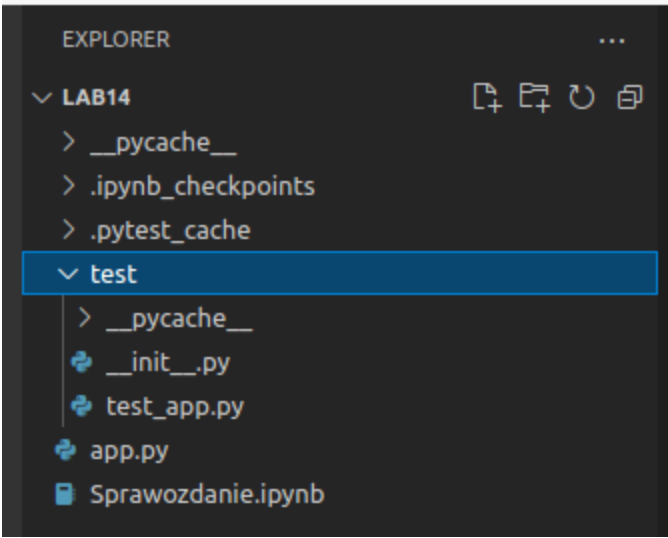


Laboratorium 14 - pytest

W zadaniu będę implementował algorytm sortowania bąbelkowego zgodnie z podejściem Test Driven Development (TDD). Zaczynam od stworzenia odpowiedniej struktury katalogów:



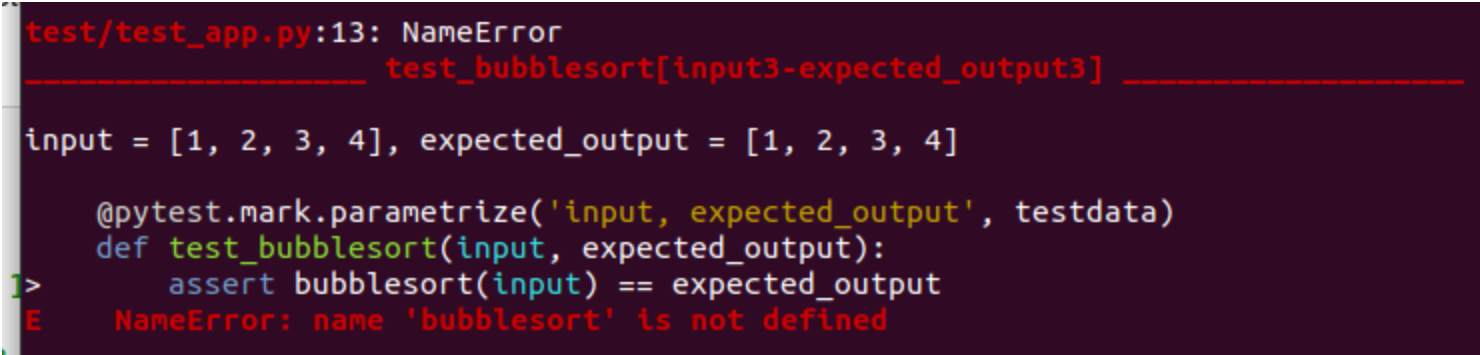
Następny krok to faza czerwona, czyli utworzenie pliku z testami jeszcze przed napisaniem samego algorytmu bubblesort. Zawartość pliku test_app.py składa się z jednego testu sprawdzającego 4 instancje:

```
In [ ]: # tests for file app.py
import pytest
from app import bubblesort

testdata = [([0, 4, 2, 7, 14], [0, 2, 4, 7, 14]),
([5, 4, 3, 2, 1], [1, 2, 3, 4, 5]),
([6, 2, 0, 4, 1, 0, 0, 3], [0, 0, 0, 1, 2, 3, 4, 6]),
([1, 2, 3, 4], [1, 2, 3, 4])]

@pytest.mark.parametrize('input, expected_output', testdata)
def test_bubblesort(input, expected_output):
    assert bubblesort(input) == expected_output
```

Oczywiście na tym etapie uruchomienie testów kończy się ich niepowodzeniem:

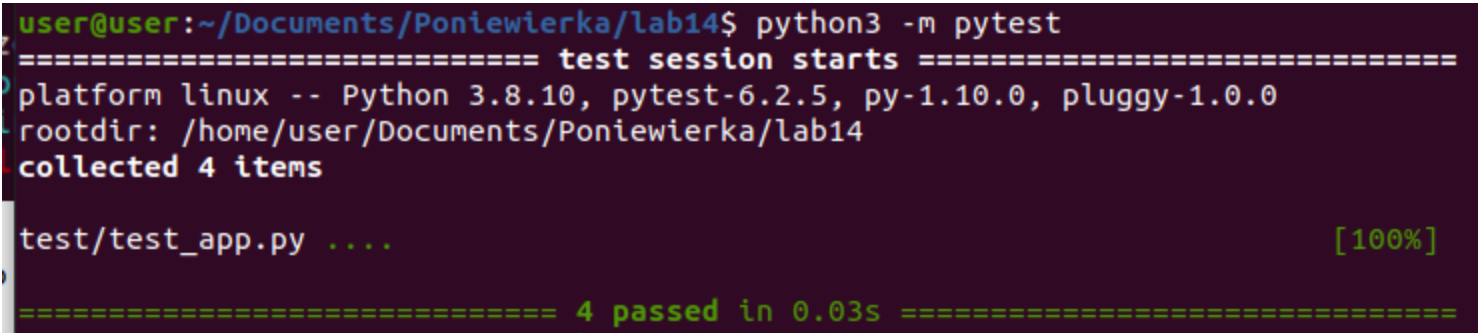


Kolejny krok to faza zielona czyli uzupełnienie pliku app.py o implementację algorytmu bubblesort. Zawatość pliku app.py jest następująca:

```
In [ ]: #bubblesort algorithm

def bubblesort(unsorted):
    l = len(unsorted)
    while l > 1:
        for i in range(0, l-1):
            if unsorted[i] > unsorted[i+1]:
                unsorted[i+1], unsorted[i] = unsorted[i], unsorted[i+1]
        l -= 1
    return unsorted
```

Sprawdzenie działania algorytmu poprzez uruchomienie testów potwierdza jego odpowiednia pracę:



Ostatnia faza to refactor, czyli poprawki do kodu. W moim przydadku kod jest na tyle prosty, że nie jest wymagana jego optymalizacja. Za to dodane zostały komentarze i podpowiedzi typów:

```
In [ ]: # bubblesort algorithm

def bubblesort(unsorted: List) -> List:
    # description: Function for sorting lists of numbers using bubblesort algorithm
    # input: list of unsorted numbers
    # returns: List of sorted numbers

    l = len(unsorted)
    while l > 1:
        for i in range(0, l-1):
            if unsorted[i] > unsorted[i+1]:
                unsorted[i+1], unsorted[i] = unsorted[i], unsorted[i+1]
        l -= 1
    return unsorted
```

Tym etapem zakończyłem pracę.