ArrayList class removeAll method

This is 16th 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() method, get(), isEmpty(), iterator(), lastIndexOf() method, listIterator(), remove(int index), remove(Object o) method.

In this post we will see removeAll(Collection<?> c) method. removeAll(Collection<?> c) method is used to remove all the elements from this list that are contained in specified Collection(as parameter).

I have written a small program to demonstrate the usage of removeAll(Collection<?> c) method.

**package** org.example.collections.list.arraylist;

**import** java.util.ArrayList;

**import** java.util.List;

**public** **class** ArrayListRemoveAll {

/\*\*

\* List<String> that contains names

\* \*/

**public** List<String> names() {

List<String> names = **new** ArrayList<String>();

names.add("Ned");

names.add("Catelyn");

names.add("Rob");

names.add("Arya");

names.add("Sansa");

**return** names;

}

/\*\*

\* List<String> contains those names that are

\* to be removed.

\* \*/

**public** List<String> namesToBeRemoved() {

List<String> names = **new** ArrayList<String>();

names.add("Ned");

names.add("Rob");

names.add("Arya");

**return** names;

}

/\*\*

\* names List will remove elements that are contained in

\* namesToBeRemoved List.

\* \*/

**public** **void** removeAllDemo(List<String> names,

List<String> namesToBeRemoved) {

names.removeAll(namesToBeRemoved);

System.***out***.print("After names.removeAll(namesToBeRemoved) call: ");

System.***out***.println(names);

}

**public** **static** **void** main(String[] args) {

ArrayListRemoveAll removeAll = **new** ArrayListRemoveAll();

List<String> names = removeAll.names();

System.***out***.println("Names: " + names);

List<String> namesToBeRemoved = removeAll.namesToBeRemoved();

System.***out***.println("Names to be removed: " + namesToBeRemoved);

removeAll.removeAllDemo(names, namesToBeRemoved);

}

}

Output

Names: [Ned, Catelyn, Rob, Arya, Sansa]

Names to be removed: [Ned, Rob, Arya]

After names.removeAll(namesToBeRemoved) call: [Catelyn, Sansa]

See in output we can see that Ned, Rob, Arya are removed this list. Now list only contains Catelyn, Sansa.

That’s all for removeAll(Collection<?> c) method.

In next post we will removeIf(Predicate<? Super E> filter) method. This method is part of Java 8 and uses [Predicate](http://data-structure-learning.blogspot.com/p/1.html) Interface to filter the elements from the List.