

МГТУ им. Н.Э. Баумана
Кафедра «Системы обработки информации и
управления»

Рубежный контроль №2
«Базовые компоненты интернет-технологий»

Студентка группы ИУ5-31Б
Котова Анастасия

Преподаватель кафедры ИУ5
Гапанюк Ю.Е.

Москва, 2022

Задание:

Рубежный контроль представляет собой разработку тестов на языке Python.

- 1) Проведите рефакторинг текста программы рубежного контроля №1 таким образом, чтобы он был пригоден для модульного тестирования.
- 2) Для текста программы рубежного контроля №1 создайте модульные тесты с применением TDD - фреймворка (3 теста).

Текст программы:

```
import unittest
from rika1 import HDD, Computer, HDDComputer, A1, A2, A3

class test(unittest.TestCase):

    def setUp(self):
        self.computers = [
            Computer(1, 'компьютер №1'),
            Computer(2, 'компьютер №2'),
            Computer(3, 'компьютер №3'),

            Computer(11, 'компьютер №4'),
            Computer(22, 'компьютер №5'),
            Computer(33, 'компьютер №6'),
        ]

        self.disks = [
            HDD(1, 'Жёсткий диск №1', 25000, 1),
            HDD(2, 'Жёсткий диск №2', 35000, 2),
            HDD(3, 'Жёсткий диск №3', 45000, 3),
            HDD(4, 'Жёсткий диск №4', 35000, 3),
            HDD(5, 'Жёсткий диск №5', 25000, 3),
        ]

        self.disks_computers = [
            HDDComputer(1, 1),
            HDDComputer(2, 2),
            HDDComputer(3, 3),
            HDDComputer(3, 4),
            HDDComputer(3, 5),

            HDDComputer(11, 1),
            HDDComputer(22, 2),
            HDDComputer(33, 3),
            HDDComputer(33, 4),
            HDDComputer(33, 5),
        ]

    def test_A1(self):
```

```

expected_result = [('Жёсткий диск №1', 25000, 'компьютер №1'),
('Жёсткий диск №2', 35000, 'компьютер №2'),
('Жёсткий диск №3', 45000, 'компьютер №3'),
('Жёсткий диск №4', 35000, 'компьютер №3'),
('Жёсткий диск №5', 25000, 'компьютер №3')]

result = A1(self.computers, self.disks)
self.assertEqual(result, expected_result)

def test_A2(self):
    expected_result = [('компьютер №3', 105000),
('компьютер №2', 35000),
('компьютер №1', 25000)]
    result = A2(self.computers, self.disks)
    self.assertEqual(result, expected_result)

def test_A3(self):
    expected_result = {'компьютер №3': ['Жёсткий диск №3', 'Жёсткий диск
№3']}
    result = A3(self.computers, self.disks, '3')
    self.assertEqual(result, expected_result)

if __name__ == '__main__':
    unittest.main()

```

Примеры выполнения:

Ran 3 tests in 0.000s

OK

```

[('Жёсткий диск №1', 25000, 'компьютер №1'), ('Жёсткий диск №2', 35000, 'компьютер №2'), ('Жёсткий диск №3', 45000, 'компьютер №3'), ('Жёсткий диск №4', 35000, 'компьютер №3'), ('Жёсткий диск №5', 25000, 'компьютер №3')]
[('компьютер №3', 105000), ('компьютер №2', 35000), ('компьютер №1', 25000)]
{'компьютер №3': ['Жёсткий диск №3', 'Жёсткий диск №3']}

```