

**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: А
Вариант предметной области: 33

Выполнил:
студент группы ИУ5-32Б
Грачев К. А.

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю. Е.

Москва, 2025 г.

Вариант запросов А. Предметная область 33.

1. «Таблица данных» и «Строка данных» связаны соотношением один-ко-многим. Выведите список всех связанных строк и таблиц, отсортированный по именам таблиц, сортировка по сотрудникам произвольная.
2. «Таблица данных» и «Строка данных» связаны соотношением один-ко-многим. Выведите список таблиц с суммой числового поля в каждой таблице, отсортированный по данной сумме.
3. «Таблица данных» и «Строка данных» связаны соотношением многие-ко-многим. Выведите список всех таблиц, у которых в названии присутствует символ «С», и содержимое текстовых полей в их строках.

Листинг программы.

```
from operator import itemgetter
```

```
class Row:
```

```
    """Строка данных"""
```

```
    def __init__(self, id, digit_field, string_field, table_id):
```

```
        self.id = id
```

```
        self.digit_field = digit_field
```

```
        self.string_field = string_field
```

```
        self.table_id = table_id
```

```
class Table:
```

```
    """Таблица данных"""
```

```
    def __init__(self, id, name):
```

```
        self.id = id
```

```
        self.name = name
```

```
class TableRow:
```

```
    """
```

```
    "Строки таблицы" для реализации  
связи многие-ко-многим  
    """
```

```
    def __init__(self, row_id, table_id):
```

```
        self.row_id = row_id
```

```
        self.table_id = table_id
```

```

tables = [
    Table(1, 'Сотрудники'),
    Table(2, 'Отделы'),
    Table(3, 'Студенты'),
    Table(4, 'Empty table')
]

rows = [
    Row(1, 100000, 'Иванов', 1),
    Row(2, 120000, 'Петров', 1),
    Row(3, 123, 'Отдел кадров', 2),
    Row(4, 1234, 'Бухгалтерия', 2),
    Row(5, 4000, 'Баранкин', 3),
    Row(6, 7000, 'Бубликов', 3)
]

table_rows = [
    TableRow(1, 1),
    TableRow(2, 1),
    TableRow(3, 2),
    TableRow(4, 2),
    TableRow(5, 3),
    TableRow(6, 3),
    TableRow(1, 3),
    TableRow(2, 3)
]

def main():
    """Основная функция"""

    # Соединение данных один-ко-многим
    one_to_many = [(table.name, row.digit_field, row.string_field)
                    for table in tables
                    for row in rows
                    if table.id == row.table_id
                    ]

    # Соединение данных многие-ко-многим
    many_to_many = [(table.name, row.digit_field, row.string_field)
                     for table_row in table_rows
                     for table in tables if table.id == table_row.table_id
                     for row in rows if row.id == table_row.row_id
                     ]

    print('Задание A1')

```

```

print(sorted(one_to_many, key=itemgetter(0)))

print('\nЗадание A2')
result = {}
for table, salary, _ in one_to_many: # Перебираем всех сотрудников
    if table in result.keys(): # Проверяем, есть ли запись о соответствующем сотруднику
отделе
        result[table] += salary # Увеличиваем сумму зарплат
    else:
        result[table] = salary # Добавляем запись об отделе
print(sorted(result.items(), key=lambda item: item[1], reverse=True)) # Выводим,
отсортированный по сумме зарплат, результат

print('\nЗадание A3')
result = {}
for table in tables:
    if 'C' in table.name:
        # Добавляем в словарь список имен сотрудников отдела
        # ключ - отдел, значение - список фамилий
        result[table.name] = [item[2] for item in filter(lambda x: x[0] == table.name,
many_to_many)]
print(result)

if __name__ == '__main__':
    main()

```

Результат выполнения.

```

(.venv) kostya@kostya-ASUS-TUF-Gaming-F15-FX506LH-FX506LH:~/ProgramsC++/Lads_3_semester/rk1$ python3 main.py
Задание A1
[('Отделы', 123, 'Отдел кадров'), ('Отделы', 1234, 'Бухгалтерия'), ('Сотрудники', 100000, 'Иванов'), ('Сотрудники', 120000, 'Петров'), ('Студенты', 4000, 'Баранкин'), ('Студенты', 7000, 'Бубликов')]

Задание A2
[('Сотрудники', 220000), ('Студенты', 11000), ('Отделы', 1357)]

Задание A3
{'Сотрудники': ['Иванов', 'Петров'], 'Студенты': ['Баранкин', 'Бубликов', 'Иванов', 'Петров']}

```

Задание A1

[('Отделы', 123, 'Отдел кадров'), ('Отделы', 1234, 'Бухгалтерия'), ('Сотрудники', 100000, 'Иванов'), ('Сотрудники', 120000, 'Петров'), ('Студенты', 4000, 'Баранкин'), ('Студенты', 7000, 'Бубликов')]

Задание A2

[('Сотрудники', 220000), ('Студенты', 11000), ('Отделы', 1357)]

Задание A3

{'Сотрудники': ['Иванов', 'Петров'], 'Студенты': ['Баранкин', 'Бубликов', 'Иванов', 'Петров']}