

Java Comparable vs. Comparator



Quang Nguyen

Sep 3, 2018 · 1 min read

When I first learned Java, I was confused about these two API because they looked and functioned as if they were the same! But there is an important distinction between them and if you want to be a Java developer, you should know what it is.

Comparable:

- provides *one* way to sort (emphasis on the *one*)
- to use, needs to `import java.lang.Comparable`
- in order to make use of Comparable, a class needs to implement `Comparable<T>` and override:

```
public int compareTo(T t) {  
    ...  
    // returns negative integer, zero, or positive integer  
    // for less than, equal, or greater than  
}
```

- when you invoke the sort method on an object, it will in turn call your `compareTo` method to do comparison and sort for you, therefore, it only provides one specific way to sort your object.

Comparator:

- provides multiple ways to sort, as long as you implement the methods
- to use, needs to `import java.util.Comparator`

- no need to implement any interface, instead just create a new `Comparator` and override, for example:

```
public static Comparator<T> customComparator = new Comparator<T>() {  
    @Override  
    public int compare(T t1, T t2) {  
        ...  
        // returns negative integer, zero, or positive integer  
        // for less than, equal, or greater than  
    }  
}
```

and substitute `T` with your custom object or class. To use it, you would do something like `Arrays.sort(YourObj, YourObj.customComparator)` .

You can create many custom comparator within your class with `Comparator` as opposed to `Comparable` which only lets you compare one way.