

DoublyLLIterator.java

```
1 package dataStructures;
2
3 class DoublyLLIterator<E> implements TwoWayIterator<E>
4 {
5
6     static final long serialVersionUID = 0L;
7
8
9     // Node with the first element in the iteration.
10    protected DListNode<E> firstNode;
11
12    // Node with the last element in the iteration.
13    protected DListNode<E> lastNode;
14
15    // Node with the next element in the iteration.
16    protected DListNode<E> nextToReturn;
17
18    // Node with the previous element in the iteration.
19    protected DListNode<E> prevToReturn;
20
21
22    public DoublyLLIterator( DListNode<E> first, DListNode<E> last )
23    {
24        firstNode = first;
25        lastNode = last;
26        this.rewind();
27    }
28
29
30    // Restarts the iteration.
31    // After rewind, if the iteration is not empty,
32    // next will return the first element in the iteration.
33    public void rewind( )
34    {
35        nextToReturn = firstNode;
36        prevToReturn = null;
37    }
38
39
40    // Restarts the iteration in the reverse direction.
41    // After fullForward, if the iteration is not empty,
```

DoublyLLIterator.java

```
42 // previous will return the last element in the iteration.
43 public void fullForward( )
44 {
45     prevToReturn = lastNode;
46     nextToReturn = null;
47 }
48
49
50 // Returns true iff the iteration has more elements.
51 // In other words, returns true if next would return an element
52 // rather than throwing an exception.
53 public boolean hasNext( )
54 {
55     return nextToReturn != null;
56 }
57
58
59 // Returns true iff the iteration has more elements
60 // in the reverse direction.
61 // In other words, returns true if previous would return an
element
62 // rather than throwing an exception.
63 public boolean hasPrevious( )
64 {
65     return prevToReturn != null;
66 }
67
68
69 // Returns the next element in the iteration.
70 public E next( ) throws NoSuchElementException
71 {
72     if ( !this.hasNext( ) )
73         throw new NoSuchElementException();
74
75     E element = nextToReturn.getElement();
76     prevToReturn = nextToReturn.getPrevious();
77     nextToReturn = nextToReturn.getNext();
78     return element;
79 }
80
81
```

DoublyLLIterator.java

```
82 // Returns the previous element in the iteration.
83 public E previous( ) throws NoSuchElementException
84 {
85     if ( !this.hasPrevious() )
86         throw new NoSuchElementException();
87
88     E element = prevToReturn.getElement();
89     nextToReturn = prevToReturn.getNext();
90     prevToReturn = prevToReturn.getPrevious();
91     return element;
92 }
93
94
95 }
96
```