**Московский государственный технический**

**университет им. Н.Э. Баумана**

**Факультет ИУ**

**Кафедра ИУ5**

**Курс «Основы информатики»**

**Отчет лабораторной работе №5**

Выполнил студент группы ИУ5-33Б:

Козлов А. А.

Подпись и дата:

Проверил преподаватель каф.:

Гапанюк Ю. Е.

Подпись и дата:

Москва, 2024 г

**Описание задания**

1. Выберите любой фрагмент кода из лабораторных работ 1 или 2 или 3-4.
2. Модифицируйте код таким образом, чтобы он был пригоден для модульного тестирования.
3. Разработайте модульные тесты. В модульных тестах необходимо применить следующие технологии:

* TDD - фреймворк (не менее 3 тестов).
* BDD - фреймворк (не менее 3 тестов).

**Текст программы (BDD – фреймворк)**

**BDD-testing.py**

from behave import given, when, then

def sortArray(array):

    for i in range( len(array)-1):

        for j in range( len(array)-i-1):

            if abs(array[j]) < abs(array[j+1]):

                array[j], array[j+1] = array[j+1], array[j]

    return array

'''First test case'''

@given('I have the array')

def step\_impl\_start1(context):

    context.array = [4, 5, 7, 34, 3, 1, 32, 782]

@when('the array is sorting')

def step\_impl\_do1(context):

    context.sortedArray = sortArray(context.array)

@then('the array is sorted')

def step\_impl\_end1(context):

    assert context.sortedArray == [782, 34, 32, 7, 5, 4, 3, 1]

'''Second test case'''

@given('I have the array with some same numbers')

def step\_impl\_start2(context):

    context.array = [782, 34, 32, 34, 5, 4, 3, 52, 34, 1, 1]

@when('the array with same same numbers is sorting')

def step\_impl\_do2(context):

    context.sortedArray = sortArray(context.array)

@then('the array with same same numbers is sorted')

def step\_impl\_end2(context):

    assert context.sortedArray == [782, 52, 34, 34, 34, 32, 5, 4, 3, 1, 1]

'''Third test case'''

@given('I have the empty array')

def step\_impl\_start3(context):

    context.array = []

@when('the empty array is sorting')

def step\_impl\_do3(context):

    context.sortedArray = sortArray(context.array)

@then('the empty array is sorted')

def stap\_impl\_end3(context):

    assert context.array == context.sortedArray

**BDD.feature**

Feature: sortArray

Scenario: let`s sort our array

Given I have the array

When the array is sorting

Then the array is sorted

Scenario: let`s sort array with some same numbers

Given I have the array with some same numbers

When the array with same same numbers is sorting

Then the array with same same numbers is sorted

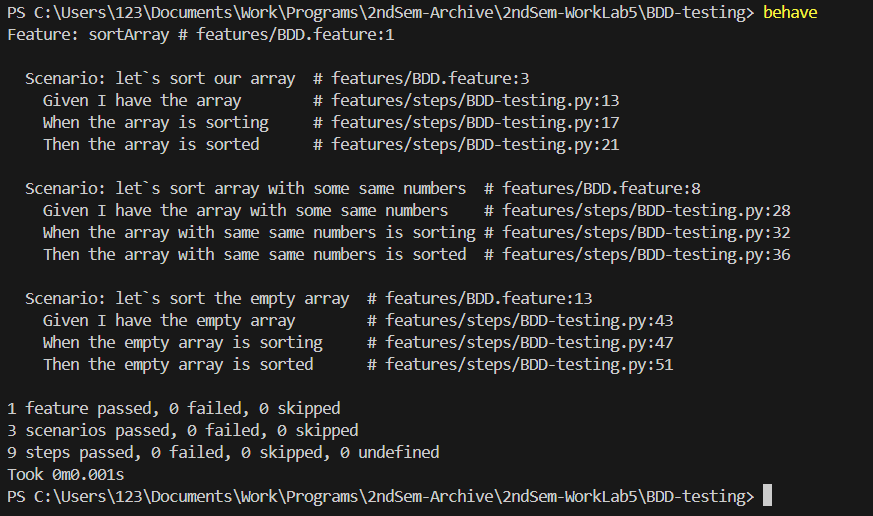
Scenario: let`s sort the empty array

Given I have the empty array

When the empty array is sorting

Then the empty array is sorted

**Выполнение программы**

****

**Текст программы (TDD – фреймворк)**

**TDD-testing.py**

import unittest

def sortArray(array):

    for i in range( len(array)-1):

        for j in range( len(array)-i-1):

            if abs(array[j]) < abs(array[j+1]):

                array[j], array[j+1] = array[j+1], array[j]

    return array

class TestMyFunction(unittest.TestCase):

    firstArr=[1,4,5,7,34,3,32,782]

    secondArr=[-1,-3,-4,-5,-7,-34,-32,-782]

    thirdArr=[1,2,-3,4,-45,23,-12,8,-5]

    def test\_sort\_positive\_numbers(self):

        self.assertEqual(sortArray(self.firstArr),[782,34,32,7,5,4,3,1])

    def test\_sort\_negative\_numbers(self):

        self.assertEqual(sortArray(self.secondArr),[-782,-34,-32,-7,-5,-4,-3,-1])

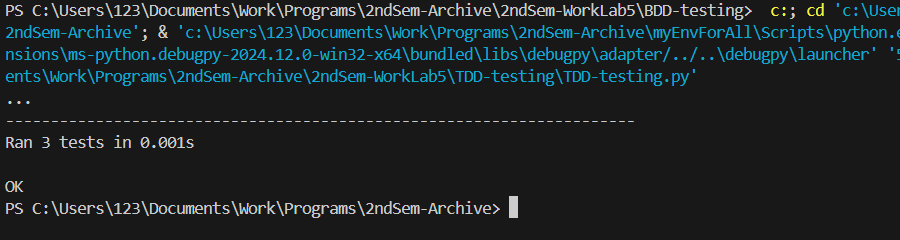
    def test\_sort\_different\_numbers(self):

        self.assertEqual(sortArray(self.thirdArr),[-45,23,-12,8,-5,4,-3,2,1])

if \_\_name\_\_ == '\_\_main\_\_':

    unittest.main()

**Выполнение программы**

****