

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).

Текст программы

```
import rk1
from operator import itemgetter
import unittest

class TestSolving(unittest.TestCase):
    def test_first_task_method(self):
        test_list = [('basic', 'basic', 'basic'), ('auto', 'auto', 'auto'),
('color', 'color', 'color')]
        self.assertEqual(rk1.first_task(test_list), sorted(test_list,
key=itemgetter(0)))

    def test_second_task_method(self):
        test_list = [('Driver1', 'id1', 'Park1'),
('Driver2', 'id2', 'Park2'),
('Driver3', 'id3', 'Park1'),
('Driver4', 'id4', 'Park3'),
('Driver5', 'id5', 'Park1')]
        self.assertEqual(rk1.second_task(test_list), [('Park1', 3),('Park2',
1),('Park3', 1)])

    def test_third_task_method(self):
        test_list = [
('V.V.Petr', 'id1', 'Park1'),
('P.P.Ivanov', 'id2', 'Park2'),
('A.A.Sidorov', 'id3', 'Park3'),
('K.K.Emelyanov', 'id4', 'Park4'),
('T.T.Smirn', 'id5', 'Park1')
]
        self.assertEqual(rk1.third_task(test_list,'ov'), [('P.P.Ivanov',
'Park2'),('A.A.Sidorov', 'Park3'),('K.K.Emelyanov', 'Park4')])

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

Вывод программы:

...

Ran 3 tests in 0.001s

OK

PS C:\Users\ann18\OneDrive\Desktop\bmstu\3sem\PCPL\ek1>