**Q- What is sorting in Java ?**

It probably means collection class sort method or the Comparable interface or the Comparator Interface.

//IntegerList

package sorting.seleniumexpress;

import java.util.ArrayList;

import java.util.Collections;

public class IntegerSortingDemo {

public static void main(String[] args) {

ArrayList <Integer> integerList = new ArrayList<Integer> ();

// here Unsorted Integer List

integerList.add(15);

integerList.add(3);

integerList.add(7);

integerList.add(1);

integerList.add(9);

integerList.add(10);

//System.out.println(integerList);

for (Integer temp : integerList) { // previously our no. was just like this 15,3,7,1,9,10

System.err.println(temp);

}

Collections.sort(integerList); // collections get sorted .

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

for (Integer temp : integerList) { //After sorting it become just like this output.

System.err.println(temp);

}}}

/\*

Steps --

1) Main method

2) Create a ArrayList

3) I am going to take as a Integer [Integer is a Wrapper Class present inside the java.lang.package]

4) Create ArrayList of Integer object

5) Insert Integer to my List called Integer List .

6) Put some Random Numbers to my list which will not be sorted and after that

I will just to try sorting them using the Collection class sort method

7) Just print out this number using for each loop . this is exactly the order that we have bee sorted.

we know that In the ArrayList whatever we insert the order is always is going to be preserved.

8) Normally we sort as a Ascending order that is called a Natural sorting order . it means Ascending Order.

not in a reverse 10,9,8,....Natural sorting order is Ascending Order.

\*\*\*\*\*

9) So I just want to sort this particular List.

So to sort it I will call a method called sort method in the collections class. and I can pass on my list and

My list is in IntegerList. [ArrayList name ]

And then My Collections get sorted and once the collection get sorted then again I can Iterate through

that particular collection. and I can see whether my collection has been done perfectly or not.

So I just copied the for each loop again from the above which I did before,

10) There are basically two sort methods ---

\*\*\*\*

11) Try to sort some string and after that we will create some custom object and try to solve them out.

12) So again I can create a class called StringSortingDemo. and create a String ArrayList again.

\*/

**// Integer**

**package** sorting.seleniumexpress;

**import** java.util.ArrayList;

**import** java.util.Collections;

**public** **class** CopyInteShortRemoveDuplicate {

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

ArrayList <Integer> integerList = **new** ArrayList<Integer> ();

// here Unsorted Integer List

integerList.add(15);

integerList.add(3);

integerList.add(7);

integerList.add(1);

integerList.add(9);

integerList.add(10);

//System.out.println(integerList);

MyUtil.*iterateIntegerList*(integerList);

Collections.*sort*(integerList); // collections get sorted .

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

MyUtil.*iterateIntegerList*(integerList);

}

**private** **static** **void** iterateIntegerList(ArrayList<Integer> integerList) {

**for** (Integer temp : integerList) { // previously our no. was just like this 15,3,7,1,9,10

System.***err***.println(temp);

}}}

/\*Note -- Utility class is a static in nature . we can use the class name to access our methods. \*

\*

1) Now I can remove this particular method from my Integersortingdemo class. and then

I will add "MyUtil" the class name.

Delete Common method and add the name of MyUtil class

2) At the time of adding an Error Message is coming. To remove this error I have to make public from Private.

3) Obviously the Utility methods basically most of the time they are static in nature.

The same thing we can do in case of String ---

====================

So What we have done ? We have .

1) Refactored our classes.

2) creates my owen Utility class named MyUtil.

3) Wherever I have a need to iterate over a list of Integer or a List of string I can actually use this method

and this method based on the type that I am passing right here.

So 4 ) For an Example In my sorting a String demo I am using the iterateStringList [MyUtil.iterateStringList(musicList1);]

method, if I do Command click . So I am using this particular method -

public static void iterateStringList(ArrayList<String> musicList1) {

for (String strList : musicList1) {

System.out.println(strList);

}

}

So I am using this particular method and Similarly If I will go to my IntegerSortingDemo ,

here I am trying to

// Two Common methods --- One for Integer and the other for String.

public static void iterateIntegerList(ArrayList<Integer> integerList) {

for (Integer temp : integerList) { // previously our no. was just like this 15,3,7,1,9,10

System.out.println(temp);

}

}

public static void iterateStringList(ArrayList<String> musicList1) {

for (String strList : musicList1) {

System.out.println(strList);

}

**Q - Can I simplify these two things further / Can I make it better/ Can I do more Refactoring over here?**

Ans - Obviously, yes.

I am basically iterating over a List. The List can be Integer and the list can be String.

So Why don't I create just one method instead of two.

Lest's say Tomorrow you want to Iterate over a list of Double or you want to iterate over a list

of Float or something else. So do you actually need to create another method here inside your

MyUtil class. not exactly . because /that's we are leaning Generic.

So basically do over here Instead tightly binding all these things to the ArrayLsit, I can remove

one of the method over here and I will just convert these method to accept or to handle any kind of list

, it may be a list of string or it may be a list of integer or character, or float or whatever.

so to make a method Generic --

a) the first thing is that I need to compiler hey and I need to java that hey this is Generic method.

b) To do that I can give a diamond bracket here and I can make it -T. but I can write anything write here.

but for making a more readable I can put here T. T means types. and then in other diamond bracket ,

this could be anything.but I can put here T also.

\*/

**StringList**

package sorting.seleniumexpress;

import java.util.ArrayList;

import java.util.Collections;

public class StringSoringDemo {

public static void main(String[] args) {

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

// this is not sorted

musicList.add("zara zara");

musicList.add("besabriyaan");

musicList.add("kaise yua");

musicList.add("aeisa desh hai mera");

musicList.add("koi lauta de woh pyare din");

for (String strList : musicList) {

System.out.println(strList);

}

System.out.println();

//---------------------

Collections.sort(musicList);

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

for (String strList : musicList) {

System.out.println(strList);

}}}

/\*

13 ) In musicList We want to add some song . so i have added some song to my music list

and then iterate over this particular things. and print it to my console.

14) If I sort all these music list perfectly , First I will get "aeisa desh hai mera" then

"Basabriyaan" then whatever---

//========================================================

//String

**package** sorting.seleniumexpress;

**import** java.util.ArrayList;

**import** java.util.Collections;

**public** **class** CopyRemovingDuplicateCodeLinesStringSortingDemo {

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

ArrayList <String> musicList = **new** ArrayList<String>();

// this is not sorted

musicList.add("zara zara");

musicList.add("besabriyaan");

musicList.add("kaise yua");

musicList.add("aeisa desh hai mera");

musicList.add("koi lauta de woh pyare din");

MyUtil.*iterateStringList*(musicList);

System.***out***.println();

//---------------------

Collections.*sort*(musicList);

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

MyUtil.*iterateStringList*(musicList); }}

/\*

About Duplicate Code ----

=====================

1) Here I am trying to Iterate ArrayList before and after sorting ,I used for each loop two times

that is called Duplicate Coding. these lines of code actually red

2) In that case we can Extract these much of code to a different method and we just can call that method.

we should not write the same code again and again everywhere.

3) so Rules of avoiding the duplicate coding --

i) Select those lines of code that you want to Extract to a method or you want to Construct a method

these lines of code then click on Ctrl +1 and then Click Extract to Method. then we will see java has created a method

for us . and the method name is |extracted| method.and then we can change the method name here.

So to give a Valid name like |iterateList| and then - Enter. and then duplicate lines code has gone.

ii) then we are just the iterateList method and we are passing our list. so my list name is musicList.

iii) Java collections.sort() | #2

===================

\*\*\*\* Removing Common Method and adding MyUtil class in which

\*

#######

Problem of Extract or Removing Double coding -------

1) Like Tomorrow I am creating a class and I have a requirement to iterate over a list

of String . At that time again I have to write some methods just like this --

in that particular class.

private static void iterateStringList(ArrayList<String> musicList1) {

for (String strList : musicList1) {

System.out.println(strList);

}

}

So what about if I am going to move these particular methods, this are my utility method.

What about if I will move this particular method to a common class and whenever I need this

I can call the method from that particular class and can use it.

So What I basically mean here is ----

a) create a new package here - myutility. and class name - MyUtil

\*/

About Duplicate Code ----

=====================

1) Here I am trying to Iterate ArrayList before and after sorting ,I used for each loop two times

that is called Duplicate Coding. these lines of code actually red

2) In that case we can Extract these much of code to a different method and we just can call that method.

we should not write the same code again and again everywhere.

3) so Rules of avoiding the duplicate coding --

i) Select those lines of code that you want to Extract to a method or you want to Construct a method

these lines of code then click on Ctrl +1 and then Click Extract to Method. then we will see java has created a method

for us . and the method name is |extracted| method.and then we can change the method name here.

So to give a Valid name like |iterateList| and then - Enter. and then duplicate lines code has gone.

ii) then we are just the iterateList method and we are passing our list. so my list name is musicList.

iii) Java collections.sort() | #2

===================

\*\*\*\* Removing Common Method and adding MyUtil class in which

\*

#######

Problem of Extract or Removing Double coding -------

1) Like Tomorrow I am creating a class and I have a requirement to iterate over a list

of String . At that time again I have to write some methods just like this --

in that particular class.

private static void iterateStringList(ArrayList<String> musicList1) {

for (String strList : musicList1) {

System.out.println(strList);

}

}

So what about if I am going to move these particular methods, this are my utility method.

What about if I will move this particular method to a common class and whenever I need this

I can call the method from that particular class and can use it.

So What I basically mean here is ----

a) create a new package here - myutility. and class name - MyUtil

/

About Duplicate Code ----

=====================

1) Here I am trying to Iterate ArrayList before and after sorting ,I used for each loop two times that is called Duplicate Coding. these lines of code actually red

2) In that case we can Extract these much of code to a different method and we just can call that method.we should not write the same code again and again everywhere.

3) so Rules of avoiding the duplicate coding --

i) Select those lines of code that you want to Extract to a method or you want to Construct a method

these lines of code then click on Ctrl +1 and then Click Extract to Method. then we will see java has created a method

for us . and the method name is |extracted| method.and then we can change the method name here.

So to give a Valid name like |iterateList| and then - Enter. and then duplicate lines code has gone.

ii) then we are just the iterateList method and we are passing our list. so my list name is musicList.

iii) Java collections.sort() | #2

\*/

/\*Note -- Utility class is a static in nature . we can use the class name to access our methods. \*

\*

\*

1) Now I can remove this particular method from my Integersortingdemo class. and then

I will add "MyUtil" the class name.

Delete Common method and add the name of MyUtil class

2) At the time of adding an Error Message is coming. To remove this error I have to make public from Private.

3) Obviously the Utility methods basically most of the time they are static in nature.

The same thing we can do in case of String ---

====================

So What we have done ? We have .

1) Refactored our classes.

2) creates my owen Utility class named MyUtil.

3) Wherever I have a need to iterate over a list of Integer or a List of string I can actually use this method

and this method based on the type that I am passing right here.

So 4 ) For an Example In my sorting a String demo I am using the iterateStringList [MyUtil.iterateStringList(musicList1);]

method, if I do Command click . So I am using this particular method -

public static void iterateStringList(ArrayList<String> musicList1) {

for (String strList : musicList1) {

System.out.println(strList);

}

}

So I am using this particular method and Similarly If I will go to my IntegerSortingDemo ,

here I am trying to

// Two Common methods --- One for Integer and the other for String.

public static void iterateIntegerList(ArrayList<Integer> integerList) {

for (Integer temp : integerList) { // previously our no. was just like this 15,3,7,1,9,10

System.out.println(temp);}}

public static void iterateStringList(ArrayList<String> musicList1) {

for (String strList : musicList1) {

System.out.println(strList);

}

**Q - Can I simplify these two things further / Can I make it better/ Can I do more Refactoring over here?**

Ans - Obviously, yes.

I am basically iterating over a List. The List can be Integer and the list can be String.

So Why don't I create just one method instead of two.

Lest's say Tomorrow you want to Iterate over a list of Double or you want to iterate over a list

of Float or something else. So do you actually need to create another method here inside your

MyUtil class. not exactly . because /that's we are leaning Generic.

So basically do over here Instead tightly binding all these things to the ArrayLsit, I can remove

one of the method over here and I will just convert these method to accept or to handle any kind of list

, it may be a list of string or it may be a list of integer or character, or float or whatever.

so to make a method Generic --

=========

a) the first thing is that I need to compiler hey and I need to java that hey this is Generic method.

b) To do that I can give a diamond bracket here and I can make it -T. but I can write anything write here.

but for making a more readable I can put here T. T means types. and then in other diamond bracket ,

this could be anything.but I can put here T also.

So It can be Access a list of any type. and I don't want make it tightly couple ,

I will just write it T here. So Done

\*/

**package** sorting.seleniumexpress;

**import** java.util.ArrayList;

**public** **class** MyUtil {

public static void iterateIntegerList(ArrayList<Integer> integerList) {

for (Integer temp : integerList) { // previously our no. was just like this 15,3,7,1,9,10

System.err.println(temp);

}

}

public static void iterateStringList(ArrayList<String> musicList) {

for (String strList : musicList) {

System.out.println(strList);

} // Before doing in Two method

===========================

**public** **static** <T> **void** iterateList(ArrayList<T> integerList) {

**for** (T temp : integerList) { // previously our no. was just like this 15,3,7,1,9,10

System.***err***.println(temp);

}

}// After Deleting one method and Add The Generic. And only one Method is applied for both type.

Now This Particular method can able to handle any kind of List and can iterate over it .

Now If I am going to pass an ArrayList of String here then T becomes String .and If I am going to pass the List of Integer and then This T becomes Integer.

So Similarly if it is float this will be flaot.

This method has no return type and the return type is void.

So Whenever I use Generic I have to Delete all

MyUtil.iterateList(integerList);

Collections.sort(integerList); // collections get sorted .

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

MyUtil.iterateList(integerList);

} }

So I have Generic method and this generic method name is iterateList and this

iterateList can take any kind of List may be a List of String , may be a List of Integer or it may be a Float or whatever

**public** **static** <T> **void** iterateList(ArrayList<T> anyList) {

**for** (T temp : anyList) { // previously our no. was just like this 15,3,7,1,9,10

System.***err***.println(temp);

}

About Duplicate Code ----

=====================

1) Here I am trying to Iterate ArrayList before and after sorting ,I used for each loop two times

that is called Duplicate Coding. these lines of code actually red

2) In that case we can Extract these much of code to a different method and we just can call that method.

we should not write the same code again and again everywhere.

3) so Rules of avoiding the duplicate coding --

i) Select those lines of code that you want to Extract to a method or you want to Construct a method

these lines of code then click on Ctrl +1 and then Click Extract to Method. then we will see java has created a method

for us . and the method name is |extracted| method.and then we can change the method name here.

So to give a Valid name like |iterateList| and then - Enter. and then duplicate lines code has gone.

ii) then we are just the iterateList method and we are passing our list. so my list name is musicList.

iii) Java collections.sort() | #2

/ Problem of Extract or Removing Double coding -------

1) Like Tomorrow I am creating a class and I have a requirement to iterate over a list

of String . At that time again I have to write some methods just like this --

in that particular class.

private static void iterateStringList(ArrayList<String> musicList1) {

for (String strList : musicList1) {

System.out.println(strList);

}}}

So what about if I am going to move these particular methods, this are my utility method.

What about if I will move this particular method to a common class and whenever I need this

I can call the method from that particular class and can use it.

So What I basically mean here is ----

a) create a new package here - myutility. and class name - MyUtil

\*/

**package** sorting.seleniumexpress;

**import** java.util.ArrayList;

**import** java.util.Collections;

**public** **class** StringSortRemoveDuplicateCode {

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

ArrayList <String> musicList = **new** ArrayList<String>();

// this is not sorted

musicList.add("zara zara");

musicList.add("besabriyaan");

musicList.add("kaise yua");

musicList.add("aeisa desh hai mera");

musicList.add("koi lauta de woh pyare din");

MyUtil.*iterateList*(musicList);

System.***out***.println();

//---------------------

Collections.*sort*(musicList);

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

MyUtil.*iterateList*(musicList);}}

}

package sorting.seleniumexpress;

import java.util.ArrayList;

import java.util.Collections;

public class IntegerSortRemoveDuplicateCode {

public static void main(String[] args) {

ArrayList <Integer> integerList = new ArrayList<Integer> ();

// here Unsorted Integer List

integerList.add(15);

integerList.add(3);

integerList.add(7);

integerList.add(1);

integerList.add(9);

integerList.add(10);

/System.out.println(integerList);

MyUtil.iterateList(integerList); //extracted(integerList); -this is not valid name.

Collections.sort(integerList); // collections get sorted .

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

MyUtil.iterateList(integerList);}}

**package** sorting.seleniumexpress;

**import** java.util.ArrayList;

**public** **class** MyUtil {

/\*public static void iterateIntegerList(ArrayList<Integer> integerList) {

for (Integer temp : integerList) { // previously our no. was just like this 15,3,7,1,9,10

System.err.println(temp);

}}

public static void iterateStringList(ArrayList<String> musicList) {

for (String strList : musicList) {

System.out.println(strList);

}\*/

**public** **static** <T> **void** iterateList(ArrayList<T> anyList) {

**for** (T temp : anyList) {

System.***err***.println(temp);

}}}