Группа: ИУ5-32Б  
Студент: Яковлев Сергей

**Рубежный контроль 1**

|  |  |  |
| --- | --- | --- |
| **24** | **Глава** | **Книга** |

Вариант Б.

1. «Книга» и «Глава» связаны соотношением один-ко-многим. Выведите список всех связанных глав и книг, отсортированный по главам, сортировка по отделам произвольная.
2. «Книга» и «Глава» связаны соотношением один-ко-многим. Выведите список книг с количеством глав в каждой книге, отсортированный по количеству глав.
3. «Книга» и «Глава» связаны соотношением многие-ко-многим. Выведите список всех глав, у которых “название” заканчивается на «2», и названия их книг. (Т.к. адекватно представить главы, имеющие одинаковые названия, которые могут присутствовать в разных книгах не представляется возможным, то иначе как “Глава\_№” их назвать невозможно)

**Текст программы:**

from operator import itemgetter

class Book:

def \_\_init\_\_(self, id, name):

self.id = id

self.name = name

class Chapter:

def \_\_init\_\_(self, id, name, length, bk\_id):

self.id = id

self.name = name

self.length = length

self.bk\_id = bk\_id

class BookChap:

def \_\_init\_\_(self, bk\_id, ch\_id):

self.bk\_id = bk\_id

self.ch\_id = ch\_id

books = [

Book(1, "All Quiet on the Western Front"),

Book(2, "Luftwaffe aces"),

Book(3, "Achtung Panzer"),

]

chapters = [

Chapter(1, "Chapter 11", 43, 1),

Chapter(2, "Chapter 24", 45, 2),

Chapter(3, "Chapter 32", 61, 3),

Chapter(4, "Chapter 47", 38, 2),

Chapter(5, "Chapter 52", 47, 2),

Chapter(6, "Chapter 27", 42, 1)

]

book\_chap = [

BookChap(1, 1),

BookChap(2, 2),

BookChap(3, 3),

BookChap(2, 4),

BookChap(2, 5),

BookChap(1, 6),

]

def task1(ch\_list):

result = sorted(ch\_list, key=itemgetter(0))

return result

def task2(ch\_list):

result = []

temp = dict()

for i in ch\_list:

if i[2] in temp:

temp[i[2]] += 1

else:

temp[i[2]] = 1

for i in temp.keys():

result.append((i, temp[i]))

result.sort(key=itemgetter(1), reverse=True)

return result

def task3(ch\_list, end\_ch):

res\_3 = [(i[0], i[2]) for i in ch\_list if i[0].endswith(end\_ch)]

return res\_3

def main():

one\_to\_many = [(ch.name, ch.length, bk.name)

for bk in books

for ch in chapters

if ch.bk\_id == bk.id]

many\_to\_many\_temp = [(bk.name, bc.bk\_id, bc.ch\_id)

for bk in books

for bc in book\_chap

if bc.bk\_id == bk.id]

many\_to\_many = [(ch.name, ch.length, bk\_name)

for bk\_name, bk\_id, ch\_id in many\_to\_many\_temp

for ch in chapters if ch.id == ch\_id]

print("Задание Б1")

print(task1(one\_to\_many))

print("\nЗадание Б2")

print(task2(one\_to\_many))

print("\nЗадание Б3")

print(task3(many\_to\_many, '2'))

if \_\_name\_\_ == '\_\_main\_\_':

main()

**Результаты выполнения:**

Задание Б1

[('Chapter 11', 43, 'All Quiet on the Western Front'), ('Chapter 24', 45, 'Luftwaffe aces'), ('Chapter 27', 42, 'All Quiet on the Western Front'), ('Chapter 32', 61, 'Achtung Panzer'), ('Chapter 47', 38, 'Luftwaffe aces'), ('Chapter 52', 47, 'Luftwaffe aces')]

Задание Б2

[('Luftwaffe aces', 3), ('All Quiet on the Western Front', 2), ('Achtung Panzer', 1)]

Задание Б3

[('Chapter 52', 'Luftwaffe aces'), ('Chapter 32', 'Achtung Panzer')]