**LAB 07**

**NAME: NOOR FATIMA**

**ROLL NO: 21sw062**

**SECTION: II**

**QUESTION#1**

(Concrete class)The Music app name is FireAir(class) music app which can perform followingfunctions•print all the liked list of songs/music names one by one.•Add new songs/music (insertion at start, end, middle)•Deletes a song/music using the number (Deletion at start, end, middle)•Searches song/music.

import java.util.Scanner;  
class Lab\_7<L> { //T,S  
 //private nested (static) class (and it is also generic class)  
 private static class Node<T> {  
 //properties  
 T data;  
 Node<T> nextNode;  
 // Node<T> preNode;  
  
 //constructor  
 Node(T data) {  
 this.data = data;  
 nextNode = null;  
 // preNode = null;  
 }  
 }  
  
 int size; //int  
 Node<L> head;  
  
 //constructor  
 Lab\_7() {  
 size = 0;  
 head = null;  
 }  
  
  
 int choice;  
 Scanner input = new Scanner(System.*in*);  
  
 int menu() {  
 System.*out*.println();  
 System.*out*.println("1. print linked list");  
 System.*out*.println("2.Add new music at first");  
 System.*out*.println("3.Add new music at middle");  
 System.*out*.println("4.Add new music at last");  
 System.*out*.println("5. Delete first song ");  
 System.*out*.println("6. Delete middle song ");  
 System.*out*.println("7. Delete last song ");  
 System.*out*.println("8. Search a song");  
 System.*out*.println("9 Exit");  
 System.*out*.println("enter your choice: ");  
 choice = input.nextInt();  
 return choice;  
 }  
  
 // By choice 1  
 public void Display() {  
 //head==null  
 if (size == 0)  
 System.*out*.println("Linked list is empty, nothing to display");  
 else {  
 Node<L> tempNode = head;  
 while (tempNode != null) {  
 System.*out*.println(tempNode.data);  
 tempNode = tempNode.nextNode;  
 }  
 }  
 }  
  
 //1. insert at first  
 public void insertAtFirst() {  
 System.*out*.println("enter the song name");  
 L song = (L) input.next();  
 //create new node  
 Node<L> newNode = new Node<L>(song); //nextNode=null  
 if (size == 0)  
 head = newNode;  
 else {  
 newNode.nextNode = head;  
 head = newNode;  
 }  
 size++;  
 }  
  
  
 public Node insertionAtLast() {  
 System.*out*.println("enter the song name");  
 L song = (L) input.next();  
 Node<L> newNode = new Node<L>(song);  
 if (head==null){  
 head = newNode;  
 newNode.nextNode=null;}  
 else {  
 Node p1=head;  
 for (p1 = head; p1.nextNode != null; p1 = p1.nextNode) {}  
 p1.nextNode=newNode;  
 }  
  
 size++;  
return head;}  
  
  
 public void insertionAtMiddle() {  
 System.*out*.println("enter the song name");  
 L song = (L) input.next();  
 Node<L> newNode = new Node<L>(song);  
 Node p = head;  
 for (int i = 0; i < size/2; i++) {  
 p = p.nextNode;  
 }  
 newNode.nextNode = p.nextNode;  
 p.nextNode=newNode;  
 size++;  
 }  
  
  
 public void deleteAtFirst() {  
 head = head.nextNode;  
 size--;  
 }  
  
  
 public void deleteAtMid() {  
 Node p = head;  
 for (int i = 0; i < size / 2; i++) {  
 p = p.nextNode;  
 }  
 p.nextNode = p.nextNode.nextNode;  
 size--;  
 }  
  
 public void deleteAtLast() {  
 Node p ;  
 for (p = head; p.nextNode.nextNode != null; p = p.nextNode) {}  
 p.nextNode=null;  
 size--;  
 }  
 public void search() {  
 System.*out*.println("enter the song name");  
 L song = (L) input.next();  
 Node p ;  
 Boolean flag=false;  
 for (p = head; p!= null; p = p.nextNode) {  
 if (p.data.equals(song)) flag=true;  
 }  
 if (flag==true) System.*out*.println("FOUND");  
 else System.*out*.println("NOT FOUND");  
  
 }}

public class Main {  
 public static void main(String[] args) {  
  
 // TASK 1  
 Lab\_7 lab=new Lab\_7();  
 while (lab.choice<7 ) {  
 switch (lab.menu()) {  
 case 1: {  
 lab.Display();  
 break;  
 }  
 case 2: {  
 lab.insertAtFirst();  
 break;  
 }  
 case 3: {  
 lab.insertionAtMiddle();  
 }  
 case 4: {  
 lab.insertionAtLast();  
 }  
 case 5:{  
 lab.deleteAtFirst();  
 }  
 case 6:{  
 lab.deleteAtMid();  
 }  
 case 7:{  
 lab.deleteAtLast();  
 }  
 case 8:{  
 lab.search();  
 }  
 case 9:{  
 System.*exit*(0);  
 } default:  
 System.*out*.print("Invalid output");  
 }//switch end  
 }//while loop

}}

**OUTPUT:**

"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.1\lib\idea\_rt.jar=54093:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.1\bin" -Dfile.encoding=UTF-8 -classpath C:\Users\hp\IdeaProjects\DSA\_ALL\_LABS\out\production\DSA\_ALL\_LABS Main  
  
 1. print linked list  
 2.Add new music at first  
 3.Add new music at middle  
 4.Add new music at last  
 5. Delete first song  
 6. Delete middle song  
 7. Delete last song  
 8. Search a song  
 9 Exit  
 enter your choice:  
 2  
 enter the song name  
 song1  
  
 1. print linked list  
 2.Add new music at first  
 3.Add new music at middle  
 4.Add new music at last  
 5. Delete first song  
 6. Delete middle song  
 7. Delete last song  
 8. Search a song  
 9 Exit  
 enter your choice:  
 2  
 enter the song name  
 son2  
  
 1. print linked list  
 2.Add new music at first  
 3.Add new music at middle  
 4.Add new music at last  
 5. Delete first song  
 6. Delete middle song  
 7. Delete last song  
 8. Search a song  
 9 Exit  
 enter your choice:  
 1  
 son2  
 song1  
  
 1. print linked list  
 2.Add new music at first  
 3.Add new music at middle  
 4.Add new music at last  
 5. Delete first song  
 6. Delete middle song  
 7. Delete last song  
 8. Search a song  
 9 Exit  
 enter your choice:  
 8  
 enter the song name  
 song1  
 FOUND  
  
 Process finished with exit code 0

**QUESTION#2**

(Concrete class)Making The Trip to Mehran UETDaily you would very often make the long trip out to Mehran UET. This route required a very specific sequence of buses, trains, and subways to get to the destination which you would follow along with using Google Maps.

// TASK 2  
class linkedList<L> { //T,S  
 //private nested (static) class (and it is also generic class)  
 private static class Node<T>{  
 //properties  
 T data;  
 Node<T> nextNode;  
 Node<T> preNode;  
 //constructor  
 Node(T data){  
 this.data=data;  
 nextNode=null;  
 preNode=null;  
 }  
 }  
  
 int size; //int  
 Node<L> head;  
 //constructor  
 linkedList(){  
 size=0;  
 head=null;  
 }  
 //operations  
 //1. insert at first  
 public void insertSource(L data){  
 //create new node  
 Node<L> newNode=new Node<L>(data); //nextNode=null  
 if(size==0)  
 head=newNode;  
 else{  
 head.preNode=newNode;  
 newNode.nextNode=head;  
 head=newNode;  
 }  
  
 size++;  
 }  
  
  
 public Node insertDesignation(L name) {  
 Node<L> newNode = new Node<L>(name);  
 if (head==null){  
 head = newNode;  
 newNode.nextNode=null;}  
 else {  
 Node p1;  
 for (p1 = head; p1.nextNode != null; p1 = p1.nextNode) {}  
 p1.nextNode=newNode;  
 }  
  
 size++;  
 return head;  
 }  
  
  
 public void MiddleCities(L name) {  
 insertDesignation(name);  
  
 }  
  
 public void Display(){  
 //head==null  
 if(size==0)  
 System.*out*.println("Linked list is empty, nothing to display");  
 else{  
 Node<L> tempNode=head;  
 while(tempNode!=null){  
 System.*out*.print(tempNode.data+" ");  
 tempNode=tempNode.nextNode;  
 }  
 }  
  
 }  
  
 } public class Main {  
 public static void main(String[] args) {  
 // TASK 2  
 linkedList<String>list=new linkedList<>();  
 list.insertSource("karachi");  
 list.MiddleCities("nooriabad");  
 list.MiddleCities("hyderAbad");  
 list.insertDesignation("tandoAdam");  
  
 list.Display();  
 }//main method  
  
 }

**OUTPUT:**

"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.1\lib\idea\_rt.jar=54127:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.1\bin" -Dfile.encoding=UTF-8 -classpath C:\Users\hp\IdeaProjects\DSA\_ALL\_LABS\out\production\DSA\_ALL\_LABS Main  
karachi nooriabad hyderAbad tandoAdam  
Process finished with exit code 0

**QUESTION#03(generic class)**

Create a generic class and implements the following operations:Task # 1:Implement the isEmpty() method in the Linked ListTask # 2:Implement the getSize() method in the Linked ListTask # 3:Implement the insertAtLast() method in the Linked ListTask # 4:Implement the insertAtLast() method in the Linked ListTask # 5:Implement the insertAtPosition() method in the Linked ListTask # 6:Implement the deleteFirst() method in the Linked ListTask # 7:Implement the deleteLast() method in the Linked ListTask # 8:Implement the deleteAtPosition() method in the Linked ListTask # 9:Implement the search() method in the Linked List to check whether the element exists in the list or not.Task # 10:delete element by value; implement deleteValue(int value)Task # 11:display method

class Node1<T> {  
 T data;  
 Node head;  
 Node next;  
 int size = 1;  
  
 public Node1(T data) {  
 this.data = data;  
 next = null;  
 }  
  
 public void insertAtFirst(T data) {  
 //create new node  
 Node<T> newNode = new Node<T>(data); //nextNode=null  
 if (size == 0)  
 head = newNode;  
 else {  
 newNode.next = head;  
 head = newNode;  
 }  
 size++;  
 }  
  
  
 public void deletionAtFirst(){  
 if (head==null){  
 System.*out*.println("Linked list is not exist");  
 }else {  
 head.next.prev=null;  
 next=head.next.next;  
 head=head.next;  
  
 }  
 size--;  
 }  
  
 public Node deletionAtLast(){  
 Node p;  
 for (p=head;p.next.next!=null;p=p.next){}  
 p.next=null;  
 size--;  
 return head;  
  
 }  
  
 public void display(){  
 for (Node p=head;p!=null;p=p.next){  
 System.*out*.print(p.data+" ");  
 }  
 }  
  
 public void getSize(){  
 System.*out*.println("SIZE: "+ size);  
 }  
  
 public boolean search(T name){  
 for (Node p=head ;p!=null;p=p.next){  
 if (p.data==name) return true;  
 }  
 return false;  
 }  
  
 public void NextElement(T name) {  
 boolean check= search(name);  
 Node p;  
 if (check==true){  
 for ( p=head ;p.next.data!=name;p=p.next){}  
 System.*out*.println(p.data);  
 }if (check==false) System.*out*.println("this image does not exist ");  
  
 }  
  
  
 public void previous(T name) {  
 boolean check= search(name);  
 if (check==true){  
 Node p;  
 for ( p=head ;p.next.data!=name;p=p.next){}  
 System.*out*.println(p);  
 }else System.*out*.println("this image does not exist ");  
  
 }  
  
 public void rename(T name, T newName){  
 boolean check= search(name);  
 if (check==true){  
 Node p;  
 for ( p=head ;p.data!=name;p=p.next){}  
 p.data=newName;  
 }else System.*out*.println("this image does not exist ");  
  
  
  
 }  
  
  
}

public class Main {  
 public static void main(String[] args) {  
NodeTASK3<Integer> i=new NodeTASK3<>(1);  
 i.insertAtFirst(1);  
 i.insertAtFirst(2);  
 i.insertAtFirst(3);  
 i.insertAtFirst(4);  
 i.display();  
 System.*out*.println(i.search(2));  
  
 i.update(2,22);  
 i.display();  
 System.*out*.print("previous: ");  
 i.previous(3);  
 i.display();  
 }//main method  
  
}

**OUTPUT:**

"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.1\lib\idea\_rt.jar=54177:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.1\bin" -Dfile.encoding=UTF-8 -classpath C:\Users\hp\IdeaProjects\DSA\_ALL\_LABS\out\production\DSA\_ALL\_LABS Main  
 4 3 2 1 true  
 4 3 22 1 previous: 4  
 4 3 22 1  
 Process finished with exit code 0