**Write a Java program to find the maximum and minimum value node from a circular linked list**

public class MinMax {

//Represents the node of list.

public class Node{

int data;

Node next;

public Node(int data) {

this.data = data;

}

}

//Declaring head and tail pointer as null.

public Node head = null;

public Node tail = null;

//This function will add the new node at the end of the list.

public void add(int data){

//Create new node

Node newNode = new Node(data);

//Checks if the list is empty.

if(head == null) {

//If list is empty, both head and tail would point to new node.

head = newNode;

tail = newNode;

newNode.next = head;

}

else {

//tail will point to new node.

tail.next = newNode;

//New node will become new tail.

tail = newNode;

//Since, it is circular linked list tail will points to head.

tail.next = head;

}

}

//Finds out the minimum value node in the list

public void minNode() {

Node current = head;

//Initializing min to initial node data

int min = head.data;

if(head == null) {

System.out.println("List is empty");

}

else {

do{

//If current node's data is smaller than min

//Then replace value of min with current node's data

if(min > current.data) {

min = current.data;

}

current= current.next;

}while(current != head);

System.out.println("Minimum value node in the list: "+ min);

}

}

//Finds out the maximum value node in the list

public void maxNode() {

Node current = head;

//Initializing max to initial node data

int max = head.data;

if(head == null) {

System.out.println("List is empty");

}

else {

do{

//If current node's data is greater than max

//Then replace value of max with current node's data

if(max < current.data) {

max = current.data;

}

current= current.next;

}while(current != head);

System.out.println("Maximum value node in the list: "+ max);

}

}

public static void main(String[] args) {

MinMax cl = new MinMax();

//Adds data to the list

cl.add(5);

cl.add(20);

cl.add(10);

cl.add(1);

//Prints the minimum value node in the list

cl.minNode();

//Prints the maximum value node in the list

cl.maxNode();

}

}

