Jon Cucci

SSW567 HW #2 - https://github.com/joncucci/Triangle567

9/18/21

Assignment Description: The objective of this assignment is for you to (a) develop a set of tests for an existing triangle classification program, (b) use those tests to find and fix defects in that program, and (c) report on your testing results for the Triangle problem.

Author: Jonathan Cucci

Summary:

	Test Run 1	Test Run 2	Test Run 3	Test Run 4
Tests Planned	3	8	8	8
Tests Executed	3	8	8	8
Tests Passed	0	3	2	8
Defects Found	3	5	6	0
Defects Fixed	0	0	0	6

My results worked out about the way I expected it to, although slightly different. I knew that there were errors in the code, however if you look at test run 2 and test run 3, I had one more failed test. Pragmatically, I was fixing bugs, however one of the bugs was essentially a False Positive (where it seems like it passed, but not for the right reasons), I took one step back to take 2 steps forward. After I kept debugging, I was able to debug all issues and have all 8 of my tests pass thoroughly.

This was a valuable experience; I had done some testing in the past but never kept track or went into as much detail. I learned, like I said before, sometimes you need to take one step back to take two step forwards, and even though something may pass, it may not be for the intended reasons, so you should always double check your code.

Honor Pledge: I pledge my honor that I have abided by the Stevens Honor System

Details: My method of dealing with this assignment, was basically to make sure to weed out any edge cases possible. I did this by following the codes input field requirements, as well as making sure I had an edge case to fall under each individual “if” statement.

See pictures below for screenshots of test run 2 and test run 4, respectively.

Test Runs 2 & 4

Test Run 2 (2 passes, 6 fails)

Test ID	Input	Expected Results	Actual Result	Pass or Fail
testEquilateralTriangles	1, 1, 1	Equilateral	InvalidInput	Fail
testInvalidInput1	-3, -4, -5	InvalidInput	InvalidInput	Pass
testInvalidInput3	1000, 1000, 1000	InvalidInput	InvalidInput	Pass
testIsoscelesTriangles	5, 5, 6	Isosceles	InvalidInput	Fail
testNotATriangle	1, 1, 100	NotATriangle	InvalidInput	Fail
testRightTriangleA	3, 4, 5	Right	InvalidInput	Fail
testRightTriangleB	5, 3, 4	Right	InvalidInput	Fail
testScaleneTriangles	2, 3, 4	Scalene	InvalidInput	Fail

Test Run 4 (8 passes, 0 fails)

Test ID	Input	Expected Results	Actual Result	Pass or Fail
testEquilateralTriangles	1, 1, 1	Equilateral	Equilateral	Pass
testInvalidInput1	-3, -4, -5	InvalidInput	InvalidInput	Pass
testInvalidInput2	1000, 1000, 1000	InvalidInput	InvalidInput	Pass
testIsocelesTriangles	5, 5, 6	Isoceles	Isoceles	Pass
testNotATriangle	1, 1, 100	NotATriangle	NotATriangle	Pass
testRightTriangleA	3, 4, 5	Right	Right	Pass
testRightTriangleB	5, 3, 4	Right	Right	Pass
testScaleneTriangles	2, 3, 4	Scalene	Scalene	Pass