

Homework Turnin

Email: rgalanos@fcps.edu
Section: 6G
Course: TJHSST APCS 2016-17
Assignment: 04-04
Receipt ID: 793b7d47f82a5f197789478632e242f9

Turnin Successful!

The following file(s) were received:

DLL_Driver.java (5578 bytes)

```
1. // name:    date:
2. public class DLL_Driver
3. {
4.     public static void main(String args[])
5.     {
6.         DLL list = new DLL();
7.
8.         list.addLast("Apple");
9.         list.addLast("Banana");
10.        list.addLast("Cucumber");
11.        list.add("Durian");
12.        list.add("Eggplant");
13.
14.        System.out.println("The list is " + list);
15.        System.out.println("Size: " + list.size());
16.        Object obj = list.remove(2);
17.        System.out.println("Remove index 2: " + obj);
18.        System.out.println("The list is " + list);
19.        System.out.println("Size: " + list.size());
20.
21.        list.add(2, "Carrot");
22.        System.out.println("Add Carrot at index 2:  " + list);
23.
24.        try
25.        {
26.            list.add(16, "Kiwi");    //out-of-bounds
27.        }
28.        catch(IndexOutOfBoundsException e)
29.        {
30.            System.out.println(e);
31.        }
32.
33.
34.        System.out.println("Get values at index 0 and First: " + list.get(0)+" and " + list.getFirst());
35.        System.out.println("No change in list: " +list);
36.
37.        list.removeFirst();
38.        System.out.println("Remove the First:  " + list);
39.
40.        list.addFirst("Artichoke");
41.        System.out.println("Add First:  " + list);
42.        System.out.println("Size: " + list.size());
43.
44.        list.set(1, "Broccoli");
45.        System.out.println("Set value at index 1:  " + list);
46.    }
47. }
48.
49. //////////////////////////////////////
50.
```

```

51. class DLL          //DoubleLinkedList
52. {
53.     private int size;
54.     private DLNode head = new DLNode(); //dummy node--very useful--simplifies the code
55.
56.     public int size()
57.     {
58.         return size;
59.     }
60.
61.     /* appends obj to end of list; increases size;
62.        @return true */
63.     public boolean add(Object obj)
64.     {
65.         addLast(obj);
66.         return true;
67.     }
68.
69.     /* inserts obj at position index. increments size.
70.        */
71.     public void add(int index, Object obj) throws IndexOutOfBoundsException //this the way the real LinkedList i
72.     {
73.         if(index > size || index < 0)
74.             throw new IndexOutOfBoundsException("Index: " + index + ", Size: " + size);
75.         /* enter your code below */
76.         for(int i=0;i<index;i++)
77.             head = head.getNext();
78.         head.setNext(new DLNode(obj, head, head.getNext()));
79.         head.getNext().setPrev(head.getNext());
80.         while(head.getValue()!=null)
81.             head = head.getNext();
82.         size++;
83.     }
84.
85.     /* return obj at position index.
86.        */
87.     public Object get(int index)
88.     {
89.         for(int i=0;i<index+1;i++)
90.             head = head.getNext();
91.         Object obj = head.getValue();
92.         while(head.getValue()!=null)
93.             head = head.getNext();
94.         return obj;
95.     }
96.
97.     /* replaces obj at position index.
98.        */
99.     public void set(int index, Object obj)
100.    {
101.        for(int i=0;i<index;i++)
102.            head = head.getNext();
103.        head.setValue(obj);
104.        while(head.getValue()!=null)
105.            head = head.getNext();
106.    }
107.
108.    /* removes the node from position index. decrements size.
109.       @return the object at position index.
110.       */
111.    public Object remove(int index)
112.    {
113.        if(index > size || index < 0)
114.            throw new IndexOutOfBoundsException("Index: " + index + ", Size: " + size);
115.
116.        for(int i=0;i<index;i++)
117.            head = head.getNext();
118.        Object obj = head.getNext().getValue();
119.        head.setNext(head.getNext().getNext());
120.        head.getNext().setPrev(head);
121.        size--;
122.        while(head.getValue()!=null)
123.            head = head.getNext();
124.        return obj;
125.    }
126.
127.    /* inserts obj at front of list; increases size;
128.       */
129.    public void addFirst(Object obj)
130.    {
131.        head.setNext(new DLNode(obj,head,head.getNext()));

```

```

132.     head.getNext().getNext().setPrev(head.getNext());
133.     size++;
134. }
135.
136. /* appends obj to end of list; increases size;
137.    */
138. public void addLast(Object obj)
139. {
140.     head.setPrev(new DLNode(obj,head.getPrev(),head));
141.     head.getPrev().getPrev().setNext(head.getPrev());
142.     size++;
143. }
144.
145. public Object getFirst()
146. {
147.     return head.getNext().getValue();
148. }
149.
150. public Object getLast()
151. {
152.     return head.getPrev().getValue();
153. }
154.
155. public Object removeFirst()
156. {
157.     Object obj = head.getNext().getValue();
158.     head.setNext(head.getNext().getNext());
159.     head.getNext().setPrev(head);
160.     size--;
161.     return obj;
162. }
163.
164. public Object removeLast()
165. {
166.     Object obj = head.getPrev().getValue();
167.     head.setPrev(head.getPrev().getPrev());
168.     head.getPrev().setNext(head);
169.     size--;
170.     return obj;
171. }
172.
173. public String toString()
174. {
175.     String list = "";
176.     head = head.getNext();
177.     list += "[";
178.     while(head.getNext().getValue()!=null)
179.     {
180.         list += head.getValue() + ", ";
181.         head = head.getNext();
182.     }
183.     list+=head.getValue();
184.     list += "]";
185.     head = head.getNext();
186.     return list;
187. }
188. }
189.
190. //////////////////////////////////////////////////
191.
192. class DLNode
193. {
194.     private Object value;
195.     private DLNode prev;
196.     private DLNode next;
197.     public DLNode(Object arg, DLNode p, DLNode n)
198.     {
199.         value=arg;
200.         prev=p;
201.         next=n;
202.     }
203.     public DLNode()
204.     {
205.         value=null;
206.         next=this;
207.         prev=this;
208.     }
209.     public void setValue(Object arg)
210.     {
211.         value=arg;
212.     }

```

```
213. public void setNext(DLNode arg)
214. {
215.     next=arg;
216. }
217. public void setPrev(DLNode arg)
218. {
219.     prev=arg;
220. }
221. public DLNode getNext()
222. {
223.     return next;
224. }
225. public DLNode getPrev()
226. {
227.     return prev;
228. }
229. public Object getValue()
230. {
231.     return value;
232. }
233. }
234.
```