

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

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

Курс «Парадигмы и конструкции языков программирования»

Отчет по рубежному контролю No1
Вар. Г-8

Выполнил:

студент группы ИУ5-
34Б

Масловский Василий

Проверил:

преподаватель каф.
ИУ5

Нардид А.Н.

Москва, 2024 г.

Код программы:

```
class HardDrive:
    def __init__(self, id: int, model: str, capacity: int, computer_id: int):
        self.id = id
        self.model = model
        self.capacity = capacity
        self.computer_id = computer_id

class Computer:
    def __init__(self, id: int, name: str):
        self.id = id
        self.name = name

class ComputerHardDrive:
    # Многое ко многим
    def __init__(self, computer_id: int, hard_drive_id: int):
        self.computer_id = computer_id
        self.hard_drive_id = hard_drive_id

# Список компьютеров
computers = [
    Computer(1, 'AlphaPC'),
    Computer(2, 'BetaPC'),
    Computer(3, 'GammaPC')
]

# Список жестких дисков
hard_drives = [
    HardDrive(1, 'Seagate', 1000, 1),
    HardDrive(2, 'WD Blue', 2000, 1),
    HardDrive(3, 'Samsung EVO', 500, 2),
    HardDrive(4, 'Toshiba', 750, 3),
    HardDrive(5, 'Hitachi', 1500, 3)
]

computer_hard_drives = [
    ComputerHardDrive(1, 1),
    ComputerHardDrive(1, 2),
    ComputerHardDrive(2, 3),
    ComputerHardDrive(3, 4),
    ComputerHardDrive(3, 5)
]

def main():
    # Создание кортежей 1:M
    one_to_many = [(hd.model, hd.capacity, comp.name)
                    for comp in computers
                    for hd in hard_drives
                    if hd.computer_id == comp.id]

    # Создание кортежей M:M
    many_to_many_temp = [(comp.name, elem.computer_id, elem.hard_drive_id)
                          for comp in computers
                          for elem in computer_hard_drives
                          if comp.id == elem.computer_id]

    many_to_many = [(hd.model, hd.capacity, comp_name)
                     for comp_name, computer_id, hard_drive_id in
                     many_to_many_temp]
```

```

        for hd in hard_drives if hd.id == hard_drive_id]

print('№1') # Компьютеры начинающиеся на "A"
res1_temp = list(filter(lambda x: x[2].startswith('A'), one_to_many))
res1 = [(model, comp) for model, _, comp in res1_temp]
print(*[': '.join(ans) for ans in res1], sep='\n')

print('№2') # максимальная емкость жестких дисков у каждого компьютера
res2_unsorted = []
for comp in computers:
    res_2_temp = list(filter(lambda x: x[2] == comp.name, one_to_many))
    if len(res_2_temp) > 0:
        drive_capacities = [capacity for _, capacity, _ in res_2_temp]
        max_capacity = max(drive_capacities)
        res2_unsorted.append((comp.name, max_capacity))

res2 = sorted(res2_unsorted, key=lambda x: x[1], reverse=True)
print(*[': '.join(list(map(str, ans))) for ans in res2], sep='\n')

print('№3') # сортируем по названию компьютера
res_3_temp = sorted(many_to_many, key=lambda x: x[2])
res_3 = [(model, comp) for model, _, comp in res_3_temp]
print(*[': '.join(ans) for ans in res_3], sep='\n')

if __name__ == '__main__':
    main()

```

Вывод:

```

№1
Seagate: AlphaPC
WD Blue: AlphaPC
№2
AlphaPC: 2000
GammaPC: 1500
BetaPC: 500
№3
Seagate: AlphaPC
WD Blue: AlphaPC
Samsung EVO: BetaPC
Toshiba: GammaPC
Hitachi: GammaPC

```