ОТЧЁТ ПО РК2
СТУДЕНТ ГРУППЫ ИУ5-31Б:
АЛЬЯНОВ ЕГОР СЕРГЕЕВИЧ

ТЕКСТ ПРОГРАММЫ:

Main.py:

```
# используется для сортировки
from operator import itemgetter


class Emp:
    """Студент"""

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


class Dep:
    """Группа"""

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


class EmpDep:
    """
    'Студенты группы' для реализации
    связи многие-ко-многим
    """

    def __init__(self, dep_id, emp_id):
        self.dep_id = dep_id
        self.emp_id = emp_id


# Группы
deps = [
    Dep(1, 'IU5-31B'),
    Dep(2, 'IU5-32B'),
    Dep(3, 'IU5-33B'),

    Dep(11, 'IU6-22B'),
    Dep(22, 'IU7-22B'),
    Dep(33, 'IU9-22B'),
]

# Студенты
emps = [
    Emp(1, 'Ivanov', 2, 11),
    Emp(2, 'Alianov', 1, 11),
    Emp(3, 'Baburin', 4, 33),
    Emp(4, 'Berdyugin', 2, 33),
    Emp(5, 'Baranov', 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),
]

def a1_solution(one_to_many):
    res_a1 = sorted(one_to_many, key=itemgetter(2))
    return res_a1


def a2_solution(one_to_many):
    res_a2_unsorted = []
    for d in deps:
        d_emps = list(filter(lambda i: i[2] == d.index, one_to_many))
        if len(d_emps) > 0:
            d_sals = [sal for _, sal, _ in d_emps]
            d_sals_sum = sum(d_sals)
            res_a2_unsorted.append((d.index, d_sals_sum))

    res_a2 = sorted(res_a2_unsorted, key=itemgetter(1), reverse=True)
    return res_a2


def a3_solution(many_to_many):
    res_a3 = {}
    for d in deps:
        if 'IU5' in d.index:
            d_emps = list(filter(lambda i: i[2] == d.index, many_to_many))
            d_emps_names = [x for x, _, _ in d_emps]
            res_a3[d.index] = d_emps_names
    return res_a3


def main():
    """Основная функция"""

    # Соединение данных один-ко-многим
    one_to_many = [(e.name, e.debts, d.index)
                   for d in deps
                   for e in emps
                   if e.dep_id == d.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(d.index, ed.dep_id, ed.emp_id)
                         for d in deps
                         for ed in emps_deps
                         if d.id == ed.dep_id]

    many_to_many = [(e.name, e.debts, dep_name)
                    for dep_name, dep_id, emp_id in many_to_many_temp
                    for e in emps if e.id == emp_id]

    print('Задание А1')
    print(a1_solution(one_to_many))
```

```
    print('\nЗадание A2')
    print(a2_solution(one_to_many))

    print('\nЗадание A3')
    print(a3_solution(many_to_many))




if __name__ == '__main__':
    main()

# Результаты выполнения:
#
# Задание A1
# [('Baranov', 8, 'IU5-32B'), ('Ivanov', 2, 'IU6-22B'), ('Alianov', 1, 'IU6-
22B'), ('Baburin', 4, 'IU9-22B'), ('Berdyugin', 2, 'IU9-22B')]
#
# Задание A2
# [('IU5-32B', 8), ('IU9-22B', 6), ('IU6-22B', 3)]
#
# Задание A3
# {'IU5-31B': ['Ivanov', 'Alianov', 'Baburin'], 'IU5-32B': ['Baranov'], 'IU5-
33B': ['Berdyugin']}
```

Tests.py:

```
import unittest
from main import *


class TestRK2(unittest.TestCase):
    # Группы
    deps = [
        Dep(1, 'IU5-31B'),
        Dep(2, 'IU5-32B'),
        Dep(3, 'IU5-33B'),

        Dep(11, 'IU6-22B'),
        Dep(22, 'IU7-22B'),
        Dep(33, 'IU9-22B'),
    ]

    # Студент
    emps = [
        Emp(1, 'Ivanov', 2, 11),
        Emp(2, 'Alianov', 1, 11),
        Emp(3, 'Baburin', 4, 33),
        Emp(4, 'Berdyugin', 2, 33),
        Emp(5, 'Baranov', 8, 2),
    ]

    def test_A1(self):
        one_to_many = [(e.name, e.debts, d.index)
                       for d in deps
                       for e in emps
                       if e.dep_id == d.id]
        self.assertEqual(a1_solution(one_to_many),
                         [('Baranov', 8, 'IU5-32B'), ('Ivanov', 2, 'IU6-
22B'), ('Alianov', 1, 'IU6-22B'),
                          ('Baburin', 4, 'IU9-22B'), ('Berdyugin', 2, 'IU9-
22B')])
```

```
    def test_A2(self):
        one_to_many = [(e.name, e.debts, d.index)
                       for d in deps
                       for e in emps
                       if e.dep_id == d.id]
        self.assertEqual(a2_solution(one_to_many),
                         [('IU5-32B', 8), ('IU9-22B', 6), ('IU6-22B', 3)])

    def test_A3(self):
        many_to_many_temp = [(d.index, ed.dep_id, ed.emp_id)
                             for d in deps
                             for ed in emps_deps
                             if d.id == ed.dep_id]

        many_to_many = [(e.name, e.debts, dep_name)
                        for dep_name, dep_id, emp_id in many_to_many_temp
                        for e in emps if e.id == emp_id]
        self.assertEqual(a3_solution(many_to_many),
                         {'IU5-31B': ['Ivanov', 'Alianov', 'Baburin'], 'IU5-
32B': ['Baranov'], 'IU5-33B': ['Berdyugin']})


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

Результаты работы:

Успешная:

```
Testing started at 14:24 ...


Ran 3 tests in 0.002s

OK
Launching unittests with arguments python -m unittest tests.TestRK2 in /Users/egoralanov/PycharmProjects/pythonProject1


Process finished with exit code 0
```

Ошибка:

```
/Users/egoralanov/PycharmProjects/pythonProject1/venv/bin/python "/Applications/PyCharm CE.app/Contents/plugins/python-ce/helpe
Testing started at 14:27 ...
Launching unittests with arguments python -m unittest tests.TestRK2 in /Users/egoralanov/PycharmProjects/pythonProject1



Ran 3 tests in 0.006s

FAILED (failures=1)


{'IU5-31B': ['Ivanov', 'Alianov', 'Baburin'],
 'IU5-33B': ['Berdyugin'],
 'IU5-3B': ['Baranov']} != {'IU5-31B': ['Ivanov', 'Alianov', 'Baburin'],
 'IU5-32B': ['Baranov'],
 'IU5-33B': ['Berdyugin']}

<Click to see difference>

Traceback (most recent call last):
  File "/Users/egoralanov/PycharmProjects/pythonProject1/tests.py", line 52, in test_A3
    self.assertEqual(a3_solution(many_to_many),
AssertionError: {'IU5[27 chars], 'Baburin'], 'IU5-32B': ['Baranov'], 'IU5-33B': ['Berdyugin']} != {'IU5[27 chars], 'Baburin'],
  {'IU5-31B': ['Ivanov', 'Alianov', 'Baburin'],
-  'IU5-32B': ['Baranov'],
-  'IU5-33B': ['Berdyugin']}
?                         ^
```

```
+  'IU5-33B': ['Berdyugin'],
?                           ^


+  'IU5-3B': ['Baranov']}




Process finished with exit code 1
```