

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

**Запрос 1**

Список всех операционных систем,  
у которых название начинается с буквы «W»,  
и список установленных на них компьютеров  
(связь "один ко многим")

**Запрос 2**

Список операционных систем с максимальной ценой компьютеров,  
отсортированный по цене (связь "один ко многим")

**Запрос 3**

Список всех связанных компьютеров и операционных систем,  
отсортированный по операционным системам (связь "многие ко многим")

**Текст программы:**

```
from operator import itemgetter

class Computer:
    """Компьютер"""

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

class OperatingSystem:
    """Операционная система"""

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

class ComputersOS:
    """
    'Компьютеры с ОС' для реализации
    связи многие-ко-многим
    """
```

```

"""

def __init__(self, os_id, comp_id):
    self.os_id = os_id
    self.comp_id = comp_id

# Операционные системы
oses = [
    OperatingSystem(1, 'Windows'),
    OperatingSystem(2, 'Linux'),
    OperatingSystem(3, 'macOS'),
    OperatingSystem(11, 'Windows Server'),
    OperatingSystem(22, 'Ubuntu'),
]

# Компьютеры
computers = [
    Computer(1, 'PC1', 1000, 1),
    Computer(2, 'PC2', 1200, 1),
    Computer(3, 'PC3', 900, 11),
    Computer(4, 'PC4', 1500, 3),
    Computer(5, 'PC5', 1100, 2),
]

# Связь многие-ко-многим (Компьютеры и ОС)
computers_os = [
    ComputersOS(1, 1),
    ComputersOS(1, 2),
    ComputersOS(2, 3),
    ComputersOS(2, 5),
    ComputersOS(3, 4),
    ComputersOS(11, 1),
    ComputersOS(22, 3),
]

def main():
    """Основная функция"""

    # Соединение данных один-ко-многим
    one_to_many = [(c.name, c.price, os.name)
                    for os in oses
                    for c in computers
                    if c.os_id == os.id]

```

```

# Соединение данных многие-ко-многим
many_to_many_temp = [(os.name, co.os_id,
co.comp_id)

                        for os in oses
                        for co in computers_os
                        if os.id == co.os_id]

many_to_many = [(c.name, c.price, os_name)
                  for os_name, os_id, comp_id in
many_to_many_temp
                  for c in computers if c.id ==
comp_id]

"""
Задание 1
Список всех операционных систем,
у которых название начинается с буквы «W»,
и список установленных на них компьютеров
(связь "один ко многим")
"""

print("Задание Г1")
# Выбираем ОС, название которых начинается с буквы
"W"

selected_os = [os for os in oses if
os.name.startswith('W')]

# Для каждой ОС находим компьютеры
result_1 = {}
for os in selected_os:
    result_1[os.name] = [comp.name for comp in
computers if comp.os_id == os.id]

print(result_1)

"""
Задание 2
Список операционных систем с максимальной ценой
компьютеров,
отсортированный по цене (связь "один ко многим")
"""

print("\nЗадание Г2")
res_2_unsorted = []
# Перебираем все ОС

```

```

    for os in oses:
        # Список компьютеров с данной ОС
        os_comps = list(filter(lambda i: i[2] ==
os.name, one_to_many))
        # Если есть компьютеры с этой ОС
        if len(os_comps) > 0:
            # Цены компьютеров с этой ОС
            os_prices = [price for _, price, _ in
os_comps]
            # Максимальная цена
            os_max_price = max(os_prices)
            res_2_unsorted.append((os.name,
os_max_price))

        # Сортировка по максимальной цене
        res_2 = sorted(res_2_unsorted, key=itemgetter(1),
reverse=True)
        print(res_2)

    """
    Задание 3
    Список всех связанных компьютеров и операционных
систем,
    отсортированный по операционным системам (связь
"многие ко многим")
    """

    # Создаем словарь для связи ОС с компьютерами
    print("\nЗадание Г3")
    result_3 = {}
    for os in oses:
        # Находим компьютеры, связанные с этой ОС
        result_3[os.name] = [comp.name for relation in
computers_os for comp in computers if
relation.os_id == os.id
and relation.comp_id == comp.id]

    # Сортировка по имени ОС
    sorted_result_3 = {os: result_3[os] for os in
sorted(result_3.keys())}

    print(sorted_result_3)

if __name__ == '__main__':
    main()

```

## **Результаты выполнения программы:**

### **Задание Г1**

**{'Windows': ['PC1', 'PC2'], 'Windows Server': ['PC3']}**

### **Задание Г2**

**[('macOS', 1500), ('Windows', 1200), ('Linux', 1100), ('Windows Server', 900)]**

### **Задание Г3**

**{'Linux': ['PC3', 'PC5'], 'Ubuntu': ['PC3'], 'Windows': ['PC1', 'PC2'],  
'Windows Server': ['PC1'], 'macOS': ['PC4']}**

**Process finished with exit code 0**