Защищено:
Гапанюк Ю.Е.

"__"_____2024 г.

Демонстрация:
Гапанюк Ю.Е.

"__"_____2024 г.

Отчет по рубежному контролю №1 по курсу Парадигмы и конструкции языков программирования

Вариант запросов: В
Вариант предметной области: 20

5
(количество листов)

ИСПОЛНИТЕЛЬ:

студент группы ИУ5Ц-53Б _____
(подпись)

Ахмеров Д.И.

"7" ноября 2024 г.

Москва, МГТУ – 2024

Вариант запросов: В. Вариант предметной области: 20

1. «Поставщик» и «Деталь» связаны соотношением один-ко-многим. Выведите список всех деталей, у которых название начинается с буквы «Т», и названия их поставщиков.
2. «Поставщик» и «Деталь» связаны соотношением один-ко-многим. Выведите список поставщиков с минимальной ценой их деталей, отсортированный по минимальной цене.
3. «Поставщик» и «Деталь» связаны соотношением многие-ко-многим. Выведите список всех связанных деталей и поставщиков, отсортированный по деталям, сортировка по поставщикам произвольная.

Текст программы

```
from operator import itemgetter


class Detail:
    def __init__(self, id, name, price, id_supplier):
        self.id = id
        self.name = name
        self.price = price
        self.id_supplier = id_supplier


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


class DetailSupplier:
    def __init__(self, id_detail, id_supplier):
        self.id_detail = id_detail
        self.id_supplier = id_supplier


details = [
    Detail(1, 'Тормозная колодка', 1000, 1),
```

```
    Detail(2, 'Свеча зажигания', 4000, 3),
    Detail(4, 'Решетка радиатора', 8000, 2),
    Detail(3, 'Подшипник', 2500, 2),
    Detail(5, 'Топливный бак', 5000, 3),
]

suppliers = [
    Supplier(1, 'Автотрейд'),
    Supplier(2, 'Берг'),
    Supplier(3, 'Автопитер')
]

details_suppliers = [
    DetailSupplier(1, 1),
    DetailSupplier(2, 3),
    DetailSupplier(4, 2),
    DetailSupplier(3, 2),
    DetailSupplier(5, 3),
]

def main():
    one_to_many = [(d.name, d.price, s.name)
        for s in suppliers
        for d in details
        if d.id_supplier == s.id
    ]

    many_to_many_temp = [(s.name, ds.id_supplier, ds.id_detail)
        for s in suppliers
```

```
    for ds in details_suppliers
    if s.id == ds.id_supplier
]

many_to_many = [(d.name, d.price, supplier_name)
    for supplier_name, id_supplier, id_detail in many_to_many_temp
    for d in details if d.id == id_detail
]

print('Задание B1')
res_11 = sorted(
    [(name, supplier) for name, _, supplier in one_to_many if name.startswith('T')],
    key=itemgetter(1)
)
print(res_11)

print('\nЗадание B2')
res_12_unsorted = []
for s in suppliers:
    s_details = list(filter(lambda i: i[2] == s.name, one_to_many))
    if s_details:
        min_price = min([price for _, price, _ in s_details])
        res_12_unsorted.append((s.name, min_price))

res_12 = sorted(res_12_unsorted, key=itemgetter(1))
print (res_12)

print('\nЗадание B3')
res_13 = {}
```

```
    for s in suppliers:
        s_details = list(filter(lambda i: i[2] == s.name, many_to_many))
        s_details_names = [x for x, _, _ in s_details]
        res_13[s.name] = sorted(s_details_names)
    print (res_13)


if __name__ == '__main__':
    main()
```

Результаты выполнения

Задание B1

[('Топливный бак', 'Автопитер'), ('Тормозная колодка', 'Автотрейд')]

Задание B2

[('Автотрейд', 1000), ('Берг', 2500), ('Автопитер', 4000)]

Задание B3

{'Автотрейд': ['Тормозная колодка'], 'Берг': ['Подшипник', 'Решетка радиатора'], 'Автопитер': ['Свеча зажигания', 'Топливный бак']}