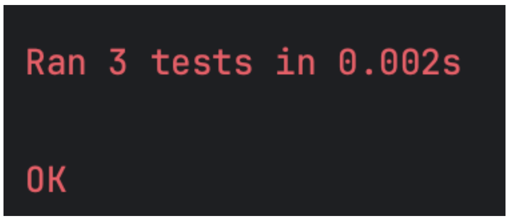
Текст программы main.py:

```
from operator import itemgetter

class prep:
    """класс преподавателя"""
    def __init__(self, id, fio, sal, course_id):
        self.id = id
        self.fio = fio
        self.sal = sal
        self.course_id = course_id

class course:
    """класс учебного курса"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class prep_course:
    """Класс для реализации многие ко многим"""
    def __init__(self, prep_id, course_id):
        self.prep_id = prep_id
        self.course_id = course_id

def one_to_many(preps, courses):
    return [(p.fio, p.sal, c.name)
               for c in courses
               for p in preps
               if p.course_id == c.id]

def many_to_many(preps, courses, courses_preps):
    many_to_many_temp = [(c.name, cp.course_id, cp.prep_id)
                         for c in courses
                         for cp in courses_preps
                         if c.id == cp.course_id]
    return [(p.fio, p.sal, c_name)
                for c_name, c_id, pr_id in many_to_many_temp
                for p in preps
                if p.id == pr_id]

def solution1(one_to_many, courses):
    answer_1 = {}
    for c in courses:
        if 'Курс' in c.name:
            c_preps = list(filter(lambda i: i[2] == c.name, one_to_many))
            only_fio = [x for x, _, _ in c_preps]
            answer_1[c.name] = only_fio
    return answer_1

def solution2(one_to_many, courses):
    answer_2 = []
    for c in courses:
        c_preps = list(filter(lambda i: i[2] == c.name, one_to_many))
        if len(c_preps) > 0:
            c_sal = [sal for _, sal, _ in c_preps]
            sr_sal = round(sum(c_sal) / len(c_preps), 2)
            answer_2.append((c.name, sr_sal))
    return sorted(answer_2, key= itemgetter(1))

def solution3(many_to_many, preps):
    answer_3 = {}
    for p in preps:
        if p.fio[0] == 'А':
            p_courses = list(filter(lambda i: i[0] == p.fio, many_to_many))
            only_cource = [x for _, _, x in p_courses]
            answer_3[p.fio] = only_cource
    return answer_3

if __name__ == '__main__':
    preps = [prep(1, 'Иванов Иван Иванович', 120, 3),
             prep(2, 'Антонов Петр Петрович', 100, 1),
             prep(3, 'Келдыш Елизавета Петровна', 110, 1),
             prep(4, 'Масленников Константин Юрьевич', 5, 2),
             prep(5, 'Афанасьев Геннадий Иванович', 6, 2)]

    courses = [course(1, 'Курс Математический анализ'),
               course(2, 'АСОИУ'),
               course(3, 'Курс Физика'),

               course(11, 'Математический анализ (дополнительный)'),
               course(22, 'Модели данных (дополнительный)'),
               course(33, 'Физика (дополнительный)')]

    courses_preps = [prep_course(1, 3),
                     prep_course(2, 1),
                     prep_course(3, 1),
                     prep_course(4, 2),
                     prep_course(5, 2),

                     prep_course(1, 11),
                     prep_course(2, 22),
                     prep_course(4, 33)]

    oneMany = one_to_many(preps, courses)
    manyMany = many_to_many(preps, courses, courses_preps)

    print("Решение 1:\n", solution1(oneMany, courses))
    print("Решение 2:\n", solution2(oneMany, courses))
    print("Решение 3:\n", solution3(manyMany, preps))
```

Текст программы tests.py:

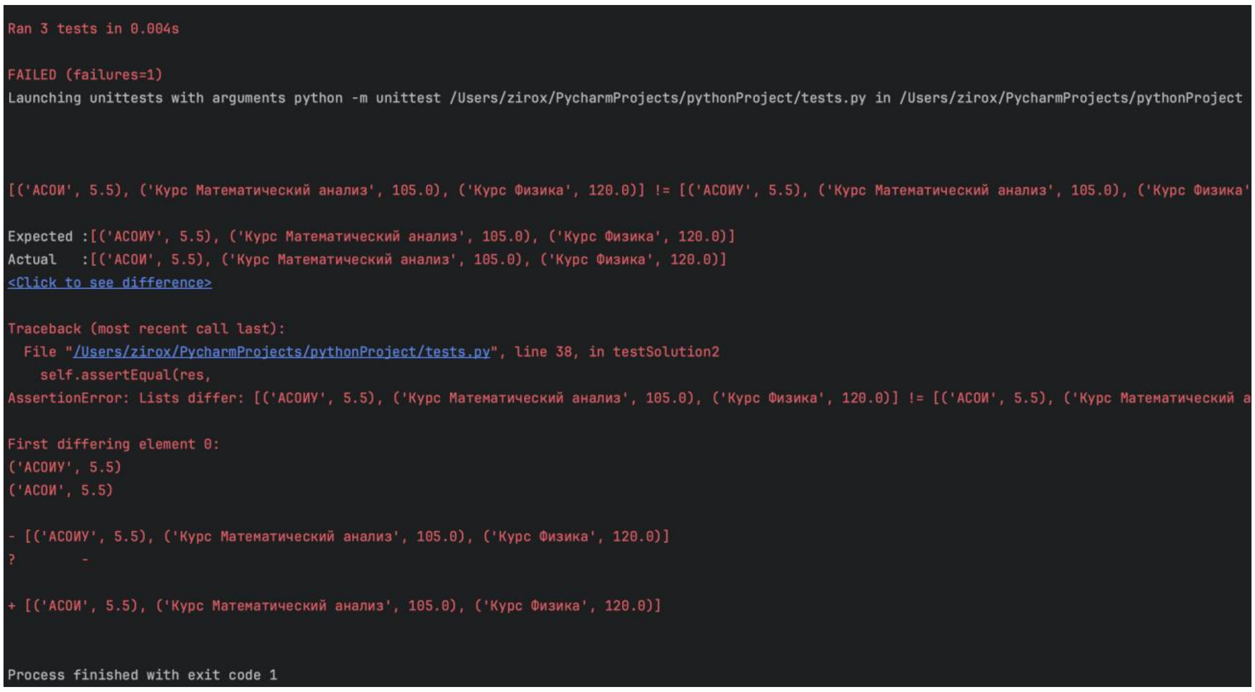
```
import unittest
from main import *

class Test(unittest.TestCase):
    def data(self):
        self.prepss = [prep(1, 'Иванов Иван Иванович', 120, 3),
             prep(2, 'Антонов Петр Петрович', 100, 1),
             prep(3, 'Келдыш Елизавета Петровна', 110, 1),
             prep(4, 'Масленников Константин Юрьевич', 5, 2),
             prep(5, 'Афанасьев Геннадий Иванович', 6, 2)]

        self.coursess = [course(1, 'Курс Математический анализ'),
               course(2, 'АСОИУ'),
               course(3, 'Курс Физика'),

               course(11, 'Математический анализ (дополнительный)'),
               course(22, 'Модели данных (дополнительный)'),
               course(33, 'Физика (дополнительный)')]

        self.courses_prepss = [prep_course(1, 3),
                     prep_course(2, 1),
                     prep_course(3, 1),
                     prep_course(4, 2),
                     prep_course(5, 2),

                     prep_course(1, 11),
                     prep_course(2, 22),
                     prep_course(4, 33)]
    def testSolution1(self):
        self.data()
        res = solution1(one_to_many(self.prepss, self.coursess),
self.coursess)
        self.assertEqual(res,
        {'Курс Математический анализ': ['Антонов Петр Петрович', 'Келдыш
Елизавета Петровна'], 'Курс Физика': ['Иванов Иван Иванович']})

    def testSolution2(self):
        self.data()
        res = solution2(one_to_many(self.prepss, self.coursess),
self.coursess)
        self.assertEqual(res,
                          [('АСОИУ', 5.5), ('Курс Математический анализ',
105.0), ('Курс Физика', 120.0)])

    def testSolution3(self):
        self.data()
        res = solution3(many_to_many(self.prepss, self.coursess,
self.courses_prepss), self.prepss)
        self.assertEqual(res,
                          {'Антонов Петр Петрович': ['Курс Математический
анализ', 'Модели данных (дополнительный)'], 'Афанасьев Геннадий Иванович':
['АСОИУ']})



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

Результат в случае успешного тестирования:



```
Ran 3 tests in 0.002s

OK
```

Пример неудачного тестирования:



```
Ran 3 tests in 0.004s

FAILED (failures=1)
Launching unittests with arguments python -m unittest /Users/zirox/PycharmProjects/pythonProject/tests.py in /Users/zirox/PycharmProjects/pythonProject



[('АСОИ', 5.5), ('Курс Математический анализ', 105.0), ('Курс Физика', 120.0)] != [('АСОИУ', 5.5), ('Курс Математический анализ', 105.0), ('Курс Физика'

Expected :[('АСОИУ', 5.5), ('Курс Математический анализ', 105.0), ('Курс Физика', 120.0)]
Actual   :[('АСОИ', 5.5), ('Курс Математический анализ', 105.0), ('Курс Физика', 120.0)]
<Click to see difference>

Traceback (most recent call last):
  File "/Users/zirox/PycharmProjects/pythonProject/tests.py", line 38, in testSolution2
    self.assertEqual(res,
AssertionError: Lists differ: [('АСОИУ', 5.5), ('Курс Математический анализ', 105.0), ('Курс Физика', 120.0)] != [('АСОИ', 5.5), ('Курс Математический а

First differing element 0:
('АСОИУ', 5.5)
('АСОИ', 5.5)

- [('АСОИУ', 5.5), ('Курс Математический анализ', 105.0), ('Курс Физика', 120.0)]
?         -

+ [('АСОИ', 5.5), ('Курс Математический анализ', 105.0), ('Курс Физика', 120.0)]


Process finished with exit code 1
```