**Comparator Interface ok**

**What is Comparator Interface ? why we need the comparator Interface ?**

**Comparable Interface –**

* It is inside java.lang package.
* It has only one method **called CompareTo()**
* **This compareTo() is actually returning an integer and that integer basically should return as a positive , negative, or a number 0 which is basically help us to sort the thing.**

**So if you use the Collections class sort() it only accept the object which is comparable. So basically by using this particular method [compareTo()] and whatever this particular method returns using that the sort() works with his logic and help us to sort all our number, String or whatever we provide to it.**

**Comparable : behind the scene –[debugging] Collection.sort().**

**Let’s say I want to sort everything in ascending order and any point of time you want to see what is happing internally.**

**We can do ---**

**We can come to the MP4 java class-**

**Like**

* **I have only two song in my list—and after that**
* **I am trying to sort this particular Collections class**
* **Then Trying to iterate it over my list**

package example.Soring;

import java.util.ArrayList;

import java.util.Collections;

import APISong.Song;

import sorting.seleniumexpress.MyUtil;

public class MP4 {

public static void main(String[] args) {

//two songs

Song song1 = new Song("take me to your heart","Michal Leans to Back",2014);

Song song2 = new Song("as long as you live mew", "Nelly", 2010);

ArrayList <Song> musicList = new ArrayList <> ();

musicList.add(song1);

musicList.add(song2);

MyUtil.iterateList(musicList); // Before sorted

**Collections.sort(musicList); // sort**

System.out.println("After Sorting---- ");

MyUtil.iterateList(musicList); //After sorted

}}

**So if you want to see what is actually happening internally** -----

always in the sort () we can break the program by debugging it and you can go into the Song class here and the song class has a compareTo().

I can put a debug point here---as well

* **return** **this**.getTitle().compareTo(anotherSong.getTitle()); // Ascending order

and After that we can debug this particular program and I can see this compare To method is returning a positive number , Negative number and how it is actually returned –

How can I degug –So I can go to my MP4 class > right click > click on Debug As > to debug in java Application.

**Distinction between Comparable and Comparator Interface –**

Like you are kind of building MP4 application and it’s kind of a music app and you are trying to build it out. So Whenever you are trying to build out a music app , you are not going to build everything from scratch rather you are going to look in the market that who has a music API. So that you can take that API, you can take that Jar. You can plug and play that particular are in your application.

And you can have a already a lot of feature which is already there in the particular library. You don’t need to develop everything from scratch.