

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 8

Подготовил:  
Студент группы РТ5-31Б  
Коваленко В.И.

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 №2 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

class HDD:
    """Жесткий диск"""
    def __init__(self, id, name_hdd, cap, pc_id):
        self.id = id
        self.name_hdd = name_hdd # имя
        self.cap = cap # вместимость диска (Гигабайт)
        self.pc_id = pc_id # в каком компьютере

class PC:
    """Компьютер"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class HddPc:
    """
    СВЯЗЬ МНОГИЕ КО МНОГИМ
    """
    def __init__(self, pc_id, hdd_id):
        self.pc_id = pc_id
        self.hdd_id = hdd_id

# Компьютеры
Pcs = [
    PC(1, 'компьютер HP'),
    PC(2, 'компьютер MSI'),
    PC(3, 'MacBook Pro'),
    PC(4, 'компьютер Alienware'),
    PC(5, 'компьютер Asus'),
    PC(6, 'MacBook Air'),
]

# Жесткие диски
hdds = [
```

```

HDD(1, 'Seagate', 5120, 2),
HDD(2, 'Samsung', 2048, 1),
HDD(3, 'Macintosh', 1024, 3),
HDD(4, 'Toshiba', 3072, 4),
HDD(5, 'WD blue', 4096, 5),
HDD(6, 'WD gold', 2048, 1),
HDD(7, 'Macintosh', 512, 6),
HDD(8, 'Seagate Baracuda', 4096, 4),
HDD(9, 'Toshiba', 4096, 5)
]

```

```

hdds_pcs = [
    HddPc(1,1),
    HddPc(1,2),
    HddPc(1,4),
    HddPc(1,8),
    HddPc(2,4),
    HddPc(2,5),
    HddPc(2,6),
    HddPc(2,1),
    HddPc(3,3),
    HddPc(4,1),
    HddPc(4,5),
    HddPc(5,4),
    HddPc(5,2),
    HddPc(6,7),

```

```

]
```

```

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

    # Соединение данных один-ко-многим
    one_to_many = [(h.name_hdd, h.cap, p.name)
                    for p in Pcs
                    for h in hdds
                    if h.pc_id==p.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(p.name, ph.pc_id, ph.hdd_id)
                          for p in Pcs
                          for ph in hdds_pcs
                          if p.id==ph.pc_id]

    many_to_many = [(h.name_hdd, h.cap, pc_name)

```

```

    for pc_name, pc_id, hdd_id in many_to_many_temp
    for h in hdds if h.id==hdd_id]

print('Задание E1')
res_E1 = []
for name_hdd, cap, name in one_to_many:
    if 'компьютер' in name: # Ищем компьютеры с ключевым словом
"компьютер"
        res_E1.append((name, name_hdd))
print(res_E1)

print('\nЗадание E2')
# находим среднюю вместимость жестких дисков
res_E2_unsorted = []
# Перебираем все компьютеры
for p in PCs:
    # Список жестких дисков компьютера
    list_hdd = list(filter(lambda i: i[2]==p.name, one_to_many))
    # Если в компьютере есть жесткий диск
    if len(list_hdd) > 0:
        # вместимомть HDD
        list_cap = [cap for _,cap,_ in list_hdd]
        # 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 жесткие диски, начинающиеся с "S" и выводим их компьютеры
res_E3 = []
for name_hdd, cap, name in many_to_many:
    if name_hdd.find("S") == 0:
        res_E3.append((name_hdd, name))
print(res_E3)

if __name__ == '__main__':
    main()

```

## Результат выполнения программы

```
GitHub — -zsh — 80x15

Задание E1
[('компьютер HP', 'Samsung'), ('компьютер HP', 'WD gold'), ('компьютер MSI', 'Seagate'), ('компьютер Alienware', 'Toshiba'), ('компьютер Alienware', 'Seagate Baracuda'), ('компьютер Asus', 'WD blue'), ('компьютер Asus', 'Toshiba')]

Задание E2
[('MacBook Air', 512.0), ('MacBook Pro', 1024.0), ('компьютер HP', 2048.0), ('компьютер Alienware', 3584.0), ('компьютер Asus', 4096.0), ('компьютер MSI', 5120.0)]

Задание E3
[('Seagate', 'компьютер HP'), ('Samsung', 'компьютер HP'), ('Seagate Baracuda', 'компьютер HP'), ('Seagate', 'компьютер MSI'), ('Seagate', 'компьютер Alienware'), ('Samsung', 'компьютер Asus')]
kovalenkov@MacBook-Air-Vlad GitHub %
```