Sort HashMap by values

In previous post we saw how to sort HashMap by keys. In this post we will see how to sort HashMap by values.

* First let us fill the Map with Key-Value pair.

Map<String, String> map =**new** HashMap<String, String>();

map.put("Robert", "Baratheon");

map.put("Roose", "Bolton");

map.put("Tyrion", "Lannister");

map.put("Eddard", "Stark");

map.put("Benjen", "Stark");

map.put("Aegon", "Targaryen");

map.put("Edmurd", "Tully");

* As we will be sorting the Map with values we need to take care of keys too, because each value is associated with a key. If we don’t take this into consideration then key-value mapping may be not be same as original map.

So we take List<E> of entrySet().

List<Entry<String, String>> entries = **new** ArrayList<Entry<String, String>>(map.entrySet());

* Now let us write a Comparator for the entry of Map.

Comparator<Entry<String,String>> valuesComparator=**new** Comparator<Entry<String,String>>(){

**public** **int** compare(Entry<String, String> s1, Entry<String, String> s2){

**return** s1.getValue().compareTo(s2.getValue());

}

};

* Now we will sort the List using the valueComparator that we just wrote.

Collections.*sort*(entries, valuesComparator);

* Now define a new LinkedHashMap<String, String> and insert all the entries from the List. As List maintains the insertion order and LinkedHashMap also maintains the insertion order we are bound to get sorted order.

Map<String, String> newMap = **new** LinkedHashMap<String, String>();

**for**(Entry<String, String> entry : entries){

newMap.put(entry.getKey(), entry.getValue());

}

Below is the code for this method.

/\*\*

\* This method is used to sort the Map.

\* \*/

**public** **static** **void** sort() {

/\*\*

\* Fill the map with key-value mappings.

\* \*/

Map<String, String> map =**new** HashMap<String, String>();

map.put("Robert", "Baratheon");

map.put("Roose", "Bolton");

map.put("Tyrion", "Lannister");

map.put("Eddard", "Stark");

map.put("Benjen", "Stark");

map.put("Aegon", "Targaryen");

map.put("Edmurd", "Tully");

/\*\*

\* Insert entire entries into the ArrayList.

\* \*/

List<Entry<String, String>> entries = **new** ArrayList<Entry<String, String>>(map.entrySet());

/\*\*

\* Write a Comparator for the values.

\* Compartor compares values in ascending order.

\* \*/

Comparator<Entry<String,String>> valuesComparator=**new** Comparator<Entry<String,String>>() {

**public** **int** compare(Entry<String, String> s1, Entry<String, String> s2){

**return** s1.getValue().compareTo(s2.getValue());

}

};

/\*\*

\* Sort the list using comparator.

\* \*/

Collections.*sort*(entries, valuesComparator);

/\*\*

\* Insert values from ArrayList<Entry<String, String>> to LinkedHashMap<String, String>

\* \*/

Map<String, String> newMap = **new** LinkedHashMap<String, String>();

**for** (Entry<String, String> entry : entries) {

newMap.put(entry.getKey(), entry.getValue());

}

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

}

Output:

Original Map {Eddard=Stark, Aegon=Targaryen, Roose=Bolton, Robert=Baratheon, Benjen=Stark, Edmurd=Tully, Tyrion=Lannister}

Sorted by Values Map {Robert=Baratheon, Roose=Bolton, Tyrion=Lannister, Eddard=Stark, Benjen=Stark, Aegon=Targaryen, Edmurd=Tully}