: if the index does not exist in the list return `the index you want is not in the list` and ask `Please enter an index to search:` again.

List Strings: Print out in this format `{str1 str2 str3 … }`(8p)
You should print one space before each string and for the last string one space after.
Exceptions:
When your list is empty and you want to print out your list just print { }. It has just one space.

List Current Size: print out the current size of your list, Example: `The current size is 2`

Quit: Terminate the programme.(2p)

Remove String: you should do everything needed for removing a node and print out "`str was removed`".(6p)
Exception: You should check the element in index exists or not. If the index does not exist notify by "`It does not exist`" and ask the "`Please enter the index of a string to remove" again.`

Reverse action: You need to reverse the list and print "`list reversed`". You do not need to print the content of the reversed list.
Example : the list is {123,789,456,321}, reversed List is {321,456,789,123}(8p)

Remove Duplicate: remove duplicate nodes, keeping the first occurrence of each node.(8p)
Exception: you should return - `there is no duplicate`, if no duplicates are not exist.

Help: show the menu again (2p)

Simple Expected Result:

```
[ds00@ds ~]$ java myAssignment
Choice          Action
------          ------
A         Add String
C         Check if Empty
E         Search for String
L         List Strings
O         List Current Size
Q         Quit
R         Remove String
T         Reverse
U         Remove Duplicate
?         Display Help

What action would you like to perform?
a
Please enter a string to add:
asd
Please enter an index to add:
3
The last index is 0
Please enter an index to add:
0
What action would you like to perform?
c
The list contains some element(s).
What action would you like to perform?
E
Please enter an index to search:
0
string at the given index is asd
What action would you like to perform?
L
{ asd }
What action would you like to perform?
O
The current size is 1
```

```
What action would you like to perform?
R
Please enter the index of a string to remove:
0
asd was removed
What action would you like to perform?
L
{ }
What action would you like to perform?
a
Please enter a string to add:
qwe
Please enter an index to add:
0
What action would you like to perform?
a
Please enter a string to add:
asd
Please enter an index to add:
1
What action would you like to perform?
a
Please enter a string to add:
asd
Please enter an index to add:
2
What action would you like to perform?
T
list reversed
What action would you like to perform?
U
duplicate removed
What action would you like to perform?
L
{ asd qwe }
What action would you like to perform?
q
```
[ds00@ds ~]$

Number 2: Arraylist vs Linked List (**40P**)

Implement the abstract data type mentioned in Problem 1 using an array; let this class be ArrayList.
We want to compare the ArrayList and Linkedlist performance in practice.
Write code to compare the performance of these operations in both ArrayList and LinkedList.
Add 100,000 random strings to the both lists and perform the following actions on them:
get()
remove()
add()

Note:

a) The main class has name Compare
b) Print out in the designated format in the example below
c) Unit of the displayed result is nanosecond.

Output example:
[ds00@ds ~]$ java Compare
\tArrayList\tLinkedList

Add\t|11609741\t|1570250

Get\t|366421333\t|1456222

Remove\t|362122817\t|11609741

/* \t is a tab character. */

Cooperation:

This assignment is to be done individually. Do not share code with others.

Assessment:

- Copying others work will result zero.
- If your program doesn't compile, then you will get a zero score.
- Make sure your program runs on **Eclipse**. Grading will be done in the Linux environment with Java 10. If you comment or use any Korean character, there will be an error during process of compiling.

Submission:

- By eTL. Make sure you write your name and student number as a comment on top of myAssignment.java.
- Compress your myAssignment.java, myLinkedList.java, LinkedList.java, myArrayList.java, ArrayList.java, Compare.java as a zip file.
- Once you have submitted assignment, pressing 'mark for grading' will prevent from making any change. Do NOT email for changing in the assignment after you have clicked 'mark for grading'.

Late submissions will not be accepted