

РК № 1

Разработка интернет приложений

Вариант 20Б

Студент: Хизриев А.А.

Группа: ИУ5-52Б

Проверил: Гапанюк Ю.Е.

## Описание задания:

- 1) Необходимо создать два класса данных в соответствии с Вашим вариантом предметной области, которые связаны отношениями один-ко-многим и многие-ко-многим.
- 2) Необходимо создать списки объектов классов, содержащих тестовые данные (3-5 записей), таким образом, чтобы первичные и вторичные ключи соответствующих записей были связаны по идентификаторам.
- 3) Необходимо разработать запросы в соответствии с Вашим вариантом. Запросы сформулированы в терминах классов «Сотрудник» и «Отдел», которые используются в примере. Вам нужно перенести эти требования в Ваш вариант предметной области. При разработке запросов необходимо по возможности использовать функциональные возможности языка Python (list/dict comprehensions, функции высших порядков).

## Вариант Б.

1. «Деталь» и «Поставщик» связаны соотношением один-ко-многим. Выведите список всех связанных деталей и поставщиков, отсортированный по компьютерам (установленным процессорам), сортировка по классам произвольная.
2. «Деталь» и «Поставщик» связаны соотношением один-ко-многим. Выведите список поставщиков с количеством поставляемых деталей, отсортированный по количеству деталей.
3. «Деталь» и «Поставщик» связаны соотношением многие-ко-многим. Выведите список всех поставщиков деталей, названия которых начинаются на «n» или «l».

## Текст программы

```
from subject_area.detail import Detail
from subject_area.provider import Provider
from subject_area.details_prov import DetailsProv

from operator import itemgetter

providers = [
    Provider(1, "Vista", "Russia"),
    Provider(2, "KROS", "Ukraine"),
    Provider(3, "SKUBA", "Litva"),
]

details = [
    Detail(1, "nut", "copper", 1),
    Detail(2, "washer", "lead", 2),
```

```

    Detail(3, "screw", "aluminum", 3),
    Detail(4, "lens", "glass", 3)
]

details_prov = [
    DetailsProv(1, 1),
    DetailsProv(1, 2),

    DetailsProv(2, 2),
    DetailsProv(2, 3),

    DetailsProv(3, 3),
    DetailsProv(3, 1),

    DetailsProv(4, 3),
    DetailsProv(4, 2)
]

def main():

    one_to_many = [(det.name, prov.name, prov.country)
                    for det in details
                    for prov in providers
                    if det.prov_id == prov.id
                    ]

    many_to_many_temp = [(prov.name, dp.prov_id, dp.detail_id)
                          for prov in providers
                          for dp in details_prov
                          if prov.id == dp.prov_id]

    many_to_many = [(det.name, prov_name)
                     for prov_name, prov_id, detail_id in many_to_many_temp
                     for det in details if det.id == detail_id]

    print('\nЗадание B1')
    task_1 = sorted(one_to_many, key=itemgetter(0))
    print(task_1)

    print('\nЗадание B2')
    task_2 = []
    for prov in providers:
        det_prov = list(filter(lambda elem: elem[1] == prov.name, one_to_many))
        task_2.append([prov.name, len(det_prov)])
    print(sorted(task_2, key=itemgetter(1)))

    print('\nЗадание B3')
    task_3 = {}
    for det in details:
        if (det.name.startswith('n') or det.name.startswith('l')):
            det_prov = list(
                filter(lambda elem: elem[0] == det.name, many_to_many))
            task_3[det.name] = [x for _, x in det_prov]
    print(task_3)

if __name__ == "__main__":
    main()

```

АКТИВНО  
Чтобы  
раздел

```
class Detail:
    def __init__(self, id, name, material, prov_id):
        self.id = id
        self.name = name
        self.material = material
        self.prov_id = prov_id
```

```
class Provider:
    def __init__(self, id, name, country):
        self.id = id
        self.name = name
        self.country = country
```

```
class DetailsProv:
    def __init__(self, detail_id, prov_id):
        self.detail_id = detail_id
        self.prov_id = prov_id
```

Задание 61  
[('lena', 'SKUBA', 'Litva'), ('nut', 'Vista', 'Russia'), ('screw', 'SKUBA', 'Litva'), ('washer', 'KROS', 'Ukraine')]

Задание 62  
[('Vista', 1), ('KROS', 1), ('SKUBA', 2)]

Задание 63  
('nut': ['Vista', 'KROS'], 'lena': ['KROS', 'SKUBA'])

Активация Windows  
Чтобы активировать  
раздел "Параметры".