

## Вариант 23.

Классы «Книга» и «Глава».

Текст программы:

```
1 from operator import itemgetter
2
3
4 class Book:
5     def __init__(self, id, title):
6         self.title = title
7         self.id = id
8
9
10 class Chapter:
11     def __init__(self, chapter_id, name, pages, book_id):
12         self.chapter_id = chapter_id
13         self.name = name
14         self.pages = pages
15         self.book_id = book_id
16
17
18 class Chap_of_book:
19     def __init__(self, book_id, chapter_id):
20         self.book_id = book_id
21         self.chapter_id = chapter_id
22
23
24 def build_relationships(books, chapters, chap_books):
25     # Создаем отношение "один ко многим"
26     one_to_many = [(ch.name, ch.pages, b.title)
27                     for b in books
28                     for ch in chapters
29                     if ch.book_id == b.id]
30
31     # Создаем временное отношение "многие ко многим"
32     many_to_many_temp = [(b.title, chb.book_id, chb.chapter_id)
33                           for b in books
34                           for chb in chap_books
35                           if b.id == chb.book_id]
36
37     # Создаем отношение "многие ко многим"
38     many_to_many = [(ch.name, ch.pages, book_title)
39                     for book_title, book_id, chapter_id in many_to_many_temp
40                     for ch in chapters if ch.chapter_id == chapter_id]
41
42     return one_to_many, many_to_many
43
```

```

44
45 # Задание A1:
46 def task_a1(one_to_many):
47     res_11 = sorted(one_to_many, key=itemgetter(2))
48     return res_11
49
50
51 # Задание A2:
52 def task_a2(books, one_to_many):
53     res_12_unsorted = []
54     # Перебираем все книги
55     for b in books:
56         # Список глав книги
57         b_chps = list(filter(lambda i: i[2] == b.title, one_to_many))
58         # Если книга с главами
59         if len(b_chps) > 0:
60             # Страницы глав этой книги
61             b_pages = [pages for _, pages, _ in b_chps]
62             # Общее количество страниц во всех главах книги
63             b_pages_sum = sum(b_pages)
64             res_12_unsorted.append((b.title, b_pages_sum))
65
66     # Сортировка по суммарному количеству страниц
67     res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
68     return res_12
69
70
71 # Задание A3:
72 def task_a3(books, many_to_many):
73     res_13 = {}
74     # Перебираем все книги
75     for b in books:
76         if 'Гарри Поттер' in b.title:
77             # Список глав книги
78             b_chps = list(filter(lambda i: i[2] == b.title, many_to_many))
79             # Только названия глав
80             b_chps_names = [x for x, _, _ in b_chps]
81             # Добавляем результат в словарь
82             # ключ - книга, значение - список глав
83             res_13[b.title] = b_chps_names
84     return res_13
85

```

```

87 def main():
88     books = [
89         Book(1, 'Город костей'),
90         Book(2, 'Гарри Поттер от издательства N'),
91         Book(3, 'Полианна'),
92         Book(22, 'Гарри Поттер от издательства M'),
93         Book(33, 'Гарри Поттер от издательства P'),
94     ]
95
96     chapters = [
97         Chapter(1, 'Хранитель ключей', 21, 2),
98         Chapter(2, 'Запретный лес', 17, 2),
99         Chapter(3, 'Тайны и ложь', 5, 1),
100        Chapter(4, 'Сумеречный охотник', 10, 1),
101        Chapter(5, 'Мисс Полли', 6, 3)
102    ]
103
104    chap_books = [
105        Chap_of_book(1, 3),
106        Chap_of_book(1, 4),
107        Chap_of_book(2, 1),
108        Chap_of_book(2, 2),
109        Chap_of_book(3, 5),
110
111        Chap_of_book(22, 1),
112        Chap_of_book(33, 2),
113    ]
114
115    one_to_many, many_to_many = build_relationships(books, chapters, chap_books)
116
117    print('\nЗадание A1:')
118    res_11 = task_a1(one_to_many)
119    for item in res_11:
120        print(item)
121
122    print('\nЗадание A2:')
123    res_12 = task_a2(books, one_to_many)
124    for item in res_12:
125        print(item)
126
127    print('\nЗадание A3:')
128    res_13 = task_a3(books, many_to_many)
129    for key, value in res_13.items():
130        print(f'{key}: {value}')

```

```

132
133 if __name__ == '__main__':
134     main()
135

```

## Результат выполнения:

```
Задание A1:  
( 'Хранитель ключей', 21, 'Гарри Поттер от издательства N')  
( 'Запретный лес', 17, 'Гарри Поттер от издательства N')  
( 'Тайны и ложь', 5, 'Город костей')  
( 'Сумеречный охотник', 10, 'Город костей')  
( 'Мисс Полли', 6, 'Полианна')  
  
Задание A2:  
( 'Гарри Поттер от издательства N', 38)  
( 'Город костей', 15)  
( 'Полианна', 6)  
  
Задание A3:  
Гарри Поттер от издательства N: [ 'Хранитель ключей', 'Запретный лес']  
Гарри Поттер от издательства M: [ 'Хранитель ключей']  
Гарри Поттер от издательства P: [ 'Запретный лес']
```

>>>