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

```
// linkStack.java
// demonstrates a stack implemented as a list
// to run this program: C>java LinkStackApp
////////////////////////////////////////////////////////////////
class Link {
   public long dData; // data item
   public Link next; // next link in list
   // -------------------------------------------------------------

   public Link(long dd) // constructor
   {
      dData = dd;
   }

   // -------------------------------------------------------------
   public void displayLink() // display ourself
   {
      System.out.print(dData + " ");
   }
} // end class Link
  ////////////////////////////////////////////////////////////////

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

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

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

   // -------------------------------------------------------------
   public void insertFirst(long dd) // insert at start of list
   { // make new link
      Link newLink = new Link(dd);
      newLink.next = first; // newLink --> old first
      first = newLink; // first --> newLink
   }

   // -------------------------------------------------------------
   public long deleteFirst() // delete first item
   { // (assumes list not empty)
      Link temp = first; // save reference to link
      first = first.next; // delete it: first-->old next
      return temp.dData; // return deleted link
```

```
52    }
53
54    // --------------------------------------------------------------
55    public void displayList() {
56       Link current = first; // start at beginning of list
57       while (current != null) // until end of list,
58       {
59          current.displayLink(); // print data
60          current = current.next; // move to next link
61       }
62       System.out.println("");
63    }
64    // --------------------------------------------------------------
65 } // end class LinkList
66   ////////////////////////////////////////////////////////////////
67
68 class LinkStack {
69    private LinkList theList;
70
71    // --------------------------------------------------------------
72    public LinkStack() // constructor
73    {
74       theList = new LinkList();
75    }
76
77    // --------------------------------------------------------------
78    public void push(long j) // put item on top of stack
79    {
80       theList.insertFirst(j);
81    }
82
83    // --------------------------------------------------------------
84    public long pop() // take item from top of stack
85    {
86       return theList.deleteFirst();
87    }
88
89    // --------------------------------------------------------------
90    public boolean isEmpty() // true if stack is empty
91    {
92       return (theList.isEmpty());
93    }
94
95    // --------------------------------------------------------------
96    public void displayStack() {
97       System.out.print("Stack (top-->bottom): ");
98       theList.displayList();
99    }
100   // --------------------------------------------------------------
101 } // end class LinkStack
102   ////////////////////////////////////////////////////////////////
103
104 class LinkStackApp {
105    public static void main(String[] args) {
```

```
106         LinkStack stack = new LinkStack(); // Create a stack
107 
108         // Original list of numbers to reverse
109         long[] numbers = { 10, 20, 30, 40, 50 };
110 
111         System.out.print("Original list: ");
112         for (long num : numbers) {
113            System.out.print(num + " ");
114            stack.push(num); // Push each number onto the stack
115         }
116         System.out.println();
117 
118         System.out.print("Reversed list: ");
119         while (!stack.isEmpty()) {
120            System.out.print(stack.pop() + " "); // Pop each number to reverse the list
121         }
122         System.out.println();
123     }
124 }
```