Отчет по РК 1 «Базовые компоненты интернет технологий»

Целуйко Ульяна ИУ5-31Б, Вариант 25 А

Задание:

1. «Документ» и «Раздел» связаны соотношением один-ко-многим. Выведите список всех связанных разделов и документов, отсортированный по документам, сортировка по разделам произвольная.
2. «Документ» и «Раздел» связаны соотношением один-ко-многим. Выведите список документов с суммарным количеством символов в каждом документе, отсортированный по количеству символов.
3. «Документ» и «Раздел» связаны соотношением многие-ко-многим. Выведите список всех документов, у которых в названии присутствует слово «Книга», и список разделов в нем.

Листинг программы:

```
from operator import itemgetter


class Sec:
    def __init__(self, Id, Names, Symbols, Id_Sec):
        self.id =Id
        self.names =Names
        self.symb=Symbols
        self.id_sec=Id_Sec


class Doc:
    def __init__(self, Id, Named):
        self.id=Id
        self.named=Named


class SecDoc:
```

```
    def __init__(self, Id_Doc, Id_Sec):
        self.id_doc=Id_Doc
        self.id_sec=Id_Sec

# Разделы
secs = [
    Sec(1, 'От автора', 2300,1 ),
    Sec(2, 'Введение', 3400,2 ),
    Sec(3, 'О зачислении', 9000,3 ),
    Sec(4, 'Глава 1', 4000,3 ),
    Sec(5, 'Заключение', 2700,3 ),
]

# Документы
docs = [
    Doc(1, 'Книга'),
    Doc(2, 'Курсовая'),
    Doc(3, 'Приказ'),
    Doc(4, 'Заметки'),
    Doc(5, 'Книга1'),
    Doc(6, 'Сборник указов '),
]

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

```
    SecDoc(5,2),
    SecDoc(6,3),
    SecDoc(6,4),
    SecDoc(6,5),
]

def main():
    one_to_many = [(s.id_doc, s.id_sec, d.named)
        for d in docs
        for s in secs_docs
        if s.id_doc==d.id]

    many_to_many_temp = [(d.named, sd.id_doc, sd.id_sec)
        for d in docs
        for sd in secs_docs
        if d.id == sd.id_doc]

    many_to_many = [(s.names, s.symb, doc_name)
        for doc_name, Id_Doc, Id_Sec in many_to_many_temp
        for s in secs if s.id == Id_Sec]



    print('Задание А1')
    res_11 = sorted(one_to_many, key=itemgetter(2))
    print(res_11)

    print('\nЗадание А2')
    res_12_unsorted = []
```

```
    for d in docs:

        d_secs = list(filter(lambda i: i[2]==d.named, one_to_many))
        if len(d_secs) > 0:
            d_symbs = [ Symbols for _,Symbols,_ in d_secs]

            d_symbs_sum = sum(d_symbs)
            res_12_unsorted.append((d.named, d_symbs_sum))


    res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
    print(res_12)

    print('\nЗадание А3')
    res_13 = {}

    for d in docs:
        if 'Книга' in d.named:

            d_secs = list(filter(lambda i: i[2]==d.named, many_to_many))

            d_secs_names = [x for x,_,_ in d_secs]

            res_13[d.named] = d_secs_names

    print(res_13)

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

Результат выполнения:

```
Задание A1
[(4, 1, 'Заметки'), (1, 1, 'Книга'), (5, 2, 'Книга1'), (2, 2, 'Курсовая'), (3, 3, 'Приказ'), (3, 4, 'Приказ'
    ), (3, 5, 'Приказ'), (6, 3, 'Сборник указов '), (6, 4, 'Сборник указов '), (6, 5, 'Сборник указов ')]
Задание A2
[('Приказ', 12), ('Сборник указов ', 12), ('Курсовая', 2), ('Книга1', 2), ('Книга', 1), ('Заметки', 1)]

Задание A3
{'Книга': ['От автора'], 'Книга1': ['Введение']}
>
```