

Selected files

5 printable files

Array.java
ClassDataArray.java
HighArray.java
LowArray.java
OrderedArray.java

Array.java

```
1 // Array.java
2 // demonstrates Java arrays
3 /////
4 class Array {
5     public static void main(String[] args) {
6         long[] arr; // Refer to array
7         arr = new long[100]; // Make array
8
9         int nElems = 0; // Number of items
10        int j;
11        int k;
12        long searchKey;
13
14        arr[0] = 77;
15        arr[1] = 99;
16        arr[2] = 44;
17        arr[3] = 55;
18        arr[4] = 22;
19        arr[5] = 88;
20        arr[6] = 11;
21        arr[7] = 00;
22        arr[8] = 66;
23        arr[9] = 33;
24        nElems = 10;
25
26        for (j = 0; j < nElems; j++) {
27            System.out.println(arr[j] + " ");
28        }
29        System.out.println("");
30
31        // Find and display
32        searchKey = 66;
33        for (j = 0; j < nElems; j++) {
34            if (arr[j] == searchKey) {
35                break;
36            }
37        }
38        if (j == nElems) {
39            System.out.println("Can not find " + searchKey);
```

```

40     } else {
41         System.out.println("Found " + searchKey);
42     }
43
44     // Find and delete
45     searchKey = 55;
46     for (j = 0; j < nElems; j++) {
47         if (arr[j] == searchKey) {
48             break;
49         }
50     }
51     for (k = j; k < nElems - 1; k++) {
52         arr[k] = arr[k + 1];
53     }
54     nElems--;
55
56     for (j = 0; j < nElems; j++) {
57         System.out.println(arr[j] + " ");
58     }
59
60     System.out.println("");
61 }
62 }

```

ClassDataArray.java

```

1 // classDataArray.java
2 // data items as class objects
3
4 class Person {
5     private String lastName;
6     private String firstName;
7     private int age;
8
9     // -----
10    public Person(String last, String first, int a) { // constructor
11        lastName = last;
12        firstName = first;
13        age = a;
14    }
15
16    // -----
17    public void displayPerson() {
18        System.out.print("    Last name: " + lastName);
19        System.out.print(", First name: " + firstName);
20        System.out.println(", Age: " + age);
21    }
22
23    public String getLast() // get last name
24    {
25        return lastName;

```

```
26     }
27 } // end class Person
28
29 class ClassDataArray {
30     private Person[] a; // reference to array
31     private int nElems; // number of data items
32
33     public ClassDataArray(int max) // constructor
34     {
35         a = new Person[max]; // create the array
36         nElems = 0; // no items yet
37     }
38
39     // -----
40     public Person find(String searchName) { // find specified value
41         int j;
42         for (j = 0; j < nElems; j++) // for each element,
43             if (a[j].getLast().equals(searchName)) // found item?
44                 break; // exit loop before end
45         if (j == nElems) // gone to end?
46             return null; // yes, can't find it
47         else
48             return a[j]; // no, found it
49     } // end find()
50     // -----
51     // put person into array
52
53     public void insert(String last, String first, int age) {
54         a[nElems] = new Person(last, first, age);
55         nElems++; // increment size
56     }
57
58     // -----
59     public boolean delete(String searchName) { // delete person from array
60         int j;
61         for (j = 0; j < nElems; j++) // look for it
62             if (a[j].getLast().equals(searchName))
63                 break;
64         if (j == nElems)
65             return false; // can't find it
66         else // found it
67         {
68             for (int k = j; k < nElems; k++) // shift down
69                 a[k] = a[k + 1];
70             nElems--; // decrement size
71             return true;
72         }
73     } // end delete()
74     // -----
75 }
```

```
76     public void displayA() // displays array contents
77     {
78         for (int j = 0; j < nElems; j++) // for each element,
79             a[j].displayPerson(); // display it
80     }
81     // -----
82 }
83
84 class ClassDataApp {
85     public static void main(String[] args) {
86         int maxSize = 100; // array size
87         ClassDataArray arr; // reference to array
88         arr = new ClassDataArray(maxSize); // create the array
89         // insert 10 items
90         arr.insert("Evans", "Patty", 24);
91         arr.insert("Smith", "Lorraine", 37);
92         arr.insert("Yee", "Tom", 43);
93         arr.insert("Adams", "Henry", 63);
94         arr.insert("Hashimoto", "Sato", 21);
95         arr.insert("Stimson", "Henry", 29);
96         arr.insert("Velasquez", "Jose", 72);
97         arr.insert("Lamarque", "Henry", 54);
98         arr.insert("Vang", "Minh", 22);
99         arr.insert("Creswell", "Lucinda", 18);
100        arr.displayA(); // display items
101        String searchKey = "Stimson"; // search for item
102        Person found;
103
104        found = arr.find(searchKey);
105        if (found != null) {
106            System.out.print("Found ");
107            found.displayPerson();
108        } else
109            System.out.println("Can't find " + searchKey);
110        System.out.println("Deleting Smith, Yee, and Creswell");
111        arr.delete("Smith"); // delete 3 items
112        arr.delete("Yee");
113        arr.delete("Creswell");
114        arr.displayA(); // display items again
115    } // end main()
116 } // end class ClassDataApp
```

HighArray.java

```
1 // HighArray.java
2 // demonstrates array class with high-level interface
3 //
4
5 public class HighArray {
6     private long[] a;
7     private int nElems;
```

```
8
9 public HighArray(int max) {
10     a = new long[max];
11     nElems = 0;
12 }
13
14 public boolean find(long searchKey) {
15     int j;
16     for (j = 0; j < nElems; j++) {
17         if (a[j] == searchKey) {
18             break;
19         }
20     }
21     if (j == nElems) {
22         return false;
23     } else {
24         return true;
25     }
26 }
27
28 public void insert(long value) {
29     a[nElems] = value;
30     nElems++;
31 }
32
33 public boolean delete(long value) {
34     int j;
35     for (j = 0; j < nElems; j++) {
36         if (value == a[j]) {
37             break;
38         }
39     }
40
41     if (j == nElems) {
42         return false;
43     }
44
45     else {
46         for (int k = j; k < nElems; k++) {
47             a[k] = a[k + 1];
48         }
49         nElems--;
50         return true;
51     }
52 }
53
54 public void display() {
55     for (int j = 0; j < nElems; j++) {
56         System.out.println(a[j] + " ");
57     }
```

```
58         System.out.println("");
59     }
60 }
61
```

LowArray.java

```
1  // LowArray.java
2  // demonstrates array class with low-level interface
3  /////
4  class LowArray {
5      private long[] a; // Refer to array a
6
7      public LowArray(int size) {
8          a = new long[size]; // Create array
9      }
10
11     public void setElem(int index, long value) {
12         a[index] = value; // Setter
13     }
14
15     public long getElem(int index) {
16         return a[index]; // Getter
17     }
18
19     public static void main(String[] args) {
20         LowArray arr;
21         arr = new LowArray(100);
22         int nElems = 0;
23         int j;
24
25         arr.setElem(0, 77); // insert 10 items
26         arr.setElem(1, 99);
27         arr.setElem(2, 44);
28         arr.setElem(3, 55);
29         arr.setElem(4, 22);
30         arr.setElem(5, 88);
31         arr.setElem(6, 11);
32         arr.setElem(7, 00);
33         arr.setElem(8, 66);
34         arr.setElem(9, 33);
35         nElems = 10;
36
37         for (j = 0; j < nElems; j++) {
38             System.err.println(arr.getElem(j) + " ");
39         }
40         System.out.println("");
41
42         // Find and display
43         int searchKey = 26;
44         for (j = 0; j < nElems; j++) {
```

```
45         if (arr.getElem(j) == searchKey) {
46             break;
47         }
48     }
49     if (j == nElems) {
50         System.out.println("Can not find " + searchKey);
51     } else {
52         System.out.println("Found" + searchKey);
53     }
54
55     // Delete 55
56     for (j = 0; j < nElems; j++) {
57         if (arr.getElem(j) == 55) {
58             break;
59         }
60     }
61
62     for (int k = j; k < nElems; k++) {
63         arr.setElem(k, arr.getElem(k + 1));
64     }
65     nElems--;
66
67     for (j = 0; j < nElems; j++) {
68         System.out.println(arr.getElem(j) + " ");
69     }
70     System.out.println("");
71 }
72 }
```

OrderedArray.java

```
1 // OrderedArray.java
2 // demonstrates ordered array class
3
4 class OrderedArray {
5     private long[] a; // ref to array a
6     private int nElems; // number of data items
7
8     public OrderedArray(int max) // constructor
9     {
10         a = new long[max]; // create array
11         nElems = 0;
12     }
13
14     public int size() {
15         return nElems;
16     }
17
18     public int find(long searchKey) {
19         int lowerBound = 0;
20         int upperBound = nElems - 1;
```

```
21     int curIn;
22     while (true) {
23         curIn = (lowerBound + upperBound) / 2;
24         if (a[curIn] == searchKey)
25             return curIn; // found it
26         else if (lowerBound > upperBound)
27             return nElems; // can't find it
28         else // divide range
29             {
30                 if (a[curIn] < searchKey)
31                     lowerBound = curIn + 1; // it's in upper half
32                 else
33                     upperBound = curIn - 1; // it's in lower half
34             } // end else divide range
35     } // end while
36 } // end find()
37
38 public void insert(long value) // put element into array
39 {
40     int j;
41     for (j = 0; j < nElems; j++) // find where it goes
42         if (a[j] > value) // (linear search)
43             break;
44     for (int k = nElems; k > j; k--) // move bigger ones up
45         a[k] = a[k - 1];
46     a[j] = value; // insert it
47     nElems++; // increment size
48 } // end insert()
49
50 public boolean delete(long value) {
51     int j = find(value);
52     if (j == nElems) // can't find it
53         return false;
54     else // found it
55     {
56         for (int k = j; k < nElems; k++) // move bigger ones down
57             a[k] = a[k + 1];
58         nElems--; // decrement size
59         return true;
60     }
61 } // end delete()
62
63 public void display() // displays array contents
64 {
65     for (int j = 0; j < nElems; j++) // for each element,
66         System.out.print(a[j] + " "); // display it
67     System.out.println("");
68 }
69 }
70
```



```
71 class OrderedApp {
72     public static void main(String[] args) {
73         int maxSize = 100; // array size
74         OrderedArray arr; // reference to array
75
76         arr = new OrderedArray(maxSize); // create the array
77         arr.insert(77); // insert 10 items
78         arr.insert(99);
79         arr.insert(44);
80         arr.insert(55);
81         arr.insert(22);
82         arr.insert(88);
83         arr.insert(11);
84         arr.insert(00);
85         arr.insert(66);
86         arr.insert(33);
87
88         int searchKey = 55; // search for item
89         if (arr.find(searchKey) != arr.size()) {
90             System.out.println("Found " + searchKey);
91         } else {
92             System.out.println("Can not find " + searchKey);
93
94         }
95         arr.display(); // display items
96         arr.delete(00); // delete 3 items
97         arr.delete(55);
98         arr.delete(99);
99         arr.display(); // display items again
100     }
101
102 } // end main()
103
```