

```

1 // foj/9/bank/class Month
2 //
3 //
4 // Copyright 2003 Bill Campbell and Ethan Bolker
5
6 import java.io.*;
7 import java.util.Calendar;
8
9 /**
10  * The Month class implements an object that keeps
11  * track of the month of the year.
12  *
13  * @version 9
14  */
15
16 public class Month
17     implements Serializable
18 {
19     private static final String[] monthName =
20         { "Jan", "Feb", "Mar", "Apr", "May", "Jun",
21           "Jul", "Aug", "Sep", "Oct", "Nov", "Dec" };
22
23     private int month;
24     private int year;
25
26     /**
27      * Month constructor constructs a Month object
28      * initialized to the current month and year.
29      */
30
31     public Month()
32     {
33         Calendar rightNow = Calendar.getInstance();
34         month = rightNow.get( Calendar.MONTH );
35         year = rightNow.get( Calendar.YEAR );
36     }
37
38     /**
39      * Advance to next month.
40      */
41
42     public void next()
43     {
44         month = (month + 1) % 12;
45         if (month == 0) {
46             year++;
47         }
48     }
49
50     /**
51      * How a Month is displayed as a String -
52      * for example, "Jan, 2003".
53      *
54      * @return String representation of the month.
55      */
56

```

```

57     public String toString()
58     {
59         return monthName[month] + ", " + year;
60     }
61
62     /**
63      * For unit testing.
64      */
65
66     public static void main( String[] args )
67     {
68         Month m = new Month();
69         for (int i=0; i < 14; i++, m.next()) {
70             System.out.println(m);
71         }
72         for (int i=0; i < 35; i++, m.next()); // no loop body
73         System.out.println( "three years later: " + m );
74         for (int i=0; i < 120; i++, m.next()); // no loop body
75         System.out.println( "ten years later: " + m );
76     }
77 }

```