Карпов И.Е., ИУ5Ц-74Б

Условие:

Вариант Г. 24

27	Преподаватель	Учебный курс

1. «Книга» и «Глава» связаны соотношением один-ко-многим. Выведите список всех Книг, у которых название начинается с буквы «А» и список глав.

2. «Книга» и «Глава» связаны соотношением один-ко-многим. Выведите список Книг с максимальным количество страниц одной главы в каждой книге, отсортированной по максимальному количеству.

3. «Книга» и «Глава» связаны соотношением многие-ко-многим. Выведите список всех связанных глав и книг, отсортированный по книгам, сортировка по главам произвольная.

```
#!/usr/bin/env python

from operator import itemgetter

class emps:
    """Глава"""
    def __init__(self, id, name, stra, empa_id):
        self.id = id
        self.name = name
        self.stra = stra
        self.empa_id = empa_id

class Dep:
    """Книга"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class EmpDep:
    """'Главы книги' для реализации связи многие-ко-многим"""
    def __init__(self, emps_id, empa_id):
        self.emps_id = emps_id
        self.empa_id = empa_id


if __name__ == "__main__":
    Dep = [
    Dep(0, 'Пхеньян'),
    Dep(1, 'Программирование с нуля'),
    Dep(2, 'Америка'),
    Dep(3, 'Автоматизация'),
```

```
Dep(4, 'C#'),
Dep(5, 'Навальный'),
]



emps = [
emps(0, 'Начало', 2, 0),
emps(1, 'Начало', 2, 1),
emps(2, 'Начало', 2, 2),
emps(3, 'Начало', 2, 3),
emps(4, 'Начало', 2, 4),
emps(5, 'Начало', 2, 5),
emps(6, 'История', 11, 2),
emps(7, 'История', 9, 3),
emps(8, 'История', 13, 2),
emps(9, 'Местоположение', 18, 2),
emps(10, 'Заключение', 50, 5),
emps(11, 'Ведение', 1, 1),
]

emps_empa = [
EmpDep(1,1),
EmpDep(2,2),
EmpDep(3,3),
EmpDep(3,4),
EmpDep(3,5),

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


 # Соединение данных один-ко-многим
one_to_many = [(d.name, d.stra, s.name)
    for s in Dep
    for d in emps
    if d.empa_id==s.id]

# Соединение данных многие-ко-многим
many_to_many_temp = [(s.name, ds.empa_id, ds.emps_id)
    for s in Dep
    for ds in emps_empa
    if s.id==ds.empa_id]
many_to_many = [(d.name, d.stra, empa_name)
    for empa_name, empa_id, emps_id in many_to_many_temp
    for d in emps if d.id==emps_id]
```

```
    print("Задание Г1")
    res_11 = {}
    selected_empa = [one_book[2] for one_book in one_to_many if
one_book[2].startswith('a') or one_book[2].startswith('A')]
    for empa_name in selected_empa:
        emps_for_book = [(one_emps[0],one_emps[1]) for one_emps in one_to_many if
one_emps[2]==empa_name]
        res_11.update({empa_name:emps_for_book})
    print(res_11)
    print()

    print("Задание Г2")
    res_12_unsorted = []
    for s in Dep:
        s_emps = list(filter(lambda i: i[2]==s.name, one_to_many))
        if len(s_emps) > 0:
            s_stras = [stra for _,stra,_ in s_emps]
            s_stra_max = max(s_stras)
            res_12_unsorted.append((s.name, s_stra_max))

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

    print("Задание Г3")
    res_13 = {}
    Dep.sort(key=lambda one_empa: one_empa.name)
    for s in Dep:
        s_emps = list(filter(lambda i: i[2]==s.name, many_to_many))
        s_emps_names = [x for x,_,_ in s_emps]
        res_13[s.name] = s_emps_names
    print(res_13)
```

```
Задание Г1
{'Америка': [('Начало', 2), ('История', 11), ('История', 13), ('Местоположение', 18)], 'Автоматизация': [('Начало', 2), ('История', 9)]}

Задание Г2
[('Навальный', 50), ('Америка', 18), ('Автоматизация', 9), ('Пхеньян', 2), ('Программирование с нуля', 2), ('C#', 2)]

Задание Г3
{'C#': ['Начало'], 'Автоматизация': ['Начало'], 'Америка': ['Начало'], 'Навальный': ['Начало'], 'Программирование с нуля': ['Начало', 'Ведение'], 'Пхеньян': []}
```