

МГТУ им. Н. Э. Баумана

Отчет по рубежному контролю №1
по курсу «Базовые компоненты и интернет-технологии»
Вариант 15.

Руководитель:
Гапанюк Ю. Е.
29.10.2022

Выполнил:
Студент группы ИУ5-34Б
Новиков Б. В.
29.10.2022

2022 г.

Полученное задание:

Необходимо создать два класса данных в соответствии с 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 – Файл, Класс 2 – Каталог файлов

Запросы:

1. «Каталог» и «Файл» связаны соотношением один-ко-многим. Выведите список всех каталогов, у которых название начинается с буквы «А», и список хранящихся в них файлов.
2. «Каталог» и «Файл» связаны соотношением один-ко-многим. Выведите список каталогов с максимальным размером файла в каждом каталоге, отсортированный по максимальному размеру.
3. «Каталог» и «Файл» связаны соотношением многие-ко-многим. Выведите список всех связанных файлов и каталогов, отсортированный по каталогам, сортировка по файлам произвольная.

Текст программы:

```
# используется для сортировки
from operator import itemgetter
```

```
class File:
    """Файл"""
    def __init__(self, id, fname, size, catalog_id):
        self.id = id
        self.fname = fname
        self.size = size
        self.catalog_id = catalog_id
```

```
class Catalog:
    """Каталог"""
    def __init__(self, id, name):
        self.id = id
        self.name = name
```

```
class FileCatalog:
    """
    'Файлы каталога' для реализации
    связи многие-ко-многим
    """
    def __init__(self, catalog_id, file_id):
        self.catalog_id = catalog_id
        self.file_id = file_id
```

```

# Каталоги
Catalogs = [
    Catalog(1, 'Downloads'),
    Catalog(2, 'Pictures'),
    Catalog(3, 'Applications'),

    Catalog(11, 'Downloads1'),
]

# Файлы
Files = [
    File(1, 'word-download.txt', 3, 1),
    File(2, 'alps.png', 7, 2),
    File(3, 'mount.jpg', 8, 2),
    File(4, 'tree.jpg', 6, 2),
    File(5, 'mario.exe', 5000, 3),
]

# Каталоги файлов
File_Catalogs = [
    FileCatalog(1,1),
    FileCatalog(2,2),
    FileCatalog(2,3),
    FileCatalog(2,4),
    FileCatalog(3,5),

    FileCatalog(11,1),
]

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

    # Соединение данных один-ко-многим
    one_to_many = [(f.fname, f.size, c.name)
                    for c in Catalogs
                    for f in Files
                    if f.catalog_id==c.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(c.name, fc.catalog_id, fc.file_id)
                          for c in Catalogs
                          for fc in File_Catalogs
                          if c.id==fc.catalog_id]

    many_to_many = [(f.fname, f.size, catalog_name)
                    for catalog_name, catalog_id, file_id in many_to_many_temp
                    for f in Files if f.id==file_id]

    print('Задание Г1')
    res_11 = {}
    # Перебираем все каталоги

```

```

for c in Catalogs:
    if c.name[0] == 'A':
        # Список файлов каталога
        c_Files = list(filter(lambda i: i[2]==c.name, one_to_many))
        # Только имя файла
        c_Files_names = [x for x,_,_ in c_Files]
        # Добавляем результат в словарь
        # ключ - каталог, значение - список файлов
        res_11[c.name] = c_Files_names
print(res_11)

print('\nЗадание Г2')
res_12_unsorted = []
# Перебираем все каталоги
for c in Catalogs:
    # Список файлов каталога
    c_Files = list(filter(lambda i: i[2]==c.name, one_to_many))
    # Если каталог не пустой
    if len(c_Files) > 0:
        # Зарплаты сотрудников отдела
        c_size = [size for _,size,_ in c_Files]
        # Суммарная зарплата сотрудников отдела
        c_size_max = max(c_size)
        res_12_unsorted.append((c.name, c_size_max))
# Сортировка по максимальному размеру файла
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))
print(res_13)

if __name__ == '__main__':
    main()

```

Результат выполнения:

```

[bogdan@ROG rkl]$ python main.py
Задание Г1
{'Applications': ['mario.exe']}

```

```

Задание Г2
[('Applications', 5000), ('Pictures', 8), ('Downloads', 3)]

```

```

Задание Г3
[('mario.exe', 5000, 'Applications'), ('word-download.txt', 3, 'Downloads'), ('word-downlo
ad.txt', 3, 'Downloads1'), ('alps.png', 7, 'Pictures'), ('mount.jpg', 8, 'Pictures'), ('tr
ee.jpg', 6, 'Pictures')]

```