

Рубежный контроль №1

Текст программы:

```
class Language:
    ''' язык программирования '''
    def __init__(self, id, name):
        self.id = id
        self.name = name

class Library:
    ''' библиотека языка программирования '''
    def __init__(self, id, name, func_num, lang_id):
        ''' func_num - количество функций в библиотеке'''
        self.id = id
        self.name = name
        self.func_num = func_num
        self.lang_id = lang_id

class Link:
    ''' связь библиотеки с языком '''
    def __init__(self, lib_id, lang_id):
        self.lib_id = lib_id
        self.lang_id = lang_id

langs = [
    Language(1, "Python"),
    Language(2, "C++"),
    Language(3, "Pascal"),

    Language(10, "Python v0"),
    Language(20, "C++ v0"),
    Language(30, "Pascal v0"),
]

libs = [
    Library(11, "random", 30, 1),
    Library(12, "math", 50, 1),
    Library(21, "vector", 40, 2),
    Library(22, "algorithm", 20, 2),
    Library(31, "Graph", 10, 3)
]

links = [
    Link(11, 1),
    Link(12, 1),
    Link(21, 2),
    Link(22, 2),
    Link(31, 3),

    Link(11, 10),
    Link(12, 10),
    Link(21, 20),
    Link(22, 20),
    Link(31, 30)
]

def main():
    ''' СВЯЗЬ ОДИН-КО-МНОГИМ'''
    one_to_many = [(lib.name, lib.func_num, lang.name)
                    for lib in libs
                    for lang in langs
                    if lib.lang_id == lang.id]

    many_to_many_temp = [(lang.name, link.lang_id, link.lib_id)
```

```

        for lang in langs
        for link in links
        if lang.id == link.lang_id]

''' СВЯЗЬ МНОГО-КО-МНОГИМ'''
many_to_many = [(lib.name, lib.func_num, lang_name)
                 for lang_name, lang_id, lib_id in many_to_many_temp
                 for lib in libs if lib.id == lib_id]

print('\nЗадание 1')
ans_1 = {}
for lib_name, x, lang_name in one_to_many:
    ''' если название языка начинается с "P" '''
    if lang_name[0] == 'P':
        if lang_name in ans_1:
            ans_1[lang_name].append(lib_name)
        else:
            ans_1[lang_name] = [lib_name]
print(*ans_1.items())

print('\nЗадание 2')
ans_2 = {}
for x, func_num, lang_name in one_to_many:
    if lang_name in ans_2:
        ans_2[lang_name] = max(ans_2[lang_name], func_num)
    else:
        ans_2[lang_name] = func_num
ans_2 = {key : value for key, value in sorted(ans_2.items(), key = lambda item:
item[1])}
print(*ans_2.items())

print('\nЗадание 3')
ans_3 = []
for lib_name, x, lang_name in many_to_many:
    ans_3.append((lang_name, lib_name))
ans_3 = sorted(ans_3, key = lambda item: item[0])
print(*ans_3)

if __name__ == '__main__':
    main()

```

Результаты выполнения программы:

Задание 1

('Python', ['random', 'math']) ('Pascal', ['Graph'])

Задание 2

('Pascal', 10) ('C++', 40) ('Python', 50)

Задание 3

('C++', 'vector') ('C++', 'algorithm') ('C++ v0', 'vector') ('C++ v0', 'algorithm') ('Pascal', 'Graph') ('Pascal v0', 'Graph') ('Python', 'random') ('Python', 'math') ('Python v0', 'random') ('Python v0', 'math')