

**Московский государственный технический
университет им. Н.Э. Баумана**

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

Курс «Парадигмы и конструкции языков программирования»

**Отчет по РК №1
Вариант запросов: В
Вариант предметной области: 1**

Выполнил:
студент группы ИУ5-33Б
Абитов М. Р.

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю. Е.

Москва, 2023 г.

Вариант запросов В.

1. «Студент» и «Группа» связаны соотношением один-ко-многим. Выведите список всех деталей, которые начинаются с «Тормоз» и названия их производителей.
2. «Студент» и «Группа» связаны соотношением один-ко-многим. Выведите список производителей с минимальной стоимостью деталей, отсортированный их по минимальной стоимости.
3. «Студент» и «Группа» связаны соотношением многие-ко-многим. Выведите список всех связанных деталей и производителей, отсортированный по производителям, сортировка деталей по цене.

Код программы

```
class Student:
    def __init__(self, student_id, name, group_id, academic_rating):
        self.student_id = student_id
        self.name = name
        self.group_id = group_id
        self.academic_rating = academic_rating

class Group:
    def __init__(self, group_id, name):
        self.group_id = group_id
        self.name = name

# Создаем объекты класса Group
group1 = Group(1, "Группа А")
group2 = Group(2, "Группа В")
group3 = Group(3, "Группа С")

# Создаем объекты класса Student
student1 = Student(1, "Андреев", 1, 90)
student2 = Student(2, "Борисов", 1, 85)
student3 = Student(3, "Абрамов", 2, 92)
student4 = Student(4, "Григорьев", 2, 88)
student5 = Student(5, "Иванов", 3, 94)
```

```
# Создаем список "Студенты и их группы" для связи один-ко-многим
student_group = [
    (student1, group1),
    (student2, group1),
    (student3, group2),
    (student4, group2),
    (student5, group3)
]

def main():
    # Задание B1
    print("Задание B1")
    for student, group in student_group:
        if student.name.startswith('A'):
            print(f"{student.name} - {group.name}")

    # Задание B2
    print("\nЗадание B2")
    group_min_ratings = {}
    for student, group in student_group:
        if group.name in group_min_ratings:
            if student.academic_rating < group_min_ratings[group.name]:
                group_min_ratings[group.name] = student.academic_rating
        else:
            group_min_ratings[group.name] = student.academic_rating

    sorted_groups = sorted(group_min_ratings.items(), key=lambda x: x[1])
    for group, min_rating in sorted_groups:
        print(f"{group} - Минимальный рейтинг: {min_rating}")

    # Задание B3
    print("\nЗадание B3")
    for student, group in student_group:
        print(f"{student.name} - {group.name}")

if __name__ == '__main__':
    main()
```

Результат

Задание В1

Андреев - Группа А

Абрамов - Группа В

Задание В2

Группа А - Минимальный рейтинг: 85

Группа В - Минимальный рейтинг: 88

Группа С - Минимальный рейтинг: 94

Задание В3

Андреев - Группа А

Борисов - Группа А

Абрамов - Группа В

Григорьев - Группа В

Иванов - Группа С