текстовый формат

main.py

```
from operator import itemgetter

class Emp:
    """Студенческая группа"""
    def __init__(self, id, name, number_of_subjects,
number_of_credits):
        self.id = id
        self.name = name
        self.number_of_subjects = number_of_subjects
        self.number_of_credits = number_of_credits

class Dep:
    """Учебный курс"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class EmpDep:
    """
    'Студенческая группа - Учебный курс' для реализации
    связи многие-ко-многим
    """
    def __init__(self, dep_id, emp_id):
        self.dep_id = dep_id
        self.emp_id = emp_id

def one_to_many(deps, emps):
    return [(e.name, e.number_of_subjects, d.name)
            for d in deps
            for e in emps
            if e.id == d.id]

def many_to_many(deps, emps_deps, emps):
```

```
    many_to_many_temp = [(d.name, ed.dep_id, ed.emp_id)
                         for d in deps
                         for ed in emps_deps
                         if d.id == ed.dep_id]

    return [(e.name, e.number_of_subjects, name)
            for name, _, emp_id in many_to_many_temp
            for e in emps if e.id == emp_id]

def sum_subjects_by_course(deps, one_to_many_data):
    res = []
    for d in deps:
        dep_subjects = list(filter(lambda i: i[2] == d.name,
one_to_many_data))
        if dep_subjects:
            dep_subject_counts = [subjects for _, subjects, _ in
dep_subjects]
            dep_subjects_sum = sum(dep_subject_counts)
            res.append((d.name, dep_subjects_sum))
    return sorted(res, key=itemgetter(1), reverse=True)

def find_subjects_by_course(deps, many_to_many_data):
    res = {}
    for d in deps:
        if 'First' in d.name:
            dep_subjects = list(filter(lambda i: i[2] == d.name,
many_to_many_data))
            dep_subject_names = [x for x, _, _ in dep_subjects]
            res[d.name] = dep_subject_names
    return res


if __name__ == "__main__":
    deps = [
        Dep(1, 'First'),
        Dep(2, 'Second'),
        Dep(3, 'Third'),
```

```
    Dep(11, 'Fourth'),
    Dep(22, 'Fifth'),
    Dep(33, 'Sixth'),
]

# Учебный курс
emps = [
    Emp(1, 'IU7-12B', 2, 11),
    Emp(2, 'IU5-11B',  1, 11),
    Emp(3, 'IU6-21B', 4, 33),
    Emp(4, 'IU7-32B', 2, 33),
    Emp(5, 'IU7-54B', 8, 2),
]

emps_deps = [
    EmpDep(1, 1),
    EmpDep(1, 2),
    EmpDep(1, 3),
    EmpDep(3, 4),
    EmpDep(2, 5),

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

o_to_m = one_to_many(deps, emps)
m_to_m = many_to_many(deps, emps_deps, emps)

print('Задание А1')
sorted_list = sorted(o_to_m  , key=itemgetter(2))
print(sorted_list)

sun_subs_by_course = sum_subjects_by_course(deps, o_to_m)
print('Задание А2')
```

```
    print(sun_subs_by_course)
    print('Задание А3')

    subs_by_course = find_subjects_by_course(deps, m_to_m)

    print(subs_by_course)

#Результаты выполнения:

    # Задание А1
    # [('IU7-12B', 2, 'First'), ('IU5-11B', 1, 'Second'), ('IU6-21B', 4,
'Third')]
    # Задание А2
    # [('Third', 4), ('First', 2), ('Second', 1)]
    # Задание А3
    # {'First': ['IU7-12B', 'IU5-11B', 'IU6-21B']}
```

tddtests.py

```
import unittest
from main import *
class TestFunctions(unittest.TestCase):
    def setUp(self):
        self.deps = [
            Dep(1, 'First'),
            Dep(2, 'Second'),
            # ... остальные данные ...
        ]

        self.emps = [
            Emp(1, 'IU7-12B', 2, 11),
            Emp(2, 'IU5-11B',  1, 11),
            # ... остальные данные ...
```

```
        ]

        self.emps_deps = [
            EmpDep(1, 1),
            EmpDep(1, 2),
            # ... остальные данные ...
        ]

    def test_one_to_many(self):
        result = one_to_many(self.deps, self.emps)
        self.assertEqual(result, [('IU7-12B', 2, 'First'), ('IU5-11B', 1,
'Second')])

    def test_many_to_many(self):
        result = many_to_many(self.deps, self.emps_deps, self.emps)
        self.assertEqual(result, [('IU7-12B', 2, 'First'), ('IU5-11B', 1,
'First')])

    def test_sum_subjects_by_course(self):
        one_to_many_data = one_to_many(self.deps, self.emps)
        result = sum_subjects_by_course(self.deps, one_to_many_data)
        self.assertEqual(result, [('First', 2), ('Second', 1)])

    def test_find_subjects_by_course(self):
        many_to_many_data = many_to_many(self.deps, self.emps_deps,
self.emps)
        result = find_subjects_by_course(self.deps, many_to_many_data)
        self.assertEqual(result, {'First': ['IU7-12B', 'IU5-11B']})

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


#Результаты выполнения:


# ----------------------------------------------------------------------
# Ran 4 tests in 0.000s
```

OK