

Рубежный контроль № 1.

Полученное задание:

1) Необходимо создать два класса данных в соответствии с Вашим вариантом предметной области, которые связаны отношениями один-ко-многим и многие-ко-многим.

2) Необходимо создать списки объектов классов, содержащих тестовые данные (3-5 записей), таким образом, чтобы первичные и вторичные ключи соответствующих записей были связаны по идентификаторам.

3) Необходимо разработать запросы в соответствии с Вашим вариантом. Запросы сформулированы в терминах классов «Сотрудник» и «Отдел», которые используются в примере. Вам нужно перенести эти требования в Ваш вариант предметной области. При разработке запросов необходимо по возможности использовать функциональные возможности языка Python (list/dict comprehensions, функции высших порядков).

Вариант Б.

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по сотрудникам, сортировка по отделам произвольная.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов с количеством сотрудников в каждом отделе, отсортированный по количеству сотрудников.
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех сотрудников, у которых фамилия заканчивается на «ов», и названия их отделов.

15	Файл	Каталог файлов
----	------	----------------

Текст программы

```
from operator import itemgetter
```

class File:

```
def __init__(self, file_id, name, type, size, catalog_id):
```

```
    self.id = file_id
```

```
    self.name = name
```

```
    self.type = type
```

```
    self.size = size
```

```
    self.catalog_id = catalog_id
```

class Catalog:

```
def __init__(self, catalog_id, name):
```

```
    self.id = catalog_id
```

```
    self.name = name
```

class CatalogFile:

```
def __init__(self, file_id, catalog_id):
```

```
    self.file_id = file_id
```

```
    self.catalog_id = catalog_id
```

files = [

```
    File(1, "some_text", "txt", 5, 1),
```

```
    File(2, "more_text", "txt", 6, 1),
```

```
    File(3, "first", "py", 3, 2),
```

```
    File(4, "second", "py", 3, 2),
```

```
    File(5, "third", "py", 5, 2)
```

```
]
```

```
catalogs = [  
    Catalog(1, "Texts"),  
    Catalog(2, "PythonCode")  
]
```

```
catalog_files = [  
    CatalogFile(1, 2),  
    CatalogFile(2, 1),  
    CatalogFile(3, 2),  
    CatalogFile(4, 2),  
    CatalogFile(5, 2),  
    CatalogFile(1, 1)  
]
```

```
def first_task(lst):  
    return sorted(lst, key=lambda x: x[0])
```

```
def second_task(lst):  
    d = {}  
    for file, file_id, catalog in lst:  
        if catalog in d:  
            d[catalog] += 1  
        else:  
            d[catalog] = 1
```

```
return sorted(d.items(), key=lambda x: x[1], reverse=True)
```

```
def third_task(lst, end):
```

```
    return [(file_name, catalog_name) for file_name, _, catalog_name in lst if
file_name.endswith(end)]
```

```
def main():
```

```
    one_to_many = [(file.name + "." + file.type, file.id, catalog.name)
```

```
                    for catalog in catalogs
```

```
                    for file in files
```

```
                    if file.catalog_id == catalog.id]
```

```
    many_to_many = [(file.name + '.' + file.type, catalog_file.file_id, catalog.name)
```

```
                    for file in files
```

```
                    for catalog in catalogs
```

```
                    for catalog_file in catalog_files
```

```
                    if file.id == catalog_file.file_id and catalog.id == catalog_file.catalog_id]
```

```
print("\nЗадание 1")
```

```
print(first_task(one_to_many))
```

```
print("\nЗадание 2")
```

```
print(second_task(one_to_many))
```

```
print("\nЗадание 3")
```

```
print(third_task(many_to_many, "txt"))
```

```
if __name__ == '__main__':
```

```
    main()
```

Вывод программы

Задание 1

```
[('first.py', 3, 'PythonCode'), ('more_text.txt', 2, 'Texts'), ('second.py', 4, 'PythonCode'), ('some_text.txt', 1, 'Texts'), ('third.py', 5, 'PythonCode')]
```

Задание 2

```
[('PythonCode', 3), ('Texts', 2)]
```

Задание 3

```
[('some_text.txt', 'Texts'), ('some_text.txt', 'PythonCode'), ('more_text.txt', 'Texts')]
```