

**Московский государственный технический
университет им. Н.Э. Баумана**

**Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»**

Курс «Казовые компоненты интернет технологий»

Рубежный контроль №1

Выполнил:
студент группы ИУ5-33Б:
Лебедева С.К.

Руководитель:
преподаватель каф. ИУ5
Гапанюк Ю.Е.

Москва, 2022 г.

Текст программы:

```
# используется для сортировки
from operator import itemgetter

class OperatingSystem:
    """Операционная система"""
    def __init__(self, id, name, version, bit_depth, comp_id):
        self.id = id
        self.name = name
        self.version = version
        self.bit_depth = bit_depth
        self.comp_id = comp_id

class Computer:
    """Компьютер"""
    def __init__(self, id, processor):
        self.id = id
        self.processor = processor
        #self.manufacturer = manufacturer

class OSComp:
    """Операционные системы в компьютерах"""
    def __init__(self, comp_id, os_id):
        self.comp_id = comp_id
        self.os_id = os_id

#Компьютеры
Computers = [
    Computer(1, 'Intel Core i7-1265'),
    Computer(2, 'AMD Ryzen 5 7600'),
    Computer(3, 'Intel Core i5-1235'),
    #Для связи М<->М
    Computer(11, 'Intel Core i5-12500'),
    Computer(22, 'Intel Core i3-10300 '),
    Computer(33, 'AMD Ryzen 3 2500')
]

#Операционные системы
OperatingSystems = [
    OperatingSystem(1, 'Microsoft Windows', '8', 16, 1),
    OperatingSystem(2, 'Microsoft Windows', '11', 64, 2),
    OperatingSystem(3, 'Linux', 'Oracle', 16, 3),
    OperatingSystem(4, 'Linux', 'Miracle', 32, 3),
    OperatingSystem(5, 'macOS', 'Mac Ventura', 64, 1)
]

#Операционные системы в компьютерах
OSComps = [
    OSComp(1, 1),
    OSComp(2, 2),
    OSComp(2, 3),
    OSComp(3, 4),
    OSComp(3, 5),
```

```

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

def task1(one_to_many):
    """
    «Компьютер» 1<->М «Операционная система».
    Вывести список всех ОС, начинающихся с буквы «L»
    и названия процессоров компьютеров, на которых
    они установлены.
    """
    for os in OperatingSystems:
        res_1 = list(filter(lambda i: i[0][0] == 'L', one_to_many))
    print(res_1)

def task2(one_to_many):
    """
    «Компьютер» 1<->М «Операционная система».
    Вывести список названий процессоров компьютеров
    с минимальной разрядностью ОС этих компьютеров,
    отсортированный по минимальной разрядности ОС.
    """
    res_min_unsorted = []

    #Перебираем все компьютеры
    """
    Другое решение
    min_bit_depth = []
    for c in Computers:
        c_os_bit_depth = list(filter(lambda i: i[3] == c.processor,
one_to_many))
        #Если на компьютере есть ОС
        min_bit_depth = []
        if len(c_os_bit_depth) > 0:
            min_bit_depth.append(min(c_os_bit_depth, key = itemgetter(2)))
            #res_min_unsorted.append(min(c_os_bit_depth, key = itemgetter(2)))
            res_min_unsorted.append((c.processor, min_bit_depth[0][2]))
            res_2 = sorted(res_min_unsorted, key = itemgetter(1))
        for r in res_2:
            print(r)
    """
    for c in Computers:
        c_os_bit_depth = [(c.processor, os_bit_depth) for os_name, os_version,
os_bit_depth, c_processor in one_to_many if c_processor == c.processor]
        #Если на компьютере есть ОС
        if len(c_os_bit_depth) > 0:
            res_min_unsorted.append(min(c_os_bit_depth, key = itemgetter(1)))
    res_2 = sorted(res_min_unsorted, key = itemgetter(1))
    for r in res_2:
        print(r)

```

```

def task3(many_to_many):
    """
    «Компьютер» М<->М «Операционная система».
    Вывести список всех связанных операционных систем
    и компьютеров, отсортированный по разрядности ОС,
    сортировка по компьютерам произвольная.
    """
    res_3 = sorted(many_to_many, key = itemgetter(2))
    for r in res_3:
        print(r)
def main():
    """Основная функция"""

    # Соединение данных 1<->М
    one_to_many = [(os.name, os.version, os.bit_depth, c.processor)
                    for c in Computers
                    for os in OperatingSystems
                    if os.comp_id == c.id]

    # Соединение данных М<->М
    many_to_many_temp = [(c.processor, osc.comp_id, osc.os_id)
                          for c in Computers
                          for osc in OSComps
                          if c.id == osc.comp_id]

    many_to_many = [(os.name, os.version, os.bit_depth, osc_name)
                     for osc_name, c_id, os_id in many_to_many_temp
                     for os in OperatingSystems if os.id == os_id]

    print('Задание B1')
    task1(one_to_many)
    print('\nЗадание B2')
    task2(one_to_many)
    print('\nЗадание B3')
    task3(many_to_many)

if __name__ == '__main__':
    main()

```

Результаты выполнения:

```
C:\Users\sophi\AppData\Local\Programs\Python\Python36-32\python.exe
Задание B1
[('Linux', 'Oracle', 16, 'Intel Core i5-1235'), ('Linux', 'Miracle', 32, 'Intel Core i5-1235')]

Задание B2
('Intel Core i7-1265', 16)
('Intel Core i5-1235', 16)
('AMD Ryzen 5 7600', 64)

Задание B3
('Microsoft Windows', '8', 16, 'Intel Core i7-1265')
('Linux', 'Oracle', 16, 'AMD Ryzen 5 7600')
('Linux', 'Oracle', 16, 'Intel Core i5-12500')
('Microsoft Windows', '8', 16, 'Intel Core i3-10300 ')
('Linux', 'Miracle', 32, 'Intel Core i5-1235')
('Linux', 'Miracle', 32, 'Intel Core i3-10300 ')
('Microsoft Windows', '11', 64, 'AMD Ryzen 5 7600')
('macOS', 'Mac Ventura', 64, 'Intel Core i5-1235')
('Microsoft Windows', '11', 64, 'Intel Core i5-12500')
('macOS', 'Mac Ventura', 64, 'AMD Ryzen 3 2500')
Press any key to continue . . .
```