CS 1301 Programming Assignment#11

3/27/2017 Pair Programming

Fares

Problem Description:

(*The Time class*) Design a class named *Time* that contains:

* The data fields hour, minute, and second that represent a time.
* A no-argument constructor that creates a Time object for the current time. (The values of the data fields will represent the current time.)
* A constructor that constructs a Time object with a specified elapsed time since midnight, January 1, 1970, in milliseconds. (The values of the data fields will represent this time.)
* A constructor that constructs a Time object with the specified hour, minute, and second.
* Three getter methods for the data fields hour, minute, and second, respectively.
* A method named setTime(long elapsedTime) that sets a new time for the object using the elapsed time. For example, if the elapsed time is 555550000 milliseconds, the hour is 10, the minute is 19, and the second is 10.
* A no-argument method named incrementSecond that accepts no argument and increase the data field second by 1. If second becomes 60, it sets second to 0 and increase minute by 1.
* An overloaded method named incrementSecond that accepts an integer value (should be less than 60) for the number of seconds to be added. If second becomes >=60, it sets second to (second + n) – 60 and increase minute by 1.
* A method named incrementMinute that accepts no argument and increase the data field minute by 1. If minute becomes 60, it sets minute to 0 and increase hour by 1.
* An overloaded method named incrementMinute that accepts an integer value (should be less than 60) for the number of minutes to be added. If minute becomes >=60, it sets minute to (minute + n) – 60 and increase hour by 1.
* A method named incrementHour that accepts no argument and increase the data field hour by 1. If hour becomes 24, it sets hour to 0.
* An overloaded method named incrementHour that accepts an integer value (should be less than 24) for the number of hours to be added. If hour becomes >=24, it sets hour to (hour + n) – 24 of next day. Otherwise, it increments hour by n.

Draw the UML diagram for the class and then implement the class.

Note:

1. Understand the attached TimeSkeleton and TimeSkeleton\_Tester classes.
2. Modify as specified in this handout.
3. Use the attached driver to test you code.

Attach (do not turn in any hardcopy) via BlazeView the following items:

* 1. Each team member should submit a single typed page with your name, class, date, and program title. The report should include what you learned from the programming assignment, problems faced, skills learned, and your observations.
  2. Simple output
  3. Do not modify or submit the driver.
  4. Each team should submit a single copy of the code for each team.
  5. Each team should submit a single copy of a sample output
  6. Zip and attach all your files
  7. Make sure that:
     1. The program is well documented and readable.
     2. The output is well labeled and aligned

Sample output:

time1 Original values: Hour: 10 Minute: 19 Second: 10

time2 Original values: Hour: 10 Minute: 19 Second: 10

Added values:

Hour + 0 Minute + 0 Second + 1

Hour: 10 Minute: 19 Second: 11

Added values:

Hour + 0 Minute + 1 Second + 0

Hour: 10 Minute: 20 Second: 11

Added values:

Hour + 1 Minute + 0 Second + 0

Hour: 11 Minute: 20 Second: 11

Added values:

Hour + 0 Minute + 0 Second + 13

Hour: 11 Minute: 20 Second: 24

Added values:

Hour + 0 Minute + 50 Second + 0

Hour: 12 Minute: 10 Second: 24

Added values:

Hour + 13 Minute + 0 Second + 0

Hour: 1 Minute: 10 Second: 24

Added values:

Hour + 10 Minute + 50 Second + 13

Hour: 12 Minute: 0 Second: 37