



**Министерство образования и науки Российской Федерации**

**Федеральное государственное бюджетное образовательное  
учреждение высшего образования**

**«Московский государственный технический университет  
имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

**Отчёт по рубежному контролю №1 по курсу**

**«Разработка Интернет-Приложений»**

**Тема работы: "Работа с классами данных в Python."**

Выполнил: Зоров В.В., РТ5-51Б

Проверил:\_\_\_\_\_

22 октября 2020 г.

ЗАЧТЕНО / НЕ ЗАЧТЕНО \_\_\_\_\_

(подпись)

- **Задание по варианту**

Вариант	Классы	Запросы
E8	<ul style="list-style-type: none"> <li>• Жесткий диск;</li> <li>• Компьютер;</li> <li>• Связь М-М между первыми двумя.</li> </ul>	<ul style="list-style-type: none"> <li>• Классы «1» и «2» имеют связь О-М. Вывести список групп компьютеров, описание которых содержит слово «для», а так же находящийся в каждом процессор;</li> <li>• Классы «1» и «2» имеют связь О-М. Вывести среднее количество ядер в процессорах, используемых в компьютерах каждой группы;</li> <li>• Классы «1» и «2» имеют связь М-М. Вывести информацию о процессорах, имеющих частоту в пределах [3,1 ГГц; 3,5 ГГц], а так же обо всех компьютерах, в которых эти процессоры используются.</li> </ul>

- **Ход выполнения работы**

class Disk:

#Жесткий диск

```
def __init__(self, id, nameD, DiskSize, comp_id):
```

```
    self.id = id
```

```
    #Наименование диска
```

```
    self.nameD = nameD
```

```
    #Размер диска
```

```
    self.DiskSize = DiskSize
```

```
    self.comp_id = comp_id
```

```
class Comp:
```

```
    #Компьютер
```

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

```
self.id = id
```

```
self.nameComp = nameComp
```

```
class DiskComp:
```

```
    #диски компьютеров
```

```
    #для реализации связи многие-ко-многим
```

```
    def __init__(self, comp_id, disk_id):
```

```
        self.comp_id = comp_id
```

```
        self.disk_id = disk_id
```

# ДИСКИ

disks = [

Disk(1, 'Western Digital', 3, 1),

Disk(2, 'Western Digital 2', 3000, 3),

Disk(3, 'Samsung EVO 970', 1000, 10),

Disk(4, 'Western Digital 3', 4, 10),

Disk(5, 'Western Digital 4', 15, 1),

Disk(6, 'Western Digital 5', 67, 2),

Disk(7, 'Western Digital 6', 108, 20),

Disk(8, 'Samsung EVO 980', 4000, 20),

Disk(9, 'Samsung EVO 960', 40000, 2),

Disk(10, 'Keenetic', 1000, 30),

Disk(11, 'Pasteroni', 30, 3),

Disk(12, 'MaddyMurk', 256, 30),

Disk(13, 'Pasteroni 2', 125, 40),

Disk(14, 'Puppet', 312, 4),

]

# Компьютеры

comps = [

Comp(1, "Компьютер ауд. 362"),

Comp(2, 'Компьютер сервер по ауд. 362'),

Comp(3, 'Компьютер админ. по сев. крылу'),

Comp(4, 'Компьютер админ. контроль ГЗ'),

Comp(10, 'Компьютер ауд. 903'),

Comp(20, 'Компьютер 306'),

Comp(30, 'Резервный админ. по сев. крылу'),

Comp(40, 'Резервный для админ. контроля ГЗ'),

]

Disk\_comps = [

DiskComp(1, 1),

DiskComp(1, 2),

DiskComp(1, 3),

DiskComp(1, 4),

DiskComp(1, 5),



DiskComp(2, 6),

DiskComp(2, 7),

DiskComp(2, 8),

DiskComp(2, 9),

DiskComp(3, 10),

DiskComp(3, 11),

DiskComp(3, 12),

DiskComp(4, 13),

DiskComp(4, 14),

DiskComp(10, 1),

DiskComp(10, 2),

DiskComp(10, 3),

DiskComp(10, 4),

DiskComp(10, 5),

DiskComp(20, 6),

DiskComp(20, 7),

DiskComp(20, 8),

DiskComp(20, 9),

DiskComp(30, 10),

DiskComp(30, 11),

DiskComp(30, 12),

DiskComp(40, 13),

DiskComp(40, 14),

]

```

def main():

    #Реализация связи один-ко-многим

    one_to_many = [(d.comp_id , d.nameD, d.DiskSize,
c.nameComp)

                    for c in comps

                    for d in disks if d.comp_id == c.id]


    #Решение задания A1

    #Выберем те компьютеры, у которых в названии есть
"админ." и выведем их наименование

    #и наименование установленных на них программ

```

```
print('\nЗадание A1\n')

res_1 = ""

for i in one_to_many:

    if "админ." in i[1]:

        res_1 = res_1 + str(i[4]) + ' с установленным диском: ' + str(i[1]) + '\n'

print(res_1)
```

**#Решение задания A2**

**#Выберем для каждого компьютера средний размер жестких дисков и выведем эти данные,**

**#предварительно отсортировав**

```

print("\nЗадание A2:\n")

c_disk_all = list()

for c in comps:

    #Выберем все диски установленные на
    рассматриваемом компьютере

    diskList = list(filter(lambda x: c.id == x[0],
one_to_many))

    c_disk = 0

    #Рассматривая каждый элемент списка всех
    программ компьютера

    for item in diskList:

        #Выбор значения количества дисков

        d = item[2]

```

```
c_disk = c_disk + d
```

```
#Находим среднее значение размеров дисков
```

```
c_disk = round(c_disk / len(diskList), 2)
```

```
#Добавляем найденное среднее значение в список  
для вывода данных
```

```
c_disk_all.append((c.nameComp, c_disk))
```

```
for item in sorted(c_disk_all, key=lambda x: x[1]):
```

```
    print("Для компьютера: {0}, в среднем размер диска  
{1} GB".format(item[0], item[1]))
```

```
#Реализация связи многие-ко-многим
```

```
many_to_many_temp = [(c.nameComp, cd.comp_id,  
cd.disk_id)
```

```
    for c in comps
```

```
        for cd in Disk_comps
```

```
            if c.id == cd.comp_id]
```

```
many_to_many = [(d.nameD, d.DiskSize, nameComp)
```

```
                for nameComp, compld, diskld in  
many_to_many_temp
```

```
                    for d in disks
```

```
                        if d.id == diskld]
```

**#Решение задания A3:**

**#Выберем данные из составленных связей многие-ко-многим, рассмотрим те диски,**

**#название которых начинается с буквы "W" и имена компьютеров**

```
print("\nЗадание A3:\n")
```

```
res_3 = "
```

```
for i in many_to_many:
```

```
    str3 = i[0]
```

```
    for k in range(len(str3)):
```

```
        if k == 0 and str3[k] == 'W':
```

```
            res_3 = res_3 + 'Диск: ' + str3 + ', установленный  
на компьютере: ' + str(i[3]) + '\n'
```

```
            break
```



```
else:
```

```
    break
```

```
print(res_3)
```

```
if __name__ == '__main__':
```

```
    main()
```

- **Результаты работы**

#### Задание A1

Компьютер админ. по сев. крылу с установленным диском: Western Digital 2  
Компьютер админ. по сев. крылу с установленным диском: Pasteroni  
Компьютер админ. контроль ГЗ с установленным диском: Puppet  
Резервный админ. по сев. крылу с установленным диском: Keenetic  
Резервный админ. по сев. крылу с установленным диском: MaddyMurk  
Резервный для админ. контроля ГЗ с установленным диском: Pasteroni 2

#### Задание A2:

Для компьютера: Компьютер ауд. 362, в среднем размер диска 9.0 GB  
Для компьютера: Резервный для админ. контроля ГЗ, в среднем размер диска 12  
Для компьютера: Компьютер админ. контроль ГЗ, в среднем размер диска 312.0  
Для компьютера: Воспомогательная ЭВМ, в среднем размер диска 502.0 GB  
Для компьютера: Резервный админ. по сев. крылу, в среднем размер диска 628.  
Для компьютера: Компьютер админ. по сев. крылу, в среднем размер диска 1515  
Для компьютера: Вспомогательный сервер, в среднем размер диска 2054.0 GB  
Для компьютера: Компьютер сервер по ауд. 362, в среднем размер диска 20033.

#### Задание A3:

#### Задание A3:

Диск: Western Digital, установленный на компьютере: Компьютер ауд. 362  
Диск: Western Digital 2, установленный на компьютере: Компьютер ауд. 362  
Диск: Western Digital 3, установленный на компьютере: Компьютер ауд. 362  
Диск: Western Digital 4, установленный на компьютере: Компьютер ауд. 362  
Диск: Western Digital 5, установленный на компьютере: Компьютер сервер по ауд.  
Диск: Western Digital 6, установленный на компьютере: Компьютер сервер по ауд.  
Диск: Western Digital, установленный на компьютере: Компьютер ауд. 903  
Диск: Western Digital 2, установленный на компьютере: Компьютер ауд. 903  
Диск: Western Digital 3, установленный на компьютере: Компьютер ауд. 903  
Диск: Western Digital 4, установленный на компьютере: Компьютер ауд. 903  
Диск: Western Digital 5, установленный на компьютере: Компьютер 306  
Диск: Western Digital 6, установленный на компьютере: Компьютер 306