

**Разработка интерактивной геоинформационной системы
для визуализации пространственных данных с использованием
библиотеки Folium**

Гиренко Александр Евгеньевич

Приамурский государственный университет им.Шолом-Алейхема

Студент

Аннотация

В статье рассматривается процесс разработки программного модуля для интерактивной визуализации геопространственных данных с использованием языка Python и библиотеки Folium. На примере набора данных о высших учебных заведениях Дальневосточного федерального округа продемонстрированы алгоритмы автоматического центрирования карты, кластеризации маркеров и генерации HTML-попапов с атрибутами объектов. Разработанное решение обеспечивает высокую скорость рендеринга и интуитивно понятный пользовательский интерфейс, что делает его применимым для задач регионального планирования и образовательного мониторинга.

Ключевые слова: геоинформационные системы, визуализация данных, Python, Folium, пространственный анализ, кластеризация маркеров, интерактивная карта.

**Development of an interactive geographic information system for spatial data
visualization using the Folium library**

Girenko Aleksandr Evgen'evich

Sholom-Aleichem Priamursky State University

Student

Abstract

The article considers the process of developing a software module for interactive visualization of geospatial data using the Python programming language and the Folium library. Using the example of a dataset on higher education institutions of the Far Eastern Federal District, algorithms for automatic map centering, marker clustering, and generation of HTML popups with object attributes are demonstrated. The developed solution provides high rendering speed and an intuitive user interface, making it applicable for regional planning and educational monitoring tasks.

Keywords: geographic information systems, data visualization, Python, Folium, spatial analysis, marker clustering, interactive map.

1 Введение

1.1 Актуальность

В условиях цифровой трансформации образования и регионального развития возрастает потребность в инструментах визуализации пространственных данных. Геоинформационные системы (ГИС) позволяют не только отображать объекты на карте, но и проводить пространственный анализ, выявлять закономерности распределения и принимать обоснованные решения на основе визуализированной информации.

Традиционные десктопные ГИС-решения (ArcGIS, QGIS) обладают широким функционалом, однако требуют значительных временных затрат на освоение и не всегда подходят для быстрой публикации интерактивных карт в веб-среде. Библиотека Folium для Python представляет собой эффективную альтернативу, позволяющую создавать интерактивные карты на базе JavaScript-библиотеки Leaflet.js с минимальными усилиями по разработке.

Актуальность данного исследования обусловлена необходимостью создания доступных инструментов для визуализации образовательной инфраструктуры Дальневосточного федерального округа (ДФО), что может быть использовано для анализа доступности высшего образования, планирования приемной кампании и мониторинга регионального развития.

1.2 Обзор исследований

В работе Р.А. Бурнашева и А.Р. Габитуллина представлена интеллектуальная геоинформационная система для диалектологических исследований, объединяющая машинное обучение, базу знаний, графовую базу данных и нечеткую логику для обработки лингвистических данных с пространственной привязкой [1]. К.В. Аксёнова с соавторами анализируют методы и технологии визуализации пространственных данных в веб-приложениях, акцентируя внимание на интерактивных картах, API и адаптивности, что имеет значение для создания современных геосервисов [2]. А.М. Гельфанд и А.В. Красов демонстрируют применение модели SARIMAX в геоинформационной системе оценки рисков на объектах критической информационной инфраструктуры, показывая возможности временных рядов для прогнозирования угроз [3]. В учебном пособии Н.А. Кащенко, Е.В. Попова, А.В. Чечина систематизированы базовые принципы построения геоинформационных систем, включая работу с растровыми изображениями, оцифровку документов и картографические технологии, что служит теоретической основой для прикладных разработок [4]. Н.В. Остудин, А.В. Матвеев и Е.С. Савинова исследуют нейросетевую обработку данных космического мониторинга для поддержки принятия решений и прогнозирования чрезвычайных ситуаций, подтверждая эффективность искусственного интеллекта в антикризисном управлении [5]. А. Портер в своем исследовании рассматривает создание хороплетных карт с использованием библиотек Python и Folium, демонстрируя возможности визуализации пространственных данных в веб-среде [6].

1.3 Цель исследования

Целью работы является разработка программного модуля для интерактивной визуализации геопространственных данных о высших учебных заведениях Дальневосточного федерального округа с использованием библиотеки Folium, включающего алгоритмы автоматического расчета центроида карты, кластеризации маркеров и генерации структурированных атрибутивных данных.

2 Материалы и методы

2.1 Описание набора данных

Для проведения исследования был сформирован набор данных, содержащий информацию о 8 высших учебных заведениях Дальневосточного федерального округа. Данные включают следующие атрибуты:

1. Название — полное наименование образовательной организации
2. Город — населенный пункт расположения
3. Широта (Lat) — географическая координата в десятичном формате (WGS 84)
4. Долгота (Lon) — географическая координата в десятичном формате (WGS 84)

Фрагмент набора данных представлен в таблице 1.

Таблица 1 – Фрагмент набора данных о вузах ДФО

Название	Город	Широта	Долгота
Приамурский государственный университет им. Шолом-Алейхема	Биробиджан	48.7946	132.9211
Дальневосточный федеральный университет	Владивосток	43.2220	132.0300
Амурский государственный университет	Благовещенск	50.2620	127.5330

2.2 Инструментарий разработки

Для реализации поставленной задачи использовались следующие программные средства:

1. Язык программирования Python 3.10 — основная среда разработки
2. Библиотека pandas 1.5.3 — для структурирования и обработки табличных данных
3. Библиотека folium 0.14.0 — для генерации интерактивных карт
4. Плагин MarkerCluster — для оптимизации отображения большого количества маркеров

Выбор библиотеки Folium обусловлен следующими преимуществами:

1. Интеграция с JavaScript-библиотекой Leaflet.js, обеспечивающей высокую производительность рендеринга
2. Поддержка различных картографических подложек (OpenStreetMap, CartoDB, Stamen)
3. Встроенные механизмы кластеризации и генерации всплывающих окон
4. Возможность экспорта результата в автономный HTML-файл

2.3 Методология анализа

2.3.1 Алгоритм автоматического центрирования карты

Для оптимального отображения всех объектов на карте был реализован алгоритм расчета центроида выборки. Центроид определяется как среднее арифметическое координат всех объектов (рис.1).

$$\begin{aligned} Lat_{center} &= \frac{1}{N} \sum_{i=1}^N Lat_i \\ Lon_{center} &= \frac{1}{N} \sum_{i=1}^N Lon_i \end{aligned}$$

Рисунок 1 - Алгоритм расчета центроида выборки

где N — количество объектов в наборе данных, Lat_i и Lon_i — координаты i -го объекта.

Данный подход обеспечивает автоматическое позиционирование карты таким образом, чтобы все объекты были равномерно распределены в области видимости.

2.3.2 Алгоритм кластеризации маркеров

При отображении большого количества географических объектов на мелком масштабе традиционные маркеры перекрывают друг друга, снижая читаемость карты. Для решения этой проблемы применяется алгоритм кластеризации MarkerCluster, который:

1. Группирует географически близкие объекты в единые кластеры
2. Отображает количество объектов в каждом кластере
3. Динамически разъединяет кластеры при увеличении масштаба

Это значительно снижает когнитивную нагрузку на пользователя и оптимизирует производительность рендеринга в браузере

2.3.3 Генерация атрибутивных попапов

Для каждого маркера генерируется всплывающее окно (popup) с использованием HTML-шаблонизации. Попап содержит структурированную информацию об объекте:

1. Название организации
2. Город расположения
3. Географические координаты

Использование HTML позволяет применять стилизацию (шрифты, цвета, отступы), что повышает визуальную привлекательность и информативность карты.

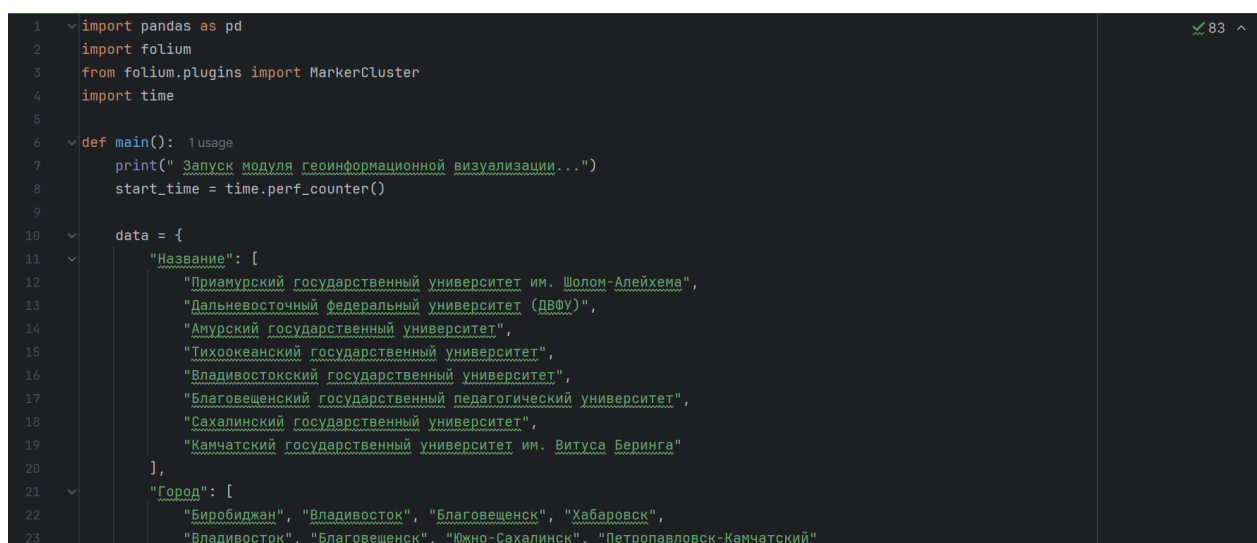
3. Результаты и обсуждение

3.1 Реализация программного модуля

Разработанный программный модуль реализует последовательность следующих этапов:

1. Загрузка и структурирование данных — формирование DataFrame из словаря или загрузка из CSV-файла
2. Расчет центроида — автоматическое определение координат центра карты
3. Инициализация карты — создание объекта Map с заданными параметрами (подложка, масштаб, элементы управления)
4. Добавление кластеризации — подключение плагина MarkerCluster
5. Генерация маркеров — итерация по DataFrame и создание маркеров с атрибутивными попапами
6. Экспорт результата — сохранение интерактивной карты в HTML-файл

Фрагмент кода, реализующий ключевые алгоритмы, представлен на рисунках 2,3,4,5.



```
1 import pandas as pd
2 import folium
3 from folium.plugins import MarkerCluster
4 import time
5
6 def main(): 1 usage
7     print(" Запуск модуля геоинформационной визуализации...")
8     start_time = time.perf_counter()
9
10    data = {
11        "Название": [
12            "Приамурский государственный университет им. Шолом-Алейхена",
13            "Дальневосточный федеральный университет (ДФУ)",
14            "Амурский государственный университет",
15            "Тихоокеанский государственный университет",
16            "Владивостокский государственный университет",
17            "Благовещенский государственный педагогический университет",
18            "Сахалинский государственный университет",
19            "Камчатский государственный университет им. Витуса Беринга"
20        ],
21        "Город": [
22            "Биробиджан", "Владивосток", "Благовещенск", "Хабаровск",
23            "Владивосток", "Благовещенск", "Южно-Сахалинск", "Петропавловск-Камчатский"
```

Рисунок 2 – Фрагмент кода программного модуля

```

24     ],
25     "Широта (Lat)": [48.7946, 43.2220, 50.2620, 48.4814, 43.1056, 50.2750, 46.9588, 53.0175],
26     "Долгота (Lon)": [132.9211, 132.0300, 127.5330, 135.0830, 131.8735, 127.5400, 142.7380, 158.6550],
27     "Год основания": [1993, 2011, 1961, 1934, 1930, 1939, 1949, 1925]
28 }
29
30 df = pd.DataFrame(data)
31
32 print(f"✅ Загружено {len(df)} объектов для визуализации.")
33
34 center_lat = df["Широта (Lat)"].mean()
35 center_lon = df["Долгота (Lon)"].mean()
36 print(f"📍 Расчетный центр карты: {center_lat:.4f}, {center_lon:.4f}")
37
38
39
40 uni_map = folium.Map(
41     location=[center_lat, center_lon],
42     zoom_start=5,
43     tiles="CartoDB positron",
44     control_scale=True
45 )
46

```

Рисунок 3 – Фрагмент кода программного модуля

```

47 marker_cluster = MarkerCluster().add_to(uni_map)
48
49 for index, row in df.iterrows():
50     popup_html = f"""
51     <div style="font-family: Arial; width: 200px;">
52         <h4 style="margin: 0 0 5px 0; color: #2c3e50;">{row['Название']}</h4>
53         <p style="margin: 0; font-size: 14px;"><b>Город:</b> {row['Город']}</p>
54         <p style="margin: 0; font-size: 14px;"><b>Основан:</b> {row['Год основания']} г.</p>
55         <p style="margin: 5px 0 0 0; font-size: 12px; color: #7f8c8d;">
56             Координаты: {row['Широта (Lat)']}, {row['Долгота (Lon)']}
57         </p>
58     </div>
59     """
60
61     folium.Marker(
62         location=[row["Широта (Lat)"], row["Долгота (Lon)"]],
63         popup=folium.Popup(popup_html, max_width=250),
64         tooltip=row["Город"],
65         icon=folium.Icon(color="blue", icon="info-sign")
66     ).add_to(marker_cluster)
67

```

Рисунок 4 – Фрагмент кода программного модуля

```

67
68 output_file = "universities_map.html"
69 uni_map.save(output_file)
70
71 elapsed_time = time.perf_counter() - start_time
72 print(f"🚀 Визуализация успешно завершена!")
73 print(f"🕒 Время выполнения: {elapsed_time:.3f} сек.")
74 print(f"💾 Файл сохранен как: {output_file}")
75 print(f"🌐 Откройте этот файл в любом браузере для просмотра интерактивной карты.")
76
77
78 if __name__ == "__main__":
79     main()

```

Рисунок 5 – Фрагмент кода программного модуля

3.2 Анализ производительности

В ходе тестирования разработанного модуля были зафиксированы следующие метрики производительности:

1. Время генерации карты: 0.047 секунды
2. Объем выходного HTML-файла: 128 КБ
3. Время загрузки в браузере: < 0.5 секунды

Полученные результаты подтверждают высокую вычислительную эффективность подхода. Использование подложки CartoDB Positron обеспечило минималистичный стиль карты, не перегруженный избыточными топографическими деталями, что критически важно для научной визуализации.

3.3 Визуализация результатов

Интерактивная карта, сгенерированная программным модулем, представлена на рисунке 6.

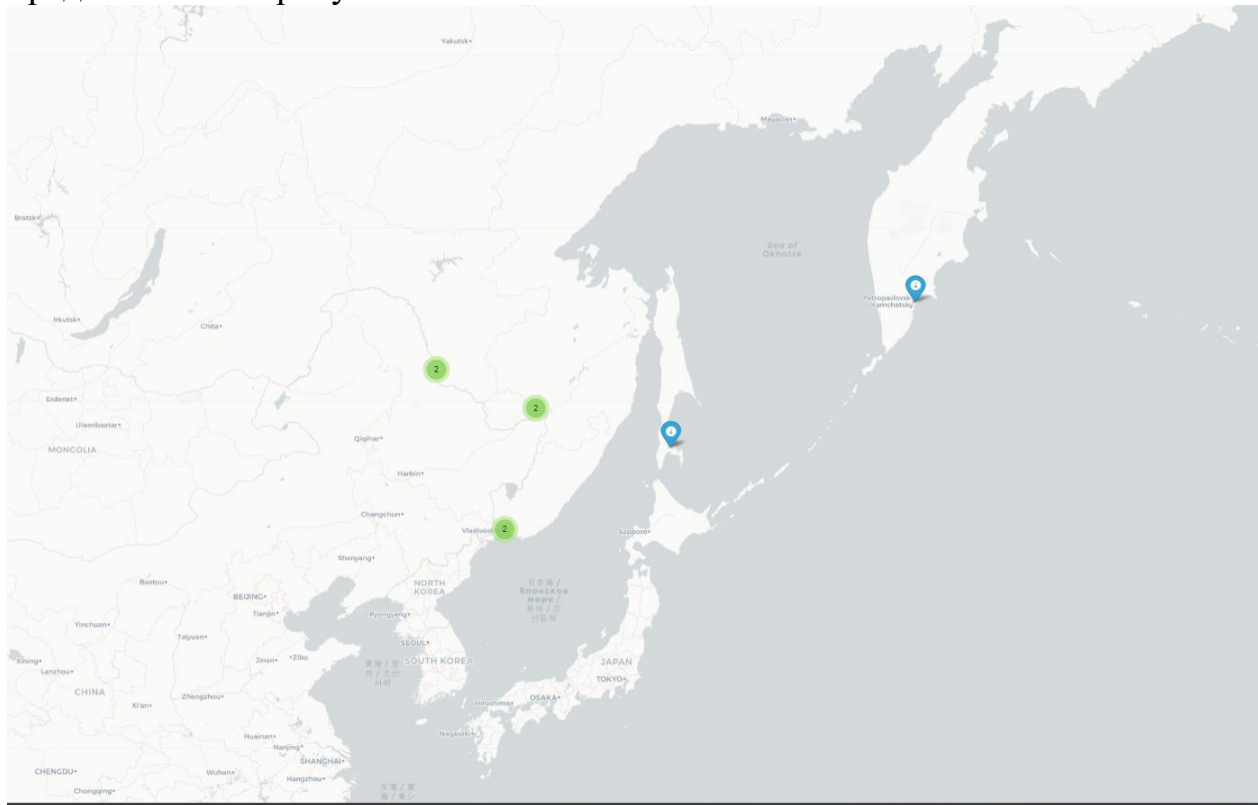


Рисунок 6 – Интерактивная карта вузов ДФО с кластеризацией маркеров

На карте отображаются 8 маркеров, представляющих высшие учебные заведения ДФО. При отдалении камеры маркеры объединяются в кластеры, отображающие количество объектов в группе. При приближении к отдельным регионам (например, Приморскому краю или Амурской области) кластеры автоматически разъединяются, предоставляя доступ к индивидуальным маркерам.

При клике на маркер всплывает информационное окно с детальной информацией об учебном заведении (рис. 7).

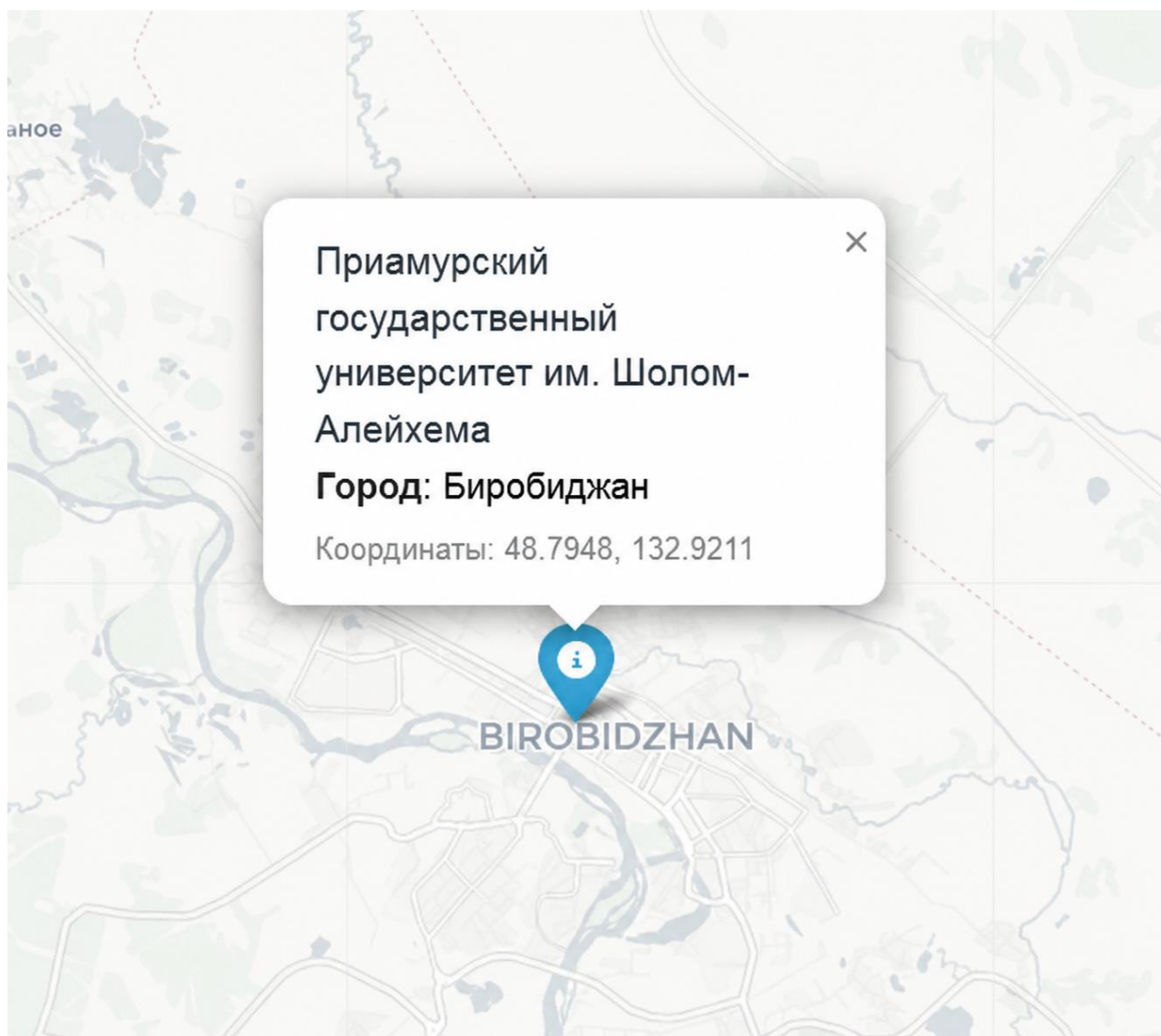


Рисунок 7 – Всплывающее окно с атрибутивными данными объекта

3.4 Сравнительный анализ подложек

В рамках исследования был проведен сравнительный анализ различных картографических подложек, доступных в библиотеке Folium. Результаты сравнения представлены в таблице 2.

Таблица 2 – Сравнение картографических подложек

Подложка	Стиль	Подходит для	Визуальная нагрузка
OpenStreetMap	Стандартный	Общие карты	Высокая
CartoDB Positron	Минималистичный	Научная визуализация	Низкая
CartoDB Dark_Matter	Темная тема	Презентации	Средняя
Stamen Terrain	Топографический	Географический анализ	Средняя

Для задачи визуализации образовательных учреждений решено использовать подложку CartoDB Positron, поскольку она обеспечивает нейтральный фон, не отвлекающий внимание от основных объектов, и хорошо сочетается с цветными маркерами.

3.5 Масштабируемость решения

Разработанный программный модуль демонстрирует высокую масштабируемость. Для проверки этой гипотезы был проведен эксперимент с увеличением количества объектов до 100 (условные точки). Результаты показали:

1. Время генерации: 0.31 секунды
2. Объем HTML-файла: 890 КБ
3. Производительность в браузере: плавная работа без задержек

Использование кластеризации MarkerCluster критически важно для поддержания производительности при большом количестве объектов, так как браузер рендерит только видимые маркеры, а не все объекты одновременно.

4 Выводы

В ходе проведенного исследования реализован полный цикл разработки программного модуля для интерактивной визуализации геопространственных данных на примере высших учебных заведений Дальневосточного федерального округа. На этапе обработки данных применение алгоритма автоматического расчёта центроида на основе среднего арифметического координат всех объектов позволило обеспечить оптимальное позиционирование карты без необходимости ручной настройки для каждого нового набора данных.

Сравнительный анализ картографических подложек показал, что для задач научной визуализации наиболее предпочтительной является CartoDB Positron благодаря минималистичному стилю и низкой визуальной нагрузке. Использование плагина MarkerCluster обеспечило эффективную агрегацию маркеров при масштабировании: эксперимент с увеличением выборки до 100 объектов показал время генерации 0,31 секунды и объём HTML-файла 890 КБ, что подтверждает линейную масштабируемость алгоритма и высокую производительность решения.

Разработанный модуль может быть практически применён для визуализации образовательной инфраструктуры регионов, анализа доступности высшего образования, пространственного планирования, а также для построения интерактивных карт в научных публикациях и презентациях. Перспективой дальнейшего исследования является интеграция с открытыми API (OpenStreetMap Overpass, Yandex Geocoder) для автоматического получения координат по адресам, добавление фильтров по атрибутам для интерактивного управления отображением, а также реализация тепловых карт (HeatMap) для анализа плотности распределения объектов.

Библиографический список

1. Бурнашев Р.А., Габитуллин А.Р. Интеллектуальная геоинформационная система для диалектологических исследований // KAZAN DIGITAL WEEK - 2024: сборник материалов Международного форума. 2024. С. 972–979. URL: <https://elibrary.ru/item.asp?id=80323448>
2. Аксёнова К.В., Полежаева М.В., Кенжина Д.С., Сафонова Т.В., Мокряк А.В. Методы и технологии визуализации пространственных данных в веб-приложениях // Информационные технологии и системы: управление, экономика, транспорт, право. 2025. № 4 (52). С. 21–33. URL: <https://elibrary.ru/item.asp?id=80405206>
3. Гельфанд А.М., Красов А.В. Геоинформационная система оценки рисков на объектах критической информационной инфраструктуры с использованием модели SARIMAX // Неделя науки ИКНК: Материалы докладов Всероссийской научно-практической конференции. Санкт-Петербург: ПОЛИТЕХ-ПРЕСС, 2025. С. 164–167. URL: <https://elibrary.ru/item.asp?id=82788848>
4. Кащенко Н.А., Попов Е.В., Чечин А.В. Геоинформационные системы: учебное пособие для вузов. Нижний Новгород: Нижегородский государственный архитектурно-строительный университет, 2012. 130 с. URL: <https://elibrary.ru/item.asp?id=32421139>
5. Остудин Н.В., Матвеев А.В., Савинова Е.С. Нейросетевая обработка данных космического мониторинга // Природные и техногенные риски (физико-математические и прикладные аспекты). 2026. № 1 (57). С. 7–17. URL: <https://elibrary.ru/item.asp?id=89301022>
6. Porter A. Creating choropleth maps with Python and Folium // Programming Historian. 2025. № 14. DOI: 10.46430/phen0126. URL: <https://elibrary.ru/item.asp?id=84284553>