



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

**Факультет «ГУИМЦ»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Дисциплина «Базовые компоненты ИТ»

ОТЧЕТ

Рубежный контроль №2

Студент: Федюнев А. Ю., группа ИУ5Ц-52Б

Преподаватель: Гапанюк Ю. Е.

2021г.

## 1. Описание задания

Вариант Б, вариант предметной области №26.

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).

## 2. Текст программы

**rk1.py**

```
from operator import itemgetter

class course:
    def __init__(self, id, number, faculty):
        self.id = id
        self.number = number
        self.faculty = faculty

class group:
    def __init__(self, id, name, amount, course_id):
        self.id = id
        self.name = name
        self.amount = amount
        self.course_id = course_id

class group_course:
    def __init__(self, group_id, course_id):
        self.group_id = group_id
        self.course_id = course_id

courses = [
    course(1, 2, 'ИУ'),
    course(2, 1, 'ИУ'),
    course(3, 3, 'ИУ'),
    course(4, 1, 'БМТ'),
    course(5, 2, 'БМТ'),
]

group_s = [
    group(1, 'ИУ-21', 20, 1),
    group(2, 'ИУ-31', 25, 3),
    group(3, 'ИУ-14', 30, 2),
    group(4, 'БМТ-12', 31, 4),
    group(5, 'ИУ-13', 28, 2),
]

group_course_s = [
    group_course(1, 2),
    group_course(2, 3),
    group_course(3, 2),
    group_course(4, 4),
```

```

        group_course(5, 2),
    ]

def sort_2(cours, one_to_many):
    res_12_unsorted = []
    for c in cours:
        courses_s = list(filter(lambda i: i[2] == c.faculty, one_to_many))
        if len(courses_s) > 0:
            courses_amount = [amount for _, amount, _ in courses_s]
            courses_amount_sum = sum(courses_amount)
            res_12_unsorted.append((c.faculty, courses_amount_sum))

    return sorted(res_12_unsorted, key=itemgetter(1), reverse=True)

def sort_3(cours, many_to_many):
    res_13 = {}
    for c in cours:
        if len(c.faculty) > 0:
            courses_s = list(filter(lambda i: i[2] == c.faculty,
many_to_many))
            courses_s_name = [x for x, _, _ in courses_s]
            res_13[c.faculty] = courses_s_name

    return res_13

def main():
    one_to_many = [(g.name, g.amount, c.faculty)
        for g in group_s
        for c in courses
        if g.course_id == c.id]

    many_to_many_temp = [(c.faculty, gc_s.course_id, gc_s.group_id)
        for c in courses
        for gc_s in group_course_s
        if c.id == gc_s.course_id]

    many_to_many = [(g.name, g.amount, course_name)
        for course_name, course_id, group_id in many_to_many_temp
        for g in group_s if g.id == group_id]

    print('Задание B1')
    res_11 = sorted(one_to_many, key=itemgetter(0))
    print(res_11)
    print('\nЗадание B2\n', sort_2(courses, one_to_many))
    print('\nЗадание B3\n', sort_3(courses, many_to_many))

if __name__ == '__main__':
    main()

```

## test.py

```

import unittest
import sys, os

sys.path.append(os.getcwd())
from rkl import *

one_to_many = [(g.name, g.amount, c.faculty)
    for g in group_s
    for c in courses
    if g.course_id == c.id]

many_to_many_temp = [(c.faculty, gc_s.course_id, gc_s.group_id)
    for c in courses
    for gc_s in group_course_s
    if c.id == gc_s.course_id]

```

```

many_to_many = [(g.name, g.amount, course_name)
                 for course_name, course_id, group_id in many_to_many_temp
                 for g in group_s if g.id == group_id]

class TestCost(unittest.TestCase):
    def test_sort_name(self):
        self.assertEqual(sorted(one_to_many, key=itemgetter(0)), [('БМТ-12',
31, 'БМТ'), ('ИУ-13', 28, 'ИУ'), ('ИУ-14', 30, 'ИУ'), ('ИУ-21', 20, 'ИУ'),
('ИУ-31', 25, 'ИУ')])

    def test_sort_numb(self):
        self.assertEqual(sort_2(courses, one_to_many), [('ИУ', 103), ('ИУ',
103), ('ИУ', 103), ('БМТ', 31), ('БМТ', 31)])

    def test_sort_courses(self):
        self.assertEqual(sort_3(courses, many_to_many), {'ИУ': ['ИУ-21', 'ИУ-
14', 'ИУ-13', 'ИУ-31'], 'БМТ': ['БМТ-12']})

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

```

### 3. Результат

```

Testing started at 1:53 ...
Launching unittests with arguments python -m unittest

Ran 3 tests in 0.002s

OK

```