**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Радиотехнический»

Кафедра «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по РК №1

Вариант запросов: Е

Вариант предметной области 24

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы РТ5-31Б: |  | преподаватель кафедры ИУ5 |
| Чиканчи В. А. |  | Гапанюк Ю. Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2024 г.

Постановка задачи

**Вариант Е.**

1. «Книга» и «Глава» связаны соотношением один-ко-многим. Выведите список всех отделов, у которых в названии присутствует слово «книга», и список глав.
2. «Книга» и «Глава» связаны соотношением один-ко-многим. Выведите список книг со средним количеством страниц в каждой книге, отсортированный по среднему количеству. Значения должна быть округлена до 2 знака после запятой (*отдельной функции вычисления среднего значения в Python нет, нужно использовать комбинацию функций вычисления суммы и количества значений; для округления необходимо использовать функцию* [*https://docs.python.org/3/library/functions.html#round*](https://docs.python.org/3/library/functions.html#round)).
3. «Книга» и «Глава» связаны соотношением многие-ко-многим. Выведите список всех Глав, у которых название начинается с буквы «К», и названия их книг.

Текст программы

**Файл «main.py»:**

# используется для сортировки

from operator import itemgetter

class Chapter:

    def **\_\_init\_\_**(self, id, name, pages, book\_id):

        self.id = id

        self.name = name

        self.pages = pages

        self.book\_id = book\_id

class Book:

    def **\_\_init\_\_**(self, id, name):

        self.id = id

        self.name = name

class ChapBook:

    def **\_\_init\_\_**(self, book\_id, chapter\_id):

        self.book\_id = book\_id

        self.chapter\_id = chapter\_id

books = [

    Book(1, 'Книга кадров'),

    Book(2, 'Книга жизни'),

    Book(3, 'джунги'),

    Book(11, 'Жизнь'),

    Book(22, 'Мир и мир'),

    Book(33, 'Какая-то ещё'),

]

chapters = [

    Chapter(1, 'Глава 1', 25000, 1),

    Chapter(2, 'Глава 2', 35000, 2),

    Chapter(3, 'КГлава 3', 45000, 3),

    Chapter(4, 'КГлава 4', 35000, 1),

    Chapter(5, 'КГлава 5', 25000, 2),

    Chapter(6, 'КГлава 6', 15324, 1),

    Chapter(7, 'Глава 7', 76585, 2),

    Chapter(8, 'Глава 8', 1734, 3)

]

chapters\_books = [

    ChapBook(1,1),

    ChapBook(2,2),

    ChapBook(3,3),

    ChapBook(3,4),

    ChapBook(3,5),

    ChapBook(11,1),

    ChapBook(22,2),

    ChapBook(33,3),

    ChapBook(33,4),

    ChapBook(33,5),

    ChapBook(33,6),

]

def **main**():

    """Основная функция"""

    one\_to\_many = [(c.name, c.pages, b.name)

        for b in books

        for c in chapters

        if b.id==c.book\_id]

    many\_to\_many\_temp = [(b.name, cb.book\_id, cb.chapter\_id)

        for b in books

        for cb in chapters\_books

        if b.id==cb.book\_id]

    many\_to\_many = [(c.name, c.pages, book\_name)

        for book\_name, book\_id, chapter\_id in many\_to\_many\_temp

        for c in chapters if c.id==chapter\_id]

**print**('Задание E1')

    res\_11 = [res

        for res in one\_to\_many

        if "книга" in res[2].**lower**()

        ]

**print**(res\_11)

**print**('\nЗадание E2')

    res\_12\_unsorted = []

    for b in books:

        b\_chapters = list(filter(lambda i: i[2]==b.name, one\_to\_many))

        if **len**(b\_chapters) > 0:

            b\_pages = [page for \_,page,\_ in b\_chapters]

            b\_page\_mid = **round**(**sum**(b\_pages)/**len**(b\_pages),2)

            res\_12\_unsorted.**append**((b.name, b\_page\_mid))

    res\_12 = **sorted**(res\_12\_unsorted, key=itemgetter(1), reverse=True)

**print**(res\_12)

**print**('\nЗадание E3')

    res\_13 = {}

    for c in chapters:

        if 'К' in c.name:

            b\_chapters = list(filter(lambda i: i[0]==c.name, many\_to\_many))

            b\_chapters\_st = [x for \_,\_,x in b\_chapters]

            res\_13[c.name] = b\_chapters\_st

**print**(res\_13)

if \_\_name\_\_ == '\_\_main\_\_':

**main**()

Экранные формы с примерами выполнения программы

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание