

```
from operator import itemgetter
```

```
class HardDrive:
```

```
    """Жесткий диск"""
```

```
    def __init__(self, id, model, remainder, computer_id):
```

```
        self.id = id
```

```
        self.model = model
```

```
        self.remainder = remainder
```

```
        self.computer_id = computer_id
```

```
class Computer:
```

```
    """Компьютер"""
```

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

```
        self.id = id
```

```
        self.name = name
```

```
# Компьютеры
```

```
computers = [Computer(1, 'Компьютер 1'),
```

```
              Computer(2, 'Компьютер 2'),
```

```
              Computer(3, 'Компьютер 3'),
```

```
]
```

```
# Жесткие диски
```

```
hard_drives = [
```

```
    HardDrive(1, 'HDD 1TB', 830, 1),
```

```
    HardDrive(2, 'SSD 500GB', 440, 2),
```

```
    HardDrive(3, 'HDD 2TB', 1050, 1),
```

```
    HardDrive(4, 'SSD 1TB', 240, 3),
```

```
]
```

```
def main():
```

```
    """Основная функция"""
```

```

# Соединение данных один-ко-многим
one_to_many = [(hd.model, hd.remaining, comp.name)
                for comp in computers
                for hd in hard_drives
                if hd.computer_id == comp.id]

print('Задание 1')
res_11 = sorted(one_to_many, key=itemgetter(1))
print(res_11)

print('\nЗадание 2')
res_12_unsorted = []

# Перебираем все компьютеры
for comp in computers:
    # Список жестких дисков компьютера
    comp_hds = list(filter(lambda i: i[2] == comp.name, one_to_many))

    # Если у компьютера есть жесткие диски
    if len(comp_hds) > 0:
        # Место, оставшееся на жестких дисках компьютера
        hd_caps = [remainder for _, remainder, _ in comp_hds]

        # Место, оставшееся на всех жестких дисках компьютера
        hd_caps_sum = sum(hd_caps)

        res_12_unsorted.append((comp.name, hd_caps_sum))

# Сортировка по суммарному свободному месту
res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
print(res_12)

if __name__ == '__main__':
    main()

```

Вывод

Задание 1

```
[('SSD 1TB', 240, 'Компьютер 3'), ('SSD 500GB', 440, 'Компьютер 2'), ('HDD 1TB', 830, 'Компьютер 1'), ('HDD 2TB', 1050, 'Компьютер 1')]
```

Задание 2

```
[('Компьютер 1', 1880), ('Компьютер 2', 440), ('Компьютер 3', 240)]
```

Код в формате скриншотов

```
1 from operator import itemgetter
2
3 class HardDrive:
4     """Жесткий диск"""
5     def __init__(self, id, model, remainder, computer_id):
6         self.id = id
7         self.model = model
8         self.remainder = remainder
9         self.computer_id = computer_id
10
11 class Computer:
12     """Компьютер"""
13     def __init__(self, id, name):
14         self.id = id
15         self.name = name
16
17 # Компьютеры
18 computers = [Computer(1, 'Компьютер 1'),
19             Computer(2, 'Компьютер 2'),
20             Computer(3, 'Компьютер 3'),
21 ]
22
23 # Жесткие диски
24 hard_drives = [
25     HardDrive(1, 'HDD 1TB', 830, 1),
26     HardDrive(2, 'SSD 500GB', 440, 2),
27     HardDrive(3, 'HDD 2TB', 1050, 1),
28     HardDrive(4, 'SSD 1TB', 240, 3),
29 ]
30
31 def main():
32     """Основная функция"""
33     # Соединение данных один-ко-многим
34     one_to_many = [(hd.model, hd.remainder, comp.name)
35                    |   |   |   |   for comp in computers
36                    |   |   |   |   for hd in hard_drives
37                    |   |   |   |   if hd.computer_id == comp.id]
38
39     print('Задание 1')
40     res_11 = sorted(one_to_many, key=itemgetter(1))
41     print(res_11)
42
43     print('\nЗадание 2')
44     res_12_unsorted = []
45     # Перебираем все компьютеры
46     for comp in computers:
47         # Список жестких дисков компьютера
48         comp_hds = list(filter(lambda i: i[2] == comp.name, one_to_many))
49         # Если у компьютера есть жесткие диски
50         if len(comp_hds) > 0:
51             # Место, оставшееся на жестких дисках компьютера
52             hd_caps = [remainder for _, remainder, _ in comp_hds]
53             # Место, оставшееся на всех жестких дисках компьютера
54             hd_caps_sum = sum(hd_caps)
55             res_12_unsorted.append((comp.name, hd_caps_sum))
56     # Сортировка по суммарному свободному месту
57     res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
58     print(res_12)
59
60 if __name__ == '__main__':
61     main()
```