

Московский государственный технический университет имени Н.Э. Баумана

**ОТЧЕТ**  
**Рубежный контроль №1**  
**Вариант 6. Запрос А.**

**Студент:** Гаммал Дмитрий  
Михайлович

**Группа:** ИБМ3-34Б

**Преподаватель:** Ю.Е. Гапанюк

**Кафедра:** ИУ-5

|   |     |       |
|---|-----|-------|
| 6 | Дом | Улица |
|---|-----|-------|

### Вариант А.

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по отделам, сортировка по сотрудникам произвольная.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов с суммарной зарплатой сотрудников в каждом отделе, отсортированный по суммарной зарплате.
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех отделов, у которых в названии присутствует слово «отдел», и список работающих в них сотрудников.

Код программы:

```
///
```

```
class Street:
```

```
def __init__(self, street_id: int, name: str):
    self.street_id = street_id
    self.name = name

def __str__(self):
    return f"Улица [ID: {self.street_id}, Название: {self.name}]"
```

```
class House:
```

```
def __init__(self, house_id: int, residents_count: int, street_id: int):
    self.house_id = house_id
    self.residents_count = residents_count
    self.street_id = street_id

def __str__(self):
    return f"Дом [ID: {self.house_id}, Жильцов: {self.residents_count}, ID улицы: {self.street_id}]"
```

```
class HouseOnStreet:
```

```
def __init__(self, relation_id: int, street_id: int, house_id: int):
    self.relation_id = relation_id
    self.street_id = street_id
    self.house_id = house_id

def __str__(self):
```

```
    return f"ДомНаУлице [ID связи: {self.relation_id}, ID улицы: {self.street_id}, ID дома: {self.house_id}]"
```

```
streets = [
    Street(1, "улица Ленина"),
    Street(2, "проспект Пушкина"),
    Street(3, "улица Гагарина"),
    Street(4, "площадь Советская"),
    Street(5, "улица Центральная")
]
```

```
houses = [
    House(1, 25, 1),
    House(2, 40, 1),
    House(3, 15, 2),
    House(4, 32, 3),
    House(5, 28, 4),
    House(6, 50, 5),
    House(7, 20, 3)
]
```

```
house_street_relations = [
    HouseOnStreet(1, 1, 1),
    HouseOnStreet(2, 1, 2),
    HouseOnStreet(3, 2, 3),
    HouseOnStreet(4, 3, 4),
    HouseOnStreet(5, 4, 5),
    HouseOnStreet(6, 5, 6),
    HouseOnStreet(7, 3, 7)
]
```

```
if __name__ == "__main__":
    print("Примеры данных")

    print("\n--- УЛИЦЫ ---")
    for street in streets:
        print(street)

    print("\n--- ДОМА ---")
    for house in houses:
        print(house)

    print("\n--- СВЯЗИ ДОМ-УЛИЦА ---")
    for relation in house_street_relations:
        print(relation)
```

```

print()
print("1. Список всех связанных домов и улиц, отсортированный по улицам -----")

street_houses = {}
for house in houses:
    street_name = next(s.name for s in streets if s.street_id == house.street_id)
    if house.street_id not in street_houses:
        street_houses[house.street_id] = {"name": street_name, "houses": []}
    street_houses[house.street_id]["houses"].append(house)

for street_id in sorted(street_houses.keys(), key=lambda x: street_houses[x]["name"]):
    street_data = street_houses[street_id]
    print(f"\n{street_data['name']}:")
    for house in street_data["houses"]:
        print(f" - Дом {house.house_id}")

print()
print("2. Улицы с суммарным количеством жильцов, отсортированные по количеству жильцов ---")
print()

street_residents = {}
for house in houses:
    street_name = next(s.name for s in streets if s.street_id == house.street_id)
    if house.street_id not in street_residents:
        street_residents[house.street_id] = {"name": street_name, "total_residents": 0}
    street_residents[house.street_id]["total_residents"] += house.residents_count

sorted_streets = sorted(street_residents.items(), key=lambda x: x[1]["total_residents"], reverse=True)

for street_id, data in sorted_streets:
    print(f"{data['name']}: {data['total_residents']} жильцов")

print()
print("3. Улицы с 'улица' в названии и их дома -----")

street_with_word = [street for street in streets if "улица" in street.name.lower()]

for street in street_with_word:
    print(f"\n{street.name}:")
    street_relations = [rel for rel in house_street_relations if rel.street_id == street.street_id]

    for relation in street_relations:
        house = next(h for h in houses if h.house_id == relation.house_id)
        print(f" - Дом {house.house_id}")

///

```

## Результат выполнения программы:

### Примеры данных

```
--- УЛИЦЫ ---
Улица [ID: 1, Название: улица Ленина]
Улица [ID: 2, Название: проспект Пушкина]
Улица [ID: 3, Название: улица Гагарина]
Улица [ID: 4, Название: площадь Советская]
Улица [ID: 5, Название: улица Центральная]

--- ДОМА ---
Дом [ID: 1, Жильцов: 25, ID улицы: 1]
Дом [ID: 2, Жильцов: 40, ID улицы: 1]
Дом [ID: 3, Жильцов: 15, ID улицы: 2]
Дом [ID: 4, Жильцов: 32, ID улицы: 3]
Дом [ID: 5, Жильцов: 28, ID улицы: 4]
Дом [ID: 6, Жильцов: 50, ID улицы: 5]
Дом [ID: 7, Жильцов: 20, ID улицы: 3]

--- СВЯЗИ ДОМ-УЛИЦА ---
ДомНаУлице [ID связи: 1, ID улицы: 1, ID дома: 1]
ДомНаУлице [ID связи: 2, ID улицы: 1, ID дома: 2]
ДомНаУлице [ID связи: 3, ID улицы: 2, ID дома: 3]
ДомНаУлице [ID связи: 4, ID улицы: 3, ID дома: 4]
ДомНаУлице [ID связи: 5, ID улицы: 4, ID дома: 5]
ДомНаУлице [ID связи: 6, ID улицы: 5, ID дома: 6]
ДомНаУлице [ID связи: 7, ID улицы: 3, ID дома: 7]
```

### 1. Список всех связанных домов и улиц, отсортированный по улицам -----

площадь Советская:  
- Дом 5

проспект Пушкина:  
- Дом 3

улица Гагарина:  
- Дом 4  
- Дом 7

улица Ленина:  
- Дом 1  
- Дом 2

улица Центральная:  
- Дом 6

### 2. Улицы с суммарным количеством жильцов, отсортированные по количеству жильцов -----

улица Ленина: 65 жильцов  
улица Гагарина: 52 жильцов  
улица Центральная: 50 жильцов  
площадь Советская: 28 жильцов  
проспект Пушкина: 15 жильцов

### 3. Улицы с 'улица' в названии и их дома -----

улица Ленина:  
- Дом 1  
- Дом 2

улица Гагарина:  
- Дом 4  
- Дом 7

улица Центральная:  
- Дом 6