List Implementation Assignment: Part #2

Re-submit Assignment

Due Apr 21, 2017 by 11:59pm **Points** 100 **Submitting** a file upload
File Types zip

Overview

This is a continuation of part #1 of our list implementation, which can be found here: **List Implementation Assignment: Part #1**. The second part of our assignment will have you implementing most of the remaining methods from the List<T> interface.

Some methods we will not implement, as can be seen from the example project I gave you. Ignore any methods that throw an UnsupportedOperationException. Also, you should note that part #2 will require you to build an iterator that works with your list class. Please review our class notes/slides on how to write an iterator.

Setup

Copy the changes from your list class from the part1 package to the part2 package. The part2 package has an updated ListTest.java file that you must use to test your work here.

Note: You must run the full set of unit tests on your new methods to pass this assignment.

Methods

public Iterator<T> iterator() Returns an iterator over the elements in this list. **Returns:** an iterator over the elements in this list in proper sequence
public void add(int index, T element)

Inserts the specified element at the specified position in this list. Shifts the element currently at that position (if any) and any subsequent elements to the right (adds one to their indices).

Parameters:

`index` - index at which the specified element is to be inserted

`element` - element to be inserted

Throws:

`IndexOutOfBoundsException (https://docs.oracle.com/javase/7/docs/api/java/lang/IndexOutOfBoundsException.html)` - if the index is out of range (index < 0 || index > size())

---

public T set(int index, T element)

Replaces the element at the specified position in this list with the specified element.

Parameters:

`index` - index of the element to replace

`element` - element to be stored at the specified position

Returns:

the element previously at the specified position

Throws:

`IndexOutOfBoundsException (https://docs.oracle.com/javase/7/docs/api/java/lang/IndexOutOfBoundsException.html)` - if the index is out of range (index < 0 || index >= size())

---

public int indexOf(Object element)

Returns the index of the first occurrence of the specified element in this list, or -1 if this list does not contain the element.

Parameters:

`element` - element to search for

Returns:

the index of the first occurrence of the specified element in this list, or -1 if this list does not contain the element

---

public T remove(int index)

Removes the element at the specified position in this list. Shifts any subsequent elements to the left (subtracts one from their indices). Returns the element that was removed from the list.

Parameters:

`index` - the index of the element to be removed

Returns:
the element previously at the specified position
Throws:
`IndexOutOfBoundsException` (https://docs.oracle.com/javase/7/docs/api/java/lang/IndexOutOfBoundsException.html) - if the index is out of range (index < 0 || index >= size())

---

public int lastIndexOf(Object element)

Returns the index of the last occurrence of the specified element in this list, or -1 if this list does not contain the element.

Parameters:
`element` - element to search for
Returns:
the index of the last occurrence of the specified element in this list, or -1 if this list does not contain the element

---

public boolean addAll(Collection<? extends T> other)

Appends all of the elements in the specified collection to the end of this list, in the order that they are returned by the specified collection's iterator.

Parameters:
`other` - collection containing elements to be added to this list
Returns:
true if this list changed as a result of the call

---

public boolean addAll(int index, Collection<? extends T> other)

Inserts all of the elements in the specified collection into this list at the specified position. Shifts the element currently at that position (if any) and any subsequent elements to the right (increases their indices). The new elements will appear in this list in the order that they are returned by the specified collection's iterator.

Parameters:
`index` - index at which to insert the first element from the specified collection
`other` - collection containing elements to be added to this list
Returns:
true if this list changed as a result of the call
Throws:
`IndexOutOfBoundsException` (https://docs.oracle.com/javase/7/docs/api/java/lang/IndexOutOfBoundsException.html) - if the index is out of range (index < 0 || index > size())

public boolean containsAll(Collection<?> other)

Returns true if this list contains all of the elements of the specified collection.

Parameters:
`other` - collection to be checked for containment in this list
Returns:
true if this list contains all of the elements of the specified collection

public boolean removeAll(Collection<?> other)

Removes from this list all of its elements that are contained in the specified collection.

Parameters:
`other` - collection containing elements to be removed from this list
Returns:
true if this list changed as a result of the call

public boolean retainAll(Collection<?> other)

Retains only the elements in this list that are contained in the specified collection. In other words, removes from this list all of its elements that are not contained in the specified collection.

Parameters:
`c` - collection containing elements to be retained in this list
Returns:
true if this list changed as a result of the call

Submission

- Each of your methods should pass the tests given in the sample project (i.e. they should be green when the tests are ran).
- Submit your entire project to the Canvas drop-box zipped.
- Remember to follow our style guide and include Javadocs for your list class!

List Implementation Part #1 Rubric (1)

Criteria	Ratings		Pts
List class contains a private inner class that uses the Iterator interface. The Iterator object returned from the iterator() method works correctly in a for-each loop.	15.0 pts Full Marks	0.0 pts No Marks	15.0 pts
add(index) inserts a new element at the given index (and for array lists it will shift elements elements to higher indices to make room). This method prevents duplicates in the collection.	10.0 pts Full Marks	0.0 pts No Marks	10.0 pts
set(index) replaces the element at the given index. This method prevents duplicates in the collection.	5.0 pts Full Marks	0.0 pts No Marks	5.0 pts
indexOf() returns the index of the element searched for, or -1 if not found.	5.0 pts Full Marks	0.0 pts No Marks	5.0 pts
remove(index) removes the element at the given index (and for array lists it will shift elements to lower indices fill the empty location in the array).	10.0 pts Full Marks	0.0 pts No Marks	10.0 pts
lastIndexOf() returns the index of the last occurrence of the element searched for, or -1 if not found.	5.0 pts Full Marks	0.0 pts No Marks	5.0 pts
Both addAll() methods add the collection to the existing collection. addAll(index) should place all elements at the given index (and for array lists this will shift elements to higher indices). These methods prevent duplicates in the collection.	10.0 pts Full Marks	0.0 pts No Marks	10.0 pts
containsAll() returns true if this list contains all elements in the given collection. removeAll() removes all elements from this list that are also contained in the given collection. retainAll() removes all elements from this list that are NOT contained in the given collection.	15.0 pts Full Marks	0.0 pts No Marks	15.0 pts
Styles: naming conventions, brace placement, spacing, commented code, redundancy, packages and magic numbers.	15.0 pts Full Marks	0.0 pts No Marks	15.0 pts
Formal documentation: full Javadocs & comment block header.	10.0 pts Full Marks	0.0 pts No Marks	10.0 pts
			Total Points: 100.0