Assignment #9: LinkedList**, CSE 205, 40 Points**

Due Date : Saturday, Nov 18th , 2017, at 11:59pm.

Important: This is an individual assignment. Please do not collaborate.

In the Assignment, you are given three files Assignment10.java, LinkedList.java, ListIterator.java. You will need to add additional methods in the LinkedList class in the LinkedList.java file. The LinkedList will be tested using **strings only.**

Specifically, the following methods must be implemented in the LinkedList class: (You should utilize listIterator() method already defined in the LinkedList class to obtain its LinkedListIterator object, and use the methods in the LinkedListIterator class to traverse from the first element to the last element of the linked list to define the following methods.)

public int size( )

The size method returns the number of strings that the linked list contains at the time when this method is called.

public int searchElement(Object element)

The *searchElement* method checks at which index the parameter element (string) is located, and returns its index. If the linked list does not contain the parameter element, then it returns -1.

public Object getElement(int index)

The *getElement* method returns an element at the given index passed as its parameter. If the parameter index is larger or smaller than the existing indices, it should throw an IndexOutOfBoundsException.

public void setElement (int index, Object element)

The *setElement* method set/replace the original element at parameter *index* with the parameter *element,* all other elements remain unchanged. If the parameter index is larger or smaller than the existing indices, it should throw an object of the IndexOutOfBoundsException class.

public Object insertElement(int index, Object element)

The *insertElement* method inserts the parameter object at the parameter index. If the index is out of bounds, throws an *IndexOutOfBoundException*. Note: the element can be inserted at the end of the linked list.

public Object removeElement(int index)

The *removeElement* method removes and returns element at parameter index and throw an *IndexOutOfBoundException* if the index is out of bound.

public int countHowMany(Object searchedObject)

The *countHowMany* method returns the number of occurences of the parameter object in the LinkedList. It returns 0 if the parameter object does not exist in the linked list.

public void removeDuplicate(Object removedObject)

The *removeDuplicate* method removes all occurences of the parameter object from the LinkedList.

public void appendAtEnd(Object element, int howManyTimes)

The *appendAtEnd* method appends the parameter object the parameter number of times at the end of the linked list. For example, a call of *list1*.*appendAtEnd*("Dog", 3) will append string "Dog" three times at the end of the linked list.

public void appendAfter(Object element1, Object element2)

The *appendAfter* method appends the second parameter object, *i.e. element2* right after the first occurence of first parameter object, *i.e. element1*. If *element1* is not inside the linked list, then *element2* will be appended at the front/head of the linked list.

public void reverseFirstFew(int howMany)

The *reverseFistFew* method reverses the first parameter number of elements inside the linked list. For example, if the original linked list is { A B C D E F G }, a call of *list.reverseFirstFew*(3) will change the linked list to { C B A D E F G }. Note: (1) You need to consider the boundary value, *i.e.*cases where *howMany <= 0 or*

howMany > *list.size*( ). (2): *list.reverseFirstFew(list.size*( )) should be able to reverse the whole linked list

Test your LinkedList class with the given Assignment10.java file.

It is recommended that you test boundary cases such as the cases when the linked list is empty, when it contains only one element, when adding/removing at the beginning or at the end of the linked list.