

## Условия рубежного контроля №2 по курсу Пик ЯП

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 sys
sys.path.append('./RK1')
import RK1
from operator import itemgetter
import unittest

class Test(unittest.TestCase):
    def test_first_task(self):
        test_list = [("Ivanov", 1, "A"), ("Petrov", 2, "B"), ("Sidorov", 3, "C")]
        self.assertEqual(RK1.first_task(test_list), sorted(test_list,
key=itemgetter(0)))

    def test_second_task(self):
        test_list = [("Ivanov", 1, "A"), ("Petrov", 2, "B"), ("Sidorov", 3, "C")]
        self.assertEqual(RK1.second_task(test_list), [("A", 1), ("B", 1), ("C",
1)])

    def test_third_task(self):
        test_list = [("Ivanov", "A"), ("Petrov", "B"), ("Sidorov", "C")]
        self.assertEqual(RK1.third_task(test_list, "ov"), [("Ivanov", "A"),
("Petrov", "B"), ("Sidorov", "C")])

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

### Результат выполнения программы

```
● tal3nt3d@tal3nt3d:~/PCPL_2024$ python3.10 RK2/RK2.py
...
-----
Ran 3 tests in 0.001s

OK
```