

```

import java.util.Iterator;
import java.util.NoSuchElementException;

public class DoublyLinkedList<T extends Comparable<T>> implements Iterable<T> {
    protected class Node<E> {
        E data;
        Node<E> previous;
        Node<E> next;
        Node() {
            data = null;
            previous = null;
            next = null;
        }
        Node(E d) {
            data = d;
            previous = null;
            next = null;
        }
        Node(E d, Node<E> p, Node<E> n) {
            data = d;
            previous = p;
            next = n;
        }
    }
}

```

```

public class ListIterator<E> implements Iterator<E> {
    private Node<E> list;
    private int count, current;

    public ListIterator(Node<E> head, int c){
        list = head;
        count = c;
        current = 0;
    }

    public boolean hasNext() {
        return current < count;
    }

    public E next() {
        if(!hasNext())
            throw new NoSuchElementException();
        E data = list.data;
        list = list.next;
        current++;
        return data;
    }
    /* add code if you need to delete while traversing
    http://docs.oracle.com/javase/7/docs/api/java/util/Iterator.html
    */
    public void remove() {
    }
}

```

```

private Node<T> head, tail;
private int size;

```

```
public Iterator<T> iterator() {  
    return new ListIterator<T>(head, size);  
}
```

```
public String toString(){  
    String str = "[";  
  
    Iterator<T> iter = iterator();  
  
    while(iter.hasNext())  
        str += iter.next() + (iter.hasNext() ? ", " : "]);  
  
    return str;  
}
```

```
...  
}
```

```
public class Sample {  
  
    public static void main(String[] args) {  
        DoublyLinkedList<String> dll = new DoublyLinkedList<String>();  
        dll.add("Hello");  
        dll.add("this");  
        dll.add("is");  
        dll.add("a");  
        dll.add("test");  
        System.out.println(dll);  
        try {  
            System.out.println(dll.first());  
            System.out.println(dll.last());  
        } catch (EmptyListException e) {  
  
            e.printStackTrace();  
        }  
  
        Iterator<String> iter = dll.iterator();  
  
        while (iter.hasNext())  
            System.out.print(iter.next() + (iter.hasNext() ? ", " : ""));  
        System.out.println();  
    }  
}
```

Output:

```
[Hello, this, is, a, test]
```

```
Hello
```

```
test
```

```
Hello, this, is, a, test
```