

**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: 2

Выполнил:  
студент группы ИУ5-33Б  
Абрамов А. В.

Проверил:  
преподаватель каф. ИУ5  
Гапанюк Ю. Е.

Москва, 2023 г.

## Вариант запросов В.

1. «Школьник» и «Класс» связаны соотношением один-ко-многим. Выведите список всех школьников, фамилия которых начинается на букву «А».
2. «Школьник» и «Класс» связаны соотношением один-ко-многим. Выведите список классов с минимальным средним баллом в каждом классе, отсортированный по минимальному среднему баллу.
3. «Школьник» и «Класс» связаны соотношением многие-ко-многим. Выведите список всех связанных школьников и классов, отсортированный по школьникам, сортировка классов произвольная.

## Код программы

```
class Student:
    def __init__(self, student_id, name, class_id, avg_rating):
        self.student_id = student_id
        self.name = name
        self.class_id = class_id
        self.avg_rating = avg_rating

class Class:
    def __init__(self, class_id, name):
        self.class_id = class_id
        self.name = name

# Создаем объекты класса Class
second_B = Class(1, "2Б")
tenth_A = Class(2, "10А")
fifth_V = Class(3, "5В")

# Создаем объекты класса Student
student1 = Student(1, "Лупарев", 1, 4.9)
student2 = Student(2, "Гукасян", 1, 3.8)
student3 = Student(3, "Абрамов", 2, 4.5)
student4 = Student(4, "Иноземцев", 2, 2.7)
student5 = Student(5, "Барсукова", 3, 5.0)

# Создаем список "Школьники и классы" для связи один-ко-многим
student_class = [
    (student1, second_B),
    (student2, second_B),
    (student3, tenth_A),
    (student4, tenth_A),
    (student5, fifth_V)
]
```

```

def query1():
    # Задание B1
    print("Задание B1")
    for student, cls in student_class:
        if student.name.startswith('A'):
            print(f"{student.name} - {cls.name}")

def query2():
    # Задание B2
    print("\nЗадание B2")
    class_min_ratings = {}
    for student, cls in student_class:
        if cls.name in class_min_ratings:
            if student.avg_rating < class_min_ratings[cls.name]:
                class_min_ratings[cls.name] = student.avg_rating
        else:
            class_min_ratings[cls.name] = student.avg_rating

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

def query3():
    # Задание B3
    print("\nЗадание B3")
    student_class.sort(key=lambda x: x[0].name)
    for student, cls in student_class:
        print(f"{student.name} - {cls.name}")

def main():
    query1()
    query2()
    query3()

if __name__ == '__main__':
    main()

```

## Результат выполнения программы

```

C:\Users\Admin\PycharmProjects\RK1\venv\Scripts\python.exe C:/Users/Admin/PycharmProjects/RK1/main.py
Задание B1
Абрамов - 10А

Задание B2
10А - Минимальный рейтинг: 2.7
2Б - Минимальный рейтинг: 3.8
5В - Минимальный рейтинг: 5.0

Задание B3
Абрамов - 10А
Барсукова - 5В
Гуксян - 2Б
Иноземцев - 10А
Лупарев - 2Б

```