Class DLinkedList<E>

java.lang.Object
    DLinkedList<E>

---

public class **DLinkedList<E>**
extends java.lang.Object

Constructor Summary

Constructors

Constructor and Description
DLinkedList()

Method Summary

All Methods	Instance Methods	Concrete Methods

Modifier and Type	Method and Description
void	**add**(int index, **E** item) Add the given item at the given index
void	**addAll**(int index, java.lang.Iterable<**E**> collection) Inserts all elements of the given collection
void	**addFirst**(**E** item) Adds the given element to the front of the list.
void	**addLast**(**E** item) Adds the given item to the end of the list
void	**clear**() clears all elements in the list return the cleared

list

java.lang.Object	**clone**()	
	returns a copy of the list	
boolean	**contains**(**E** items)	
	Determines if the list contains the given item	
boolean	**containsIter**(**E** items)	
	Determines if the list contains the given item	
boolean	**equals**(java.lang.Object list)	
	determine is this list is equal to the given list	
E	**getFirst**()	
	Returns the first element in the list.	
E	**getLast**()	
	Returns the last element in the list.	
boolean	**isEmpty**()	
	If the list is empty	
java.util.Iterator<**E**>	**iterator**()	
void	**remove**()	
	Remove the element in the list	
boolean	**removeAll**(**E** item)	
	removes all occurrences of the given item in the list	
boolean	**removeAllIter**(**E** item)	
	Same as removeAll but uses an iterator	
boolean	**removeItem**(**E** item)	
	removes the first occurence of the given item in the list	
void	**test_iter_fails**()	
java.lang.String	**toString**()	
	changing integers to string return string representation of the list	
java.lang.String	**toStringRev**()	
	reversing toString method	

Methods inherited from class java.lang.Object

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

Constructor Detail

DLinkedList

public DLinkedList()

Method Detail

isEmpty

public boolean isEmpty()

If the list is empty

Returns:
the data null;

getFirst

public E getFirst()

Returns the first element in the list.

Returns:
the first element in the list

Throws:
java.util.NoSuchElementException - when the list is empty

getLast

public E getLast()

Returns the last element in the list.

Returns:

the last element in the list.

Throws:

java.util.NoSuchElementException - when list is empty

addLast

public void addLast(E item)

Adds the given item to the end of the list

Parameters:

item - the element to add

addFirst

public void addFirst(E item)

Adds the given element to the front of the list.

Parameters:

data - the element to add

add

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

Add the given item at the given index

Parameters:

index - to the node that you are adding

item - added to the list

contains

public boolean contains(E items)

Determines if the list contains the given item

Parameters:

items - in the given list

Returns:
```
true
```

clear

```
public void clear()
```

clears all elements in the list return the cleared list

toString

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

changing integers to string return string representation of the list

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

toStringRev

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

reversing toString method

Returns:
```
the string
```

containsIter

```
public boolean containsIter(E items)
```

Determines if the list contains the given item

Parameters:
```
items - visit the nodes as it searches for the item
```

Returns:
```
null
```

iterator

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

removeItem

```
public boolean removeItem(E item)
```

removes the first occurence of the given item in the list

Parameters:
```
item - is the given item
```
Returns:
```
true if the element is removed
```

removeAll

```
public boolean removeAll(E item)
```

removes all occurrences of the given item in the list

Parameters:
```
item - is the item we remove in all occurences
```
Returns:
```
true if the value of the element was removed
```

equals

```
public boolean equals(java.lang.Object list)
```

determine is this list is equal to the given list

Overrides:
```
equals in class java.lang.Object
```
Parameters:
```
list - return
```

clone

```
public java.lang.Object clone()
```

returns a copy of the list

Overrides:

clone in class java.lang.Object

---

addAll

public void addAll(int index,
                   java.lang.Iterable<E> collection)

Inserts all elements of the given collection

Parameters:

index - in which we give the collection

collection - all elements that were inserted

---

remove

public void remove()

Remove the element in the list

---

removeAllIter

public boolean removeAllIter(E item)

Same as removeAll but uses an iterator

Parameters:

item - that is being removed

Returns:

the item being removed

---

test_iter_fails

public void test_iter_fails()