Кулаков Н.М.

ИУ5-36Б

Рубежный контроль №1.

Вариант 16(Д).

1. Исходный текст

```python
from operator import itemgetter
from statistics import mean


class Book:
    """Класс Книга"""
    def __init__(self, id, title, price, bookstore_id):
        self.id = id
        self.title = title
        self.price = price
        self.bookstore_id = bookstore_id


class Bookstore:
    """Класс Книжный магазин"""
    def __init__(self, id, name):
        self.id = id
        self.name = name


class BookBookstore:
    """Связь Книги и Книжного магазина для реализации многие-ко-многим"""
```

```
    def __init__(self, bookstore_id, book_id):
        self.bookstore_id = bookstore_id
        self.book_id = book_id


# Создание тестовых данных для книжных магазинов
bookstores = [
    Bookstore(1, "Академическая литература"),
    Bookstore(2, "Книжный мир"),
    Bookstore(3, "Азбука знаний")
]


# Создание тестовых данных для книг
books = [
    Book(1, "Философия науки", 500, 1),
    Book(2, "Математика для всех", 400, 1),
    Book(3, "История России", 700, 2),
    Book(4, "Алгебра и анализ", 450, 2),
    Book(5, "Биология", 550, 3)
]


# Связь многие-ко-многим
books_bookstores = [
    BookBookstore(1, 1),
    BookBookstore(1, 2),
    BookBookstore(2, 3),
```

```
    BookBookstore(2, 4),
    BookBookstore(3, 5)
]


def main():
    # Запрос 1: Все книги, которые есть в каждом магазине (связь один-ко-многим)
    one_to_many = [(b.title, b.price, s.name)
                   for s in bookstores
                   for b in books
                   if b.bookstore_id == s.id]


    # Запрос 2: Средняя цена книг в каждом магазине, отсортированная по средней цене
    res_2_unsorted = []
    for s in bookstores:
        # Все книги в текущем магазине
        s_books = list(filter(lambda i: i[2] == s.name, one_to_many))
        # Если книги есть
        if len(s_books) > 0:
            # Цены книг в магазине
            s_prices = [price for _, price, _ in s_books]
            # Средняя цена
            avg_price = sum(s_prices) / len(s_prices)
            res_2_unsorted.append((s.name, avg_price))
```

```
    # Сортировка по средней цене
    res_2 = sorted(res_2_unsorted, key=itemgetter(1))

    # Запрос 3: Список всех магазинов, название которых начинается на "А", и список книг в них
    res_3 = {}
    for s in bookstores:
        if s.name.startswith("А"):
            # Книги в данном магазине
            s_books = list(filter(lambda i: i[2] == s.name, one_to_many))
            # Список названий книг
            s_books_titles = [x for x, _, _ in s_books]
            res_3[s.name] = s_books_titles
    # Вывод результатов
    print("Запрос 1: Список книг и магазинов (связь один-ко-многим):")
    print(one_to_many)

    print("\nЗапрос 2: Средняя цена книг в каждом магазине, отсортированная по средней цене:")
    print(res_2)

    print("\nЗапрос 3: Магазины, название которых начинается с 'А', и список книг в них:")
    print(res_3)


if __name__ == "__main__":
    main()
```

Вывод:

Запрос 1: Список книг и магазинов (связь один-ко-многим):

[('Философия науки', 500, 'Академическая литература'), ('Математика для всех', 400, 'Академическая литература'), ('История России', 700, 'Книжный мир'), ('Алгебра и анализ', 450, 'Книжный мир'), ('Биология', 550, 'Азбука знаний')]

Запрос 2: Средняя цена книг в каждом магазине, отсортированная по средней цене:

[('Академическая литература', 450.0), ('Азбука знаний', 550.0), ('Книжный мир', 575.0)]

Запрос 3: Магазины, название которых начинается с 'А', и список книг в них:

{'Академическая литература': ['Философия науки', 'Математика для всех'], 'Азбука знаний': ['Биология']}