

МГТУ им Н.Э.Баумана  
«Системы обработки информации и управления»

**Отчет**

«Рубежный контроль №1»

Дисциплина: Парадигмы и Конструкции Языков Программирования

Студент: Керимова Жанна Руслановна

Группа: ИУ5-31Б

## Текст программы

```
from operator import itemgetter

class C:
    def __init__(self, id, model, ram, os_id):
        self.id = id
        self.model = model
        self.ram = ram
        self.os_id = os_id

class OS:
    def __init__(self, id, name):
        self.id = id
        self.name = name

class C_OS:
    def __init__(self, os_id, comp_id):
        self.os_id = os_id
        self.comp_id = comp_id

Oski = [
    OS(1, 'Windows 10'),
    OS(2, 'Linux'),
    OS(3, 'macOS'),

    OS(11, 'Windows 11'),
    OS(22, 'Ubuntu'),
    OS(33, 'macOS Big Sur'),
]

Comps = [
    C(1, 'Dell XPS 13', 16, 1),
    C(2, 'MacBook Air', 32, 3),
    C(3, 'Lenovo YOGA', 8, 2),
    C(4, 'HP Spectre', 16, 3),
    C(5, 'Asus TUF GAMING F15', 32, 2),
]

comps_os = [
    C_OS(1, 1),
    C_OS(2, 2),
    C_OS(3, 3),
    C_OS(3, 4),
    C_OS(3, 5),

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

def main():
    one_to_many = [(c.model, c.ram, o.name)
                    for o in Oski]
```

```

        for c in Comps
        if c.os_id == o.id]

many_to_many_temp = [(o.name, co.os_id, co.comp_id)
                      for o in Oski
                      for co in comps_os
                      if o.id == co.os_id]

many_to_many = [(c.model, c.ram, os_name)
                 for os_name, os_id, comp_id in many_to_many_temp
                 for c in Comps if c.id == comp_id]

print('Задание A1')
res_11 = sorted(one_to_many, key=itemgetter(2))
print(res_11)

print('Задание A2')
res_12_unsorted = []

for o in Oski:

    o_comps = list(filter(lambda i: i[2] == o.name, one_to_many))

    if len(o_comps) > 0:
        o_rams = [ram for _, ram, _ in o_comps]

        o_rams_sum = sum(o_rams)
        res_12_unsorted.append((o.name, o_rams_sum))
res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=False)
print(res_12)

print('Задание A3')
res_13 = {}
for o in Oski:
    if 'Windows' in o.name or 'macOS' in o.name:
        o_comps = list(filter(lambda i: i[2] == o.name, many_to_many))
        o_comps_models = [x for x, _, _ in o_comps]
        res_13[o.name] = o_comps_models
print(res_13)

# print('Задание A3')
# res_13 = {}
# for o in Oski:
#     if 'Windows' in o.name or 'macOS' in o.name:
#         o_comps = list(filter(lambda i: i[2] == o.name, many_to_many))
#         o_comps_models = [x for x, _, _ in o_comps]
#         o_comps_models_sorted = sorted(o_comps_models)
#         res_13[o.name] = o_comps_models_sorted
# for os_name, models in res_13.items():
#     print(f'{os_name}: {', '.join(models)}")

if __name__ == '__main__':
    main()

```

## Описание Программы:

Программа работает с предметной областью по варианту (в моём случае 9)(компьютеры и операционная система).

Сначала производится определение классов: OS, C и C\_OS, каждый из которых наделён следующими атрибутами:

C: id, model, ram, os\_id

OS: id, name

C\_OS: осуществляет связь "многие-ко-многим", используя os\_id, comp\_id

В main() ведётся основная обработка данных: один ко многим и многие ко многим, далее происходит выполнение заданий 1-3. Первое задание выводит отсортированный список моделей компьютеров с объемом оперативной памяти и названием операционной системы. Второе задание выводит список операционных систем с суммарным объемом оперативной памяти всех компьютеров с этой системой, отсортированный по возрастанию суммарного объема. Третье- выводит словарь, где ключ - название операционной системы, значение - список моделей компьютеров с этой системой.

## Интересное:

Используем функции высшего порядка filter, sorted.

Я попробовала немного изменить задание 3, поэтому часть кода закомментирована, если её подставить рядом с заданием 3 в исходном коде, выведется результат как около пометки (дополнительное).

## Результат работы программы:

Задание A1

[('Lenovo YOGA', 8, 'Linux'), ('Asus TUF GAMING F15', 32, 'Linux'), ('Dell XPS 13', 16, 'Windows 10'), ('MacBook Air', 32, 'macOS'), ('HP Spectre', 16, 'macOS')]

Задание A2

[('Windows 10', 16), ('Linux', 40), ('macOS', 48)]

Задание A3

{'Windows 10': ['Dell XPS 13'], 'macOS': ['Lenovo YOGA', 'HP Spectre', 'Asus TUF GAMING F15'], 'Windows 11': ['Dell XPS 13'], 'macOS Big Sur': ['Lenovo YOGA', 'HP Spectre', 'Asus TUF GAMING F15']}

Задание A3 (дополнительное)

Windows 10: Dell XPS 13

macOS: Asus TUF GAMING F15, HP Spectre, Lenovo YOGA

Windows 11: Dell XPS 13

macOS Big Sur: Asus TUF GAMING F15, HP Spectre, Lenovo YOGA

Process finished with exit code 0

