**LinkedList**

1. **Singly LinkedList:**

**public** **class** Main {

**public** **static** **void** main(String[] args) {

LinkedList list = **new** LinkedList();

list.addNode(20);

list.addToStart(10);

list.addToEnd(30);

list.addToEnd(40);

list.search(30);

list.display();

}

}

**public** **class** Node {

Object Data;

Node next;

**public** Node(Object data) {

Data = data;

}

}

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

Node head;

**int** size;

// 1- add Node...

**public** **void** addNode(Object data) {

Node newNode = **new** Node(data);

head = newNode;

size++;

}

// // 2- addFirst

// public void addFirst(Object data) {

// Node newNode = new Node(data);

// if(isEmpty()) {

// addNode(data);

// return;

// }else {

// newNode.next = head;

// head = newNode;

// size++;

// }

// }

// Add To First

**public** **void** addToStart(Object data) {

**if**(isEmpty()) {

System.***out***.println("List is Empty");

**return**;

}

Node curr = head;

**while**(curr != **null**) {

Node newNode = **new** Node(data);

newNode.next = head;

head = newNode;

size++;

**break**;

}

}

// 3- addLast......

// public void addLast(Object data) {

// Node newNode = new Node(data);

// if(isEmpty()) {

// addNode(data);

// return;

// }

// Node currentNode = head;

// while (currentNode.next != null) {

// currentNode = currentNode.next;

// }

// currentNode.next = newNode;

// newNode = currentNode;

// size++;

// }

// Add To End

**public** **void** addToEnd(Object data) {

**if**(isEmpty()) {

System.***out***.println("List is Empty");

**return**;

}

Node curr = head;

**while**(curr != **null**) {

**if**(curr.next == **null**) {

Node newNode = **new** Node(data);

curr.next = newNode;

size++;

**break**;

}

curr = curr.next;

}

}

//4- Add After specific node

**public** **void** addAfter(Object valueAfter, Object data) {

**if**(isEmpty()) {

System.***out***.println("List is Empty");

**return**;

}

Node curr = head;

**while**(curr != **null**) {

**if**(curr.Data == valueAfter) {

Node newNode = **new** Node(data);

newNode.next=curr.next;

curr.next=newNode;

**break**;

}

curr = curr.next;

}

}

//5- Add between To Nodes

**public** **void** addBetween(Object first, Object second, Object data) {

**if**(isEmpty()) {

System.***out***.println("List is Empty");

**return**;

}

Node curr = head;

**while**(curr != **null**) {

**if**(curr.Data == first && curr.next.Data == second) {

Node newNode = **new** Node(data);

newNode.next = curr.next;

curr.next = newNode;

**break**;

}

curr = curr.next;

}

}

// 5- deleteFirst....

**public** **void** deleteFirstNode() {

**if**(isEmpty()) {

System.***out***.println("List is Empty");

**return**;

}

head = head.next;

size--;

}

// 6- deleteLast

**public** **void** deleteLastNode() {

**if**(isEmpty()) {

System.***out***.println("List is Empty");

**return**;

}

Node curr = head;

Node nextNode = curr.next;

**while**(curr.next != **null**) {

**if**(nextNode.next == **null**) {

curr.next = **null**;

size--;

}

curr = nextNode;

nextNode = nextNode.next;

}

deleteFirstNode();

}

// search

**public** **void** search(Object value) {

Node curr = head;

**int** index = 0;

**boolean** flag = **false**;

**if**(isEmpty()) {

System.***out***.println("list is Empty");

}

**else** {

**while**(curr != **null**) {

**if**(curr.Data == value) {

flag = **true**;

**break**;

}

index++;

curr = curr.next;

}

}

**if**(flag) {

System.***out***.println("The Element "+curr.Data +" is present in index "+ index);

}

**else** {

System.***out***.println("The Element is not present in the list");

}

}

// 6- deleteLast....

// public void deletLast() {

// Node current = head;

// if(isEmpty()) {

// System.out.println("List is Empty");

// return;

// }

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

// current = current.next;

// }

// current.next = current;

// current.next = null;

// size--;

// }

// delete a specific node

**public** **void** deleteSpecificNode(Object value) {

Node current = head, prev = **null**;

**if**(current != **null** && current.Data == value) {

deleteFirstNode();

**return**;

}

**while**(current != **null** && current.Data != value) {

prev = current;

current = current.next;

}

**if**(current == **null**) {

System.***out***.println("The value is not Exist!");

}**else** {

prev.next = current.next;

size--;

}

}

// 7- boolean... isEmpty

**boolean** isEmpty() {

**return** head == **null**;

}

// 8- display list

**public** **void** display() {

**if**(isEmpty()) {

System.***out***.println("List is Empty");

}**else** {

Node currentNode = head;

**while**(currentNode != **null**) {

System.***out***.print(currentNode.Data + " -> ");

currentNode = currentNode.next;

}

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

System.***out***.println("Size : "+size);

}

}

}

1. **Doubly LinkedList:**

**public** **class** Main {

**public** **static** **void** main(String[] args) {

DoublyLinkedList list = **new** DoublyLinkedList();

list.addNode(30);

list.addToStart(20);

list.addToEnd(40);

list.addToEnd(50);

// list.addAfter(40, 44);

// list.addBetween(30, 40, 32);

// list.deleteNode(40);

list.search(40);

list.display();

}

}

**public** **class** Node {

Object Data;

Node next;

Node prev;

**public** Node(Object data) {

Data = data;

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

**this**.prev = **null**;

}

}

**public** **class** DoublyLinkedList {

Node head;

Node tail;

**int** size;

// 1- add Node...

**public** **void** addNode(Object data) {

Node newNode = **new** Node(data);

tail = head = newNode;

size++;

}

// 2- addFirst

**public** **void** addToStart(Object data) {

Node newNode = **new** Node(data);

**if**(isEmpty()) {

addNode(data);

**return**;

}

newNode.next = head;

head.prev = newNode;

head = newNode;

size++;

}

// 3- addLast

**public** **void** addToEnd(Object data) {

Node newNode = **new** Node(data);

**if**(isEmpty()) {

addNode(data);

**return**;

}

tail.next = newNode;

newNode.prev = tail;

tail = newNode;

size++;

}

// 4- add between tow nodes

**public** **void** addBetween(Object first, Object second, Object data) {

Node newNode = **new** Node(data);

**if**(isEmpty()) {

System.***out***.println("The List is Empty");

}

Node curr = head;

**while**(curr != **null**) {

**if**(curr.Data == first && curr.next.Data == second) {

Node temp;

temp = curr.next;

curr.next = newNode;

newNode.prev = curr;

newNode.next = temp;

temp.prev = newNode;

size++;

**return**;

}

curr = curr.next;

}

System.***out***.println("Elements Not Found");

}

// 5- add after specific node

**public** **void** addAfter(Object value , Object data) {

Node newNode = **new** Node(data);

**if**(isEmpty()) {

System.***out***.println("The List is Empty");

}

Node curr = head;

**while**(curr != **null**) {

**if**(curr.Data == value) {

Node temp;

temp = curr.next;

curr.next = newNode;

newNode.prev = curr;

newNode.next = temp;

temp.prev = newNode;

size++;

**return**;

}

curr = curr.next;

}

System.***out***.println("Elements Not Found");

}

// 6- Delete First node

**public** **void** deleteFirst() {

**if**(isEmpty()) {

System.***out***.println("The List is Empty");

}

head = head.next;

head.prev = **null**;

size--;

}

// 7- Delete Last node

**public** **void** deleteLast() {

**if**(isEmpty()) {

System.***out***.println("The List is Empty");

}

tail = tail.prev;

tail.next = **null**;

size--;

}

// 8- Delete specific node

**public** **void** deleteNode(Object value) {

**if**(isEmpty()) {

System.***out***.println("The List is Empty");

}

Node curr = head;

**while**(curr != **null**) {

**if**(curr.Data == value) {

Node tempn, tempp;

tempn = curr.next;

tempp = curr.prev;

tempp.next = tempn;

tempn.prev = tempp;

size--;

**return**;

}

curr = curr.next;

}

}

// 9- search

**public** **void** search(Object value) {

Node curr = head;

**int** index = 0;

**boolean** flag = **false**;

**if**(isEmpty()) {

System.***out***.println("list is Empty");

}

**else** {

**while**(curr != **null**) {

**if**(curr.Data == value) {

flag = **true**;

**break**;

}

index++;

curr = curr.next;

}

}

**if**(flag) {

System.***out***.println("The Element "+curr.Data +" is present in index "+ index);

}

**else** {

System.***out***.println("The Element is not present in the list");

}

}

// boolean... isEmpty

**boolean** isEmpty() {

**return** head == **null**;

}

// display list

**public** **void** display() {

**if**(isEmpty()) {

System.***out***.println("List is Empty");

}**else** {

Node currentNode = head;

**while**(currentNode != **null**) {

System.***out***.print(currentNode.Data + " -> ");

currentNode = currentNode.next;

}

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

System.***out***.println("Size : "+size);

}

}

}