

**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»**

**Рубежный контроль №2.**

Выполнил:  
студент группы ИУ5-34Б:  
Белозеров Дмитрий Сергеевич  
Подпись и дата:

Проверил:  
преподаватель каф. ИУ5  
Гапанюк Юрий Евгеньевич  
Подпись и дата:

## Условия рубежного контроля №2 по курсу БКИТ

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).

### Код программы

#### main.py

```
import unittest

class RK2test(unittest.TestCase):
    def setUp(self):
        self.one_to_many = [(e.fio, e.avg_mark, d.name)
                             for d in groups
                             for e in pups
                             if e.group_id == d.id]
        self.many_to_many = [(e.fio, e.avg_mark, dep_name)
                              for dep_name, dep_id, emp_id in many_to_many_temp
                              for e in pups if e.id == emp_id]

    def test_result_1(self):
        self.assertEqual(task_1(self.one_to_many), {'A1, класс':
['Артамонов'],
                                                    'A2, класс': ['Иваненко',
'Иванов', 'Иванин']})

    def test_result_2(self):
        self.assertEqual(task_2(self.one_to_many), [('A1, класс', 5.0), ('B1,
класс', 4.7), ('A2, класс', 4.6)])

    def test_result_3(self):
        self.assertEqual(task_3(self.many_to_many), [('A1, класс',
['Артамонов']), ('A1, класс v1', ['Артамонов']),
                                                    ('A2, класс',
['Иваненко', 'Иванов', 'Иванин']),
                                                    ('A2, класс v1',
['Иваненко', 'Иванов', 'Иванин']),
                                                    ('B1, класс',
['Петров']), ('B1, класс v1', ['Петров'])])

class Pupils:
    def __init__(self, id, fio, avg_mark, group_id):
        self.id = id
        self.fio = fio
        self.avg_mark = avg_mark
        self.group_id = group_id

class Group:
    def __init__(self, id, name):
        self.id = id
        self.name = name
```

```

class PupGroup:
    def __init__(self, group_id, pup_id):
        self.group_id = group_id
        self.pup_id = pup_id

groups = [
    Group(1, 'A1, класс'),
    Group(2, 'B1, класс'),
    Group(3, 'A2, класс'),
    Group(11, 'A1, класс v1'),
    Group(22, 'B1, класс v1'),
    Group(33, 'A2, класс v1'),
]

pups = [
    Pupils(1, 'Артамонов', 5.0, 1),
    Pupils(2, 'Петров', 4.7, 2),
    Pupils(3, 'Иваненко', 3.2, 3),
    Pupils(4, 'Иванов', 2.9, 3),
    Pupils(5, 'Иванин', 4.6, 3),
]

pups_groups = [
    PupGroup(1, 1),
    PupGroup(2, 2),
    PupGroup(3, 3),
    PupGroup(3, 4),
    PupGroup(3, 5),
    PupGroup(11, 1),
    PupGroup(22, 2),
    PupGroup(33, 3),
    PupGroup(33, 4),
    PupGroup(33, 5),
]

one_to_many = [(e.fio, e.avg_mark, d.name)
                for d in groups
                for e in pups
                if e.group_id == d.id]

many_to_many_temp = [(d.name, ed.group_id, ed.pup_id)
                     for d in groups
                     for ed in pups_groups
                     if d.id == ed.group_id]

many_to_many = [(e.fio, e.avg_mark, dep_name)
                 for dep_name, dep_id, emp_id in many_to_many_temp
                 for e in pups if e.id == emp_id]

def main():
    print('Задание Г1')
    print(task_1(one_to_many))
    print('\nЗадание Г2')
    print(task_2(one_to_many))
    print('\nЗадание Г3')
    print(task_3(many_to_many))

def task_1(one_to_many):
    res_11 = {}
    for fio, avg, name in one_to_many:
        if name[0] == 'A':
            if name in res_11:

```

```

        res_11[name].append(fio)
    else:
        res_11[name] = [fio]
return res_11

def task_2(one_to_many):
    res_12 = []
    for g in groups:
        g_groups = list(filter(lambda i: i[2] == g.name, one_to_many))
        if len(g_groups) > 0:
            g_mark = [sal for _, sal, _ in g_groups]
            g_mark_max = max(g_mark)
            res_12.append((g.name, g_mark_max))
    return sorted(res_12, key=lambda x: x[1], reverse=True)

def task_3(many_to_many):
    res_13_unsorted = {}
    for fio, avg, name in many_to_many:
        if name in res_13_unsorted:
            res_13_unsorted[name].append(fio)
        else:
            res_13_unsorted[name] = [fio]
    res_13 = sorted(res_13_unsorted.items(), key=lambda x: x[0])
    return res_13

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

```

## Результат выполнения

```

✔ Tests passed: 3 of 3 tests - 3 ms

C:\Users\beloz\PycharmProjects\RK_2\venv\Scripts\python.exe "C:/Program Files/JetBrains/PyCharm 2022.2.1/plugins/python/helpers/pycharm/_
Testing started at 20:31 ...
Launching unittests with arguments python -m unittest C:/Users/beloz/PycharmProjects/RK_2/main.py in C:\Users\beloz\PycharmProjects\RK_2

Ran 3 tests in 0.004s

OK

Process finished with exit code 0

```