

МГТУ им. Н.Э. Баумана

Отчёт по рубежному контролю №1
по курсу «Базовые компоненты и интернет-технологии»
Вариант 14.

Руководитель
Гапанюк Ю.Е.
28.10.2022

Студент группы ИУ5-34Б
Нигматуллин А.Р.
28.10.2022

2022 г.

Полученное задание:

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

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

Необходимо разработать запросы в соответствии с Вашим вариантом.

Предметная область: класс_1 – CD-диск, класс_2 – Библиотека CD-дисков, вариант запросов: Г.

Запросы:

1. «Библиотека CD-дисков» и «CD-диск» связаны соотношением один-ко-многим. Выведите список всех библиотек CD-дисков, у которых название начинается с буквы «А», и список содержащихся в них CD-дисков.
2. «Библиотека CD-дисков» и «CD-диск» связаны соотношением один-ко-многим. Выведите список библиотек CD-дисков с максимальным размером CD-дисков в каждой библиотеке CD-дисков, отсортированный по максимальному размеру.
3. «Библиотека CD-дисков» и «CD-диск» связаны соотношением многие-ко-многим. Выведите список всех связанных CD-дисков и библиотек CD-дисков, отсортированный по библиотекам CD-дисков, сортировка по CD-дискам произвольная.

Текст программы:

```
# используется для сортировки
from operator import itemgetter

class Disk:
    """CD-диск"""

    def __init__(self, id, maker, memory, lib_id):
        self.id = id
        self.maker = maker
        self.memory = memory
        self.lib_id = lib_id

class Lib:
    """Библиотека CD-дисков"""

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

```

        self.name = name

class DiskLib:
    """
    'CD-диски библиотек' для реализации
    связи многие-ко-многим
    """

    def __init__(self, lib_id, disk_id):
        self.lib_id = lib_id
        self.disk_id = disk_id

# Библиотеки CD-дисков
libs = [
    Lib(1, 'архивы'),
    Lib(2, 'фильмы'),
    Lib(3, 'игры'),
    Lib(4, 'музыка'),
    Lib(5, 'утилиты'),
    Lib(6, 'прочее'),
]

# CD-диски
disks = [
    Disk(1, 'Verbatim', 700, 1),
    Disk(2, 'Ritek', 650, 2),
    Disk(3, 'Sonnen', 500, 3),
    Disk(4, 'Mirex', 250, 1),
    Disk(5, 'Vs', 350, 3),
]

disks_libs = [
    DiskLib(1, 1),
    DiskLib(2, 2),
    DiskLib(3, 3),
    DiskLib(3, 4),
    DiskLib(3, 5),
    DiskLib(4, 1),
    DiskLib(5, 2),
    DiskLib(6, 3),
    DiskLib(4, 4),
    DiskLib(3, 5),
]

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

    # Соединение данных один-ко-многим
    one_to_many = [(d.make, d.memory, l.name)
                    for l in libs
                    for d in disks
                    if d.lib_id == l.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(l.name, ld.lib_id, ld.disk_id)
                          for l in libs
                          for ld in disks_libs
                          if l.id == ld.lib_id]

    many_to_many = [(d.make, d.memory, lib_name)
                     for lib_name, lib_id, disk_id in many_to_many_temp

```

```

        for d in disks if d.id == disk_id]

print('Задание Г1')
res_11 = {}
for l in libs:
    if l.name[0] == 'a':
        l_disks = list(filter(lambda i: i[2] == l.name, one_to_many))
        l_disks_names = [x for x, _, _ in l_disks]
        res_11[l.name] = l_disks_names

print(res_11)

print('\nЗадание Г2')
res_12_unsorted = []
for l in libs:
    l_disks = list(filter(lambda i: i[2] == l.name, one_to_many))
    if len(l_disks) > 0:
        l_members = [mem for _, mem, _ in l_disks]
        l_members_max = max(l_members)
        res_12_unsorted.append((l.name, l_members_max))

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

print('\nЗадание Г3')
res_13 = sorted(many_to_many, key=itemgetter(2))
for i in res_13:
    print (i)

if __name__ == '__main__':
    main()

```

Результаты выполнения:

```

Задание Г1
{'архивы': ['Verbatim', 'Mirex']}

Задание Г2
[('архивы', 700), ('фильмы', 650), ('игры', 500)]

Задание Г3
('Verbatim', 700, 'архивы')
('Sonnen', 500, 'игры')
('Mirex', 250, 'игры')
('Vs', 350, 'игры')
('Vs', 350, 'игры')
('Verbatim', 700, 'музыка')
('Mirex', 250, 'музыка')
('Sonnen', 500, 'прочее')
('Ritek', 650, 'утилиты')
('Ritek', 650, 'фильмы')

```