***Collections.sort() in Java***

The java.util.Collections.sort() method is present in java.util.Collections class. It is used to sort the elements present in the specified list of Collection in ascending order. It works similar to java.util.Arrays.sort() method but it is better then as it can sort the elements of Array as well as a linked list, AbstractList queue and much more present implementing List interface.  
  
**Example 1:** Working of Collection.sort() and how to reverse sort.  
java

// Java program to demonstrate

// working of Collections.sort()

import java.util.\*;

public class Collectionsorting

{

public static void main(String[] args)

{

// Create a list of strings

List<Integer> list = new ArrayList<Integer>();

list.add(10);

list.add(5);

list.add(20);

// List is sorted in the natural order

Collections.sort(list);

// Displaying the sorted list

System.out.println(list);

// Sorting the list in reverse order

Collections.sort(list, Collections.reverseOrder());

// Let us print the sorted list

System.out.println(list);

}

}

**Output:**

[5, 10, 20]

[20, 10, 5]

**Note:** The wrapper classes like Integer, Character etc implement Comparable interface. Here Collections.reverse() uses a compareTo() function to return a comparator that helps to reverse sort the list.  
  
**Implementing Comparable interface to sort a Collection**  
**Example 2:** Using a user-defined class implementing Comparable interface, where the compareTo() function uses the x-coordinate to sort the passed list of points.  
java

// Java program to demonstrate

// working of Collections.sort()

import java.util.\*;

import java.lang.\*;

import java.io.\*;

// A user-defined Point class implementing

// Comparable interface

class Point implements Comparable<Point>

{

int x, y;

Point(int x, int y)

{

this.x = x;

this.y = y;

}

// compareTo() function defining the

// nature of sorting i.e., according to

// x-coordinate

public int compareTo(Point P)

{

return this.x - P.x;

}

}

class GfG

{

public static void main(String[] args)

{

// Create a list of Integers

List<Point> list = new ArrayList<Point>();

list.add(new Point(5, 10));

list.add(new Point(2, 20));

list.add(new Point(10, 30));

// List is sorted in the natural order

Collections.sort(list);

// Displaying the points

for (Point p: list)

System.out.println(p.x + " " + p.y);

}

}

**Output:**

2 20

5 10

10 30

**Implementing Comparator interface to sort a Collection**  
**Example 3:** Using a class implementing Comparator interface, to sort the passed list of points according to x-coordinates.  
java

// Java program to demonstrate

// working of Collections.sort()

import java.util.\*;

import java.lang.\*;

import java.io.\*;

// Point class which does not implement

// Comparable interface. Thus the objects

// of this class are not comparable.

class Point

{

int x, y;

Point(int x, int y)

{

this.x = x;

this.y = y;

}

}

// This class implements

// Comparator interface

class MyCmp implements Comparator<Point>

{

// Sorts the Point objects according

// to x-coordinates in natural order

public int compare(Point p1, Point p2)

{

return p1.x - p2.x;

}

}

// Main Class

class GfG

{

public static void main(String[] args)

{

// Create a list of Integers

List<Point> list = new ArrayList<Point>();

list.add(new Point(5, 10));

list.add(new Point(2, 20));

list.add(new Point(10, 30));

// List is sorted in natural order

// Passing the list and MyCmp object

Collections.sort(list, new MyCmp());

// Displaying the points

for (Point p: list)

System.out.println(p.x + " " + p.y);

}

}

**Output:**

2 20

5 10

10 30

**Arrays.sort() vs Collections.sort()**

* Arrays.sort works for arrays which can be of primitive data type also. Collections.sort() works for objects Collections like ArrayList, LinkedList, etc.
* Arrays.sort can be used in two cases i.e., to sort the whole array or sort a subarray and along with this it provides multiple declarations for different primitive data types but Collections.sort can be used in two cases i.e., using a comparator and not using a comparator for all data types.

Mark as Read

Report An Issue