Personal Information

Name: Khan Muhammad

CMS ID: 023-22-0199

**Assignment 4**

Develop a linked list (Doubly) with the following methods

1.       Add an element after a given node in a linked list.

2.       Add an element before a given node in a linked list.

3.       Remove an element after a given node in a linked list.

4.       Remove an element before a given node in a linked list.

5.       Update an element after a given node in a linked list.

6.       Update an element before a given node in a linked list.

**Source Code:**

class Node

{

    int data;

    Node next;

    Node prev;

    Node(int data)

    {

        this.data = data;

        this.next = null;

        this.prev = null;

    }

}

class LinkedList

{

    Node head;

    Node tail;

    LinkedList()

    {

        this.head = null;

        this.tail = null;

    }

    LinkedList(int data)

    {

        Node newNode = new Node(data);

        head = newNode;

        tail = newNode;

    }

    void addAfter(Node afterNode, int data)

    {

        Node newNode = new Node(data);

        newNode.next = afterNode.next;

        afterNode.next = newNode;

    }

    void addBefore(Node beforeNode, int data)

    {

        Node newNode = new Node(data);

        newNode.next = beforeNode;

        newNode.prev = beforeNode.prev;

        if(beforeNode.prev != null)

        {

            beforeNode.prev.next = newNode;

        }

        beforeNode.prev = newNode;

    }

    void removeAfter(Node afterNode)

    {

        if(afterNode.next == null)

        {

            System.out.print("No Node Found After Given Node!");

            return;

        }

        if(afterNode.next.next == null)

        {

            afterNode.next = null;

            tail = afterNode;

            afterNode.next.prev = null;

            return;

        }

        afterNode.next.next.prev = afterNode;

        afterNode.next = afterNode.next.next;

    }

    void removeBefore(Node beforeNode)

    {

        if(beforeNode.prev == null)

        {

            System.out.print("No Node Found Before Given Node!");

            return;

        }

        if(beforeNode.prev.prev == null)

        {

            beforeNode.prev.next = null;

            beforeNode.prev = null;

            head = beforeNode;

            return;

        }

        beforeNode.prev.prev.next = beforeNode;

        beforeNode.prev = beforeNode.prev.prev;

    }

    void updateElementAfter(Node afterNode, int data)

    {

        if(afterNode.next==null)

        {

            System.out.print("No Node Found After Given Node!");

            return;

        }

        afterNode.next.data = data;

    }

    void updateElementBefore(Node beforeNode, int data)

    {

        if(beforeNode.prev==null)

        {

            System.out.print("No Node Found Before Given Node!");

            return;

        }

        beforeNode.prev.data = data;

    }

    void printList()                // Print List

    {

        Node temp = head;

        System.out.print("{");

        while(temp!=null)

        {

            System.out.print(temp.data+", ");

            temp = temp.next;

        }

        System.out.println("\b\b}");

    }

}

class Assignment4

{

    public static void main(String [] args)

    {

        LinkedList l1 = new LinkedList(10);

        Node second = new Node(20);

        Node third = new Node(30);

        Node forth = new Node(40);

        l1.head.next = second;

        second.next = third;

        second.prev = l1.head;

        third.next = forth;

        third.prev = second;

        forth.prev = third;

        System.out.print("\nList: ");

        l1.printList();

        System.out.print("\n1. Adding 35 After Third Node: ");

        l1.addAfter(third, 35);

        l1.printList();

        System.out.print("\n2. Adding 15 Before Second Node: ");

        l1.addBefore(second, 15);

        l1.printList();

        System.out.print("\n3. Removing After Forth Node: ");

        l1.removeAfter(third);      // 'third' because the Forth one in the list now, is third

        l1.printList();

        System.out.print("\n4. Removing Before Third Node: ");

        l1.removeBefore(second);      // 'second' because the Third one in the list now, is second

        l1.printList();

        System.out.print("\n5. Updating Element To '25' After Second Node: ");

        l1.updateElementAfter(second, 25);

        l1.printList();

        System.out.print("\n6. Updating Element To '15' Before Second Node: ");

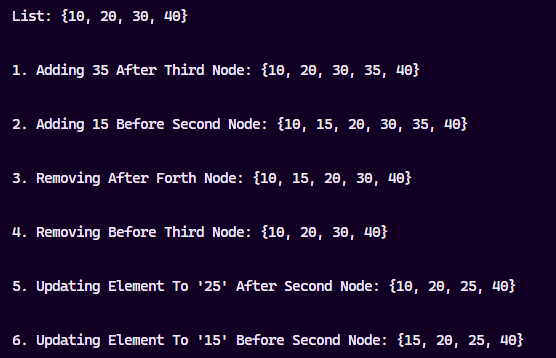
        l1.updateElementBefore(second, 15);

        l1.printList();

    }

}

**Result:**

****