

	<p align="center"> Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана) </p>
---	--

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ
 КАФЕДРА СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ

Отчет по рубежному контролю №1. Вариант №14(А)

Студент Мартынова Полина Владимировна
фамилия, имя, отчество

Группа ИУ5-51Б

Студент 18.10.2021 Мартынова П.В.
подпись, дата фамилия, и.о.

Преподаватель 18.10.2021 Гапанюк Ю.Е.
подпись, дата фамилия, и.о.

2021г.

Задание:

- 1) "CD-диск" и "Библиотека CD-дисков" связаны соотношением один-ко-многим. Выведите список всех связанных дисков и библиотек, отсортированный по библиотекам, сортировка по дискам произвольная.
- 2) "CD-диск" и "Библиотека CD-дисков" связаны соотношением один-ко-многим. Выведите список дисков с суммарной ёмкостью в библиотеке дисков, отсортированный по суммарному занимаемому месту.
- 3) "CD-диск" и "Библиотека CD-дисков" связаны соотношением многие-ко-многим. Выведите список всех библиотек, у которых в названии присутствует слово «Library», и список их дисков.

Код:

```
# используется для сортировки
from operator import itemgetter

class CD:
    """cd-диск"""
    def __init__(self, id, capacity, recording_layer_type, lib_id):
        self.id = id
        self.capacity = capacity
        self.recording_layer_type = recording_layer_type
        self.lib_id = lib_id

class CD_lib:
    """библиотека cd-дисков"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

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

# библиотеки cd-дисков
libs = [
    CD_lib(1, 'CDDDB'),
    CD_lib(2, 'Dacal CD Library'),
    CD_lib(3, 'Proceed')
]

# диски
disks = [
    CD(1, 650, 'Cyanine', 1),
    CD(2, 702, 'Cyanine', 1),
    CD(3, 790, 'Cyanine', 1),
    CD(4, 870, 'Cyanine', 2),
    CD(5, 650, 'Phthalocyanine', 2),
    CD(6, 650, 'Phthalocyanine', 2),
    CD(7, 702, 'Phthalocyanine', 3),
    CD(8, 702, 'Phthalocyanine', 3),
    CD(9, 650, 'Phthalocyanine', 4),
    CD(10, 650, 'Phthalocyanine', 4)
]

lib_disks = [
    LibDisk(1, 1),
```

```

LibDisk(1, 2),
LibDisk(1, 3),
LibDisk(2, 4),
LibDisk(2, 5),
LibDisk(2, 6),
LibDisk(3, 7),
LibDisk(3, 8),
LibDisk(4, 9),
LibDisk(4, 10),
]

def main():
    # Соединение данных один-ко-многим
    one_to_many = [(e.recording_layer_type, e.capacity, d.name)
                    for d in libs
                    for e in disks
                    if e.lib_id == d.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(d.name, ed.lib_id, ed.disk_id)
                          for d in libs
                          for ed in lib_disks
                          if d.id == ed.lib_id]

    many_to_many = [(e.recording_layer_type, e.capacity, lib_name)
                     for lib_name, lib_id, disk_id in many_to_many_temp
                     for e in disks if e.id == disk_id]

    print('Задание A1')
    res_11 = sorted(one_to_many, key=itemgetter(2))
    print(res_11)

    print('\nЗадание A2')
    res_12_unsorted = []
    # Перебираем все библиотеки
    for d in libs:
        # Список дисков в библиотеке
        d_disks = list(filter(lambda i: i[2] == d.name, one_to_many))
        # Если библиотека не пустая
        if len(d_disks) > 0:
            # Ёмкости дисков в библиотеке
            d_capacitys = [capacity for _, capacity, _ in d_disks]
            # суммарная ёмкость дисков в библиотеке
            d_capacitys_sum = sum(d_capacitys)
            res_12_unsorted.append((d.name, d_capacitys_sum))

    # Сортировка по суммарной ёмкости
    res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
    print(res_12)

    print('\nЗадание A3')
    res_13 = {}
    # Перебираем все библиотеки
    for d in libs:
        if 'Library' in d.name:
            # Список дисков в библиотеке
            d_disks = list(filter(lambda i: i[2] == d.name, many_to_many))
            # только тип записывающего слоя
            d_disks_id = [x for x, _, _ in d_disks]
            # Добавляем результат в словарь
            # ключ - библиотека, значение - список id
            res_13[d.name] = d_disks_id

    print(res_13)

```

```
if __name__ == '__main__':  
    main()
```

Результаты выполнения:

```
Задание A1  
[('Cyanine', 650, 'CDDb'), ('Cyanine', 702, 'CDDb'), ('Cyanine', 790, 'CDDb'), ('Cyanine', 870, 'Dacal CD Library'), ('Phthalocyanine', 650, 'Dacal CD Library'), ('Phthalocyanine', 650, 'Dacal CD Library'), ('Phthalocyanine', 702, 'Proceed'), ('Phthalocyanine', 702, 'Proceed')]  
  
Задание A2  
[('Dacal CD Library', 2170), ('CDDb', 2142), ('Proceed', 1404)]  
  
Задание A3  
{'Dacal CD Library': ['Cyanine', 'Phthalocyanine', 'Phthalocyanine']}  
  
Process finished with exit code 0
```