

Рубежный контроль №2

- 1) Рефакторинг текста программы рубежного контроля №1, чтобы он был пригоден для модульного тестирования:

```
from operator import itemgetter
class Line:
    def __init__(self, id, name, val, dep_id):
        self.id = id
        self.name = name
        self.val = val
        self.dep_id = dep_id
class Table:
    def __init__(self, id, name):
        self.id = id
        self.name = name
class LineTable:
    def __init__(self, dep_id, line_id):
        self.dep_id = dep_id
        self.line_id = line_id
tables = [
    Table(1, 'Анализируемая таблица'),
    Table(2, 'В ожидании'),
    Table(3, 'Таблица пропущенных'),
    Table(11, 'Анализируемая таблица 2'),
    Table(22, 'В ожидании 2'),
    Table(33, 'Таблица пропущенных 2')
]
lines = [
    Line(1, 'Рыбак', 604397, 1),
    Line(2, 'Шестерочка', 565819, 2),
    Line(3, 'Диски', 124781, 3),
    Line(4, 'Сложно - и точка', 617325, 4),
    Line(5, 'Ну-Ну', 981458, 5)
]
lines_tables = [
    LineTable(1, 1),
    LineTable(2, 2),
    LineTable(3, 3),
    LineTable(3, 4),
    LineTable(3, 5),
    LineTable(11, 1),
    LineTable(22, 2),
    LineTable(33, 3),
    LineTable(33, 4),
    LineTable(33, 5)
]
def one_to_many_func(tables, lines):
    return [(e.name, e.val, d.name) for d in tables for e in lines if
e.dep_id == d.id]
```

```

def many_to_many_func(tables, lines_tables, lines):
    many_to_many_temp = [(d.name, ed.dep_id, ed.line_id) for d in tables for
ed in lines_tables if d.id == ed.dep_id]
    return [(l.name, l.val, dep_name) for dep_name, dep_id, line_id in
many_to_many_temp for l in lines if l.id == line_id]
def starting_a(one_to_many):
    return list(filter(lambda i: i[2][0] == 'A', one_to_many))
def max_val_table(tables, one_to_many):
    ms = []
    for t in tables:
        t_lines = list(filter(lambda i: i[2] == t.name, one_to_many))
        mx = max((line[1] for line in t_lines), default=0)
        if t_lines:
            ms.append([mx, t_lines[0][2]])
    return sorted(ms, key=lambda x: x[0], reverse=True)
def main():
    one_to_many = one_to_many_func(tables, lines)
    print('Задание Г1')
    t_lines = starting_a(one_to_many)
    print(t_lines)
    if t_lines:
        names = [name for name, _, _ in t_lines]
        print(names)
    print('\nЗадание Г2')
    sms = max_val_table(tables, one_to_many)
    print(sms)
    print('\nЗадание Г3')
    many_to_many = many_to_many_func(tables, lines_tables, lines)
    print(sorted(list((x[0], x[2]) for x in many_to_many),
key=itemgetter(1)))
if __name__ == '__main__':
    main()

```

2) Модульные тесты с применением TDD - фреймворка (3 теста):

```

import unittest
class TestMappingFunctions(unittest.TestCase):
    def setUp(self):
        self.tables = [
            Table(1, 'Анализируемая таблица'),
            Table(2, 'В ожидании'),
            Table(3, 'Таблица пропущенных')
        ]
        self.lines = [
            Line(1, 'PyМак', 604397, 1),
            Line(2, 'Шестерочка', 565819, 2)
        ]
        self.lines_tables = [
            LineTable(1, 1),
            LineTable(2, 2)
        ]
    def test_one_to_many_func(self):

```

```

        result = one_to_many_func(self.tables, self.lines)
        expected = [('ПуМак', 604397, 'Анализируемая таблица'),
                    ('Шестерочка', 565819, 'В ожидании')]
        self.assertEqual(result, expected)
    def test_many_to_many_func(self):
        result = many_to_many_func(self.tables, self.lines_tables,
self.lines)
        expected = [('ПуМак', 604397, 'Анализируемая таблица'),
                    ('Шестерочка', 565819, 'В ожидании')]
        self.assertEqual(result, expected)
    def test_max_val_table(self):
        one_to_many_data = [('ПуМак', 604397, 'Анализируемая таблица'),
                            ('Шестерочка', 565819, 'В ожидании')]
        result = max_val_table(self.tables, one_to_many_data)
        expected = [[604397, 'Анализируемая таблица'],
                    [565819, 'В ожидании']]
        self.assertEqual(result, expected)
if __name__ == '__main__':
    unittest.main()

```

Результаты работы программы и тестов:

Задание Г1

```

[('ПуМак', 604397, 'Анализируемая таблица')]
['ПуМак']

```

Задание Г2

```

[[604397, 'Анализируемая таблица'], [565819, 'В ожидании'], [124781,
'Таблица пропущенных']]

```

Задание Г3

```

[('ПуМак', 'Анализируемая таблица'), ('ПуМак', 'Анализируемая таблица
2'), ('Шестерочка', 'В ожидании'), ('Шестерочка', 'В ожидании 2'), ('Диски',
'Таблица пропущенных'), ('Сложно - и точка', 'Таблица пропущенных'), ('Ну-
Ну', 'Таблица пропущенных'), ('Диски', 'Таблица пропущенных 2'), ('Сложно - и
точка', 'Таблица пропущенных 2'), ('Ну-Ну', 'Таблица пропущенных 2')]
...

```

Ran 3 tests in 0.003s

OK