

Московский государственный технический университет им. Н.Э. Баумана

Факультет “Радиотехнический”
Кафедра ИУ5 “Системы обработки информации и управления”

Отчет по РК1 по курсу **Базовые компоненты интернет технологий**

Вариант 14

Подготовил:

Студент группы РТ5-31Б

Мамаев Т.Э.

Проверил:

Доцент кафедры ИУ5

Гапанюк Ю.Е.

24 Октябрь 2021г.

Задание

1) Необходимо создать два класса данных в соответствии с Вашим вариантом предметной области, которые связаны отношениями один-ко-многим и многие-ко-многим.

Пример классов данных для предметной области
Сотрудник-Отдел:

1. Класс «Сотрудник», содержащий поля:

- ID записи о сотруднике;
- Фамилия сотрудника;
- Зарплата (количественный признак);
- ID записи об отделе. (для реализации связи один-ко-многим)

2. Класс «Отдел», содержащий поля:

- ID записи об отделе;
- Наименование отдела.

3. (Для реализации связи многие-ко-многим) Класс «Сотрудники отдела», содержащий поля:

- ID записи о сотруднике;

- ID записи об отделе.
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). Для реализации запроса No2 введите в класс, находящийся на стороне связи «много», произвольный количественный признак, например, «зарплата сотрудника».
Результатом рубежного контроля является документ в формате PDF, который содержит текст программы и результаты ее выполнения.
Вариант Е.

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех отделов, у которых в названии присутствует слово «отдел», и список работающих в них сотрудников.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов со средней

зарплатой сотрудников в каждом отделе, отсортированный по средней зарплате. Средняя зарплата 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>).

3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех сотрудников, у которых фамилия начинается с буквы «А», и названия их отделов.

Текст программы

```
from operator import itemgetter

"""CD-диск"""

class CD:

    def __init__(self, id, name_cd, cap, lib_id):

        self.id = id

        self.name_cd = name_cd # имя

        self.cap = cap # вместимость диска (Мегабайт)

        self.lib_id = lib_id # в какой библиотеке

"""Библиотека CD-дисков"""

class lib:
```

```
def __init__(self, id, name):
```

```
    self.id = id
```

```
    self.name = name
```

```
class CdLib:
```

```
    """СВЯЗЬ МНОГИЕ КО МНОГИМ"""
```

```
    def __init__(self, lib_id, cd_id):
```

```
        self.lib_id = lib_id
```

```
        self.cd_id = cd_id
```

```
# библиотеки
```

```
libs = [
```

```
    lib(1, 'Первая Библиотека'),
```

```
    lib(2, 'Вторая Библиотека'), lib(3, 'Третья Библиотека'),
```

```
    lib(4, 'Четвертая Библиотека'), lib(5, 'Пятая Библиотека'),  
    lib(6, 'Шестая Библиотека'),
```

```
]
```

```
# CD-диски
```

```
cds = [
```

```
    CD(2, 'Kodack', 2048, 1), CD(3, 'Samsung', 1024, 3), CD(4,  
    'Sharp', 3072, 4),
```

```
    CD(5, 'Sony', 4096, 5), CD(6, 'Piratsky', 2048, 1), CD(7,  
    'Palm', 512, 6), CD(8, 'IBM', 4096, 4), CD(9, 'Toshiba', 4096, 5)
```

```
]
```

```
cds_libs = [ CdLib(1,1), CdLib(1,2), CdLib(1,4), CdLib(1,8),  
CdLib(2,4), CdLib(2,5), CdLib(2,6), CdLib(2,1), CdLib(3,3),  
CdLib(4,1), CdLib(4,5), CdLib(5,4), CdLib(5,2), CdLib(6,7),  
]
```

```
def main():
```

```
    """Основная функция"""
```

```
    # Соединение данных один-ко-многим
```

```
    one_to_many = [(h.name_cd, h.cap, p.name)
```

```
        for p in libs
```

```
        for h in cds
```

```
        if h.lib_id==p.id]
```

```
    # Соединение данных многие-ко-многим
```

```
    many_to_many_temp = [(p.name, ph.lib_id, ph.cd_id)
```

```
        for p in libs
```

```
        for ph in cds_libs if p.id==ph.lib_id]
```

```
    many_to_many = [(h.name_cd, h.cap, lib_name)
```

```
        for lib_name, lib_id, cd_id in many_to_many_temp for h  
in cds if h.id==cd_id]
```

```
    print('Задание E1')
```

```
    res_E1 = []
```

```
    for name_cd, cap, name in one_to_many:
```

```
        if 'Первая' in name: # Ищем библиотеки с ключевым  
словом "Вторая"
```

```

        res_E1.append((name, name_cd))

print(res_E1)

print('\nЗадание E2') # находим среднюю вместимость
ДИСКОВ

res_E2_unsorted = [] # Перебираем все библиотеки

for p in libs: # Список дисков библиотеки

    list_cd = list(filter(lambda i: i[2]==p.name, one_to_many))
# Если в библиотеке есть диск

    if len(list_cd) > 0: # вместимость CD

        list_cap = [cap for _, cap, _ in list_cd]

        # средняя вместимость

        avg_sum = sum(list_cap)/len(list_cap)

        res_E2_unsorted.append((p.name, avg_sum))

res_E2 = sorted(res_E2_unsorted, key=itemgetter(1))

print(res_E2)

print('\nЗадание E3')

# находим диски, начинающиеся с "P" и выводим их
библиотеки

res_E3 = []

for name_cd, cap, name in many_to_many:

    if name_cd.find("P") == 0:

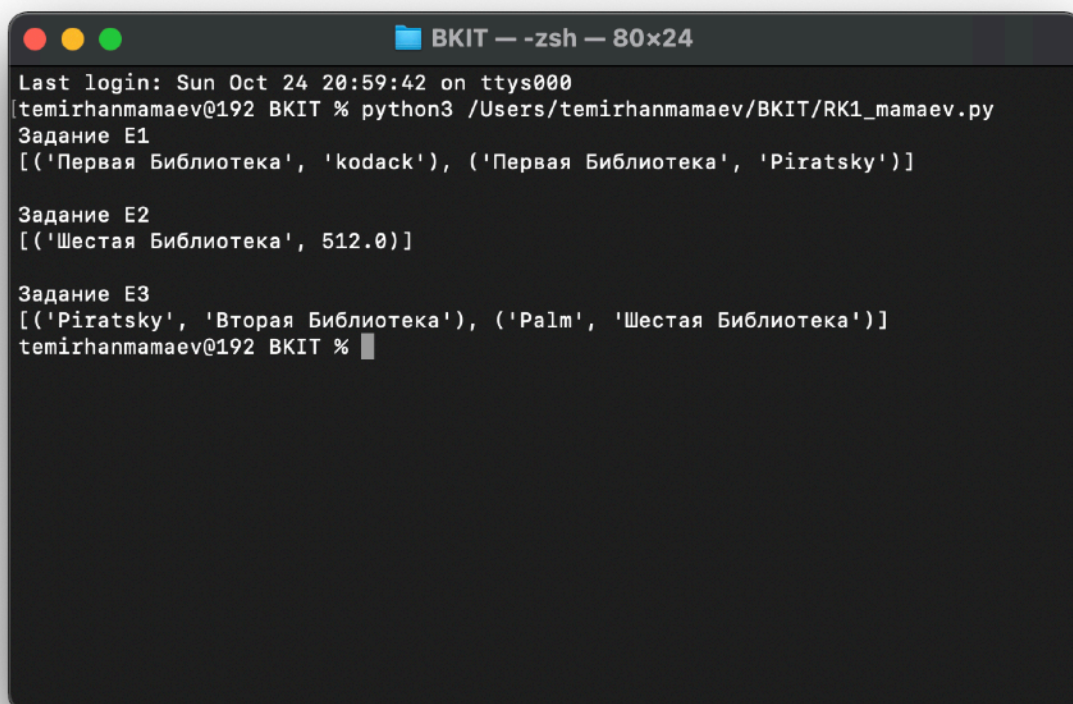
        res_E3.append((name_cd, name))

```

```
print(res_E3)

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

Результат выполнения программы

A terminal window titled "BKIT — -zsh — 80x24" with standard macOS window controls (red, yellow, green buttons). The terminal shows the output of a Python script. It starts with a login message, then shows the command to run the script. The output consists of three tasks: Task E1 shows a list of two tuples, Task E2 shows a list with one tuple, and Task E3 shows a list of two tuples. The prompt returns to the user after each task.

```
Last login: Sun Oct 24 20:59:42 on ttys000
temirhanmamaev@192 BKIT % python3 /Users/temirhanmamaev/BKIT/RK1_mamaev.py
Задание E1
[('Первая Библиотека', 'kodack'), ('Первая Библиотека', 'Piratsky')]

Задание E2
[('Шестая Библиотека', 512.0)]

Задание E3
[('Piratsky', 'Вторая Библиотека'), ('Palm', 'Шестая Библиотека')]
temirhanmamaev@192 BKIT %
```