Course ENSE 374-Lab

Assignment #1

Linked List Code

By

Mohamed Aly (200393686)

# URL link to GitHub

<https://github.com/mma012/ENSE-374-lab/branches>

Java Code

1. Main

package ListElement.src;

import ListElement.src.ListElement;

import java.util.Scanner;

public class MainClass

{

public static void main(String[] args)

{

// Adding new listElements to the list

System.out.println("How many elements you want to add to the list");

Scanner userinput = new Scanner(System.in);

while(!userinput.hasNextInt())

{

System.out.println("'" + userinput.nextLine() + "'" + " is not a valid entry. Please enter a number: ");

}

int NumOfElements = userinput.nextInt();

ListElement [] Le = new ListElement[NumOfElements];

System.out.println("Enter the data of 1st element");

int FirstListData = userinput.nextInt();

// Defining a new LinkedList and putting a value in the head list

LinkedList list = new LinkedList(FirstListData);

for (int i=1; i< Le.length; i++)

{

System.out.println("Add Data of element # " + (i+1));

int Listdata = userinput.nextInt();

Le[i] = new ListElement();

Le[i].setData(Listdata);

list.addElement(Le[i]);

}

//Printing the data of the Linkedlist from head to tail

System.out.println("Elements in the list from Head to Tail");

list.printLinkedListHead();

//Printing the data of the Linkedlist from tail to head

System.out.println("Elements in the list from Tail to Head");

list.printLinkedListTail();

//Deleting an element in the list

System.out.println("Which list you want to delete? ");

int listtodelete = userinput.nextInt()-1;

list.deleteElement(listtodelete);

System.out.println("Now the Elements in the list from Head to Tail are");

list.printLinkedListHead();

System.out.println("Now the Elements in the list from Tail to Head are");

list.printLinkedListTail();

}

}

1. ListElement.java

**package** ListElement.src;

// Creating ListElement class under package ListElement.src

**public** **class** ListElement {

ListElement next ;

ListElement previous;

**private** **int** data;

**public** Object todelete;

// Constractor for new object initialization

**public** ListElement() {

**this**.data = 0 ;

**this**.next = **null**;

**this**.previous = **null**;

}

// Defining Setters and Getters for retrieving and input data

// from/to the list

**public** ListElement getNext() {

**return** next;

}

**public** **void** setNext(ListElement next) {

**this**.next = next;

}

**public** ListElement getPrevious() {

**return** previous;

}

**public** **void** setPrevious(ListElement previous) {

**this**.previous = previous;

}

**public** **int** getData() {

**return** data;

}

**public** **void** setData(**int** data) {

**this**.data = data;

}

}

1. LinkedList.java

**package** ListElement.src;

**import** ListElement.src.ListElement;

// Creating LinkedList class

**public** **class** LinkedList {

ListElement head =**new** ListElement();

// Constractor for new object "head" initialization

**public** LinkedList(**int** n){

head.setData(n);

head.setNext(**null**);

head.setPrevious(**null**);

}

// Function to add new element in the linkedlist

**public** **void** addElement(ListElement le)

{

// Start by going to the first element in the linkedlist

ListElement last =head;

// As long as the last has a next go to the next one

**while**(hasNext(last))

{

last=last.getNext();

}

//Connect the last element with the new element

last.setNext(le);

le.setPrevious(last);

}

//New function to retrieve the element corresponding to the index

**public** ListElement getElement(**int** index)

{

**int** count =0;

ListElement currentElement = head;

**while**(index !=count)

{

count++;

currentElement =currentElement.getNext();

}

**return** currentElement;

}

//New function to delete the element corresponding to the index

**public** ListElement deleteElement(**int** index)

{

ListElement todelete = getElement(index);

ListElement previous = todelete.getPrevious();

ListElement next = todelete.getNext();

**if** (index == 0 ) {

System.***out***.println("You have deleted the head");

head = head.getNext();

head.setPrevious(**null**);

**return** head;

}

**else** **if**(todelete.getNext() == **null**) {

System.***out***.println("You have deleted the tail");

previous.next = **null**;}

**else** {

System.***out***.println("You have deleted list #" +(index+1));

next.setPrevious(previous);

previous.setNext(next);}

**return** next;

}

// Function to print the list from head to tail

**public** **void** printLinkedListHead()

{

ListElement currentelement = head;

System.***out***.print(head.getData() +" ");

**while**(hasNext(currentelement))

{

currentelement = currentelement.getNext();

System.***out***.print(currentelement.getData() +" ");

}

System.***out***.println();

}

//Function to print the list from tail to head

**public** **void** printLinkedListTail()

{

ListElement tail = head;

// System.out.print(head.getData() +" ");

**while**(hasNext(tail))

{

tail = tail.getNext();

}

System.***out***.print(tail.getData() + " ");

**while** (hasPrevious(tail))

{

tail = tail.getPrevious();

System.***out***.print(tail.getData() +" ");

}

System.***out***.println();

}