

**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 №9В**

**Выполнил:**

студент группы ИУ5-53Б  
Воронова Ольга

**Проверил:**

преподаватель каф. ИУ5  
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 В.

1. «Операционная система» и «Компьютер» связаны соотношением один-ко-многим. Выведите список всех компьютеров, у которых название начинается с буквы «А», и их операционные системы.
2. «Операционная система» и «Компьютер» связаны соотношением один-ко-многим. Выведите список ОС с минимальной памятью компьютеров для каждой ОС, отсортированный по минимальной памяти.
3. «Операционная система» и «Компьютер» связаны соотношением многие-ко-многим. Выведите список всех связанных компьютеров и ОС, отсортированный по компьютерам, сортировка по ОС производная.

## Текст программы

```
from operator import itemgetter
```

```
class OS:
```

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

```
class Computer:
```

```
    def __init__(self, id, brand, memory_gb, os_id):
```

```
self.id = id
self.brand = brand
self.memory_gb = memory_gb
self.os_id = os_id
```

```
class ComputerOS:
```

```
def __init__(self, computer_id, id_os):
    self.computer_id = computer_id
    self.id_os = id_os
```

```
operating_system = [
    OS(1, 'Windows 10'),
    OS(2, 'Linux Ubuntu 21.10'),
    OS(3, 'MacOS Big Sur 11.6'),
    OS(11, 'Windows 8'),
    OS(22, 'Linux Ubuntu 20.04'),
    OS(33, 'MacOS Big Sur 11.5'),
]
```

```
computers = [
    Computer(1, 'Asus', 64, 2),
    Computer(2, 'Lenovo', 128, 2),
    Computer(3, 'Acer', 128, 1),
    Computer(4, 'Dell', 64, 1),
    Computer(5, 'Acer', 256, 2),
    Computer(6, 'HP', 128, 1),
    Computer(7, 'Apple', 128, 3),
    Computer(8, 'Xiaomi', 256, 1),
    Computer(9, 'Apple', 512, 3),
]
```

```
computers_os = [
    ComputerOS(1, 2),
    ComputerOS(2, 2),
    ComputerOS(3, 1),
    ComputerOS(4, 1),
    ComputerOS(5, 2),
    ComputerOS(6, 1),
    ComputerOS(7, 3),
    ComputerOS(8, 1),
    ComputerOS(9, 3),
    ComputerOS(1, 22),
    ComputerOS(2, 22),
    ComputerOS(3, 11),
    ComputerOS(4, 11),
    ComputerOS(5, 22),
    ComputerOS(6, 11),
]
```

```
ComputerOS(7, 33),
ComputerOS(8, 11),
ComputerOS(9, 33),
]
```

```
def main():
```

```
    one_to_many = [(comp.brand, comp.memory_gb, os.name)
                    for comp in computers
                    for os in operating_system
                    if comp.os_id == os.id]
```

```
    many_to_many_temp = [(os.name, comp_os.computer_id, comp_os.id_os)
                          for os in operating_system
                          for comp_os in computers_os
                          if os.id == comp_os.id_os]
```

```
    many_to_many = [(comp.brand, os_name)
                    for os_name, comp_id, os_id in many_to_many_temp
                    for comp in computers if comp.id == comp_id]
```

```
    print("\n\nЗадание B1')
```

```
    res = list(filter(lambda i: str(i[0]).startswith('A'), one_to_many))
```

```
    res = [
        (elem[0], elem[2])
        for elem in res
    ]
```

```
    print(res)
```

```
    print("\n\nЗадание B2')
```

```
    res = sorted(one_to_many, key=itemgetter(1))
```

```
    res = [
        (elem[2], elem[1])
        for elem in res
    ]
```

```
    print(res)
```

```
    print("\n\nЗадание B3')
```

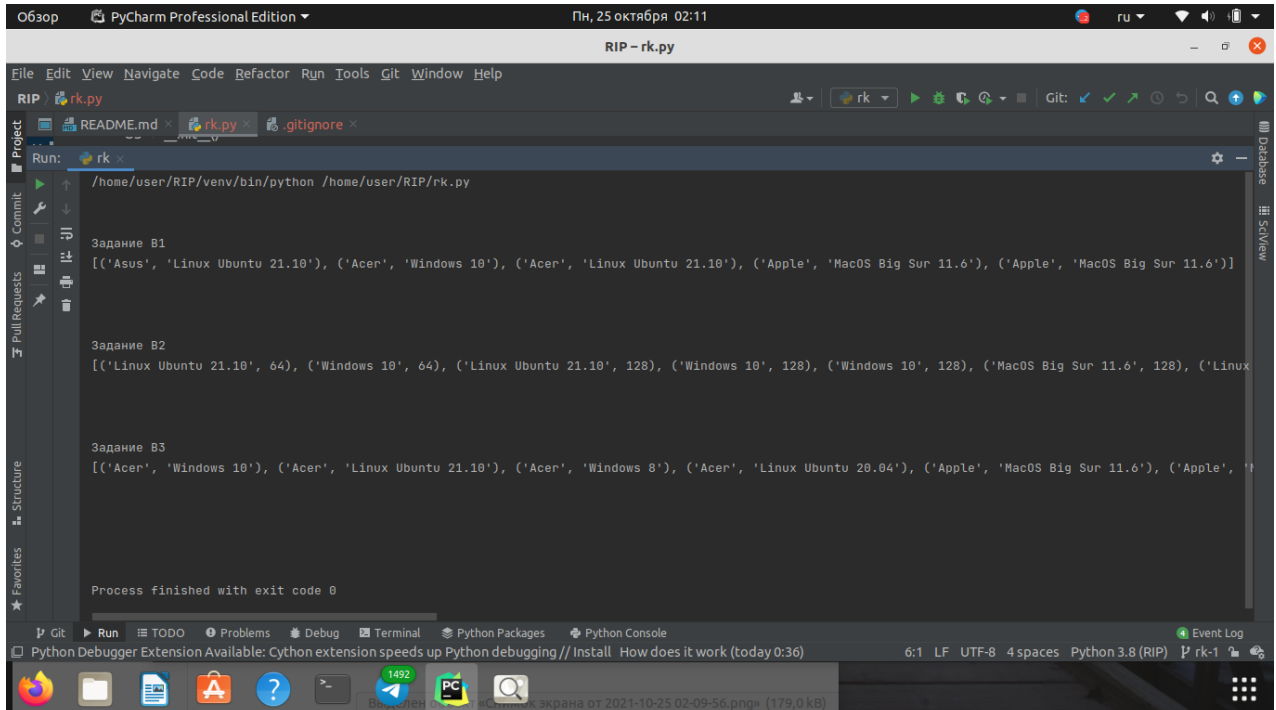
```
    res = sorted(many_to_many, key=itemgetter(0))
```

```
    print(res)
```

```
    print("\n\n')
```

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

## Результат выполнения программы:



```
Обзор PyCharm Professional Edition Пн, 25 октября 02:11
RIP - rk.py
File Edit View Navigate Code Refactor Run Tools Git Window Help
RIP > rk.py
Project: rk.py
Run: rk
/home/user/RIP/venv/bin/python /home/user/RIP/rk.py

Задание B1
[('Asus', 'Linux Ubuntu 21.10'), ('Acer', 'Windows 10'), ('Acer', 'Linux Ubuntu 21.10'), ('Apple', 'MacOS Big Sur 11.6'), ('Apple', 'MacOS Big Sur 11.6')]

Задание B2
[('Linux Ubuntu 21.10', 64), ('Windows 10', 64), ('Linux Ubuntu 21.10', 128), ('Windows 10', 128), ('Windows 10', 128), ('MacOS Big Sur 11.6', 128), ('Linux Ubuntu 21.10', 256), ('Windows 10', 256), ('MacOS Big Sur 11.6', 512)]

Задание B3
[('Acer', 'Windows 10'), ('Acer', 'Linux Ubuntu 21.10'), ('Acer', 'Windows 8'), ('Acer', 'Linux Ubuntu 20.04'), ('Apple', 'MacOS Big Sur 11.6'), ('Apple', 'MacOS Big Sur 11.5'), ('Apple', 'MacOS Big Sur 11.5'), ('Asus', 'Linux Ubuntu 21.10'), ('Asus', 'Linux Ubuntu 20.04'), ('Dell', 'Windows 10'), ('Dell', 'Windows 8'), ('HP', 'Windows 10'), ('HP', 'Windows 8'), ('Lenovo', 'Linux Ubuntu 21.10'), ('Lenovo', 'Linux Ubuntu 20.04'), ('Xiaomi', 'Windows 10'), ('Xiaomi', 'Windows 8')]

Process finished with exit code 0
```

### Задание B1

[('Asus', 'Linux Ubuntu 21.10'), ('Acer', 'Windows 10'), ('Acer', 'Linux Ubuntu 21.10'), ('Apple', 'MacOS Big Sur 11.6'), ('Apple', 'MacOS Big Sur 11.6')]

### Задание B2

[('Linux Ubuntu 21.10', 64), ('Windows 10', 64), ('Linux Ubuntu 21.10', 128), ('Windows 10', 128), ('Windows 10', 128), ('MacOS Big Sur 11.6', 128), ('Linux Ubuntu 21.10', 256), ('Windows 10', 256), ('MacOS Big Sur 11.6', 512)]

### Задание B3

[('Acer', 'Windows 10'), ('Acer', 'Linux Ubuntu 21.10'), ('Acer', 'Windows 8'), ('Acer', 'Linux Ubuntu 20.04'), ('Apple', 'MacOS Big Sur 11.6'), ('Apple', 'MacOS Big Sur 11.6'), ('Apple', 'MacOS Big Sur 11.5'), ('Apple', 'MacOS Big Sur 11.5'), ('Asus', 'Linux Ubuntu 21.10'), ('Asus', 'Linux Ubuntu 20.04'), ('Dell', 'Windows 10'), ('Dell', 'Windows 8'), ('HP', 'Windows 10'), ('HP', 'Windows 8'), ('Lenovo', 'Linux Ubuntu 21.10'), ('Lenovo', 'Linux Ubuntu 20.04'), ('Xiaomi', 'Windows 10'), ('Xiaomi', 'Windows 8')]

Process finished with exit code 0