

МГТУ им. Н.Э. Баумана

Отчёт по рубежному контролю №1
по курсу «Базовые компоненты и интернет-технологии»
Вариант 17.

Преподаватель
Гапанюк Ю.Е.
29.10.2022

Студент группы ИУ5-34Б
Лавренов М.А.
29.10.2022

2022 г.

Полученное задание:

Необходимо создать два класса данных в соответствии с Вашим вариантом предметной области, которые связаны отношениями один-ко-многим и многие-ко-многим.

Необходимо создать списки объектов классов, содержащих тестовые данные (3-5 записей), таким образом, чтобы первичные и вторичные ключи соответствующих записей были связаны по идентификаторам.

Необходимо разработать запросы в соответствии с Вашим вариантом.

Предметная область: класс_1 – Операционная система, класс_2 – Компьютер, вариант запросов: Г.

Запросы:

1. «ОС» и «ПК» связаны соотношением один-ко-многим. Выведите список всех ПК, у которых модель начинается с буквы «А», и список установленных на нем ОС.
2. «ОС» и «ПК» связаны соотношением один-ко-многим. Выведите список ПК с максимальным требованием к оперативной памяти ОС, отсортированный по максимальному требуемому объему памяти.
3. «ОС» и «ПК» связаны соотношением многие-ко-многим. Выведите список всех связанных ОС и ПК, отсортированный по моделям ПК, сортировка по ОС произвольная.

Текст программы:

```
from operator import itemgetter

class OS:
    """ОС"""
    def __init__(self, id, name, minram, PC_id):
        self.id = id
        self.name = name
        self.minram = minram
        self.PC_id = PC_id
class PC:
    """ПК"""
    def __init__(self, id, model):
        self.id = id
        self.model = model
class OSPC:
    """
    'Операционные системы компьютеров' для реализации
    СВЯЗИ МНОГИЕ-КО-МНОГИМ
    """
    def __init__(self, OS_id, PC_id):
        self.OS_id = OS_id
        self.PC_id = PC_id

# ОС
OSs = [
    OS(1, 'Ubuntu', 150000, 1),
    OS(2, 'Kali', 2800, 2),
    OS(3, 'MacOS', 100000, 2),
    OS(4, 'Windows 7', 86400, 3),
    OS(5, 'Windows Wista', 44444, 4)
]
# Компьютеры
PCs = [
    PC(1, 'Lenovo a123'),
    PC(2, 'Abobus M1'),
    PC(3, 'qwertyijk'),
    PC(4, 'HYPERPC OVERPRICE 2.0')
]
OSs_PCs = [
    OSPC(1, 1),
    OSPC(2, 2),
    OSPC(3, 3),
    OSPC(4, 4),
    OSPC(5, 1),
    OSPC(5, 1),
    OSPC(2, 3),
    OSPC(3, 2),
    OSPC(4, 1),
    OSPC(5, 4)
]

def main():
    # Соединение данных ОДИН-КО-МНОГИМ
    one_to_many = [(OS.name, OS.minram, PC.model)
                    for PC in PCs
                    for OS in OSs
                    if OS.PC_id == PC.id]
    # Соединение данных МНОГИЕ-КО-МНОГИМ
```

```

many_to_many_temp = [(PC.model, OS.PC_id, OS.OS_id)
                      for PC in PCs
                      for OS in OSs_PCs
                      if PC.id == OS.PC_id]
many_to_many = [(OS.name, OS.minram, PC_name)
                 for PC_name, PC_id, OS_id in many_to_many_temp
                 for OS in OSs if OS.id == OS_id]

print('\nЗадание Г1')
t_1 = {}
for PC in PCs:
    if PC.model[0] == 'A':
        o_emps = list(filter(lambda i: i[2] == PC.model, one_to_many))
o_emps_names = [x for x, _, _ in o_emps]
t_1[o_emps[0][2]] = o_emps_names
print(t_1)

print('\nЗадание Г2')
t_2_unsorted = []
for PC in PCs:
    PC_OSs = list(filter(lambda i: i[2] == PC.model, one_to_many))
    if len(PC_OSs) > 0:
        PC_minrams = [minram for _, minram, _ in PC_OSs]
        PC_minrams_sum = max(PC_minrams)
        t_2_unsorted.append((PC.model, PC_minrams_sum))
t_2 = sorted(t_2_unsorted, key=itemgetter(1), reverse=True)
print(t_2)

print('\nЗадание Г3')
t_3 = sorted(many_to_many, key=itemgetter(2))
for i in t_3:
    print(i)

if __name__ == '__main__':
    main()

```

Результаты выполнения:

```

Задание Г1
{'Abobus M1': ['Kali', 'MacOS']}

Задание Г2
[('Lenovo a123', 150000), ('Abobus M1', 100000), ('qwertyijk', 86400), ('HYPERPC OVERPRICE 2.0', 44444)]

Задание Г3
('Kali', 2800, 'Abobus M1')
('MacOS', 100000, 'Abobus M1')
('Windows 7', 86400, 'HYPERPC OVERPRICE 2.0')
('Windows Wista', 44444, 'HYPERPC OVERPRICE 2.0')
('Ubuntu', 150000, 'Lenovo a123')
('Windows Wista', 44444, 'Lenovo a123')
('Windows Wista', 44444, 'Lenovo a123')
('Windows 7', 86400, 'Lenovo a123')
('MacOS', 100000, 'qwertyijk')
('Kali', 2800, 'qwertyijk')

```