JavaScript is disabled on your browser.

[Skip navigation links](#1fob9te)

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/BasicLinkedList.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-files/index-1.html)
* [Help](http://docs.google.com/help-doc.html)
* Prev Class
* [Next Class](http://docs.google.com/listClasses/SortedLinkedList.html)
* [Frames](http://docs.google.com/index.html?listClasses/BasicLinkedList.html)
* [No Frames](http://docs.google.com/BasicLinkedList.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#3znysh7) |
* [Method](#2et92p0)
* Detail:
* Field |
* [Constr](#1t3h5sf) |
* [Method](#2s8eyo1)

listClasses

## Class BasicLinkedList<T>

* java.lang.Object
  + listClasses.BasicLinkedList<T>
* All Implemented Interfaces: java.lang.Iterable<T> Direct Known Subclasses: [SortedLinkedList](http://docs.google.com/listClasses/SortedLinkedList.html)  
    
  public class BasicLinkedList<T>  
  extends java.lang.Object  
  implements java.lang.Iterable<T>  
  This generic singly-linked list relies on a head (reference to first element of the list) and tail (reference to the last element of the list). Both are set to null when the list is empty. Both point to the same element when there is only one element in the list. A node structure has only two fields: data and next reference. The class must only define the following entities: a class Node, head and tail references and an integer representing the list size. All the entities are defined as protected so they can be accessed by the subclass.Author: Dept of Computer Science, UMCP

### Constructor SummaryConstructors

|  |
| --- |
| * + Constructor and Description |
| * + [BasicLinkedList](http://docs.google.com/listClasses/BasicLinkedList.html#BasicLinkedList--)() Defines an empty linked list. |

### Method SummaryAll Methods Instance Methods Concrete Methods

|  |  |
| --- | --- |
| * + Modifier and Type | * + Method and Description |
| * + [BasicLinkedList](http://docs.google.com/listClasses/BasicLinkedList.html)<[T](http://docs.google.com/listClasses/BasicLinkedList.html)> | * + [addToEnd](http://docs.google.com/listClasses/BasicLinkedList.html#addToEnd-T-)([T](http://docs.google.com/listClasses/BasicLinkedList.html) data) Adds element to the end of the list. |
| * + [BasicLinkedList](http://docs.google.com/listClasses/BasicLinkedList.html)<[T](http://docs.google.com/listClasses/BasicLinkedList.html)> | * + [addToFront](http://docs.google.com/listClasses/BasicLinkedList.html#addToFront-T-)([T](http://docs.google.com/listClasses/BasicLinkedList.html) data) Adds element to the front of the list. |
| * + [T](http://docs.google.com/listClasses/BasicLinkedList.html) | * + [getFirst](http://docs.google.com/listClasses/BasicLinkedList.html#getFirst--)() Returns but does not remove the first element from the list. |
| * + [T](http://docs.google.com/listClasses/BasicLinkedList.html) | * + [getLast](http://docs.google.com/listClasses/BasicLinkedList.html#getLast--)() Returns but does not remove the last element from the list. |
| * + java.util.ArrayList<[T](http://docs.google.com/listClasses/BasicLinkedList.html)> | * + [getReverseArrayList](http://docs.google.com/listClasses/BasicLinkedList.html#getReverseArrayList--)() Returns an ArrayList with the element of the linked list in reverse order. |
| * + [BasicLinkedList](http://docs.google.com/listClasses/BasicLinkedList.html)<[T](http://docs.google.com/listClasses/BasicLinkedList.html)> | * + [getReverseList](http://docs.google.com/listClasses/BasicLinkedList.html#getReverseList--)() Returns a new list with the elements of the current list in reverse order. |
| * + int | * + [getSize](http://docs.google.com/listClasses/BasicLinkedList.html#getSize--)() Notice you must not traverse the list to compute the size. |
| * + java.util.Iterator<[T](http://docs.google.com/listClasses/BasicLinkedList.html)> | * + [iterator](http://docs.google.com/listClasses/BasicLinkedList.html#iterator--)() This method must be implemented using an anonymous inner class that defines the iterator. |
| * + [BasicLinkedList](http://docs.google.com/listClasses/BasicLinkedList.html)<[T](http://docs.google.com/listClasses/BasicLinkedList.html)> | * + [remove](http://docs.google.com/listClasses/BasicLinkedList.html#remove-T-java.util.Comparator-)([T](http://docs.google.com/listClasses/BasicLinkedList.html) targetData, java.util.Comparator<[T](http://docs.google.com/listClasses/BasicLinkedList.html)> comparator) Removes ALL instances of targetData from the list. |
| * + [T](http://docs.google.com/listClasses/BasicLinkedList.html) | * + [retrieveFirstElement](http://docs.google.com/listClasses/BasicLinkedList.html#retrieveFirstElement--)() Removes and returns the first element from the list. |
| * + [T](http://docs.google.com/listClasses/BasicLinkedList.html) | * + [retrieveLastElement](http://docs.google.com/listClasses/BasicLinkedList.html#retrieveLastElement--)() Removes and returns the last element from the list. |

### Methods inherited from class java.lang.Objectequals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Methods inherited from interface java.lang.IterableforEach, spliterator

### Constructor Detail

#### BasicLinkedList public BasicLinkedList() Defines an empty linked list. No nodes are created. We do not use dummy nodes for this list (if you don't know what a dummy list is don't worry about it).

### Method Detail

#### getSize public int getSize() Notice you must not traverse the list to compute the size. This method just returns the value of the instance variable you use to keep track of size.Returns: list size

#### addToEnd public [BasicLinkedList](http://docs.google.com/listClasses/BasicLinkedList.html)<[T](http://docs.google.com/listClasses/BasicLinkedList.html)> addToEnd([T](http://docs.google.com/listClasses/BasicLinkedList.html) data) Adds element to the end of the list. Do not use iterators to implement this method.Parameters: data - Returns: reference to current object

#### addToFront public [BasicLinkedList](http://docs.google.com/listClasses/BasicLinkedList.html)<[T](http://docs.google.com/listClasses/BasicLinkedList.html)> addToFront([T](http://docs.google.com/listClasses/BasicLinkedList.html) data) Adds element to the front of the list. Do not use iterators to implement this method.Parameters: data - Returns: reference to current object

#### getFirst public [T](http://docs.google.com/listClasses/BasicLinkedList.html) getFirst() Returns but does not remove the first element from the list. If there are no elements the method returns null. Do not implement this method using iterators.Returns: data element or null

#### getLast public [T](http://docs.google.com/listClasses/BasicLinkedList.html) getLast() Returns but does not remove the last element from the list. If there are no elements the method returns null. Do not implement this method using iterators.Returns: data element or null

#### retrieveFirstElement public [T](http://docs.google.com/listClasses/BasicLinkedList.html) retrieveFirstElement() Removes and returns the first element from the list. If there are no elements the method returns null. Do not implement this method using iterators.Returns: data element or null

#### retrieveLastElement public [T](http://docs.google.com/listClasses/BasicLinkedList.html) retrieveLastElement() Removes and returns the last element from the list. If there are no elements the method returns null. Do not implement implement this method using iterators.Returns: data element or null

#### remove public [BasicLinkedList](http://docs.google.com/listClasses/BasicLinkedList.html)<[T](http://docs.google.com/listClasses/BasicLinkedList.html)> remove([T](http://docs.google.com/listClasses/BasicLinkedList.html) targetData, java.util.Comparator<[T](http://docs.google.com/listClasses/BasicLinkedList.html)> comparator) Removes ALL instances of targetData from the list. Notice that you must remove the elements by performing a single traversal over the list. You may not use any of the other retrieval methods associated with the class in order to complete the removal process. You must use the provided comparator (do not use equals) to find those elements that match the target. Do not implement implement this method using iterators.Parameters: targetData - comparator - Returns: reference to current object

#### iterator public java.util.Iterator<[T](http://docs.google.com/listClasses/BasicLinkedList.html)> iterator() This method must be implemented using an anonymous inner class that defines the iterator. Remember that we should be able to call the hasNext() method as many times as we want without changing what is considered the next element. Only students in the honor section will implement the iterator's remove() method. If you don't need to implement the iterator's remove() method, throw UnsupportedOperationException.Specified by: iterator in interface java.lang.Iterable<[T](http://docs.google.com/listClasses/BasicLinkedList.html)> Returns: iterator

#### getReverseArrayList public java.util.ArrayList<[T](http://docs.google.com/listClasses/BasicLinkedList.html)> getReverseArrayList() Returns an ArrayList with the element of the linked list in reverse order. This method must be implemented using recursion.Returns: ArrayList

#### getReverseList public [BasicLinkedList](http://docs.google.com/listClasses/BasicLinkedList.html)<[T](http://docs.google.com/listClasses/BasicLinkedList.html)> getReverseList() Returns a new list with the elements of the current list in reverse order. You can assume sharing of data of each node is fine. This method must be implemented using recursion.Returns: list

[Skip navigation links](#qsh70q)

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/BasicLinkedList.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-files/index-1.html)
* [Help](http://docs.google.com/help-doc.html)
* Prev Class
* [Next Class](http://docs.google.com/listClasses/SortedLinkedList.html)
* [Frames](http://docs.google.com/index.html?listClasses/BasicLinkedList.html)
* [No Frames](http://docs.google.com/BasicLinkedList.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#3znysh7) |
* [Method](#2et92p0)
* Detail:
* Field |
* [Constr](#1t3h5sf) |
* [Method](#2s8eyo1)