Data Structures and Algorithms

CSE2001

Assignment - 2 -Missing Question

Yashwanth Reddy
19BCE7362
Date- 17thJuly2021

Problem : A train is travelling from new Delhi to Trivandrum, initially it started with 20 compartments, after some hours of journey (at Bhopal) delinked 9 and 12 compartments at Nagpur link some more compartments at the end of the train. Each compartment will have 60 members, store 60 member's details for each compartments. Write the algorithm to execute it, write the corresponding JAVA code.

Code

```
import java.util.Scanner;
class Member{
  int member_id;
  String member_name;
  String start_location;
  String destination_location;
  Member(int member_id,String member_name,String start_location,String
destination_location){
      this.member_id=member_id;
      this.member_name=member_name;
      this.start_location=start_location;
      this.destination_location=destination_location;
```

```
  }
  public void display_member(){
     System.out.println("member id :"+this.member_id);
     System.out.println("member name :"+this.member_name);
     System.out.println("member start location :"+this.start_location);
     System.out.println("member destination location :"+this.destination_location);
  }
}
class Compartment{
  int c_no;
  Member m[]=new Member[60];
  Compartment next;
  Compartment(int c_no,Member[] m){
     this.c_no=c_no;
     this.m=m;
     next=null;
  }
}
class Train{
  Compartment head;
  Train(){
     this.head=null;
  }
  public void append_compartment(int c_no,Member[] m){
     Compartment new_c=new Compartment(c_no,m);
     if(this.head==null){
        this.head=new_c;
     }
     else{
        Compartment d_head=head;
        while(this.head.next!=null){
```

```
            this.head=d_head.next;
        }
        this.head.next=new_c;
        this.head=d_head;
    }
}
public void delink_compartment(int c_no){
    if(this.head==null){
        System.out.println("Train is Empty");
    }
    else if(this.head.c_no==c_no){
        this.head=this.head.next;
    }
    else{
        Compartment d_head=head;
        Compartment next_c=this.head.next;
        while(next_c!=null){
            if(next_c.c_no==c_no){
                this.head.next=next_c.next;
            }
            this.head=this.head.next;
            next_c=next_c.next;
        }
        this.head=d_head;
    }
}
public void display(){
    Compartment d_head=head;
    System.out.println("Compartments in a train are:");
    while(this.head!=null){
        System.out.println(this.head.c_no);
```

```
        this.head=this.head.next;
      }
      this.head=d_head;
   }
}
class SingleLinkedList{
   public Member[] create_member_list(){
      Scanner sc=new Scanner(System.in);
      int member_id;
      String member_name,start_location,destination_location;
      Member[] m=new Member[60];
      for(int i=0;i<60;i++){
         System.out.println("Member "+(i+1)+"\n--------");
         System.out.println("Enter member id :");
         member_id=sc.nextInt();
         System.out.println("Enter member name :");
         member_name=sc.next();
         System.out.println("Enter member start location :");
         start_location=sc.next();
         System.out.println("Enter member destination location :");
         destination_location=sc.next();
         m[i]=new
Member(member_id,member_name,start_location,destination_location);
      }
      return m;
   }
   public static void main(String args[]){
      Train t=new Train();
      t.append_compartment(1,null);
      t.append_compartment(2,null);
      t.display();
```

```
        t.delink_compartment(2);
        t.display();
    }
}
```

Output