**Министерство образования и науки Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего профессионального образования**

**«Московский государственный технический университет имени Н.Э. Баумана»**

**(МГТУ им. Н.Э. Баумана)**

|  |  |
| --- | --- |
| **ФАКУЛЬТЕТ** | **«Информатики и систем управления»** |
| **КАФЕДРА** | Системы обработки информации и управления |

Дисциплина «Разработка Интернет-Приложений»

**РУБЕЖНЫЙ КОНТРОЛЬ №1**

Вариант 24

|  |  |
| --- | --- |
| Студент | Сахарова Е. К. ИУ5-52Б |
| Преподаватель | Гапанюк Ю. Е. |

**1. Задание**

|  |  |  |
| --- | --- | --- |
| **№ варианта** | **Класс 1** | **Класс 2** |
| 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 книгам 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 авторов, имена которых заканчивается на «ов», и главы, которые встречаются в их книгах.

**2. Предметная область**

Класс данных для предметной области Глава-Книга:

1. Класс «Глава», содержащий поля:
   * ID записи о главах;
   * Название главы;
   * ID записи о книге (для реализации связи один-ко-многим).
2. Класс «Книга», содержащий поля:
   * ID записи о книгах;
   * Название книги;
   * Фамилия автора книги;
   * Количество страниц книги.
3. (Для реализации связи многие-ко-многим)

Класс «Главы книги», содержащий поля:

* + ID записи о главе;
  + ID записи о книге.

**3. Текст программы**

**from** operator **import** itemgetter  
  
  
**class** Chapter:  
 *"""Глава"""* **def** \_\_init\_\_(self, id, name, book\_id):  
 self.id = id  
 self.name = name  
 self.book\_id = book\_id  
  
  
**class** Book:  
 *"""Книга"""* **def** \_\_init\_\_(self, id, name, author, pages):  
 self.id = id  
 self.name = name  
 self.author = author  
 self.pages = pages  
  
  
**class** ChapterBook:  
 *"""Главы книги"""* **def** \_\_init\_\_(self, book\_id, chapter\_id):  
 self.book\_id = book\_id  
 self.chapter\_id = chapter\_id  
  
  
books = [  
 Book(1, **'Преступление и наказание'**, **'Достоевский'**, 672),  
 Book(2, **'Война и мир'**, **'Толстой'**, 1225),  
 Book(3, **'Доктор Живаго'**, **'Пастернак'**, 592),  
 Book(5, **'Лезвие бритвы'**, **'Ефремов'**, 604),  
 Book(9, **'Мастер и Маргарита'**, **'Булгаков'**, 445),  
 Book(15, **'Два капитана'**, **'Каверин'**, 624),  
]  
  
chapters = [  
 Chapter(1, 0, 9),Chapter(2, 15, 2),Chapter(3, 1, 15),Chapter(4, 3, 1),Chapter(5, 99, 5),]  
  
chapters\_books = [  
 ChapterBook(1, 1),  
 ChapterBook(1, 2),  
 ChapterBook(1, 4),  
  
 ChapterBook(2, 1),  
 ChapterBook(2, 2),  
 ChapterBook(2, 4),  
  
 ChapterBook(3, 2),  
 ChapterBook(3, 4),  
  
 ChapterBook(5, 3),  
 ChapterBook(5, 5),  
  
 ChapterBook(9, 1),  
 ChapterBook(9, 3),  
  
 ChapterBook(15, 1),  
 ChapterBook(15, 3),  
]  
  
  
**def** main():  
  
 *# Соединение данных 1:М* one\_to\_many = [(c.name, b.name, b.author, b.pages)  
 **for** b **in** books  
 **for** c **in** chapters  
 **if** c.book\_id == b.id]  
  
 *# Соединение данных М:М* many\_to\_many\_temp = [(b.name, b.author, b.pages, cp.book\_id, cp.chapter\_id)  
 **for** b **in** books  
 **for** cp **in** chapters\_books  
 **if** b.id == cp.book\_id]  
  
 many\_to\_many = [(c.name, book\_name, book\_author, book\_pages)  
 **for** book\_name, book\_author, book\_pages, book\_id, chapter\_id **in** many\_to\_many\_temp  
 **for** c **in** chapters  
 **if** c.id == chapter\_id]  
  
 print(**'Задание Б1'**)  
 res\_1 = sorted(one\_to\_many, key=itemgetter(0))  
 print(res\_1)  
  
 print(**'\nЗадание Б2'**)  
 res\_2\_unsorted = []  
 *# Перебираем все книги* **for** b **in** books:  
 *# Список глав книги* b\_chapters = list(filter(**lambda** i: i[1] == b.name, one\_to\_many))  
 res\_2\_unsorted.append((b.name, len(b\_chapters)))  
  
 *# Сортировка по главам* res\_2 = sorted(res\_2\_unsorted, key=itemgetter(1), reverse=**True**)  
 print(res\_2)  
  
 print(**'\nЗадание Б3'**)  
 res\_3 = {}  
 *# Перебираем все книги* **for** b **in** books:  
 **if 'ов' in** b.author:  
 *# Список глав книги* b\_chapters = list(filter(**lambda** i: i[2] == b.author, many\_to\_many))  
 *# Только названия книг* b\_chapters\_names = [x **for** x, \_, \_, \_ **in** b\_chapters]  
 *# Добавляем результат в словарь  
 # ключ - фамилия автора, значение - список глав* res\_3[b.author] = b\_chapters\_names  
 print(res\_3)  
  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 main()

**4. Результат выполнения программы**

Задание Б1

[(0, 'Мастер и Маргарита', 'Булгаков', 445), (1, 'Два капитана', 'Каверин', 624), (3, 'Преступление и наказание', 'Достоевский', 672), (15, 'Война и мир', 'Толстой', 1225), (99, 'Лезвие бритвы', 'Ефремов', 604)]

Задание Б2

[('Преступление и наказание', 1), ('Война и мир', 1), ('Лезвие бритвы', 1), ('Мастер и Маргарита', 1), ('Два капитана', 1), ('Доктор Живаго', 0)]

Задание Б3

{'Ефремов': [1, 99], 'Булгаков': [0, 1]}