

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

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

Курс «Разработка интернет приложений»

**Отчет по рубежному контролю №1
Вариант В-13**

Выполнил:
студент группы ИУ5-53Б
Кузнецов Г.И.

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.

Москва, 2021 г.

Описание предметной области:

Существуют две сущности: Книга и Библиотека.

Книга имеет следующие атрибуты: номер (id), фамилия автора (author_name), количество страниц (pages), жанр (genre) и номер связанной библиотеки для связи один-ко-многим (lib_id).

Библиотека имеет следующие атрибуты: номер (id) и название (name).

Задание сущностей и список их значений в Python:

```
class library:
    def __init__(self, id, name):
        self.id = id
        self.name = name

    def values(self):
        return (self.id, self.name)

class book:
    def __init__(self, id, author_name, pages_amount, genre, library_id):
        self.id = id
        self.author_name = author_name
        self.pages = pages_amount
        self.genre = genre
        self.lib_id = library_id

class books_libs:
    def __init__(self, book_id, lib_id):
        self.book_id = book_id
        self.lib_id = lib_id

#Библиотеки
libs = [
    library(1, 'Библиотека 1'),
    library(2, 'Библиотека 2'),
    library(3, 'Библиотека 3')
]

#Книги
books = [
    book(1, 'Иванов', 450, 'Пособия', 1),
    book(2, 'Петров', 300, 'Пособия', 1),
    book(3, 'Абаимов', 276, 'Романы', 2),
    book(4, 'Авдонин', 354, 'Романы', 3),
    book(5, 'Васнецов', 125, 'Поэмы', 3)
]
```

Задание В1:

«Библиотека» и «Книга» связаны соотношением один-ко-многим. Выведите список всех книг, фамилия автора которых начинается с буквы «А», и названия их библиотек.

Реализация на Python:

```
one_to_many = [(lib.id, lib.name, b.author_name, b.pages, b.genre)
                for lib in libs
                for b in books
                if b.lib_id == lib.id]

#Задание A1
print('Задание B1')
a1 = list(filter(lambda x : (str)(x[2]).startswith('A'), one_to_many))
a1 = [(el[2], el[1]) for el in a1]
print(a1)
```

Результат:

Задание B1

[('Абаимов', 'Библиотека 2'), ('Авдонин', 'Библиотека 3')]

Задание B2:

«Библиотека» и «Книга» связаны соотношением один-ко-многим. Выведите список библиотек с минимальным количеством страниц из всех книг в соответствующей библиотеке, отсортированный по минимальному количеству страниц.

Реализация на Python:

```
#Задание B2
print('Задание B2')
a2 = []
for lib in libs:
    lib_books = list(filter(lambda x: x.lib_id == lib.id, books))
    if len(lib_books) > 0:
        min_pages = min([b.pages for b in lib_books])
        a2.append((*lib.values(), min_pages))

a2 = sorted(a2, key= lambda x: x[2])
print(a2)
```

Результат:

Задание B2

[(3, 'Библиотека 3', 125), (2, 'Библиотека 2', 276), (1, 'Библиотека 1', 300)]

Задание B3:

«Библиотека» и «Книга» связаны соотношением многие-ко-многим. Выведите список всех связанных книг и библиотек, отсортированный по книгам, сортировка по библиотекам произвольная.

Для реализации связи многие-ко-многим зададим промежуточную сущность:

```
class books_libs:
    def __init__(self, book_id, lib_id):
        self.book_id = book_id
        self.lib_id = lib_id
```

Список связей:

```
#Книги в библиотеках (много-ко-многим)
books_in_libs = [
    books_libs(1, 1),
    books_libs(1, 2),
    books_libs(2, 2),
    books_libs(2, 3),
    books_libs(3, 1),
    books_libs(3, 2),
    books_libs(3, 3),
    books_libs(4, 1),
    books_libs(5, 2),
    books_libs(5, 3)
]
```

Реализация задания:

```
#Задание В3
print('Задание В3')
a3 = {}
for book in books:
    booklib = list(filter(lambda x: x.book_id == book.id, books_in_libs))
    a3[book.author_name] = [
        l.name for l in libs
        for bl in booklib
        if bl.lib_id == l.id]
a3 = {i[0]: i[1] for i in sorted(a3.items(), key=lambda x: x[0])}
print(a3)
```

Результат:

```
{'Абаимов': ['Библиотека 1', 'Библиотека 2', 'Библиотека 3'], 'Авдонин':
['Библиотека 1'], 'Васнецов': ['Библиотека 2', 'Библиотека 3'], 'Иванов':
['Библиотека 1', 'Библиотека 2'], 'Петров': ['Библиотека 2',
'Библиотека 3']}
```