



**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)**

**Отчет
РК №1
По курсу «Разработка интернет-приложений»**

Вариант № А-24

Студент Дубянский А. И., ИУ5Ц-71Б.

(Подпись, дата)

Преподаватель Гапанюк Ю.Е.

(Подпись, дата)

Москва - 2020

Вариант А-24

1. «Книга» и «Глава» связаны соотношением один-ко-многим. Выведите список всех связанных глав и книг, отсортированный по книгам, сортировка по главам произвольная.
2. «Книга» и «Глава» связаны соотношением один-ко-многим. Выведите список книг с суммарной стоимостью глав, отсортированный по суммарному количеству глав.
3. «Книга» и «Глава» связаны соотношением многие-ко-многим. Выведите список всех книг, у которых в названии присутствует слово «Книга», и список глав.

Код программы:

```
from operator import itemgetter

class Book:
    '''Книга'''
    def __init__(self, id, title):
        self.id = id
        self.title = title

class Chapter:
    '''Глава'''
    def __init__(self, id, bookId, price):
        self.bookId = bookId
        self.id = id
        self.price = price

class BookChapter:
    '''Связь между главой и книгой'''
    def __init__(self, bookId, chapterId):
        self.bookId = bookId
        self.chapterId = chapterId
```

```
# Книги
books = [
    Book(1, 'Властелин колец'),
    Book(2, 'Гамлет'),
    Book(3, 'Книга Еноха'),

    Book(4, 'Властелин колец (на языке оригинала)'),
    Book(5, 'Гамлет (на языке оригинала)'),
    Book(6, 'Книга Еноха (на языке оригинала)'),
    Book(7, 'Астрология для чайников')
]

# Главы
chapters = [
    Chapter(1, 1, 500),
    Chapter(2, 1, 400),

    Chapter(3, 2, 200),
    Chapter(4, 2, 1000),

    Chapter(5, 3, 500),
    Chapter(6, 3, 600),
    Chapter(7, 3, 40),

    Chapter(8, 7, 40000)
]

# Связи
booksChapters = [
    BookChapter(1, 1),
    BookChapter(1, 2),
    BookChapter(2, 3),
    BookChapter(2, 4),
    BookChapter(3, 5),
    BookChapter(3, 6),
    BookChapter(3, 7),
```

```
BookChapter(7, 8),
```

```
BookChapter(4, 1),
```

```
BookChapter(4, 2),
```

```
BookChapter(5, 3),
```

```
BookChapter(5, 4),
```

```
BookChapter(6, 5),
```

```
BookChapter(6, 6),
```

```
BookChapter(6, 7),
```

```
]
```

```
def main():
```

```
    oneToMany = [
```

```
        (book.id, book.title, chapter.id, chapter.price)
```

```
        for book in books for chapter in chapters if book.id ==  
chapter.bookId
```

```
    ]
```

```
    manyToManyPrevious = [
```

```
        (book.title, relation.bookId, relation.chapterId)
```

```
        for book in books for relation in booksChapters if book.id ==  
relation.bookId
```

```
    ]
```

```
    manyToMany = [
```

```
        (bookId, title, chapterId, chapter.price)
```

```
        for title, bookId, chapterId in manyToManyPrevious for chapter  
in chapters if chapter.id == chapterId
```

```
    ]
```

```
# Задание A-24-1
```

```
print('Задание A-24-1\n')
```

```
result_1_temp = sorted(oneToMany, key=itemgetter(1))
```

```
result_1 = [(title, chapter, price) for _, title, chapter, price in  
result_1_temp]
```

```

print(result_1, '\n')

print('Задание A-24-2\n')
result_2_temp = []
for book in books:
    chaptersOfTheBook = list(filter(lambda x: x[0] == book.id,
oneToMany))
    if len(chaptersOfTheBook) > 0:
        priceList = [price for _, _, _, price in chaptersOfTheBook]
        generalPrice = sum(priceList)
        chaptersCount = len(priceList)
        result_2_temp.append((book.id, book.title, generalPrice,
chaptersCount))

    result_2_temp = sorted(result_2_temp, key=itemgetter(3),
reverse=True)
    result_2 = [(title, generalPrice) for _, title, generalPrice, _ in
result_2_temp]
    print(result_2, '\n')

# Задание A-24-3

print('Задание A-24-3\n')
result_3 = {}
token = 'Книга'
for book in books:
    if token in book.title:
        necessaryBookChapters = list(filter(lambda x: book.id ==
x[0], manyToMany))
        necessaryChapters = [x for _, _, x, _ in
necessaryBookChapters]
        result_3[book.title] = necessaryChapters

    print(result_3, '\n')

if __name__ == '__main__':
    main()

```

Вывод программы:

Задание А-24-1

[('Астрология для чайников', 8, 40000), ('Властелин колец', 1, 500), ('Властелин колец', 2, 400), ('Гамлет', 3, 200), ('Гамлет', 4, 1000), ('Книга Еноха', 5, 500), ('Книга Еноха', 6, 600), ('Книга Еноха', 7, 40)]

Задание А-24-2

[('Книга Еноха', 1140), ('Властелин колец', 900), ('Гамлет', 1200), ('Астрология для чайников', 40000)]

Задание А-24-3

{'Книга Еноха': [5, 6, 7], 'Книга Еноха (на языке оригинала)': [5, 6, 7]}