November 14, 2017

Deliverable #4 Completed Automated Testing Framework

1. EXPERIENCES

A. For this deliverable, our primary goal was to make any changes that were requested by the customer and to continue testing in earnest. We were told to update our script to use relative class paths instead of absolute class paths, in order to ensure that it would work on a different machine. We updated our HTML output file to include a table to display our output data horizontally instead of being text based and vertical. This will increase readability of our testing framework if we had a larger test case base. From Deliverable 3, we had compiled our Drivers using a jar file for our imports and dependencies, this is something we were asked to fix. To go from a jar file as an import, we just had to change the code in our script to make this change. At first it was giving us a hard time but we eventually figured it out.

2. TESTING FRAMEWORK CHANGES

A. We changed class path dependencies to be relative so that they will run successfully on other machines. We finalized our test report outputs, the html file, to show our outputs in a table to make it more readable. That way if we added say, ten thousand more tests, it would be easy to read through them all in a single html file, rather than have them listed vertically and almost impossible to search through.

3. TEST CASE SAMPLES

A. We did not change our test case format. Our last deliverable showed our first five test cases, all of which were performed on one method inside of one class. Here are a few other test cases that are shown to test a separate method in the NumberUtils class. We now have 25 test cases for our testing framework. The test case format has been left below to aid in understanding each sample test case:

i. <test case ID>
ii. <description of the method's purpose>
iii. <class file name *.java>
iv. <method name with any parameters>
v. <input>
vi. <expected output>

B. Test Case #6

i. Test #0006
ii. Returns a ratio from a number and decimal ratio.
iii. NumberUtils.java
iv. ratio(Number n, Number in)
v. 2,0.0
vi. 0.0

C. Test Case #10

i. Test #010
ii. Returns a ratio from a number and decimal ratio.
iii. NumberUtils.java
iv. ratio(Number n, Number in)
v. 1,1.5
vi. 66.66

D. Test Case #13

i. Test #013
ii. Returns ratio as a String percentage.
iii. NumberUtils.java
iv. ratioAsString(Number n, Number in)
v. 2,0.5
vi. 400.0 %

E. Test Case #21

i. Test #021
ii. Returns a truncated number.
iii. NumberUtils.java
iv. truncate(Number n, int decimals)
v. 2.1234,2
vi. 2.12