

**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 4

Выполнил:
студент группы ИУ5-33Б
Просвирякова Д. С.

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю. Е.

Москва, 2025 г.

Листинг кода

```
from operator import itemgetter

class Computer:
    def __init__(self, id, model, ram_gb, classroom_id):
        self.id = id
        self.model = model
        self.ram_gb = ram_gb
        self.classroom_id = classroom_id

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

class ComputerClassroom:
    def __init__(self, computer_id, classroom_id):
        self.computer_id = computer_id
        self.classroom_id = classroom_id

class DataManager:
    def __init__(self):
        self.classrooms = [
            Classroom(1, "Основной компьютерный класс"),
            Classroom(2, "Графическая лаборатория"),
            Classroom(3, "Программирование"),
            Classroom(4, "Мультимедийный центр"),
        ]

        self.computers = [
            Computer(1, "Dell Optiplex", 16, 1),
            Computer(2, "HP EliteDesk", 32, 1),
            Computer(3, "Lenovo ThinkCentre", 8, 2),
            Computer(4, "Apple iMac", 16, 2),
            Computer(5, "Asus ProArt", 64, 3),
            Computer(6, "Acer Aspire", 8, 3),
            Computer(7, "MSI Creator", 32, 4),
            Computer(8, "Acer Predator", 16, 2),
        ]

        self.computer_classrooms = [
            ComputerClassroom(1, 1),
            ComputerClassroom(2, 1),
            ComputerClassroom(3, 2),
            ComputerClassroom(4, 2),
            ComputerClassroom(5, 3),
            ComputerClassroom(6, 3),
            ComputerClassroom(7, 4),
            ComputerClassroom(8, 2),
            ComputerClassroom(1, 2),
            ComputerClassroom(5, 4),
        ]
```

```

        ComputerClassroom(8, 1),
    ]

    def get_one_to_many(self):
        return [
            (c.model, c.ram_gb, r.name)
            for r in self.classrooms
            for c in self.computers
            if c.classroom_id == r.id
        ]

    def get_many_to_many(self):
        many_to_many_temp = [
            (r.name, cr.classroom_id, cr.computer_id)
            for r in self.classrooms
            for cr in self.computer_classrooms
            if r.id == cr.classroom_id
        ]

        return [
            (c.model, c.ram_gb, classroom_name)
            for classroom_name, classroom_id, computer_id in
many_to_many_temp
            for c in self.computers if c.id == computer_id
        ]

    def task1_computers_starting_with_A(self):

        one_to_many = self.get_one_to_many()

        result = [
            (model, ram, classroom_name)
            for model, ram, classroom_name in one_to_many
            if model.startswith('A')
        ]

        return sorted(result, key=itemgetter(0))

    def task2_classrooms_min_ram(self):

        one_to_many = self.get_one_to_many()

        min_ram_by_classroom = {}

        for model, ram, classroom_name in one_to_many:
            if classroom_name not in min_ram_by_classroom:
                min_ram_by_classroom[classroom_name] = ram
            else:
                min_ram_by_classroom[classroom_name] =
min(min_ram_by_classroom[classroom_name], ram)

        result = sorted(

```

```

        [(classroom_name, min_ram) for classroom_name,
min_ram in min_ram_by_classroom.items()],
        key=itemgetter(1)
    )

    return result

def task3_all_relationships_sorted(self):

    many_to_many = self.get_many_to_many()

    result = sorted(many_to_many, key=itemgetter(0))

    return result

def main():
    manager = DataManager()

    print("=" * 70)
    print("РУБЕЖНЫЙ КОНТРОЛЬ №1 ПО КУРСУ ПИК ЯП")
    print("Предметная область: Компьютер – Дисплейный класс")
    print("Вариант В")
    print("=" * 70)

    print("\n1. Компьютеры, модель которых начинается с буквы
'A':")
    print("-" * 60)
    result1 = manager.task1_computers_starting_with_A()
    if result1:
        for model, ram, classroom_name in result1:
            print(f"Компьютер: {model}, RAM: {ram}GB, Класс:
{classroom_name}")
    else:
        print("Компьютеры с моделями на 'A' не найдены")

    print("\n2. Дисплейные классы с минимальным RAM
компьютеров:")
    print("-" * 60)
    result2 = manager.task2_classrooms_min_ram()
    for classroom_name, min_ram in result2:
        print(f"Класс: {classroom_name}, Мин. RAM: {min_ram}GB")

    print("\n3. Все связи компьютеров и дисплейных классов
(отсортировано по компьютерам):")
    print("-" * 60)
    result3 = manager.task3_all_relationships_sorted()
    for model, ram, classroom_name in result3:
        print(f"Компьютер: {model} ({ram}GB) -> Класс:
{classroom_name}")

if __name__ == '__main__':
    main()

```

Скриншот работы приложения:

```
● daraprosvirakova@MacBook-Air-Dara Prosviryakova-PCPL % python3 a.py
=====
РУБЕЖНЫЙ КОНТРОЛЬ №1 ПО КУРСУ ПИК ЯП
Предметная область: Компьютер – Дисплейный класс
Вариант В
=====

1. Компьютеры, модель которых начинается с буквы 'A':
=====
Компьютер: Acer Aspire, RAM: 8GB, Класс: Программирование
Компьютер: Acer Predator, RAM: 16GB, Класс: Графическая лаборатория
Компьютер: Apple iMac, RAM: 16GB, Класс: Графическая лаборатория
Компьютер: Asus ProArt, RAM: 64GB, Класс: Программирование

2. Дисплейные классы с минимальным RAM компьютеров:
=====
Класс: Графическая лаборатория, Мин. RAM: 8GB
Класс: Программирование, Мин. RAM: 8GB
Класс: Основной компьютерный класс, Мин. RAM: 16GB
Класс: Мультимедийный центр, Мин. RAM: 32GB

3. Все связи компьютеров и дисплейных классов (отсортировано по компьютерам):
=====
Компьютер: Acer Aspire (8GB) -> Класс: Программирование
Компьютер: Acer Predator (16GB) -> Класс: Основной компьютерный класс
Компьютер: Acer Predator (16GB) -> Класс: Графическая лаборатория
Компьютер: Apple iMac (16GB) -> Класс: Графическая лаборатория
Компьютер: Asus ProArt (64GB) -> Класс: Программирование
Компьютер: Asus ProArt (64GB) -> Класс: Мультимедийный центр
Компьютер: Dell Optiplex (16GB) -> Класс: Основной компьютерный класс
Компьютер: Dell Optiplex (16GB) -> Класс: Графическая лаборатория
Компьютер: HP EliteDesk (32GB) -> Класс: Основной компьютерный класс
Компьютер: Lenovo ThinkCentre (8GB) -> Класс: Графическая лаборатория
Компьютер: MSI Creator (32GB) -> Класс: Мультимедийный центр
```