

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

Факультет информатика и системы управления
Кафедра системы обработки информации и управления

Курс «Парадигмы и конструкции языков программирования»

Отчет по рубежному контролю №1

Вариант Б9

Выполнил:
студент группы ИУ5-32Б:
Неустроева А.В.
Подпись и дата:

Проверил:
преподаватель каф. ИУ5:
Гапанюк Ю.Е.
Подпись и дата:

Код программы

```
from dataclasses import dataclass
from typing import List


@dataclass
class OperatingSystem:
    id: int
    name: str
    version: str


@dataclass
class Computer:
    id: int
    brand: str
    ram_gb: int
    os_id: int


@dataclass
class ComputerOS:
    computer_id: int
    os_id: int


class DataManager:
    def __init__(self):
        self.operating_systems = [
            OperatingSystem(1, "Windows", "11"),
            OperatingSystem(2, "Linux", "Ubuntu 22.04"),
            OperatingSystem(3, "macOS", "Ventura"),
            OperatingSystem(4, "Windows", "10"),
        ]

        self.computers = [
            Computer(1, "Dell", 16, 1),
            Computer(2, "HP", 8, 1),
            Computer(3, "Lenovo", 32, 2),
        ]
```

```
        Computer(4, "Apple", 16, 3),
        Computer(5, "Asus", 8, 4),
    ]

self.computer_os_relations = [
    ComputerOS(1, 1),
    ComputerOS(1, 2),
    ComputerOS(2, 1),
    ComputerOS(3, 2),
    ComputerOS(3, 4),
    ComputerOS(4, 3),
    ComputerOS(5, 1),
    ComputerOS(5, 4),
]

def task_1_one_to_many(self):
    print("== ЗАПРОС 1 ==")
    print("Список всех связанных компьютеров и ОС (сортировка по компьютерам):")
    print("-" * 50)

    result = [
        (computer, next(os for os in self.operating_systems if os.id == computer.os_id))
        for computer in self.computers
    ]

    result.sort(key=lambda x: x[0].brand)

    for computer, os in result:
        print(f"Компьютер: {computer.brand} (ОЗУ: {computer.ram_gb}GB) -> "
              f"ОС: {os.name} {os.version}")
    print()

def task_2_one_to_many(self):
    print("== ЗАПРОС 2 ==")
    print("Список ОС с количеством компьютеров (сортировка по количеству):")
```

```
print("-" * 50)

os_computer_count = {
    os.name: len([comp for comp in self.computers if comp.os_id == os.id])
    for os in self.operating_systems
}

sorted_os = sorted(os_computer_count.items(), key=lambda x: x[1],
reverse=True)

for os_name, count in sorted_os:
    print(f"ОС: {os_name} -> Количество компьютеров: {count}")
print()

def task_3_many_to_many(self):
    print("==== ЗАПРОС 3 ===")
    print("Список компьютеров с брендом, заканчивающимся на 'ов', и их ОС:")
    print("-" * 50)

    computers_with_ov = [comp for comp in self.computers if
comp.brand.endswith('ов')]

    result = []
    for computer in computers_with_ov:
        os_relations = [rel for rel in self.computer_os_relations
                        if rel.computer_id == computer.id]

        operating_systems_for_computer = [
            next(os for os in self.operating_systems if os.id == rel.os_id)
            for rel in os_relations
        ]

        result.append((computer, operating_systems_for_computer))

    for computer, os_list in result:
        os_names = ", ".join([f"{os.name} {os.version}" for os in os_list])
```

```
        print(f"Компьютер: {computer.brand} (ОЗУ: {computer.ram_gb}GB) -> "
              f"Установленные ОС: {os_names}")

    if not result:
        print("Компьютеры с брендом, заканчивающимся на 'ов', не найдены.")

    print()

def display_all_data(self):
    print("==== ИСХОДНЫЕ ДАННЫЕ ===")
    print("Операционные системы:")
    for os in self.operating_systems:
        print(f" ID: {os.id}, Название: {os.name}, Версия: {os.version}")

    print("\nКомпьютеры (связь один-ко-многим):")
    for comp in self.computers:
        os_name = next(os.name for os in self.operating_systems if os.id ==
comp.os_id)
        print(f" ID: {comp.id}, Бренд: {comp.brand}, ОЗУ: {comp.ram_gb}GB, "
              f"Основная ОС: {os_name}")

    print("\nСвязи многие-ко-многим:")
    for rel in self.computer_os_relations:
        comp = next(c for c in self.computers if c.id == rel.computer_id)
        os = next(o for o in self.operating_systems if o.id == rel.os_id)
        print(f" Компьютер: {comp.brand} -> ОС: {os.name} {os.version}")

    print()

def main():
    manager = DataManager()

    manager.display_all_data()

    manager.task_1_one_to_many()
    manager.task_2_one_to_many()
    manager.task_3_many_to_many()
```

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

Результаты программы

==== ЗАПРОС 1 ====

Список всех связанных компьютеров и ОС (сортировка по компьютерам):

```
-----  
Компьютер: Apple (ОЗУ: 16GB) -> ОС: macOS Ventura  
Компьютер: Asus (ОЗУ: 8GB) -> ОС: Windows 10  
Компьютер: Dell (ОЗУ: 16GB) -> ОС: Windows 11  
Компьютер: HP (ОЗУ: 8GB) -> ОС: Windows 11  
Компьютер: Lenovo (ОЗУ: 32GB) -> ОС: Linux Ubuntu 22.04
```

==== ЗАПРОС 2 ====

Список ОС с количеством компьютеров (сортировка по количеству):

```
ОС: Linux -> Количество компьютеров: 1  
ОС: macOS -> Количество компьютеров: 1
```

==== ЗАПРОС 3 ====

Список компьютеров с брендом, заканчивающимся на 'ов', и их ОС:

```
-----  
Компьютеры с брендом, заканчивающимся на 'ов', не найдены.
```