How to use different types of Collections?

Solution:

Following example uses different types of collection classes and adds an element in those collections.

import java.util.Map;

import java.util.Set;

import java.util.SortedMap;

import java.util.SortedSet;

import java.util.TreeMap;

import java.util.TreeSet;

import java.util.ArrayList;

import java.util.Collection;

import java.util.HashMap;

import java.util.HashSet;

import java.util.Iterator;

import java.util.LinkedHashMap;

import java.util.LinkedHashSet;

import java.util.LinkedList;

import java.util.List;

public class Main {

public static void main(String[] args) {

List lnkLst = new LinkedList();

lnkLst.add("element1");

lnkLst.add("element2");

lnkLst.add("element3");

lnkLst.add("element4");

displayAll(lnkLst);

List aryLst = new ArrayList();

aryLst.add("x");

aryLst.add("y");

aryLst.add("z");

aryLst.add("w");

displayAll(aryLst);

Set hashSet = new HashSet();

hashSet.add("set1");

hashSet.add("set2");

hashSet.add("set3");

hashSet.add("set4");

displayAll(hashSet);

SortedSet treeSet = new TreeSet();

treeSet.add("1");

treeSet.add("2");

treeSet.add("3");

treeSet.add("4");

displayAll(treeSet);

LinkedHashSet lnkHashset = new LinkedHashSet();

lnkHashset.add("one");

lnkHashset.add("two");

lnkHashset.add("three");

lnkHashset.add("four");

displayAll(lnkHashset);

Map ma p1 = new HashMap();

map1.put("key1", "J");

map1.put("key2", "K");

map1.put("key3", "L");

map1.put("key4", "M");

displayAll(map1.keySet());

displayAll(map1.values());

SortedMap map2 = new TreeMap();

map2.put("key1", "JJ");

map2.put("key2", "KK");

map2.put("key3", "LL");

map2.put("key4", "MM");

displayAll(map2.keySet());

displayAll(map2.values());

LinkedHashMap map3 = new LinkedHashMap();

map3.put("key1", "JJJ");

map3.put("key2", "KKK");

map3.put("key3", "LLL");

map3.put("key4", "MMM");

displayAll(map3.keySet());

displayAll(map3.values());

}

static void displayAll(Collection col) {

Iterator itr = col.iterator();

while (itr.hasNext()) {

String str = (String) itr.next();

System.out.print(str + " ");

}

System.out.println();

}

}