ArrayList class retainAll method

This is 19th 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), removeAll(Collection<?> c), removeIf(Predicate<? super T> E) and replaceAll(UnaryOperator<E> operator) method.

In this post we will see retainAll(Collection<?> c) method. retainAll(Collection<?> c) is opposite of removeAll(Collection<?> c). In removeAll method we removed all the elements from this Collection that are specified in parameter Collection. In retainAll method **we keep only those elements** in this Collection that are in specified Collection (as parameter).

Let us take for example two different lists.

Kids name(kidsName): [Robb, Sansa, John, Arya, Bran]

Family Members name(familyName): [Robb, Catelyn, Ned, Lyanna, Sansa, John, Arya, Bran]

Now let us run retainAll method.

familyName.retainAll(kidsName) : This mean that retainAll kids names from family name. Now family name will consist of following:

[Robb, Sansa, John, Arya, Bran]

Below is the program that uses retainAll operation.

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

**import** java.util.ArrayList;

**import** java.util.List;

**public** **class** ArrayListRetainAllDemo {

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

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

kids.add("Robb");

kids.add("Sansa");

kids.add("John");

kids.add("Arya");

kids.add("Bran");

**return** kids;

}

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

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

family.add("Robb");

family.add("Catelyn");

family.add("Ned");

family.add("Lyanna");

family.add("Sansa");

family.add("John");

family.add("Arya");

family.add("Bran");

**return** family;

}

**public** **void** retainAllDemo(List<String> familyNames, List<String> kidsNames){

System.***out***.println("Family: "+familyNames);

System.***out***.println("Kids: "+kidsNames);

familyNames.retainAll(kidsNames);

System.***out***.println("Family after retainAll: "+familyNames);

}

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

ArrayListRetainAllDemo retainAllDemo = **new** ArrayListRetainAllDemo();

List<String> kidsNames = retainAllDemo.kidsNames();

List<String> familyNames = retainAllDemo.familyNames();

retainAllDemo.retainAllDemo(familyNames, kidsNames);

}

}

Output

Family: [Robb, Catelyn, Ned, Lyanna, Sansa, John, Arya, Bran]

Kids: [Robb, Sansa, John, Arya, Bran]

Family after retainAll: [Robb, Sansa, John, Arya, Bran]

That’s all on retainAll(Collection<?> c) method. In next post we will see set(int index, E element) method which is used to replace the element at given index.