

# Жидков Е.И. ИУ5-53Б 8 Вариант РК№1

## Вариант 8

### Вариант В.

1. «Компьютер» и «Жёсткий диск» связаны соотношением один-ко-многим. Выведите список всех сотрудников, у которых фамилия начинается с буквы «А», и названия их отделов.
2. «Компьютер» и «Жёсткий диск» связаны соотношением один-ко-многим. Выведите список отделов с минимальной зарплатой сотрудников в каждом отделе, отсортированный по минимальной зарплате.
3. «Компьютер» и «Жёсткий диск» связаны соотношением многие-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по сотрудникам, сортировка по отделам произвольная.

## Исходный код:

```
from operator import itemgetter

class Computer:

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

class HDD:

    def __init__(self, id, name, size, id_of_computer):
        self.id = id
        self.name = name
        self.size = size
        self.comp_id = id_of_computer

class HDD_Computer:

    def __init__(self, id_of_computer, id_of_HDD):
        self.id_of_computer = id_of_computer
        self.id_of_HDD = id_of_HDD

# Отделы

Computer = [
    Computer(1, 'Первый компьютер'),
    Computer(2, 'Второй компьютер'),
    Computer(3, 'Третий компьютер'),
```

```

Computer(33, 'Мощный компьютер'),
Computer(11, 'Главный компьютер'),
Computer(22, 'Удаленный компьютер')
]

# Сотрудники
HDD = [
    HDD(1, 'Samsung', 1000, 1),
    HDD(2, 'SEAGATE', 2500, 3),
    HDD(3, 'TOSHIBA', 500, 3),
    HDD(4, 'HITACHI', 5000, 2),
    HDD(5, 'WESTERN DIGITAL', 3000, 3)
]

HDD_Computer = [
    HDD_Computer(1,1),
    HDD_Computer(2,2),
    HDD_Computer(3,3),
    HDD_Computer(3,4),
    HDD_Computer(5,5),

    HDD_Computer(11,1),
    HDD_Computer(22,2),
    HDD_Computer(33,3),
    HDD_Computer(33,4),
    HDD_Computer(33,5)
]

def main():

    one_to_many = [(b.name, b.size, a.name)
                    for a in Computer
                    for b in HDD
                    if b.comp_id == a.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(c.name, d.id_of_computer, d.id_of_HDD)
                          for c in Computer
                          for d in HDD_Computer
                          if c.id == d.id_of_computer]

    many_to_many = [(b.name, b.size, name_of_computer)
                    for name_of_computer, id_of_computer, id_of_HDD in many_to_many_temp
                    for b in HDD if b.id == id_of_HDD]

    print('Задание B1')
    first_res = {}

    for l in HDD:

```

```

    if 'S' == l.name[0]:
        hdd = list((filter(lambda i: i[0] == l.name, one_to_many)))
        l_comp_names = [f[2] for f in hdd]
        first_res[l.name] = l_comp_names

print(first_res)

print("\nЗадание B2")
second_res_unsort = []

for d in Computer:

    hdd = list(filter(lambda i: i[2]==d.name, one_to_many))

    if len(hdd) > 0:
        second_res_unsort.append((d.name, min([a[1] for a in hdd])))
second_res = sorted(second_res_unsort, key=itemgetter(1))
print(second_res)

print("\nЗадание B3")
third_res = sorted(many_to_many, key=itemgetter(0))
print(third_res)

if __name__ == '__main__':
    main()

```

## Вывод программы:

```

егорzhidkov@MacBook-Pro-Egor RK1 % /Library/Frameworks/Python.framework/Versions/3.8/bin/python3 /Users/egorzhidkov/Desktop/ПИП/RK1/test.py
Задание B1
{'Samsung': ['Первый компьютер'], 'SEAGATE': ['Третий компьютер']}

Задание B2
[('Третий компьютер', 500), ('Первый компьютер', 1000), ('Второй компьютер', 5000)]

Задание B3
[('HITACHI', 5000, 'Третий компьютер'), ('HITACHI', 5000, 'Мощный компьютер'), ('SEAGATE', 2500, 'Второй компьютер'), ('SEAGATE', 2500, 'Удаленный компьютер'), ('Samsung', 1000, 'Первый компьютер'), ('Samsung', 1000, 'Главный компьютер'), ('TOSHIBA', 500, 'Третий компьютер'), ('TOSHIBA', 500, 'Мощный компьютер'), ('WESTERN DIGITAL', 3000, 'Мощный компьютер')]
егорzhidkov@MacBook-Pro-Egor RK1 %

```

64-bit 0 0 0 Строка 114, столбец 1 (выбрано 2487) Пробелов: 4 UTF-8 LF Python