Report for Assignment 1

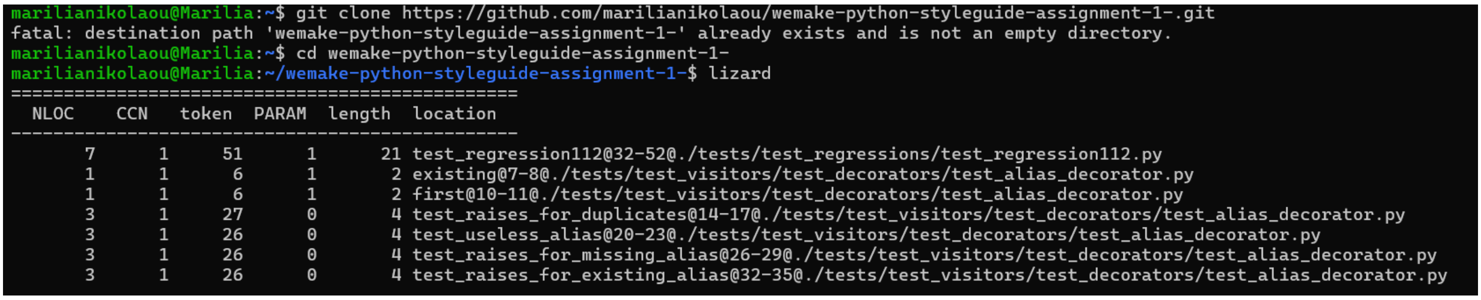
Project chosen:

Name: wemake-python-styleguide (the strictest and most opinionated python linter ever!)

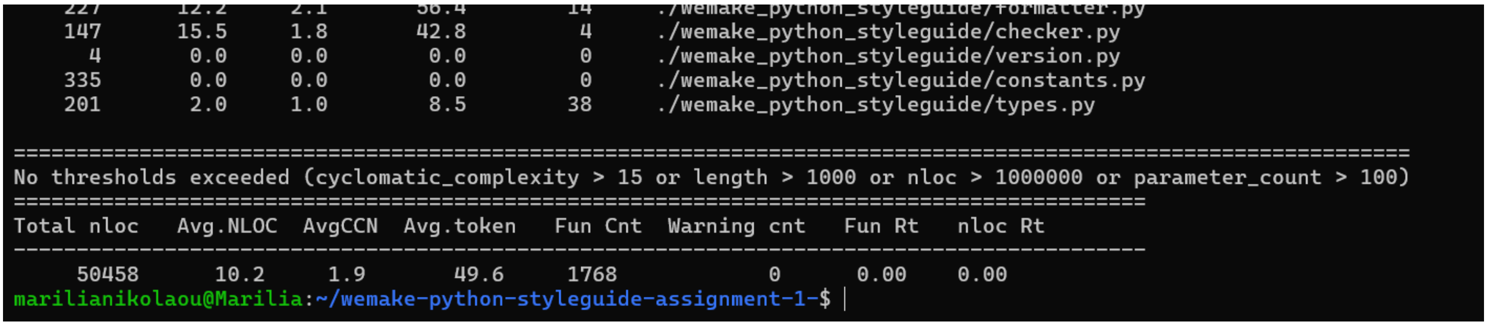
URL: https://github.com/wemake-services/wemake-python-styleguide

Number of lines of code and the tool used to count it:

In order to count the total number of lines of code, the “lizard” tool was used in our clone public repository. As it can be observed from the screenshots below, the whole project consists of 50458 lines of code.



```
marilianikolaou@Marilia:~$ git clone https://github.com/marilianikolaou/wemake-python-styleguide-assignment-1-.git
fatal: destination path 'wemake-python-styleguide-assignment-1-' already exists and is not an empty directory.
marilianikolaou@Marilia:~$ cd wemake-python-styleguide-assignment-1-
marilianikolaou@Marilia:~/wemake-python-styleguide-assignment-1-$ lizard
================================================
  NLOC    CCN   token  PARAM  length  location
------------------------------------------------
       7      1     51      1      21 test_regression112@32-52@./tests/test_regressions/test_regression112.py
       1      1      6      1       2 existing@7-8@./tests/test_visitors/test_decorators/test_alias_decorator.py
       1      1      6      1       2 first@10-11@./tests/test_visitors/test_decorators/test_alias_decorator.py
       3      1     27      0       4 test_raises_for_duplicates@14-17@./tests/test_visitors/test_decorators/test_alias_decorator.py
       3      1     26      0       4 test_useless_alias@20-23@./tests/test_visitors/test_decorators/test_alias_decorator.py
       3      1     26      0       4 test_raises_for_missing_alias@26-29@./tests/test_visitors/test_decorators/test_alias_decorator.py
       3      1     26      0       4 test_raises_for_existing_alias@32-35@./tests/test_visitors/test_decorators/test_alias_decorator.py
```



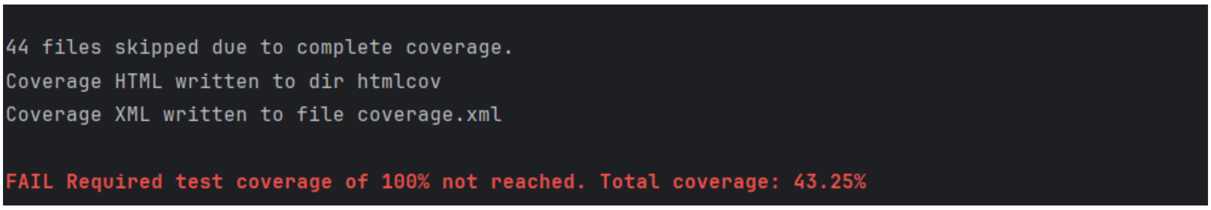
```
    227      12.2     2.1       56.4       14     ./wemake_python_styleguide/formatter.py
    147      15.5     1.8       42.8        4     ./wemake_python_styleguide/checker.py
      4       0.0     0.0        0.0        0     ./wemake_python_styleguide/version.py
    335       0.0     0.0        0.0        0     ./wemake_python_styleguide/constants.py
    201       2.0     1.0        8.5       38     ./wemake_python_styleguide/types.py

=========================================================================================
No thresholds exceeded (cyclomatic_complexity > 15 or length > 1000 or nloc > 1000000 or parameter_count > 100)
==========================================================================================
Total nloc   Avg.NLOC  AvgCCN  Avg.token   Fun Cnt  Warning cnt   Fun Rt   nloc Rt
------------------------------------------------------------------------------------------
     50458      10.2     1.9        49.6     1768            0      0.00    0.00
marilianikolaou@Marilia:~/wemake-python-styleguide-assignment-1-$ |
```

Programming language: Python

Coverage measurement:

Existing tool:

The code coverage tool that was used is “coverage.py”, which is a popular tool for measuring code coverage in Python projects. In order to execute this tool, we first need to navigate to our project directory, and then with the command “coverage run -m pytest tests” we can display the coverage results onto the command line interface. As it can be observed from the report, the code coverage is 43.25%.



```
44 files skipped due to complete coverage.
Coverage HTML written to dir htmlcov
Coverage XML written to file coverage.xml

FAIL Required test coverage of 100% not reached. Total coverage: 43.25%
```

Your own coverage tool:

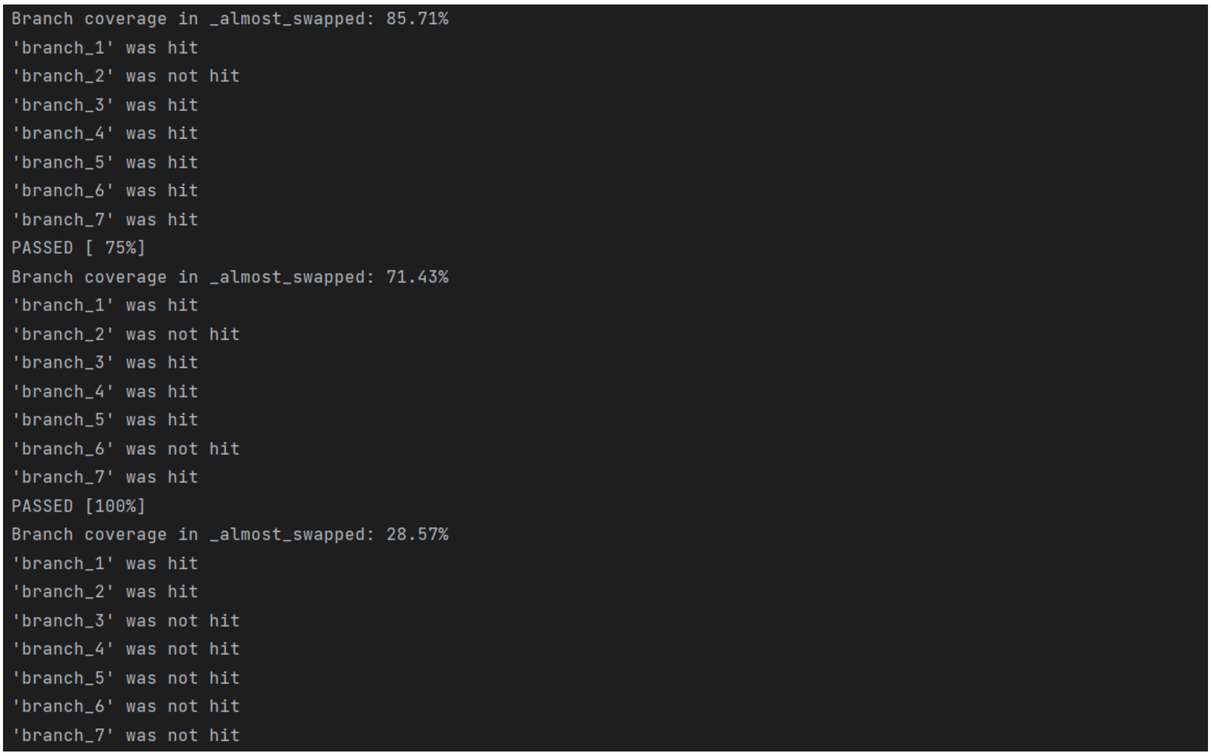
Marilia Nikolaou:

Function 1:

Name: `_almost_swapped`

Commit Link:
https://github.com/marilianikolaou/wemake-python-styleguide-assignment-1-/blob/master/wemake_python_styleguide/visitors/ast/statements.py

Screenshot of the coverage measurements:



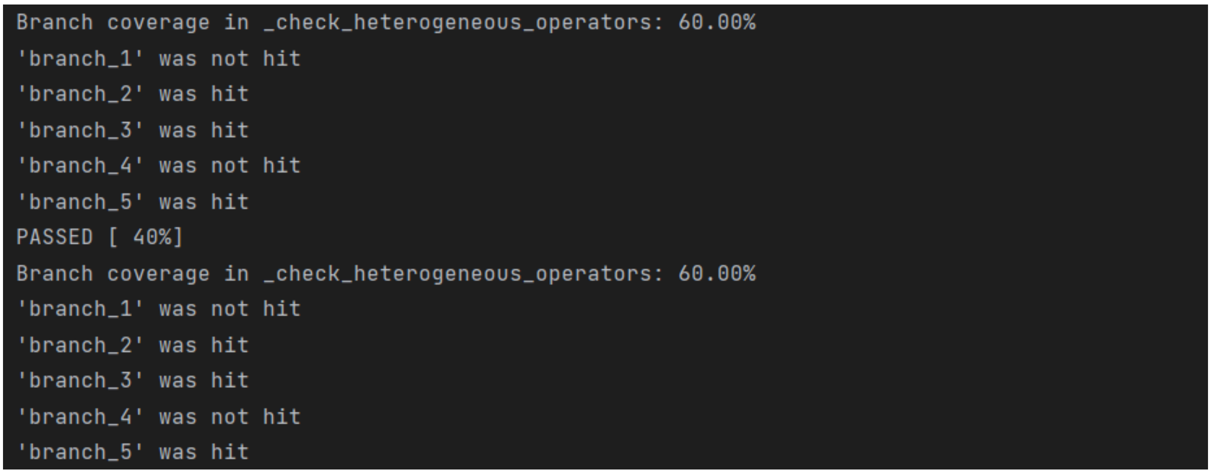
```
Branch coverage in _almost_swapped: 85.71%
'branch_1' was hit
'branch_2' was not hit
'branch_3' was hit
'branch_4' was hit
'branch_5' was hit
'branch_6' was hit
'branch_7' was hit
PASSED [ 75%]
Branch coverage in _almost_swapped: 71.43%
'branch_1' was hit
'branch_2' was not hit
'branch_3' was hit
'branch_4' was hit
'branch_5' was hit
'branch_6' was not hit
'branch_7' was hit
PASSED [100%]
Branch coverage in _almost_swapped: 28.57%
'branch_1' was hit
'branch_2' was hit
'branch_3' was not hit
'branch_4' was not hit
'branch_5' was not hit
'branch_6' was not hit
'branch_7' was not hit
```

Function 2:

Name: `_check_heterogeneous_operators`

Commit Link:
https://github.com/marilianikolaou/wemake-python-styleguide-assignment-1-/blob/master/wemake_python_styleguide/visitors/ast/compares.py

Screenshot of the coverage measurements:



```
Branch coverage in _check_heterogeneous_operators: 60.00%
'branch_1' was not hit
'branch_2' was hit
'branch_3' was hit
'branch_4' was not hit
'branch_5' was hit
PASSED [ 40%]
Branch coverage in _check_heterogeneous_operators: 60.00%
'branch_1' was not hit
'branch_2' was hit
'branch_3' was hit
'branch_4' was not hit
'branch_5' was hit
```

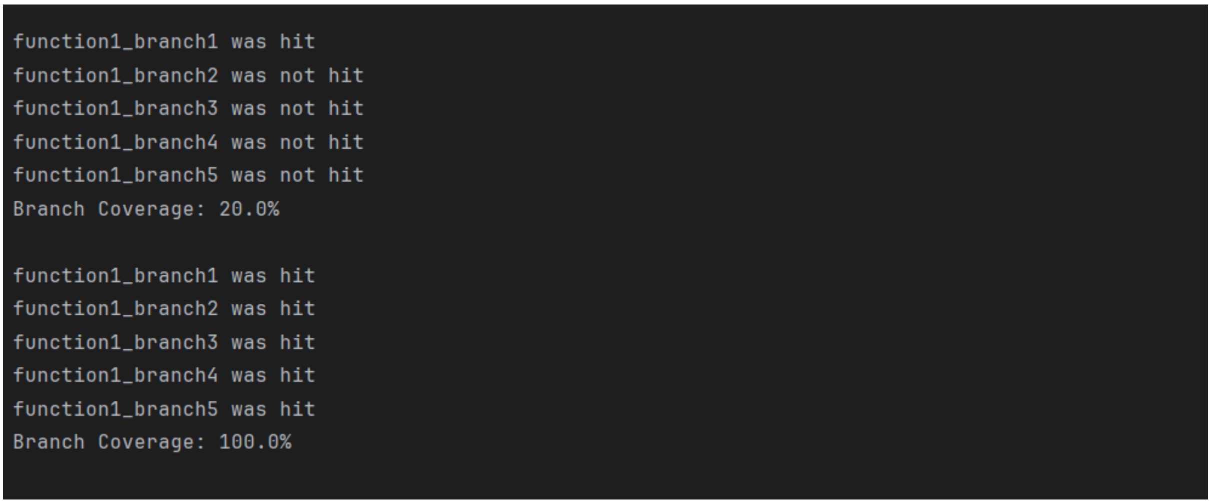
Melani Evangelou:

<u>Function 1:</u>

Name: `process_child_nodes`

Commit Link:
https://github.com/marilianikolaou/wemake-python-styleguide-assignment-1-/commit/cf63f3c5d17299256fc0dd054be0c12251c173c6

Screenshot of the coverage measurements:



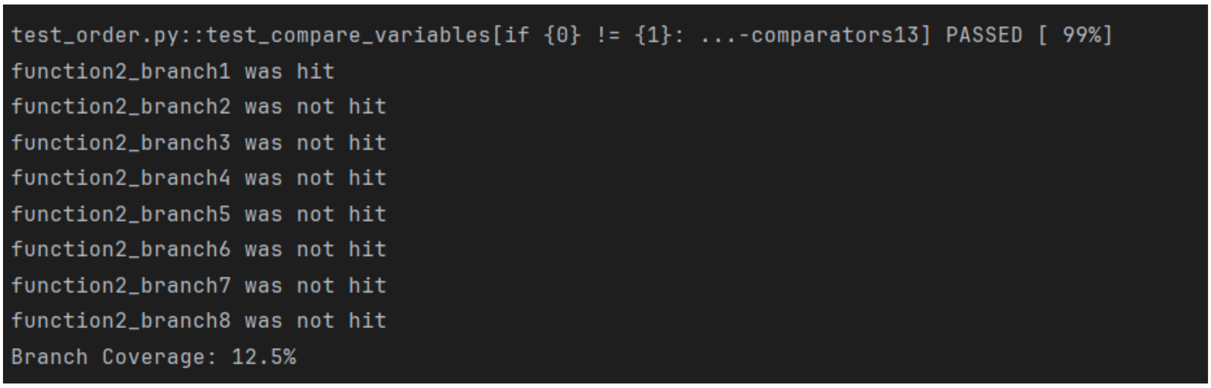
```
function1_branch1 was hit
function1_branch2 was not hit
function1_branch3 was not hit
function1_branch4 was not hit
function1_branch5 was not hit
Branch Coverage: 20.0%

function1_branch1 was hit
function1_branch2 was hit
function1_branch3 was hit
function1_branch4 was hit
function1_branch5 was hit
Branch Coverage: 100.0%
```

<u>Function 2:</u>

Name: `_check_ordering`

Commit Link:
https://github.com/marilianikolaou/wemake-python-styleguide-assignment-1-/commit/5967351c4e5a335a94bad9345aaa593a67485319

Screenshot of the coverage measurements:



```
test_order.py::test_compare_variables[if {0} != {1}: ...-comparators13] PASSED [ 99%]
function2_branch1 was hit
function2_branch2 was not hit
function2_branch3 was not hit
function2_branch4 was not hit
function2_branch5 was not hit
function2_branch6 was not hit
function2_branch7 was not hit
function2_branch8 was not hit
Branch Coverage: 12.5%
```

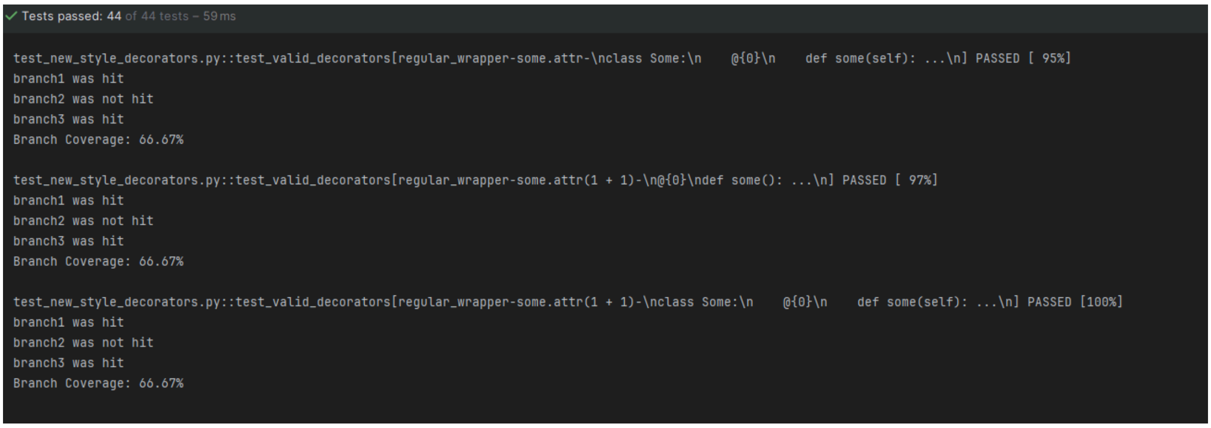
Christos Mouskallis:

Function 1:

Name:_check_new_decorator_syntax

Commit Link: https://github.com/marilianikolaou/wemake-python-styleguide-assignment-1-/commit/c8e21b80e25c9e114592a4c473d82284e74a80ed

Screenshot of the coverage measurements:



```
✓ Tests passed: 44 of 44 tests – 59 ms

test_new_style_decorators.py::test_valid_decorators[regular_wrapper-some.attr-\nclass Some:\n    @{0}\n    def some(self): ...\n] PASSED [ 95%]
branch1 was hit
branch2 was not hit
branch3 was hit
Branch Coverage: 66.67%

test_new_style_decorators.py::test_valid_decorators[regular_wrapper-some.attr(1 + 1)-\n@{0}\ndef some(): ...\n] PASSED [ 97%]
branch1 was hit
branch2 was not hit
branch3 was hit
Branch Coverage: 66.67%

test_new_style_decorators.py::test_valid_decorators[regular_wrapper-some.attr(1 + 1)-\nclass Some:\n    @{0}\n    def some(self): ...\n] PASSED [100%]
branch1 was hit
branch2 was not hit
branch3 was hit
Branch Coverage: 66.67%
```

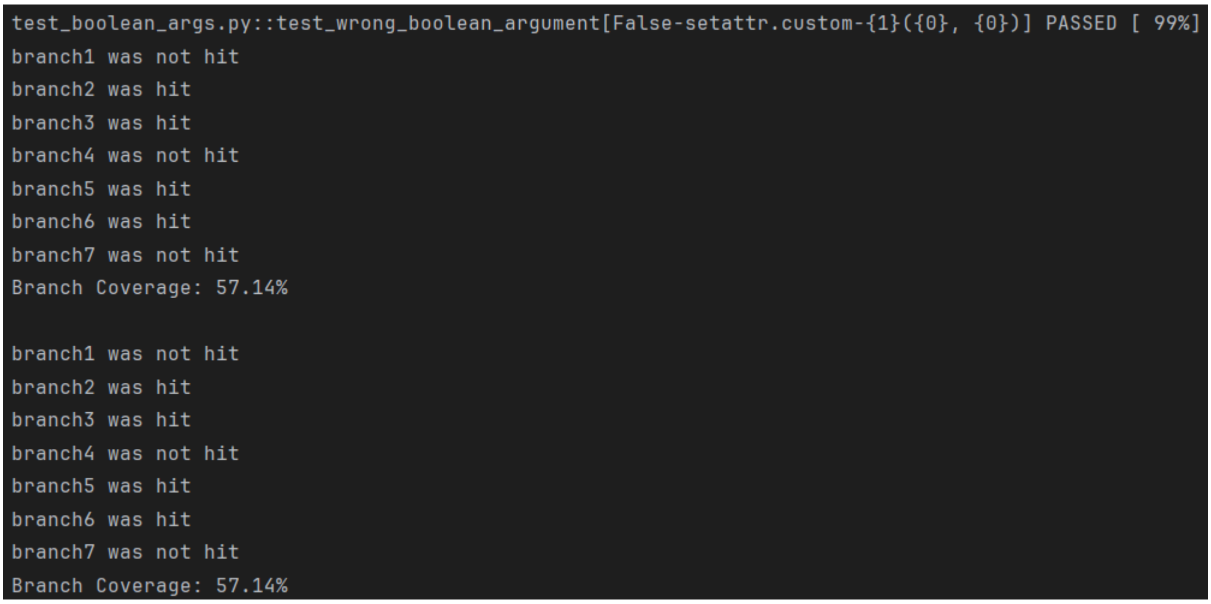
Function 2:

Name: _check_boolean_arguments

Commit Link:

https://github.com/marilianikolaou/wemake-python-styleguide-assignment-1-/commit/b5decf72338d4d8a9f6046b0a62a58a649d94c51

Screenshot of the coverage measurements:



```
test_boolean_args.py::test_wrong_boolean_argument[False-setattr.custom-{1}({0}, {0})] PASSED [ 99%]
branch1 was not hit
branch2 was hit
branch3 was hit
branch4 was not hit
branch5 was hit
branch6 was hit
branch7 was not hit
Branch Coverage: 57.14%

branch1 was not hit
branch2 was hit
branch3 was hit
branch4 was not hit
branch5 was hit
branch6 was hit
branch7 was not hit
Branch Coverage: 57.14%
```

Coverage improvement:

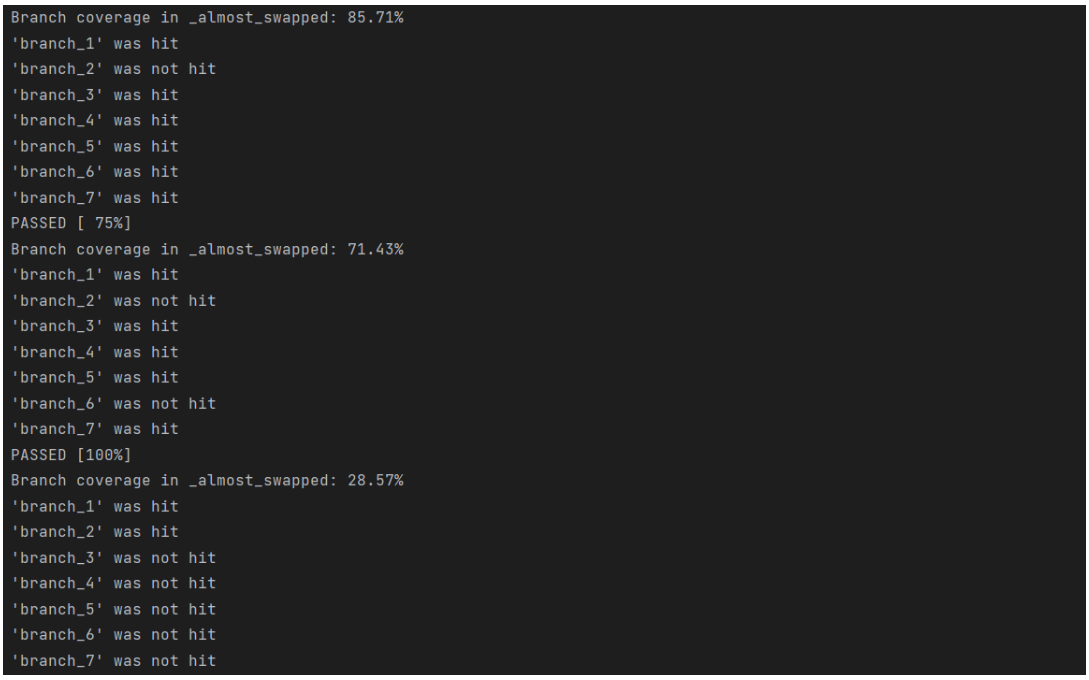
Marilia Nikolaou:

Test 1:

Name: `def test_wrong_swapped_variables`

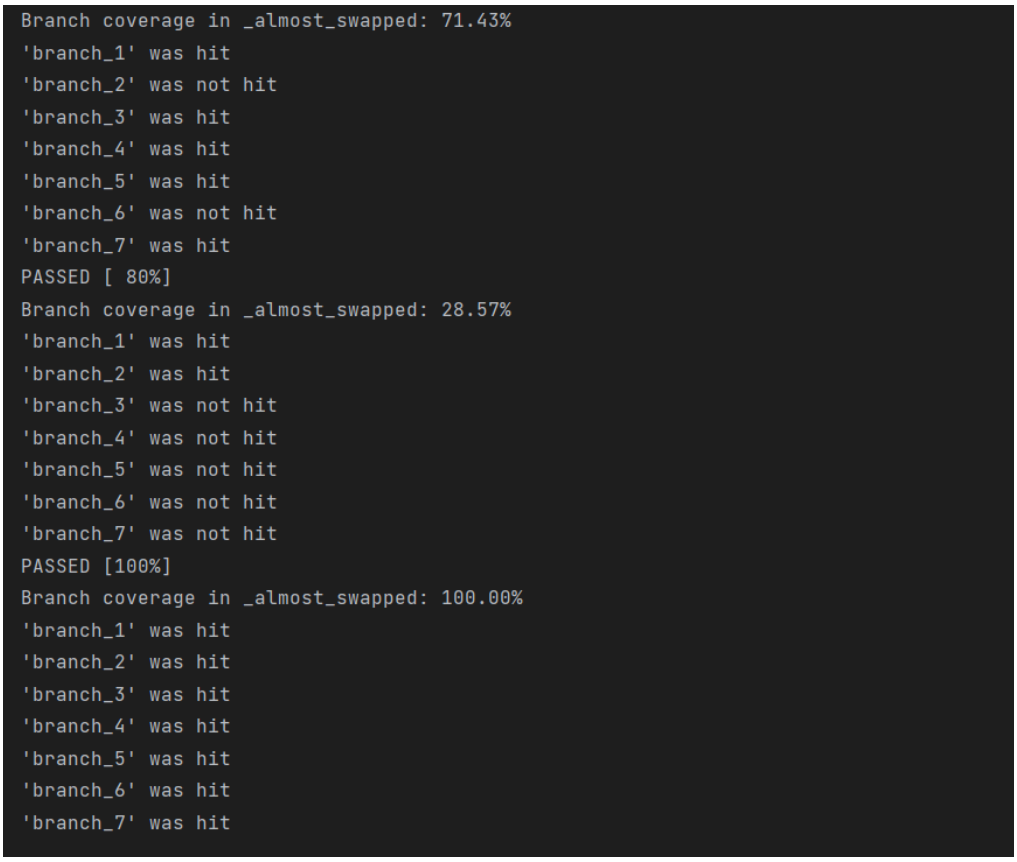
Commit Link:
https://github.com/marilianikolaou/wemake-python-styleguide-assignment-1-/blob/master/tests/test_visitors/test_ast/test_statements/test_almost_swapped.py

Old coverage results:



```
Branch coverage in _almost_swapped: 85.71%
'branch_1' was hit
'branch_2' was not hit
'branch_3' was hit
'branch_4' was hit
'branch_5' was hit
'branch_6' was hit
'branch_7' was hit
PASSED [ 75%]
Branch coverage in _almost_swapped: 71.43%
'branch_1' was hit
'branch_2' was not hit
'branch_3' was hit
'branch_4' was hit
'branch_5' was hit
'branch_6' was not hit
'branch_7' was hit
PASSED [100%]
Branch coverage in _almost_swapped: 28.57%
'branch_1' was hit
'branch_2' was hit
'branch_3' was not hit
'branch_4' was not hit
'branch_5' was not hit
'branch_6' was not hit
'branch_7' was not hit
```

New coverage results:



```
Branch coverage in _almost_swapped: 71.43%
'branch_1' was hit
'branch_2' was not hit
'branch_3' was hit
'branch_4' was hit
'branch_5' was hit
'branch_6' was not hit
'branch_7' was hit
PASSED [ 80%]
Branch coverage in _almost_swapped: 28.57%
'branch_1' was hit
'branch_2' was hit
'branch_3' was not hit
'branch_4' was not hit
'branch_5' was not hit
'branch_6' was not hit
'branch_7' was not hit
PASSED [100%]
Branch coverage in _almost_swapped: 100.00%
'branch_1' was hit
'branch_2' was hit
'branch_3' was hit
'branch_4' was hit
'branch_5' was hit
'branch_6' was hit
'branch_7' was hit
```

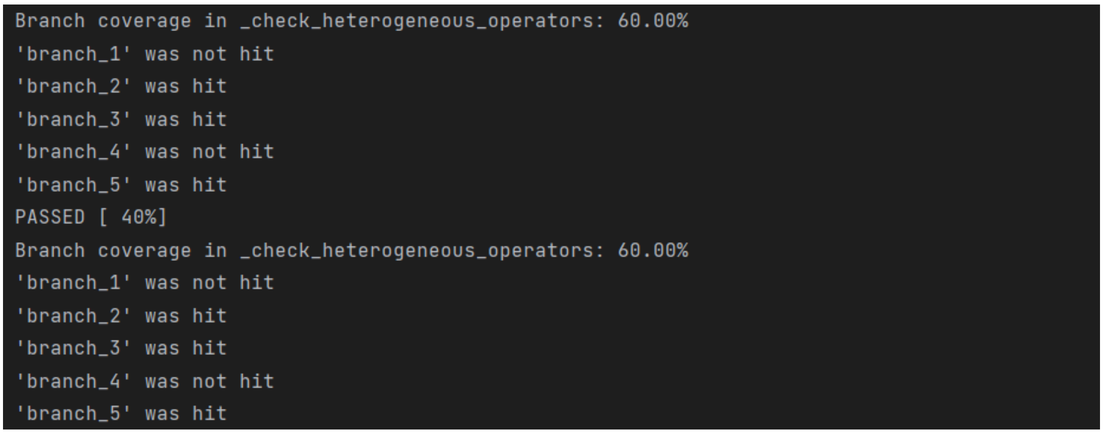
The branch coverage in the function **"_almost_swapped"** has improved significantly, because as it can be seen from the screenshots above, initially the coverage is 85.71%, 71.43%, and 28.57% in different test cases, and the final test case has branch coverage 100%. This indicates, that the addition of the new test case, covers all the previously untested branches and all the branches are hit successfully.

Test 2:

Name: `def test_correct_compare_operators`

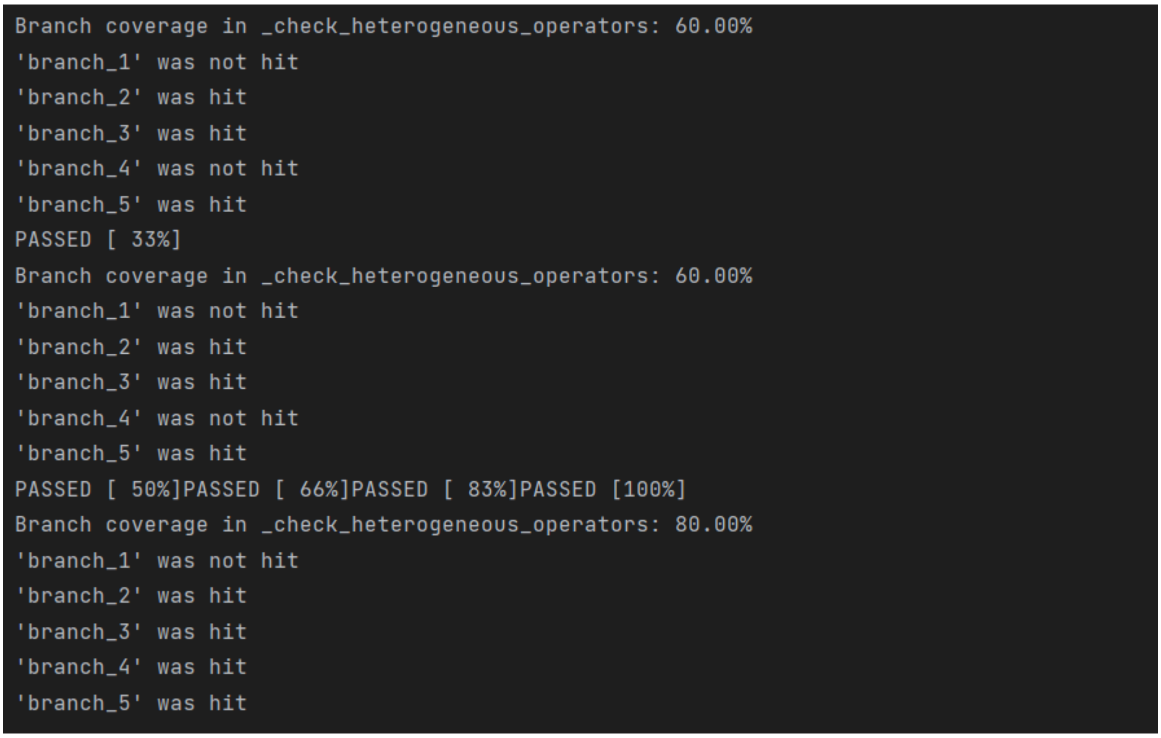
Commit Link:
https://github.com/marilianikolaou/wemake-python-styleguide-assignment-1-/blob/master/tests/test_visitors/test_ast/test_compares/test_heterogenous_compare.py

Old coverage results:



```
Branch coverage in _check_heterogeneous_operators: 60.00%
'branch_1' was not hit
'branch_2' was hit
'branch_3' was hit
'branch_4' was not hit
'branch_5' was hit
PASSED [ 40%]
Branch coverage in _check_heterogeneous_operators: 60.00%
'branch_1' was not hit
'branch_2' was hit
'branch_3' was hit
'branch_4' was not hit
'branch_5' was hit
```

New coverage results:



```
Branch coverage in _check_heterogeneous_operators: 60.00%
'branch_1' was not hit
'branch_2' was hit
'branch_3' was hit
'branch_4' was not hit
'branch_5' was hit
PASSED [ 33%]
Branch coverage in _check_heterogeneous_operators: 60.00%
'branch_1' was not hit
'branch_2' was hit
'branch_3' was hit
'branch_4' was not hit
'branch_5' was hit
PASSED [ 50%]PASSED [ 66%]PASSED [ 83%]PASSED [100%]
Branch coverage in _check_heterogeneous_operators: 80.00%
'branch_1' was not hit
'branch_2' was hit
'branch_3' was hit
'branch_4' was hit
'branch_5' was hit
```

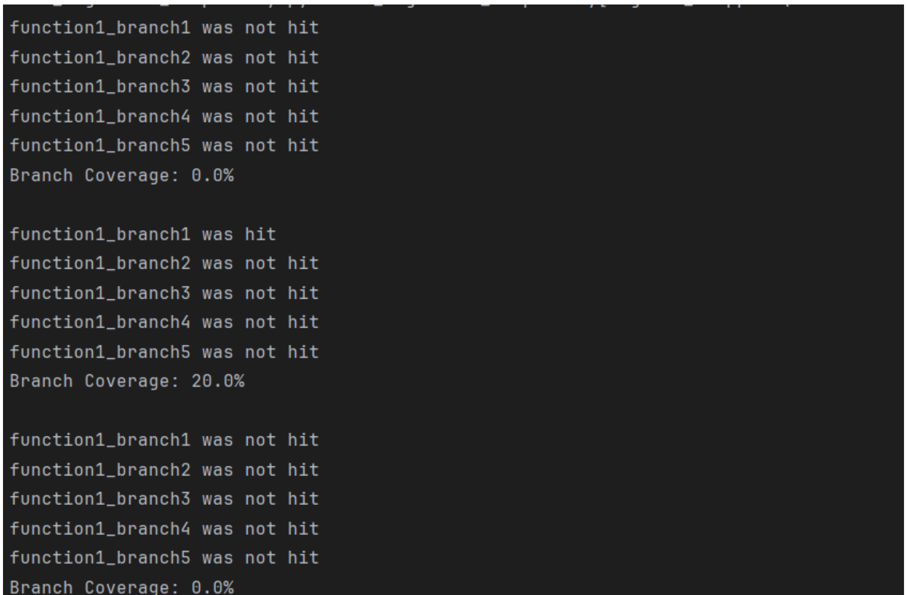
The branch coverage in the function **"_check_heterogeneous_operators"** has improved significantly, because as it can be seen from the screenshots above, initially the coverage is 60% in different test cases, and the final test case has branch coverage 80%. This indicates, that the addition of the new test case, covers all the untested branches instead of the first one.

Melani Evangelou:

Test 1:
Name: `test_cognitive_complexity`

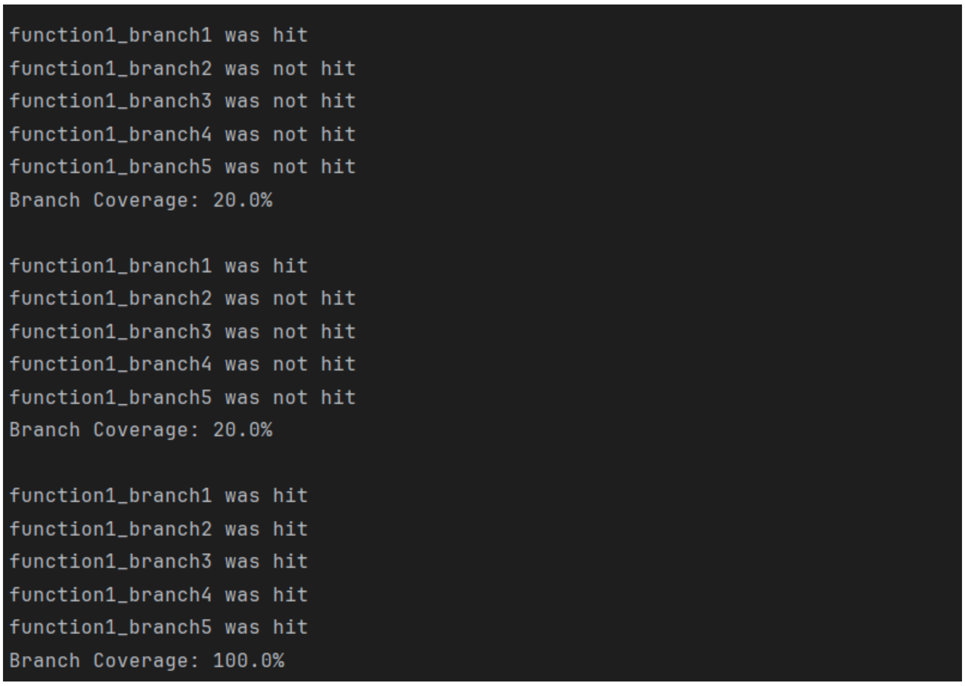
Commit Link:
https://github.com/marilianikolaou/wemake-python-styleguide-assignment-1-/commit/b2216d6b1f99e5f13a403de8fc11d70f5600f830

Old coverage results:



```
function1_branch1 was not hit
function1_branch2 was not hit
function1_branch3 was not hit
function1_branch4 was not hit
function1_branch5 was not hit
Branch Coverage: 0.0%

function1_branch1 was hit
function1_branch2 was not hit
function1_branch3 was not hit
function1_branch4 was not hit
function1_branch5 was not hit
Branch Coverage: 20.0%

function1_branch1 was not hit
function1_branch2 was not hit
function1_branch3 was not hit
function1_branch4 was not hit
function1_branch5 was not hit
Branch Coverage: 0.0%
```

New coverage results:



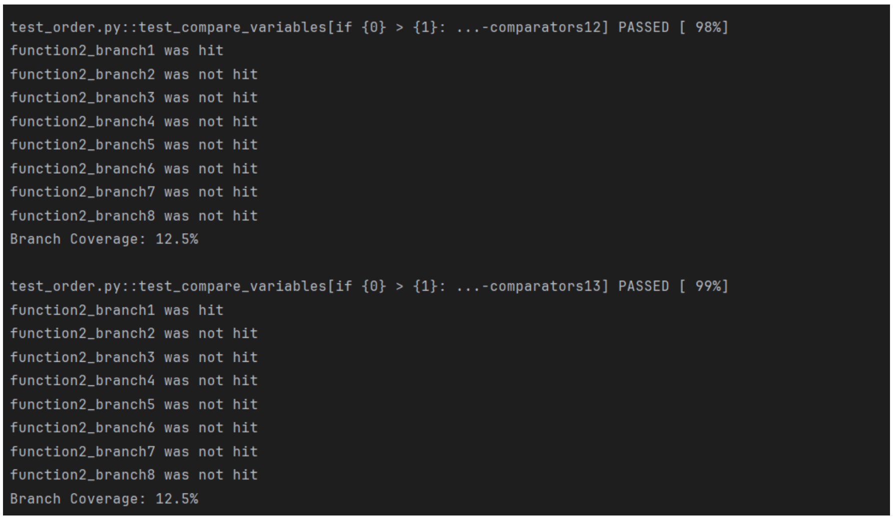
```
function1_branch1 was hit
function1_branch2 was not hit
function1_branch3 was not hit
function1_branch4 was not hit
function1_branch5 was not hit
Branch Coverage: 20.0%

function1_branch1 was hit
function1_branch2 was not hit
function1_branch3 was not hit
function1_branch4 was not hit
function1_branch5 was not hit
Branch Coverage: 20.0%

function1_branch1 was hit
function1_branch2 was hit
function1_branch3 was hit
function1_branch4 was hit
function1_branch5 was hit
Branch Coverage: 100.0%
```

The branch coverage in the function "`test_cognitive_complexity`" has improved significantly, because as it can be seen from the screenshots above, initially the highest coverage shown in different test cases of the test we are examining was 20% , and now I have created a test case that has branch coverage 100%. The reason why the coverage was improved significantly is because the already existing test case did not include a try-catch block that was necessary to hit this branch `if isinstance(node, ast.Try)` and consequently the one nested inside of it.

Test 2:
Name: `test_compare_variables`

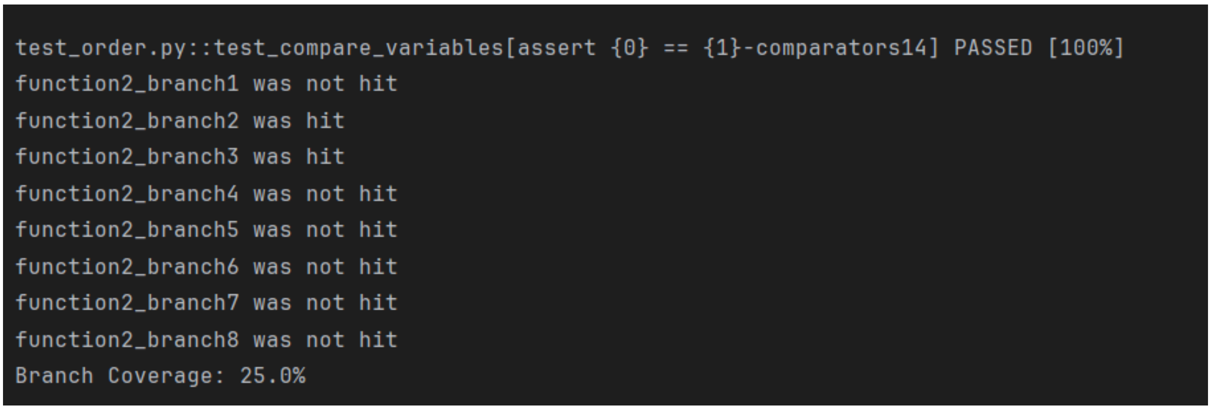
Commit Link:
https://github.com/marilianikolaou/wemake-python-styleguide-assignment-1-/commit/491b99f51a1a4cdec4cfe9f40b68454916d2f053

Old coverage results:



```
test_order.py::test_compare_variables[if {0} > {1}: ...-comparators12] PASSED [ 98%]
function2_branch1 was hit
function2_branch2 was not hit
function2_branch3 was not hit
function2_branch4 was not hit
function2_branch5 was not hit
function2_branch6 was not hit
function2_branch7 was not hit
function2_branch8 was not hit
Branch Coverage: 12.5%

test_order.py::test_compare_variables[if {0} > {1}: ...-comparators13] PASSED [ 99%]
function2_branch1 was hit
function2_branch2 was not hit
function2_branch3 was not hit
function2_branch4 was not hit
function2_branch5 was not hit
function2_branch6 was not hit
function2_branch7 was not hit
function2_branch8 was not hit
Branch Coverage: 12.5%
```

New coverage results:



```
test_order.py::test_compare_variables[assert {0} == {1}-comparators14] PASSED [100%]
function2_branch1 was not hit
function2_branch2 was hit
function2_branch3 was hit
function2_branch4 was not hit
function2_branch5 was not hit
function2_branch6 was not hit
function2_branch7 was not hit
function2_branch8 was not hit
Branch Coverage: 25.0%
```

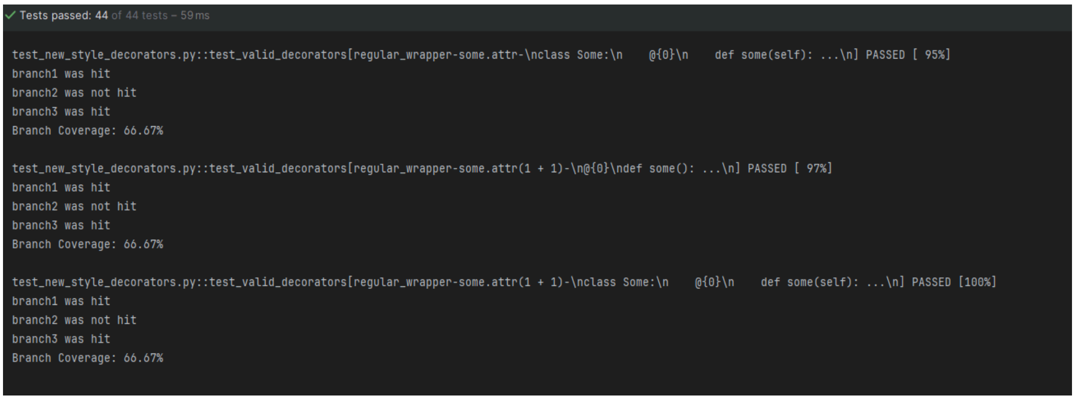
The branch coverage in the function “**`test_compare_variables`**” has improved, because as it can be seen from the screenshots above, initially branch 3 was never hit and now I have created a test case that manages to hit it. The reason why the coverage has improved it is because we needed to include a special cade node like in or not in in order to hit branch3 `if self._is_special_case(node).`

Christos Mouskallis:

Test 1:
Name: `test_new_style_decorators.py`

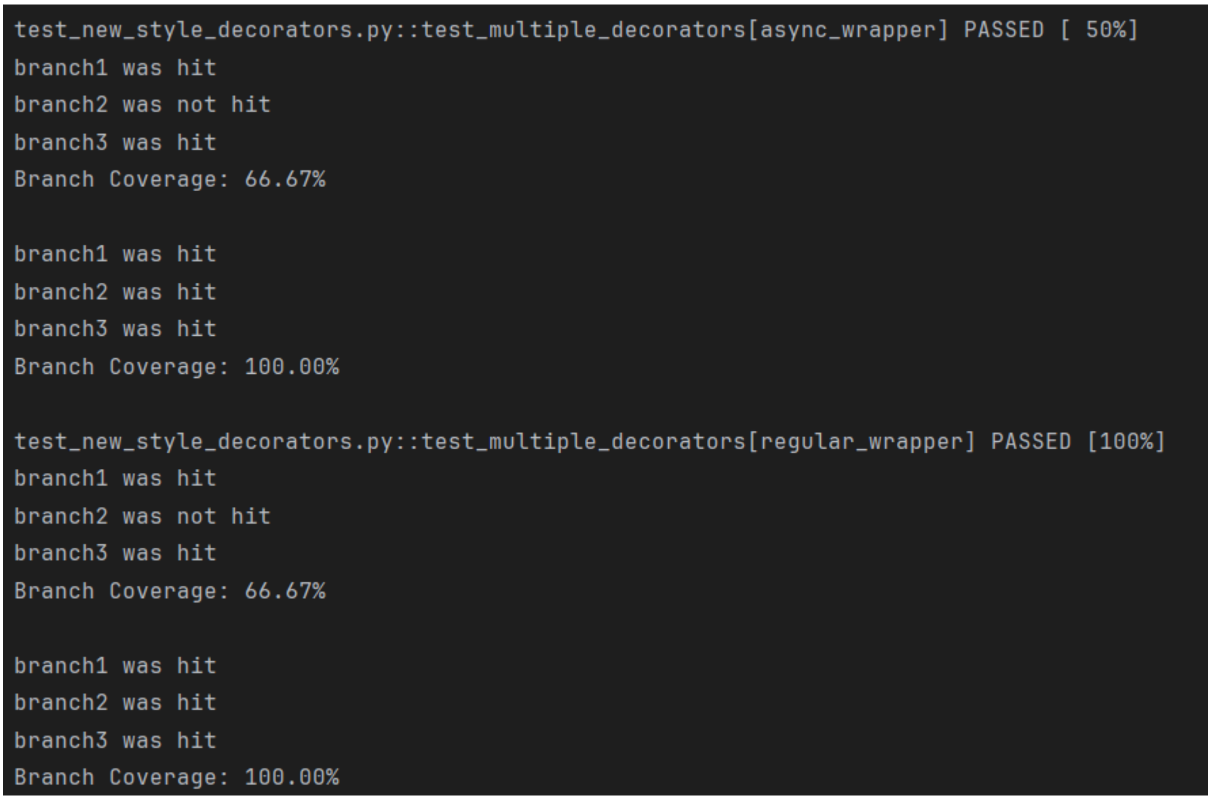
Commit Link:
https://github.com/marilianikolaou/wemake-python-styleguide-assignment-1-/commit/c8e21b80e25c9e114592a4c473d82284e74a80ed

Old coverage results:



```
test_new_style_decorators.py::test_valid_decorators[regular_wrapper-some.attr-\nclass Some:\n    @{0}\n    def some(self): ...\n] PASSED [ 95%]
branch1 was hit
branch2 was not hit
branch3 was hit
Branch Coverage: 66.67%

test_new_style_decorators.py::test_valid_decorators[regular_wrapper-some.attr(1 + 1)-\n@{0}\ndef some(): ...\n] PASSED [ 97%]
branch1 was hit
branch2 was not hit
branch3 was hit
Branch Coverage: 66.67%

test_new_style_decorators.py::test_valid_decorators[regular_wrapper-some.attr(1 + 1)-\nclass Some:\n    @{0}\n    def some(self): ...\n] PASSED [100%]
branch1 was hit
branch2 was not hit
branch3 was hit
Branch Coverage: 66.67%
```

New coverage results:



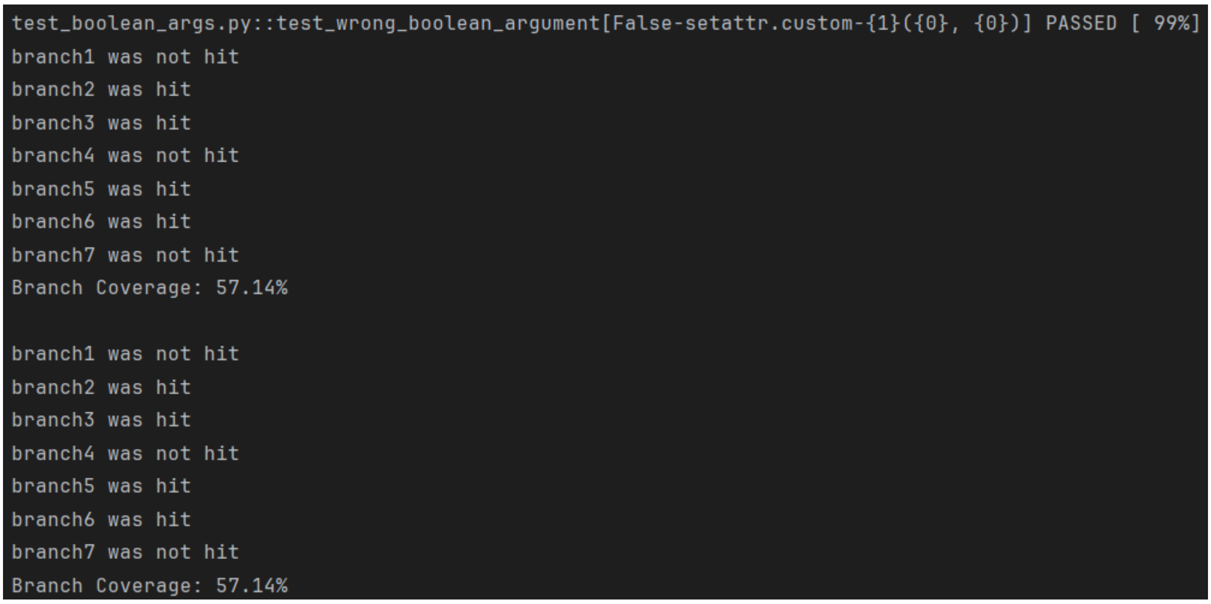
```
test_new_style_decorators.py::test_multiple_decorators[async_wrapper] PASSED [ 50%]
branch1 was hit
branch2 was not hit
branch3 was hit
Branch Coverage: 66.67%

branch1 was hit
branch2 was hit
branch3 was hit
Branch Coverage: 100.00%

test_new_style_decorators.py::test_multiple_decorators[regular_wrapper] PASSED [100%]
branch1 was hit
branch2 was not hit
branch3 was hit
Branch Coverage: 66.67%

branch1 was hit
branch2 was hit
branch3 was hit
Branch Coverage: 100.00%
```

The branch coverage in the function "`test_new_style_decorators.py`" has improved, because as it can be seen from the screenshots above, initially branch 2 was never hit and now I have created a test case that manages to hit it. Initially, the branch coverage was 66.67% and now with hitting branch 2 it went up to 100%.

Test 2:
Name: `test_boolean_args.py`

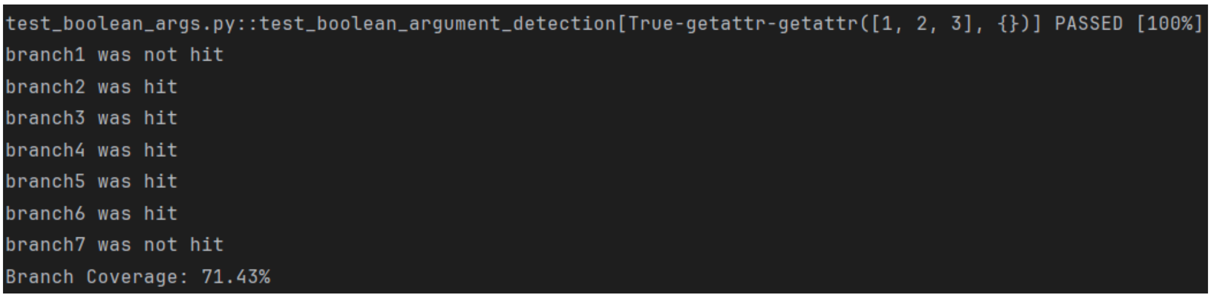
Commit Link:
https://github.com/marilianikolaou/wemake-python-styleguide-assignment-1-/commit/b5decf72338d4d8a9f6046b0a62a58a649d94c51

Old coverage results:



```
test_boolean_args.py::test_wrong_boolean_argument[False-setattr.custom-{1}({0}, {0})] PASSED [ 99%]
branch1 was not hit
branch2 was hit
branch3 was hit
branch4 was not hit
branch5 was hit
branch6 was hit
branch7 was not hit
Branch Coverage: 57.14%

branch1 was not hit
branch2 was hit
branch3 was hit
branch4 was not hit
branch5 was hit
branch6 was hit
branch7 was not hit
Branch Coverage: 57.14%
```

New coverage results:

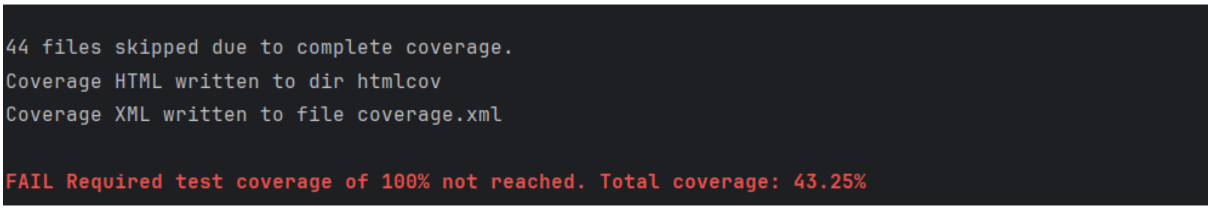


```
test_boolean_args.py::test_boolean_argument_detection[True-getattr-getattr([1, 2, 3], {})] PASSED [100%]
branch1 was not hit
branch2 was hit
branch3 was hit
branch4 was hit
branch5 was hit
branch6 was hit
branch7 was not hit
Branch Coverage: 71.43%
```

The branch coverage in the function "`test_boolean_args.py`" has improved, because as it can be seen from the screenshots above, initially branch 4 was never hit and now I have created a test case that manages to hit it. Initially, the branch coverage was 57.14% and now with hitting branch 4 it went up to 71.43%.

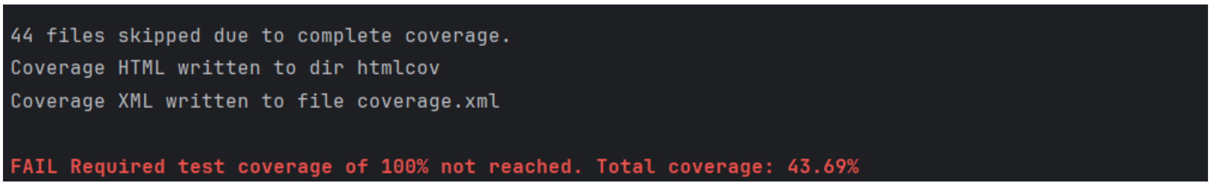
Overall:

Old coverage results:



```
44 files skipped due to complete coverage.
Coverage HTML written to dir htmlcov
Coverage XML written to file coverage.xml

FAIL Required test coverage of 100% not reached. Total coverage: 43.25%
```

New coverage results:



```
44 files skipped due to complete coverage.
Coverage HTML written to dir htmlcov
Coverage XML written to file coverage.xml

FAIL Required test coverage of 100% not reached. Total coverage: 43.69%
```

Statement of individual contributions:

Marilia Nikolaou: ran the "lizard" tool to measure the number of lines of code, measured the coverage for the functions **"_almost_swapped"**, and **"_check_heterogeneous_operators"**, and improved the test cases **"test_wrong_swapped_variables"**, and **"test_correct_compare_operators"** for the functions respectively.

Melani Evangelou:
measured the coverage for the functions **“process_child_nodes”**, and “**_check_ordering**”, and improved the test cases “**test_cognitive_complexity**”, and “**test_compare_variables**” for the functions respectively.

Christos Mouskallis:
measured the coverage for the functions **“_check_new_decorator_syntax”**, and “ **_check_boolean_arguments**”, and improved the test cases “**test_new_style_decorators.py**”, and “**test_boolean_args.py**” for the functions respectively.