

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

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

Курс «Базовые компоненты интернет технологий»

Отчёт по рубежному контролю №1

Выполнил:

студент группы ИУ5-31Б

Прошкин Георгий

Подпись и дата:

Проверил:

преподаватель каф. ИУ5

Подпись и дата:

Москва, 2021 г.

Вариант – А16

Условия рубежного контроля №1 по курсу БКИТ

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

1) Необходимо создать два класса данных в соответствии с предметной областью “Книга – Книжный магазин”, которые связаны отношениями один-ко-многим и многие-ко-многим.

1. Класс «Книга», содержащий поля:

- ID записи о книге;
- Название книги;
- Цена (количественный признак);
- ID записи о книжном магазине. (для реализации связи один-ко-многим)

2. Класс «Книжный магазин», содержащий поля:

- ID записи о книжном магазине;
- Название книжного магазина

3. (Для реализации связи многие-ко-многим) Класс «Книга в книжном магазине», содержащий поля:

- ID записи о книге;
- ID записи о книжном магазине.

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 предметной областью. При разработке запросов необходимо по возможности использовать функциональные возможности языка Python (list/dict comprehensions, функции высшего порядка).

Запросы

1. «Книжный магазин» и «Книга» связаны соотношением один-ко-многим. Выведите список всех связанных книг и книжных магазинов, отсортированный по книжным магазинам, сортировка по книгам произвольная.
2. «Книжный магазин» и «Книга» связаны соотношением один-ко-многим. Выведите список книжных магазинов со средней стоимостью книги в каждом книжном магазине, отсортированный по средней стоимости (от меньшей к большей). Для более объективного сравнения средних стоимостей книг в магазинах, во всех магазинах продаются одинаковые книги.
3. «Книжный магазин» и «Книга» связаны соотношением многие-ко-многим. Выведите список всех книжных магазинов, у которых в названии присутствует слово «магазин», и список их ассортимента.

Текст программы

main.py

```
from operator import itemgetter

class Book:
    """Книга"""

    def __init__(self, id, b_name, price, shop_id):
        self.id = id
        self.b_name = b_name
        self.price = price
        self.shop_id = shop_id

class Shop:
    """Магазин"""

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

class BookShop:
    """
    'Книга в магазине' для реализации
    связи многие-ко-многим
    """

    def __init__(self, shop_id, book_id):
        self.shop_id = shop_id
        self.book_id = book_id

# Магазины
shops = [
```

```

    Shop(1, 'центральный магазин'),
    Shop(2, 'магазин знаний'),
    Shop(3, 'дом книги'),
]

# Книги
books = [
    Book(1, 'Преступление и наказание', 500, 1),
    Book(2, 'На дне', 700, 2),
    Book(3, 'Мцыри', 450, 3),
    Book(4, 'Преступление и наказание', 480, 2),
    Book(5, 'На дне', 710, 3),
    Book(6, 'Мцыри', 465, 1),
    Book(7, 'Преступление и наказание', 480, 3),
    Book(8, 'На дне', 690, 1),
    Book(9, 'Мцыри', 445, 2),
]

books_shops = [
    BookShop(1, 1),
    BookShop(1, 6),
    BookShop(1, 8),
    BookShop(2, 2),
    BookShop(2, 4),
    BookShop(2, 9),
    BookShop(3, 3),
    BookShop(3, 5),
    BookShop(3, 7),
]

def main():
    one_to_many = [(b.b_name, b.price, s.name)
                    for s in shops
                    for b in books
                    if b.shop_id == s.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(s.name, b.shop_id, b.book_id)
                          for s in shops
                          for b in books_shops
                          if s.id == b.shop_id]

    many_to_many = [(b.b_name, b.price, shop_name)
                     for shop_name, shop_id, book_id in many_to_many_temp
                     for b in books if b.id == book_id]

    print('Задание A1')
    res_11 = sorted(one_to_many, key=itemgetter(2))
    print(res_11)

    print('\nЗадание A2')
    res_12_unsorted = []
    # Перебираем все магазины
    for s in shops:
        # Список книг в магазине
        s_books = list(filter(lambda i: i[2] == s.name, one_to_many))
        # Если магазин не пустой
        if len(s_books) > 0:
            # Цены книг в магазине
            s_price = [price for _, price, _ in s_books]
            # Средняя цена книги в магазине

```

```

        s_price_sum_sr = sum(s_price)//len(s_books)
        res_12_unsorted.append((s.name, s_price_sum_sr))

# Сортировка по средней цене
res_12 = sorted(res_12_unsorted, key=itemgetter(1))
print(res_12)

print('\nЗадание A3')
res_13 = {}
# Перебираем все магазины
for s in shops:
    if 'магазин' in s.name:
        # Список книг магазина
        s_books = list(filter(lambda i: i[2] == s.name, many_to_many))
        # Только название книг
        s_books_names = [x for x, _, _ in s_books]
        # Добавляем результат в словарь
        # ключ - магазин, значение - список названий книг
        res_13[s.name] = s_books_names

print(res_13)

if __name__ == '__main__':
    main()

```

Результаты выполнения программы

Задание A1

[('Мцыри', 450, 'дом книги'), ('На дне', 710, 'дом книги'), ('Преступление и наказание', 480, 'дом книги'), ('На дне', 700, 'магазин знаний'), ('Преступление и наказание', 480, 'магазин знаний'), ('Мцыри', 445, 'магазин знаний'), ('Преступление и наказание', 500, 'центральный магазин'), ('Мцыри', 465, 'центральный магазин'), ('На дне', 690, 'центральный магазин')]

Задание A2

[('магазин знаний', 541), ('дом книги', 546), ('центральный магазин', 551)]

Задание A3

{'центральный магазин': ['Преступление и наказание', 'Мцыри', 'На дне'], 'магазин знаний': ['На дне', 'Преступление и наказание', 'Мцыри']}