**Национальный исследовательский технологический университет «МИСИС»**

**Институт Информационных технологий и Компьютерных наук**

**(ИТКН)**

**Курс «Карты Folium»**

**Лабораторная работа №1**

Выполнили: студенты группы БИВТ-22-ИСАД-1

Ильющеня Дмитрий Павлович

Иванюта Михаил Игоревич

Изображение выглядит как черный, темнота

Контент, сгенерированный ИИ, может содержать ошибки.

Проверила: Прокофьева Е.Н.

**Москва, 2025**

**Сервис расчета стоимости недвижимости**

**Основные этапы реализации**

**1. Загрузка и геокодирование данных**

Пользователь загружает файл Excel с данными по квартирам. Далее запускается геокодирование адресов через библиотеку geocoder.\

1. y = geocoder.osm(new\_adress)

2. if y.latlng is None:

3. continue

4.

**2. Занесение данных в базу uploaded\_files**

Каждая строка Excel-файла превращается в объект с характеристиками:

* тип недвижимости (estate\_type);
* этаж/этажность;
* наличие балкона;
* удаленность от метро и пр.

1. u\_db.create(address=new\_adress, rooms\_count=rooms, estate\_type=...,

2. coordinates\_lat=y.latlng[0], coordinates\_lng=y.latlng[1])

3.

**3. Создание карты с помощью folium**

Основная карта создается в функции index, по центру — Москва:

1. m = folium.Map(location=[55.75, 37.62], height='100%', width='100%')

2.

Добавлены тайлы разных стилей (темная, светлая и контрастная), слои районов Москвы (moscow.geojson), а также плагины:

* MiniMap;
* Fullscreen;
* HeatMap (тепловая карта по цене);
* MarkerCluster для группировки объектов.

**4. Отображение объектов недвижимости**

**Старый и новый фонд** визуализируется с разной окраской иконок:

1. marker = folium.Marker([lat, lng], icon=DivIcon(...), popup=f"{...}")

2. cluster\_old.add\_child(marker)

3.

Также отображаются загруженные пользователем объекты (cluster\_uploaded).

**5. Расчет стоимости объекта по аналогам**

Для каждой загруженной квартиры рассчитывается средняя цена на основе найденных аналогов в пределах 3 км:

1. dist = geopy.distance.geodesic(coords, target)

2. if dist <= 3:

3. fitable\_obj\_list.append(g)

4.

Используется метод **PriceCalculation** для оценки:

1. a = PriceCalculation(main\_object, dict\_of\_appartments[key])

2.

**6. Отображение рассчитанных объектов**

На карту добавляются маркеры с подсказками и рассчитанной ценой:

1. upl\_marker = folium.Marker([...], popup=f"... ЦЕНА:{final\_appartment\_price}")

2. cluster\_uploaded.add\_child(upl\_marker)

3.

**Полученный результат**

* Реализована полная цепочка обработки пользовательских данных.
* Карта с кластерами и тепловой картой по ценам недвижимости.
* Механизм оценки стоимости новых квартир по аналогам с использованием геопозиции.
* Автоматическое геокодирование и сохранение данных.