

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
им. Н.Э. Баумана

Факультет "Информатика и системы управления"  
Кафедра "Системы обработки информации и управления"



Дисциплина "Парадигмы и конструкции языков программирования"

Отчет по рубежному контролю №1  
Вариант 15Г

**Выполнил:**  
Студент группы ИУ5-35Б  
Королев М.О.  
**Преподаватель:**  
Гапанюк Ю.Е.

Москва 2025

### **Задание:**

Рубежный контроль представляет собой разработку программы на языке Python, которая выполняет следующие действия:

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 15:**

Класс 1: Файл

Класс 2: Каталог файлов

### **Вариант Г.**

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

## Листинг кода:

```
# main.py
#####
Вариант 15Г: Класс 1 -- Файл, Класс 2 -- Каталог файлов
Запросы: 1) ...название начинается с буквы «А», и список...
          2) ...список каталогов с максимальным размером файлов в каждом каталоге,
          отсортированный по максимальному размеру.
          3) ...список всех связанных файлов и каталогов, отсортированный по отделам...
#####

class File:
    def __init__(self, id, name, size, catalogue_id):
        self.id = id
        self.name = name
        self.size = size
        self.catalogue_id = catalogue_id

    def __repr__(self):
        return f"File(id={self.id}, name='{self.name}', size={self.size},
catalogue_id={self.catalogue_id})"

class Catalogue:
    def __init__(self, id, name):
        self.id = id
        self.name = name

    def __repr__(self):
        return f"Catalogue(id={self.id}, name='{self.name}')"

class FileCatalogue:
    def __init__(self, file_id, catalogue_id):
        self.file_id = file_id
        self.catalogue_id = catalogue_id

files = [
    File(1, "report.txt", 1024, 1),
    File(2, "Ivanov_document.pdf", 2048, 2),
    File(3, "presentation.pptx", 3072, 1),
    File(4, "Sidorov_data.txt", 512, 2),
    File(5, "Petrov_notes.doc", 4096, 3),
    File(6, "image.jpg", 1536, 2),
    File(7, "archive.zip", 10240, 3)
]

catalogues = [
    Catalogue(1, "Documents"),
    Catalogue(2, "User Files"),
    Catalogue(3, "Archives")
]

file_catalogue = [
    FileCatalogue(1, 1),
    FileCatalogue(2, 2),
    FileCatalogue(2, 1),
    FileCatalogue(3, 1),
    FileCatalogue(4, 2),
    FileCatalogue(5, 3),
    FileCatalogue(5, 2),
    FileCatalogue(6, 2),
]
```

```

FileCatalogue(7, 3)
]

def first_task(output=True):
    ret = {}
    if output: print("--- TASK ONE ---")

    for catalogue in [c for c in catalogues if c.name.startswith('A')]:
        if output: print(f"Catalogue: {catalogue.name}")
        files_in_catalogue = [f for f in files if f.catalogue_id == catalogue.id]
        ret[catalogue.name] = files_in_catalogue
        if files_in_catalogue:
            for f in files_in_catalogue:
                if output: print("    ", f)
        else:
            if output: print("    is empty")

    if output: print("--- TASK END ---")
    return ret

def second_task(output=True):
    ret = []
    if output: print("--- TASK TWO ---")

    cats = sorted(catalogues, key=lambda cat: max(
        (f.size for f in files if f.catalogue_id == cat.id),
        default = 0
    ), reverse=True)

    for c in cats:
        size = max((f.size for f in files if f.catalogue_id == c.id), default = 0)
        if size > 0:
            if output: print(f"{str(c).ljust(34, ' ')} with max file size of {str(size).rjust(5, ' ')} bytes")
        ret.append((c, size))

    if output: print("--- TASK END ---")
    return ret

def third_task(output=True):
    if output: print("-- THIRD TASK --")
    cells_interlinked = sorted([
        (catalogue, file)
        for relation in file_catalogue
        for catalogue in catalogues
        if catalogue.id == relation.catalogue_id
        for file in files
        if file.id == relation.file_id
    ],
    key=lambda x: (x[0].name, -x[1].size))

    for x in cells_interlinked:
        if output: print(f"{str(x[0]).ljust(34, ' ')} {x[1]}")

    if output: print("--- TASK END ---")
    return cells_interlinked

if __name__ == "__main__":

```

```
print()
first_task()
print()
second_task()
print()
third_task()
```

## Результат работы программы:

[запуск программы]

```
--- TASK ONE ---
Catalogue: Archives
    File(id=5, name='Petrov_notes.doc', size=4096, catalogue_id=3)
    File(id=7, name='archive.zip', size=10240, catalogue_id=3)
--- TASK END ---

--- TASK TWO ---
Catalogue(id=3, name='Archives')      with max file size of 10240 bytes
Catalogue(id=1, name='Documents')     with max file size of 3072 bytes
Catalogue(id=2, name='User Files')    with max file size of 2048 bytes
--- TASK END ---

-- THIRD TASK --
Catalogue(id=3, name='Archives')      File(id=7, name='archive.zip', size=10240,
catalogue_id=3)                      File(id=5, name='Petrov_notes.doc',
                                         size=4096, catalogue_id=3)
Catalogue(id=3, name='Archives')      File(id=3, name='presentation.pptx',
size=4096, catalogue_id=3)           File(id=2, name='Ivanov_document.pdf',
                                         size=3072, catalogue_id=1)
Catalogue(id=1, name='Documents')     File(id=1, name='report.txt', size=1024,
size=3072, catalogue_id=1)           Catalogue(id=1, name='Documents')      File(id=5, name='Petrov_notes.doc',
                                         size=2048, catalogue_id=2)
Catalogue(id=1, name='Documents')     File(id=2, name='Ivanov_document.pdf',
                                         size=2048, catalogue_id=2)
Catalogue(id=1, name='Documents')     File(id=6, name='image.jpg', size=1536,
                                         catalogue_id=1)
Catalogue(id=2, name='User Files')    File(id=4, name='Sidorov_data.txt',
                                         size=512, catalogue_id=2)
--- TASK END ---

[программа завершила выполнение]
```