

Sets and Maps

Sources

Horstmann, Chapter 16

Summary:

Sets:

A Set is an unordered collection of elements, similar to an ArrayList. However, there are a few crucial differences between a set and an ArrayList. In a set, the elements may not remain in the same order in which they were inserted. Also, there are no duplicates allowed in a set. Similar to a list of printers, you only care that all elements are present and do not have any duplicates. Order is not a concern.

There are 2 main types of sets, HashSets and TreeSet. In HashSets, all elements must provide a hashCode() method. In a TreeSet, the elements are placed in an order based on the compareTo() method. As a rule of thumb, a HashSet should be used, unless you need to visit the elements in a sorted order.

The iterator for a Set is not the same as a list iterator. It lacks the add() and the previous() methods. The main reason is that when iterating through a set, there is no explicit order to the elements.

Maps:

A Map keeps associations between key and value objects. It is similar to a dictionary. You look up a word (the key) and get the definition (the value). In a map, every key has a single value. Each value, on the other hand, can be associated with several keys. All the keys in a Map are stored in one set (the key set), and all the values are stored in another set (the value set).

Just like Sets there are 2 kinds of implementations of Maps. A HashMap stores its elements in a seemingly random order, while a TreeMap visits its elements in a sorted order.

Example: (See next page)

```

1 public static void main(String[] args) {
2     Set<String> hashNames = new HashSet<String>();
3     Set<String> treeNames = new TreeSet<String>();
4
5     //Add some fake data
6     hashNames.add("John");
7     treeNames.add("John");
8
9     hashNames.add("Paul");
10    treeNames.add("Paul");
11
12    hashNames.add("Karl");
13    treeNames.add("Karl");
14
15    //These are not sorted
16    for (String name : hashNames) {
17        System.out.println(name);
18    }
19    System.out.println("----");
20    //These are sorted
21    for (String name : treeNames) {
22        System.out.println(name);
23    }
24
25    System.out.println("-----");
26
27    Map <String, Color> hashColorMap = new HashMap<String, Color>();
28    Map <String, Color> treeColorMap = new TreeMap<String, Color>();
29
30    //Add some fake data
31    hashColorMap.put("Green",Color.GREEN);
32    treeColorMap.put("Green",Color.GREEN);
33
34    hashColorMap.put("Red",Color.RED);
35    treeColorMap.put("Red",Color.RED);
36
37    hashColorMap.put("Blue",Color.BLUE);
38    treeColorMap.put("Blue",Color.BLUE);
39
40    hashColorMap.put("Yellow",Color.YELLOW);
41    treeColorMap.put("Yellow",Color.YELLOW);
42
43    //This map is unsorted
44    Set<String> hashColorKeys = hashColorMap.keySet();
45    for (String colorName : hashColorKeys) {
46        Color color = hashColorMap.get(colorName);
47        System.out.println(colorName + " : " + color.toString());
48    }

```

```
49 System.out.println("----");
50 //While this one is
51 Set<String> treeColorKeys = treeColorMap.keySet();
52 for (String colorName : treeColorKeys) {
53     Color color = treeColorMap.get(colorName);
54     System.out.println(colorName + " : " + color.toString());
55 }
56 }
```