

```

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

```

```

57     public String toString()
58     {
59         return monthName[month] + ", " + year;
60     }
61
62     /**
63      * For unit testing.
64     */
65     public static void main( String[] args )
66     {
67         Month m = new Month();
68         for (int i=0; i < 14; i++, m.next()) {
69             System.out.println(m);
70
71             for (int i=0; i < 3; i++, m.next()); // no loop body
72             System.out.println( "three years later: " + m );
73             for (int i=0; i < 120; i++, m.next()); // no loop body
74             System.out.println( "ten years later: " + m );
75
76     }
77

```