Collections class copy(..) method

copy(..) is a static method in Collections class. It is used to copy the elements of source list to destination list.

We have covered [addAll(..)](http://data-structure-learning.blogspot.com/2015/05/collections-class-addall-method.html), [swap(..)](http://data-structure-learning.blogspot.com/2015/06/collections-class-swap-method.html), [binarySearch(..)](http://data-structure-learning.blogspot.com/2015/06/collections-class-binarysearch-method.html) and [synchronized](http://data-structure-learning.blogspot.com/2015/06/synchronized-collections-by-collections.html) in previous posts.

Once this operation is completed he index of elements copied are same in source and destination list.

The destination list must be at least as long as source list. If the destination list is larger than additional elements are not affected.

Now regarding exceptions thrown by this method.

IndexOutOfBoundsException: This exception is thrown if the source list size is greater than destination list as source lists elements won’t fit in destination list.

UnsupportedOperationException: This exception is thrown if the destination lists list-iterator does not support set operations.

First we will take example that will run properly.

Then we will take example that will throw IndexOutOfBoundsException.

Last, we will take example that will throw UnsupportedOperationException.

Below example works fine.

**public** **static** **void** copydemo(){

List<String> languages = *populateLanguages*();

List<String> databases = *populateDatabases*();

System.***out***.println("Languages list "+languages);

System.***out***.println("Databases list "+databases);

Collections.*copy*(databases, languages);

System.***out***.println("After Copy Operation");

System.***out***.println("Databases list "+databases);

}

Output:

Languages list [Java, JavaScript, C#, Python]

Databases list [MySql, SqlLite, MongoDB, Neo4j, Oracle]

After Copy Operation

Databases list [Java, JavaScript, C#, Python, Oracle]

Observe the output carefully. Languages list has size 4 and Database list has size 5.   
 Now Database list’s first 4 elements are same as Languages and last element is not affected.

Now we will take an example that will throw IndexOutOfBoundsException.

**public** **static** **void** copydemo(){

List<String> languages = *populateLanguages*();

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

System.***out***.println("Languages list "+languages);

System.***out***.println("Databases list "+databases);

Collections.*copy*(databases, languages);

System.***out***.println("After Copy Operation");

System.***out***.println("Databases list "+databases);

}

Output:

Languages list [Java, JavaScript, C#, Python]

Databases list []

Exception in thread "main" java.lang.IndexOutOfBoundsException: Source does not fit in dest

at java.util.Collections.copy(Unknown Source)

at org.collections.CollectionClass.copydemo(CollectionClass.java:74)

at org.collections.CollectionClass.collectionsAddAllDemo(CollectionClass.java:30)

at org.collections.CollectionClass.main(CollectionClass.java:12)

We have languages list with size 4 and databases list with size 0. So this will throw IndexOutOfBoundsException because rules apply that destination list must be atleast as long as source list.

Now we will take an example that will throw UnsupportedOperationException.

**public** **static** **void** copydemo() {

List<String> languages = *populateLanguages*();

List<String> databases = *populateDatabases*();

System.***out***.println("Languages list " + languages);

System.***out***.println("Databases list " + databases);

databases = Collections.*unmodifiableList*(databases);

Collections.*copy*(databases, languages);

System.***out***.println("After Copy Operation");

System.***out***.println("Databases list " + databases);

}

Output:

Languages list [Java, JavaScript, C#, Python]

Databases list [MySql, SqlLite, MongoDB, Neo4j, Oracle]

Exception in thread "main" java.lang.UnsupportedOperationException

at java.util.Collections$UnmodifiableList.set(Unknown Source)

at java.util.Collections.copy(Unknown Source)

at org.collections.CollectionClass.copydemo(CollectionClass.java:75)

at org.collections.CollectionClass.collectionsAddAllDemo(CollectionClass.java:30)

at org.collections.CollectionClass.main(CollectionClass.java:12)

What we did here was we created an unmodifiable list using the static method unmodifiableList(list). Then we are trying to change it using copy method. This will result in UnsupportedOperationException.

In next posts we will saw other methods for Collections class.