Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по рубежному контролю №2 по курсу БКИТ

Выполнил:
студент группы ИУ5-34Б
Жданова Яна
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

2022

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:

main.py

```
from operator import itemgetter

class Musician:
    """Музыкант"""
    def __init__(self, id, surname, salary, orch_id):
        self.id = id
        self.surname = surname
        self.salary = salary
        self.orch_id = orch_id

class Orchestra:
    """Оркестр"""
    def __init__(self, id, name):
        self.id = id
        self.name = name


class MusOrch:
    """
    'Музыканты оркестра' для реализации
    связи многие-ко-многим
    """

    def __init__(self, mus_id, orch_id):
        self.mus_id = mus_id
        self.orch_id = orch_id


# Оркестры
orchestras = [
    Orchestra(1, 'Дуделки и свистелки'),
    Orchestra(2, 'Стучалки и бренчалки'),
    Orchestra(3, 'Баян или плохая шутка'),
    Orchestra(4, 'Ансамбль песнопелки и плясалки')
]

# Музыканты
musicians = [
    Musician(1, 'Соколов', 500000, 2),
    Musician(2, 'Жданова', 250000, 4),
    Musician(3, 'Тумановский', 300000, 3),
    Musician(4, 'Савицкая', 150000, 4),
```

```
    Musician(5, 'Теряева', 4050, 1)
]

mus_orches = [
    MusOrch(1, 1),
    MusOrch(2, 2),
    MusOrch(3, 3),
    MusOrch(4, 4),
    MusOrch(5, 1),

    MusOrch(1, 4),
    MusOrch(2, 3),
    MusOrch(3, 2),
    MusOrch(4, 2),
    MusOrch(5, 4),
]

# Соединение данных один-ко-многим
one_to_many = [(mus.surname, mus.salary, orch.name)
               for orch in orchestras
               for mus in musicians
               if mus.orch_id == orch.id]

# Соединение данных многие-ко-многим
many_to_many_temp = [(orch.name, mu_or.orch_id, mu_or.mus_id)
                     for orch in orchestras
                     for mu_or in mus_orches
                     if orch.id == mu_or.orch_id]

many_to_many = [(mus.surname, mus.salary, orch_name)
               for orch_name, orch_id, mus_id in many_to_many_temp
               for mus in musicians if mus.id == mus_id]

def task_1(one_to_many):
    print ('Задание Г1')
    res_1 = {}
    for orch in orchestras:
        if orch.name[0] == 'A':
            orch_muses = list(filter(lambda i: i[2] == orch.name, one_to_many))
            orch_muses_names = [x for x, _, _ in orch_muses]
            res_1[orch.name] = orch_muses_names
    print (res_1)
    return res_1

def task_2(one_to_many):
    print ('Задание Г2')
    res_2 = []
    for orch in orchestras:
        orch_muses = list(filter(lambda i: i[2] == orch.name, one_to_many))
        if len(orch_muses) > 0:
            orch_salaries = [sal for _, sal, _ in orch_muses]
```

```
            orch_salaries_max = max(orch_salaries)
            res_2.append((orch.name, orch_salaries_max))
    res_2_sort = sorted(res_2, key=itemgetter(1))
    for i in res_2_sort:
        print(i)
    return res_2_sort

def task_3(many_to_many):
    print ('Задание Г3')
    res_3_sort = sorted (many_to_many, key=itemgetter(2))
    for i in res_3_sort:
        print(i)
    return res_3_sort
```

tests.py

```
import unittest
from main import *

class TestingTasks(unittest.TestCase):

    def test_task_1(self):
        self.assertEqual(task_1(one_to_many), {'Ансамбль песнопелки и плясалки':
['Жданова', 'Савицкая']})

    def test_task_2(self):
        self.assertEqual(task_2(one_to_many), [('Дуделки и свистелки', 4050),
                                               ('Ансамбль песнопелки и плясалки',
250000),
                                               ('Баян или плохая шутка', 300000),
                                               ('Стучалки и бренчалки', 500000)])

    def test_task_3(self):
        self.assertEqual(task_3(many_to_many), [('Савицкая', 150000, 'Ансамбль
песнопелки и плясалки'),
                                                ('Соколов', 500000, 'Ансамбль
песнопелки и плясалки'),
                                                ('Теряева', 4050, 'Ансамбль
песнопелки и плясалки'),
                                                ('Тумановский', 300000, 'Баян или
плохая шутка'),
                                                ('Жданова', 250000, 'Баян или
плохая шутка'),
                                                ('Соколов', 500000, 'Дуделки и
свистелки'),
                                                ('Теряева', 4050, 'Дуделки и
свистелки'),
                                                ('Жданова', 250000, 'Стучалки и
бренчалки'),
                                                ('Тумановский', 300000, 'Стучалки
и бренчалки'),
```

```
                                                ('Савицкая', 150000, 'Стучалки и
бренчалки')])

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

Результат выполнения программы:

```
ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ

PS C:\Users\user\Desktop\РК2 БКИТ>  & 'C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python37_64\
bugpy\adapter/../..\debugpy\launcher' '51236' '--' 'c:\Users\user\Desktop\РК2 БКИТ\tests.py'
Задание Г1
{'Ансамбль песнопелки и плясалки': ['Жданова', 'Савицкая']}
.Задание Г2
('Дуделки и свистелки', 4050)
('Ансамбль песнопелки и плясалки', 250000)
('Баян или плохая шутка', 300000)
('Стучалки и бренчалки', 500000)
.Задание Г3
('Савицкая', 150000, 'Ансамбль песнопелки и плясалки')
('Соколов', 500000, 'Ансамбль песнопелки и плясалки')
('Теряева', 4050, 'Ансамбль песнопелки и плясалки')
('Тумановский', 300000, 'Баян или плохая шутка')
('Жданова', 250000, 'Баян или плохая шутка')
('Соколов', 500000, 'Дуделки и свистелки')
('Теряева', 4050, 'Дуделки и свистелки')
('Жданова', 250000, 'Стучалки и бренчалки')
('Тумановский', 300000, 'Стучалки и бренчалки')
('Савицкая', 150000, 'Стучалки и бренчалки')
.
----------------------------------------------------------------------
Ran 3 tests in 0.013s

OK
PS C:\Users\user\Desktop\РК2 БКИТ>
```