

Отчет по РК1 по дисциплине

“Парадигмы и конструкции языков программирования”

Вариант запросов Г. Вариант предметной области 14 (CD-диск – Библиотека CD-дисков)

Запросы:

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех отделов, у которых название начинается с буквы «А», и список работающих в них сотрудников.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов с максимальной зарплатой сотрудников в каждом отделе, отсортированный по максимальной зарплате.
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по отделам, сортировка по сотрудникам произвольная.

Текст программы:

```
from operator import itemgetter
```

```
class CD_disk:
```

```
    def init(self, id, name, size, library_id):
```

```
        self.id = id
```

```
        self.name = name
```

```
        self.size = size
```

```
        self.library_id = library_id
```

```
class CD_library:
```

```
    def init(self, id, name):
```

```
        self.id = id
```

```
        self.name = name
```

```
class Disks_in_libraries:
    def init(self, library_id, disk_id):
        self.library_id = library_id
        self.disk_id = disk_id
```

```
disks = [
    CD_disk(1, "The Silence of the Lambs", 512, 1),
    CD_disk(2, "Men in Black", 1024, 2),
    CD_disk(3, "Home Alone", 256, 2),
    CD_disk(4, "Avatar", 2048, 3),
    CD_disk(5, "Interstellar", 4096, 3)
]
```

```
libraries = [
    CD_library(1, "Thriller"),
    CD_library(2, "Comedy"),
    CD_library(3, "Fiction"),
    CD_library(4, "Drama")
]
```

```
disks_in_libraries = [
    Disks_in_libraries(1, 1),
    Disks_in_libraries(2, 2),
    Disks_in_libraries(2, 3),
```

```

Disks_in_libraries(3, 4),
Disks_in_libraries(3, 5),

Disks_in_libraries(4, 4),
Disks_in_libraries(5, 4)
]

```

```
def main():
```

```

    one_to_many = [(d.name, d.size, l.name)
                    for l in libraries
                    for d in disks
                    if d.library_id == l.id]

```

```

    many_to_many_temp = [(l.name, dl.library_id, dl.disk_id)
                          for l in libraries
                          for dl in disks_in_libraries
                          if l.id==dl.library_id]

```

```

    many_to_many = [(d.name, d.size, lib_name)
                    for lib_name, library_id, disk_id in many_to_many_temp
                    for d in disks
                    if d.id==disk_id]

```

```
print("Задание Г1:")
```

```

res1 = [item
        for item in one_to_many
        if item[2].startswith('C')]

```

```

print(res1)

print("\nЗадание Г2:")
res2_unsorted = []
for l in libraries:
    lib_disks = list(filter(lambda i: i[2]==l.name, one_to_many))
    if len(lib_disks) > 0:
        d_sizes = [size for _,size,_ in lib_disks]
        l_max_sizes = max(d_sizes)
        res2_unsorted.append((l.name, l_max_sizes))
res2 = sorted(res2_unsorted, key=itemgetter(1), reverse=True)
print(res2)

```

```

print("\nЗадание Г3')
res3 = {}
for l in libraries:
    lib_disks = list(filter(lambda i: i[2] == l.name, many_to_many))
    l_disk_name = [name for name, _, _ in lib_disks]
    res3[l.name] = l_disk_name
print(res3)

```

```

if name == 'main':
    main()

```

Результат выполнения программы:

```

Задание Г1:
[('Men in Black', 1024, 'Comedy'), ('Home Alone', 256, 'Comedy')]

Задание Г2:
[('Fiction', 4096), ('Comedy', 1024), ('Thriller', 512)]

Задание Г3
{'Thriller': ['The Silence of the Lambs'], 'Comedy': ['Men in Black', 'Home Alone'], 'Fiction': ['Avatar', 'Interstellar'], 'Drama': ['Avatar']}

```