



**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 «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»**

Рубежный контроль №1

по дисциплине «Базовые компоненты интернет-технологий»

Выполнил:

студент группы ИУ5-35Б

Сыса П.В.

2021 г.

Рубежный контроль представляет собой разработку программы на языке Python, которая выполняет следующие действия:

1) Необходимо создать два класса данных в соответствии с Вашим вариантом предметной области, которые связаны отношениями один-ко-многим и многие-ко-многим.

2) Необходимо создать списки объектов классов, содержащих тестовые данные (3-5 записей), таким образом, чтобы первичные и вторичные ключи соответствующих записей были связаны по идентификаторам.

3) Необходимо разработать запросы в соответствии с Вашим вариантом. Запросы сформулированы в терминах классов «Сотрудник» и «Отдел», которые используются в примере. Вам нужно перенести эти требования в Ваш вариант предметной области. При разработке запросов необходимо по возможности использовать функциональные возможности языка Python (list/dict comprehensions, функции высших порядков).

Вариант Д.

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех сотрудников, у которых фамилия заканчивается на «ов», и названия их отделов.

2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов со средней зарплатой сотрудников в каждом отделе, отсортированный по средней зарплате (отдельной функции вычисления среднего значения в Python нет, нужно использовать комбинацию функций вычисления суммы и количества значений).

3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех отделов, у которых название начинается с буквы «А», и список работающих в них сотрудников.

Вариант предметной области 13.

Книга-Библиотека

В соответствии с предметной областью, задание было немного изменено:

3. «Библиотека» и «Книга» связаны соотношением многие-ко-многим. Выведите список всех отделов, у которых название начинается с буквы «б», и список работающих в них сотрудников.

```

# используется для сортировки
from operator import itemgetter

class Book:
    """Книга"""

    def __init__(self, id, name, cost, LibID):
        self.id = id
        self.name = name
        self.cost = cost
        self.LibID = LibID

class Lib:
    """Библиотека"""

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

class LibBook:
    """
    'Книги библиотеки' для реализации
    связи многие-ко-многим
    """

    def __init__(self, libID, bookID):
        self.libID = libID
        self.bookID = bookID

libs = [
    Lib(1, 'Московская библиотека'),
    Lib(2, 'Петербургская библиотека'),
    Lib(3, 'библиотека имени Пушкина'),
    # для связи многие-ко-многим:
    Lib(11, 'Новгородская библиотека'),
    Lib(22, 'библиотека имени Лермонтова'),
    Lib(33, 'библиотека имени Толстого'),
]

# Сотрудники
books = [
    Book(1, 'Война и мир', 1000, 1),
    Book(2, 'Капитанская дочка', 200, 2),
    Book(3, 'Богач, бедняк', 600, 2),
    Book(4, 'Преступление и наказание', 500, 3),
    Book(5, 'Гордость и предубеждение', 400, 3),
]

libBooks = [
    LibBook(3, 1),
    LibBook(2, 2),
    LibBook(2, 2),
    LibBook(3, 3),
    LibBook(3, 3),

    LibBook(11, 1),
    LibBook(22, 2),
    LibBook(22, 3),
    LibBook(33, 4),
    LibBook(33, 5),
]

```

```

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

    # Соединение данных один-ко-многим
    one_to_many = [(h.name, h.cost, l.name)
                    for l in libs
                    for h in books
                    if h.LibID == l.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(l.name, lb.libID, lb.bookID)
                          for l in libs
                          for lb in libBooks
                          if l.id == lb.libID]

    many_to_many = [(b.name, b.cost, bookName)
                    for bookName, streetID, bookID in many_to_many_temp
                    for b in books if b.id == bookID]

    print('Задание D1')
    res1 = list(filter(lambda x: x[0].endswith("ие"), one_to_many))
    print(res1)

    print('\nЗадание D2')
    res2unsorted = []
    # Перебираем все библиотеки
    for l in libs:
        # Список книг в библиотеке
        bookss = list(filter(lambda i: i[2] == l.name, one_to_many))
        # Если на улице есть дома
        if len(bookss) > 0:
            # Все цены книг в библиотеке
            allCosts = [sal for _, sal, _ in bookss]
            # Средняя цена книги в библиотеке
            averageCosts = round(sum(allCosts) / len(allCosts), 2)
            res2unsorted.append((l.name, averageCosts))

    # Сортировка по средней стоимости
    res2 = sorted(res2unsorted, key=itemgetter(1), reverse=True)
    print(res2)
    print('\nЗадание D3')
    res3 = {}
    for l in libs:
        if l.name.startswith("б"):
            # Список книг в библиотеке
            bookss = list(filter(lambda i: i[2] == l.name, many_to_many))
            # Только имя книг
            booksNames = [x for x, _, _ in bookss]
            # Добавляем результат в словарь
            # ключ - библиотека, значение - список названий книг
            res3[l.name] = booksNames

    print(res3)

if __name__ == '__main__':
    main()

```

Результаты выполнения:

```
Задание D1
[['Преступление и наказание', 500, 'библиотека имени Пушкина'], ('Гордость и предубеждение', 400, 'библиотека имени Пушкина')]

Задание D2
[['Московская библиотека', 1000.0), ('библиотека имени Пушкина', 450.0), ('Петербургская библиотека', 400.0)]

Задание D3
{'библиотека имени Пушкина': ['Война и мир', 'Богач, бедняк', 'Богач, бедняк'], 'библиотека имени Лермонтова': ['Капитанская дочка', 'Богач, бедняк'],
```

Продолжение результата выполнения задания D3

```
'Богач, бедняк'], 'библиотека имени Толстого': ['Преступление и наказание', 'Гордость и предубеждение']}]
```