

Создание нового слоя объектов, содержащих определённые значения атрибутов, в другом слое используя программу Qgis и язык Python

Вихляев Дмитрий Романович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье рассматривается создание программы в среде Qgis имеющей функции работы с векторными слоями и их атрибутами. Программа написана на языке программирования Python. Результатом исследования станет готовая программа, создающая новый векторный слой из объектов старого.

Ключевые слова: Qgis, ГИС, Python, векторные данные, атрибуты.

Creating a new layer of objects containing certain attribute values in another layer using the Qgis program and Python language

Vikhlyaev Dmitry Romanovich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article discusses the creation of a program in the Qgis environment that has functions for working with vector layers and their attributes. The program is written in the Python programming language. The result of the study will be a ready-made program that creates a new vector layer from the objects of the old one.

Keywords: Qgis, GIS, Python, vector data, attributes.

1 Введение

1.1 Актуальность

Атрибуты в векторных географических файлах - это информация, предназначенная для описания и характеристики географических объектов, представленных в виде точек, линий или полигонов. Эти данные хранятся в специальной таблице, где каждая строка соответствует отдельному геометрическому объекту, а каждый столбец содержит определенный атрибут. Полезность атрибутов в векторных географических файлах заключается в их способности обеспечивать интерпретацию и анализ географических данных. Например, атрибуты могут содержать информацию о названии объекта, его типе, координатах, высоте над уровнем моря, населении и других характеристиках. Это позволяет пользователям легко понимать и работать с географическими данными. Атрибуты позволяют осуществлять фильтрацию и выборку данных, визуализировать информацию

на картах, проводить геоинформационный анализ, а также интегрировать и обмениваться данными между различными геоинформационными системами. Применение атрибутов в векторных географических файлах охватывает широкий спектр областей, включая географическую информационную науку, геоинформационные системы, картографию, геодезию, экологию, транспортное планирование, а также различные отрасли, где требуется работа с пространственными данными и анализ географических объектов.

1.2 Обзор исследований

Ю.С.Борисова в своей работе создала условные знаки в геоинформационной среде Qgis для изображения сетей коммуникаций на крупномасштабных топографических планах [1]. А.О.Белоусов разработал технологию формирования информационной базы о состоянии земель сельскохозяйственных организаций в геоинформационной системе Qgis [2]. И.Б.Гинзбург использовал автономное приложение при работе с данными дистанционного зондирования земли и векторными кадастровыми слоями [3]. Е.Н.Дремов изучил метод распознавания значений высот на векторных слоях геопространственных данных [4]. И.Н.Беляева, А.В. Федоров создали векторный слой интерактивной карты из растрового изображения плана здания [5]. С.В.Смирнов, С.С.Наговицын, Ю.В.Русских Спроектировали векторные слои для геоинформационных систем [6].

1.3 Цель исследования

Цель исследования – написать скрипт программы, отделяющей объекты в слое по значению атрибутов и создать новый слой, состоящий только из выявленных объектов.

2 Материалы и методы

Для реализации используется геоинформационная система Qgis 3.36.1 с встроенным интерпретатором Python. В качестве векторных данных используются лесные кварталы Сахалинской области.

3 Результаты и обсуждения

Атрибуты в QGIS представляют собой ключевые данные, связанные с географическими объектами на карте. Эти данные описывают характеристики, такие как названия, числовые значения, даты и другую информацию. Атрибуты хранятся в таблице атрибутов каждого векторного слоя в проекте QGIS. Атрибуты позволяют интегрировать данные из различных источников и работать с ними в рамках одного проекта QGIS. Это обеспечивает удобство работы с разнообразными данными на карте. QGIS позволяет выполнять пространственный анализ на основе данных атрибутов. Например, можно проводить статистический анализ значений атрибутов в определенных областях на карте.

В качестве примера взяты лесные кварталы Сахалинской области. Слой представляет собой массив полигонов, каждый из которых является неким объектом со своими значениями атрибутов (рис.1).

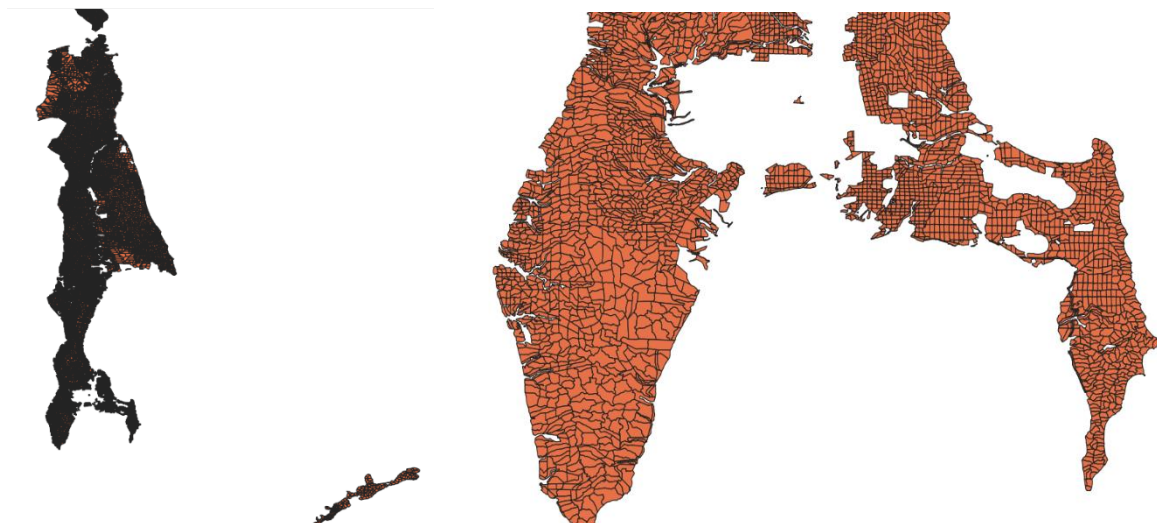


Рис. 1. Векторный слой

В Qgis для каждого слоя можно посмотреть таблицу атрибутов. В данном примере таблица имеет 4 колонки. Интересующая колонка «NAME_LV» содержит название лесничеств каждого полигона (рис.2).

	НАИМЕН_УР	ПЕР_КВАРТ	NAME_LV	NAME_ULV
1	NULL		1 Александровск...	Александровск...
2	NULL		1 Александровск...	Хознское
3	часть 1		1 Анивское	Анивское
4	часть 2 (б.Брян...		1 Анивское	Анивское
5	NULL		1 Анивское	Таранайское
6	часть 1		1 Долинское	Долинское
7	часть 2 (б.Парк...		1 Долинское	Долинское
8	часть 3 (б.Найб...		1 Долинское	Долинское
9	часть 1		1 Долинское	Фирсовское
10	часть 2 (б.Айск...		1 Долинское	Фирсовское
11	часть 1		1 Корсаковское	Озерское
12	часть 2 (б.Нов...		1 Корсаковское	Озерское
13	часть 1 (б.Лесн...		1 Корсаковское	Охотское
14	часть 1		1 Красногорское	Ильинское
15	часть 2 (б.Пенз...		1 Красногорское	Ильинское

Рис. 2. Таблица атрибутов

Для начала необходимо создать функцию, которая получает значения колонки в таблице. Используются Python и встроенные модули «QgsProject» и «QgsVectorLayer» из библиотеки «qgis.core». Код, представленный на рисунке 3, позволяет извлекать значения конкретного атрибута из всех объектов векторного слоя в QGIS. Он выполняет поиск слоя по имени, определяет индекс поля атрибута по его имени и затем перебирает все объекты слоя, чтобы собрать и вернуть значения этого атрибута в виде списка. Статический метод QgsProject.instance(), возвращает единственный

экземпляр проекта QGIS. QgsProject используется для управления открытыми проектами и слоями. Поиск слоя в проекте по имени осуществляется через метод mapLayersByName(layer_name). Возвращает список всех слоев с указанным именем. Если слой с таким именем есть, они будут возвращены в списке. Нулевой элемент массива означает, что берётся первый элемент из списка слоев с указанным именем. Если такого слоя нет или если список пустой, возникнет ошибка. Получить список объектов через layer.getFeatures(). Этот метод возвращает итератор по всем объектам (features) в слое. Объекты представляют собой географические объекты, такие как точки, линии или полигоны. Затем из каждого объекта получить атрибуты, используя feature.attributes(). И извлечь определённый атрибут, указанный в переменной field_index, в методе feature.attributes()[field_index]. Чтобы получить список только уникальных значений можно использовать структуру данных «set» встроенную в Python.

```
#получить список значений атрибута по его имени и названию слоя
def get_list_attribute_value(layer_name, attribute_name):

    # Получаем слой по его имени
    layer = QgsProject.instance().mapLayersByName(layer_name)[0]

    # Получаем индекс поля атрибута
    field_index = layer.fields().indexFromName(attribute_name)

    if field_index != -1:
        # Создаем пустой список для сохранения значений атрибута
        attribute_values = []

        # Перебираем все объекты слоя и сохраняем значения атрибута в список
        for feature in layer.getFeatures():
            value = feature.attributes()[field_index]
            attribute_values.append(value)

        # Выводим список значений атрибута
        return attribute_values
    else:
        print(f"Атрибут с именем '{attribute_name}' не найден в слое '{layer_name}'.")
        return []

Local/Temp/tmp3uhha3tn').read_text())
Невельское
Александровское
Корсаковское
Поронайское
Холмское
Смирныховское
Охинское
Южно-Сахалинское
Красногорское
Долинское
Южно-Курильское
Курильское
Углегорское
Ногиликское
Макаровское
Анивское
Тымовское
```

Рис. 3. Функция получения списка значений колонки атрибутов и вывод в уникальных значениях

Для создания нового векторного слоя в QGIS используется конструктор QgsVectorLayer. В данном случае, строка 'Polygon?crs=' + layer2.crs().toWkt() определяет тип создаваемого слоя (многоугольный) и его координатную систему (CRS), взятую из существующего слоя layer2. Имя нового атрибута задается переменной attribute_value, которая преобразуется в строку для использования в качестве имени слоя.

Для обеспечения согласованности структуры атрибутов в новом векторном слое с уже существующим слоем layer2, сначала необходимо получить список атрибутов (полей) из слоя. Это выполняется с помощью метода fields().toList(), который возвращает список атрибутов в удобном формате для дальнейшего использования. Полученная структура атрибутов затем добавляется в dataProvider нового слоя result_layer с использованием метода addAttributes(). Этот шаг гарантирует, что новый слой будет иметь такую же структуру атрибутов, как и слой layer2, что важно для совместимости и последующей обработки данных. После добавления атрибутов, необходимо выполнить обновление полей с помощью метода updateFields(). Это обеспечивает актуализацию метаданных нового слоя

result_layer, что позволяет QGIS корректно интерпретировать и работать с атрибутами (рис.4).

```
#Создать новый обрезаемый слой содержащий только объекты с определённым
#значением атрибута, параметры название слоя, название поля атрибута, значение атрибута
def create_cut_layer_for_attribute_value(layer_name2, attribute_name, attribute_value, output_path):

    # Получаем второй слой по его имени
    layer2 = QgsProject.instance().mapLayersByName(layer_name2)
    if not layer2:
        print(f"Слой '{layer_name2}' не найден.")
        return

    layer2 = layer2[0]

    # Создаем новый пустой векторный слой для результата
    result_layer = QgsVectorLayer('Polygon?crs=' + layer2.crs().toWkt(), str(attribute_value), 'memory')
    if not result_layer.isValid():
        print("Ошибка при создании слоя.")
        return

    # Создаем атрибуты в новом слое на основе структуры атрибутов второго слоя
    attributes = layer2.fields().toList()
    result_layer.dataProvider().addAttributes(attributes)
    result_layer.updateFields()
```

Рис. 4. Создание нового слоя

Для выполнения операции фильтрации объектов векторного слоя по значению определённого атрибута и их последующего добавления в новый слой, используется следующий подход. Создаётся пустой список, который будет содержать объекты, соответствующие заданному условию. Происходит итерация по всем объектам (features) в слое с помощью метода getFeatures(). Проверяется, соответствует ли значение атрибута в текущем объекте. Если условие выполняется, текущий объект добавляется в список. После завершения цикла проверяется список selected_features. Если список пуст, выводится сообщение о том, что не было найдено объектов со значением для атрибута. Это гарантирует, что пользователь будет проинформирован о результате операции. Если список не пуст, используется метод addFeatures() провайдера данных (dataProvider) нового слоя для добавления всех объектов из selected_features (рис.5).

```
# Фильтруем объекты второго слоя и добавляем их в новый слой
selected_features = []
for feature in layer2.getFeatures():
    if feature[attribute_name] == attribute_value:
        selected_features.append(feature)

if not selected_features:
    print(f"Нет объектов со значением '{attribute_value}' для атрибута '{attribute_name}'.")
    return

result_layer.dataProvider().addFeatures(selected_features)

# Добавляем новый слой на карту
```

Рис. 5. Определение нужных объектов и добавление в новый слой

В конечном итоге все слои сохраняются на машине пользователя в формате geojson. Кодировка файла установлена на UTF-8 для поддержки многоязычных данных. Координатная система задана как EPSG:4326 (WGS 84), что является стандартом для географических данных. Функция QgsVectorFileWriter.writeAsVectorFormat выполняет операцию записи в указанный формат geojson с заданными параметрами (рис.6,7).

```
# Сохраняем слой в формате GeoJSON
layer = .result_layer
output = .output_path+'\\'+str(attribute_value)
coder = .'utf-8'
coordinate_system = .QgsCoordinateReferenceSystem('EPSG:4326')
format = .'GeoJSON'

writer = .QgsVectorFileWriter.writeAsVectorFormat(layer, .output, .coder, .coordinate_system, .format)
```

Рис. 6. Сохранение слоя в файл

Александровское.geojson	10.06.2024 21:45	Файл "GEOJSON"	9 868 КБ
Анивское.geojson	10.06.2024 21:45	Файл "GEOJSON"	4 223 КБ
Долинское.geojson	10.06.2024 21:45	Файл "GEOJSON"	5 304 КБ
Корсаковское.geojson	10.06.2024 21:45	Файл "GEOJSON"	5 700 КБ
Красногорское.geojson	10.06.2024 21:45	Файл "GEOJSON"	7 516 КБ
Курильское.geojson	10.06.2024 21:45	Файл "GEOJSON"	5 667 КБ
Макаровское.geojson	10.06.2024 21:45	Файл "GEOJSON"	4 836 КБ
Невельское.geojson	10.06.2024 21:45	Файл "GEOJSON"	2 213 КБ
Ногликское.geojson	10.06.2024 21:45	Файл "GEOJSON"	18 725 КБ
Охинское.geojson	10.06.2024 21:45	Файл "GEOJSON"	21 862 КБ
Поронайское.geojson	10.06.2024 21:45	Файл "GEOJSON"	10 258 КБ
Смирныховское.geojson	10.06.2024 21:45	Файл "GEOJSON"	17 105 КБ
Тымовское.geojson	10.06.2024 21:45	Файл "GEOJSON"	9 173 КБ
Углегорское.geojson	10.06.2024 21:45	Файл "GEOJSON"	6 949 КБ
Холмское.geojson	10.06.2024 21:45	Файл "GEOJSON"	4 799 КБ
Южно-Курильское.geojson	10.06.2024 21:45	Файл "GEOJSON"	1 247 КБ
Южно-Сахалинское.geojson	10.06.2024 21:45	Файл "GEOJSON"	1 310 КБ

Рис. 7. Сохранённые файлы

Для визуализации можно раскрасить все новые слои и вывести на карту, получится отображение исходного слоя (рис.8).



Рис. 8. Цветовое разделение участков

4 Выводы

В результате исследования была написана программа для разделения объектов слоя по значению поля атрибута. Описаны функции и методы, способствующие достижению конечной цели. Приведён пример на

практической задаче из реальной жизни, что обеспечивает актуальность работы.

Библиографический список

1. Борисова Ю.С. Создание условных знаков в геоинформационной среде Qgis для изображения сетей коммуникаций на крупномасштабных топографических планах // В сборнике: Актуальные вопросы землепользования и управления недвижимостью. Сборник статей III Национальной научно-практической конференции. Екатеринбург, 2021. С. 52-61.
2. Белоусов А.О. Технология формирования информационной базы о состоянии земель сельскохозяйственных организаций в геоинформационной системе Qgis // В сборнике: Рациональное использование земельных ресурсов в условиях современного развития АПК. Сборник материалов Всероссийской (национальной) научно-практической конференции. Тюмень, 2021. С. 15-21.
3. Гинзбург И.Б. Использование автономного веб-приложения при работе с данными дистанционного зондирования земли и векторными кадастровыми слоями // Альманах мировой науки. 2015. № 1-1 (1). С. 79-80.
4. Дремов Е.Н. Метод распознавания значений высот на векторных слоях геопространственных данных // В сборнике: Оптико-электронные приборы и устройства в системах распознавания образов, обработки изображений и символьной информации. Распознавание - 2017. сборник материалов XIII Международной научно-технической конференции. 2017. С. 134-136.
5. Беляева И.Н., Федоров А.В. Создание векторного слоя интерактивной карты из растрового изображения плана здания // В сборнике: Актуальные проблемы развития науки и современного образования. Сборник материалов Международной научно-практической конференции. 2017. С. 77-79.
6. Смирнов С.В., Наговицын С.С., Русских Ю.В. Проектирование векторных слоев как начальная стадия создания геоинформационных систем // Социально-экономическое управление: теория и практика. 2019. № 2 (37). С. 72-75.
7. Серебренников И.В. Построение архитектуры программного комплекса на основе Qgis // В сборнике: Наука, знания, интеллект. Сборник статей III Международного научно-исследовательского конкурса. Пенза, 2023. С. 47-50.