CSC 115 A01 (Java)
Fall 2012 L.Jackson

Assignment #3

Due

Milestone (Submit Part 1): before 3:30 pm on Friday, October 19
Final Submission (Submit Part 2): before 3:30pm on Friday, October 26

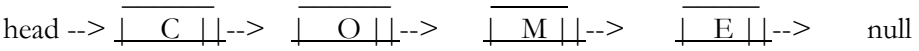
Learning Outcomes: Upon successful completion of Assignment #3 you will be able to:

- Use an interface to implement an ADT
- Implement a reference based Linked List data structure, including displaying the contents of a list, inserting a node into a specified position of the list, determining the length of the list
- Compare and contrast an array versus a linked list.

Part 1

Problem statement:

Consider the following reference based linked list that stores characters: 'C', 'O', 'M' and 'E'.



In Part 1 of this assignment you will given a write a basic linked list class (call it `MyList.java`) that will store the above data.

Specifications:

- Examine the `Node.java` class that is provided under the Assignment 2 link of the Resources part of the course connex site. Observe that each 'Node' has two attributes an 'Object' called `value` and a reference, called `next`, to another `Node`. Although it would be more convenient for this particular program if the Node's `value` were of type char rather than Object, this Object can store a char as required by the problem statement above.
- Examine the TesterA3P1.java program that is provided under the Assignment 2 link of the Resources part of the course connex site. It will test the linked list that you create an, if correct, will output the characters : 'C', 'O', 'M' and 'E'. Your task is to write the linked list.
- Your linked list class has only one attribute, the head, which is a reference to a Node.
- Your linked list class has the following methods:
    - A default constructor that creates an empty list (ie, the head is assigned `null`).

- A method 'addHead' that accepts a character parameter and inserts a new Node to the head of the list whose value is that character and whose `next` reference points to the former head of the list.
- A 'toString' methods that displays the characters of the list on the terminal. If the list is empty it outputs `"this is an empty list..."`

- Test your MyList.java class with the TesterA3P1.java program: Correct output should be COME.

Submitting your Solution:

When complete, submit your MyList.java class to the CSc 115 Connex Site Assignment 3 Milestone link before 3:30 on Friday, October 19, 2012.

This milestone is a formative exercise: It must be completed and will be examined by the instructor but will only be used to give you information about your learning.

Part 2

Problem statement:

Design a program to **implement** String operations, using linked list objects designed by you (as opposed to the array based implementations in the String class provided in Java: where each string is implemented as an array of characters). In our code, however, each string will be implemented as a reference based linked list where each character is stored in a node whose info field is of type Object.

Your assignment will consist of 4 classes:
1. `Node.java`: represents the Nodes in the linked list.
2. `ADT.java`: An interface file that specifies the set of methods that must be implemented.
3. `StringADT.java`: written by you and implements the linked list. It must begin with the line: `public class StringADT implements ADT {`
4. `TesterA3P2.java`: A client (tester) class that will act as a driver for your StringADT class. (Do not write this class; just use the provided one to test your code.)

Of the 4 classes, you will only need to hand in `StringADT.java`.

Specifications:

- In the class StringADT, be sure to *implement* the abstract data type (ADT.java) that is provided via the Assignment 2 link of the Resources part of the course connex site.
- Please make **sure** to use linked list operations to implement the copy, concat, substring, leng, and repeats methods. Although it is possible to achieve the desired functionality by calling toString and assign from all your methods, this is not the expected technique and will **not** yield any grades.
- The only methods that should be using calls to the Java String API are assign (where you need to get the data out of a String you've been given) and toString.
- write the methods of the StringADT class, implemented with reference based linked lists, as follows:
  `x.assign("nyu")` – is equivalent to `x = "nyu"`

`z.concat (x , y)` – is equivalent is to concatenation of Strings x and y into a String z
`w.repeats(z , 12)` – is equivalent to repeating String z 12 times in String w
`w.substring (x, 2,5)` – is equivalent to taking Substring of x from character 2 to 5 and storing it in w.
`u.copy (x)` – is equivalent to copying x and storing it in u.
`w.leng()` – returns the length of w as an integer
`toString()` – you should know what this does!

Submitting your Solution:

- Don't forget to adjust the comments so they contain your name and student number.
- When complete submit your **StringADT.java** file to the CSc 115 Connex Site using the Assignments: Assignment 3 Submission link before 3:30 on Friday, October 26, 2012.