

```
1  /*
2  Interface for Date objects to be used by the Year 3000 driver program
3  @author Jon Sorenson
4  */
5
6  public interface DateInterface
7  {
8      /** @return the day of the month(1-31)*/
9      public int getDay();
10     /** @return the day of the week(0-6) */
11     public int getDow();
12     /** @return the month of the year(1-12) */
13     public int getMonth();
14     /** @return the year (four digits) */
15     public int getYear();
16     /** sets the date
17      * @param m the month of the year (1-12)
18      * @param d the day of the month (1-31)
19      * @param y the year (four digits)
20      * @param dow the day of the week (0-6) */
21     public void set (int m, int d, int y, int dow);
22     /** moves the date forward by exactly one day */
23     public void tomorrow();
24     /** @return the date as a String in the format "Monday March 18, 2002" */
25     public String toString();
26     /**sets the date to today;\
27      * make this empty {} unless you do the extra credit.
28      */
29     public void today();
30     /** Moves the date backward by exactly one day;
31      * make this empty {} unless you do the extra credit.
32      */
33     public void yesterday();
34 }
```

```
1 import java.io.*;
2
3 /**
4  * Driver class for The Dating Game programming assignment.
5  * @author Jon Sorenson
6  * edited by Casey Sobecks to include extra credit
7  */
8
9 public class Year3000
10 {
11     public static void main(String [] args) throws IOException
12     {
13         DateInterface d= new MyDate();
14         d.today(); //sets the date to current date used by computer
15         System.out.println(d); //prints the current day
16         while(d.getYear()<3000) //moves forward until the day Jan 1 3000 is reached
17         {
18             d.tomorrow();
19         }
20         System.out.println(d); //print out the date and day of week of Jan 1, 3000
21
22         d.today(); //reset the date to today
23         while(d.getYear()>1799) //move backwards until Dec 31 1799 is reached
24         {
25             d.yesterday();
26         }
27         d.tomorrow(); //move forward a day to reach Jan 1, 1800
28         System.out.println(d); //print out the day of the week of Jan 1, 1800
29     }
30 }
31 }
```

```
1 import java.util.*;
2 /*
3 This is the Class file for the Dating Game Assignment
4 CS248 Object Oriented Programming
5 @author Casey Sobecks
6 */
7
8 public class MyDate implements DateInterface
9 {
10     //data memers(variables)
11     int month;
12     int day;
13     int year;
14     int dayofweek;
15     String dayname;
16     String monthname;
17
18     //constructor
19     /** @param m month
20      * @param d day
21      * @param y year
22      * @param dow day of the week */
23     public void set (int m, int d, int y, int dow)
24     {
25         month=m; //months in 1-12
26         day=d; //day in 1-31
27         year=y; //year four digits long
28         dayofweek=dow; //day in 0-6
29     }
30
31     //accessors
32     public int getDay()
33     {
34         return day; //returns the day
35     }
36
37     public int getDow()
38     {
39         return dayofweek; //returns the day of the week
40     }
41
42     public int getMonth()
43     {
44         return month; //returns the month
45     }
46
47     public int getYear()
48     {
49         return year; //returns the year
50     }
51
52     //mutators
53     public void tomorrow()
54     {
55         //First checks the day of the week
56         if(getDow()==6)
57         {
58             dayofweek=0;
59         }
60         else
```

```
61     {
62         dayofweek++;
63     }
64
65     //Next checks the day of the month
66
67     if(getDay()==28) //February
68     {
69         if(getMonth()==2) //Check for Feb
70         {
71             if((getYear()%100==0) && (getYear()%400==0)) //Check for leap year
72             {day++;}
73             else if((getYear()%100!=0) && (getYear()%4==0)) //check for a leap
year
74                 {day++;}
75             else //if not a leap year
76             {
77                 day=1;
78                 month++;
79             }
80         }
81         else //not february
82         {
83             day++;
84         }
85     }
86     else if(getDay()==29) //leap year
87     {
88         if(getMonth()==2) //check for Feb
89         {
90             day=1;
91             month++;
92         }
93         else //not february
94         {
95             day++;
96         }
97     }
98
99     else if(getDay()==30) //30 day months
100    {
101        if((getMonth()==4)|| (getMonth()==6)|| (getMonth()==9)|| (getMonth()==11))
//check if month has 30 days
102        {
103            day=1;
104            month++;
105        }
106        else //month has 31 days
107        {
108            day++;
109        }
110    }
111    else if(getDay()==31) //31 day months
112    {
113        if(getMonth()==12) //check if it's a new year
114        {
115            day=1;
116            month=1;
117            year++;
118        }
119    }
```

```
119         else //not a new year
120         {
121             day=1;
122             month++;
123         }
124     }
125     else //if not end of the month, add a day
126     {
127         day++;
128     }
129 }
130
131 public String toString()
132 {
133     //convert the day of the week number to a string name of the week
134     if(getDow()==0)
135     {
136         dayname="Sunday";
137     }
138     if(getDow()==1)
139     {
140         dayname="Monday";
141     }
142     if(getDow()==2)
143     {
144         dayname="Tuesday";
145     }
146     if(getDow()==3)
147     {
148         dayname="Wednesday";
149     }
150     if(getDow()==4)
151     {
152         dayname="Thursday";
153     }
154     if(getDow()==5)
155     {
156         dayname="Friday";
157     }
158     if(getDow()==6)
159     {
160         dayname="Saturday";
161     }
162
163     //convert the month number into a string name of the month
164     if(getMonth()==1)
165     {
166         monthname="January";
167     }
168     if(getMonth()==2)
169     {
170         monthname="February";
171     }
172     if(getMonth()==3)
173     {
174         monthname="March";
175     }
176     if(getMonth()==4)
177     {
178         monthname="April";
```

```
179     }
180     if(getMonth()==5)
181     {
182         monthname="May";
183     }
184     if(getMonth()==6)
185     {
186         monthname="June";
187     }
188     if(getMonth()==7)
189     {
190         monthname="July";
191     }
192     if(getMonth()==8)
193     {
194         monthname="August";
195     }
196     if(getMonth()==9)
197     {
198         monthname="September";
199     }
200     if(getMonth()==10)
201     {
202         monthname="October";
203     }
204     if(getMonth()==11)
205     {
206         monthname="November";
207     }
208     if(getMonth()==12)
209     {
210         monthname="December";
211     }
212
213     //print out the current date that the program has stored
214     System.out.println("The date is " +dayname + " " +monthname + " " +day +", "
+year);
215     return " "; //a nice space to separate the lines of the outputted code for
easier reading
216 }
217
218 public void today() //get the current date being used by the computer that is
running this program
219 {
220     Date x= new Date(); //get current date in variable
221     int yr=x.getYear()+1900; //extract year from date (year0=1900 so add 1900)
222     int mn=x.getMonth()+1; //extract month from date (January =0 so add one)
223     int da=x.getDate(); //extract date of day from date
224     int daowe=x.getDay(); //extract day of week from date
225     set(mn,da,yr,daowe); //set current date to supplied info from the computer
226 }
227
228
229 public void yesterday() //this code will take the code to the day before
230 {
231     //day of the week
232     if(getDow()==0) //beginning of week
233     {
234         dayofweek=6;
235     }
```

```
236     else //during the week
237     {
238         dayofweek--;
239     }
240
241     //day of the month
242     if(getDay()==1) //first day of the month, so moving back to the last day of
another month
243     {
244         if(getMonth()==3)
245         {
246             if(getYear()%100==0 && getYear()%400==0) //check for leap year
247             {
248                 day=29;
249                 month--;
250             }
251             else if(getYear()%100!=0 && getYear()%4==0) //check for leap year
252             {
253                 day=29;
254                 month--;
255             }
256             else //not leap year
257             {
258                 day=28;
259                 month--;
260             }
261         }
262         else if(getMonth()==12 || getMonth()==10 || getMonth()==7 || getMonth()==5)
//going to 30 day months
263         {
264             day=30;
265             month--;
266         }
267         else //going to 31 day months
268         {
269             if(getMonth()==1)
270             {
271                 day=31;
272                 month=12;
273                 year--;
274             }
275             else
276             {
277                 day=31;
278                 month--;
279             }
280         }
281     }
282     else //some day in the month, not the first
283     {
284         day--;
285     }
286 }
287 }
```