WE NEED A BETTER TESTING FRAMEWORK

NOAM TENNE

$ WHOAMI

HACKING AROUND FOR THE PAST ~15 YEARS

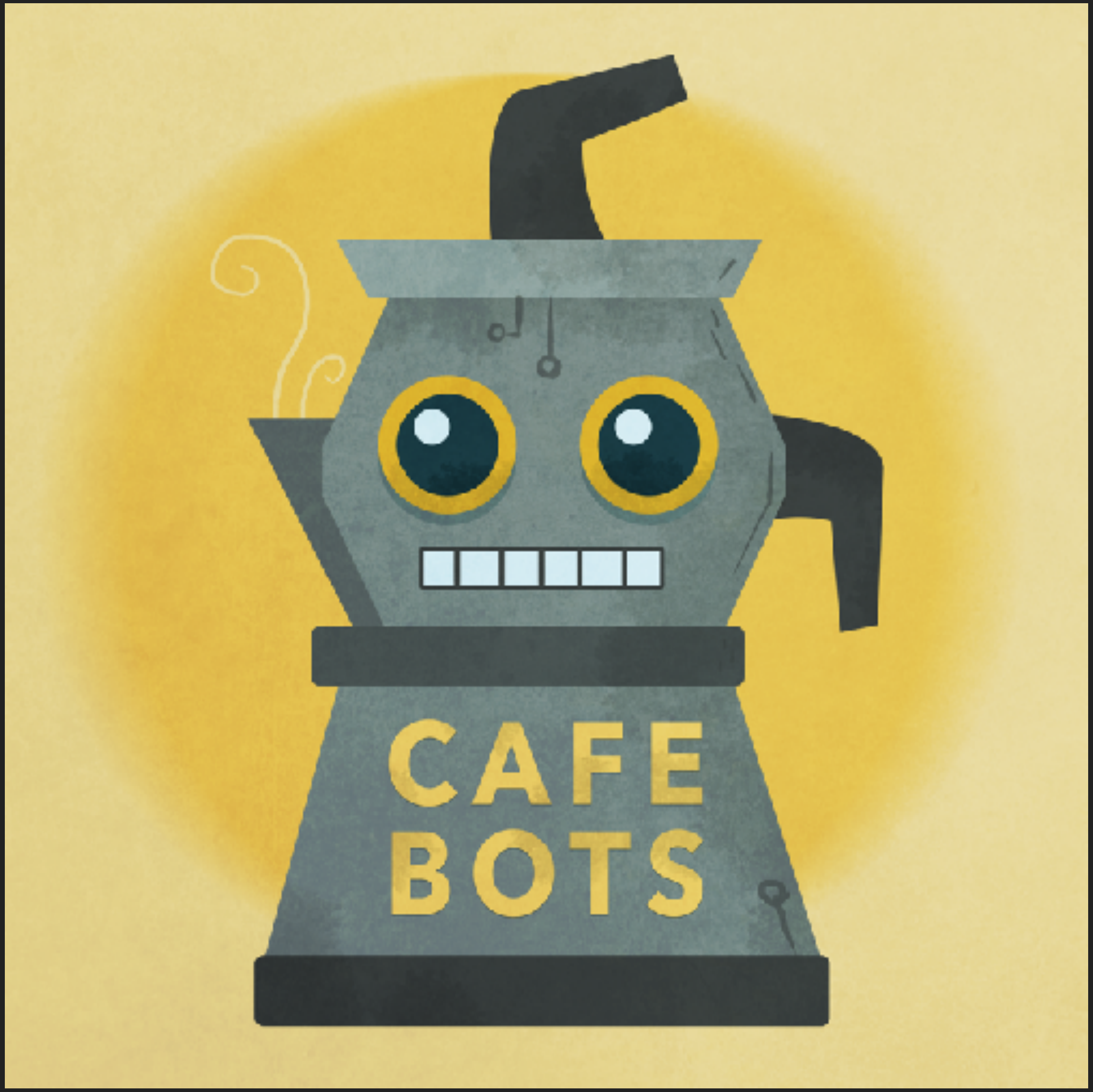
HEALTHY.IO ← Best Company Ever!!!

@NOAMTENNE

GITHUB.COM/NOAMT

MEDIUM.COM/@NOAMT

```
$ WHOAMI
```



PODCASTING “CAFE BOTS”

WITH YOAV LUFT AND TOM KAMINSKI

$ CONFESSION

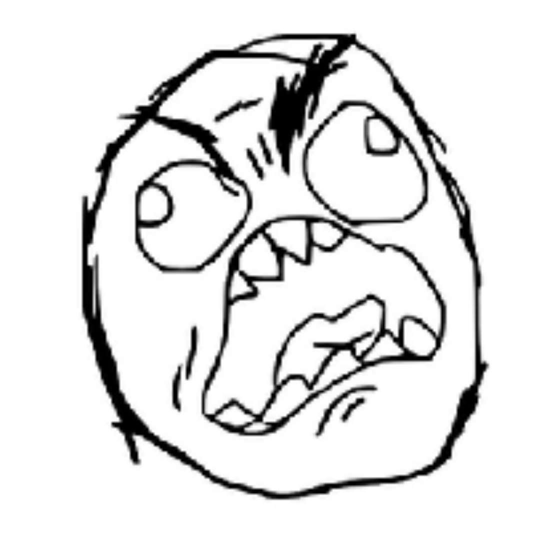
NIMOY

$ CLARIFICATIONS

CONSTRUCTIVE FEEDBACK

$ CLARIFICATIONS

CONSTRUCTIVE FEEDBACK



I DON'T HATE PYTHON

$ CLARIFICATIONS

CONSTRUCTIVE FEEDBACK

I DON’T HATE PYTHON

SINCERE

$ QUESTIONS

DO YOU TDD?

$ QUESTIONS

DO YOU BDD?

$ WHICH BDD

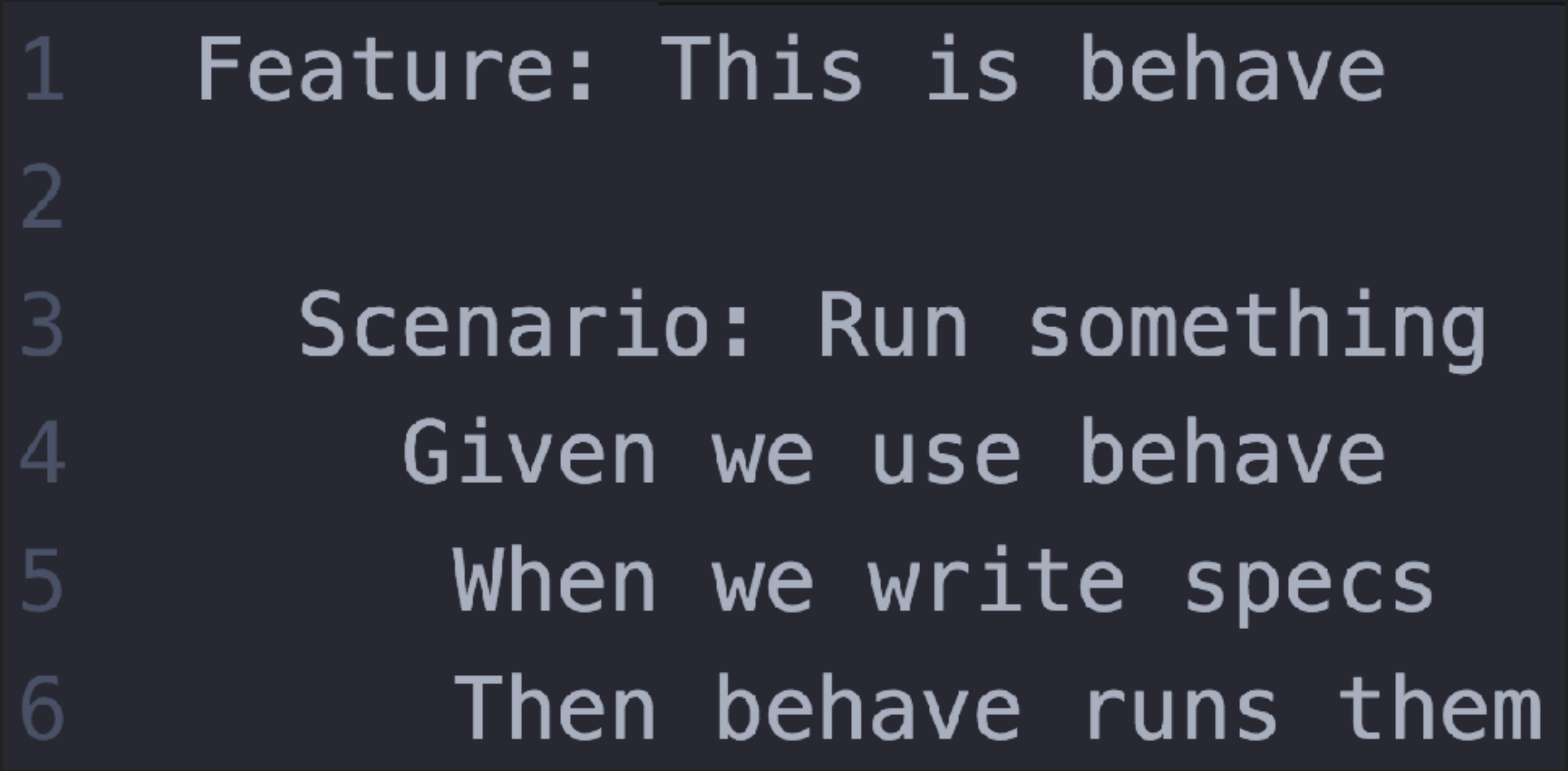
SCENARIO: DESCRIPTION

GIVEN: A STATE

WHEN: AN ACTION IS PERFORMED

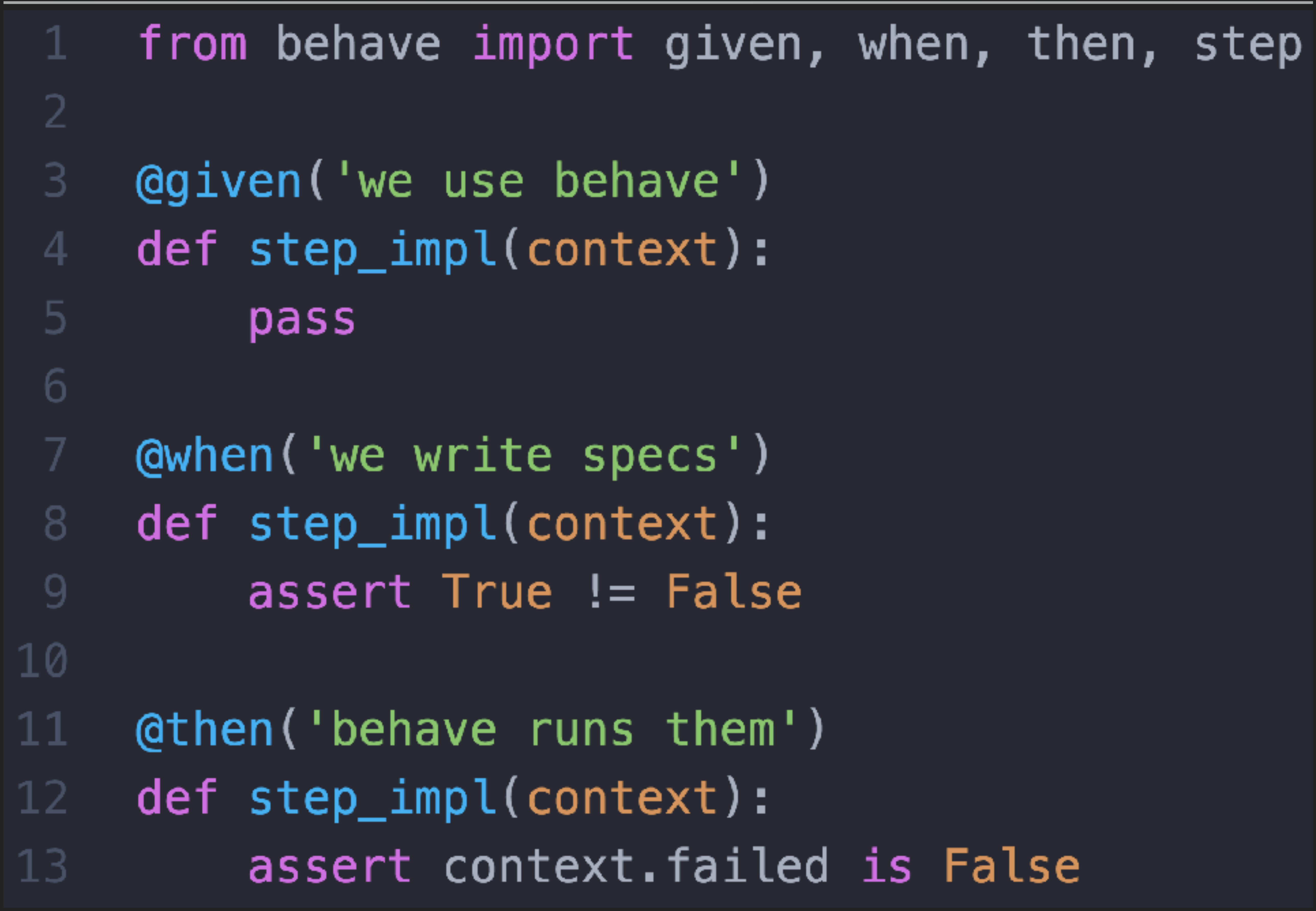
EXPECT: AN OUTCOME

$ VIM BEHAVE.FEATURE



```
Feature: This is behave

  Scenario: Run something
    Given we use behave
     When we write specs
     Then behave runs them
```

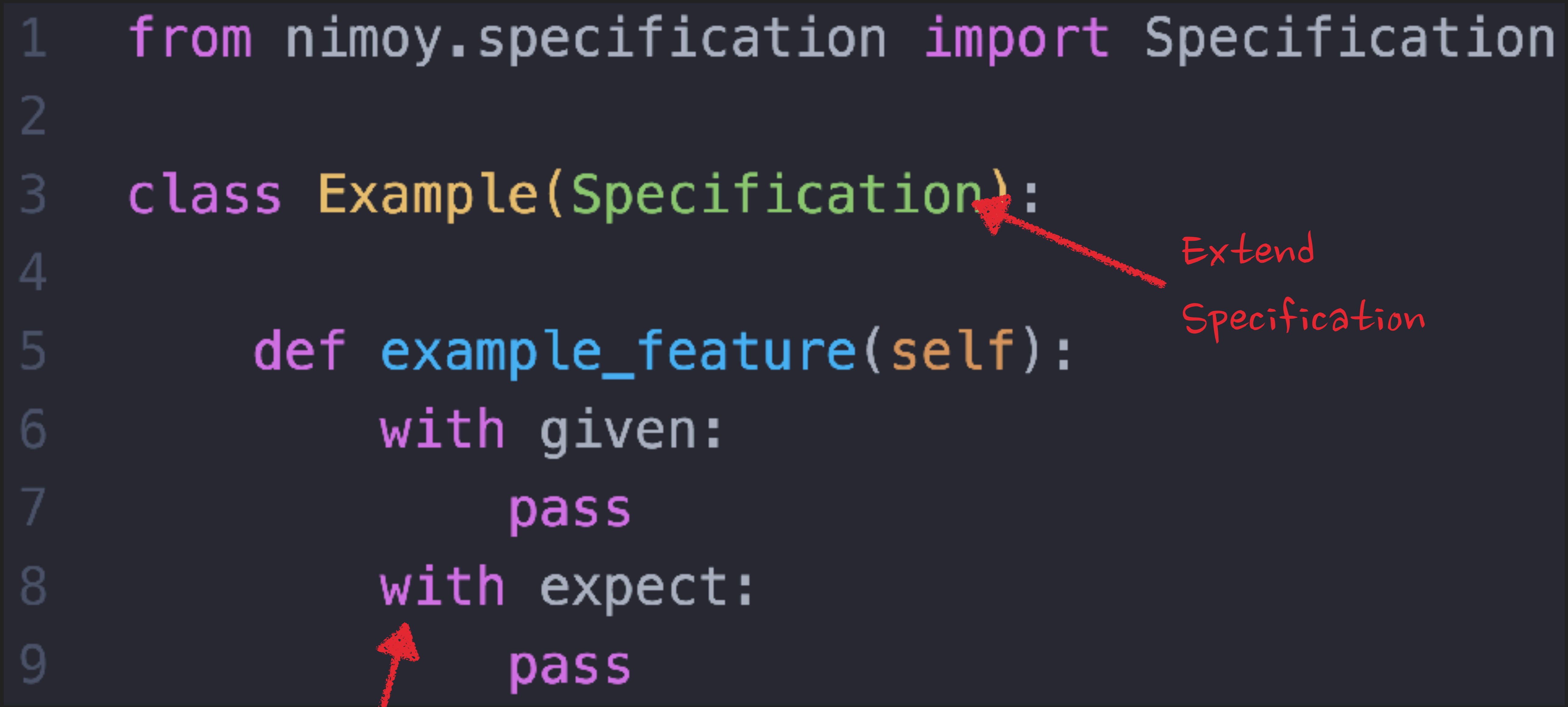
$ VIM STEPS/BEHAVE_STEPS.PY



```python
from behave import given, when, then, step

@given('we use behave')
def step_impl(context):
    pass

@when('we write specs')
def step_impl(context):
    assert True != False

@then('behave runs them')
def step_impl(context):
    assert context.failed is False
```

$ VIM NIMOY_SPEC.PY

End with _spec.py

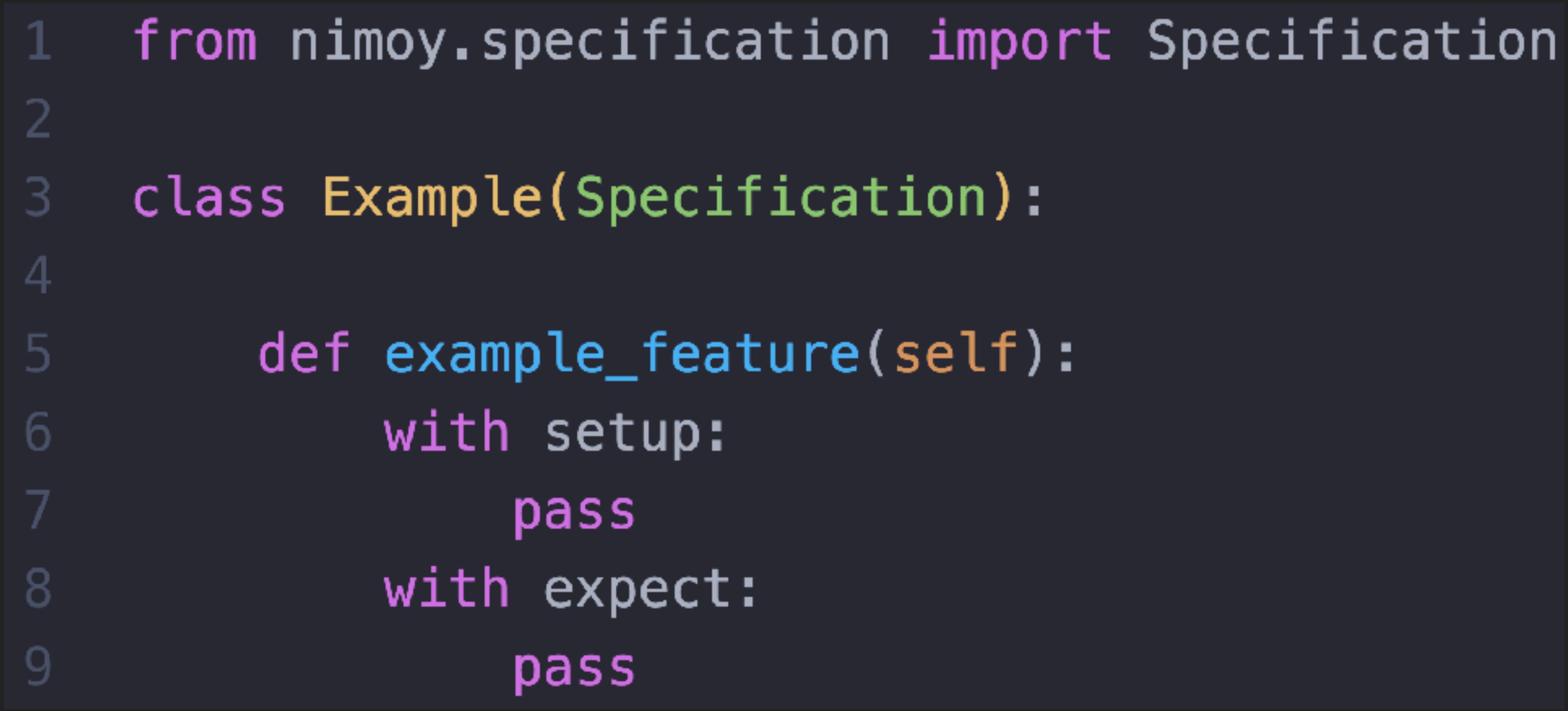


```
from nimoy.specification import Specification

class Example(Specification):

    def example_feature(self):
        with given:
            pass
        with expect:
            pass
```

Extend Specification

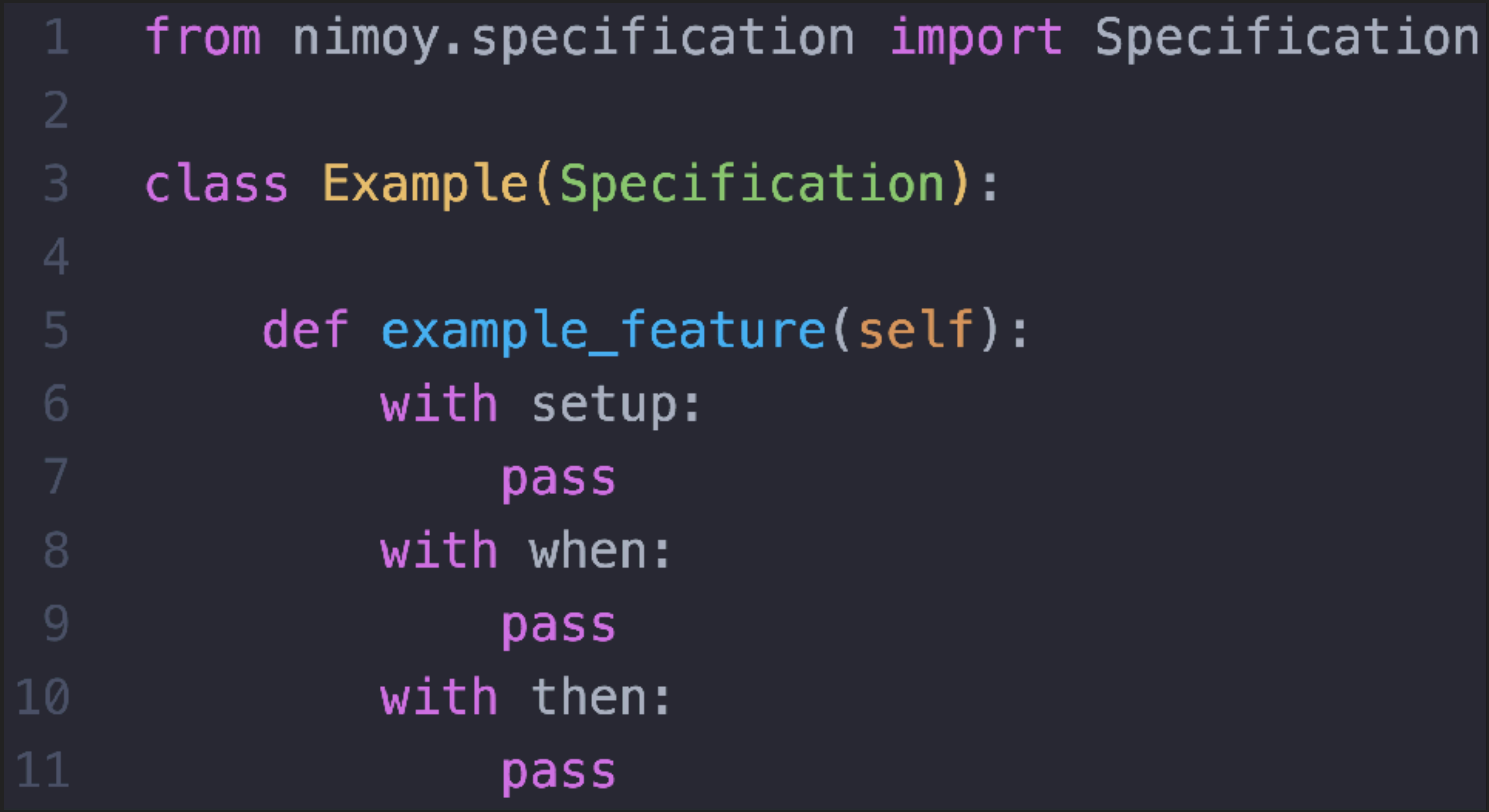
Contain steps using "with"

$ VIM NIMOY_SPEC.PY



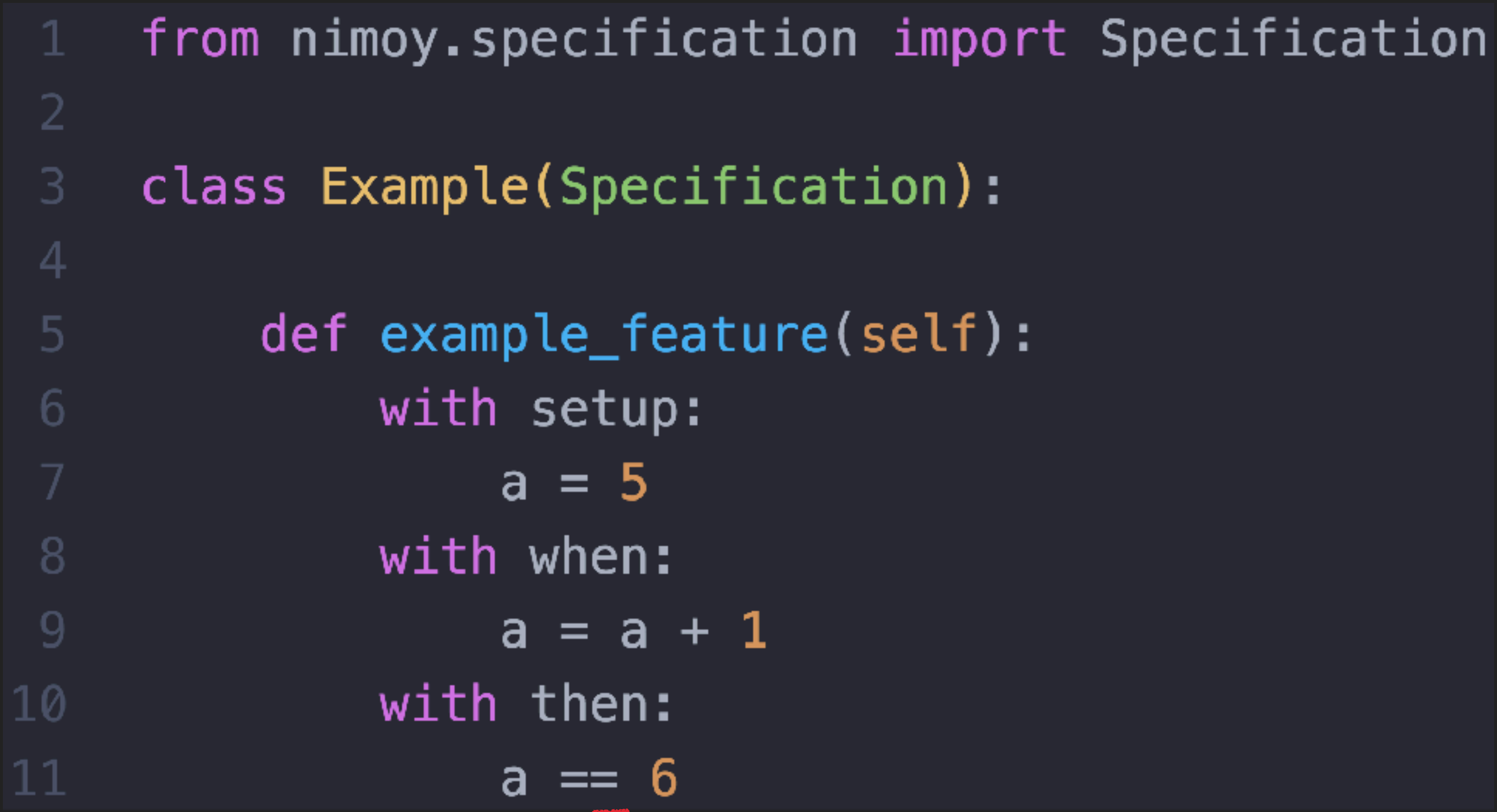
```python
from nimoy.specification import Specification

class Example(Specification):

    def example_feature(self):
        with setup:
            pass
        with expect:
            pass
```

$ VIM NIMOY_SAR_SPEC.PY



```python
from nimoy.specification import Specification

class Example(Specification):

    def example_feature(self):
        with setup:
            pass
        with when:
            pass
        with then:
            pass
```

$ VIM NIMOY_SAR_SPEC.PY



```python
from nimoy.specification import Specification

class Example(Specification):

    def example_feature(self):
        with setup:
            a = 5
        with when:
            a = a + 1
        with then:
            a == 6
```

Look, mom! No asserts!

$ QUESTIONS

DO YOU DDT?

$ WHICH DDT

SCENARIO: DESCRIPTION

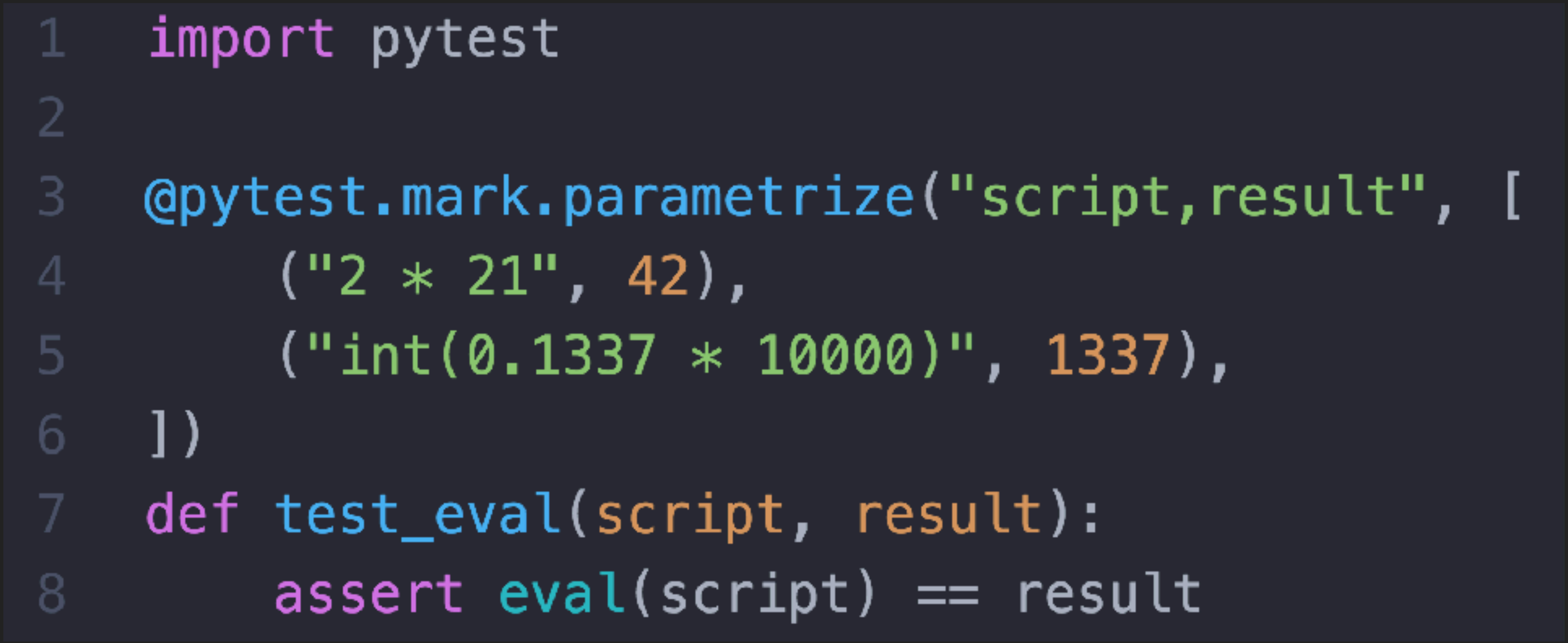
GIVEN: A STATE

WHEN: AN ACTION IS PERFORMED

EXPECT: AN OUTCOME

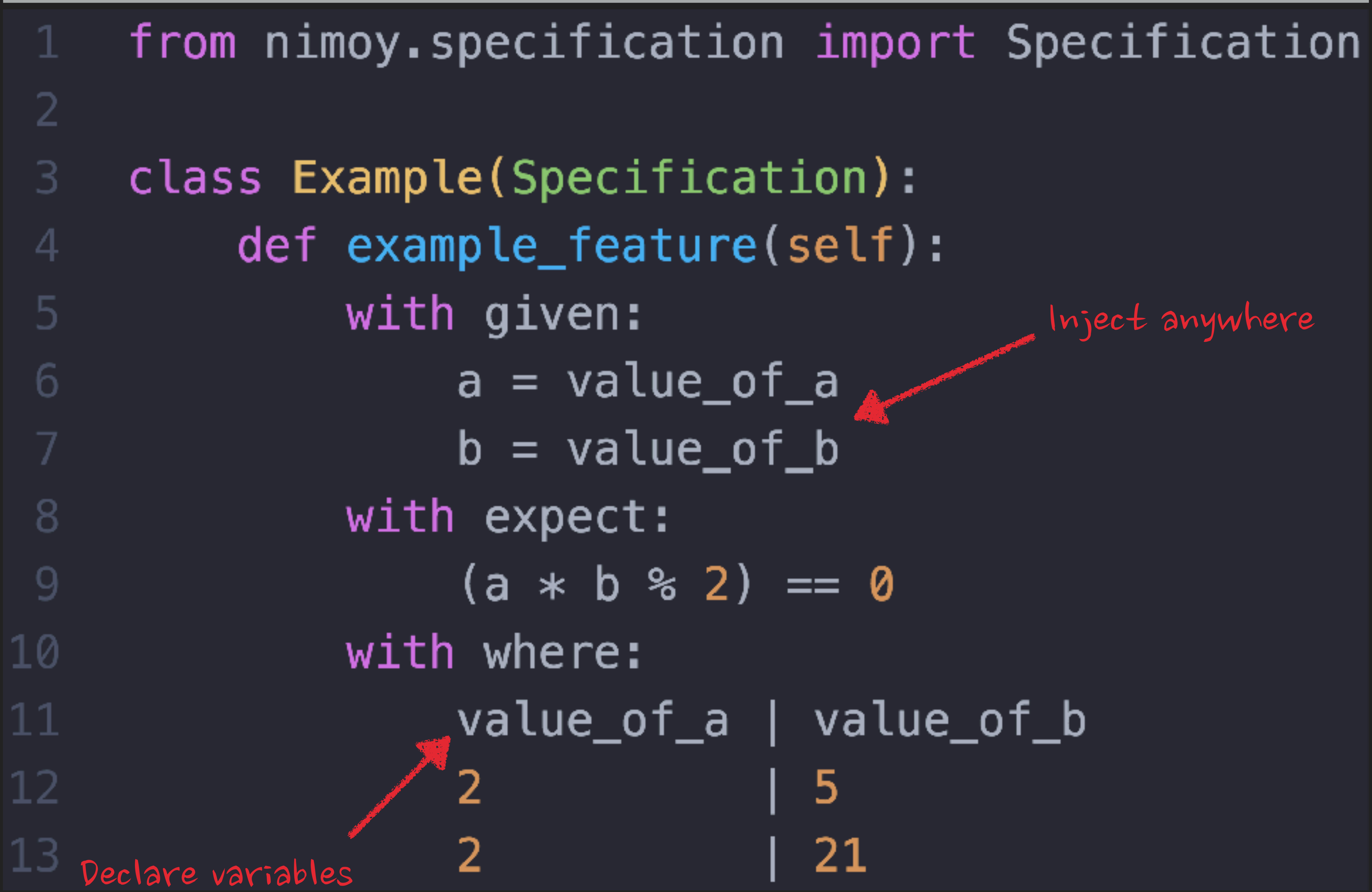
WHERE: VARIABLES EQUAL VALUES

$ VIM TEST_PYTEST_PARAMS.PY



```python
import pytest

@pytest.mark.parametrize("script,result", [
    ("2 * 21", 42),
    ("int(0.1337 * 10000)", 1337),
])
def test_eval(script, result):
    assert eval(script) == result
```

$ VIM NIMOY_MATRIX_SPEC.PY

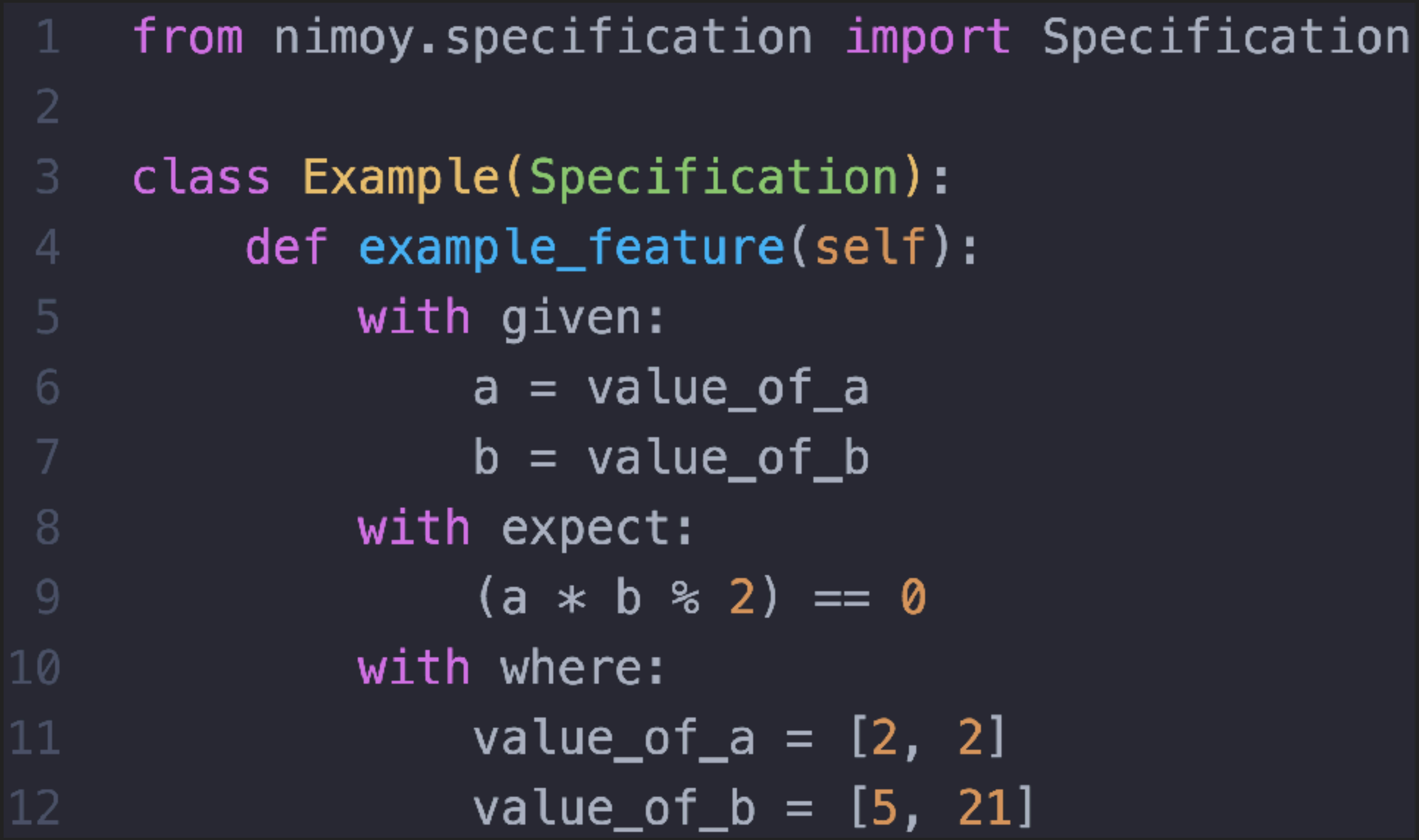


```python
from nimoy.specification import Specification

class Example(Specification):
    def example_feature(self):
        with given:
            a = value_of_a
            b = value_of_b
        with expect:
            (a * b % 2) == 0
        with where:
            value_of_a | value_of_b
            2          | 5
            2          | 21
```

Inject anywhere

Declare variables

$ VIM NIMOY_LIST_SPEC.PY

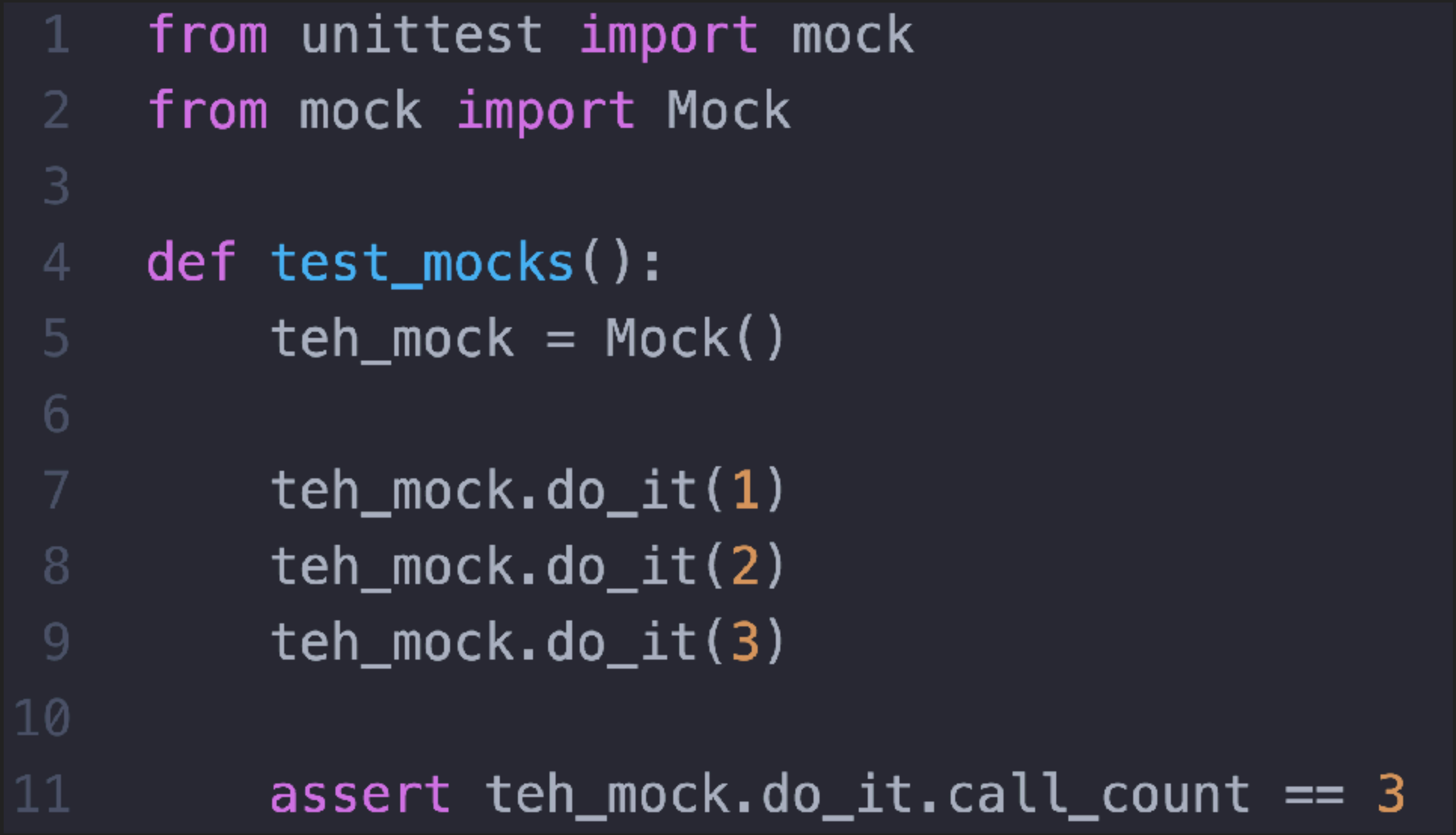


```python
from nimoy.specification import Specification

class Example(Specification):
    def example_feature(self):
        with given:
            a = value_of_a
            b = value_of_b
        with expect:
            (a * b % 2) == 0
        with where:
            value_of_a = [2, 2]
            value_of_b = [5, 21]
```

$ QUESTIONS

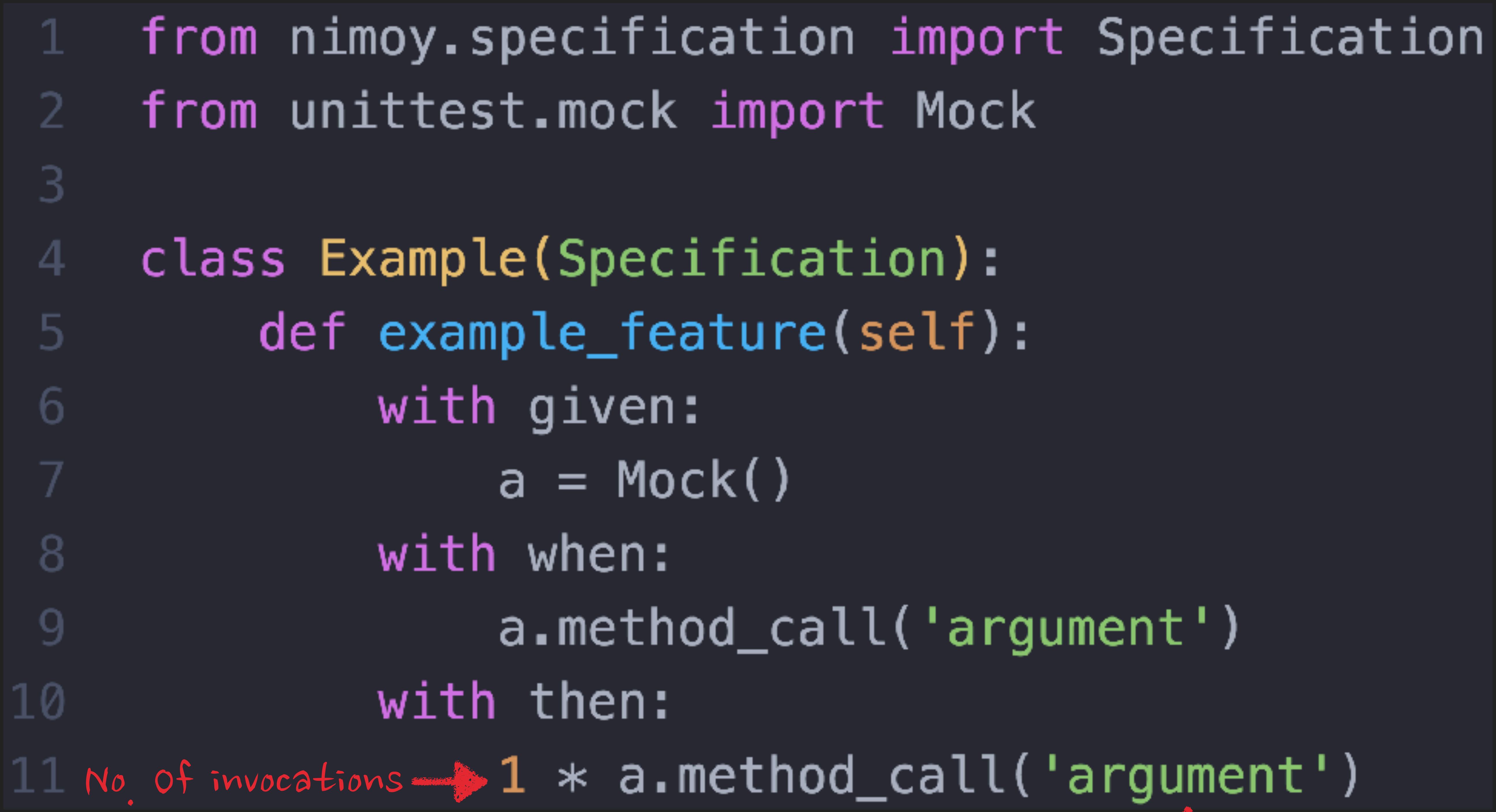
DO YOU MOCK?

$ VIM PYTEST_MOCKS.PY



```python
from unittest import mock
from mock import Mock

def test_mocks():
    teh_mock = Mock()

    teh_mock.do_it(1)
    teh_mock.do_it(2)
    teh_mock.do_it(3)

    assert teh_mock.do_it.call_count == 3
```

$ VIM NIMOY_MOCK_SPEC.PY



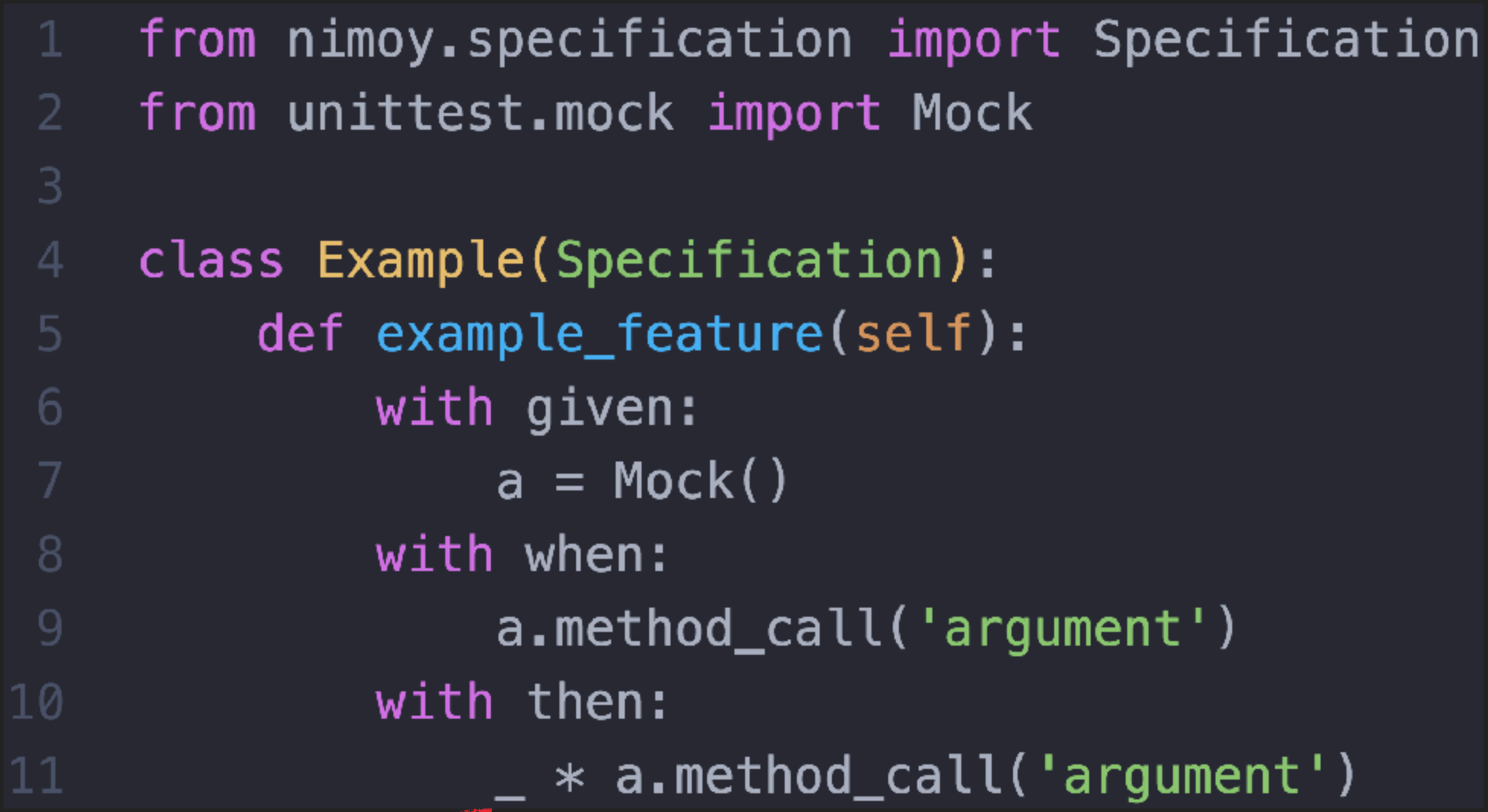
```python
from nimoy.specification import Specification
from unittest.mock import Mock

class Example(Specification):
    def example_feature(self):
        with given:
            a = Mock()
        with when:
            a.method_call('argument')
        with then:
            1 * a.method_call('argument')
```

No. of invocations

Target mock

Expected arguments

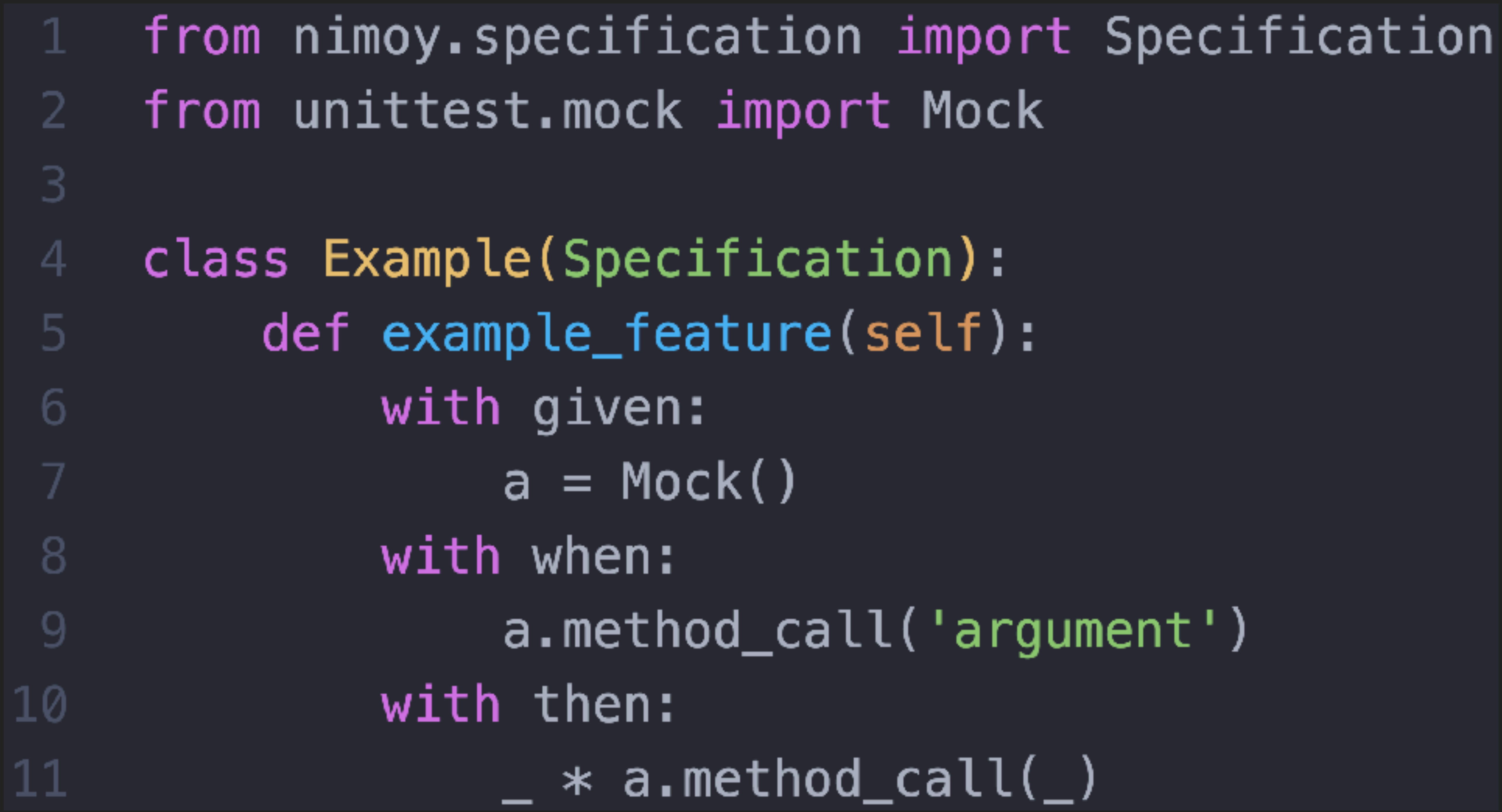
$ VIM NIMOY_MOCK_SPEC.PY



```python
from nimoy.specification import Specification
from unittest.mock import Mock

class Example(Specification):
    def example_feature(self):
        with given:
            a = Mock()
        with when:
            a.method_call('argument')
        with then:
            _ * a.method_call('argument')
```

Any no. of invocations

$ VIM NIMOY_MOCK_SPEC.PY



```python
from nimoy.specification import Specification
from unittest.mock import Mock

class Example(Specification):
    def example_feature(self):
        with given:
            a = Mock()
        with when:
            a.method_call('argument')
        with then:
            _ * a.method_call(_)
```

Any arg value

$ WHICH NIMOY

PYTHON 3

UNITTEST

AST

$ DEMO?

github.com/browncoat-ninjas/nimoy

Version: 0.0.1b4

Search Medium for “Building Nimoy”

QUESTIONS?

THANKS!