**Java Program to Find the Middle Node of a Linked list in a Single-pass**

public class LinkedListTest {

public static void main(String args[]) {

//creating LinkedList with 5 elements including head

LinkedList linkedList = new LinkedList();

LinkedList.Node head = linkedList.head();

linkedList.add( new LinkedList.Node("1"));

linkedList.add( new LinkedList.Node("2"));

linkedList.add( new LinkedList.Node("3"));

linkedList.add( new LinkedList.Node("4"));

//finding middle element of LinkedList in single pass

LinkedList.Node current = head;

int length = 0;

LinkedList.Node middle = head;

while(current.next() != null){

length++;

if(length%2 ==0){

middle = middle.next();

}

current = current.next();

}

if(length%2 == 1){

middle = middle.next();

}

System.out.println("length of LinkedList: " + length);

System.out.println("middle element of LinkedList : " + middle);

}

}

class LinkedList{

private Node head;

private Node tail;

public LinkedList(){

this.head = new Node("head");

tail = head;

}

public Node head(){

return head;

}

public void add(Node node){

tail.next = node;

tail = node;

}

public static class Node{

private Node next;

private String data;

public Node(String data){

this.data = data;

}

public String data() {

return data;

}

public void setData(String data) {

this.data = data;

}

public Node next() {

return next;

}

public void setNext(Node next) {

this.next = next;

}

public String toString(){

return this.data;

}

}

}

OUTPUT:

