Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Рубежный контроль №2

Вариант №18

Выполнил:

Студент ИУ5-34Б Тумановский А. М.

Подпись и дата: _______________

Проверил:

Преподаватель каф. ИУ5 Гапанюк Ю. Е.

Подпись и дата: ______________

Москва, 2023 г.

Условия рубежного контроля №2 по курсу ПиК ЯП

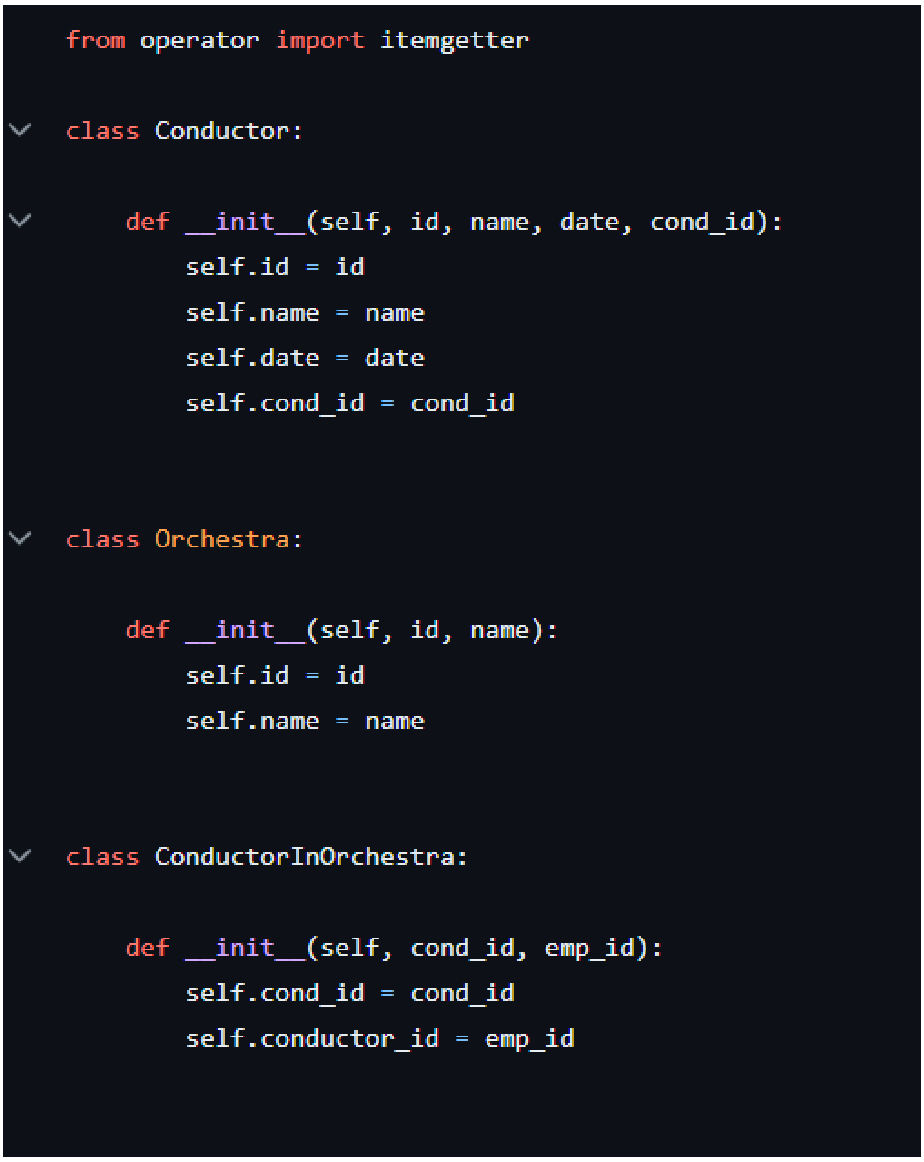
Рубежный контроль представляет собой разработку тестов на языке Python.

1) Проведите рефакторинг текста программы рубежного контроля №1 таким образом, чтобы он был пригоден для модульного тестирования.

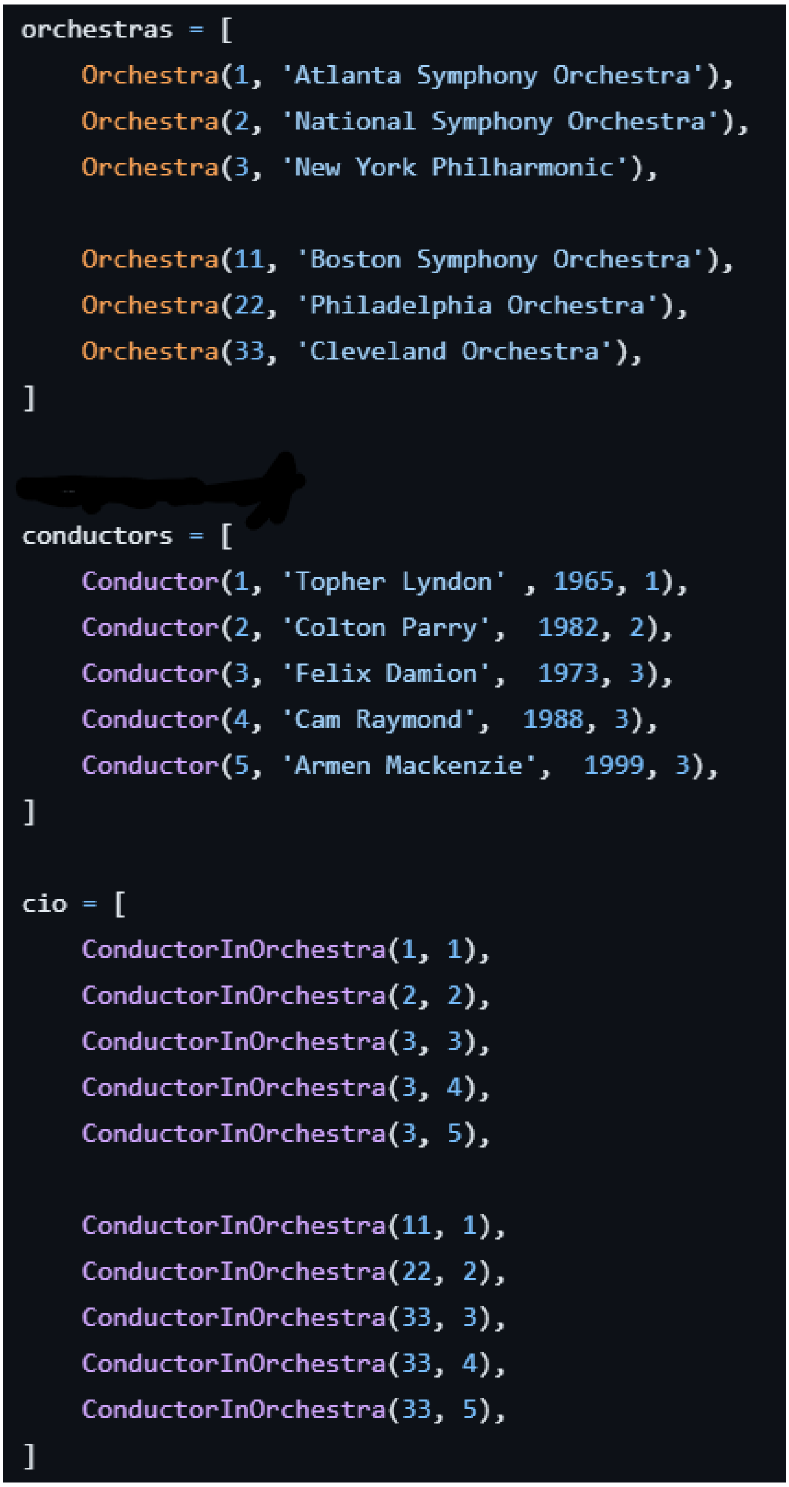
2) Для текста программы рубежного контроля №1 создайте модульные тесты с применением TDD - фреймворка (3 теста).

Текст программы:

main.py:



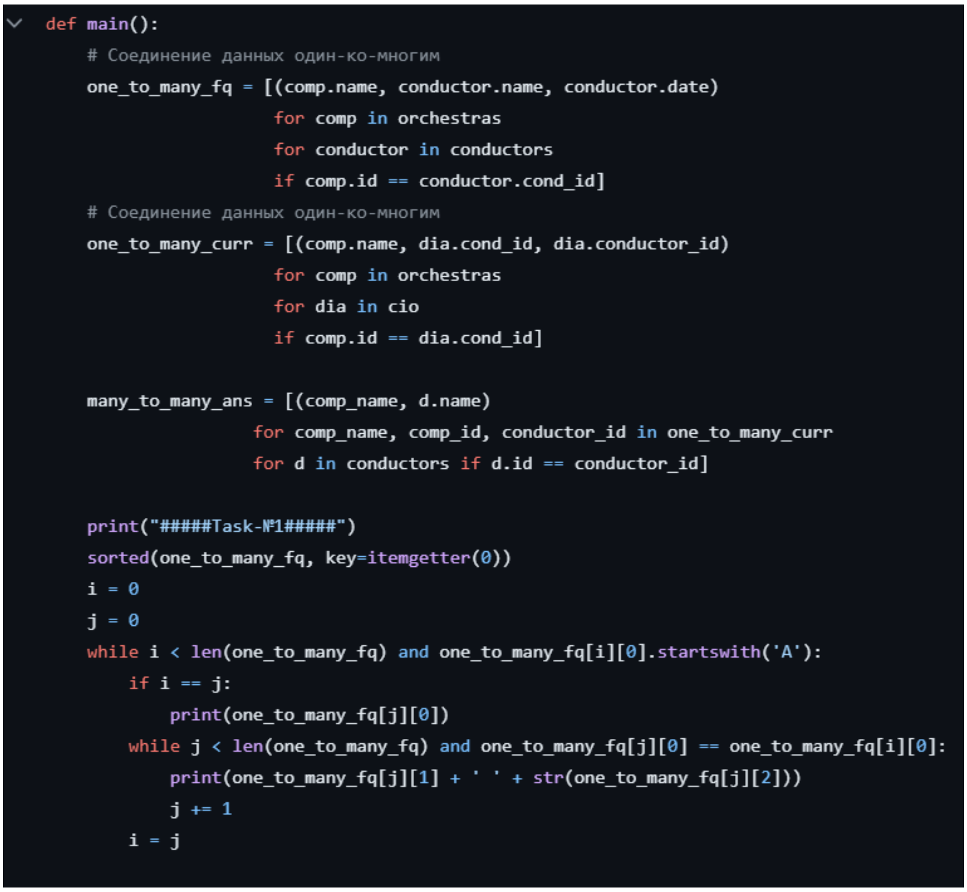
```python
from operator import itemgetter

class Conductor:

    def __init__(self, id, name, date, cond_id):
        self.id = id
        self.name = name
        self.date = date
        self.cond_id = cond_id


class Orchestra:

    def __init__(self, id, name):
        self.id = id
        self.name = name


class ConductorInOrchestra:

    def __init__(self, cond_id, emp_id):
        self.cond_id = cond_id
        self.conductor_id = emp_id
```



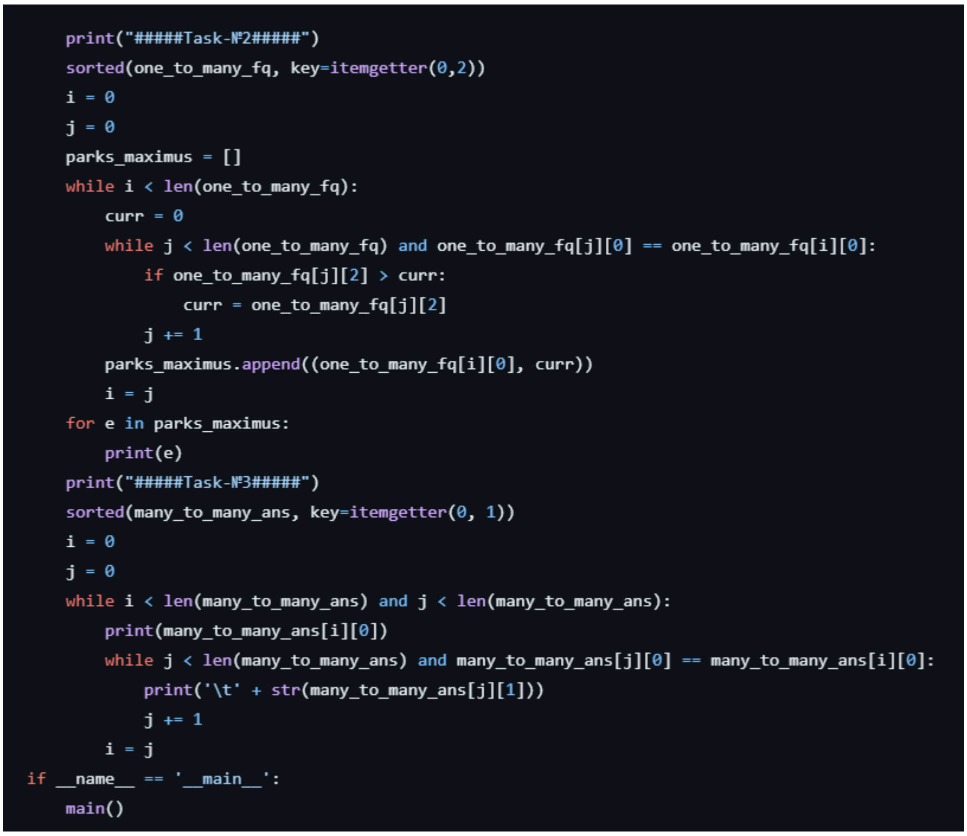
```
orchestras = [
    Orchestra(1, 'Atlanta Symphony Orchestra'),
    Orchestra(2, 'National Symphony Orchestra'),
    Orchestra(3, 'New York Philharmonic'),

    Orchestra(11, 'Boston Symphony Orchestra'),
    Orchestra(22, 'Philadelphia Orchestra'),
    Orchestra(33, 'Cleveland Orchestra'),
]
```

[redacted]

```
conductors = [
    Conductor(1, 'Topher Lyndon' , 1965, 1),
    Conductor(2, 'Colton Parry',  1982, 2),
    Conductor(3, 'Felix Damion',  1973, 3),
    Conductor(4, 'Cam Raymond',  1988, 3),
    Conductor(5, 'Armen Mackenzie',  1999, 3),
]

cio = [
    ConductorInOrchestra(1, 1),
    ConductorInOrchestra(2, 2),
    ConductorInOrchestra(3, 3),
    ConductorInOrchestra(3, 4),
    ConductorInOrchestra(3, 5),

    ConductorInOrchestra(11, 1),
    ConductorInOrchestra(22, 2),
    ConductorInOrchestra(33, 3),
    ConductorInOrchestra(33, 4),
    ConductorInOrchestra(33, 5),
]
```

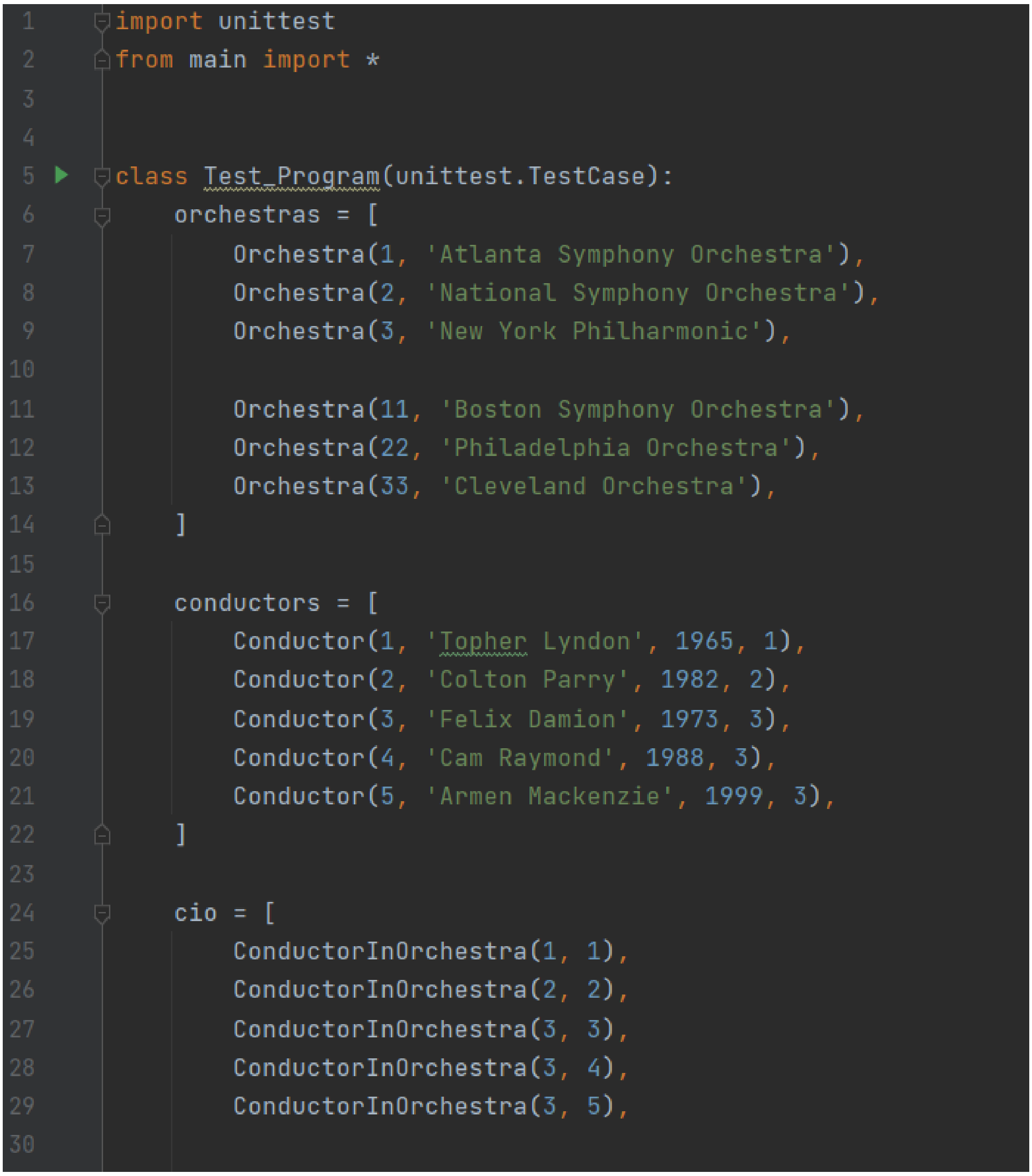


```python
def main():
    # Соединение данных один-ко-многим
    one_to_many_fq = [(comp.name, conductor.name, conductor.date)
                      for comp in orchestras
                      for conductor in conductors
                      if comp.id == conductor.cond_id]
    # Соединение данных один-ко-многим
    one_to_many_curr = [(comp.name, dia.cond_id, dia.conductor_id)
                      for comp in orchestras
                      for dia in cio
                      if comp.id == dia.cond_id]

    many_to_many_ans = [(comp_name, d.name)
                    for comp_name, comp_id, conductor_id in one_to_many_curr
                    for d in conductors if d.id == conductor_id]

    print("#####Task-№1#####")
    sorted(one_to_many_fq, key=itemgetter(0))
    i = 0
    j = 0
    while i < len(one_to_many_fq) and one_to_many_fq[i][0].startswith('A'):
        if i == j:
            print(one_to_many_fq[j][0])
        while j < len(one_to_many_fq) and one_to_many_fq[j][0] == one_to_many_fq[i][0]:
            print(one_to_many_fq[j][1] + ' ' + str(one_to_many_fq[j][2]))
            j += 1
        i = j
```

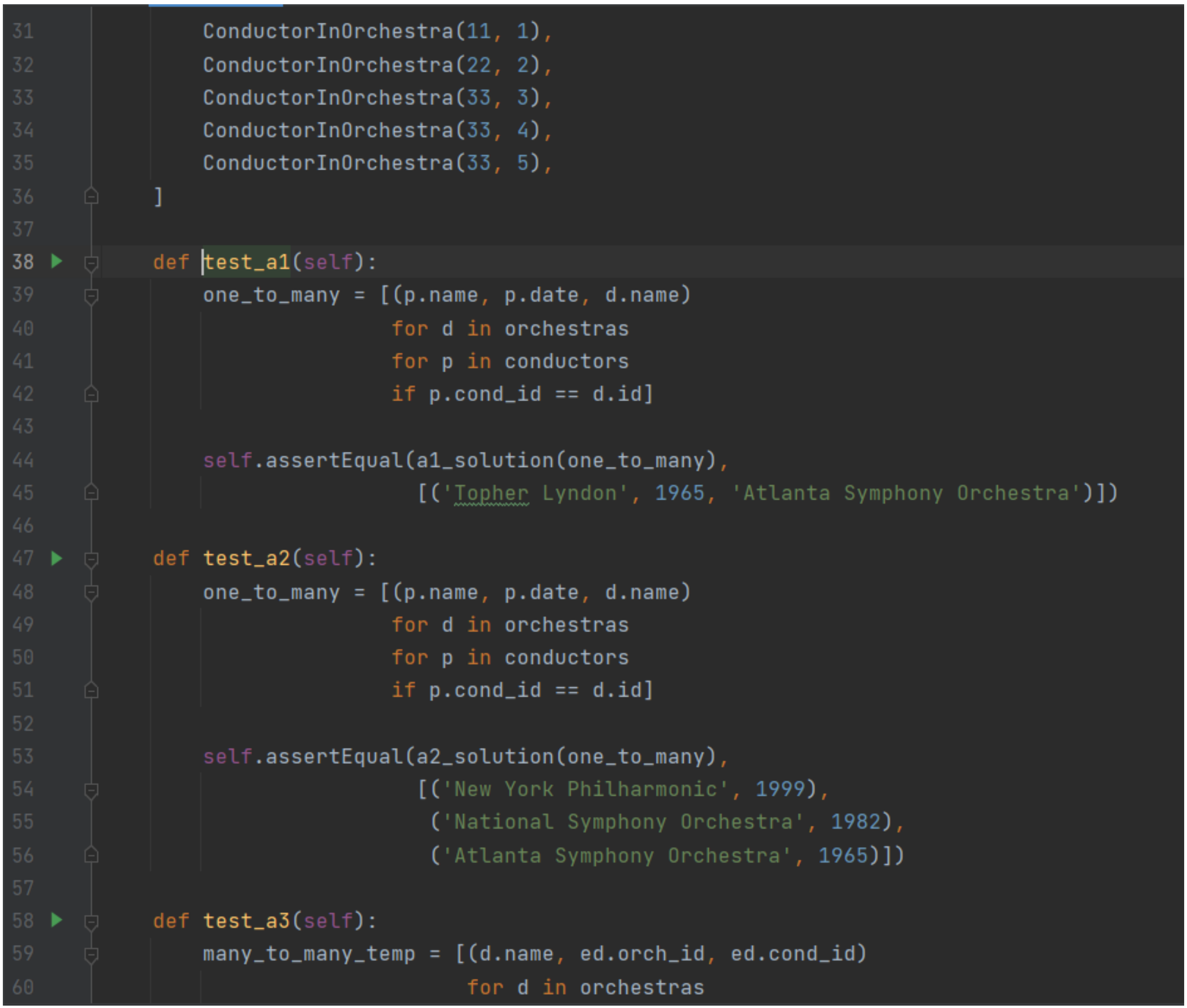


```python
    print("#####Task-№2#####")
    sorted(one_to_many_fq, key=itemgetter(0,2))
    i = 0
    j = 0
    parks_maximus = []
    while i < len(one_to_many_fq):
        curr = 0
        while j < len(one_to_many_fq) and one_to_many_fq[j][0] == one_to_many_fq[i][0]:
            if one_to_many_fq[j][2] > curr:
                curr = one_to_many_fq[j][2]
            j += 1
        parks_maximus.append((one_to_many_fq[i][0], curr))
        i = j
    for e in parks_maximus:
        print(e)
    print("#####Task-№3#####")
    sorted(many_to_many_ans, key=itemgetter(0, 1))
    i = 0
    j = 0
    while i < len(many_to_many_ans) and j < len(many_to_many_ans):
        print(many_to_many_ans[i][0])
        while j < len(many_to_many_ans) and many_to_many_ans[j][0] == many_to_many_ans[i][0]:
            print('\t' + str(many_to_many_ans[j][1]))
            j += 1
        i = j
if __name__ == '__main__':
    main()
```

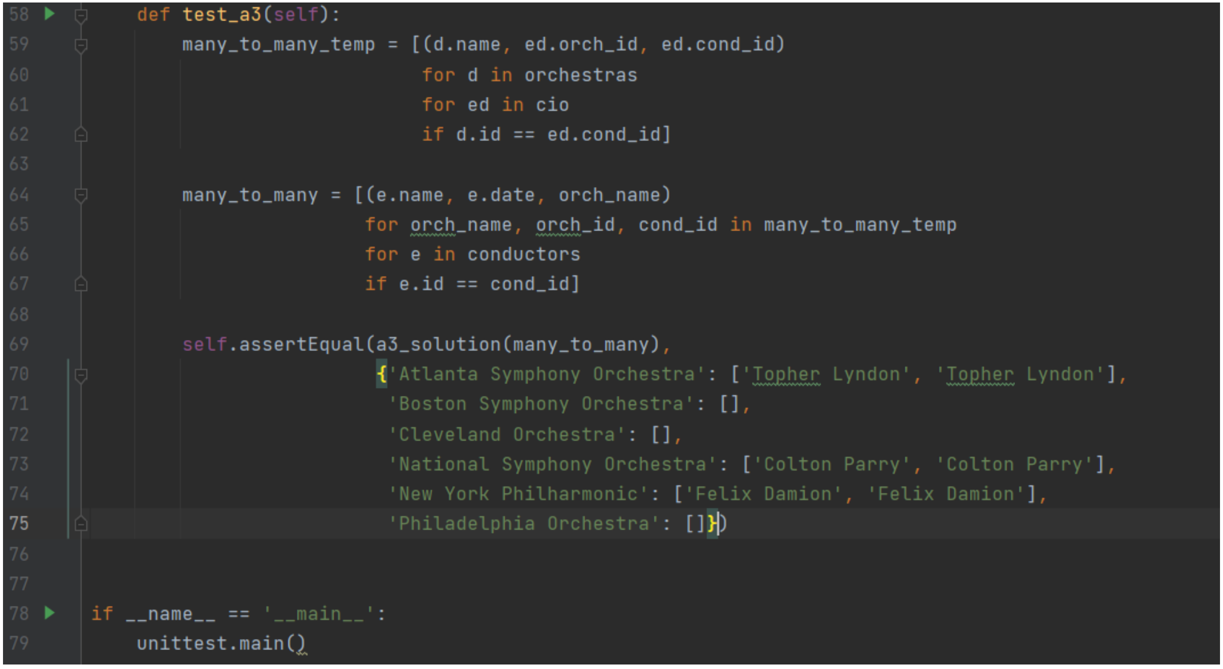
Test.py:



```python
import unittest
from main import *


class Test_Program(unittest.TestCase):
    orchestras = [
        Orchestra(1, 'Atlanta Symphony Orchestra'),
        Orchestra(2, 'National Symphony Orchestra'),
        Orchestra(3, 'New York Philharmonic'),

        Orchestra(11, 'Boston Symphony Orchestra'),
        Orchestra(22, 'Philadelphia Orchestra'),
        Orchestra(33, 'Cleveland Orchestra'),
    ]

    conductors = [
        Conductor(1, 'Topher Lyndon', 1965, 1),
        Conductor(2, 'Colton Parry', 1982, 2),
        Conductor(3, 'Felix Damion', 1973, 3),
        Conductor(4, 'Cam Raymond', 1988, 3),
        Conductor(5, 'Armen Mackenzie', 1999, 3),
    ]

    cio = [
        ConductorInOrchestra(1, 1),
        ConductorInOrchestra(2, 2),
        ConductorInOrchestra(3, 3),
        ConductorInOrchestra(3, 4),
        ConductorInOrchestra(3, 5),

```

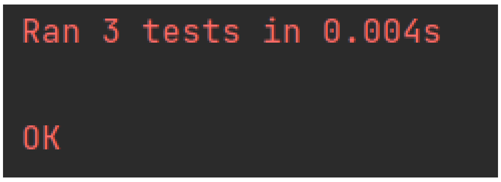


```python
        ConductorInOrchestra(11, 1),
        ConductorInOrchestra(22, 2),
        ConductorInOrchestra(33, 3),
        ConductorInOrchestra(33, 4),
        ConductorInOrchestra(33, 5),
    ]

    def test_a1(self):
        one_to_many = [(p.name, p.date, d.name)
                       for d in orchestras
                       for p in conductors
                       if p.cond_id == d.id]

        self.assertEqual(a1_solution(one_to_many),
                         [('Topher Lyndon', 1965, 'Atlanta Symphony Orchestra')])

    def test_a2(self):
        one_to_many = [(p.name, p.date, d.name)
                       for d in orchestras
                       for p in conductors
                       if p.cond_id == d.id]

        self.assertEqual(a2_solution(one_to_many),
                         [('New York Philharmonic', 1999),
                          ('National Symphony Orchestra', 1982),
                          ('Atlanta Symphony Orchestra', 1965)])

    def test_a3(self):
        many_to_many_temp = [(d.name, ed.orch_id, ed.cond_id)
                             for d in orchestras
```



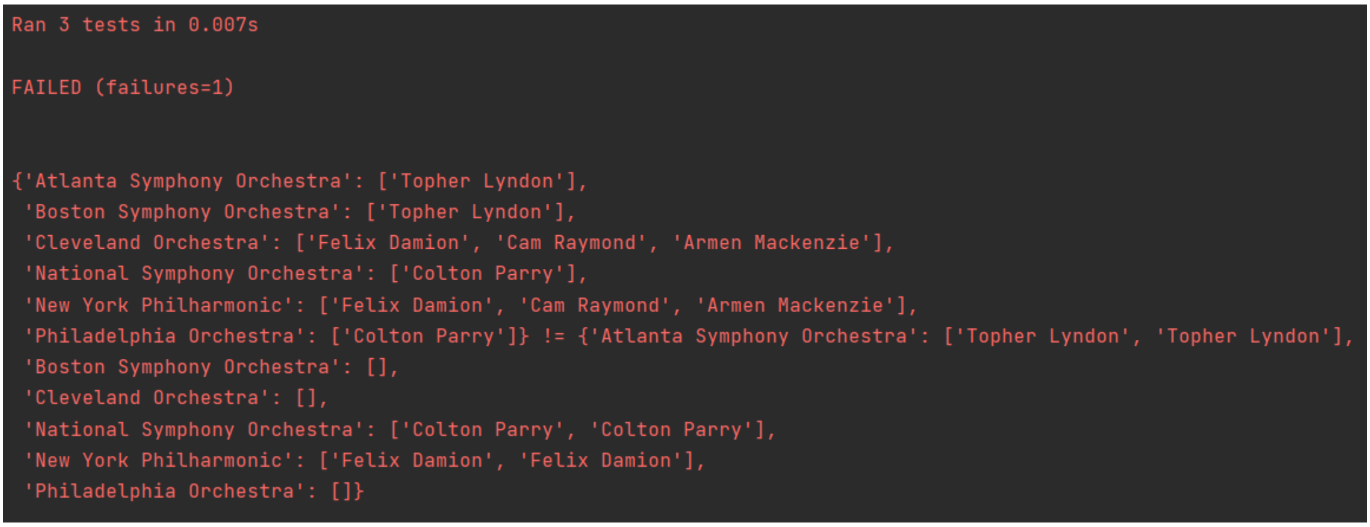
```python
    def test_a3(self):
        many_to_many_temp = [(d.name, ed.orch_id, ed.cond_id)
                             for d in orchestras
                             for ed in cio
                             if d.id == ed.cond_id]

        many_to_many = [(e.name, e.date, orch_name)
                        for orch_name, orch_id, cond_id in many_to_many_temp
                        for e in conductors
                        if e.id == cond_id]

        self.assertEqual(a3_solution(many_to_many),
                         {'Atlanta Symphony Orchestra': ['Topher Lyndon', 'Topher Lyndon'],
                          'Boston Symphony Orchestra': [],
                          'Cleveland Orchestra': [],
                          'National Symphony Orchestra': ['Colton Parry', 'Colton Parry'],
                          'New York Philharmonic': ['Felix Damion', 'Felix Damion'],
                          'Philadelphia Orchestra': []})


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

Результаты тестов:



```
Ran 3 tests in 0.004s

OK
```

Если при проверки функции a3_solution намеренно сделать ошибку в тестах, то в результатах теста увидим ошибку:



```
Ran 3 tests in 0.007s

FAILED (failures=1)


{'Atlanta Symphony Orchestra': ['Topher Lyndon'],
 'Boston Symphony Orchestra': ['Topher Lyndon'],
 'Cleveland Orchestra': ['Felix Damion', 'Cam Raymond', 'Armen Mackenzie'],
 'National Symphony Orchestra': ['Colton Parry'],
 'New York Philharmonic': ['Felix Damion', 'Cam Raymond', 'Armen Mackenzie'],
 'Philadelphia Orchestra': ['Colton Parry']} != {'Atlanta Symphony Orchestra': ['Topher Lyndon', 'Topher Lyndon'],
 'Boston Symphony Orchestra': [],
 'Cleveland Orchestra': [],
 'National Symphony Orchestra': ['Colton Parry', 'Colton Parry'],
 'New York Philharmonic': ['Felix Damion', 'Felix Damion'],
 'Philadelphia Orchestra': []}
```