Difference between Comparable<> and Comparator<>

We have seen [Comparable<>](http://data-structure-learning.blogspot.com/2015/06/comparable-interface-for-sorting.html) and [Comparator<>](http://data-structure-learning.blogspot.com/2015/06/comparator-interface.html) in previous post. In this post we will see difference between both of this interface. This is one of the most important [Collections interview question](http://data-structure-learning.blogspot.com/p/java-collections_26.html) in Interview.

Previously, I had written posts on difference between [fail-safe vs fail-fast iterator](http://data-structure-learning.blogspot.com/2015/05/difference-between-fail-safe-iterator.html), [Collection vs Collections](http://data-structure-learning.blogspot.com/2015/05/difference-between-collection-and.html), [HashMap vs Hashtable](http://data-structure-learning.blogspot.com/2015/05/difference-between-hashtableand-hashmap.html), [ArrayList vs LinkedList](http://data-structure-learning.blogspot.com/2015/05/difference-between-arraylist-and_28.html), [ArrayList vs Vector](http://data-structure-learning.blogspot.com/2015/05/difference-between-arraylist-and-vector.html), [List vs Set](http://data-structure-learning.blogspot.com/2015/05/difference-between-list-and-set.html), [poll() vs remove() method in Queue](http://data-structure-learning.blogspot.com/2015/05/difference-between-poll-and-remove-in.html) and [Iterator vs ListIterator](http://data-structure-learning.blogspot.com/2015/05/5-difference-between-iterator-and.html).

Comparable<> and Comparator<> are two tools provided to use for sorting of instances in Collection.

We will now see the differences between these two interfaces.

Sort Sequence: In Comparable interface we can implement only one sort sequence. If other sort sequence is to be implemented then we need to change the class. In Comparator interface we can implement any number of sorting sequence.

Objects for Comparison: Comparable compares “this” reference with the parameter in compareTo(T o) method. So one object is provided that is compared to “this” object. Comparable interface takes 2 object of type T in compare(T o1, To2) method.

Methods Used: Comparable interface uses compareTo(T o) method while Comparator interface uses compare(T o1, T o2) method. Both of the methods returns same details.

For **Comparable** interface

|  |  |
| --- | --- |
| **Returns** | **Condition** |
| Negative | thisObject < anotherObject |
| Zero | thisObject == anotherObject |
| Positive | thisObject > anotherObject |
|  |  |

For **Comparator** interface

|  |  |
| --- | --- |
| **Returns** | **Condition** |
| Negative | object1 < object2 |
| Zero | object1 == object2 |
| Positive | object1 > object2 |
|  |  |

Package: Comparable interface is in java.lang package because it should be used as default sorting strategy. Comparator interface is in java.util package because it should be used as utility for sorting.

Modification of Classes: If using Comparable interface we can implement only one sort sequence on the class. Also we use “this” reference. Now we want to implement another sort sequence then we need to modify the class. Now, if we use Comparator interface then we can implement different sort sequence on different attributes of same instance. We saw that in Comparator interface post.

In next post we will see some points to remember while using Comparator and Comparable.