Московский государственный технический университет им. Н.Э. Баумана
Факультет «Информатика и системы управления»
Кафедра «Системы обработки информации и управления»

Курс «Разработка интернет-приложений»
Отчет по рубежному контролю №1

Выполнил: Студент группы ИУ5-53Б
Борисов Андрей Михайлович

Москва - 2021

Задание

Вариант 6В

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

Пример классов данных для предметной области Сотрудник-Отдел:

1. Класс «Дом», содержащий поля:
   - ID записи о доме;
   - Номер дома;
   - Число этажей;
   - ID записи об улице, которая через него проходит. (для реализации связи один-ко-многим)
2. Класс «Улица», содержащий поля:
   - ID записи об улице;
   - Наименование улицы.
3. (Для реализации связи многие-ко-многим) Класс «Дома улицы», содержащий поля:
   - ID записи об улице;
   - ID записи о доме.

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).

Код программы

Класс House.py:

```
class House:
    """House"""

    def __init__(self, id, number, floor, street_id):
        self.id = id
        self.number = number
        self.floor = floor
        self.street_id = street_id
```

Класс Street.py:

```
class Street:
    """Street"""

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

КлассHouseStreet.py:

```
class HouseStreet:

    def __init__(self, street_id, house_id):
        self.street_id = street_id
        self.house_id = house_id
```

Класс main.py:

```
from entity.House import House
from entity.HouseStreet import HouseStreet
from entity.Street import Street


def main():
    # Улицы
    streets = [
        Street(1, "Бауманская"),
        Street(2, "Никольская"),
        Street(3, "Семеновская"),
        Street(4, "Даниловская"),
    ]

    # Дома
    houses = [
        House(1, 33, 5, 1),
        House(2, 21, 1, 2),
        House(3, 56, 12, 3),
        House(4, 77, 2, 4),
        House(5, 66, 8, 2),
        House(6, 69, 9, 4),
        House(7, 12, 5, 3),
        House(8, 11, 2, 1),
```

```
    ]

    house_street = [
        HouseStreet(1, 1),
        HouseStreet(2, 2),
        HouseStreet(3, 3),
        HouseStreet(3, 1),
        HouseStreet(4, 2),
        HouseStreet(4, 4),
        HouseStreet(2, 3),
        HouseStreet(1, 3),
    ]

    # Связь один ко многим(улица - дом)
    one_to_many = [(h.number, h.floor, s.name)
                   for s in streets
                   for h in houses
                   if h.street_id == s.id]

    # Объединяем streets - house_street
    many_to_many_temp = [(s.name, hs.street_id, hs.house_id)
                         for s in streets
                         for hs in house_street
                         if s.id == hs.street_id]

    # Соединение данных многие ко многим
    many_to_many = [(h.number, h.floor, s_name)
                    for s_name, str_id, h_id in many_to_many_temp
                    for h in houses
                    if h.id == h_id]

    print(many_to_many)

    print('Задание А1')
    # Вывести список всех домов, у которых этажность больше 3, и названия улиц на
которых они расположены

    res_1 = list(filter(lambda i: i[1] > 3, one_to_many))
    print(res_1)

    print('Задание А2')
    # Вывести список улиц с минимальной этажностью дома на каждой улице,
отсортированный по минимальной этажности

    res_2 = []
    for s in streets:
        # список домов улицы
        s_houses = list(filter(lambda i: i[2] == s.name, one_to_many))
        if len(s_houses) > 0:
            res_2.append(min(s_houses, key=lambda i: i[1]))

    # сортиировка по этажности
    res_2 = sorted(res_2, key=lambda i: i[1])
    print(res_2)

    print('Задание А3')
    # Вывести список всех связанных домов и улиц, отсортированный по домам(сортировка
по этажности дома)
    res_3 = sorted(many_to_many, key=lambda i: i[1])
    print(res_3)

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

Результаты работы программы

Задание А1

[(33, 5, 'Бауманская'), (66, 8, 'Никольская'), (56, 12, 'Семеновская'), (12, 5, 'Семеновская'), (69, 9, 'Даниловская')]

Задание А2

[(21, 1, 'Никольская'), (11, 2, 'Бауманская'), (77, 2, 'Даниловская'), (12, 5, 'Семеновская')]

Задание А3

[(21, 1, 'Никольская'), (21, 1, 'Даниловская'), (77, 2, 'Даниловская'), (33, 5, 'Бауманская'), (33, 5, 'Семеновская'), (56, 12, 'Бауманская'), (56, 12, 'Никольская'), (56, 12, 'Семеновская')]

Скриншот результатов: