

## L13.Jakub\_Szczypek , nr. 405912

### Faza RED

Pierwszym naszym zadaniem było napisanie testu, który miał za zadanie wywołać sortowanie bąbelkowe.

```
5 data_to_sort = [  
6     ([1, 2, 3, 4, 5, 21, 7, 4], [1, 1, 0, 10, 5, 7]),  
7     ([20, 1, 5, 3, 6, 7, 2, 9], [5, 3, 2, 1, 10, 9, 8])  
8 ]  
9  
10 @pytest.mark.parametrize('lst, sorted', data_to_sort)  
11 def test_bubble_sort(lst, sorted):  
12  
13     assert bubble_sort(lst) == sorted  
14
```

Jak widać poniżej 2 testy sortowania bąbelkowego nie wykonały się, ponieważ funkcja ta nie została jeszcze przez nas zaimplementowana:

```
Tests failed: 2 of 2 tests - 0 ms  
collecting ... collected 2 items  
  
test_app.py::test_bubble_sort[lst0-sorted0] FAILED [ 50%]  
test_app.py:9 (test_bubble_sort[lst0-sorted0])  
lst = [1, 2, 3, 4, 5, 21, ...], sorted = [1, 1, 0, 10, 5, 7]  
  
    @pytest.mark.parametrize('lst, sorted', data_to_sort)  
    def test_bubble_sort(lst, sorted):  
  
>         assert bubble_sort(lst) == sorted  
E         NameError: name 'bubble_sort' is not defined  
  
test_app.py:13: NameError  
FAILED [100%]  
test_app.py:9 (test_bubble_sort[lst1-sorted1])  
lst = [20, 1, 5, 3, 6, 7, ...], sorted = [5, 3, 2, 1, 10, 9, ...]  
  
    @pytest.mark.parametrize('lst, sorted', data_to_sort)  
    def test_bubble_sort(lst, sorted):  
  
>         assert bubble_sort(lst) == sorted  
E         NameError: name 'bubble_sort' is not defined  
  
test_app.py:13: NameError  
  
test_app.py::test_bubble_sort[lst1-sorted1]  
  
===== 2 failed in 0.16s =====  
  
Process finished with exit code 1
```

## FAZA GREEN

Naszym następnym zadaniem było napisanie funkcji `bubblesort()` w pliku `app.py`, tak aby testy zostały wykonane prawidłowo.

```
4  def bubble_sort(arr):
5      n = len(arr)
6
7      for i in range(n):
8          for j in range(0, n - i - 1):
9              if arr[j] > arr[j + 1]:
10                 arr[j], arr[j + 1] = arr[j + 1], arr[j]
11      return arr
```

Stworzona funkcja pozwala nam na wykonanie się testów.

```
===== test session starts =====
collecting ... collected 2 items

test_app.py::test_bubble_sort[lst0-sorted0] PASSED [ 50%]
test_app.py::test_bubble_sort[lst1-sorted1] PASSED [100%]

===== 2 passed in 0.54s =====

Process finished with exit code 0
```

## FAZA REFACTOR

W fazie Refactor mamy za zadanie ulepszyć istniejący kod. Ulepszania dokonujemy w celu uzyskania bardziej wydajnej metody, czy też aby metoda była bardziej czytelna. Jak widać w kodzie poniżej dodałem zmienną „changed”, dzięki którym sprawdzamy czy podczas pojedynczej iteracji, w całym ciągu wartości doszło przynajmniej raz do zmiany pozycji między dwoma wartościami. Jeśli nie zaszła taka zmiana to można przedwcześnie zakończyć obliczenia. Dzięki tej zmianie nasz kod staje się wydajniejszy.

```
4  def bubble_sort(arr):
5      n = len(arr)
6
7      for i in range(n):
8          changed = False
9          for j in range(0, n - i - 1):
10             if arr[j] > arr[j + 1]:
11                 arr[j], arr[j + 1] = arr[j + 1], arr[j]
12                 changed = True
13             if not changed:
14                 return arr
15         return arr
16
```

Jak widać poniżej testy dalej się wykonują.

```
===== test session starts =====
collecting ... collected 2 items

test_app.py::test_bubble_sort[lst0-sorted0] PASSED [ 50%]
test_app.py::test_bubble_sort[lst1-sorted1] PASSED [100%]

===== 2 passed in 0.45s =====

Process finished with exit code 0
```