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 «ГУИМЦ»

Кафедра ИУ5 «Системы обработки информации и управления»

Дисциплина «Базовые компоненты ИТ»

ОТЧЕТ

Рубежный контроль №1

Студент: Близнева А.Е., группа ИУ5Ц-51Б

Преподаватель: Гапанюк Ю.Е.

2021г.

Описание задания:

Вариант А, вариант предметной области №25.

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

Класс «Раздел», содержащий поля:
- id раздела (id)
- название раздела (name)

Класс «Документ», содержащий поля:
- id документа (id)
- название (name)
- кол-во документов (number)
- id раздела, для реализации связи один-ко-многим (sect_id)

Класс «Документы раздела» (для реализации связи один-ко-многим), содержащий поля:
- id раздела (sect_id)
- id документа (doc_id)

Листинг программы:

```
"""РК№1, Близнева Ангелина, группа ИУ5Ц-51Б
Вариант А, вариант предметной области 25"""

# используется для сортировки
from operator import itemgetter


class Doc:
    """Документ"""

    def __init__(self, id, name, number, sect_id):
        self.id = id
        self.name = name
        self.number = number
        self.doc_id = sect_id


class Sect:
    """Раздел"""

    def __init__(self, id, name):
        self.id = id
        self.name = name


class SectDoc:
    """
    'Документы раздела' для реализации
    связи многие-ко-многим
    """

    def __init__(self, sect_id, doc_id):
        self.sect_id = sect_id
        self.doc_id = doc_id


# Разделы
sects = [
    Sect(1, 'Приказы'),
    Sect(2, 'Распоряжения'),
    Sect(3, 'Протоколы'),

    Sect(11, 'Приказы (филиал компании)'),
    Sect(22, 'Распоряжения (филиал компании)'),
    Sect(33, 'Протоколы (филиал компании)'),
]

# Документы
docs = [
    Doc(1, 'Приказ о ежегодном отпуске', 333, 1),
    Doc(2, 'Распоряжение для выполнения задания', 1231, 2),
    Doc(3, 'Протокол совещания', 56, 3),
    Doc(4, 'Протокол общего собрания учредителей', 34, 3),
    Doc(5, 'Протокол заседания экспертной комиссии', 25, 3),
]

sects_docs = [
    SectDoc(1, 1),
    SectDoc(2, 2),
    SectDoc(3, 3),
```

```
    SectDoc(3, 4),
    SectDoc(3, 5),

    SectDoc(11, 1),
    SectDoc(22, 2),
    SectDoc(33, 3),
    SectDoc(33, 4),
]


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

    # Соединение данных один-ко-многим
    one_to_many = [(d.name, d.number, s.name)
                   for s in sects
                   for d in docs
                   if d.doc_id == s.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(s.name, sd.sect_id, sd.doc_id)
                         for s in sects
                         for sd in sects_docs
                         if s.id == sd.sect_id]

    many_to_many = [(d.name, d.number, sect_name)
                    for sect_name, sect_id, emp_id in many_to_many_temp
                    for d in docs if d.id == emp_id]

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

    print('\nЗадание А2')
    res_12_unsorted = []
    # Перебираем все разделы
    for s in sects:
        # Список документов раздела
        d_docs = list(filter(lambda i: i[2] == s.name, one_to_many))
        # Если раздел не пустой
        if len(d_docs) > 0:
            # Кол-во документов раздела
            d_nums = [sal for _, sal, _ in d_docs]
            # Суммарное кол-во документов раздела
            d_nums_sum = sum(d_nums)
            res_12_unsorted.append((s.name, d_nums_sum))

    # Сортировка по суммарному кол-ву документов
    res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
    print(res_12)

    print('\nЗадание А3')
    res_13 = {}
    # Перебираем все разделы
    for s in sects:
        if 'Протокол' in s.name:
            # Список документов раздела
            d_docs = list(filter(lambda i: i[2] == s.name, many_to_many))
            # Только наименование документа
            d_docs_names = [x for x, _, _ in d_docs]
            # Добавляем результат в словарь
            # ключ - раздел, значение - список документов
            res_13[s.name] = d_docs_names
```

```
    print(res_13)


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

Результат выполнения программы:

```
Задание A1
[('Приказ о ежегодном отпуске', 333, 'Приказы'), ('Протокол совещания', 56, 'Протоколы'), ('Протокол общего собрания учредителей', 34, 'Протоколы'),
('Протокол заседания экспертной комиссии', 25, 'Протоколы'), ('Распоряжение для выполнения задания', 1231, 'Распоряжения')]

Задание A2
[('Распоряжения', 1231), ('Приказы', 333), ('Протоколы', 115)]

Задание A3
{'Протоколы': ['Протокол совещания', 'Протокол общего собрания учредителей', 'Протокол заседания экспертной комиссии'], 'Протоколы (филиал компании)'
: ['Протокол совещания', 'Протокол общего собрания учредителей']}
```