

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Радиотехнический»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка интернет-приложений»
Рубежный контроль №1
Вариант Е11

Выполнил: студент группы РТ5-51Б
Незаметдинов Т. И.

Проверил: преподаватель кафедры ИУ5
Гапанюк Ю. Е.

Подпись и дата:

Подпись и дата:

Описание задания

1. «Программа» и «Компьютер» связаны соотношением один-ко-многим. Выведите список всех программ и компьютеров, отсортированных по Операционной системе компьютера, сортировка по программам отсутствует.
2. «Программа» и «Компьютер» связаны соотношением один-ко-многим. Выведите список компьютеров, отсортированный по кол-ву занимаемой программами памяти в порядке убывания.
3. «Программа» и «Компьютер» связаны соотношением многие-ко-многим. Выведите список всех Компьютеров, на операционной системе “Windows” и список работающих в них программ.

Текст программы

main.py

```
from operator import itemgetter
```

```
# Создаем класс-таблицу Компьютер
```

```
class PC:
```

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

```
        self.id = id
```

```
        self.name = name
```

```
        self.os = os
```

```
# Создаем класс-таблицу Программа
```

```
class Prog:
```

```
    def __init__(self, id, name, id pc, storage usage):
```

```
        self.id = id
```

```
        self.name = name
```

```
        self.storage usage = storage usage
```

```
        self.id pc = id pc
```

```
# Создаем класс-таблицу Программа-Компьютер для  
реализации связи М-М
```

```
class ProgPC:
```

```
def __init__(self, id_pc, id_prog):  
    self.id_pc = id_pc  
    self.id_prog = id_prog
```

```
# Заполняем класс-таблицу Программа тестовыми данными  
programs = [  
    Prog(1, 'VSCode', 3, 1.5),  
    Prog(2, 'PYCharm', 1, 2),  
    Prog(3, 'Google Chrome', 4, 3),  
    Prog(4, 'VM VirtualBox', 2, 10),  
    Prog(5, 'Slack', 5, 4),  
    Prog(6, 'Spotify', 3, 3.3),  
    Prog(7, 'Zoom', 1, 2.4),  
    Prog(8, 'Telegram', 3, 2.4),  
    Prog(9, 'Discord', 2, 4),  
]
```

```
# Заполняем класс-таблицу Компьютер тестовыми данными  
comps = [  
    PC(1, 'Asus', 'Windows'),  
    PC(2, 'Acer', 'Windows'),  
    PC(3, 'MacBook', 'MacOS'),  
    PC(4, 'Lenovo', 'Windows'),  
    PC(5, 'PC551', 'Linux'),  
]
```

```
# Заполняем класс-таблицу Программа-Компьютер тестовыми  
данными  
prog_pc = [  
    ProgPC(1, 2),  
    ProgPC(1, 7),  
    ProgPC(2, 4),  
    ProgPC(2, 9),  
    ProgPC(3, 1),  
    ProgPC(3, 6),  
    ProgPC(3, 8),  
    ProgPC(4, 3),
```

```
ProgPC(5, 5),
```

```
]
```

```
def main():
```

```
    print('Задание E1')
```

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

```
    one_to_many = [(pr.name, pr.storage_usage, pc.name, pc.os)
```

```
                    for pc in comps
```

```
                    for pr in programs
```

```
                    if pr.id == pc.id]
```

```
    res_1 = sorted(one_to_many, key=itemgetter(3)) #  
    Сортировка данных по операционной системе Компьютера
```

```
    print(res_1)
```

```
    print('Задание E2')
```

```
    res_2 = []
```

```
    for pc in comps:
```

```
        storage_usage_sum = sum([pr.storage_usage for pr  
in programs if (
```

```
        pr.id == pc.id)]) # Считаем суммарно  
        занимаемую программами память для каждого компьютера
```

```
        storage_usage_sum = round(storage_usage_sum, 1)  
# Округляем до 1 знака после запятой
```

```
        res_2.append([pc.name, storage_usage_sum]) #  
        Заполняем список новыми данными
```

```
        res_2 = sorted(res_2, key=itemgetter(1), #  
        Сортируем список относительно занимаемой памяти в  
        порядке убывания
```

```
        reverse=True)
```

```
    print(res_2)
```

```

print('Задание E3')

# Соединение данных много-ко-многим
many to many temp = [(pc.name, pc.os, pr pc.id prog)
                      for pc in comps
                      for pr pc in prog pc
                      if pc.id == pr pc.id pc
                      ]

many to many = [(pr.name, pc name, pc os)
                for pc name, pc os, prog id in
many to many temp
                for pr in programs if pr.id ==
prog id
                ]

res 3 = {}
for pc in comps:
    if 'Windows' in pc.os:
        pc_list = list(filter(lambda x: x[1] ==
pc.name, many_to_many)) # Из таблицы М-М достаем записи,
связанные с текущим индексом цикла for
        prog_list = [prog_name for prog_name,
pc_name, pc_os in pc_list] # Из полученных записей
формируем список работающих приложений для текущего
индекса for
        res 3[pc.name] = prog list
print(res 3)

if name == " main ":
    main()

```

Результат работы программы

```
C:\Users\Tim\Desktop\RK1\Scripts\python.exe C:/Users/Tim/Desktop/IU5-Web/RK1/main.py
Задание E1
[('Slack', 4, 'PC551', 'Linux'), ('VSCode', 1.5, 'MacBook', 'MacOS'), ('Spotify', 3.3, 'MacBook', 'MacOS'), ('Telegram', 2.4, 'MacBook', 'MacOS'), ('PYCharm', 2, 'Asus', 'Windows'), ('Zoom', 2.4, 'Asus', 'Windows'), ('VM
VirtualBox', 10, 'Acer', 'Windows'), ('Discord', 4, 'Acer', 'Windows'), ('Google Chrome', 3, 'Lenovo', 'Windows')]
Задание E2
[['Acer', 14], ['MacBook', 7.2], ['Asus', 4.4], ['PC551', 4], ['Lenovo', 3]]
Задание E3
{'Asus': ['PYCharm', 'Zoom'], 'Acer': ['VM VirtualBox', 'Discord'], 'Lenovo': ['Google Chrome']}
```