Class MyLinkedList.MyListIterator

java.lang.Object
    MyLinkedList.MyListIterator

All Implemented Interfaces:

`java.util.Iterator<E>, java.util.ListIterator<E>`

Enclosing class:

`MyLinkedList<E>`

```
protected class MyLinkedList.MyListIterator
extends java.lang.Object
implements java.util.ListIterator<E>
```

This is an inner class, called MyListIterator. It has access to all of the MyLinkedList instance variables, including private variables. To access the set() method of the MyLinkedList instance from within a methodhere use MyLinkedList.this.set(i,e);

Field Summary

Fields

Modifier and Type	Field and Description
(package private) `MyLinkedList.Node<E>`	**`back`**
(package private) `MyLinkedList.Node<E>`	**`front`**
(package private) int	**`index`**
(package private) boolean	**`nextcall`**
(package private) boolean	**`prevcall`**

Constructor Summary

Constructors

Constructor and Description
`MyListIterator()` no-arg constructor for MyListIterator

Method Summary

All Methods | Instance Methods | Concrete Methods

Modifier and Type	Method and Description
void	**add**(**E** e) Method: add Insert the given item into the list immediately before whatever would have been returned by a call to next()
boolean	**hasNext**() Method: hasNext Return true if there are more elements when going in the forward direction
boolean	**hasPrevious**() Method: hasPrevious Return true if there are more elements when going in the reverse direction
E	**next**() Method: next Return the next element in the list when going forward
int	**nextIndex**() Method: nextIndex Return the index of the element that would be returned by a call to next()
E	**previous**() Method: previous Return the next element in the list when going backwards
int	**previousIndex**() Method: previousIndex Return the index of the element that would be returned by a call to previous()
void	**remove**() Method: remove Remove the last element returned by the most recent call to either next/previous
void	**set**(**E** e) Method: set Change the value in the node returned by the most recent next/previous with the new value

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface java.util.Iterator

forEachRemaining

Field Detail

front

```
MyLinkedList.Node<E> front
```

back

```
MyLinkedList.Node<E> back
```

index

```
int index
```

nextcall

```
boolean nextcall
```

prevcall

```
boolean prevcall
```

Constructor Detail

MyListIterator

```
public MyListIterator()
```

no-arg constructor for MyListIterator

Method Detail

add

```
public void add(E e)
         throws java.lang.NullPointerException
```

Method: add Insert the given item into the list immediately before whatever would have been returned by a call to next()

Specified by:

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

Parameters:

`e - the element to insert`

Throws:

`java.lang.NullPointerException - if e is null`

hasNext

```
public boolean hasNext()
```

Method: hasNext Return true if there are more elements when going in the forward direction

Specified by:

`hasNext in interface java.util.Iterator<E>`

Specified by:

`hasNext in interface java.util.ListIterator<E>`

Returns:

`true if more element forward`

hasPrevious

```
public boolean hasPrevious()
```

Method: hasPrevious Return true if there are more elements when going in the reverse direction

Specified by:

`hasPrevious in interface java.util.ListIterator<E>`

Returns:

`true if more element backward`

next

```
public E next()
       throws java.util.NoSuchElementException
```

Method: next Return the next element in the list when going forward

Specified by:

`next in interface java.util.Iterator<E>`

Specified by:

`next in interface java.util.ListIterator<E>`

Returns:

`the next element in the list`

Throws:

`java.util.NoSuchElementException - if the iteration has no next element`

nextIndex

```
public int nextIndex()
```

Method: nextIndex Return the index of the element that would be returned by a call to next()

Specified by:

`nextIndex in interface java.util.ListIterator<E>`

Returns:

`the index of the element that would be returned by a subsequent call to next; or the list size if at the end of the list`

previous

```
public E previous()
           throws java.util.NoSuchElementException
```

Method: previous Return the next element in the list when going backwards

Specified by:

`previous in interface java.util.ListIterator<E>`

Returns:

`the previous element in the list`

Throws:

`java.util.NoSuchElementException - if the iteration has no previous element`

previousIndex

```
public int previousIndex()
```

Method: previousIndex Return the index of the element that would be returned by a call to previous()

Specified by:

`previousIndex in interface java.util.ListIterator<E>`

Returns:

`the index of the element backward; or -1 if at the start of the list`

remove

```
public void remove()
            throws java.lang.IllegalStateException
```

Method: remove Remove the last element returned by the most recent call to either next/previous

Specified by:

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

Specified by:

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

Throws:

`java.lang.IllegalStateException`

set

```
public void set(E e)
```

Method: set Change the value in the node returned by the most recent next/previous with the new value

Specified by:

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

Parameters:

`e - the element with which to replace the last element returned by next or previous`