Different ways to traverse the Map<K, V>

We saw what is [Map<K, V> interface](http://data-structure-learning.blogspot.com/2015/05/java-collections-part-4map-interface.html) and its methods.

We see 4 different ways to traverse the Map<K, V>.

First, we take all the keys from map using keySet() method. Then we can get the value of Key using the get(Object key) method.

/\*\*

\* keySet() is one of the Collection views to see the data in Map.

\* keySet() returns the Set of keys and we can get value of keys

\* by get(Object key) method.

\* \*/

**public** **static** **void** traverseByKeys(Map<String, String> map) {

// Returns the Set view of keys

Set<String> keySet = map.keySet();

**for** (String key : keySet) {

System.***out***.println(key + " " + map.get(key));

}

}

Second, we again use keySet() method to get the set of keys. Then we take the iterator from the keySet() and iterate over the Map.

/\*\*

\* The above method is used to traverse using enhanced for loop on Set

\* collection. In this method we take iterator of set and then

\* traverse the map.

\* \*/

**public** **static** **void** traverseByKeysIterator(Map<String, String> map) {

Iterator<String> keySet = map.keySet().iterator();

**while** (keySet.hasNext()) {

String K = keySet.next();

System.***out***.println(K + " " + map.get(K));

}

}

Third, we take entire entry of Key and Value from Map. This entire entry is returned in Set. Now we use the enhanced for loop to print the key and value.

/\*\*

\* entrySet is another view that is provided by Map Interface.

\* entrySet returns the Set of Map.Entry.

\* We can then traverse using the enhanced for loop.

\* \*/

**public** **static** **void** traverseByEntrySet(Map<String, String> map) {

Set<Map.Entry<String, String>> entries = map.entrySet();

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

System.***out***.println(entry.getKey() + " " + entry.getValue());

}

}

Fourth, we again take the entrySet and then we take iterator of entrySet. Then we can iterate and print the entry.

/\*\*

\* entrySet is another view that is provided by Map Interface.

\* entrySet returns the Set of Map.Entry.

\* Then we take the iterator for the Entry<K,V> and iterate

\* through the map.

\* \*/

**public** **static** **void** traverseByEntrySetIterator(Map<String, String> map) {

Iterator<Entry<String, String>> entries = map.entrySet().iterator();

**while** (entries.hasNext()) {

Entry<String, String> entry = entries.next();

System.***out***.println(entry.getKey() + " " + entry.getValue());

}

}

Below is the code for entire program:

**package** org.collections;

**import** java.util.HashMap;

**import** java.util.Iterator;

**import** java.util.Map;

**import** java.util.Map.Entry;

**import** java.util.Set;

/\*\*

\* This class demonstrates the traversals ways for HashMap.

\* \*/

**public** **class** TraverseMap {

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

Map<String, String> map = *populateMap*();

*traverseByKeys*(map);

*traverseByKeysIterator*(map);

*traverseByEntrySet*(map);

*traverseByEntrySetIterator*(map);

}

/\*\*

\* This method is used to enter data into HashMap.

\* \*/

**public** **static** Map<String, String> populateMap() {

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

map.put("Jonny", "A+");

map.put("John", "B");

map.put("Leo", "B+");

map.put("Josh", "A-");

map.put("Kevin", "A+");

map.put("Sam", "B-");

map.put("Dean", "A");

**return** map;

}

/\*\*

\* keySet() is one of the Collection views to see the data in Map.

\* keySet() returns the Set of keys and we can get value of keys

\* by get(Object key) method.

\* \*/

**public** **static** **void** traverseByKeys(Map<String, String> map) {

// Returns the Set view of keys

Set<String> keySet = map.keySet();

**for** (String key : keySet) {

System.***out***.println(key + " " + map.get(key));

}

}

/\*\*

\* The above method is used to traverse using enhanced for loop on Set

\* collection. In this method we take iterator of set and then

\* traverse the map.

\* \*/

**public** **static** **void** traverseByKeysIterator(Map<String, String> map) {

Iterator<String> keySet = map.keySet().iterator();

**while** (keySet.hasNext()) {

String K = keySet.next();

System.***out***.println(K + " " + map.get(K));

}

}

/\*\*

\* entrySet is another view that is provided by Map Interface.

\* entrySet returns the Set of Map.Entry.

\* We can then traverse using the enhanced for loop.

\* \*/

**public** **static** **void** traverseByEntrySet(Map<String, String> map) {

Set<Map.Entry<String, String>> entries = map.entrySet();

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

System.***out***.println(entry.getKey() + " " + entry.getValue());

}

}

/\*\*

\* entrySet is another view that is provided by Map Interface.

\* entrySet returns the Set of Map.Entry.

\* Then we take the iterator for the Entry<K,V> and iterate

\* through the map.

\* \*/

**public** **static** **void** traverseByEntrySetIterator(Map<String, String> map) {

Iterator<Entry<String, String>> entries = map.entrySet().iterator();

**while** (entries.hasNext()) {

Entry<String, String> entry = entries.next();

System.***out***.println(entry.getKey() + " " + entry.getValue());

}

}

}