

```

1 // Example 4.4 joi/examples/TreeMapDemo.java
2 //
3 //
4 // Copyright 2003 Bill Campbell and Ethan Bolker
5
6 import java.util.TreeMap;
7 import java.util.Iterator;
8 import java.util.Set;
9 import java.util.Collection;
10 import java.util.Map;
11
12 // A class illustrating the use of TreeMap. A typical run:
13 //
14 // %> java TreeMapDemo
15 // Store 3 wrapped ints, keys "one", "two", "three".
16 // The wrapped int stored for "two" is 2
17 //
18 // Iterate over keys, get each value.
19 // Note that key order is alphabetical:
20 // The value for key one is 1
21 // The value for key three is 3
22 // The value for key two is 2
23 //
24 // Iterate over the values:
25 // 1
26 // 3
27 // 2
28 //
29 // Iterate over the key-value pairs:
30 // The value for the entry with key one is 1
31 // The value for the entry with key three is 3
32 // The value for the entry with key two is 2
33 //
34 // How a TreeMap represents itself as a String:
35 // {one=1, three=3, two=2}
36 //
37 // Store a different value at key "two"
38 // {one=1, three=3, two=2222}
39 //
40 // Store map.get( "one" ) at key "two"
41 // {one=1, three=3, two=1}
42 //
43 // A TreeMap with Integer keys mapping to String values
44 // {1=I, 2=II, 3=III}
45 // %>
46
47 public class TreeMapDemo
48 {
49     public static void main( String[] args )
50     {
51         Terminal terminal = new Terminal(); // for input and output
52
53         TreeMap map = new TreeMap();
54
55         // Put in some ints (each wrapped up as an Integer object)
56         terminal.println(

```

```

57         // Store 3 wrapped ints, keys \"one\", \"two\", \"three\".");
58         map.put("one", new Integer(1) );
59         map.put("two", new Integer(2) );
60         map.put("three", new Integer(3) );
61
62         // get the value associated with a key;
63         // notice the required cast.
64         Integer wrappedInt = (Integer) map.get( "two" );
65
66         // And print the wrapped int
67         terminal.println( "The wrapped int stored for \"two\" is "
68             + wrappedInt);
69
70
71         // The set of keys.
72         Set keys = map.keySet();
73         // The iterator over this "set" of keys will return
74         // the keys in key-order.
75         terminal.println( "\nIterate over keys, get each value. " );
76         terminal.println( "Note that key order is alphabetical:" );
77         Iterator keyIterator = keys.iterator();
78         while ( keyIterator.hasNext() ) {
79             String key = (String) keyIterator.next();
80             terminal.println( "The value for key " + key + " is "
81                 + ((Integer) map.get( key )) );
82         }
83
84         // Iterate over the collection of values;
85         // notice the order is the same (ie the key-order).
86         terminal.println( "\nIterate over the values:" );
87         Iterator valueIterator = map.values().iterator();
88         while ( valueIterator.hasNext() ) {
89             terminal.println( ((Integer) valueIterator.next());
90         }
91
92         // The set of Map.Entry objects (key-value pairs);
93         // Map.Entry is an inner class of Map.
94
95         // Iterate over the entries.
96         terminal.println( "\nIterate over the key-value pairs:" );
97         Iterator entryIterator = map.entrySet().iterator();
98         while ( entryIterator.hasNext() ) {
99             Map.Entry entry = (Map.Entry) entryIterator.next();
100             terminal.println( "The value for the entry with key "
101                 + entry.getKey() + " is "
102                 + ((Integer) entry.getValue());
103         }
104
105         // how a TreeMap represents itself as a String:
106         terminal.println(
107             "\nHow a TreeMap represents itself as a String:" );
108         terminal.println( map.toString() );
109         terminal.println();
110
111         // We can overwrite the value stored under a key
112         terminal.println(

```

```
113     "Store a different value at key \"two\"");
114     map.put("two", new Integer(2222));
115     terminal.println(map.toString());
116     terminal.println();
117
118     // We can store the same value under two keys
119     terminal.println(
120         "Store map.get( \"one\" ) at key \"two\"");
121     map.put("two", map.get( "one" ) );
122     terminal.println(map.toString());
123     terminal.println();
124
125     // And keys don't necessarily have to be Strings;
126     // Here's a TreeMap mapping Integers to strings.
127     terminal.println(
128         "A TreeMap with Integer keys mapping to String values");
129     map = new TreeMap();
130     map.put( new Integer( 1 ), "I" );
131     map.put( new Integer( 2 ), "II" );
132     map.put( new Integer( 3 ), "III" );
133     terminal.println(map.toString());
134
135 }
136 }
```