PACKAGE CLASS TREE DEPRECATED INDEX HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES ALL CLASSES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Class MyLinkedList<E>

```
java.lang.Object
    java.util.AbstractCollection<E>
        java.util.AbstractList<E>
            MyLinkedList<E>
```

All Implemented Interfaces:

`java.lang.Iterable<E>, java.util.Collection<E>, java.util.List<E>`

```
public class MyLinkedList<E>
extends java.util.AbstractList<E>
```

The class MyLinkedList extends the AbstractList

Nested Class Summary

Nested Classes

Modifier and Type	Class and Description
`protected class`	**`MyLinkedList.MyListIterator`** This is an inner class, called MyListIterator.
`protected class`	**`MyLinkedList.Node<E>`** This is an inner class, called Node, to represent a node in the linked list.

Field Summary

Fields

Modifier and Type	Field and Description
`private MyLinkedList.Node<E>`	**`head`**
`private int`	**`size`**
`private MyLinkedList.Node<E>`	**`tail`**

Fields inherited from class java.util.AbstractList

`modCount`

Constructor Summary

Constructors

Constructor and Description
MyLinkedList() The no-arg constructor for MyLinkedList that creates an empty list and initializes all the necessary variables to track the list

Method Summary

All Methods **Instance Methods** **Concrete Methods**

Modifier and Type	Method and Description
boolean	**add**(**E** data) Method: add A boolean add method that will presumably always return true; optional operation
void	**add**(int index, **E** data) Method: add A method that inserts the specified element at the specified position in the list; optional operation
void	**clear**() Method: clear A method that removes all of the elements from the list
E	**get**(int index) Method: get A method that gets contents at position [index]
private **MyLinkedList.Node**	**getNth**(int index) Method: getNth A method that returns the Node at a specified index, not the content
boolean	**isEmpty**() Method: isEmpty A method that determines if the list is empty
java.util.Iterator<**E**>	**iterator**()
java.util.ListIterator<**E**>	**listIterator**()
java.util.Iterator<**E**>	**QQQiterator**()
java.util.ListIterator<**E**>	**QQQlistIterator**()
E	**remove**(int index) Method: remove A method that removes the element from position [i] in the list
E	**set**(int index, **E** element)

	Method: set A method that sets the value at index [index] to [element]
`int`	`size()` Method: size A method that returns the number of elements being stored

Methods inherited from class java.util.AbstractList

`addAll, equals, hashCode, indexOf, lastIndexOf, listIterator, removeRange, subList`

Methods inherited from class java.util.AbstractCollection

`addAll, contains, containsAll, remove, removeAll, retainAll, toArray, toArray, toString`

Methods inherited from class java.lang.Object

`clone, finalize, getClass, notify, notifyAll, wait, wait, wait`

Methods inherited from interface java.util.List

`addAll, contains, containsAll, remove, removeAll, replaceAll, retainAll, sort, spliterator, toArray, toArray`

Methods inherited from interface java.util.Collection

`parallelStream, removeIf, stream`

Methods inherited from interface java.lang.Iterable

`forEach`

Field Detail

size

```
private int size
```

head

```
private MyLinkedList.Node<E> head
```

tail

```
private MyLinkedList.Node<E> tail
```

Constructor Detail

MyLinkedList

```
public MyLinkedList()
```

The no-arg constructor for MyLinkedList that creates an empty list and initializes all the necessary variables to track the list

Method Detail

getNth

```
private MyLinkedList.Node getNth(int index)
                          throws java.lang.IndexOutOfBoundsException
```

Method: getNth A method that returns the Node at a specified index, not the content

Parameters:

`index: - the index at which the Node is returned`

Returns:

`Node at a specified index`

Throws:

`java.lang.IndexOutOfBoundsException - if the index is out of range`

add

```
public boolean add(E data)
            throws java.lang.NullPointerException
```

Method: add A boolean add method that will presumably always return true; optional operation

Specified by:

`add in interface java.util.Collection<E>`

Specified by:

`add in interface java.util.List<E>`

Overrides:

`add in class java.util.AbstractList<E>`

Parameters:

`data: - the specified element to be appended to the end of the list`

Returns:

`true, as specified by Collection.add(E)`

Throws:

`java.lang.NullPointerException - if the specified element is null and this list does not permit null elements`

add

```
public void add(int index,
                E data)
```

Method: add A method that inserts the specified element at the specified position in the list; optional operation

Specified by:

`add in interface java.util.List<E>`

Overrides:

`add in class java.util.AbstractList<E>`

Parameters:

`index: - the specified position`

`data: - the specified element to be added`

Throws:

`java.lang.IndexOutOfBoundsException - if the index is out of range`

`java.lang.NullPointerException - if the specified element is null and this list does not permit null elements`

get

```
public E get(int index)
      throws java.lang.IndexOutOfBoundsException
```

Method: get A method that gets contents at position [index]

Specified by:

`get in interface java.util.List<E>`

Specified by:

`get in class java.util.AbstractList<E>`

Parameters:

`index: - the index that specifies the position`

Returns:

`the contents at the position`

Throws:

`java.lang.IndexOutOfBoundsException` - if the index is out of range

set

```
public E set(int index,
             E element)
```

Method: set A method that sets the value at index [index] to [element]

Specified by:

`set` in interface `java.util.List<E>`

Overrides:

`set` in class `java.util.AbstractList<E>`

Parameters:

`index:` - the position at which the value is set

`element:` - the value to be set

Returns:

the element previously at the specified position

Throws:

`java.lang.NullPointerException` - if the specified element is null and this list does not permit null elements

`java.lang.IndexOutOfBoundsException` - if the index is out of range

remove

```
public E remove(int index)
```

Method: remove A method that removes the element from position [i] in the list

Specified by:

`remove` in interface `java.util.List<E>`

Overrides:

`remove` in class `java.util.AbstractList<E>`

Parameters:

`index` - the index of the element to be removed

Returns:

the element previously at the specified position

Throws:

`java.lang.IndexOutOfBoundsException` - if index is out of range

clear

```
public void clear()
```

Method: clear A method that removes all of the elements from the list

Specified by:

`clear in interface java.util.Collection<E>`

Specified by:

`clear in interface java.util.List<E>`

Overrides:

`clear in class java.util.AbstractList<E>`

isEmpty

```
public boolean isEmpty()
```

Method: isEmpty A method that determines if the list is empty

Specified by:

`isEmpty in interface java.util.Collection<E>`

Specified by:

`isEmpty in interface java.util.List<E>`

Overrides:

`isEmpty in class java.util.AbstractCollection<E>`

Returns:

`true if empty false otherwise`

size

```
public int size()
```

Method: size A method that returns the number of elements being stored

Specified by:

`size in interface java.util.Collection<E>`

Specified by:

`size in interface java.util.List<E>`

Specified by:

`size in class java.util.AbstractCollection<E>`

QQQiterator

```
public java.util.Iterator<E> QQQiterator()
```

QQQlistIterator

```
public java.util.ListIterator<E> QQQlistIterator()
```

iterator

```
public java.util.Iterator<E> iterator()
```

Specified by:

`iterator in interface java.lang.Iterable<E>`

Specified by:

`iterator in interface java.util.Collection<E>`

Specified by:

`iterator in interface java.util.List<E>`

Overrides:

`iterator in class java.util.AbstractList<E>`

listIterator

```
public java.util.ListIterator<E> listIterator()
```

Specified by:

`listIterator in interface java.util.List<E>`

Overrides:

`listIterator in class java.util.AbstractList<E>`