

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
им. Н.Э. Баумана

Факультет “Информатика и системы управления”  
Кафедра “Системы обработки информации и управления”



Дисциплина “Парадигмы и конструкции языков программирования”

Отчет по рубежному контролю № 1  
Вариант Д4

**Выполнил:**  
Студент группы ИУ5-36Б  
Канаев А.И.  
**Преподаватель:**  
Гапанюк Ю.Е.

Москва 2025

# Report for RK\_1

## Variant

№ Варианта	Класс 1	Класс 2
4	Компьютер	Дисплейный класс

## Code

```
class DisplayClass:
    """Дисплейный класс"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class Computer:
    """Компьютер"""
    def __init__(self, id, name, cost, class_id):
        self.id = id
        self.name = name
        self.cost = cost
        self.class_id = class_id

class CompClass:
    """'Компьютеры в классах' для связи многие-ко-многим"""
    def __init__(self, comp_id, class_id):
        self.comp_id = comp_id
        self.class_id = class_id

# Дисплейные классы
classes = [
    DisplayClass(1, 'Аудитория 501'),
    DisplayClass(2, 'Лаборатория сетей'),
    DisplayClass(3, 'Аудитория 333'),
]

# Компьютеры
computers = [
    Computer(1, 'HP Pavilion 15s', 95000, 1),
    Computer(2, 'MacBook Pro M4', 250000, 2),
    Computer(3, 'Dell XPS 13', 150000, 1),
    Computer(4, 'Asus ROG Zephyrus', 180000, 3),
    Computer(5, 'Lenovo IdeaPad s340s', 80000, 3),
]

# Связи многие-ко-многим
comps_classes = [
    CompClass(1, 1), CompClass(2, 2), CompClass(3, 1),
    CompClass(4, 3), CompClass(5, 3),
    # Добавим для примера, что MacBook есть и в 1-м классе
    CompClass(2, 1), # mac в 1 классе
]

def main():
    """Основная функция"""

    # Д1
    # Список компьютеров, название которых заканчивается на s, и их классов
    print('Задание Д1')
    one_to_many = [(comp.name, cls.name)
                    for comp in computers
```

```

        for comp in computers
        for cls in classes
        if comp.class_id == cls.id and comp.name.endswith('s')]

print(one_to_many)
print()

# Д2
# Список классов со средней стоимостью компьютеров в каждом, отсортированный по убыванию
print('Задание Д2')
res_d2 = []
for cls in classes:
    class_comps = list(filter(lambda x: x.class_id == cls.id, computers))
    if len(class_comps) > 0:
        avg_cost = sum([comp.cost for comp in class_comps]) / len(class_comps) # сред
        res_d2.append((cls.name, int(avg_cost)))

res_d2 = sorted(res_d2, key=lambda item: item[1], reverse=True) # no dec
print(res_d2)
print()

# Д3
# Список классов, начинающихся на 'A', и список компьютеров в них
print('Задание Д3')

# {id: Computer}
comps_dict = {comp.id: comp for comp in computers}
# {id: DisplayClass}
classes_dict = {cls.id: cls for cls in classes}

many_to_many = {}
for cc in comps_classes:
    class_name = classes_dict[cc.class_id].name
    comp_name = comps_dict[cc.comp_id].name

    if class_name not in many_to_many:
        many_to_many[class_name] = []
    many_to_many[class_name].append(comp_name)

# фильтр по названию с A
res_d3 = {key: val for key, val in many_to_many.items() if key.startswith('A')}
print(res_d3)

if __name__ == '__main__':
    main()

```

## Output

Задание Д1  
 [('HP Pavilion 15s', 'Аудитория 501'), ('Asus ROG Zephyrus', 'Аудитория 333'), ('Lenovo IdeaPad s340s', 'Аудитория 333')]

Задание Д2  
 [('Лаборатория сетей', 250000), ('Аудитория 333', 130000), ('Аудитория 501', 122500)]

Задание Д3  
 {'Аудитория 501': ['HP Pavilion 15s', 'Dell XPS 13', 'MacBook Pro M4'], 'Аудитория 333': ['Asus ROG Zephyrus', 'Lenovo IdeaPad s340s']}