

**Отчет по РК1 по дисциплине
«Парадигмы и конструкции языков программирования»**

Вариант запросов В. Вариант предметной области 7 (Микропроцессор - Компьютер).

Вариант В.

1. «Микропроцессор» и «Компьютер» связаны соотношением один-ко-многим. Выведите список всех компьютеров, у которых название начинается с буквы «А», и названия их микропроцессоров.

2. «Микропроцессор» и «Компьютер» связаны соотношением один-ко-многим. Выведите список микропроцессоров с минимальной мощностью компьютера в каждом микропроцессоре, отсортированный по минимальной мощности.

3. «Микропроцессор» и «Компьютер» связаны соотношением многие-ко-многим. Выведите список всех связанных микропроцессоров и компьютеров, отсортированный по компьютерам, сортировка по микропроцессорам произвольная.

Текст программы

```
from operator import itemgetter

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

class comp:
    def __init__(self, id, name, power, id_micr):
        self.id = id
        self.name = name
        self.power = power
        self.id_micr = id_micr

class micr_comp:
    def __init__(self, id_micr, id_comp):
        self.id_comp = id_comp
        self.id_micr = id_micr

micrs = [
    micr(1, "intel"),
    micr(2, 'intel 5'),
    micr(3, 'intel 4'),
    micr(44, 'intel 2'),
    micr(55, 'intel 9'),
]

comps = [
    comp(1, 'mac', 2000, 1),
    comp(2, 'hp', 3000, 2),
    comp(3, 'asus', 5000, 3),
```

```
    comp(4, 'asus 2', 4000, 3),
    comp(5, 'asus 3', 6000, 3),
]
```

```
micrs_comps = [
    micr_comp(1,1),
    micr_comp(2,2),
    micr_comp(3,3),
    micr_comp(44,1),
    micr_comp(55,2),
    micr_comp(3,4),
    micr_comp(3,5),
]
```

```
def main():
    one_to_many = [(c.name, c.power, m.name)
                   for m in micrs
                   for c in comps
                   if c.id_micr == m.id]

    many_to_many_temp = [(m.name, mc.id_micr, mc.id_comp)
                          for m in micrs
                          for mc in micrs_comps
                          if m.id == mc.id_micr]

    many_to_many = [(c.name, c.power, micr_name)
                    for micr_name, id_micr, id_comp in many_to_many_temp
                    for c in comps if c.id == id_comp]
```

```
#задание 1
print('Задание 1:')
for comp_name, comp_power, micr_name in one_to_many:
    if comp_name[0] == 'a':
        print(comp_name, micr_name)
```

```
#задание 2
print('\nЗадание 2:')
micrs_power = [(m_name, c_power)
               for c_name, c_power, m_name in one_to_many]

res = sorted(micrs_power, key=itemgetter(1), reverse=False)
temp = ""

for mc_name, mc_power in res:
    if mc_name == temp:
        continue
    else:
        temp = mc_name
        print(mc_name, mc_power)
```

#задание 3

```
print('\nЗадание 3:')
res_2 = sorted(many_to_many, key=itemgetter(1))
for i in range(0, len(res_2)):
    print(res_2[i])
if __name__ == '__main__':
    main()
```

Результаты выполнения программы

```
C:\Users\UserPC\PycharmProjects\pythonProject6\venv\Scripts\python.exe C:\Users\UserPC\PycharmProjects\pythonProject6\main.py
Задание 1:
asus intel 4
asus 2 intel 4
asus 3 intel 4

Задание 2:
intel 2000
intel 5 3000
intel 4 4000

Задание 3:
('asus', 5000, 'intel 4')
('asus 2', 4000, 'intel 4')
('asus 3', 6000, 'intel 4')
('hp', 3000, 'intel 5')
('hp', 3000, 'intel 9')
('mac', 2000, 'intel')
('mac', 2000, 'intel 2')

Process finished with exit code 0
```