Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка интернет приложений» Рубежный контроль №1

Вариант №15Б

|  |  |
| --- | --- |
| Выполнил: | Проверил: |
| студент группы ИУ5-52Б | преподаватель каф. ИУ5 |
| Никитина Ксения | Гапанюк Ю.Е. |

Москва, 2021 г.

**Описание задания:**

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 Б.**

1. «Каталог» и «Файл» связаны соотношением один-ко-многим. Выведите список всех связанных файлов и каталогов, отсортированный по файлам, сортировка по каталогам произвольная.
2. «Каталог» и «Файл» связаны соотношением один-ко-многим. Выведите список каталогов с количеством файлов в нем, отсортированный по количеству файлов.
3. «Каталог» и «Файл» связаны соотношением многие-ко-многим (если каталоги вложенные, то один файл принадлежит нескольким каталогам). Выведите список всех файлов, у которых расширение png и названия их каталогов.

# Текст программы

|  |
| --- |
| class File: |
|  | def \_\_init\_\_(self, id, name, size, extension, id\_catalog): |
|  | self.id = id |
|  | self.name = name |
|  | self.size = size |
|  | self.extension = extension |
|  | self.id\_catalog = id\_catalog |
|  |  |
|  | def full\_name(self): |
|  | return f'{self.name}{self.extension}' |
|  |  |
|  |  |
|  | class Catalog: |
|  | def \_\_init\_\_(self, id, name, ): |
|  | self.id = id |
|  | self.name = name |
|  |  |
|  |  |
|  | class FileCatalog: |
|  | def \_\_init\_\_(self, file\_id, catalog\_id): |
|  | self.file\_id = file\_id |
|  | self.catalog\_id = catalog\_id |
|  |  |
|  |  |
|  | files = [ |
|  | File(1, 'Реферат', 310, '.docx', 5), |
|  | File(2, 'Фото1', 63, '.png', 2), |
|  | File(3, 'Песня', 46, '.mp3', 4), |
|  | File(4, 'Методичка', 22450, '.pdf', 3), |
|  | File(5, 'Фото2', 78, '.png', 2), |
|  | File(6, 'Отчет', 458, '.docx', 5), |
|  | File(7, 'Фильм', 24780, '.mp4', 1), |
|  | File(8, 'Фото3', 57, '.png', 3), |
|  | ] |
|  |  |
|  | catalogs = [ |
|  | Catalog(1, 'Видео'), |
|  | Catalog(2, 'Изображения'), |
|  | Catalog(3, 'Загрузки'), |
|  | Catalog(4, 'Музыка'), |
|  | Catalog(5, 'Документы'), |
|  | ] |
|  |  |
|  | file\_catalog = [ |
|  | FileCatalog(1, 3), |
|  | FileCatalog(1, 5), |
|  | FileCatalog(2, 1), |
|  | FileCatalog(2, 3), |
|  | FileCatalog(3, 2), |
|  | FileCatalog(4, 3), |
|  | FileCatalog(4, 5), |
|  | FileCatalog(5, 1), |
|  | FileCatalog(5, 3), |
|  | FileCatalog(6, 3), |
|  | FileCatalog(8, 5), |
|  | FileCatalog(8, 1), |
|  | ] |
|  |  |
|  |  |
|  | def main(): |
|  | print('Задание Б1') |
|  | res = sorted([(f.name, c.name) for f in |
|  | files for c in catalogs if c.id == f.id\_catalog], key=lambda x: x[0]) |
|  | for r in res: |
|  | print(r) |
|  |  |
|  | print('\nЗадание Б2') |
|  | res = sorted({c.name: len(list(filter(lambda x: x.id\_catalog == c.id, files))) for c in catalogs}.items(), |
|  | key=lambda x: x[1], |
|  | reverse=True) |
|  | for r in res: |
|  | print(r) |
|  |  |
|  | print('\nЗадание Б3') |
|  | res = {f.full\_name(): [c.name for c in catalogs if c.id in |
|  | [f\_c.catalog\_id for f\_c in file\_catalog if |
|  | f\_c.file\_id == f.id]] for f in files if |
|  | str(f.full\_name()).endswith('.png')}.items() |
|  | for r in res: |
|  | print(r) |
|  |  |
|  |  |
|  | if \_\_name\_\_ == '\_\_main\_\_': |
|  | main() |

# Экранная форма с результатом выполнения программы:

