

Рубежного контроль №1 по курсу БКИТ. Вариант Д25.

1. Класс «Раздел», содержащий поля:
 - ID записи о разделе;
 - Оглавление раздела;
 - Количество страниц (количественный признак);
 - ID записи о документе. (для реализации связи один-ко-многим)
2. Класс «Документ», содержащий поля:
 - ID записи о документе;
 - Наименование документа.
3. (Для реализации связи многие-ко-многим) Класс «Раздел документа», содержащий поля:
 - ID записи о разделе;
 - ID записи о документе.

1. «Документ» и «Раздел» связаны соотношением один-ко-многим. Выведите список оглавления разделов, которые заканчиваются на «ов», и названия их документа.
2. «Документ» и «Раздел» связаны соотношением один-ко-многим. Выведите список документов по среднему количеству страниц раздела в каждом документе, отсортированный по среднему количеству страниц.
3. «Документ» и «Раздел» связаны соотношением многие-ко-многим. Выведите список всех документов, у которых название начинается с буквы «А», и список разделов в нем.

Код программы.

```
from operator import itemgetter
class Sec:
    """Раздел"""
    def __init__(self, id, contents, pages, doc_id):
        self.id = id
        self.cont = contents
        self.pages = pages
        self.doc_id = doc_id
class Doc:
    """Документ"""
    def __init__(self, id, name):
        self.id = id
        self.name = name
class SecDoc:
    """Класс ^Раздел документа^ (Для реализации связи многие-ко-многим) """
    def __init__(self, sec_id, doc_id):
        self.doc_id = doc_id
        self.sec_id = sec_id
```

```

# Документы
docs = [
    Doc(1, 'Доход'),
    Doc(2, 'Аппартаменты'),
    Doc(3, 'Города'),

    Doc(11, '(другой) Доход'),
    Doc(22, '(другие) Аппартаменты'),
    Doc(33, '(другие) Города')
]

# Разделы
secs = [
    Sec(1, 'Деньги', 1000, 1),
    Sec(2, 'Здания', 1200, 2),
    Sec(3, 'Азов', 500, 3),
    Sec(4, 'Псков', 450, 3),
    Sec(5, 'Проценты', 600, 1),
    Sec(6, 'Сады', 550, 2)
]

secs_docs = [
    SecDoc(1, 1),
    SecDoc(5, 1),
    SecDoc(2, 2),
    SecDoc(6, 2),
    SecDoc(3, 3),
    SecDoc(4, 3),

    SecDoc(1, 11),
    SecDoc(5, 11),
    SecDoc(2, 22),
    SecDoc(6, 22),
    SecDoc(3, 33),
    SecDoc(4, 33)
]

def main():
    """Основная функция"""
    # Соединения данных один ко многим
    one_to_many = [(s.cont, s.pages, d.name)
                    for d in docs
                    for s in secs
                    if s.doc_id == d.id]

    # Соединение данных многие ко многим
    many_to_many_temp = [(d.name, sd.doc_id, sd.sec_id)
                           for d in docs
                           for sd in secs_docs
                           if d.id == sd.doc_id]
    many_to_many = [(s.cont, s.pages, doc_name)
                     for doc_name, doc_id, sec_id in many_to_many_temp
                     for s in secs
                     if s.id == sec_id]

    print('Задание Д1')
    for s in secs:
        if s.cont[len(s.cont) - 1] == 'в' and s.cont[len(s.cont) - 2] == 'о':
            print(s.cont, docs[(s.doc_id) - 1].name)

    print('\nЗадание Д2')
    res_12_unsorted = []
    # Перебираем все документы
    for d in docs:

```

```

# Список разделов в документе
d_secs = list(filter(lambda i: i[2] == d.name, one_to_many))
# Если разделы не пусты
if len(d_secs) > 0:
    # Количество страниц в разделах
    d_pages = [pages for _, pages, _ in d_secs]
    # Суммарное количество страниц
    d_pages_sum = sum(d_pages)
    # Среднее количество страниц
    d_pages_avg = d_pages_sum / len(d_pages)
    res_12_unsorted.append((d.name, d_pages_avg))
# Сортировка по среднему количеству страниц
res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
print(res_12)

print('\n Задание Д3')
res_13 = {}
# Перебираем все документы
for d in docs:
    if 'A' in d.name:
        # Список разделов в документе
        d_secs = list(filter(lambda i: i[2] == d.name, many_to_many))
        # Только оглавления разделов
        d_secs_conts = [x for x, _, _ in d_secs]
        # Добавляем результат в словарь
        # ключ - документ, значение - список оглавлений
        res_13[d.name] = d_secs_conts
print(res_13)

if __name__ == "__main__":
    main()

```

Результат выполнения программы.

Задание Д1

Азов Города

Псков Города

Задание Д2

[('Апартаменты', 875.0), ('Доход', 800.0), ('Города', 475.0)]

Задание Д3

{'Апартаменты': ['Здания', 'Сады'], '(другие) Апартаменты': ['Здания', 'Сады']}