~\OneDrive - VietNam National University - HCM INTERNATIONAL UNIVERSITY\Desktop\DSA\DSA LAB NEW\Lab 4 Linked List\ITITSB22029_DoMinhDuy_Lab4\LinkListApp\LinkListApp.java

```
class Link {
   public int iData; // data item
   public double dData; // data item
   public Link next; // next link in list

   public Link(int id, double dd) { // constructor
      iData = id; // initialize data
      dData = dd; // ('next' is automatically set to null)
   }

   public void displayLink() { // display ourself
      System.out.print("{" + iData + ", " + dData + "} ");
   }

   @Override
   public String toString() {
      return "{" + iData + ", " + dData + "}";
   }
}

class LinkList {
   private Link first; // ref to first link on list

   public LinkList() { // constructor
      first = null; // no links on list yet
   }

   public boolean isEmpty() { // true if list is empty
      return (first == null);
   }

   public void insertFirst(int id, double dd) { // insert at start of list
      Link newLink = new Link(id, dd);
      newLink.next = first; // newLink --> old first
      first = newLink; // first --> newLink
   }

   public Link deleteFirst() { // delete first item
      if (isEmpty()) {
         return null; // if empty, nothing to delete
      }
      Link temp = first; // save reference to link
      first = first.next; // delete it: first-->old next
      return temp; // return deleted link
   }

   public Link getFirst() { // get the first element
      return first;
   }

   public Link getLast() { // get the last element
```

```
52        if (isEmpty()) {
53           return null;
54        }
55        Link current = first;
56        while (current.next != null) { // traverse to the last link
57           current = current.next;
58        }
59        return current;
60     }
61
62     public void displayList() {
63        System.out.print("List (first-->last): ");
64        Link current = first; // start at beginning of list
65        while (current != null) { // until end of list,
66           current.displayLink(); // print data
67           current = current.next; // move to next link
68        }
69        System.out.println("");
70     }
71
72     @Override
73     public String toString() { // Override toString for easy printing
74        StringBuilder listStr = new StringBuilder("List (first-->last): ");
75        Link current = first;
76        while (current != null) {
77           listStr.append(current.toString()).append(" ");
78           current = current.next;
79        }
80        return listStr.toString();
81     }
82  }
83
84  class LinkListApp {
85     public static void main(String[] args) {
86        LinkList theList = new LinkList(); // make new list
87
88        theList.insertFirst(22, 2.99); // insert four items
89        theList.insertFirst(44, 4.99);
90        theList.insertFirst(66, 6.99);
91        theList.insertFirst(88, 8.99);
92
93        System.out.println(theList); // display list using toString()
94
95        while (!theList.isEmpty()) { // until it's empty,
96           Link aLink = theList.deleteFirst(); // delete link
97           System.out.print("Deleted "); // display it
98           System.out.println(aLink);
99        }
100       System.out.println(theList); // display empty list using toString()
101
102       // Example usage of getFirst() and getLast()
103       theList.insertFirst(55, 5.99); // insert one item
104       System.out.println("First element: " + theList.getFirst());
105       System.out.println("Last element: " + theList.getLast());
```

```
    }
}

```