



**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: Смирнов Д.И., группа ИУ5Ц-51Б

Преподаватель: Гапанюк Ю.Е.

2022г.

## **Описание задания:**

Вариант А, вариант предметной области №29.

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).

## Листинг программы:

```
rk2.py
# Вариант 29 А Смирнов Д.И. ИУ5Ц-51Б

from operator import itemgetter


class
    """Кафедра"""

    def __init__(self, id, name, count_sotrud, fac_id)
        self.id = id
        self.name = name
        self.sotr = count_sotrud
        self.fac_id = fac_id


class
    """Факультет"""

    def __init__(self, id, name)
        self.id = id
        self.name = name


class
    """
    'Кафедры факультетов' для реализации
    связи многие-ко-многим
    """

    def __init__(self, id_kaf, id_fac)
        self.kaf_id = id_kaf
        self.fac_id = id_fac


#Кафедры
Kafedres
Kafedra(1, 'ИУ1-Системы автоматического управления', 12, 1),
Kafedra(2, 'ИУ2 - Приборы и системы ориентации, стабилизации и навигации', 15, 1),
Kafedra(3, 'Э9- Экология и промышленная безопасность', 10, 3),
Kafedra(4, 'ИБМ5 - Финансы', 11, 2),
Kafedra(5, 'ИБМ6 - Предпринимательство и внешнеэкономическая деятельность', 22, 2),


Facultets
Facultet(1, "Факультет: Информатика, искусственный интеллект и системы управления"),
Facultet(2, "Факультет: Инженерный бизнес и менеджмент"),
Facultet(3, "Факультет: Энергомашиностроение"),
Facultet(11, "Факультет: Информатика, искусственный интеллект и системы
управления (другие кафедры)"),
Facultet(22, "Факультет: Инженерный бизнес и менеджмент (другие кафедры)"),
Facultet(33, "Факультет: Энергомашиностроение (другие кафедры)"),


Kaf_Fac
KafFac(1, 1),
KafFac(2, 1),
KafFac(3, 3),
KafFac(4, 2),
KafFac(5, 2),
KafFac(1, 11),
KafFac(2, 11),
KafFac(3, 33),
KafFac(4, 22),
KafFac(5, 22),
```

```

# Соединение данных один-ко-многим
def      (Facultets,Kafedres)
    return  (kaf.name, kaf.sotr, fac.name)
            for fac in Facultets
            for kaf in Kafedres
            if kaf.fac_id == fac.id

def      (Facultets, Kafedres)
    print('Задание A1')
    a1 = sorted(one_to_many(Facultets,Kafedres), key=itemgetter(2))
    print(a1)
    return list(a1)
def      (Facultets, Kafedres)
    print('Задание A2')
    res_2_unsorted

    for fac in Facultets

        fac_kaf = list(filter(lambda i: i[2] == fac.name, one_to_many(Facultets,Kafedres)))
        # Если факультет не пустой
        if len(fac_kaf) > 0

            kaf_count = {}
            for _, sotr, _ in fac_kaf:
                # Суммарное количество сотрудников факультета
                fac_sotr_sum[sotr] = fac_sotr_sum.get(sotr, 0) + 1
            res_2_unsorted.append((fac.name, fac_sotr_sum))

# Сортировка по суммарному количеству
res_2 = sorted(res_2_unsorted, key=itemgetter(1), reverse=True)

print(res_2)
return list(res_2)

if __name__ == '__main__':
    a1(Facultets, Kafedres)
    a2(Facultets, Kafedres)

```

testrk2.py

```

import unittest
from rk2 import
expect_result_for_a1 = (('ИБМ5 - Финансы', 11, 'Факультет: Инженерный бизнес и менеджмент'),
                        ('ИБМ6 - Предпринимательство и внешнеэкономическая деятельность', 22, 'Факультет: Инженерный бизнес и менеджмент'),
                        ('ИУ1-Системы автоматического управления', 12, 'Факультет: Информатика, искусственный интеллект и системы управления'),
                        ('ИУ2 - Приборы и системы ориентации, стабилизации и навигации', 15, 'Факультет: Информатика, искусственный интеллект и системы управления'),
                        ('Э9- Экология и промышленная безопасность', 10, 'Факультет: Энергомашиностроение'))
expect_result_for_a2 = ('Факультет: Инженерный бизнес и менеджмент', 33),
                        ('Факультет: Информатика, искусственный интеллект и системы управления', 27),
                        ('Факультет: Энергомашиностроение', 10))
expect_result_for_func = ('ИУ1-Системы автоматического управления', 12, 'Факультет: Информатика, искусственный интеллект и системы управления'),
                          ('ИУ2 - Приборы и системы ориентации, стабилизации и навигации', 15, 'Факультет: Информатика, искусственный интеллект и системы управления'),
                          ('ИБМ5 - Финансы', 11, 'Факультет: Инженерный бизнес и менеджмент'), ('ИБМ6 - Предпринимательство и внешнеэкономическая деятельность', 22, 'Факультет: Инженерный бизнес и менеджмент'),
                          ('Э9- Экология и промышленная безопасность', 10, 'Факультет: Энергомашиностроение'))
class      (unittest.TestCase)
    def      (self)

```

```

        self.assertEqual(a1(Facultets,Kafedres),expect_result_for_a1)
def      (self)
        self.assertEqual(a2(Facultets,Kafedres),expect_result_for_a2)
def      (self)
        self.assertEqual(one_to_many(Facultets,Kafedres),expect_result_for_func)

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

```

## Результат выполнения программы:

