

Отчет по РК2 по дисциплине
“Парадигмы и конструкции языков программирования”

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).

Рефакторинг кода заключался в перенос логики задания в функции для последующего тестирования

Текст программы (unit_test.py):

```
import main
```

```
from operator import itemgetter
```

```
import unittest
```

```
class TestMainMethods(unittest.TestCase):
```

```
    def test_func_1(self):
```

```
        test_list = [
```

```
            ('review', 300, 'document_1'),
```

```
            ('results', 100, 'document_2'),
```

```
            ('introduction', 200, 'document_3'),
```

```
        ]
```

```
        result = main.func_1(test_list)
```

```
        reference = [item for item in test_list if item[0].startswith('r')]
```

```
        self.assertEqual(result, reference)
```

```
    def test_func_2(self):
```

```
        test_list = [
```

```
            ('introduction', 200, 'doc_1'),
```

```
            ('review', 300, 'doc_1'),
```

```

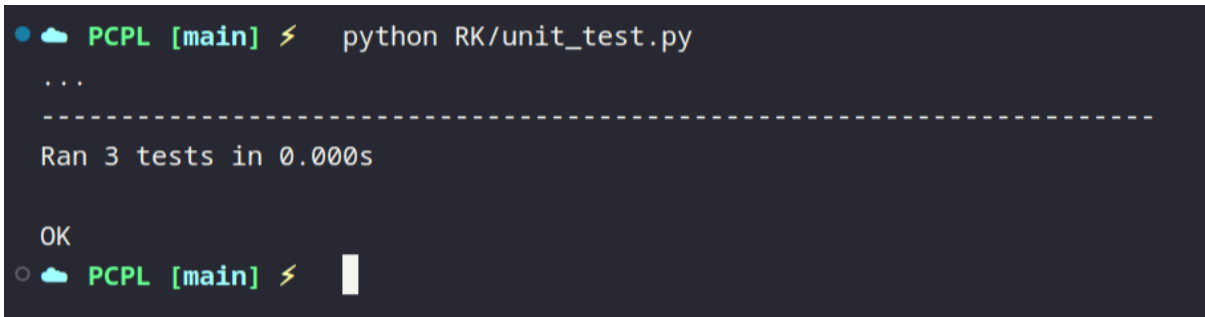
        ('results', 100, 'doc_2'),
    ]
    result = main.func_2(test_list)
    reference = [('doc_2', 100), ('doc_1', 200)]
    self.assertEqual(result, reference)

def test_func_3(self):
    test_list = [
        ('results', 100, 'doc_2'),
        ('review', 300, 'doc_1'),
        ('introduction', 200, 'doc_3'),
    ]
    result = main.func_3(test_list)
    reference = sorted(test_list, key=itemgetter(0))
    self.assertEqual(result, reference)

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

```

Результат выполнения:



```

● ☁ PCPL [main] ⚡ python RK/unit_test.py
...
-----
Ran 3 tests in 0.000s

OK
○ ☁ PCPL [main] ⚡ █

```