



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ОТЧЕТ
По лабораторной работе №1
По дисциплине «Парадигмы и конструкции
языков программирования»
«Вариант 9»

Выполнил студент: Денефельд Валерия Максимовна
фамилия, имя, отчество

Группа: ИБМЗ-34Б

Преподаватель

Гапанюк Ю.Е.

подпись, дата

2024 г.

Код программы:

```
from operator import itemgetter

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

class Computer:
    """Класс Компьютер"""
    def __init__(self, id, model, os_id):
        self.id = id
        self.model = model
        self.os_id = os_id

class ComputerOS:
    """
    Класс для связи многие-ко-многим между компьютерами и операционными
    системами
    """
    def __init__(self, os_id, computer_id):
        self.os_id = os_id
        self.computer_id = computer_id

# Список операционных систем
operating_systems = [
    OperatingSystem(1, "Windows 10"),
    OperatingSystem(2, "Ubuntu"),
    OperatingSystem(3, "macOS"),
]

# Список компьютеров
computers = [
    Computer(1, "Dell XPS", 1),
    Computer(2, "MacBook Pro", 3),
    Computer(3, "HP Spectre", 1),
    Computer(4, "ThinkPad", 2),
    Computer(5, "Mac Mini", 3),
]

# Связь многие-ко-многим между компьютерами и операционными системами
computers_os = [
    ComputerOS(1, 1),
    ComputerOS(1, 3),
    ComputerOS(2, 4),
    ComputerOS(3, 2),
    ComputerOS(3, 5),
]

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

    # Связь один-ко-многим: Операционная система -> Компьютер
    one_to_many = [(c.model, o.name)
                    for o in operating_systems
                    for c in computers
                    if c.os_id == o.id]

    # Связь многие-ко-многим: Операционная система -> Компьютеры
    many_to_many_temp = [(o.name, co.os_id, co.computer_id)
                          for o in operating_systems
                          for co in computers_os]
```

```

        if o.id == co.os_id]

many_to_many = [(c.model, os_name)
                 for os_name, os_id, computer_id in many_to_many_temp
                 for c in computers
                 if c.id == computer_id]

# Запрос 1: Все операционные системы и связанные с ними компьютеры,
отсортированные по ОС
print("Задание 1:")
res_1 = sorted(one_to_many, key=itemgetter(1))
print(res_1)

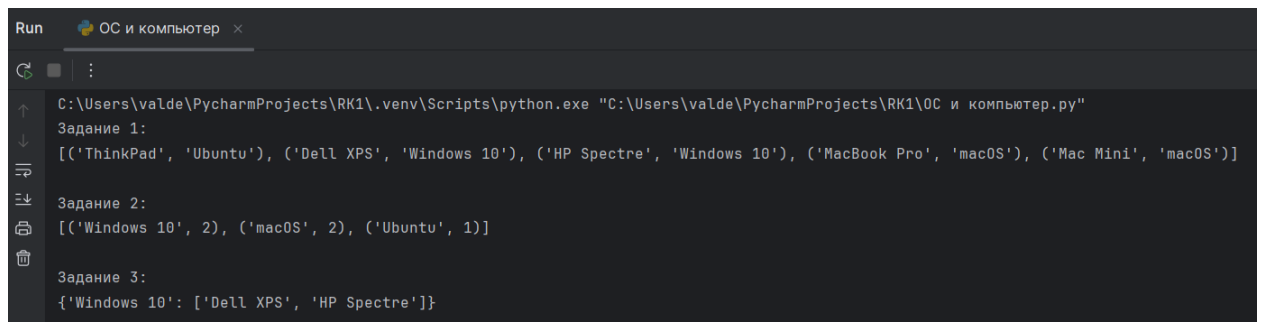
# Запрос 2: Количество компьютеров для каждой операционной системы
print("\nЗадание 2:")
res_2 = []
for o in operating_systems:
    # Список компьютеров для данной ОС
    o_computers = list(filter(lambda i: i[1] == o.name, one_to_many))
    # Количество компьютеров
    res_2.append((o.name, len(o_computers)))
res_2_sorted = sorted(res_2, key=itemgetter(1), reverse=True)
print(res_2_sorted)

# Запрос 3: Все операционные системы, содержащие слово "Windows", и их
компьютеры
print("\nЗадание 3:")
res_3 = {}
for o in operating_systems:
    if "Windows" in o.name:
        o_computers = list(filter(lambda i: i[1] == o.name,
many_to_many))
        o_computers_names = [x[0] for x in o_computers]
        res_3[o.name] = o_computers_names
print(res_3)

if __name__ == "__main__":
    main()

```

Вывод программы:



```

Run  ОС и компьютер  ×
C:\Users\valde\PycharmProjects\RK1\.venv\Scripts\python.exe "C:\Users\valde\PycharmProjects\RK1\ОС и компьютер.py"
Задание 1:
[('ThinkPad', 'Ubuntu'), ('Dell XPS', 'Windows 10'), ('HP Spectre', 'Windows 10'), ('MacBook Pro', 'macOS'), ('Mac Mini', 'macOS')]
Задание 2:
[('Windows 10', 2), ('macOS', 2), ('Ubuntu', 1)]
Задание 3:
{'Windows 10': ['Dell XPS', 'HP Spectre']}

```