Inspiring Greatness
ISD
INDEPENDENCE SCHOOL DISTRICT

Computer Science Virtual Learning

HS Computer Science A

May 1st, 2020

Lesson: Free Response Friday

Objective/Learning Target:

Students will apply what they’ve learned this far in order to write code in response to an open ended free response question

Free Response Question

You will implement two unrelated methods for a `Time` class that keeps track of the time using a 24 hour clock. Consider the code for the `Time` class provided on the next slide

```
/**
 * Objects of the Time class hold a time value for
 * a European-style 24 hour clock.
 * The value consists of hours, minutes and seconds.
 * The range of the value is 00:00:00 (midnight)
 * to 23:59:59 (one second before midnight).
 */
public class Time
{
   // The values of the three parts of the time
   private int hours;
   private int minutes;
   private int seconds;
   /**
    * Creates a new Time object set to 00:00:00
    * Do not change this constructor.
    */
   public Time()
   {
      this.hours = 0;
      this.minutes = 0;
      this.seconds = 0;
   }
   /**
    * Constructor for objects of class Time.
    * Creates a new Time object set to h:m:s.
    * Assumes, without checking, that the parameter
values are
    * within bounds.
    * For this task, you don't need to worry about
invalid parameter values.
    * Do not change this constructor.
    */
```

```
   public Time(int h, int m, int s)
   {
      this.hours = h;
      this.minutes = m;
      this.seconds = s;
   }
   /**
    * Add one second to the current time.
    * When the seconds value reaches 60, it rolls over
to zero.
    * When the seconds roll over to zero, the minutes
advance.
    * So 00:00:59 rolls over to 00:01:00.
    * When the minutes reach 60, they roll over and the
hours advance.
    * So 00:59:59 rolls over to 01:00:00.
    * When the hours reach 24, they roll over to zero.
    * So 23:59:59 rolls over to 00:00:00.
    */
   public void tick()
   {
      // Part a: complete the tick() method
   }
   /**
    * Add an offset to this Time.
    * Rolls over the hours, minutes and seconds fields
when needed.
    */
   public void add(Time offset)
   {
        // Part b: complete the add method
   }
```

```
  public String toString()
   {
       return pad(hours) + ":" +
pad(minutes) + ":" + pad(seconds);
   }

    /**
     * Returns a string representing
the argument value, adding a leading
     * "0" if needed to make it at
least two digits long.
     * Do not change this.
     */
   private String pad(int value)
   {
      String sign = "";
      if (value < 0)
      {
         sign = "-";
         value = -value;
       }
       if (value < 10) {
         return sign + "0" + value;
       } else {
         return sign + value;
       }
    }
  }
```

Part A.

Write the method `tick` which increases the number of seconds by one. If the number of seconds is 60 it adds one to the number of minutes and resets seconds to 0. If the number of minutes is 59 it adds one to the number of hours and resets the number of minutes to 0. If the number of hours reaches 24 it should be reset to 0.

Part A (Hint)

The first thing to do is try to solve the examples by hand. The question tells us that when the value of minutes is 0, and seconds is 59 the method tick should result in minutes = 1 and seconds = 0. When the value of minutes is 59 and the value of seconds is also 59 and the method tick is called the number of hours should increase and the minutes reset to 0. If the number of hours reaches 24 it should be reset to 0.

Use conditionals (if statements) to check for each of these conditions and take the appropriate actions when each condition is true.

Part B

Write the method `add(Time offset)` which adds the seconds together and makes sure the result is 59 or less (incrementing the minutes as needed), adds the minutes together and makes sure the result is 59 or less (increments the hours as needed), and adds the hours together (resetting the hours to 0 if it reaches 24). When you have finished writing the method, click “Run” to test your solution. The main method has code that will test your solution using several different times.

For More Resources and to Check Answers

Go to: https://runestone.academy/runestone/books/published/apcsareview/Conditionals/timeFRQ.html