

Группа: ИУ5-31Б

Студент: Нагдасёв Дмитрий

Вариант: 16А

Текст программы:

```
from operator import itemgetter

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

class BookShop:
    """Книжный магазин"""
    def __init__(self, id, shop_name):
        self.id = id
        self.shop_name = shop_name

class Sales:
    """'Книги в магазинах для реализации связи многие-ко-многим'"""
    def __init__(self, book_id, shop_id):
        self.book_id = book_id
        self.shop_id = shop_id

books = [
    Book(1, "Python 101", 25.99, 1),
    Book(2, "Advanced Python", 35.50, 2),
    Book(3, "Learning AI", 40.00, 1),
    Book(21, "Advanced C++", 50.0, 4),
    Book(22, "Rust For Beginners", 35.99, 5),
    Book(23, "ML: Profile Level", 49.99, 3)
]

bookshops = [
    BookShop(1, "Tech Books"),
    BookShop(2, "Programming Hub"),
    BookShop(3, "AI Books"),
    BookShop(4, "IT Heaven"),
    BookShop(5, "Book++")
]

sales = [
    Sales(1, 1),
    Sales(2, 2),
```

```

    Sales(3, 1),
    Sales(3, 2),
    Sales(3, 5),
    Sales(21, 4),
    Sales(22, 2),
    Sales(22, 1),
    Sales(23, 3),
    Sales(23, 3)
]

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

    # Соединение данных один-ко-многим
    one_to_many = [(b.title, b.price, bs.shop_name)
                    for bs in bookshops
                    for b in books
                    if b.shop_id == bs.id
                   ]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(bs.shop_name, bb.shop_id, bb.book_id)
                          for bs in bookshops
                          for bb in sales
                          if bs.id == bb.shop_id
                         ]

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

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

    print("\nA2")
    res_12_unsorted = []
    for bs in bookshops:
        bs_books = list(filter(lambda i: i[2] == bs.shop_name, one_to_many))
        if len(bs_books) > 0:
            bs_prices = [price for _, price, _ in bs_books]
            bs_prices_sum = sum(bs_prices)
            res_12_unsorted.append((bs.shop_name, bs_prices_sum))

    res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
    print(res_12)

```

```

print("\nA3")
res_13 = {}
for bs in bookshops:
    if "Book" in bs.shop_name:
        bs_books = list(filter(lambda i: i[2] == bs.shop_name,
many_to_many))
        bs_books_titles = [x for x,_,_ in bs_books]
        res_13[bs.shop_name] = bs_books_titles

print(res_13)

if __name__ == '__main__':
    main()

```

Результат выполнения:

A1

[('ML: Profile Level', 49.99, 'AI Books'), ('Rust For Beginners', 35.99, 'Book++'), ('Advanced C++', 50.0, 'IT Heaven'), ('Advanced Python', 35.5, 'Programming Hub'), ('Python 101', 25.99, 'Tech Books'), ('Learning AI', 40.0, 'Tech Books')]

A2

[('Tech Books', 65.99), ('IT Heaven', 50.0), ('AI Books', 49.99), ('Book++', 35.99), ('Programming Hub', 35.5)]

A3

{'Tech Books': ['Python 101', 'Learning AI', 'Rust For Beginners'], 'AI Books': ['ML: Profile Level', 'ML: Profile Level'], 'Book++': ['Learning AI']}

```

A1
[('ML: Profile Level', 49.99, 'AI Books'), ('Rust For Beginners', 35.99, 'Book++'), ('Advanced C++', 50.0, 'IT Heaven'), ('Advanced Python', 35.5, 'Programming Hub'), ('Python 101', 25.99, 'Tech Books'), ('Learning AI', 40.0, 'Tech Books')]

A2
[('Tech Books', 65.99), ('IT Heaven', 50.0), ('AI Books', 49.99), ('Book++', 35.99), ('Programming Hub', 35.5)]

A3
{'Tech Books': ['Python 101', 'Learning AI', 'Rust For Beginners'], 'AI Books': ['ML: Profile Level', 'ML: Profile Level'], 'Book++': ['Learning AI']}

```