~\OneDrive - VietNam National University - HCM INTERNATIONAL UNIVERSITY\Desktop\DSA\DSA LAB NEW\Lab 2 Simple sorting\ITITSB22029_DoMinhDuy_Lab2\Problem 1\BubbleSortApp.java

```
1  class ArrayBub {
2     private long[] a; // ref to array a
3     private int nElems; // number of data items
4     private int nSwaps; // number of swaps
5     private int nComparisons; // number of comparisons
6     // --------------------------------------------------------------
7
8     public ArrayBub(int max) // constructor
9     {
10       a = new long[max]; // create the array
11       nElems = 0; // no items yet
12       nSwaps = 0; // no swaps yet
13       nComparisons = 0; // no comparisons yet
14    }
15
16    // --------------------------------------------------------------
17    public void insert(long value) // put element into array
18    {
19       a[nElems] = value; // insert it
20       nElems++; // increment size
21    }
22
23    // --------------------------------------------------------------
24    public void display() // displays array contents
25    {
26       for (int j = 0; j < nElems; j++) // for each element,
27          System.out.print(a[j] + " "); // display it
28       System.out.println("");
29    }
30
31    // --------------------------------------------------------------
32    public void bubbleSort() {
33       int out, in;
34
35       for (out = nElems - 1; out > 0; out--) { // outer loop (backward)
36          System.out.println("Outer loop at index " + out + ":");
37          for (in = 0; in < out; in++) { // inner loop (forward)
38             nComparisons++; // increment number of comparisons
39             if (a[in] > a[in + 1]) // out of order?
40                swap(in, in + 1); // swap them
41          }
42          display(); // display array after each outer loop
43          System.out.println("Number of swaps so far: " + nSwaps);
44          System.out.println("Number of comparisons so far: " + nComparisons);
45       }
46       System.out.println("Total comparisons made: " + nComparisons);
47    } // end bubbleSort()
```

```
48      // --------------------------------------------------------------
49
50    private void swap(int one, int two) {
51       long temp = a[one];
52       a[one] = a[two];
53       a[two] = temp;
54
55       nSwaps++; // increase number of swaps by 1
56    }
57
58    public int getSwapNumber() {
59       return nSwaps;
60    }
61
62    public int getComparisonNumber() {
63       return nComparisons;
64    }
65    // --------------------------------------------------------------
66 } // end class ArrayBub
67   ////////////////////////////////////////////////////////////////
68
69 class BubbleSortApp {
70    public static void main(String[] args) {
71       int maxSize = 100; // array size
72       ArrayBub arr; // reference to array
73       arr = new ArrayBub(maxSize); // create the array
74
75       arr.insert(77); // insert 10 items
76       arr.insert(99);
77       arr.insert(44);
78       arr.insert(55);
79       arr.insert(22);
80       arr.insert(88);
81       arr.insert(11);
82       arr.insert(00);
83       arr.insert(66);
84       arr.insert(33);
85
86       arr.display(); // display items before sorting
87
88       arr.bubbleSort(); // bubble sort them
89
90       arr.display(); // display items after sorting
91
92       // display the number of swaps and comparisons
93       System.out.println("The number of swaps = " + arr.getSwapNumber());
94       System.out.println("The total number of comparisons = " + arr.getComparisonNumber());
95
96       // The algorithm complexity estimation
97       int n = arr.getComparisonNumber();
```

```
98        System.out.println("Estimated complexity (n*(n-1)/2): " + n * (n - 1) / 2);
99        System.out.println("The algorithm has O(n^2) complexity.");
100     } // end main()
101 } // end class BubbleSortApp
102   ////////////////////////////////////////////////////////////////
103
```