Problems Javadoc Declaration Console

```
runner [Java Application] C:\Users\Nick Kokott\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre
Hello user! Please enter the following data.
Would you like to reset the system (0 for yes and 1 for no)
1
What is your temperature reading?
67
What is your humidity reading?
45
For Relative Humidity:
 1. Current Humidity: 45%
 2. Max Humidity: 45%
3. Minimum Humidity: 45%
 4. Humidity Trend: Stable
 5. Humidity Check: OK
 For Temperature:
 1. Current Temperature: 67 degrees.
 2. Maximum Temp: 67 degrees
3. Minumum Temp: 67 degrees.
 4. Temperature Trend: Stable
```

First reading

Problems Javadoc Declaration Console

```
runner [Java Application] C:\Users\Nick Kokott\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v20211116-1657\jre\bin\javaw.exe  (Mar 2
3. Minumum Temp: 67 degrees.
 4. Temperature Trend: Stable
Hello user! Please enter the following data.
Would you like to reset the system (0 for yes and 1 for no)
1
What is your temperature reading?
85
What is your humidity reading?
43
For Relative Humidity:
 1. Current Humidity: 43%
 2. Max Humidity: 45%
3. Minimum Humidity: 43%
 4. Humidity Trend: Down
 5. Humidity Check: OK
 For Temperature:
 1. Current Temperature: 85 degrees.
 2. Maximum Temp: 85 degrees
3. Minumum Temp: 67 degrees.
 4. Temperature Trend: Up
```

Second reading

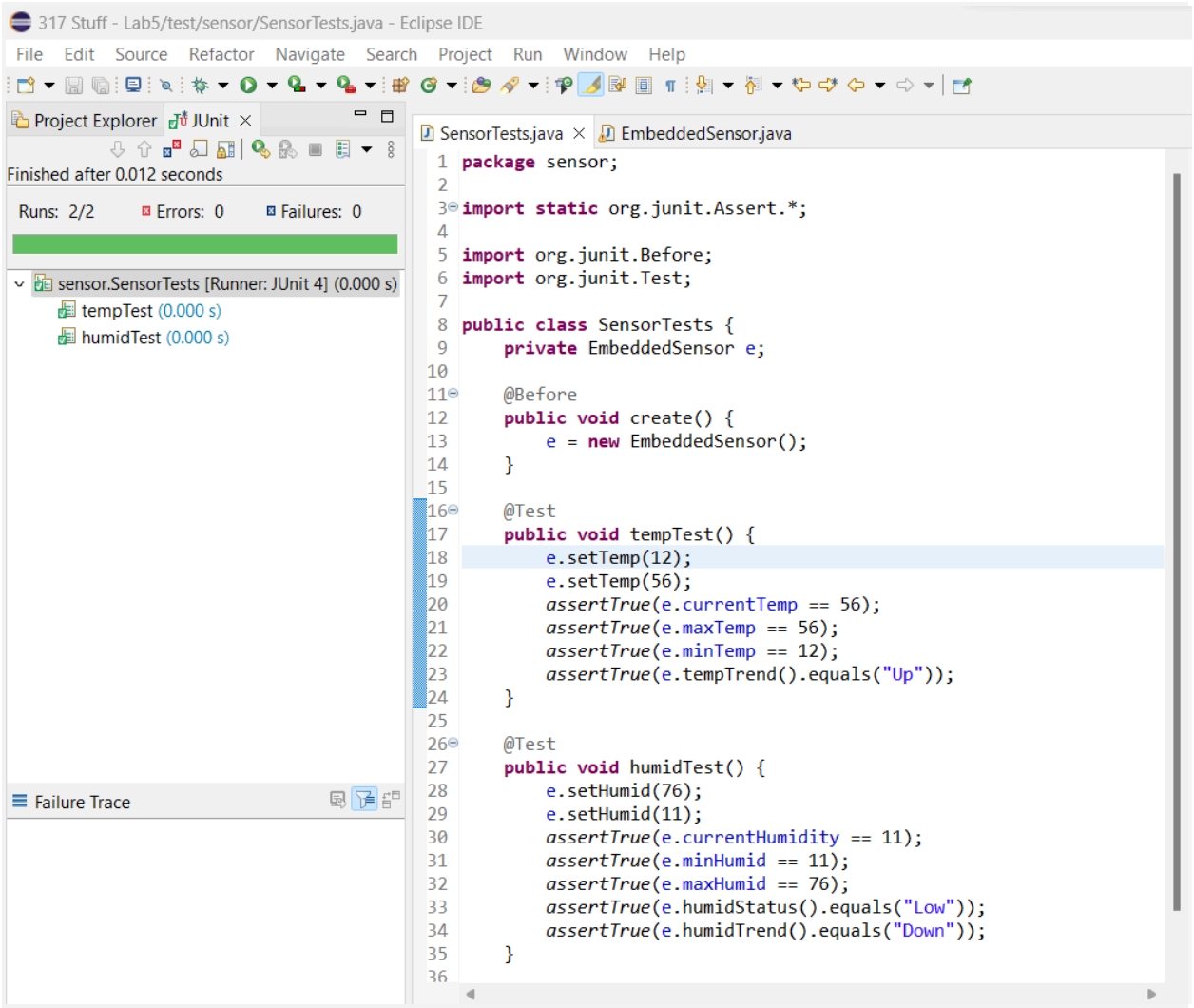
```
3. Minumum Temp: 67 degrees.
 4. Temperature Trend: Up
Hello user! Please enter the following data.
Would you like to reset the system (0 for yes and 1 for no)
1
What is your temperature reading?
42
What is your humidity reading?
89
For Relative Humidity:
 1. Current Humidity: 89%
 2. Max Humidity: 89%
3. Minimum Humidity: 43%
 4. Humidity Trend: Up
 5. Humidity Check: High
 For Temperature:
 1. Current Temperature: 42 degrees.
 2. Maximum Temp: 85 degrees
3. Minumum Temp: 42 degrees.
 4. Temperature Trend: Down
```

Third reading
Question a)
Design at least two test cases (using assertion) to test each of the output values separately

Hint: To test trend, max, and min, you need to input at least two values for each test before you check the output.

Each test will include test code (script) with input values or input sequences as needed for each test

Run your tests and take a screenshot of each one of them.



Submit your test scripts.

Question b) Data-Driven Testing to avoid test code bloating:

Hint: Review chapter 3 slides of the textbook, the “adding two numbers” example”, @Parameters

Write the following tests using the given values in two different testing styles, style 1 & style 2:

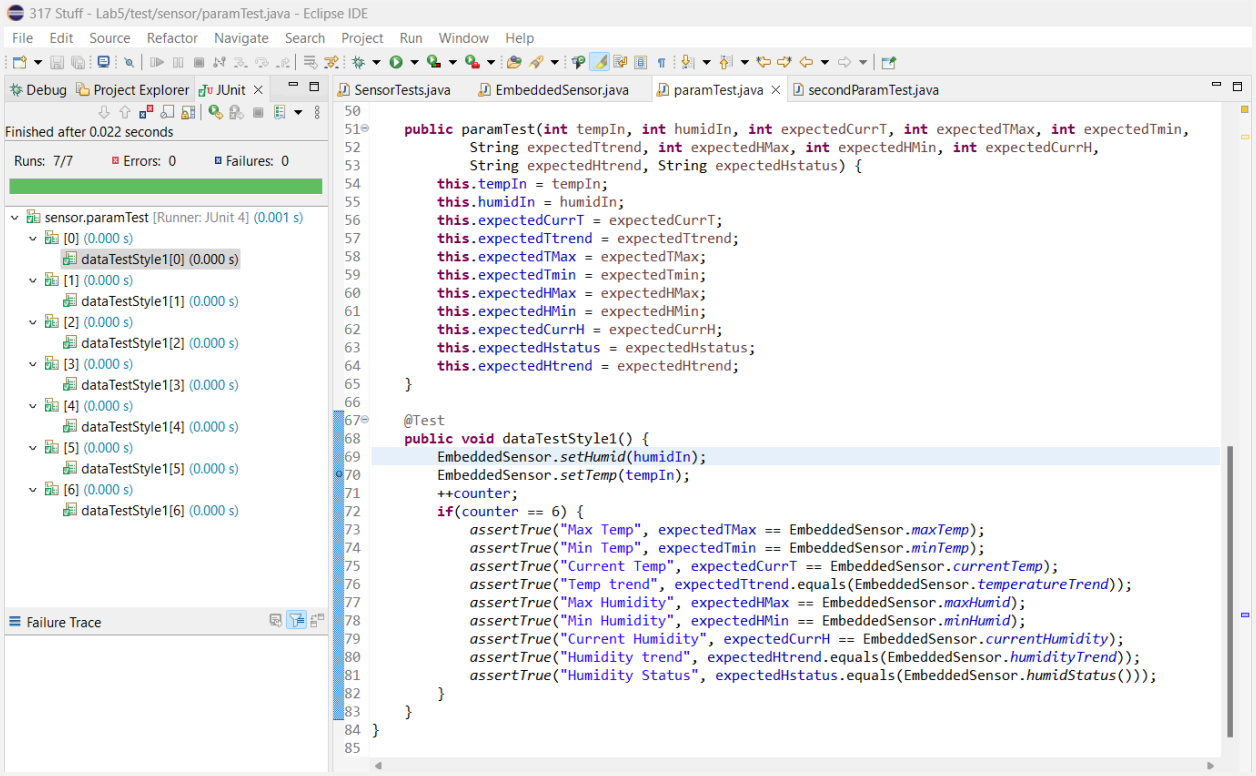
Style 1:
One sequence of seven “Temperature, Humidity” pairs, followed by ONE test at the end of the given sequence (i.e. one test for the entire sequence of the seven input pairs)

Style 2:
One pair of “Temperature, Humidity” input followed by a test, repeat seven times (i.e. seven independent tests: One test for each pair of input values).

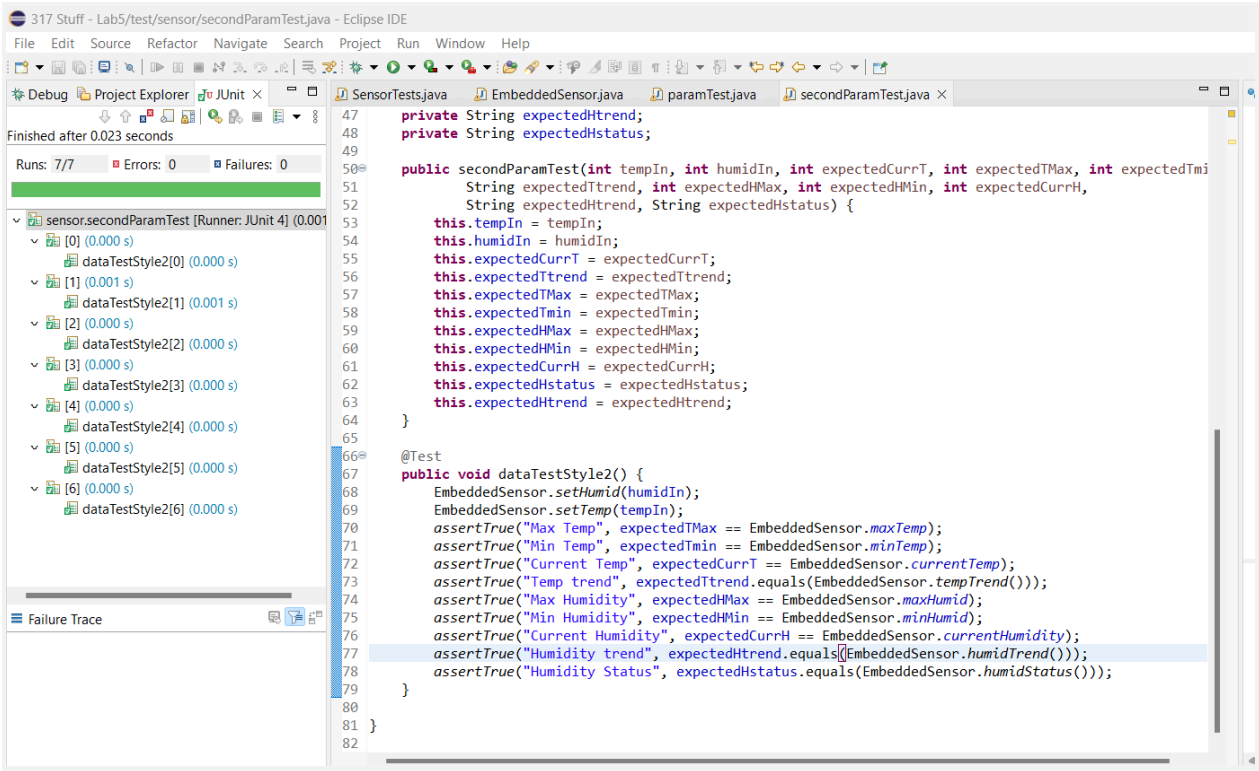
I. Use the following temperature sequence after a reset: 66, 68, 69, 67, 63, 59, 53
(Use any arbitrary value for the corresponding humidity readings)

Submit your test scripts for both style 1 & style 2

Take a screenshot of the output values after each test is run (style 1 & 2) and include them in your deliverable



This is style 1



This is style 2

ii.
Use the following relative humidity sequence (in %) after a reset: 53, 51, 48, 49, 54, 56, 56

(Use any arbitrary value for the corresponding temperature readings)

Submit your test scripts for both style 1 & style 2

Take a screenshot of the output values after each test is run (style 1 & 2) and include them in your deliverable.

Question c) Refactoring:
You will notice that the algorithms used for humidity and temperature are similar (for calculating the max, the min, and the trend).

Refactor your code (if needed) to use one algorithm that could be used in both humidity and temperature calculations.

```java
    //My refactoring for the inputs and calculation of the min, max, and trend of both temperature and humidity
    public void doAllTempandHumidCalculations(int temp, int humid) {
        //Temperature calculations happen in this chunk for min and max.
        prevCurrTemp = currentTemp;
        currentTemp = temp;
        if(started == 0) {
            prevCurrTemp = temp;
        }
        if(temp > maxTemp || maxTemp == 0) {
            maxTemp = temp;
        }if(temp < minTemp || minTemp == 0) {
            minTemp = temp;
        }
        //Humidity calculations happen in this chunk for min and max.
        prevCurrHumid = currentHumidity;
        currentHumidity = humid;
        if(started == 0) {
            prevCurrHumid = humid;
        }
        if(humid > maxHumid || maxHumid == 0) {
            maxHumid = humid;
        }if(humid < minHumid || minHumid == 0) {
            minHumid = humid;
        }
        started = 1;
        //The trend calculations begin here
        //This is the humidity trend calculation
        if(currentHumidity < 55 && currentHumidity > 25) {
            humidityTrend = "OK";
        }else if(currentHumidity >= 55) {
            humidityTrend = "High";
        }else {
            humidityTrend = "Low";
        }
        //This is the temperatureTrend calculation
        if(currentTemp > prevCurrTemp) {
            temperatureTrend = "Up";
        }else if(currentTemp < prevCurrTemp) {
            temperatureTrend = "Down";
        }else {
            temperatureTrend = "Stable";
        }
    }
```

I've refactored my setting methods and my trend calculation methods into one major function that does all the calculations. It takes in the temperature and the humidity to be set and used, and from there it starts by finding the temperature calculations of max and min. It then moves onto humidity calculations for max and min. The function then moves onto trend calculations for humidity and temperature respectively. This makes it so that all functions are in one function.

Part 2 Deliverables:

i)
Submit your test scripts as explained above

ii)
Include one screenshot of the 9 output values after each of the two sequences (question a) are read by the system in both

styles 1 & 2

iii)
Submit your refactored code (with comments)

iv)
Answer the following question:

a. What is the difference between testing the 7 inputs in a sequence and testing them individually. How are the two test cases designed? (use narrative description, no test code needed.)

The difference between testing after all 7 and one by one, is that with the 7 in a row, we test the end result after all elements are added to the sensor. With the ability to test one by one we get to test the values after every change. This allows us to possibly determine if there was an issue within the internal state at one of the entries. Potentially with the 7 in a row and one test we may not be able to actually see the fault happen, but when testing one by one we are able to see any potential faults that can occur throughout all of the values being inserted.

b.As the same person who developed, refactored, and tested the code, does your refactored code make it easier or harder to test the system, explain with examples.

It really doesn't make a difference, I still end up needing the same number of lines for each test. For example prior to my refactorization I simply called one set of either temperature or humidity and from there I would get the rest of the data through function calls. Now I get that data just by calling the public variable. It really didn't change all that much.

C. Does your refactored code parametrize your tests ? Explain.

It does a little more, instead of having just one parameter pass through into each setting function it's now required that you pass both a temperature and a humidity into the one method. before this I had two separate setting methods that used just temperature or humidity, now the one function uses both of them as parameters. So overall it does end up making my functions slightly more parameterized.

D. If you received the refactored code (written by another developer) to just test it, would it be easier or harder than case iv) above?

I think it would be harder as I would have to figure out what parameters are being passed into the function. I already know what will need to be passed into the function since I wrote the code, with another developer I would need to analyze the code in order to figure out how their code is called and what should be expected.

E. Would you prefer to test the original code or the refactored code, if both were written by another developer?

Yes I would prefer to test both if they were written by another developer. I think that when I test my own code that I write I end up testing it a certain way where I don't consider all of the criteria the best way because I want my code to be perfect. When I am testing other code I am not timid when looking for new errors.