ArrayList class listIterator method

This is 13th post in series of ArrayList class. Previously we have seen [ArrayList introduction](http://data-structure-learning.blogspot.com/2015/08/arraylist-class-introduction-and-how-it.html), ArrayList class [constructors](http://data-structure-learning.blogspot.com/2015/08/arraylist-class-constructors.html), [add](http://data-structure-learning.blogspot.com/2015/09/arraylist-class-add-methods.html)() method, [addAll](http://data-structure-learning.blogspot.com/2015/09/arraylist-class-addall-methods.html)() method, [clear](http://data-structure-learning.blogspot.com/2015/09/arraylist-class-clear-method.html)() method, [indexOf](http://data-structure-learning.blogspot.com/2015/09/arraylist-class-indexof-method.html)() method, [contains](http://data-structure-learning.blogspot.com/2015/09/arraylist-class-contains-method.html)() method, [forEach](http://data-structure-learning.blogspot.com/2015/09/arraylist-class-foreach-method.html)() method, [get](http://data-structure-learning.blogspot.com/2015/09/arraylist-class-get-method.html)(), [isEmpty](http://data-structure-learning.blogspot.com/2015/09/arraylist-class-isempty-method.html)(), [iterator](http://data-structure-learning.blogspot.com/2015/09/arraylist-class-iterator-method.html)() and [lastIndexOf](http://data-structure-learning.blogspot.com/2015/09/arraylist-class-lastindexof-method.html)() method.

listIterator() method is enhanced version of iterator().

Below are the methods that can be used while traversing collection by listIterator.

|  |  |
| --- | --- |
| **Method** | **Description** |
| **boolean** hasNext(); | Returns true if there are more elements while list is being traversed in forward direction. |
| **boolean** hasPrevious(); | Returns true if there are more elements when list is being traversed backwards. |
| E previous(); | Returns the previous element in iteration while being traversed in backward direction. It will throw NoSuchElementException if there are no more elements. |
| E next(); | Returns the next element in iteration while being traversed in forward direction. It will throw NoSuchElementException if there are no more elements. |
| **int** nextIndex(); | Returns the index of element that will be returned by next() |
| **int** previousIndex(); | Returns the index of element that will be returned by previous() |
| **void** remove(); | removes the element from the list which was returned by next() or previous() |
| **void** set(E e); | Replaces the last element that was returned by next() or previous(). |
| **void** add(E e); | Inserts element e in the list |

Previously, I had written a post on [ListIterator interface](http://data-structure-learning.blogspot.com/2015/05/java-collections-part-9-listiterator.html). [Continue](http://data-structure-learning.blogspot.com/2015/05/java-collections-part-9-listiterator.html) to that post to read further on listIterator() method.

In next post we will the remove() method. remove() method is overloaded method which takes either index or Object itself.