Tsegazeab Berhie

CMSC 204

03/07/2021

Pseudocode

Use the class BasicDoubleLinkedList and extends SortedDoubleLinkedList

public class **BasicDoubleLinkedList<T>**

extends java.lang.Object

implements java.lang.Iterable<T>

Use the constructor [**BasicDoubleLinkedList**](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378352_1/courses/23685.202120/BasicDoubleLinkedList.html#BasicDoubleLinkedList())()

Method [BasicDoubleLinkedList](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378352_1/courses/23685.202120/BasicDoubleLinkedList.html)<[T](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378352_1/courses/23685.202120/BasicDoubleLinkedList.html)>

[**addToEnd**](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378352_1/courses/23685.202120/BasicDoubleLinkedList.html#addToEnd(T))([T](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378352_1/courses/23685.202120/BasicDoubleLinkedList.html) data)  
 Adds an element to the end of the list.

**Parameters:**

data - the data for the Node within the linked list

**Returns:**

reference to the current object

[**addToFront**](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378352_1/courses/23685.202120/BasicDoubleLinkedList.html#addToFront(T))([T](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378352_1/courses/23685.202120/BasicDoubleLinkedList.html) data)  
 Adds element to the front of the list.

[**getFirst**](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378352_1/courses/23685.202120/BasicDoubleLinkedList.html#getFirst())()  
 Returns but does not remove the first element from the list.

[**getLast**](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378352_1/courses/23685.202120/BasicDoubleLinkedList.html#getLast())()  
 Returns but does not remove the last element from the list.

[**getSize**](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378352_1/courses/23685.202120/BasicDoubleLinkedList.html#getSize())()  
 **Returns:** the size of the linked list

[**iterator**](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378352_1/courses/23685.202120/BasicDoubleLinkedList.html#iterator())()  
 This method must be implemented using an inner class that implements ListIterator and defines the methods of hasNext(), next(), hasPrevious() and previous().

[**remove**](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378352_1/courses/23685.202120/BasicDoubleLinkedList.html#remove(T,%20java.util.Comparator))([T](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378352_1/courses/23685.202120/BasicDoubleLinkedList.html) targetData, java.util.Comparator<[T](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378352_1/courses/23685.202120/BasicDoubleLinkedList.html)> comparator)  
 Removes the first instance of the targetData from the list.

[**retrieveFirstElement**](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378352_1/courses/23685.202120/BasicDoubleLinkedList.html#retrieveFirstElement())()  
 Removes and returns the first element from the list.

[**retrieveLastElement**](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378352_1/courses/23685.202120/BasicDoubleLinkedList.html#retrieveLastElement())()  
 Removes and returns the last element from the list.

[**toArrayList**](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378352_1/courses/23685.202120/BasicDoubleLinkedList.html#toArrayList())()  
 Returns an arraylist of the items in the list from head of list to tail of list

public class **SortedDoubleLinkedList<T>**

extends [BasicDoubleLinkedList](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378351_1/courses/23685.202120/BasicDoubleLinkedList.html)<T>

Implements a generic sorted double list using a provided Comparator. It extends BasicDoubleLinkedList class.

### Use the constructor SortedDoubleLinkedList()

public **SortedDoubleLinkedList**(Comparator<[T](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378351_1/courses/23685.202120/SortedDoubleLinkedList.html)> comparator2)

Creates an empty list that is associated with the specified comparator.

**Parameters:**

comparator2 - Comparator to compare data elements

Methods

Add

Inserts the specified element at the correct position in the sorted list

public [SortedDoubleLinkedList](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378351_1/courses/23685.202120/SortedDoubleLinkedList.html)<[T](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378351_1/courses/23685.202120/SortedDoubleLinkedList.html)> **add**([T](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378351_1/courses/23685.202120/SortedDoubleLinkedList.html) data)

Returns:

a reference to the current object

AddToEnd

public [BasicDoubleLinkedList](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378351_1/courses/23685.202120/BasicDoubleLinkedList.html)<[T](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378351_1/courses/23685.202120/SortedDoubleLinkedList.html)> **addToEnd**([T](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378351_1/courses/23685.202120/SortedDoubleLinkedList.html) data)

Overrides:

addToEnd in class BasicDoubleLinkedList<T>

AddToFront

public [BasicDoubleLinkedList](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378351_1/courses/23685.202120/BasicDoubleLinkedList.html)<[T](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378351_1/courses/23685.202120/SortedDoubleLinkedList.html)> **addToFront**([T](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378351_1/courses/23685.202120/SortedDoubleLinkedList.html) data)

Overrides:

addToFront in class BasicDoubleLinkedList<T>

Iterator

Implements the iterator by calling the super class iterator method.

public java.util.ListIterator<[T](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378351_1/courses/23685.202120/SortedDoubleLinkedList.html)> **iterator**()

Overrides:

iterator in class BasicDoubleLinkedList<T>

Remove

Implements the remove operation by calling the super class remove method.

public [SortedDoubleLinkedList](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378351_1/courses/23685.202120/SortedDoubleLinkedList.html)<[T](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378351_1/courses/23685.202120/SortedDoubleLinkedList.html)> **remove**([T](https://bb-montgomerycollege.blackboard.com/bbcswebdav/pid-6068684-dt-content-rid-83378351_1/courses/23685.202120/SortedDoubleLinkedList.html) data)

Overrides:

remove in class BasicDoubleLinkedList<T>

Create a JUnit Test – BasicDoubleLinkedList\_STUDENT\_Test.

And create a JUnit Test – SortedDoubleLinkedList\_STUDENT\_Test.

Test table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case # | Input | Actual  input | Expected  output | Actual  output | Did the test pass? |
| 1 | Tsega, Boogie, Abe, Berhie  Add to first  Mulu  (Basic) | Tsega, Boogie, Abe, Berhie, Mulu | Mulu,  Tsega, Boogie, Abe, Berhie | Mulu, Tsega, Boogie, Abe, Berhie | yes |
| 2 | Red, Yellow, Blue  Add to end  black  (Basic) | Red, Yellow, Blue, Black | Red, Yellow, Blue  Black | Red, Yellow, Blue, Black | yes |
| 3 | Tsega, Boogie, Abe, Berhie  Add  Mulu  (Sorted) | Tsega, Boogie, Abe, Berhie  Add  Mulu  (Sorted) | Abe  Berhie  Boogie  Mulu  Tsega | Abe  Berhie  Boogie  Mulu  Tsega | yes |
| 4 | Red, Yellow, Blue  (sorted) | Red, Yellow, Blue  (sorted) | Blue  Red  Yellow | Blue  Red  Yellow | yes |

Learning Experience

Assignment 3 was challenging, but a good practice on solving real life problems. Doing Assignment 3 helped me study all the materials we learned so far to work for my assignment. The algorithm to sort and display them as doubly liked list was a great lesson. It really helped me get back to my studies and look at ways to fix bugs and conjugate my knowledge to work on the assignment

I had some difficulties on getting the main point of the assignment and I finally find a way to do the assignments by using the UML given. After I created two of the classes which are BasicDoubleLinkedList and SortedDoublyLinkedList, I used the assignment sheet and the html to come up the codes.

All in all, the assignment was helpful to finalize what we learned so far. I wasn’t able to submit the assignment on Sunday, and I used my last chance to work on it and submit on Monday. I will study on the next materials on spring break and make my self ready for class.