CENG 202 Data Structures

Teaching Assistants

- Res. Asst. Elif ŞANLIALP
- Res. Asst. Ahmet Esad TOP
- Res. Asst. Elif ERÇEK

Lab Assignment 2: Linked Lists

Aim: Implement some functions for Linked List data structure.

LinkList.java

You can download Test.java file. You have the linklist structure below and you will be implemented the following functions for LinkList:

- sortedInsert()
- insertToSpecificPosition()
- delete()

You will get 1 point for each correct function, but only correct solutions will be accepted.

```
class Node {

    public String data;
    public Node next;

    public Node(String nodeData) {
        this.data = nodeData;
        this.next = null;
    }
}
```

```
class LinkList {

    public Node head; // First item of LinkList

    public LinkList() {
        this.head = null;
    }

    void printLinkList() {
        Node current = head;
        if (current == null) {
            System.out.println("The list is empty!");
            return;
        }
        while (current != null) {
            System.out.print(current.data + " -> ");
            current = current.next;
        }
        System.out.println("");
    }

    void insertToHead(String nodeData) {
        Node newNode = new Node(nodeData);

        newNode.next = head;
        head = newNode;
    }

    // Complete the insertToSpecificPosition function below.
    // Indexes are starting from 0.
    // A->B->C->null
    // Want to add D to position 1   A -> D -> B -> C
    void insertToSpecificPosition(String nodeData, int position) {

    }

    // Delete if a node has same data with key value
    // Return deleted Node or null if it is not found.
    public Node delete(String key)
    {

    }

    // Complete the sortedInsert function below which inserts items alphabetically
sorted.
    void sortedInsert(String nodeData) {

    }
}
```

Output

Please check your output with the correct one below and try to make your solutions the same exactly matched. Do not forget to **show** your solutions after finish to your TA.

```
------------------------------
Sorted Insert Operations
------------------------------
The list is empty!
D ->
B -> D ->
B -> C -> D ->
A -> B -> C -> D ->
A -> B -> C -> D -> F ->
A -> B -> C -> D -> E -> F ->
------------------------------
Insert to a Position Operations
------------------------------
A -> B -> C -> D -> E -> F -> Z ->
A -> B -> C -> T -> D -> E -> F -> Z ->
X -> A -> B -> C -> T -> D -> E -> F -> Z ->
Can not add G to this position!
------------------------------
Delete Operations
------------------------------
X -> A -> B -> C -> T -> D -> E -> F ->
X -> A -> B -> C -> D -> E -> F ->
A -> B -> C -> D -> E -> F ->
L is not found
------------------------------
```