Casey Sobecks

CS248

Assignment 3: The Dating Game

Terminal Output:

PS C:\Users\csobecks\OneDrive\Documents\Butler\(13) 2020 Spring\CS248\Assignment 3> java Year3000

The date is Saturday February 1, 2020

The date is Wednesday January 1, 3000

The date is Wednesday January 1, 1800

PS C:\Users\csobecks\OneDrive\Documents\Butler\(13) 2020 Spring\CS248\Assignment 3>

```java
/*
Interface for Date objects to be used by the Year 3000 driver program
@author Jon Sorenson
 */

public interface DateInterface
{
    /** @return the day of the month(1-31)*/
    public int getDay();
    /** @return the day of the week(0-6) */
    public int getDow();
    /** @return the month of the year(1-12) */
    public int getMonth();
    /** @return the year (four digits) */
    public int getYear();
    /** sets the date
     *  @param m the month of the year (1-12)
     *  @param d the day of the month (1-31)
     *  @param y the year (four digits)
     *  @param dow the day of the week (0-6) */
    public void set (int m, int d, int y, int dow);
    /** moves the date forward by exactly one day */
    public void tomorrow();
    /** @return the date as a String in the format "Monday March 18, 2002" */
    public String toString();
    /**sets the date to today;\
     * make this empty {} unless you do the extra credit.
     */
    public void today();
    /** Moves the date backword by exactly one day;
     *  make this empty {} unless you do the extra credit.
     */
    public void yesterday();
}
```

```java
import java.io.*;

/**
 * Driver class for The Dating Game programming assignment.
 * @author Jon Sorenson
 * editted by Casey Sobecks to include extra credit
 */

public class Year3000
{
    public static void main(String [] args) throws IOException
    {
        DateInterface d= new MyDate();
        d.today(); //sets the date to current date used by computer
        System.out.println(d); //prints the current day
        while(d.getYear()<3000) //moves forward until the day Jan 1 3000 is reached
        {
            d.tomorrow();
        }
        System.out.println(d); //print out the date and day of week of Jan 1, 3000

        d.today(); //reset the date to today
        while(d.getYear()>1799) //move backwards until Dec 31 1799 is reached
        {
            d.yesterday();
        }
        d.tomorrow(); //move forward a day to reach Jan 1, 1800
        System.out.println(d); //print out the day of the week of Jan 1, 1800

    }
}
```

```java
import java.util.*;
/*
This is the Class file for the Dating Game Assignment
CS248 Object Oriented Programming
@author Casey Sobecks
*/

public class MyDate implements DateInterface
{
    //data memers(variables)
    int month;
    int day;
    int year;
    int dayofweek;
    String dayname;
    String monthname;

    //constructor
    /** @param m month
     * @param d day
     * @param y year
     * @param dow day of the week */
    public void set (int m, int d, int y, int dow)
    {
        month=m; //months in 1-12
        day=d;  //day in 1-31
        year=y; //year four digits long
        dayofweek=dow; //day in 0-6
    }

    //accessors
    public int getDay()
    {
        return day; //returns the day
    }

    public int getDow()
    {
        return dayofweek; //returns the day of the week
    }

    public int getMonth()
    {
        return month; //returns the month
    }

    public int getYear()
    {
        return year; //returns the year
    }

    //mutators
    public void tomorrow()
    {
        //First checks the day of the week
        if(getDow()==6)
        {
            dayofweek=0;
        }
        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              }
94              else //not february
95              {
96                  day++;
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             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 }
```