



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

**Рубежный контроль № 2
по дисциплине «Базовые компоненты интернет-технологий»**

**Выполнил:
студент группы ИУ5-32Б
Кульктна Д.А.**

**Проверил:
Гапанюк Ю.Е.**

2021 г.

Полученное задание:

Вариант Б.

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по сотрудникам, сортировка по отделам произвольная.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов с количеством сотрудников в каждом отделе, отсортированный по количеству сотрудников.
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех сотрудников, у которых фамилия заканчивается на «ов», и названия их отделов.

11	Программа	Компьютер
----	-----------	-----------

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. 7.
- 2) Для текста программы рубежного контроля №1 создайте модульные тесты с применением TDD - фреймворка (3 теста).

Текст кода: Main.py

```
import Classes

computers = [
    Classes.Computer(1, 'Asus'),
    Classes.Computer(2, 'HP'),
    Classes.Computer(3, 'Mac'),
    Classes.Computer(4, 'Acer')
]

programs = [
    Classes.Program(1, 'Sims4', 6, 1),
    Classes.Program(2, 'Minecraft', 4, 1),
    Classes.Program(3, 'Overcooked2', 3, 2),
    Classes.Program(4, 'NFS Head', 50, 2),
    Classes.Program(5, 'GTA', 65, 2),
    Classes.Program(6, 'Diablo3', 25, 3),
    Classes.Program(7, 'Doom Eternal', 50, 3),
    Classes.Program(8, 'Outlast', 6, 4),
    Classes.Program(9, 'Resident Evil', 45, 4),
    Classes.Program(10, 'The Last of us2', 78, 4)
]
```

```

computer_program = [
    Classes.ProCom(1, 1),
    Classes.ProCom(1, 2),
    Classes.ProCom(1, 3),
    Classes.ProCom(1, 6),
    Classes.ProCom(2, 2),
    Classes.ProCom(2, 4),
    Classes.ProCom(2, 9),
    Classes.ProCom(2, 10),
    Classes.ProCom(3, 3),
    Classes.ProCom(3, 4),
    Classes.ProCom(3, 5),
    Classes.ProCom(3, 6),
    Classes.ProCom(4, 7),
    Classes.ProCom(4, 8),
    Classes.ProCom(4, 3),
    Classes.ProCom(4, 6)
]

def task1():
    task_list = {comp.name: [prog.name for prog in programs if prog.com_id ==
comp.id] for comp in computers}
    return task_list

def task2(task1_list):
    task_list = dict(sorted(task1_list.items(), key=lambda x: len(x[1])))
    task_list = {comp: len(task_list[comp]) for comp in task_list.keys()}
    return task_list

def task3():
    task3_list={}
    for prog in programs:
        if prog.name[-1].isdigit():
            index_list = [comp.id_comp for comp in computer_program if
comp.id_prog == prog.id]
            task3_list[prog.name] = [comp.name for comp in computers if
comp.id in index_list]
    return task3_list

def main():
    print('task1')
    task1_list = task1()
    for comp, prog in task1_list.items():
        print(f'{comp}: {prog}')

    print('\n\ntask2')
    task2_list = task2(task1_list)
    for comp, prog in task2_list.items():
        print(f'{comp}: {prog}')

    print('\n\ntask3')
    task3_list = task3()
    for prog, comp in task3_list.items():
        print(f'{prog}: {comp}')

if __name__ == '__main__':
    main()

```

Файл Classes.py:

```
class Computer:
    """
        класс Компьютер
        id: id компьютера
        name: название модели
    """
    def __init__(self, id, name):
        self.id = id
        self.name = name

class Program:
    """
        класс Программа
        id: id программы
        name: название программы
        size: размер
        com_id: id компьютера, на котором установлена программа
    """
    def __init__(self, id, name, size, com_id):
        self.id = id
        self.name = name
        self.size = size
        self.com_id = com_id

class ProCom:
    """
        Класс программы-компьютеры
        id_com: id компьютера
        id_prog: id программы
    """
    def __init__(self, id_com, id_prog):
        self.id_comp = id_com
        self.id_prog = id_prog
```

test.py

```
import unittest
import main

class Test_tasks(unittest.TestCase):
    def test_task1(self):
        test_list = main.task1()
        self.assertEqual(test_list, {'Asus': ['Sims4', 'Minecraft'],
                                         'HP': ['Overcooked2', 'NFS Head', 'GTA'],
                                         'Mac': ['Diablo3', 'Doom Eternal'],
                                         'Acer': ['Outlast', 'Resident Evil', 'The
Last of us2']})

    def test_task2(self):
        temp_list = {'Asus': ['Sims4', 'Minecraft'], 'HP': ['Overcooked2',
'NFS Head', 'GTA'], 'Mac': ['Diablo3', 'Doom Eternal'], 'Acer': ['Outlast',
'Resident Evil', 'The Last of us2']}
        test_list = main.task2(temp_list)
        self.assertEqual(test_list, {'Asus': 2,
                                         'Mac': 2,
                                         'HP': 3,
                                         'Acer': 3})

    def test_task3(self):
```

```
test_list = main.task3()
self.assertEqual(test_list, {'Sims4': ['Asus'],
                              'Overcooked2': ['Asus', 'Mac', 'Acer'],
                              'Diablo3': ['Asus', 'Mac', 'Acer'],
                              'The Last of us2': ['HP']})
```

Результаты выполнения программы:

task1

Asus: ['Sims4', 'Minecraft']

HP: ['Overcooked2', 'NFS Head', 'GTA']

Mac: ['Diablo3', 'Doom Eternal']

Acer: ['Outlast', 'Resident Evil', 'The Last of us2']

task2

Asus: 2

Mac: 2

HP: 3

Acer: 3

task3

Sims4: ['Asus']

Overcooked2: ['Asus', 'Mac', 'Acer']

Diablo3: ['Asus', 'Mac', 'Acer']

The Last of us2: ['HP']

