|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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)** |

**Факультет «Информатика и системы управления»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Курс «Разработка интернет-приложений»

Отчет по РК1

Выполнил:

студент группы ИУ5-53Б

Жуков Д.Д.

Москва, 2020 г.

Условия задания:

|  |  |  |
| --- | --- | --- |
| **№ варианта** | **Класс 1** | **Класс 2** |
| 9 | Глава | Книга |

Вариант В.

1. «Книга» и «Глава» связаны соотношением один-ко-многим. Выведите список всех глав в алфавитном порядке, начиная с буквы «А» и названия их книг.
2. «Книга» и «Глава» связаны соотношением один-ко-многим. Выведите список книг с минимальным объемом глав в каждой книге, отсортированный по минимальному объему.
3. «Книга» и «Глава» связаны соотношением многие-ко-многим.

Выведите список всех связанных глав и книг, отсортированный по главам, сортировка по книгам – произвольная.

Код программы

*# используется для сортировки*

**from** operator **import** itemgetter

*# ================= Класс главы книги =================*

**class** Chapter: """Глава"""

**def** init (self, id, title, vall, book\_id): self.id = id

self.title = title self.vall = vall self.book\_id = book\_id

*# ======================================================*

*# ============= Класс книги ===============*

**class** Book:

"""Книга"""

**def** init (self, id, title): self.id = id

self.title = title

*# =========================================*

*# ============== Класс связи ==============*

**class** ChapterBook: """

'Главы книги' для реализации связи многие-ко-многим

"""

**def** init (self, book\_id, chapter\_id): self.book\_id = book\_id self.chapter\_id = chapter\_id

*# =========================================*

*# ================= Книги ====================*

books = [

Book(1, 'Война и мир'), Book(2, 'Мастер и Маргарита'), Book(3, 'Муму'),

Book(4, 'Три мушкетера'),

Book(5, 'Энциклопедия комнатных растений'),

]

*#==============================================*

*# ================== Главы =================================*

chapters = [

Chapter(1, 'АГлава 1', 13, 1), *# глава с началом на "А"*

Chapter(2, 'Глава 2', 35, 1),

Chapter(3, 'Глава 3', 16, 2),

Chapter(4, 'АГлава 4', 21, 2), *# глава с началом на "А"*

Chapter(5, 'Глава 5', 20, 4),

]

*# ==========================================================*

*# ========== связь глав с книгами ========*

chapters\_books = [ ChapterBook(1,1), ChapterBook(1,2), ChapterBook(2,5), ChapterBook(3,2), ChapterBook(3,3),

ChapterBook(4,1), ChapterBook(4,2), ChapterBook(4,4), ChapterBook(5,1), ChapterBook(5,5),

]

*# =============================================*

*# ================== главная функция =================*

**def** main():

"""Основная функция"""

*# Соединение данных один-ко-многим*

one\_to\_many = [(c.title, c.vall, b.title)

**for** b **in** books **for** c **in** chapters **if** c.id == b.id]

*# Соединение данных многие-ко-многим*

many\_to\_many\_temp = [(b.title, cb.book\_id, cb.chapter\_id)

**for** b **in** books

**for** cb **in** chapters\_books

**if** b.id==cb.book\_id]

many\_to\_many = [(c.title, c.vall, book\_name)

**for** book\_name, book\_id, chapter\_id **in** many\_to\_many\_temp

**for** c **in** chapters **if** c.id==chapter\_id]

print('Задание В1:')

res\_11 = list(filter(**lambda** x: x[0].startswith('А'),one\_to\_many)) print(res\_11, '\n')

print('Задание В2:') res\_12\_unsorted = [] **for** b **in** books:

b\_chapter = list(filter(**lambda** x: x[2] == b.title, one\_to\_many))

**if** len(b\_chapter) > 0:

c\_vall = [vall **for** \_,vall,\_ **in** b\_chapter] c\_vall\_min = min(c\_vall)

res\_12\_unsorted.append((b.title, c\_vall\_min))

res\_12 = sorted(res\_12\_unsorted, key=itemgetter(1), reverse=**False**) print(res\_12, end='\n')

print('\nЗадание B3')

res\_13 = sorted(many\_to\_many, key=itemgetter(0)) print(res\_13)

*# =========================================================*

**if** name == ' main ': main()

# Что выводит:

