```
/*
PROGRAM NAME: Clock
PROGRAMMER: Samuel Jentsch
CLASS: CSC 241.001
INSTRUCTOR: Dr. D. Dunn
DATE STARTED: 9/13/2013
DATE DUE: 9/18/2013

REFERENCES:
      Data Abstraction and Problem Solving with Java
            Janet J. Prichard & Frank M. Carrano
      Lab 2 Assignment Sheet
            Dr. Debra Dunn

PROGRAM PURPOSE:
  a. This class is designed to hold all of the data associated with
displaying a time using
      hours,  minutes, and seconds.
  b. The class provides operations allowing for the modification of the data
mentioned above.
  c. The class provides methods to display the time stored by the client in
either a 24 hour
      or 12 hour format. (HOURS:MINUTES:SECONDS)

VARIABLE DICTIONARY:
  -hours - int, holds the value of hours for use in the clock.
  -minutes - int, holds the value of minutes for use in the clock.
  -seconds - int, holds the valie of seconds for use in the clock.

Files Used:
  NONE.

Test Cases:
 1. Creating a Clock with a default constructor creates a Clock with an hours
value of 0,
      minutes value of 0, and a seconds value of 0.
 2. A clock with a value of 00:00:00 should have a value of 00:00:01 after
calling increment seconds.
 3. Calling setTime(11,12,13) should change the value of the Clock so that
when displayTime24hr() is called
            11:12:13 is printed.
 4. Calling addMinutes(80) on a Clock with initial value of 00:00:00, should
change the time to 1:20:00.
 5. Calling incrementSeconds() on a Clock with a value of 11:59:59 should
change the value to 12:00:00.

 */
```

```
public class Clock {

      //Private data fields
      private int hours;
      private int minutes;
      private int seconds;

      public Clock() {
            //-------------------------------------------------------------------
----------------------------------------
            //Constructor to initialize a clock object with a default value.
            //Preconditions: none.
            //Postconditions: The values of hours, minutes, and seconds are
initialized with the default value of 0.
            //-------------------------------------------------------------------
----------------------------------------

            this.setHours(0);
            this.setMinutes(0);
            this.setSeconds(0);

      }//default constructor

      public Clock(int hours, int minutes, int seconds) {
            //-------------------------------------------------------------------
-------------------------------------------
            //Constructor to initialize hours, minutes, and seconds with user
specified values passed as parameters.
            //Preconditions: Integer values passed as parameters for hours,
minutes, and seconds. Where:
            //          0 <= hours <= 12, 0 <= minutes <= 60; 0 <= seconds <=
60
            //Postconditions: The values for hours, minutes, and seconds are
set to the user specified values. If the
            //                values passed to not meet the requirements, the
values are set to a default value of 0.
            //-------------------------------------------------------------------
-------------------------------------------

            this.setHours(hours);
            this.setMinutes(minutes);
            this.setSeconds(seconds);

      }//constructor with specified values

      private int getHours() {
```

```
		//-------------------------------------------------------------
		//Preconditions: none.
		//Postconditions: returns the value for the data field hours.
		//-------------------------------------------------------------

		return this.hours;

	}//getHours

	private void setHours(int hours) {
		//-----------------------------------------------------------------
-----
		//Precondition: int value passed as parameter. Where:
		//          0 <= hours <= 24
		//Postcondition: The value for the class datafield hours is
		//                 changed to the parameter passed to the method.
		//-----------------------------------------------------------------
------
		if(0 <= hours && hours <= 24)
			this.hours = hours;

	}//setHours

	private int getMinutes() {
		//---------------------------------------------------------------
		//Preconditions: none.
		//Postconditions: returns the value for the data field minutes.
		//---------------------------------------------------------------

		return this.minutes;

	}//getMinutes

	private void setMinutes(int minutes) {
		//----------------------------------------------------------------
		//Precondition: int value passed as parameter. Where:
		//           0 <= minutes  <= 60
		//Postcondition: The value for the class datafield minutes is
		                 changed to the parameter passed to the method.
		//----------------------------------------------------------------

		if(0 <= minutes && minutes <= 60)
			this.minutes = minutes;
	}//setMinutes

	private int getSeconds() {
		//---------------------------------------------------------------
		//Preconditions: none.
		//Postconditions: returns the value for the data field seconds.
		//---------------------------------------------------------------

		return this.seconds;

	}//getMinutes

	private void setSeconds(int seconds) {
		//-----------------------------------------------------------------
		//Precondition: int value passed as parameter. Where:
		//	    0 <= minutes  <= 60
		//Postcondition: The value for the class datafield seconds is
		//		    changed to the parameter passed to the method.
		//-----------------------------------------------------------------

		if(0 <= seconds && seconds  <= 60)
			this.seconds = seconds;

	}//setMinutes

	public void setTime(int hours, int minutes, int seconds) {
		//-----------------------------------------------------------------
-----------------------------------
		//Method to set hours, minutes, and seconds with user specified
values passed as parameters.
		//Preconditions: Integer values passed as parameters for hours,
minutes, and seconds. Where:
		//	    0 <= hours <= 12, 0 <= minutes <= 60; 0 <= seconds
<= 60
		//Postconditions: The values for hours, minutes, and seconds are
set to the user specified values.
		//		If the values passed do not meet the
requirements, the values are left unchanged.
		//-----------------------------------------------------------------
-------------------------------------

		this.setHours(hours);
		this.setMinutes(minutes);
		this.setSeconds(seconds);

	}//setTime

	public void incrementSeconds() {
		//-----------------------------------------------------------------
-----------------------------
		//Method to increment the seconds value by one. The values for
```

```
hours and minutes are updated
            //appropriately based on new seconds value.
            //If the seconds >= 60 after being incremented, the minutes value
is incremented and the
            //seconds value is set to 0.
            //If the minutes value is >= 60 after being incremented, the
hours value is incremented by one
            //and the minutes value is set to 0.
            //Preconditions: hours, minutes, seconds each have value
            //Postconditions: The seconds value is incremented by one. If
necessary, the minutes and hours
            //                values are incremented by one as dictated by
the method description above.
            //--------------------------------------------------------------
------------------------------
            if(this.getSeconds() + 1 != 60) {
                  this.setSeconds(this.getSeconds() + 1);
            }//end if
            else if(this.getMinutes() + 1 != 60) {
                  this.setSeconds(0);
                  this.setMinutes(this.getMinutes() + 1);
            } //end else if
            else if(this.getHours() + 1 != 25) {
                  this.setSeconds(0);
                  this.setMinutes(0);
                  this.setHours(getHours() + 1);
            }//end else if
            else {
                  setSeconds(0);
                  setMinutes(0);
                  setHours(1);
            }

      }//incrementSeconds

      public void addMinutes(int minutes) {
            //--------------------------------------------------------------
-----------------------------------------
            //Method to add the integer value passed as a parameter to the
minutes value.
            //The value for hours is updated appropriately based on the new
minutes value after the addition.
            //If the minutes value is >= 60 after being incremented, the
hours value is incremented by minutes/60
            //and the minutes value is set to the remaining amount of
minutes.
            //Preconditions: hours, minutes, seconds each have a value. The
```

```
minutes parameter passed must be positive.
            //Postconditions: The minutes value parameter is added to minutes 
value for Clock. If necessary, the hours
            //                value is incremented as dictated by the method 
description above
            //-----------------------------------------------------------------
------------------------------------------

            int newMinutesValue = this.getMinutes() + minutes;

            int hoursFromMinutes = newMinutesValue / 60;
            int minutesRemainder = newMinutesValue % 60;

            this.setMinutes(minutesRemainder);

            //Use a loop to add amount of hours to the Clock.
            for(int i = 0; i < hoursFromMinutes; i++) {
                  if(getHours() + 1 != 25) {
                        setHours(getHours() + 1);
                  }
                  else {
                        setHours(1);
                  }
            }//for loop

      }//addMinutes

      public static int findDifference(Clock c1, Clock c2) {
            //-----------------------------------------------------------------
----------------------------------
            //Calculates the difference between the times of two Clock 
objects passed as parameters.
            //The difference is returned in seconds as the absolute value of 
the difference between the times.
            //Preconditions: Two initialized Clock objects passed as 
parameters.
            //Postconditions: The absolute value of the difference between 
the two times in seconds is returned
            //                as an integer.
            //-----------------------------------------------------------------
-----------------------------------

            int hoursDifference = c1.getHours() - c2.getHours();
            int minutesDifference = c1.getMinutes() - c2.getMinutes();
            int secondsDifference = c1.getSeconds() - c2.getSeconds();

            int difference = (hoursDifference * 60 * 60) + (minutesDifference
* 60) + secondsDifference;

            //set difference to absolute value
            if(difference < 0)
                  difference *= -1;

            return difference;

      }//end findDifference

      public void displayTime24hr() {
            //----------------------------------------------------------------
------------------------------
            //Displays the values stored in hours, minutes, and seconds in a
24 hour time format 00:00:00.
            //If the value for one of the three values is less than 10, a 0
is appended to the time string.
            //Preconditions: The data fields hours, minutes, and seconds must
be positive and have values.
            //Postconditions: The values for hours, minutes, and seconds are
printed to the console in a
            //                24 hour time format.
            //----------------------------------------------------------------
-------------------------------

            String time = "";

            if(hours < 10)
                  time += "0";
            time += "" + hours + ":";

            if(minutes < 10)
                  time += "0";
            time += "" + minutes + ":";

            if(seconds < 10)
                  time += "0";
            time += "" + seconds;

            System.out.print(time);

      }//displayTime24hr

      public void displayTime12hr() {
            //----------------------------------------------------------------
-------------------------------
            //Displays the values stored in hours, minutes, and seconds in a
```

```
12 hour time format 00:00:00.
            //If the value for one of the three values is less than 10, a 0
is appended to the time string.
            // If the value for hours is greater than 12, 12 is subtracted
from hours and pm is appended to
            //the end of the time string. If the value for hours is less than
12, hours is left unaltered
            // and am is appended to the end of the time string.
            //Preconditions: The data fields hours, minutes, and seconds must
be positive and have values.
            //Postconditions: The values for hours, minutes, and seconds are
printed to the console in a 1
            //                  2 hour time format.
            //----------------------------------------------------------------
-------------------------------

            String time = "";

            boolean isAM = true;

            int printHours = this.getHours();
            if(printHours < 10) {
                  time += "0";
            }
            else if(printHours > 12) {
                  isAM = false;
                  printHours -= 12;
            }

            time += "" + printHours + ":";

            if(this.getMinutes() < 10)
                  time += "0";
            time += "" + this.getMinutes() + ":";

            if(this.getSeconds() < 10)
                  time += "0";
            time += "" + this.getSeconds();

            if(isAM) {
                  time += " AM";
            }
            else {
                  time += " PM";
            }

            System.out.print(time);
```

```
    }//displayTime12hr

}//class
```