

Package [utility](#)

Class `LinkedList<E>`

`java.lang.Object`
`utility.LinkedList<E>`

All Implemented Interfaces:

`List<E>`

```
public class LinkedList<E>
    extends java.lang.Object
    implements List<E>
```

Nested Class Summary

Nested Classes

Modifier and Type	Class	Description
class	<code>LinkedList.LinkedListIterator</code>	

Constructor Summary

Constructors

Constructor	Description
<code>LinkedList()</code>	creates linked list object

Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
void	<code>add</code> (int index, E item)	inserts the item at the given index in the list.
boolean	<code>add(E item)</code>	appends the item specified to the end of the list.
void	<code>clear()</code>	clears list of all elements, return size back to zero.
boolean	<code>contains</code> (E item)	searches for an item and returns true if in the array,

Modifier and Type	Method	Description
E	get (int index)	returns the item at the specified position in the list.
int	indexOf (E item)	searches for an item and returns the first occurrence in the array, otherwise returns -1, if NOT found.
boolean	isEmpty()	returns true, if the list is empty,
Iterator<E>	iterator()	returns an object used to traverse the elements in list
E	remove (int index)	removes the item at the given index in the list.
boolean	remove (E item)	removes the first occurrence of the specified item from the list, if present.
E	set (int index, E item)	replaces the item at the specified position with the one passed.
int	size()	returns the number of the elements in the list.
java.lang.String	toString()	displays the contents of the list.

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Details

LinkedList

```
public LinkedList()
```

creates linked list object

Method Details

add

```
public boolean add(E item)
```

appends the item specified to the end of the list.

Specified by:

add in interface `List<E>`

Parameters:

item - in the list

Returns:

boolean value if successful.

add

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

inserts the item at the given index in the list.

Specified by:

add in interface `List<E>`

Parameters:

index - given in the list.

item - in the list.

clear

```
public void clear()
```

clears list of all elements, return size back to zero.

Specified by:

clear in interface `List<E>`

contains

```
public boolean contains(E item)
```

searches for an item and returns true if in the array,

Specified by:

contains in interface `List<E>`

Parameters:

item - to search for in list.

Returns:

boolean value.

get

```
public E get(int index)
```

returns the item at the specified position in the list.

Specified by:

`get` in interface `List<E>`

Parameters:

index - of item in list.

Returns:

item at index.

indexOf

```
public int indexOf(E item)
```

searches for an item and returns the first occurrence in the array, otherwise returns -1, if NOT found.

Specified by:

`indexOf` in interface `List<E>`

Parameters:

item - to search for in list.

Returns:

location of item, if found.

isEmpty

```
public boolean isEmpty()
```

returns true, if the list is empty,

Specified by:

`isEmpty` in interface `List<E>`

Returns:

boolean value

iterator

```
public Iterator<E> iterator()
```

returns an object used to traverse the elements in list

Specified by:

`iterator` in interface `List<E>`

Returns:

iterator for list

remove

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

removes the item at the given index in the list.

Specified by:

[remove](#) in interface `List<E>`

Parameters:

index - of item in list.

Returns:

old value in list.

remove

```
public boolean remove(E item)
```

removes the first occurrence of the specified item from the list, if present.

Specified by:

[remove](#) in interface `List<E>`

Parameters:

item - to remove from list.

Returns:

boolean value.

set

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

replaces the item at the specified position with the one passed.

Specified by:

[set](#) in interface `List<E>`

Parameters:

index - to replace list item.

item - that replaces one in list.

Returns:

old item.

size

```
public int size()
```

returns the number of the elements in the list.

Specified by:

`size` in interface `List<E>`

Returns:

size of the list.

toString

```
public java.lang.String toString()
```

displays the contents of the list.

Overrides:

`toString` in class `java.lang.Object`

Returns:

string representation of list