



**Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Московский государственный технический
университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика и системы управления»
Кафедра «Системы обработки информации и
управления»**

Рубежный контроль №2 по курсу
«Базовые компоненты интернет технологий»

Выполнил:
студент группы № ИУ5-33Б
Балюк А.В

Проверил:
Преподаватель
Гапанюк Ю.Е

2022 г.

Листинг

1) Рефакторинг текста:

```
from operator import itemgetter

class Student:
    """Студент"""

    def __init__(self, id, Fio, age, rec_book, group_id):
        self.id = id
        self.Fio = Fio
        self.age = age
        self.rec_book = rec_book
        self.group_id = group_id

class Group:
    """Группа"""

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

class StudentsGroup:
    """
    Студенты в группе
    """

    def __init__(self, group_id, student_id):
        self.student_id = student_id
        self.group_id = group_id

# Группы
groups = [
    Group(1, 'ИУ5-31Б'),
    Group(2, 'ИУ5-32Б'),
    Group(3, 'ИУ5-33Б'),

    # Для связи М-М
    Group(11, 'РК7-11Б'),
    Group(22, 'РК7-22Б'),
    Group(33, 'РК7-33Б'),
]

# Студенты
students = [
    Student(1, 'Петров Ю.Г', 19, '38A180', 1),
    Student(2, 'Иванченко В.О', 21, '38A181', 1),
    Student(3, 'Шалапин Н.Г', 17, '38A182', 2),
    Student(4, 'Сколов Н.А', 24, '38A183', 2),
    Student(5, 'Клемент А.Н', 18, '38A184', 3),
    Student(6, 'Поляков Д.Ю', 19, '38A185', 3)
]

# Студенты в группах (связь М-М)
students_group = [
    StudentsGroup(1, 1),
    StudentsGroup(1, 2),
    StudentsGroup(2, 3),
    StudentsGroup(2, 4),
```

```

StudentsGroup(3, 5),
StudentsGroup(3, 6),

StudentsGroup(11, 1),
StudentsGroup(11, 2),
StudentsGroup(22, 3),
StudentsGroup(22, 4),
StudentsGroup(33, 5),
StudentsGroup(33, 6),
]

def one_to_many(students, groups):
    return ([
        (s.Fio, s.age, s.rec_book, g.index)
        for s in students
        for g in groups
        if s.group_id == g.id])

def many_to_many(students, students_group):
    return ([
        (s.Fio, s.age, s.rec_book, [g.index for g in groups if g.id ==
s_g.group_id][0])
        for s in students
        for s_g in students_group
        if s.id == s_g.student_id
    ])

def task1(students, groups):
    res_1 = list(filter(lambda x: x[0][0] == 'П', one_to_many(students,
groups)))
    return res_1

def task2(students, groups):
    res_2 = []
    s_age = []
    s_fio = {}
    for g in groups:
        st_g = list(filter(lambda x: x[3] == g.index,
one_to_many(students, groups)))
        if len(st_g) > 0:
            for i in st_g:
                s_age.append(i[1])
                s_fio[i[1]] = i[0]
            s_age_min = min(s_age)
            s_age.clear()
            res_2.append((g.index, s_age_min, s_fio[s_age_min]))
    res_2 = sorted(res_2, key=itemgetter(1), reverse=False)
    return res_2

def task3(students, students_group):
    res_3 = sorted(many_to_many(students, students_group), key=itemgetter(2))
    return res_3

def main():
    print('Задание №1')
    task1(students, groups)
    print('Задание №2')
    task2(students, groups)
    print('Задание №3')
    task3(students, students_group)

if __name__ == '__main__':
    main()

```

2) Текст программы:

```
import unittest
import rk1

class testRk(unittest.TestCase):
    def setUp(self):
        self.test1 = [('Петров Ю.Г', 19, '38A180', 'ИУ5-31Б'),
                       ('Поляков Д.Ю', 19, '38A185', 'ИУ5-33Б')]
        self.test2 = [('ИУ5-32Б', 17, 'Шаляпин Н.Г'),
                       ('ИУ5-33Б', 18, 'Клемент А.Н'),
                       ('ИУ5-31Б', 19, 'Петров Ю.Г')]
        self.test3 = [('Петров Ю.Г', 19, '38A180', 'ИУ5-31Б'),
                       ('Петров Ю.Г', 19, '38A180', 'РК7-11Б'),
                       ('Иванченко В.О', 21, '38A181', 'ИУ5-31Б'),
                       ('Иванченко В.О', 21, '38A181', 'РК7-11Б'),
                       ('Шаляпин Н.Г', 17, '38A182', 'ИУ5-32Б'),
                       ('Шаляпин Н.Г', 17, '38A182', 'РК7-22Б'),
                       ('Сколов Н.А', 24, '38A183', 'ИУ5-32Б'),
                       ('Сколов Н.А', 24, '38A183', 'РК7-22Б'),
                       ('Клемент А.Н', 18, '38A184', 'ИУ5-33Б'),
                       ('Клемент А.Н', 18, '38A184', 'РК7-33Б'),
                       ('Поляков Д.Ю', 19, '38A185', 'ИУ5-33Б'),
                       ('Поляков Д.Ю', 19, '38A185', 'РК7-33Б')]

    def test1_rk(self):
        self.assertEqual(rk1.task1(rk1.students, rk1.groups), self.test1)
    def test2_rk(self):
        self.assertEqual(rk1.task2(rk1.students, rk1.groups), self.test2)
    def test3_rk(self):
        self.assertEqual(rk1.task3(rk1.students, rk1.students_group),
                          self.test3)

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

Результат:

```
(venv) C:\Users\Андрей\Desktop\rk2>main.py
...
-----
Ran 3 tests in 0.000s

OK

(venv) C:\Users\Андрей\Desktop\rk2>
```