



**Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

---

ФАКУЛЬТЕТ «ИНЖЕНЕРНЫЙ БИЗНЕС И МЕНЕДЖМЕНТ»

КАФЕДРА «ПРОМЫШЛЕННАЯ ЛОГИСТИКА» (ИБМ-3)

## **Рубежный контроль №1**

**«Парадигмы и конструкции языков программирования»**

38.03.05 «Бизнес-информатика» (бакалавриат)

Студент ИБМ3-24Б

\_\_\_\_\_  
(Подпись, дата)

А. А. Лютикова

Руководитель

\_\_\_\_\_  
(Подпись, дата)

Ю. Е. Гапанюк

2024 г.

## Вариант 13 (А)

Необходимо составить 3 запроса для предметной области, состоящей из двух классов: «Книга», «Библиотека».

### Варианты запросов:

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.

### Текст программы:

```
from operator import itemgetter

class Book:
    """Книга"""

    def __init__(self, book_id, title, author):
        self.book_id = book_id
        self.title = title
        self.author = author

class Library:
    """Библиотека"""

    def __init__(self, library_id, name):
        self.library_id = library_id
        self.name = name

class LibraryBook:
    """Связь между библиотекой и книгой"""

    def __init__(self, book_id, library_id, quantity):
        self.book_id = book_id
        self.library_id = library_id
        self.quantity = quantity

#Библиотеки
libraries = [
    Library(1, 'Центральная библиотека'),
    Library(2, 'Студенческая библиотека'),
]
```

```

#Книги
books = [
    Book(1, '1984', 'George Orwell'),
    Book(2, 'Ulysses', 'James Joyce'),
    Book(3, 'The Bead Game', 'Hermann Hesse'),
]

library_books = [
    LibraryBook(1, 1, 3), #3 копии "1984" в Центральной библиотеке
    LibraryBook(2, 1, 2), #2 копии "Ulysses" в Центральной библиотеке
    LibraryBook(3, 2, 1), #1 копия "The Bead Game" в Студенческой библиотеке
]

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

    #Соединение данных один-ко-многим
    one_to_many = [(lb.library_id, lb.quantity, b.title)
                    for lb in library_books
                    for b in books if b.book_id == lb.book_id]

    #Соединение данных многие-ко-многим
    many_to_many_temp = [(lib.name, lb.library_id, lb.book_id)
                           for lib in libraries
                           for lb in library_books if lib.library_id ==
lb.library_id]

    many_to_many = [(b.title, b.author, lib_name)
                     for lib_name, lib_id, book_id in many_to_many_temp
                     for b in books if b.book_id == book_id]

    print('Задание A1 (экземпляры книг в каждой библиотеке с указанием
идентификатора библиотеки и количества конкретного экземпляра)')
    res_1 = sorted(one_to_many, key=itemgetter(0))
    print(res_1)

    print('\nЗадание A2 (общее количество книг в каждой библиотеке)')
    res_2_unsorted = []

    #Перебираем все библиотеки
    for lib in libraries:
        #Список книг в библиотеке
        lib_books = list(filter(lambda i: i[0] == lib.library_id,
one_to_many))
        #Если библиотека не пустая
        if len(lib_books) > 0:
            #Количество книг в библиотеке
            total_quantity = sum(qty for _, qty, _ in lib_books)
            res_2_unsorted.append((lib.name, total_quantity))

    #Сортировка по количеству книг
    res_2 = sorted(res_2_unsorted, key=itemgetter(1), reverse=True)
    print(res_2)

    print('\nЗадание A3 (названия уникальных книг, хранящихся в каждой из
библиотек)')
    res_3 = {}
    #Перебираем все библиотеки
    for lib in libraries:
        if 'библиотека' in lib.name.lower():
            #Список книг в библиотеке
            lib_books = list(filter(lambda i: i[2] == lib.name,

```

```

many_to_many))
    #Только названия книг
    book_titles = [title for title, _, _ in lib_books]
    #Добавляем результат в словарь
    res_3[lib.name] = book_titles

print(res_3)

if __name__ == '__main__':
    main()

```

## Результат выполнения:

```

"/Users/User/Desktop/ПикЯП/ПК №1/.venv/bin/python" /Users/User/Desktop/ПикЯП/ПК №1/Задания 1-3 (Вариант 13).py
Задание A1 (экземпляры книг в каждой библиотеке с указанием идентификатора библиотеки и количества конкретного экземпляра)
[(1, 3, '1984'), (1, 2, 'Ulysses'), (2, 1, 'The Bead Game')]

Задание A2 (общее количество книг в каждой библиотеке)
[('Центральная библиотека', 5), ('Студенческая библиотека', 1)]

Задание A3 (названия уникальных книг, хранящихся в каждой из библиотек)
{'Центральная библиотека': ['1984', 'Ulysses'], 'Студенческая библиотека': ['The Bead Game']}

Process finished with exit code 0

```