

**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»**

**Отчет по рубежному контролю № 2**

Выполнила:

студентка группы ИУ5-32Б  
Андреева Анастасия

Проверил:

преподаватель каф. ИУ5  
Гапанюк Ю. Е.

Подпись и дата:

Подпись и дата:

Москва, 2021 г.

## **Описание задания**

(Вариант запросов - Б, вариант предметной области – 1)

1. «Группа» и «Студент» связаны соотношением один-ко-многим. Выведите список всех связанных студентов и групп, отсортированный по студентам, сортировка по группам произвольная.
2. «Группа» и «Студент» связаны соотношением один-ко-многим. Выведите список групп с количеством студентов в каждом отделе, отсортированный по количеству студентов.
3. «Группа» и «Студент» связаны соотношением многие-ко-многим. Выведите список всех студентов, у которых фамилия заканчивается на «ов», и названия их групп.

## **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).

## 1) Текст программы main.py

```
from operator import itemgetter

class Stud:
    """Студент"""
    def __init__(self, id, fio, scr, grp_id):
        self.id = id
        self.fio = fio
        self.scr = scr
        self.grp_id = grp_id

class Grp:
    """Группа"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class StudGrp:
    """ 'Студенты группы' для реализации связи многие-ко-многим """
    def __init__(self, stud_id, grp_id):
        self.grp_id = grp_id
        self.stud_id = stud_id

# Группы
groups = [
    Grp(1, 'ИУ5-31Б'),
    Grp(2, 'ИУ5-32Б'),
    Grp(3, 'ИУ5-33Б'),
    Grp(4, 'ИУ5-34Б'),
    Grp(5, 'ИУ5-35Б')
]

# Студенты
studs = [
    Stud(1, 'Иванов', 80, 1),
    Stud(2, 'Петрова', 85, 1),
    Stud(3, 'Андреева', 99, 2),
    Stud(4, 'Коваленко', 71, 2),
    Stud(5, 'Бодров', 100, 2),
    Stud(6, 'Степанова', 60, 3),
    Stud(7, 'Платонов', 89, 3),
    Stud(8, 'Сидоров', 90, 3),
    Stud(9, 'Гладких', 67, 3),
    Stud(10, 'Батянов', 100, 4),
    Stud(11, 'Золина', 70, 5)
]

studs_groups = [
    StudGrp(1, 1),
    StudGrp(2, 1),
    StudGrp(3, 2),
    StudGrp(4, 2),
    StudGrp(5, 2),
    StudGrp(6, 3),
    StudGrp(7, 3),
    StudGrp(8, 3),
    StudGrp(9, 3),
    StudGrp(10, 4),
    StudGrp(11, 5)
]

def bl(one_to_many):
```

```

res1 = sorted(one_to_many, key=itemgetter(0))
return res1

def b2(one_to_many):
    res2_temp = []
    # Перебираем все группы
    for g in groups:
        # Список студентов группы
        g_studs = list(filter(lambda i: i[2] == g.name, one_to_many))
        if len(g_studs) > 0:
            res2_temp.append((g.name, len(g_studs)))
    # Сортировка по количеству студентов в группе
    res2 = sorted(res2_temp, key=itemgetter(1), reverse=True)
    return res2

def b3(many_to_many):
    res3 = {}
    for s in studs:
        if s.fio.endswith('ов'):
            # Список студентов с фамилией на "ов"
            s_groups = list(filter(lambda i: i[0] == s.fio, many_to_many))
            # Получаем их имена
            s_groups_names = [x[2] for x in s_groups]
            res3[s.fio] = s_groups_names
    return res3

def main():
    # Соединение данных один-ко-многим
    one_to_many = [(s.fio, s.scr, g.name)
                    for s in studs
                    for g in groups
                    if s.grp_id==g.id]
    # Соединение данных многие-ко-многим
    many_to_many_temp = [(g.name, sg.grp_id, sg.stud_id)
                          for g in groups
                          for sg in studs_groups
                          if g.id==sg.grp_id]

    many_to_many = [(s.fio, s.scr, grp_name)
                     for grp_name, grp_id, stud_id in many_to_many_temp
                     for s in studs if s.id == stud_id]

    print('Задание Б1')
    print(b1(one_to_many))

    print('\nЗадание Б2')

    print(b2(one_to_many))

    print('\nЗадание Б3')

    print(b3(many_to_many))

if __name__ == '__main__':
    main()

```

## Результат выполнения программы main.py

```
/Users/anastasia/PycharmProjects/BKIT_RK1/venv/bin/python /Users/anastasia/PycharmProjects/BKIT_RK1/main.py
Задание Б1
('Андреева', 99, 'ИУ5-32Б')
('Батянов', 100, 'ИУ5-34Б')
('Бодров', 100, 'ИУ5-32Б')
('Гладких', 67, 'ИУ5-33Б')
('Золина', 70, 'ИУ5-35Б')
('Иванов', 80, 'ИУ5-31Б')
('Коваленко', 71, 'ИУ5-32Б')
('Петрова', 85, 'ИУ5-31Б')
('Платонов', 89, 'ИУ5-33Б')
('Сидоров', 90, 'ИУ5-33Б')
('Степанова', 60, 'ИУ5-33Б')

Задание Б2
('ИУ5-33Б', 4)
('ИУ5-32Б', 3)
('ИУ5-31Б', 2)
('ИУ5-34Б', 1)
('ИУ5-35Б', 1)

Задание Б3
Иванов ['ИУ5-31Б']
Бодров ['ИУ5-32Б']
Платонов ['ИУ5-33Б']
Сидоров ['ИУ5-33Б']
Батянов ['ИУ5-34Б']

Process finished with exit code 0
```

## 2) Текст программы test\_tdd.py

```
import unittest
from main import *

class test_cases(unittest.TestCase):
    def test_b1(self):
        one_to_many = [(s.fio, s.scr, g.name)
                        for s in studs
                        for g in groups
                        if s.grp_id == g.id]

        self.assertEqual(b1(one_to_many), [ ('Андреева', 99, 'ИУ5-32Б'),
                                             ('Батынов', 100, 'ИУ5-34Б'),
                                             ('Бодров', 100, 'ИУ5-32Б'),
                                             ('Гладких', 67, 'ИУ5-33Б'),
                                             ('Золкина', 70, 'ИУ5-35Б'),
                                             ('Иванов', 80, 'ИУ5-31Б'),
                                             ('Коваленко', 71, 'ИУ5-32Б'),
                                             ('Петрова', 85, 'ИУ5-31Б'),
                                             ('Платонов', 89, 'ИУ5-33Б'),
                                             ('Сидоров', 90, 'ИУ5-33Б'),
                                             ('Степанова', 60, 'ИУ5-33Б')])

    def test_b2(self):
        one_to_many = [(s.fio, s.scr, g.name)
                        for s in studs
                        for g in groups
                        if s.grp_id == g.id]
        self.assertEqual(b2(one_to_many), [ ('ИУ5-33Б', 4),
                                             ('ИУ5-32Б', 3),
                                             ('ИУ5-31Б', 2),
                                             ('ИУ5-34Б', 1),
                                             ('ИУ5-35Б', 1)])

    def test_b3(self):
        many_to_many_temp = [(g.name, sg.grp_id, sg.stud_id)
                              for g in groups
                              for sg in studs_groups
                              if g.id == sg.grp_id]

        many_to_many = [(s.fio, s.scr, grp_name)
                          for grp_name, grp_id, stud_id in many_to_many_temp
                          for s in studs if s.id == stud_id]
        self.assertEqual(b3(many_to_many), {'Иванов': ['ИУ5-31Б'],
                                             'Бодров': ['ИУ5-32Б'],
                                             'Платонов': ['ИУ5-33Б'],
                                             'Сидоров': ['ИУ5-33Б'],
                                             'Батынов': ['ИУ5-34Б']})

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

## Результат выполнения программы test\_tdd.py

...

-----  
Ran 3 tests in 0.000s

OK

Process finished with exit code 0