

Homework 5a: Static Code Analysis

Author: Syed Zain Raza

Description:

The objective of this assignment is to apply the techniques from the lecture to static testing of your Triangles program. Specifically:

- You will run a static code analyzer on your code, e.g. Pylint, identify and fix any problems reported by the static code analyzer;
- You will run a code coverage tool on your code, e.g. Coverage.py, and extend your test cases to demonstrate at least 80% code coverage;

Github Link:

https://github.com/ZainRaza14/software_Testing/tree/master/classifyingTriangle_2

Static Code Analyzer Tool: Pylint

Code Coverage Tool: Coverage

Old Test Cases:

```
def testRightTriangleA(self):
```

```
    self.assertEqual(classifyTriangle(3,4,5),'Right','3,4,5 is a Right triangle')
```

```
def testRightTriangleB(self):
```

```
    self.assertEqual(classifyTriangle(5,3,4),'Right','5,3,4 is a Right triangle')
```

```
def testEquilateralTriangles(self):
```

```
    self.assertEqual(classifyTriangle(1,1,1),'Equilateral','1,1,1 should be equilateral')
```

New Test Cases Added:

```
def testnotaTriangle(self):
```

```
self.assertEqual(classifyTriangle(9,4,4),'NotATriangle','9,4,4 should not be a triangle')
```

```
def testScaleneTriangle(self):
```

```
self.assertEqual(classifyTriangle(9,5,6),'Scalene','9,5,6 should be equilateral')
```

```
def testIsocelesTriangle(self):
```

```
self.assertEqual(classifyTriangle(9,5,5),'Isoceles','9,5,5 should be equilateral')
```

Pylint before changes:

```
(base) C:\Users\Syed Zain Raza\Documents\Software Testing\assig2>pylint Triangle
***** Module Triangle
Triangle.py:12:22: C0326: Exactly one space required after comma
def classifyTriangle(a,b,c):
                    ^ (bad-whitespace)
Triangle.py:12:24: C0326: Exactly one space required after comma
def classifyTriangle(a,b,c):
                    ^ (bad-whitespace)
Triangle.py:14:91: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:15:19: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:16:0: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:19:0: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:26:0: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:33:0: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:36:0: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:37:43: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:39:23: C0326: Exactly one space required after comma
    if not(isinstance(a,int) and isinstance(b,int) and isinstance(c,int)):
                ^ (bad-whitespace)
Triangle.py:39:45: C0326: Exactly one space required after comma
    if not(isinstance(a,int) and isinstance(b,int) and isinstance(c,int)):
                ^ (bad-whitespace)
Triangle.py:39:67: C0326: Exactly one space required after comma
    if not(isinstance(a,int) and isinstance(b,int) and isinstance(c,int)):
                ^ (bad-whitespace)
Triangle.py:40:0: W0301: Unnecessary semicolon (unnecessary-semicolon)
Triangle.py:41:0: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:42:59: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:48:0: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:49:47: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:1:0: C0103: Module name "Triangle" doesn't conform to snake_case naming style (invalid-name)
Triangle.py:12:0: C0103: Function name "classifyTriangle" doesn't conform to snake_case naming style (invalid-name)
Triangle.py:12:0: C0103: Argument name "a" doesn't conform to snake_case naming style (invalid-name)
Triangle.py:12:0: C0103: Argument name "b" doesn't conform to snake_case naming style (invalid-name)
Triangle.py:12:0: C0103: Argument name "c" doesn't conform to snake_case naming style (invalid-name)
Triangle.py:34:17: R0124: Redundant comparison - b <= b (comparison-with-itself)
Triangle.py:50:4: R1705: Unnecessary "elif" after "return" (no-else-return)
Triangle.py:54:9: R1714: Consider merging these comparisons with "in" to 'b not in (a, c)' (consider-using-in)
Triangle.py:12:0: R0911: Too many return statements (8/6) (too-many-return-statements)

-----
Your code has been rated at -6.88/10
```

Pylint after changes:

Anaconda Prompt

```
(base) C:\Users\Syed Zain Raza\Documents\Software Testing\assig2>pylint Triangle
***** Module Triangle
Triangle.py:12:22: C0326: Exactly one space required after comma
def classifyTriangle(a,b,c):
    ^ (bad-whitespace)
Triangle.py:12:24: C0326: Exactly one space required after comma
def classifyTriangle(a,b,c):
    ^ (bad-whitespace)
Triangle.py:14:91: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:15:19: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:16:0: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:19:0: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:26:0: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:33:0: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:36:0: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:37:43: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:41:0: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:42:59: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:48:0: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:49:47: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:52:0: C0301: Line too long (117/100) (line-too-long)
Triangle.py:1:0: C0103: Module name "Triangle" doesn't conform to snake_case naming style (invalid-name)
Triangle.py:12:0: C0103: Function name "classifyTriangle" doesn't conform to snake_case naming style (invalid-name)
Triangle.py:12:0: C0103: Argument name "a" doesn't conform to snake_case naming style (invalid-name)
Triangle.py:12:0: C0103: Argument name "b" doesn't conform to snake_case naming style (invalid-name)
Triangle.py:12:0: C0103: Argument name "c" doesn't conform to snake_case naming style (invalid-name)
Triangle.py:50:4: R1705: Unnecessary "elif" after "return" (no-else-return)
Triangle.py:12:0: R0911: Too many return statements (8/6) (too-many-return-statements)

-----
Your code has been rated at -3.75/10 (previous run: -5.62/10, +1.88)
```

Code coverage before changes:

```

(base) C:\Users\Syed Zain Raza\Documents\Software Testing\assig2>coverage run TestTriangle.py
Running unit tests
FFF
=====
FAIL: testEquilateralTriangles (__main__.TestTriangles)
-----
Traceback (most recent call last):
  File "TestTriangle.py", line 27, in testEquilateralTriangles
    self.assertEqual(classifyTriangle(1,1,1),'Equilateral','1,1,1 should be equilateral')
AssertionError: 'InvalidInput' != 'Equilateral'
- InvalidInput
+ Equilateral
: 1,1,1 should be equilateral

=====
FAIL: testRightTriangleA (__main__.TestTriangles)
-----
Traceback (most recent call last):
  File "TestTriangle.py", line 21, in testRightTriangleA
    self.assertEqual(classifyTriangle(3,4,5),'Right','3,4,5 is a Right triangle')
AssertionError: 'InvalidInput' != 'Right'
- InvalidInput
+ Right
: 3,4,5 is a Right triangle

=====
FAIL: testRightTriangleB (__main__.TestTriangles)
-----
Traceback (most recent call last):
  File "TestTriangle.py", line 24, in testRightTriangleB
    self.assertEqual(classifyTriangle(5,3,4),'Right','5,3,4 is a Right triangle')
AssertionError: 'InvalidInput' != 'Right'
- InvalidInput
+ Right
: 5,3,4 is a Right triangle

-----
Ran 3 tests in 0.002s

FAILED (failures=3)

(base) C:\Users\Syed Zain Raza\Documents\Software Testing\assig2>coverage report -m
Name           Stmts  Miss  Cover   Missing
-----
TestTriangle.py    13     0   100%
Triangle.py       17    12    29%   32, 39-57
-----
TOTAL              30    12    60%

```

Code coverage after changes:

```
(base) C:\Users\Syed Zain Raza\Documents\Software Testing\assig2>coverage run TestTriangle.py
Running unit tests
```

```
.....
```

```
-----
Ran 6 tests in 0.002s
```

```
OK
```

```
(base) C:\Users\Syed Zain Raza\Documents\Software Testing\assig2>coverage report -m
```

Name	Stmts	Miss	Cover	Missing
TestTriangle.py	19	0	100%	
Triangle.py	17	3	82%	32, 35, 40
TOTAL	36	3	92%	

```
-----
(base) C:\Users\Syed Zain Raza\Documents\Software Testing\assig2>
```

Honor Pledge:

"I pledge my honor that I have abided by the Stevens Honor System."