Laboratorium 13 – Jakub Wisła

TDD – Test Driven Development

Faza Red:

Napisanie testu, który na początku daje wynik negatywny ze względu na to, że testowana funkcjonalność jeszcze nie istnieje.

```
test\test_app.py FFF                                                                                [100%]
================================================ FAILURES ================================================
_______________________________________________ test_islist _______________________________________________

    def test_islist():
>       got = bubblesort(3)
E       NameError: name 'bubblesort' is not defined

test\test_app.py:3: NameError
_______________________________________________ test_sort1 ________________________________________________

    def test_sort1():
>       got = bubblesort([6, 4, 3, 1, 2])
E       NameError: name 'bubblesort' is not defined

test\test_app.py:8: NameError
_______________________________________________ test_sort2 ________________________________________________

    def test_sort2():
>       got = bubblesort([10, 9, 8, 7, 6, 5, 4, 3, 2, 1])
E       NameError: name 'bubblesort' is not defined

test\test_app.py:13: NameError
======================================== short test summary info =========================================
FAILED test/test_app.py::test_islist - NameError: name 'bubblesort' is not defined
FAILED test/test_app.py::test_sort1 - NameError: name 'bubblesort' is not defined
FAILED test/test_app.py::test_sort2 - NameError: name 'bubblesort' is not defined
============================================ 3 failed in 0.29s ============================================
```

Faza Green:

Napisanie kodu, który implementuje testowaną funkcjonalność. Na tym etapie rozwiązanie nie jest konieczne implementowanie całego rozwiązania - wystarczy, aby przechodziły wszystkie testy.

```
test\test_app.py ...                                                                                [100%]

============================================ 3 passed in 0.06s ============================================
```

Faza Refactor:

Ulepszenie kodu poprzez poprawienie jego czytelności i zmniejszenie złożoności obliczeniowej.

Kod przez fazą Refactor:

```python
def bubblesort(input_array: List) -> List:
    if not isinstance(input_array, List):
        return None
    n = len(input_array)
    copy_list = input_array
    while n > 1:
        for i in range(1, n):
            if copy_list[i - 1] > copy_list[i]:
                copy_list[i - 1], copy_list[i] = copy_list[i], copy_list[i - 1]
        n -= 1
    return copy_list
```

Kod po fazie Refactor:

```python
def bubblesort(input_array: List) -> List:
    if not isinstance(input_array, List): # sprawdzenie czy argument funkcji jest listą
        return None
    n = len(input_array)
    copy_list = input_array
    while n > 1: # pętla, w której następuje sortowanie
        changes = 0 # zmienna wprowadzona, aby pętla zatrzymała się, gdy lista jest posortowana
        for i in range(1, n):
            if copy_list[i - 1] > copy_list[i]:
                copy_list[i - 1], copy_list[i] = copy_list[i], copy_list[i - 1]
                changes += 1
        if changes == 0:
            break # przerwanie pętli w momencie posortowania listy
        n -= 1
    return copy_list
```

Testy po wykonaniu fazy Refactor powinny wszystkie dawać wynik pozytywny, co potwierdza poniższy zrzut ekranu.