

Рубежный контроль №1 по курсу БКИТ
Вариант Д13

Классы: «Книга» – «Библиотека»

Запросы:

1. «Библиотека» и «Книга» связаны соотношением один-ко-многим. Выведите список всех Книг, у которых название начинается на «Ме», и названия их Библиотек.
2. «Библиотека» и «Книга» связаны соотношением один-ко-многим. Выведите список Библиотек со средней ценой Книг в каждой Библиотеке, отсортированный по средней цене.
3. «Библиотека» и «Книга» связаны соотношением многие-ко-многим. Выведите список всех Библиотек, у которых название начинается с буквы «К», и список имеющихся в них Книг.

Текст программы:

```
"""Предметная область: Книга-Библиотека, вариант 13Д"""  
from operator import itemgetter
```

```
class Book:  
    """Книга"""  
    def __init__(self, id, title, price, lib_id):  
        self.id = id  
        self.title = title  
        self.price = price  
        self.lib_id = lib_id
```

```
class Lib:  
    """Библиотека"""  
    def __init__(self, id, name):  
        self.id = id  
        self.name = name
```

```
class BookLibRelation:
```

```

        """Книги в библиотеке"""
def __init__(self, book_id, lib_id):
    self.book_id = book_id
    self.lib_id = lib_id

# Библиотеки
libs = [
    Lib(1, 'Книга и чернила'),
    Lib(2, 'Книжное место'),
    Lib(3, 'Перо'),

]

# Книги
books = [
    Book(1, 'Метель', 500, 1),
    Book(2, 'На реке', 200, 1),
    Book(3, 'Муму', 300, 2),
    Book(4, 'Мертвые души', 450, 2),
    Book(5, 'Цыган', 150, 2),
    Book(6, 'Обломов', 370, 3),
]

# Связи
books_libs = [
    BookLibRelation(1, 1),
    BookLibRelation(2, 1),
    BookLibRelation(3, 2),
    BookLibRelation(4, 2),
    BookLibRelation(5, 2),
    BookLibRelation(6, 3),
]

def main():
    """Основная функция"""

    # Соединение данных один ко многим

```

```

one_to_many = [(b.title, b.price, l.name)
                for l in libs
                for b in books
                if b.lib_id == l.id
                ]

# Соединение данных многие ко многим
many_to_many_temp = [(l.name, bl.lib_id, bl.book_id)
                      for l in libs
                      for bl in books_libs
                      if l.id == bl.lib_id
                      ]

many_to_many = [(b.title, b.price, lib_name)
                 for lib_name, lib_id, book_id in many_to_many_temp
                 for b in books
                 if b.id == book_id
                 ]

# Вывод книг, начинающихся на "Me" а так же их библиотеки

print("Задание Д1")
res_1 = [(b[0], b[2]) for b in list(filter(lambda x: x[0][:2] == "Me",
one_to_many))]
print(*res_1, sep="\n", end="\n\n")

# Вывод названий библиотек и средних цен на книги
print("Задание Д2")
res2 = []
for l in libs:
    # список книг в одной конкретной библиотеке
    books_in_lib = list(filter(lambda x: x[2] == l.name, one_to_many))
    if len(books_in_lib) > 0:
        sum_price = sum(int(b[1])
                        for b in books_in_lib
                        )
        avg = sum_price/len(books_in_lib)
        # добавляем название библиотеки и среднюю цену
        res2.append((books_in_lib[0][2], avg))
res2.sort(key=itemgetter(1))
print(*res2, sep="\n", end="\n\n")

```

```
# Вывод библиотек начинающихся на "К", а так же список книг в них
print("Задание Д3")
res3 = {}
for l in libs:
    if l.name[0] == "К":
        books_in_lib = list(filter(lambda x: x[2] == l.name, many_to_many))
        res3[l.name] = [b[0] for b in books_in_lib]
print(res3)
```

```
if __name__ == '__main__':
    main()
```

Результаты выполнения:

Задание Д1

('Метель', 'Книга и чернила')

('Мертвые души', 'Книжное место')

ы

Задание Д2

('Книжное место', 300.0)

('Книга и чернила', 350.0)

('Перо', 370.0)

Задание Д3

{'Книга и чернила': ['Метель', 'На реке'], 'Книжное место': ['Муму', 'Мертвые души', 'Цыган']}