Date: Sunday, July 14, 2019

Submission Date:

Title: Linked List Implementation in java

Aims:

- Getting practice with Linked List
- Getting practice with insert a data in front, end, between and delete.

Tasks:

1. Execute the following code,
   a. Node class (To create Node template)

```
public class Node {
    int data;
    Node next;

      Node(int d)
      {
         data=d;
         next=null;
      }
}
```

   b. LinkedList Class

```
public class linkedlList {
   Node head;

   public void display()
   {
      Node n=head;
      while(n!=null)
      {
         System.out.println(n.data);
         n=n.next;
      }
   }

}
```

   c. Main class

```
public class main_method {
   public static void main(String[] args) {
```

```
        linkedlList link=new linkedlList();
        link.head=new Node(4);
        Node second=new Node(5);
        Node third=new Node(6);

        link.head.next=second;
        second.next=third;

        link.display();

    }
}
```

2. Insert new head element

```
public void insertfront(int newdata)
  {
    Node newnode=new Node(newdata);
    newnode.next=head;
    head=newnode;
  }
```

3. Insert at any point

```
public void insertatAny(Node pre_node, int newdata)
  {
    if(pre_node==null)
    {
      System.out.println("It can not be null");
    }
    else
    {
      Node newnode=new Node(newdata);
      newnode.next=pre_node.next;
      pre_node.next=newnode;
    }
  }
```

4. Insert node at ending point

```
public void insertLast(int newdata)
  {
    Node newnode=new Node(newdata);
    if(head==null)
    {
```

```
            head=new Node(newdata);
        }
        else
        {
            newnode.next=null;
            Node end=head;
            while(end.next!=null)
            {
                end=end.next;
            }
            end.next=newnode;
        }
    }
```

5. Delete Node at any point

```
public void deleteNode(int dele_node)
    {
        Node temp=head, prev=null;

        if(temp!=null && temp.data==dele_node)
        {
            head=temp.next;

        }
        while(temp!=null && temp.data!=dele_node)
        {
            prev=temp;
            temp=temp.next;
        }
        if(temp==null)
        return;
        prev.next=temp.next;

    }
```

Exercise:

1. Create Linked list with four elements using three classes.

2. Display all four Linked list element by using display() method.

3. Insert two more elements in between 1st and 2nd element, 3rd and 4th elements

4. Insert new node as a head of the linked list

5. Insert new tail or end node

6. Delete the 3rd linked list element and insert new element of that position.

7. Also write java program to reverse the Linked list element by using reverse() method.

8. Display the reversed output.

Discussion:

- Linked List Advantages and Disadvantages
- Linked List Vs Arrays