```
 1
 2 import java.util.Random;
 3 /**
 4  * An Updated simple test for SortedIntList.
 5  *
 6  * <ul>
 7  * <li> Name: SimpleTest.java
 8  * <li> Description: Sorted Int List Simple Test
 9  * <li> Class: Java 145
10  * <li> Instructor: Ken Hang
11  * <li> Date: Feb 2 2015
12  * </ul>
13  *
14  * @author  Hai H Nguyen (Bill)
15  * @version Winter 2015
16  */
17 public class SimpleTest {
18
19     //Random Duplicate Indicator
20     public static final int MAX_INT = 9;
21
22     public static void main(String[] args) {
23         test1();
24
25         test2();
26
27         //ExtraTest.test3();  //These invoke
28
29         //ExtraTest.test4();  //test cases extracted
30
31         //ExtraTest.test5();  //from model tests on canvas
32     }
33
34     //First Test
35     //Test Uniqueness and Sort order
36     public static void test1() {
37         boolean unique = false;
38
39         SortedIntList list = new SortedIntList(unique);
40
41         fill(list, 18);
42
43         list.setUnique(unique = true);
44
45         fill(list, 18);
46
47         boolean failFlag = ((list.getUnique() != unique) || (!isSorted(list)));
48
49         list.clear();
50
51         failFlag = failFlag || (!list.toString().equals("[]"));
52
53         if (failFlag){
54             System.out.println("\nTest 1 Failed!\n");
55         } else{
56             System.out.println("\nTest 1 Passed!\n");
57         }
58     }
59
60     //Second Test
61     //Test toString, min, max and size
62     public static void test2() {
63         if (checkToString() && checkMinMax() && checkSize()) {
64             System.out.println("\nTest 2 Passed!");
65         } else {
66             System.out.println("\nTest 2 Failed!");
67         }
68     }
```

```
69
70      // returns true if list is sorted, false otherwise
71      public static boolean isSorted(SortedIntList list) {
72          for (int i = 0; i < list.size() - 1; i++) {
73              if (list.get(i) > list.get(i + 1)) {
74                  return false;
75              }
76          }
77
78          return true;
79      }
80
81      // Test the toString Method.
82      public static boolean checkToString() {
83          SortedIntList list1 = new SortedIntList(10);
84
85          fillNine(list1);
86
87          list1.add(0);
88
89          String test1 =
90                  "[-9, -8, -7, -6, -5, -4, -3, -2, -1, " +
91                  "0, 1, 2, 3, 4, 5, 6, 7, 8, 9]";
92
93          SortedIntList list2 = new SortedIntList(true);
94
95          for (int i = 9; i > 0; --i) {
96              for (int j = 0; j < 9; ++j) {
97                  list2.add(i * 9);
98              }
99          }
100
101         String test2 = "[9, 18, 27, 36, 45, 54, 63, 72, 81]";
102
103         boolean failFlag = !( test1.equals(list1.toString()) &&
104                               test2.equals(list2.toString()));
105
106         System.out.println(failFlag ?
107                 "toString Failed!" : "toString Ok!");
108
109         return (!failFlag);
110     }
111
112     // Test the min/max methods.
113     public static boolean checkMinMax() {
114         //Should be -9 and 9
115
116         SortedIntList list = new SortedIntList();
117
118         fillNine(list);
119
120         boolean failFlag = list.min() != -9 && list.max() != 9;
121
122         System.out.println(failFlag ?
123                 "min and max Failed!" : "min and max Ok!");
124
125         return (!failFlag);
126     }
127
128     // Checks to see if the list has the appropriate size.
129     public static boolean checkSize() {
130         SortedIntList list = new SortedIntList();
131
132         fillNine(list);
133
134         boolean failFlag = list.size() != 18;
135
136         System.out.println(failFlag ?
```

```
137                 "size Failed!" : "size Ok!");
138
139         return (!failFlag);
140     }
141
142     //pre: list is a SortedIntList, size will not exceed l's capacity
143     //post: list is filled with TEST_SIZE random values
144     public static void fill(SortedIntList list, int size) {
145         Random rand = new Random();
146
147         list.clear();
148
149         for (int i = 0; i < size; i++) {
150             list.add(rand.nextInt(MAX_INT));
151         }
152     }
153
154     // Add integers from -9 to 9 into the list
155     public static void fillNine(SortedIntList list){
156         for(int i = 9; i > 0; --i){
157             list.add(i);
158
159             list.add(-i);
160         }
161     }
162 }
```