

Рубежный контроль № 2

Полученное задание:

- 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).

Код программы

```
import unittest

from operator import itemgetter

from rk1 import first_task, second_task, third_task


class TestStreetTasks(unittest.TestCase):

    def setUp(self):

        self.one_to_many = [

            ("Bataskiy Proezd", 43, 1),

            ("Lugovoi proezd", 21, 25),

            ("Maryinskiy bulvar", 61, 30),

            ("Rahovo", 38, 30),

            ("Leninskiy Prospekt", 47, 1),

            ("Ulica Lenina", 42, 1)

        ]

        self.many_to_many = [

            ("Bataskiy Proezd", 43, 1),

            ("Lugovoi proezd", 21, 25),

            ("Maryinskiy bulvar", 61, 30),

            ("Rahovo", 38, 30),

            ("Leninskiy Prospekt", 47, 1),

        ]
```

```
def test_first_task(self):
    expected = [
        ("Bataskiy Proezd", 43, 1),
        ("Leninskiy Prospekt", 47, 1),
        ("Ulica Lenina", 42, 1),
        ("Lugovoi proezd", 21, 25),
        ("Maryinskiy bulvar", 61, 30),
        ("Rahovo", 38, 30)
    ]
    result = first_task(self.one_to_many)
    self.assertEqual(result, expected)
```

```
def test_second_task(self):
    expected = [
        (1, 3),
        (30, 2),
        (25, 1)
    ]
    result = second_task(self.one_to_many)
    self.assertEqual(result, expected)
```

```
def test_third_task(self):
    expected = [
        ('Lugovoi proezd', 25),
        ('Maryinskiy bulvar', 30)
    ]
    result = third_task(self.many_to_many, '1')
    self.assertEqual(result, expected)
```

```
if __name__ == "__main__":
    unittest.main()
```

Результаты выполнения

```
Ran 3 tests in 0.004s
```

```
OK
```

```
Process finished with exit code 0
```