

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Навчально-науковий
інститут електроенергетики
(інститут)
Факультет інформаційних технологій
(факультет)
Кафедра інформаційних технологій та комп'ютерної інженерії
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня бакалавра

здобувача Ждимори Артема Сергійовича
(ІПБ)

академічної групи 126-21-1
(шифр)

спеціальності 126 Інформаційні системи та технології
(код і назва спеціальності)

за освітньо-професійною програмою Інформаційні системи та технології
(офіційна назва)

на тему «Розробка додатку для побудови діаграм розподілу азимутів лінійних»
об'єктів у програмному середовищі QGIS
(назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	проф. Нікулін С.Л.			
розділів:				
Рецензент				
Нормоконтролер	проф. Коротенко Г.М.			

Дніпро
2025

ЗАТВЕРДЖЕНО:

завідувач кафедри
інформаційних технологій
та комп'ютерної інженерії

(повна назва)

Гнатушенко В.В.

(підпис)

(прізвище, ініціали)

« ____ » _____ 2025 року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня бакалавра

здобувача Ждиморі А.С. академічної групи 126-21-1
(прізвище та ініціали) (шифр)

спеціальності 126 Інформаційні системи та технології

за освітньо-професійною програмою Інформаційні системи та технології
(офіційна назва)

на тему «Розробка додатку для побудови діаграм розподілу азимутів лінійних об'єктів у програмному середовищі QGIS»

затверджену наказом ректора НТУ «Дніпровська політехніка» від 05.05.2025 № 336-с

Розділ	Зміст	Термін виконання
Розділ 1	Аналіз стану області рішення задачі	25.01.2025 – 28.02.2025
Розділ 2	Аналіз та проектування додатку для побудови діаграм розподілу азимутів	01.03.2025 – 30.04.2025
Розділ 3	Розробка плагіну для побудови діаграм розподілу азимутів лінійних об'єктів у програмному середовищі QGIS мовою Python	01.05.2025 – 30.05.2025

Завдання видано _____ С.Л. Нікулін
(підпис керівника) (прізвище, ініціали)

Дата видачі 25.02.2025

Дата подання до екзаменаційної комісії 23.06.2025

Прийнято до виконання _____ Ждимора А.С.
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 60 стор., 16 рис., 2 додатки, 11 джерело.

Об'єкт розробки: інструмент для побудови діаграм розподілу азимутів лінійних об'єктів у програмному середовищі QGIS.

Мета кваліфікаційної роботи: розробка додатку для побудови діаграм розподілу азимутів лінійних об'єктів в середовищі QGIS за допомогою мови програмування Python, який дозволяє користувачу в зручному форматі отримувати дані, необхідні для аналізу.

У вступі наведено стан проблеми та обґрунтована її актуальність.

Перший розділ включає в себе постановку завдання і характеристику предметної області.

У другому розділі було розглянуто теперішній стан використання діаграм розподілу азимутів, мова програмування Python та інструменти для аналізу в QGIS.

У третьому розділі розроблений плагін для побудови діаграм розподілу азимутів лінійних об'єктів мовою Python. Продемонстрована інструкція користувача з використанням елементів інтерфейсу користувача для наглядного прикладу роботи додатку.

Практичне значення роботи полягає у створенні додатку (плагіна) за допомогою мови програмування Python для побудови діаграм розподілу азимутів лінійних об'єктів в середовищі QGIS.

Список ключових слів: ГЕОДАНИ, UML, ДІАГРАМА РОЗПОДІЛУ АЗИМУТІВ, PYTHON, QGIS, ДОДАТОК, АНАЛІЗ, API

ABSTRACT

Explanatory note: 60 pages, 16 figures, 2 appendices, 11 sources.

The object of development: a tool for constructing azimuth distribution diagrams of linear objects in the QGIS software environment.

The purpose of the thesis: developing an application for constructing diagrams of the distribution of azimuths of linear objects in the QGIS environment using the Python programming language, which allows the user to obtain the data necessary for analysis in a convenient format.

The introduction presents the state of the problem and justifies its relevance.

The first section includes the statement of the problem and the description of the subject area.

The second section considers the current state of use of azimuth distribution diagrams, the Python programming language, and tools for analysis in QGIS.

The third section develops a plugin for constructing azimuth distribution diagrams of linear objects in Python. A user manual using user interface elements is demonstrated for a visual example of the application's operation.

The practical significance of the work lies in creating an application (plugin) using the Python programming language for constructing azimuth distribution diagrams of linear objects in the QGIS environment.

Keywords: GEO DATA, UML, AZIMUTH DISTRIBUTION DIAGRAM, PYTHON, QGIS, APPLICATION, ANALYSIS, API

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ	7
ВСТУП.....	8
1 АНАЛІЗ СТАНУ ОБЛАСТІ РІШЕННЯ ЗАДАЧІ	10
1.1 Актуальність задачі.....	10
1.2 Програмне забезпечення QGIS як інструмент для просторового аналізу	11
1.2.1 Огляд можливостей QGIS	11
1.2.2 Архітектура QGIS та розширення функціоналу (плагіни)	12
1.3 Лінійні об'єкти в ГІС.....	14
1.4 Види діаграм, їх застосування. Полярний графік.....	15
1.5 Концепція азимуту та його значення у просторовому аналізі	17
1.5.1 Визначення азимуту та одиниці вимірювання.....	17
1.5.2 Роль азимуту в дослідженні напрямків лінійних об'єктів	18
1.6 Висновки	19
2 АНАЛІЗ ТА ПРОЄКТУВАННЯ ДОДАТКУ ДЛЯ ПОБУДОВИ ДІАГРАМ РОЗПОДІЛУ АЗИМУТІВ.....	20
2.1 Аналіз існуючих інструментів та підходів до аналізу азимутів у ГІС	20
2.1.1 Огляд вбудованих функцій QGIS	20
2.1.2 Аналіз наявних плагінів та їх функціоналу.....	22
2.2 Проєктування архітектури додатку (плагіну)	23
2.2.1 Вибір мови програмування Python.....	23
2.2.2 Вибір бібліотек та фреймворків	25
2.3 Діаграма класів	30
2.4 Висновки	32
3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ДОДАТКУ	34
3.1 Опис середовища розробки та встановлення необхідних компонентів	34
3.2 Моделювання бізнес-процесу функціонування додатку для побудови діаграм розподілу азимутів	35
3.3 Створення плагіну для QGIS.....	38

3.4 Висновки	44
ВИСНОВКИ.....	46
ПЕРЕЛІК ПОСИЛАНЬ	47

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

OGC - Open Geospatial Consortium;

WMS – Web Mapping Service;

SLD – Styled Layer Descriptor;

TJS - Table Joining Service;

CSV - Comma-Separated Values;

ДГІ – добровільної географічної інформації;

ГІС – географічні інформаційні системи;

IDE – Integrated Drive Electronics;

QGIS - Quantum GIS.

ВСТУП

Сучасний розвиток геоінформаційних систем (ГІС) відкриває широкі можливості для ефективного збору, зберігання, аналізу та візуалізації просторових даних. Серед безлічі ГІС-інструментів особливе місце посідає програмне забезпечення з відкритим вихідним кодом, таке як QGIS, яке завдяки своїй доступності та розширюваності завоювало значну популярність у професійному та освітньому середовищах. Одним з ключових аспектів просторового аналізу є дослідження характеристик лінійних об'єктів, які відіграють важливу роль у відображенні різноманітних природних та антропогенних процесів – від тектонічних розломів та гідрографічної мережі до транспортних шляхів та комунікацій, [1].

Вивчення напрямків (азимутів) лінійних об'єктів дозволяє виявляти приховані закономірності, оцінювати вплив географічних факторів на розподіл об'єктів, а також обґрунтовувати просторові рішення. Традиційні методи аналізу азимутів часто є трудомісткими та вимагають використання спеціалізованого програмного забезпечення або ручних розрахунків. Незважаючи на наявність певних інструментів у ГІС-пакетах, існує потреба у розробці спеціалізованих, зручних у використанні додатків, які можуть інтегруватися безпосередньо у робоче середовище ГІС та надавати якісну візуалізацію результатів.

Актуальність роботи полягає у зростаючій потребі в автоматизованих та ефективних інструментах для просторового аналізу лінійних об'єктів у ГІС, зокрема, у візуалізації розподілу їхніх азимутів. Розробка такого додатку для QGIS дозволить розширити функціональні можливості цього програмного забезпечення, надаючи користувачам потужний засіб для досліджень у геології, гідрології, екології, містобудуванні та інших галузях, де орієнтація лінійних структур має вирішальне значення.

Об'єктом дослідження є інструмент просторового аналізу лінійних об'єктів у ГІС.

Метою роботи є розробка та реалізація додатку (плагіну) для програмного середовища QGIS, призначеного для побудови діаграм розподілу азимутів лінійних об'єктів, що забезпечить ефективний просторовий аналіз та наочну візуалізацію даних.

Для досягнення поставленої мети визначено наступні завдання:

- проаналізувати теоретичні основи геоінформаційних систем, особливості представлення лінійних об'єктів та методи розрахунку азимутів;
- провести огляд та оцінку існуючих інструментів QGIS та сторонніх рішень для аналізу напрямків, виявити їхні обмеження;
- розробити архітектуру та спроектувати інтерфейс користувача додатку, обравши відповідні мови програмування та бібліотеки;
- реалізувати основний функціонал додатку для QGIS, включаючи алгоритми розрахунку азимутів та інтеграцію засобів візуалізації;
- провести тестування розробленого додатку на реальних геопросторових даних для підтвердження його коректності та ефективності.

1 АНАЛІЗ СТАНУ ОБЛАСТІ РІШЕННЯ ЗАДАЧІ

1.1 Актуальність задачі

Сучасний розвиток геоінформаційних систем (ГІС) та їх широке впровадження у різноманітні сфери людської діяльності суттєво змінило підходи до аналізу та візуалізації просторових даних. В умовах зростаючої складності геопросторових завдань, особливої актуальності набуває розробка спеціалізованих інструментів, що дозволяють ефективно опрацьовувати та інтерпретувати великі обсяги інформації. Однією з таких критично важливих задач є аналіз лінійних об'єктів, які є невід'ємною частиною будь-якої ГІС-моделі світу – від доріг, річок та комунікацій до геологічних розломів та урбаністичних структур.

Знання орієнтації лінійних об'єктів у просторі, тобто їхніх азимутів, є фундаментальним для багатьох наукових та прикладних досліджень. Наприклад, у геології аналіз розподілу азимутів тектонічних тріщин та розломів дозволяє реконструювати палеонапруження та розуміти динаміку земної кори. У гідрології вивчення орієнтації водотоків допомагає моделювати стоки та оцінювати ерозійні процеси. У містобудуванні та транспортному плануванні аналіз азимутів доріг може оптимізувати прокладання нових маршрутів та оцінити вплив існуючої інфраструктури на ландшафт.

Незважаючи на значні можливості сучасних ГІС, інтегровані інструменти для комплексного аналізу та візуалізації розподілу азимутів лінійних об'єктів часто є обмеженими або вимагають використання кількох окремих програмних модулів. Існуючі комерційні ГІС-пакети можуть пропонувати подібні функції, але їх вартість та закритий вихідний код обмежують доступність та можливість адаптації для широкого кола користувачів та дослідників, [1].

У цьому контексті програмне забезпечення з відкритим вихідним кодом, таке як QGIS, набуває особливого значення. QGIS надає потужну платформу для геопросторового аналізу, а його архітектура, що підтримує розробку

плагінів за допомогою мови Python, дозволяє створювати спеціалізовані інструменти, які відповідають конкретним потребам користувачів. Це відкриває шлях до розробки доступних, гнучких та розширюваних рішень.

Таким чином, розробка спеціалізованого додатку для QGIS, що дозволяє автоматично розраховувати та візуалізувати діаграми розподілу азимутів лінійних об'єктів, є актуальною задачею. Вона спрямована на розширення аналітичних можливостей QGIS, підвищення ефективності геопросторових досліджень та надання фахівцям зручного та наочного інструменту для інтерпретації просторових закономірностей у різних галузях науки та практики.

1.2 Програмне забезпечення QGIS як інструмент для просторового аналізу

QGIS, або Quantum GIS, є потужною геоінформаційною системою з відкритим вихідним кодом (Open Source Geographic Information System), яка відіграє центральну роль у сучасному просторовому аналізі. Вона є вільною та безкоштовною у використанні, що робить її доступною для широкого кола користувачів – від студентів та дослідників до державних установ та комерційних організацій по всьому світу. Ця доступність є однією з ключових переваг QGIS у порівнянні з комерційними ГІС-продуктами.

1.2.1 Огляд можливостей QGIS

QGIS пропонує надзвичайно широкий спектр можливостей для роботи з геопросторовими даними, що робить її комплексним інструментом для просторового аналізу. До основних функціональних блоків QGIS можна віднести, [2, 5]:

1. Перегляд та візуалізація даних. QGIS дозволяє відображати та досліджувати векторні (точки, лінії, полігони) та растрові (супутникові знімки, DEM) дані у різних форматах. Користувачі можуть налаштовувати символіку,

створювати тематичні карти, застосовувати різні стилі відображення для ефективною візуалізації просторової інформації.

2. Управління даними. Програмне забезпечення підтримує широкий спектр форматів геоданих (Shapefile, GeoJSON, KML, GeoPackage, PostGIS, Oracle Spatial тощо), дозволяючи імпортувати, експортувати та редагувати дані. Вбудовані інструменти управління атрибутивними таблицями дозволяють додавати, видаляти та модифікувати поля, виконувати запити та фільтрувати дані.

3. Геообробка та просторовий аналіз. QGIS має вбудований потужний фреймворк геообробки, який включає сотні алгоритмів для виконання різноманітних просторових аналізів. Це дозволяє здійснювати буферизацію, накладання шарів (intersection, union, clip), об'єднання, роз'єднання, аналіз близькості, інструменти для роботи з рельєфом (ухил, експозиція), мережевий аналіз, геостатистику та багато інших операцій.

4. Створення карт (картографія). QGIS надає професійні інструменти для створення високоякісних друкованих карт та атласів. Макетувальник друку дозволяє додавати заголовки, легенди, масштабні лінійки, сітки координат, текстові блоки та інші картографічні елементи.

5. Підтримка зовнішніх джерел даних. QGIS може підключатися до баз даних (PostGIS, SpatiaLite, MS SQL Server, Oracle Spatial), веб-сервісів (WMS, WMTS, WFS, ArcGIS REST services), дозволяючи працювати з даними, що зберігаються віддалено.

1.2.2 Архітектура QGIS та розширення функціоналу (плагіни)

Архітектура QGIS є модульною та розширюваною, що є однією з її найсильніших сторін. Вона побудована на мові програмування C++ з використанням бібліотеки Qt для графічного інтерфейсу та багатьох інших бібліотек для роботи з геопросторовими даними (наприклад, GDAL/OGR для форматів даних, GEOS для геометричних операцій, Proj для систем

координат). Однак, ключовим елементом її гнучкості та розширюваності є можливість розробки плагінів (модулів).

Модульна архітектура. QGIS має ядро, яке надає основну функціональність, але більша частина складних інструментів та можливостей реалізована через плагіни. Це дозволяє користувачам встановлювати лише ті функції, які їм потрібні, підтримуючи базову програму легкою та ефективною, [2, 5].

Розширення функціоналу через плагіни. Плагіни є основними засобами розширення функціоналу QGIS. Вони дозволяють додавати нові інструменти, алгоритми обробки, формати даних, покращувати інтерфейс користувача та автоматизувати рутинні завдання.

Розробка плагінів на Python. Переважна більшість плагінів QGIS розробляється з використанням мови програмування Python та бібліотеки PyQt5 (прив'язок Python до Qt). Це значно спрощує процес розробки, оскільки Python є відносно легким для вивчення та має велику кількість бібліотек, придатних для геопросторових обчислень та візуалізації (наприклад, matplotlib для побудови діаграм, як у нашому випадку).

QGIS API (Application Programming Interface). QGIS надає потужний API на Python, який дозволяє розробникам плагінів взаємодіяти з ядром QGIS, отримувати доступ до шарів, об'єктів, атрибутів, створювати нові елементи інтерфейсу та виконувати операції геообробки. Це дозволяє створювати інструменти, які глибоко інтегровані в середовище QGIS.

Менеджер модулів. QGIS має вбудований менеджер модулів, який дозволяє користувачам легко знаходити, встановлювати, оновлювати та керувати плагінами з офіційного репозиторію. Це створює живу екосистему, де розробники можуть ділитися своїми інструментами зі спільнотою.

Завдяки своїй модульній архітектурі та можливостям розширення через плагіни, QGIS перетворюється з базового ГІС-переглядача на гнучку та адаптовану платформу, здатну вирішувати широкий спектр складних завдань

просторового аналізу, включаючи створення спеціалізованих інструментів, таких як аналіз розподілу азимутів.

1.3 Лінійні об'єкти в ГІС

У світі геоінформаційних систем (ГІС) лінійні об'єкти є одним з трьох базових типів векторних даних (поряд з точками та полігонами), які використовуються для представлення реальних об'єктів та явищ на поверхні Землі. Ці об'єкти складаються з послідовності двох або більше пов'язаних вершин (точок), які утворюють сегменти лінії. Кожен сегмент з'єднує дві послідовні вершини, а сукупність сегментів формує лінійний об'єкт.

Основні характеристики лінійних об'єктів:

– наявність початкової та кінцевої точок. Кожна лінія має чітко визначені початок і кінець, навіть якщо вона є частиною більшої полілінії.

– довжина. Лінійні об'єкти мають вимірну довжину, яка обчислюється як сума довжин усіх її сегментів.

– напрямок (азимут). Кожен сегмент, а також лінія в цілому (якщо вона пряма або її можна узагальнити), має певний азимут або орієнтацію щодо сторін світу.

– зв'язність. Лінійні об'єкти можуть бути зв'язані між собою, утворюючи мережі (наприклад, дорожні, річкові, комунікаційні).

Лінійні об'єкти використовуються для представлення широкого спектра географічних явищ, які мають виражену довжину, але не значну ширину чи площу на масштабі відображення. До них належать: транспортні мережі, гідрографічні мережі, комунікації (лінії електропередач, трубопроводи, телефонні кабелі), природні об'єкти, адміністративні межі.

Лінійні об'єкти є фундаментальними для багатьох видів просторового аналізу, оскільки вони представляють зв'язки, потоки та шляхи. З їхньою допомогою можна виконувати мережевий аналіз, який дозволяє знаходити найкоротші шляхи, оптимальні маршрути, зони обслуговування, визначати найближчі об'єкти (наприклад, пожежні станції до місця виклику) та багато

іншого. Аналіз потоків, що досліджує рух ресурсів, транспорту, води або інформації вздовж лінійних мереж. Аналіз щільності ліній, що визначає ділянки з високою або низькою щільністю лінійних об'єктів (наприклад, дорожньої мережі). Аналіз орієнтації (азимутальний аналіз) який дозволяє визначити та провести візуалізацію переважних напрямків лінійних об'єктів, що є критично важливим у геології (напрямок розломів), гідрології (напрямок течій) та інших сферах. Вимірювання, що дозволяє обчислювати довжини лінійних об'єктів або їхніх частин, [1].

Таким чином, лінійні об'єкти в ГІС є не просто географічними представленнями, а потужними інструментами для просторового аналізу, що дозволяють отримувати цінні інсайти щодо зв'язків, руху та орієнтації явищ у реальному світі.

1.4 Види діаграм, їх застосування. Полярний графік

Діаграми є найбільш поширеним і ефективним видом графічного зображення статистичної інформації, що використовуються для аналізу та порівняння даних у більшості сучасних сфер діяльності. Попри різноманіття доступних видів діаграм, вибір найбільш підходящого може бути складним завданням, оскільки універсальна діаграма не може ефективно відобразити всі аспекти та типи даних, що існують. Кожен вид діаграми спроектований для вирішення конкретних візуалізаційних завдань.

Серед основних популярних видів діаграм вирізняють: гістограму, яка використовується для порівняння різних елементів або їх динаміки за певний проміжок часу; лінійну діаграму, що відображає тенденції або прогрес і застосовується для візуалізації багатьох категорій даних, особливо при аналізі даних, зібраних протягом тривалого часу; діаграму з подвійною віссю, яка дозволяє відображати дані з використанням двох осей (X і Y), що є корисним для порівняння різних наборів даних та демонстрації кореляції; кругову діаграму, що відображає статичну частку та співвідношення частин до цілого, показуючи числа у відсотках; та діаграму розсіювання, де значення атрибутів

використовуються як координати X та Y для відображення точок, а їх розподіл показує взаємозв'язок між значеннями. Всі ці діаграми є популярними та зручними для користувачів, їх можна будувати за допомогою QGIS. Однак, іноді навіть цих стандартних типів діаграм недостатньо для представлення специфічних даних, що вимагає інших підходів до візуалізації.

Одним з таких спеціалізованих видів є полярна діаграма, яка зображує багатовимірні дані у вигляді двовимірної діаграми більш ніж трьох змінних, представлених на осях, що виходять з однієї центральної точки, і будується на полярній системі координат. Цей тип діаграм є особливо актуальним для візуалізації спрямованих даних, таких як азимут. Азимут – це один із кутів орієнтування, який застосовується в геології та геодезії. Багато характеристик середовища проживання та поведінки дикої природи мають напрямок. Зазвичай для визначення цього напрямку використовують азимут.

У більшості випадків напрямок або значення азимуту легко визначити із застосуванням математики або з компасом. Ускладнення виникають, якщо цікавлять азимути на великих відстанях. Прямолінійний сегмент на довгій відстані найкраще наближається геодезичною кривою, що представляє собою відстань довкола кола між двома точками. Проблема полягає в тому, що за деякими рідкісними винятками, значення азимуту постійно змінюються в ході геодезичної кривої, [1].

Через круговий характер спрямованих даних зазвичай використовуються кругові ділянки для відображення розподілу. Більшість цих круглих ділянок є варіаціями на стандартній гістограмі, за винятком обернутого в колі так, щоб максимально можливе значення осі X оберталось назад до дотику мінімально можливого значення. Одним з найбільш відомих і історично цікавих видів діаграм є Роза Діаграма, яка називається такою, оскільки набори гістограми нагадують пелюстки квітки.

1.5 Концепція азимуту та його значення у просторовому аналізі

1.5.1 Визначення азимуту та одиниці вимірювання

Азимут – це кут, який вимірюється за годинниковою стрілкою від північного напрямку (або іншого встановленого базового напрямку) до заданого об'єкта чи лінії. Це один з фундаментальних показників, що характеризує орієнтацію об'єкта у просторі. Азимут є ключовим поняттям у геодезії, навігації, картографії та геоінформаційних системах, [1].

Існують різні типи азимутів залежно від використовуваного північного напрямку:

– істинний (географічний) азимут, який вимірюється від істинного географічного Півночі (напрямку на Північний полюс Землі).

– магнітний азимут, який вимірюється від магнітного Півночі (напрямку, на який вказує магнітна стрілка компаса). Він відрізняється від істинного азимута на величину магнітного схилення, рис. 1.1.

– дирекційний кут (румб), який вимірюється від осьового меридіана проєкції (наприклад, центрального меридіана зони UTM) і використовується в проєкційних системах координат.

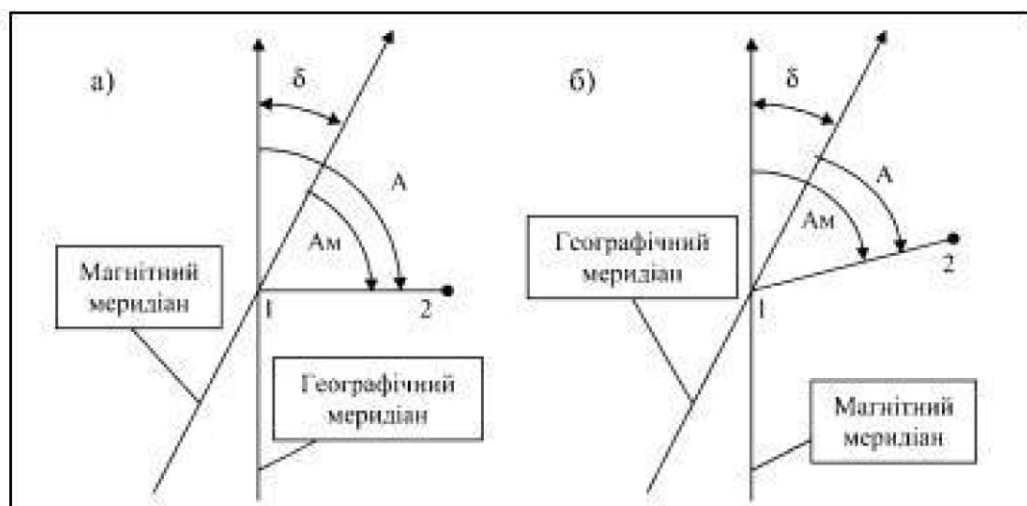


Рисунок 1.1 – Азимут: а) географічний; б) магнітний

Азимут найчастіше вимірюється в градусах. Повний коло становить 360 градусів. Відлік зазвичай починається від Півночі (0° або 360°) і йде за годинниковою стрілкою:

- Північ: 0° (або 360°);
- Схід: 90° ;
- Південь: 180° ;
- Захід: 270° .

Для просторового аналізу в ГІС найбільш поширеною та універсальною одиницею вимірювання азимуту є градуси.

1.5.2 Роль азимуту в дослідженні напрямків лінійних об'єктів

Азимут відіграє критично важливу роль у просторовому аналізі, особливо при дослідженні лінійних об'єктів. Він дозволяє кількісно та якісно оцінювати орієнтацію цих об'єктів, виявляти закономірності та робити висновки щодо процесів, які їх сформували або на них впливають.

Азимут надає числове значення для напрямку лінії, що дозволяє проводити математичний та статистичний аналіз. Одним з найважливіших застосувань азимуту є виявлення домінуючих (переважних) напрямків у сукупності лінійних об'єктів. Азимутальний аналіз дозволяє порівнювати орієнтацію лінійних об'єктів у різних регіонах, геологічних провінціях або в різні історичні періоди. Побудова діаграм розподілу азимутів, таких як кругові гістограми (рози вітрів), є потужним інструментом для візуалізації цих закономірностей. Вони дозволяють наочно представити частоту зустрічальності напрямків, що значно полегшує інтерпретацію складних просторових даних. Отримані азимутальні характеристики можуть бути використані як вхідні параметри для більш складних геоінформаційних моделей, наприклад, при моделюванні водних потоків, плануванні транспортних мереж або аналізі ризиків, [1].

Таким чином, концепція азимуту та його точне визначення дозволяють перетворити візуальні спостереження за напрямками лінійних об'єктів на

кількісні дані, які є основою для глибокого просторового аналізу, виявлення прихованих закономірностей та прийняття обґрунтованих рішень у широкому спектрі наукових та практичних завдань.

1.6 Висновки

Геоінформаційні системи (ГІС) відіграють центральну та незамінну роль у сучасному просторовому аналізі, оскільки вони надають комплексні можливості для збору, зберігання, управління, аналізу та візуалізації географічних даних, що дозволяє інтегрувати різноманітну інформацію та виявляти складні просторові закономірності. QGIS є ефективним, доступним та потужним інструментом для просторового аналізу завдяки своїй відкритій ліцензії та широкому спектру вбудованих функцій. Модульна архітектура QGIS та підтримка розробки плагінів на Python значно розширюють його функціонал, дозволяючи створювати спеціалізовані інструменти, які глибоко інтегровані в середовище QGIS та використовують його API, а також потужні сторонні бібліотеки, такі як Matplotlib.

Лінійні об'єкти є ключовим елементом просторового аналізу, представляючи транспортні, гідрографічні, комунікаційні мережі та інші елементи, що мають протяжність, і є основою для мережевого, потокового та азимутального аналізу. Концепція азимуту є фундаментальною для дослідження напрямків лінійних об'єктів, оскільки азимут надає кількісну характеристику орієнтації, дозволяє виявляти переважні напрямки, проводити порівняльний аналіз та є важливим вхідним параметром для моделювання. Вимірюється він у градусах, за годинниковою стрілкою від Півночі. Візуалізація даних напрямків за допомогою кругових гістограм (рози вітрів) є дуже ефективною, забезпечуючи наочність, дозволяючи виявляти мультимодальність та оцінювати розсіювання даних, що значно полегшує їх інтерпретацію та комунікацію результатів..

Коротко описано про початкові інструменти які потрібні для розробки.

2 АНАЛІЗ ТА ПРОЄКТУВАННЯ ДОДАТКУ ДЛЯ ПОБУДОВИ ДІАГРАМ РОЗПОДІЛУ АЗИМУТІВ

2.1 Аналіз існуючих інструментів та підходів до аналізу азимутів у ГІС

Аналіз азимутів у геоінформаційних системах є важливою задачею для розуміння орієнтації лінійних об'єктів та виявлення прихованих просторових закономірностей. Перед тим, як розпочати проєктування власного додатку для побудови діаграм розподілу азимутів, необхідно провести детальний огляд існуючих інструментів та підходів, доступних у програмному забезпеченні QGIS. Це дозволить ідентифікувати переваги та недоліки поточних рішень, уникнути дублювання функціоналу та визначити унікальні можливості для розроблюваного інструменту.

2.1.1 Огляд вбудованих функцій QGIS

QGIS, як потужна ГІС з відкритим вихідним кодом, пропонує ряд вбудованих функцій, які, хоча і не надають прямого інструменту для побудови кругової гістограми розподілу азимутів, але можуть бути використані як складові частини для вирішення цієї задачі або для отримання вихідних даних.

Один з найпотужніших вбудованих інструментів QGIS для маніпуляцій з атрибутивними даними шарів є «Калькулятор поля (Field Calculator)». Він дозволяє створювати нові поля або оновлювати існуючі за допомогою виразів. У контексті аналізу азимутів, «Калькулятор поля» є незамінним для обчислення азимуту для кожного лінійного сегмента. Завдяки підтримці мови Python, користувач може написати спеціальні функції, що отримують координати початкової та кінцевої точок лінії (`$geometry.start_point().x()`, `$geometry.end_point().y()`) та обчислюють кут між ними за допомогою тригонометричних функцій (наприклад, `atan2` з модуля `math`). Цей метод вимагає від користувача знання синтаксису виразів Python та базових математичних формул, але надає велику гнучкість, [5].

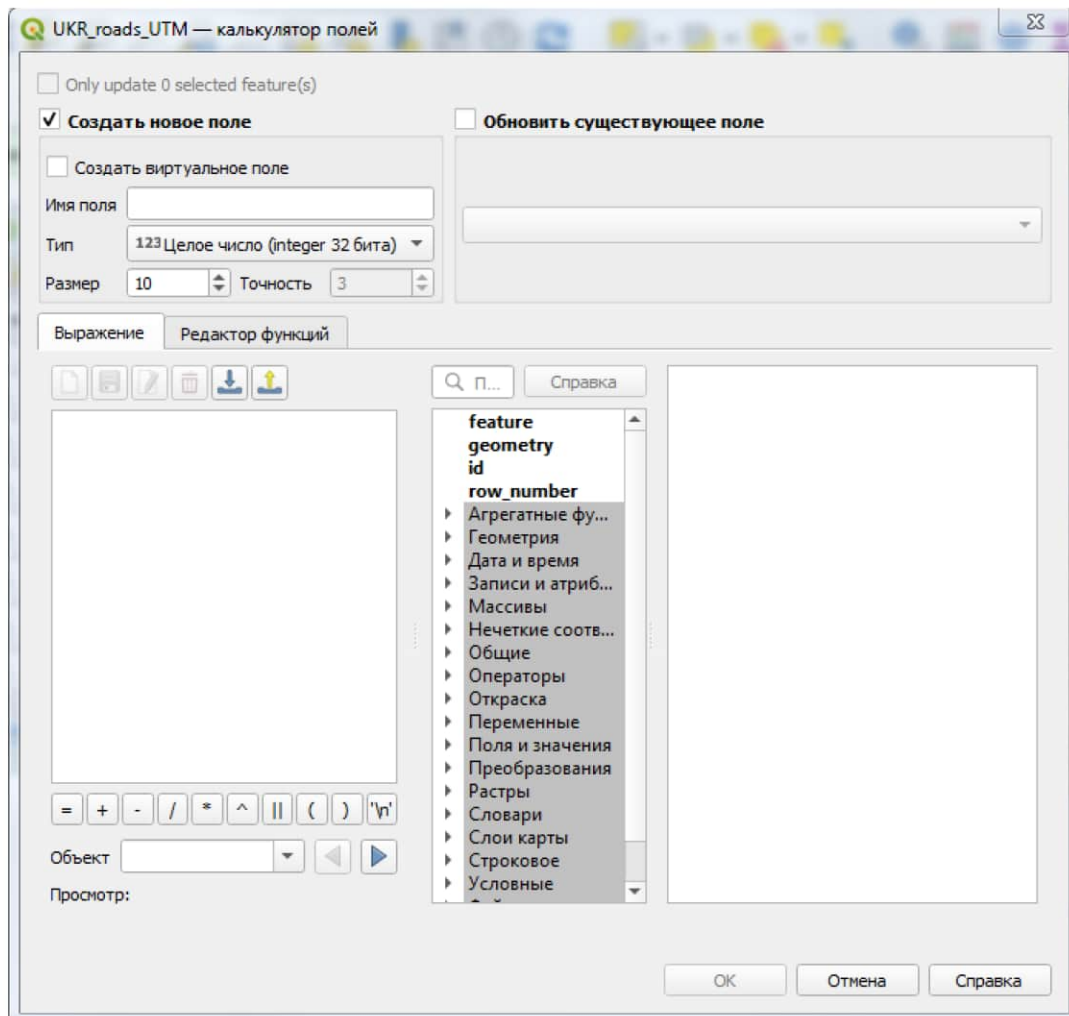


Рисунок 2.1 – «Калькулятор поля (Field Calculator)» в QGIS

Інструменти геообробки (Processing Toolbox), панель інструментів обробки в QGIS інтегрує сотні алгоритмів, включаючи інструменти з таких бібліотек, як GRASS, SAGA, GDAL.

«Розбити лінії по частинам» (Explode lines). Цей інструмент є критично важливим для попередньої підготовки даних, якщо вихідний шар містить полілінії. Він дозволяє розділити багатосегментні лінійні об'єкти на окремі двоточкові сегменти, кожен з яких потім може бути проаналізований індивідуально на предмет азимуту. Це забезпечує точність розрахунків для кожного прямолінійного відрізка.

Інструменти для вимірювання. QGIS має базові інструменти вимірювання, які можуть показувати довжину та кут, але вони призначені для

ручного вимірювання окремих об'єктів, а не для пакетної обробки та аналізу розподілу.

Інструменти візуалізації та символіки. QGIS дозволяє налаштовувати символіку лінійних шарів на основі атрибутів. Хоча це не пряма діаграма розподілу азимутів, користувач може створювати тематичні карти, де лінії з певними діапазонами азимутів будуть відображатися різними кольорами або товщиною, що дає певне візуальне уявлення про їхню орієнтацію. Однак, це не є повноцінною кількісною діаграмою розподілу.

2.1.2 Аналіз наявних плагінів та їх функціоналу

Офіційний репозиторій плагінів QGIS є багатим джерелом розширень, що розробляються спільнотою. Аналіз доступних плагінів, пов'язаних з азимутами та напрямками, показує, що хоча існують інструменти для вимірювання та обчислення азимутів, комплексного рішення для побудови діаграм розподілу напрямків у вигляді кругової гістограми «з коробки» бракує.

Плагін Azimuth Measurement дозволяє користувачеві вимірювати азимут під час малювання лінії. Він зручний для інтерактивного отримання значення кута, але не надає функціоналу для пакетної обробки вже існуючих лінійних об'єктів та, тим більше, не будує діаграм розподілу.

Плагін Azimuth and Distance Calculator, обчислює азимути та відстані для вибраних об'єктів (ліній або полігонів). Він дозволяє додавати обчислені значення до атрибутивної таблиці. Це може бути корисним для отримання числових даних азимутів, аналогічно «Калькулятору поля», але без можливостей візуалізації розподілу, [3].

Плагін AzimuthTool фокусується на створенні нових векторних лінійних шарів на основі заданих азимутів, румбів та відстаней. Його функціонал протилежний завданню аналізу існуючих лінійних об'єктів та їх азимутів.

QGIS має всі необхідні базові компоненти для розрахунку азимутів (Калькулятор поля, інструменти геообробки для розбиття ліній). Але відсутній прямий вбудований інструмент або широкодоступний плагін, який би

автоматично створював кругову гістограму розподілу азимутів з векторного шару. Існуючі плагіни зосереджені на вимірюванні або створенні ліній, а не на статистичній візуалізації розподілу напрямків.

Це обґрунтовує актуальність та необхідність розробки власного додатку, який би об'єднав етапи розрахунку азимутів з їхньою наочною візуалізацією у вигляді кругової гістограми, надаючи користувачам QGIS комплексний та зручний інструмент для просторового аналізу напрямків лінійних об'єктів. Такий інструмент дозволить автоматизувати процес, що наразі вимагає ручного поєднання кількох вбудованих функцій.

2.2 Проектування архітектури додатку (плагіну)

2.2.1 Вибір мови програмування Python

Вибір мови програмування Python для розробки додатку в середовищі QGIS є не лише обґрунтованим, а й стратегічно вигідним рішенням, зумовленим його унікальними перевагами, що роблять його стандартом де-факто для розширення функціоналу цієї геоінформаційної системи, [4, 7].

Насамперед, нативна інтеграція з QGIS є ключовим фактором. QGIS розроблений таким чином, що його ядро (написане на C++) надає потужний та добре документований API (Application Programming Interface), який безпосередньо експонується для Python. Це означає, що розробник може легко взаємодіяти з усіма внутрішніми компонентами QGIS: отримувати доступ до завантажених шарів, маніпулювати географічними об'єктами, читати та змінювати їхні атрибути, викликати інструменти геообробки, керувати системою координат та навіть впливати на елементи користувацького інтерфейсу QGIS. Ця глибока інтеграція дозволяє створювати плагіни, які відчуються як органічна частина самої програми, а не як окремі, незграбні додатки.

По-друге, простота вивчення та використання Python є значною перевагою, особливо для розробників, які можуть не мати великого досвіду у програмуванні, але добре розуміються на ГІС-концепціях. Python відомий

своїм чистим, інтуїтивно зрозумілим синтаксисом та високою читабельністю коду. Це дозволяє зосередитися на логіці просторового аналізу та реалізації функціоналу, а не на складнощах мови програмування чи низькорівневих деталях. Швидкість розробки та прототипування на Python значно вища порівняно з мовами нижчого рівня, такими як C++, що є важливим у динамічному середовищі розробки ГІС-додатків.

По-третє, величезна екосистема бібліотек Python є неперевершеним ресурсом для будь-якого розробника. Для даного додатку, що фокусується на аналізі азимутів та візуалізації, особливу цінність мають:

- `math`: стандартний модуль Python, що містить необхідні математичні функції для точних тригонометричних розрахунків кутів та перетворень (наприклад, `atan2` для обчислення азимуту, `degrees` для перетворення радіан у градуси), [7].

- `numpy`: фундаментальна бібліотека для наукових обчислень у Python. Вона надає високоефективні масиви даних та функції для роботи з ними, що є критично важливим для статистичної обробки великих наборів азимутів (наприклад, для побудови гістограми за допомогою `np.histogram` та перетворення кутів за допомогою `np.radians`), [3].

- `matplotlib`: де-факто стандарт для створення статичних, інтерактивних та анімованих візуалізацій у Python. Ця бібліотека є ідеальним вибором для побудови складних графіків, таких як кругові гістограми (рози вітрів) у полярній системі координат. Її гнучкість дозволяє повністю контролювати вигляд графіка: налаштовувати осі, мітки, кольори, заголовки та інші естетичні елементи для створення професійних та інформативних діаграм, [9].

По-четверте, кросплатформність Python гарантує, що розроблений додаток буде безперебійно працювати в QGIS на різних операційних системах – Windows, Linux та macOS – без необхідності вносити суттєві зміни в код. Це значно розширює коло потенційних користувачів плагіна та спрощує його розповсюдження.

Нарешті, активна спільнота розробників є потужною підтримкою. QGIS має одну з найбільших та найактивніших спільнот розробників серед ГІС з відкритим вихідним кодом. Це означає доступ до величезної кількості документації, навчальних матеріалів, прикладів коду, форумів та каналів підтримки. Також існують численні розробники Python та PyQt, що забезпечує легкість пошуку рішень для будь-яких технічних викликів, що можуть виникнути під час створення плагіна.

У сукупності, ці переваги роблять Python оптимальним вибором для розробки додатку, що не тільки відповідає технічним вимогам до аналізу азимутів та візуалізації, а й забезпечує ефективність, гнучкість та довгострокову підтримку рішення в екосистемі QGIS, [10].

2.2.2 Вибір бібліотек та фреймворків

Для реалізації функціоналу додатку «Побудова Діаграм Розподілу Азимутів» обрано наступні ключові бібліотеки та фреймворки, кожен з яких виконує свою унікальну та незамінну роль:

- 1) PyQt5 – це бібліотека Python-прив'язок до кросплатформного фреймворку Qt. Вона призначена для створення графічного інтерфейсу користувача (GUI). Вона є фундаментом для всіх візуальних елементів плагіна. Також використовується для побудови головного діалогового вікна (QDialog), яке є основним інтерфейсом взаємодії з користувачем, створення та управління елементами керування, такими як, [8]:
 - QComboBox: випадаючі списки для вибору векторних шарів та полів;
 - QPushButton: кнопки, що ініціюють дії (наприклад, «Побудувати діаграму»);
 - QLabel: мітки для відображення інструкцій та текстової інформації;
 - QSpinBox: поле для введення числових значень (наприклад, кількість інтервалів для гістограми);
 - QWidget / QFrame: контейнери, які використовуються для розміщення інших елементів, зокрема, для вбудовування графіка Matplotlib.

Ще одним із її призначень є організація елементів на вікні за допомогою макетів (`QVBoxLayout`, `QHBoxLayout`), що забезпечує адаптивність інтерфейсу до різних розмірів вікна. А також обробки сигналів і слотів – механізму Qt для обробки подій (наприклад, натискання кнопки, зміна вибору у випадяючому списку).

2) `Matplotlib` – це бібліотека Python для побудови графіків. За її допомогою можна створити статичні, анімовані та інтерактивні візуалізації. В нашому додатку `Matplotlib` є візуалізаційним двигуном плагіна. Він використовується для, [9]:

- створення головної фігури (`matplotlib.pyplot.Figure`), яка є полотном для графіка.

- побудови полярної гістограми за допомогою функціоналу полярної проєкції (`add_subplot(projection='polar')`).

- відображення розподілу азимутів за секторами (стовпчиками) на полярному графіку (`ax.bar()`).

- налаштування візуальних аспектів графіка, таких як: встановлення напрямку (Північ, за годинниковою стрілкою), додавання міток осей (наприклад, N, NE, E тощо), встановлення заголовка діаграми, керування кольорами, прозорістю (`alpha`), межами стовпчиків (`edgcolor`).

- інтеграція з PyQt5. За допомогою `FigureCanvasQTAgg` фігура `Matplotlib` вбудовується безпосередньо в PyQt5-віджет. Це дозволяє відображати графік у діалоговому вікні.

- Інтерактивність - `NavigationToolbar2QT` надає стандартні інструменти `Matplotlib` для взаємодії з графіком (масштабування, панорамування, збереження у файл).

3) `QGIS API` – це набір Python-модулів, програмний доступ до функціоналу `QGIS`, маніпуляція з геопросторовими даними та взаємодія з інтерфейсом `QGIS`. В нашому випадку це:

- `qgis.core` - основний модуль для роботи з геопросторовими даними та проєктом; `QgsProject` - представляє поточний проєкт `QGIS`. Використовується

для отримання списку всіх завантажених шарів (`QgsProject.instance().mapLayers()`), доступу до конкретних шарів за їхнім ID (`QgsProject.instance().mapLayer()`); `QgsVectorLayer` представляє векторні шари. Використовується для визначення типу шару (`layer.type()`); Визначення типу геометрії (`layer.geometryType()`), наприклад, `QgsWkbTypes.LineGeometry`, `QgsWkbTypes.MultiLineGeometry`); ітерації по об'єктах шару (`layer.getFeatures()`), [11].

`QgsFeature`, представляє окремий об'єкт (рядок у таблиці атрибутів + геометрія) у векторному шарі. Він дозволяє отримати геометрію об'єкта (`feature.geometry()`), де `QgsGeometry` представляє геометрію об'єкта. Він використовується для визначення типу геометрії (`geom.type()`) та її отримання у потрібному форматі (`geom.asPolyline()`, `geom.asMultiPolyline()`).

`QgsPointXY`, представляє точку з X та Y координатами. Використовується для визначення кінців лінійних сегментів.

`QgsDistanceArea` – це клас для виконання точних геодезичних розрахунків, таких як відстані та азимути. Використовується для обчислення азимуту між двома точками (`distance_area.bearing(p1, p2)`), враховуючи еліпсоїд Землі.

– `qgis.gui`, це модуль, що містить класи для інтеграції GUI-компонентів QGIS та взаємодії з вікном карти.

`QgisInterface` надає доступ до головного вікна QGIS, його меню, тулбарів, панелі повідомлень (`iface.messageBar()`) та інших елементів інтерфейсу. Використовується для додавання плагіна в меню, відображення повідомлень користувачу тощо.

– `qgis.utils` містить допоміжні функції, зокрема для доступу до `iface`.

Структурна схема додатку наведена на рисунку 2.2:

Azimuth Distribution Analysis

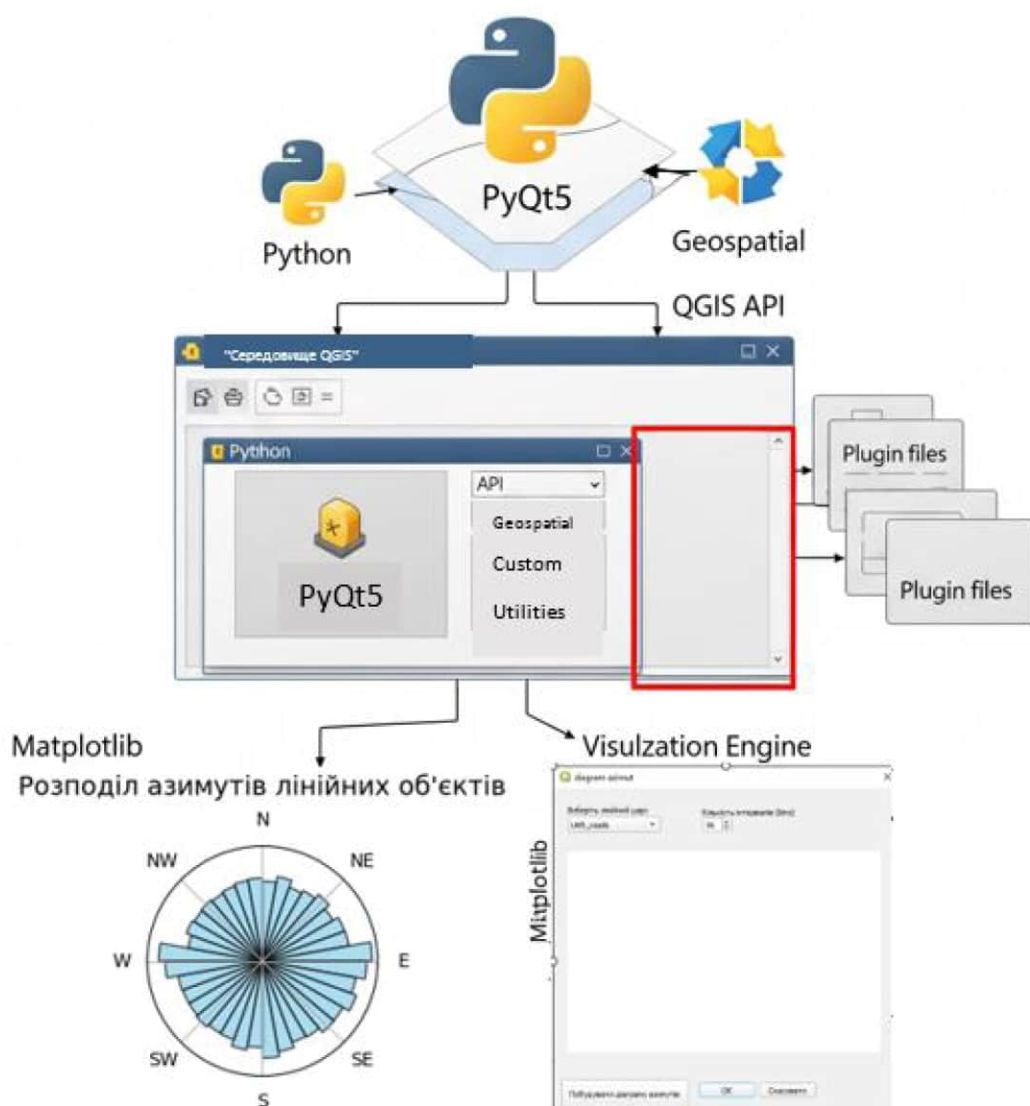


Рисунок 2.2 – Структурна схема додатку (плагіну)

Ця діаграма візуалізує архітектуру та взаємодію ключових технологій, що використовуються у розробленому інструменті для аналізу розподілу азимутів. Вона показує, як різні програмні компоненти співпрацюють для забезпечення функціоналу плагіна QGIS.

Верхній рівень (Базові технології та Інтерфейси):

Python. Слугує основною мовою програмування, яка забезпечує логіку, обчислення та загальну інтеграцію системи.

PyQt5. Використовується як фреймворк для розробки графічного інтерфейсу користувача, що створює всі візуальні елементи плагіна та QGIS загалом.

Геопросторовий (Geospatial). Цей термін вказує на використання географічних даних та відповідних бібліотек, які QGIS інтегрує (наприклад, GDAL/OGR для роботи з форматами даних, GEOS для геометричних операцій).

QGIS API. Це інтерфейс програмування додатків QGIS, який дозволяє Python-коду плагіна взаємодіяти безпосередньо з ядром QGIS, отримувати доступ до шарів, об'єктів, інструментів геообробки та керувати інтерфейсом.

Середній рівень (Середовище QGIS та Плагін):

Зображено вікно Середовища QGIS (або "Інтерфейс QGIS"), що символізує основне вікно програми QGIS, в якому виконується плагін.

Внутрішнє вікно з написом «Python» та «PyQt5» представляє загальне середовище виконання Python-коду, що використовує PyQt5 для створення свого інтерфейсу. Елементи на ньому, такі як «API», «Геопросторовий» і т.д. символізують різні функціональні блоки та категорії інструментів, до яких Python-скрипт може отримати доступ.

Файли плагіна. Це окремі модулі, що містять код вашого плагіна. Він завантажується та виконується в середовищі QGIS.

Нижній рівень (Візуалізація та Результати):

Рушій візуалізації. Це компонент, відповідальний за перетворення оброблених даних на графічні зображення.

Matplotlib. Є ключовою бібліотекою, яка реалізує функціонал рушія візуалізації. Вона використовується для побудови складних графіків, зокрема діаграми розподілу азимутів.

Діаграма розподілу азимутів представляє собою кінцевий візуальний результат аналізу – графік «Розподіл азимутів лінійних об'єктів», що наочно демонструє орієнтацію лінійних об'єктів.

2.3 Діаграма класів

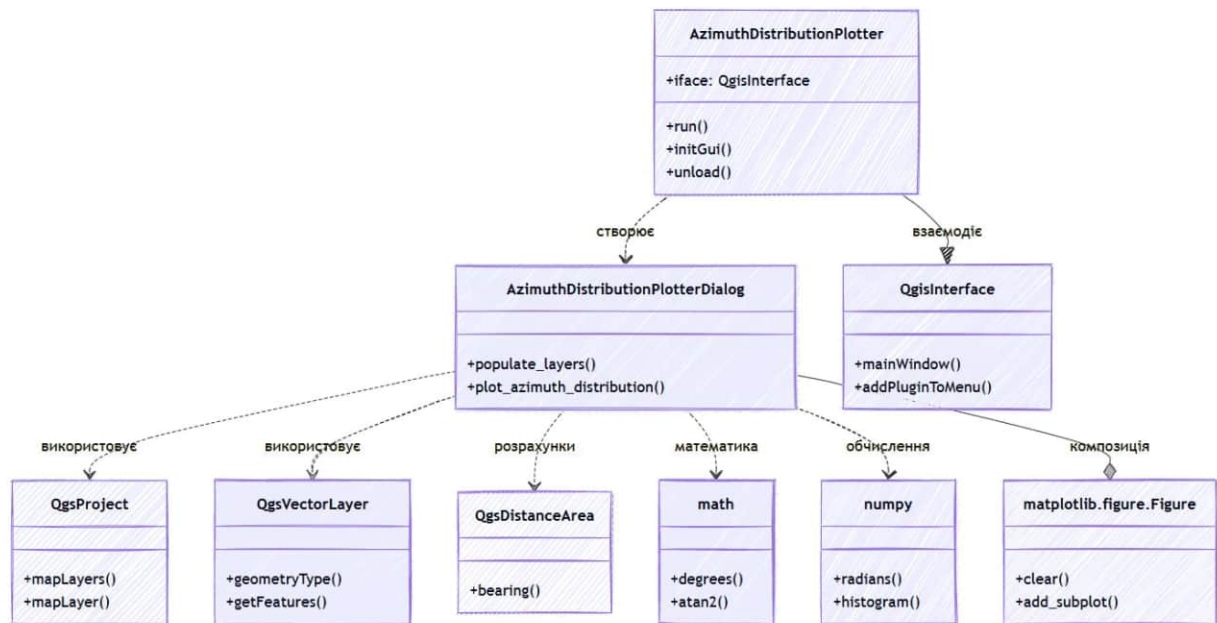


Рисунок 2.3 – Діаграма класів додатку (плагіну)

Ця UML-діаграма класів (рис. 2.3) візуалізує статичну структуру основних компонентів плагіна «Діаграма розподілу азимутів», показуючи їхні атрибути, методи та взаємозв'язки для повнішого розуміння архітектури коду. У центрі діаграми знаходяться два розроблені класи: `AzimuthDistributionPlotter` та `AzimuthDistributionPlotterDialog`, які є ключовими для функціонування плагіна.

Клас `AzimuthDistributionPlotter` є головним класом плагіна, що слугує точкою входу для QGIS. Він відповідає за ініціалізацію плагіна в середовищі QGIS, додавання його до меню та панелей інструментів, а також за керування життєвим циклом діалогового вікна. Цей клас має такі ключові методи, як `run()`, що запускає діалогове вікно; `initGui()`, що ініціалізує графічний інтерфейс плагіна в QGIS; та `unload()`, що видаляє елементи плагіна з GUI QGIS при його вивантаженні. Клас `AzimuthDistributionPlotter` створює екземпляр класу `AzimuthDistributionPlotterDialog`, і це є ключовим зв'язком, що показує залежність головного плагіна від його діалогового інтерфейсу. Крім того, `AzimuthDistributionPlotter` взаємодіє з класом `QgisInterface`, що є

інтерфейсом QGIS API і надає доступ до функцій головного вікна програми, його меню та панелей інструментів.

Клас `AzimuthDistributionPlotterDialog` представляє собою основний графічний інтерфейс користувача плагіна. Це діалогове вікно відповідає за збір вхідних даних від користувача, виконання основних розрахунків азимутів та візуалізацію результатів. Серед його ключових методів є такі як: `populate_layers()`, що заповнює список шарів; та `plot_azimuth_distribution()`, що виконує розрахунок азимутів та побудову діаграми. Клас `AzimuthDistributionPlotterDialog` успадковує від класу `QDialog`, який є базовим класом для всіх діалогових вікон у фреймворку `PyQt5`. Він також успадковує від класу `Ui_AzimuthDistributionPlotterDialogBase`, що є згенерованим класом з UI-дизайнера, який відповідає за налаштування візуальних елементів інтерфейсу.

Залежності та взаємодії з QGIS API та бібліотеками:

Клас `AzimuthDistributionPlotterDialog` має численні залежності від ключових компонентів QGIS API та зовнішніх бібліотек, що дозволяє йому виконувати свої функції. Він використовує клас `QgsProject`, що представляє поточний QGIS-проект і надає доступ до його шарів. Для роботи з векторними даними `AzimuthDistributionPlotterDialog` використовує клас `QgsVectorLayer`, що є моделлю векторного шару. Розрахунки азимутів виконуються за допомогою класу `QgsDistanceArea`, який є обчислювальним модулем QGIS API для точних геодезичних розрахунків. Для математичних операцій, необхідних при розрахунку та нормалізації кутів, плагін використовує стандартний модуль `Python math`. Обробка числових даних, зокрема для побудови гістограм, здійснюється за допомогою бібліотеки `numpy`. Для візуалізації `AzimuthDistributionPlotterDialog` компонує `matplotlib.figure.Figure`, що є основним об'єктом для графіка. Ця фігура, своєю чергою, компонує осі для побудови діаграми.

Ця діаграма чітко візуалізує структуру нашого плагіна, показуючи, як розроблені класи взаємодіють з ключовими компонентами QGIS API, `PyQt` та

бібліотеки Matplotlib для виконання своєї основної функції – аналізу та візуалізації розподілу азимутів.

2.4 Висновки

У рамках другого розділу було проведено всебічний аналіз та проєктування додатку для побудови діаграм розподілу азимутів у програмному середовищі QGIS. Детальний огляд існуючих інструментів та підходів до аналізу азимутів у ГІС показав, що попри наявність вбудованих функцій QGIS, таких як Калькулятор поля та інструменти геообробки ("Розбити лінії по частинам"), вони не надають комплексного рішення для автоматичної візуалізації розподілу азимутів у вигляді кругової гістограми. Аналіз наявних плагінів QGIS також виявив відсутність інструменту, який би поєднував автоматичний розрахунок азимутів безпосередньо з геометрії об'єктів та їх візуалізацію у вигляді рози вітрів. Це обґрунтувало актуальність та необхідність розробки власного додатку.

Проєктування архітектури додатку базувалося на обґрунтованому виборі мови програмування Python, що є стандартом для розробки плагінів QGIS завдяки його нативній інтеграції з QGIS API, простоті вивчення, кросплатформності та великій екосистемі бібліотек. Для реалізації функціоналу були обрані ключові бібліотеки та фреймворки: PyQt5 для створення графічного інтерфейсу користувача, Matplotlib для потужної візуалізації (зокрема, полярних графіків) та QGIS API (модулі qgis.core, qgis.gui, qgis.utils) для взаємодії з геопросторовими даними та середовищем QGIS. Структурна схема додатку передбачає модульну архітектуру, де чітко розмежовані відповідальності між компонентами, такими як модуль плагіна, діалогове вікно, обчислювальні модулі (за участю QgsDistanceArea, math, NumPy) та модуль візуалізації.

Алгоритм розрахунку азимуту для лінійних об'єктів включає етапи підготовки даних, такі як обов'язкова перевірка та перепроєкція шару у відповідну проєкційну систему координат (наприклад, UTM), розбиття

поліліній на окремі сегменти, та програмне отримання координат початкової та кінцевої точок кожного сегмента. Математичний апарат обчислення азимуту базується на використанні функції `math.atan2` та `QgsDistanceArea.bearing()`, забезпечуючи точний розрахунок кута та його нормалізацію до діапазону 0-360 градусів. Проєктування інтерфейсу користувача додатку зосереджено на розробці інтуїтивно зрозумілого макету діалогового вікна з чіткими елементами керування (`QComboBox`, `QSpinBox`, `QPushButton`), що дозволяють користувачеві легко вибирати шар, поля координат та ініціювати побудову діаграми. Розміщення `Matplotlib`-графіка з інтерактивною панеллю інструментів у вікні забезпечує зручну візуалізацію результатів. Загальне середовище розробки включає `QGIS LTR`-версії, інтегрований `Python` та необхідні бібліотеки, встановлення яких забезпечується через `OSGeo4W Shell`.

Таким чином, у даному розділі було повністю проаналізовано поточний стан інструментарію та розроблено детальну архітектуру та алгоритми для створення ефективного та зручного плагіна `QGIS` для аналізу та візуалізації розподілу азимутів лінійних об'єктів.

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ДОДАТКУ

3.1 Опис середовища розробки та встановлення необхідних компонентів

Для розробки додатку (плагіна) для QGIS, який виконує розрахунок та візуалізацію азимутів лінійних об'єктів, використовується інтегроване середовище, що складається з кількох ключових компонентів.

Python - це високорівнева інтерпретована мова програмування. Ми обрали її для реалізації проекту завдяки нашому досвіду роботи з нею, простоті вивчення та великому асортименту сторонніх пакетів. Використовувалася версія 3.10.

1. Програмне забезпечення QGIS:

QGIS є основною платформою, у якій функціонуватиме розроблений плагін. Він надає доступ до геопросторових даних, шарів, проекту та основного інтерфейсу, з яким взаємодіятиме плагін через QGIS API.

QGIS завантажується безкоштовно з офіційного сайту <https://qgis.org/>. Встановлення здійснюється за допомогою інсталятора, який включає всі необхідні компоненти, зокрема інтегровану версію Python.

2. Середовище виконання Python:

Python є мовою програмування, на якій розробляються плагіни QGIS. QGIS постачається зі своєю власною інтегрованою версією Python, що є найкращим варіантом для забезпечення сумісності.

Зазвичай, окрема установка Python не потрібна, оскільки вона входить до складу інсталятора QGIS. Якщо виникає потреба в встановленні додаткових бібліотек, це робиться через `pip` у консолі OSGeo4W Shell, що постачається з QGIS.

3. Редактор коду / Інтегроване середовище розробки (IDE):

Редактор коду використовується для написання та редагування файлів `.py` та `.txt`, які складають плагін.

4. Необхідні бібліотеки Python:

– PyQt5. Прив'язки Python до бібліотеки Qt, що є основою GUI QGIS. Встановлюється разом з QGIS;

– Matplotlib - бібліотека для побудови статичних, інтерактивних та анімованих графіків;

– Модуль math. Стандартний модуль Python, що містить математичні функції (для тригонометричних розрахунків). Не потребує окремого встановлення.

Процес налаштування середовища розробки:

– встановлення QGIS (включаючи Python та PyQt5);
 – встановлення обраного редактора коду/IDE;
 – перевірка наявності matplotlib у Python-середовищі QGIS. За потреби, встановлення matplotlib через pip в OSGeo4W Shell;

– створення структури папок плагіна (azimuth_distribution/, azimuth_distribution.py, metadata.txt, icon/);

– розміщення папки плагіна в директорії модулів QGIS.

Таким чином, середовище розробки для плагіна QGIS є комплексним, але добре інтегрованим. Воно базується на стабільній версії QGIS та його інтегрованому Python-середовищі, доповненому спеціалізованими бібліотеками для GUI та візуалізації, що забезпечує всі необхідні інструменти для ефективної розробки.

3.2 Моделювання бізнес-процесу функціонування додатку для побудови діаграм розподілу азимутів

Контекстна діаграма бізнес-процесу (рис.3.1) демонструє взаємодію програми з зовнішніми сутностями та основні функції, які вона виконує.

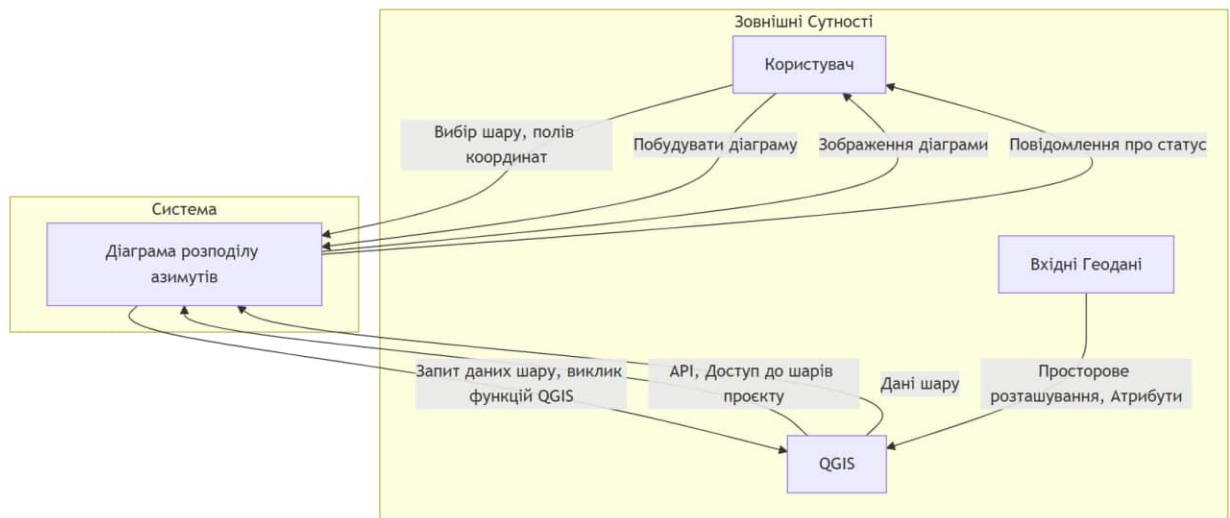


Рисунок 3.1 – Контекстна діаграма додатку для побудови діаграм розподілу азимутів

На діаграмі зображені наступні ключові компоненти та їхні взаємозв'язки:

1. Зовнішні Сутності (External Entities):

Це об'єкти або системи, які взаємодіють з нашим додатком, але знаходяться поза його прямим контролем. На діаграмі вони представлені прямокутними блоками:

Користувач – це людина, яка взаємодіє з плагіном. Користувач є ініціатором дій, надає вхідні дані та отримує вихідні результати.

QGIS (платформа) - це основне програмне середовище, в якому функціонує ваш плагін. QGIS надає плагіну доступ до геопросторових даних, свого API та інших функцій.

Вхідні Геодані (Шар ліній) - це джерело даних, необхідних для аналізу (в нашому випадку файл UKR_roads.shp). Він містить просторове розташування та атрибути лінійних об'єктів, які завантажуються в QGIS.

2. Система (System):

Це безпосередньо розроблений додаток, представлений як єдиний прямокутний блок.

Плагін "Діаграма розподілу азимутів" - це центральний функціональний блок, що інкапсулює всю внутрішню логіку плагіна – від збору вхідних даних та обчислення азимутів до побудови діаграми та її візуалізації.

3. Потоки Даних (Data Flows):

Стрілки на діаграмі показують напрямок обміну інформацією між сутностями та системою.

Користувач → Плагін:

- вибір шару, полів координат: користувач передає плагіну інформацію про те, який шар і які поля використовувати для аналізу;

- команда «Побудувати діаграму»: користувач ініціює процес аналізу та візуалізації.

Плагін → Користувач:

- зображення діаграми: плагін відображає згенеровану діаграму користувачеві;

- повідомлення про статус: плагін надає зворотний зв'язок користувачеві (попередження, помилки, підтвердження виконання).

QGIS (платформа) → Плагін:

- API, доступ до шарів проєкту: QGIS надає плагіну програмний інтерфейс для доступу до даних та функцій проєкту;

- дані шару: QGIS передає плагіну фактичні дані (геометрію та атрибути) з вибраного шару.

Плагін → QGIS (платформа):

- запит даних шару, виклик функцій QGIS: плагін робить запити до QGIS API для отримання необхідної інформації та використання його сервісів.

Вхідні Геодані → QGIS (платформа):

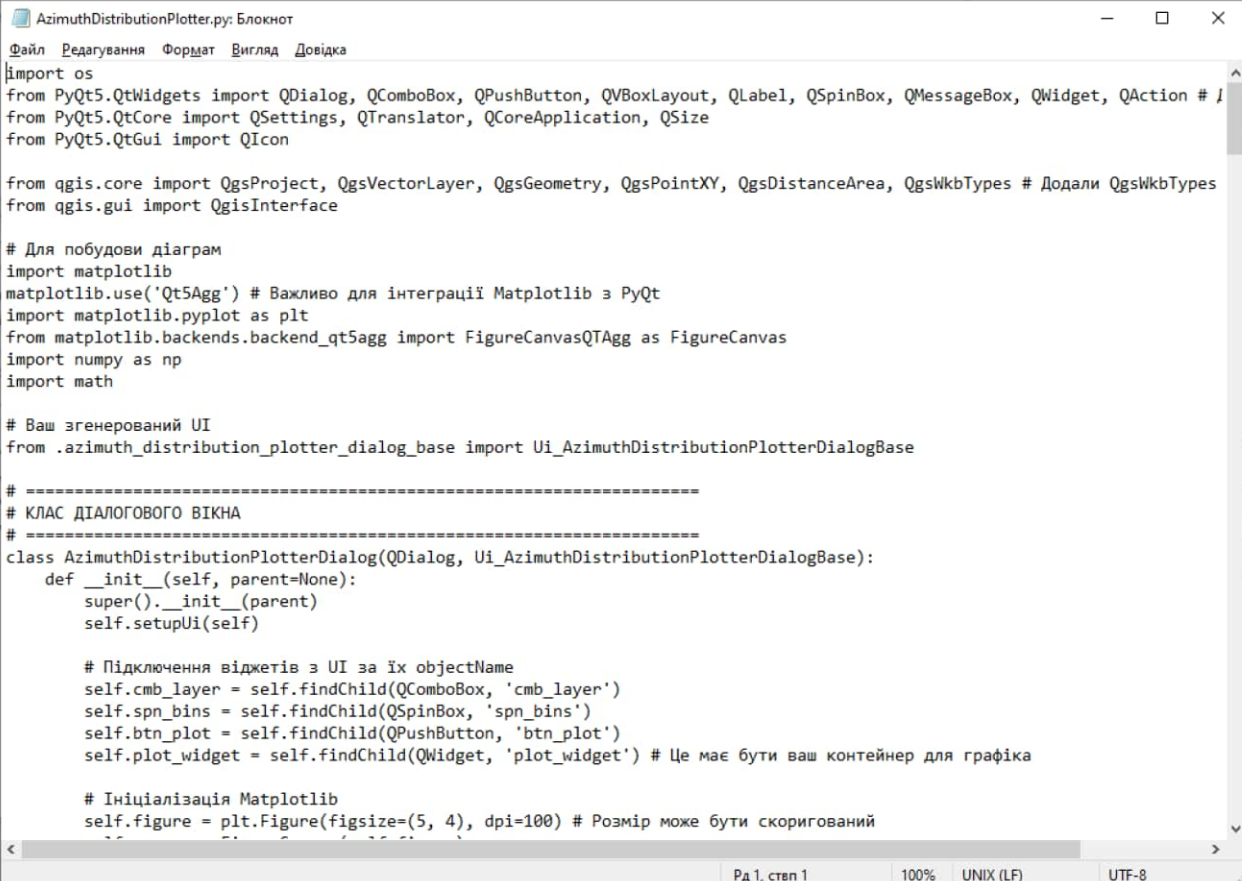
- просторове розташування, атрибути: геодані спочатку завантажуються в QGIS, який потім робить їх доступними для плагіна.

Ця Контекстна діаграма надає чітке та високорівневе уявлення про те, як плагін взаємодіє зі своїм зовнішнім оточенням, візуалізуючи основні входи, виходи та процеси обміну інформацією.

3.3 Створення плагіну для QGIS

Створення плагіну в QGIS складається з декількох ключових етапів, починаючи від підготовки робочого середовища до написання коду та його налагодження.

Основна логіка плагіну реалізується в файлі `azimuth_distribution_plotter.py`. У цьому файлі визначається клас діалогового вікна, який успадковується від `QDialog` та згенерованого класу `Ui_AzimuthDistributionPlotterDialogBase` (що відповідає за дизайн інтерфейсу). В ініціалізаторі діалогового вікна відбувається налаштування віджетів (таких як `QComboBox` для вибору шару, `QSpinBox` для кількості інтервалів, `QPushButton` для запуску розрахунків) та ініціалізація `Matplotlib` для побудови графіків (рис.3.3). Зокрема, створюється об'єкт `Figure` та `FigureCanvas`, який вбудовується в `QWidget` для відображення діаграми.



```

AzimuthDistributionPlotter.py: Блокнот
Файл Редагування Формат Вигляд Довідка
import os
from PyQt5.QtWidgets import QDialog, QComboBox, QPushButton, QVBoxLayout, QLabel, QSpinBox, QMessageBox, QWidget, QAction # I
from PyQt5.QtCore import QSettings, QTranslator, QCoreApplication, QSize
from PyQt5.QtGui import QIcon

from qgis.core import QgsProject, QgsVectorLayer, QgsGeometry, QgsPointXY, QgsDistanceArea, QgsWkbTypes # Додали QgsWkbTypes
from qgis.gui import QgsInterface

# Для побудови діаграм
import matplotlib
matplotlib.use('Qt5Agg') # Важливо для інтеграції Matplotlib з PyQt
import matplotlib.pyplot as plt
from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg as FigureCanvas
import numpy as np
import math

# Ваш згенерований UI
from .azimuth_distribution_plotter_dialog_base import Ui_AzimuthDistributionPlotterDialogBase

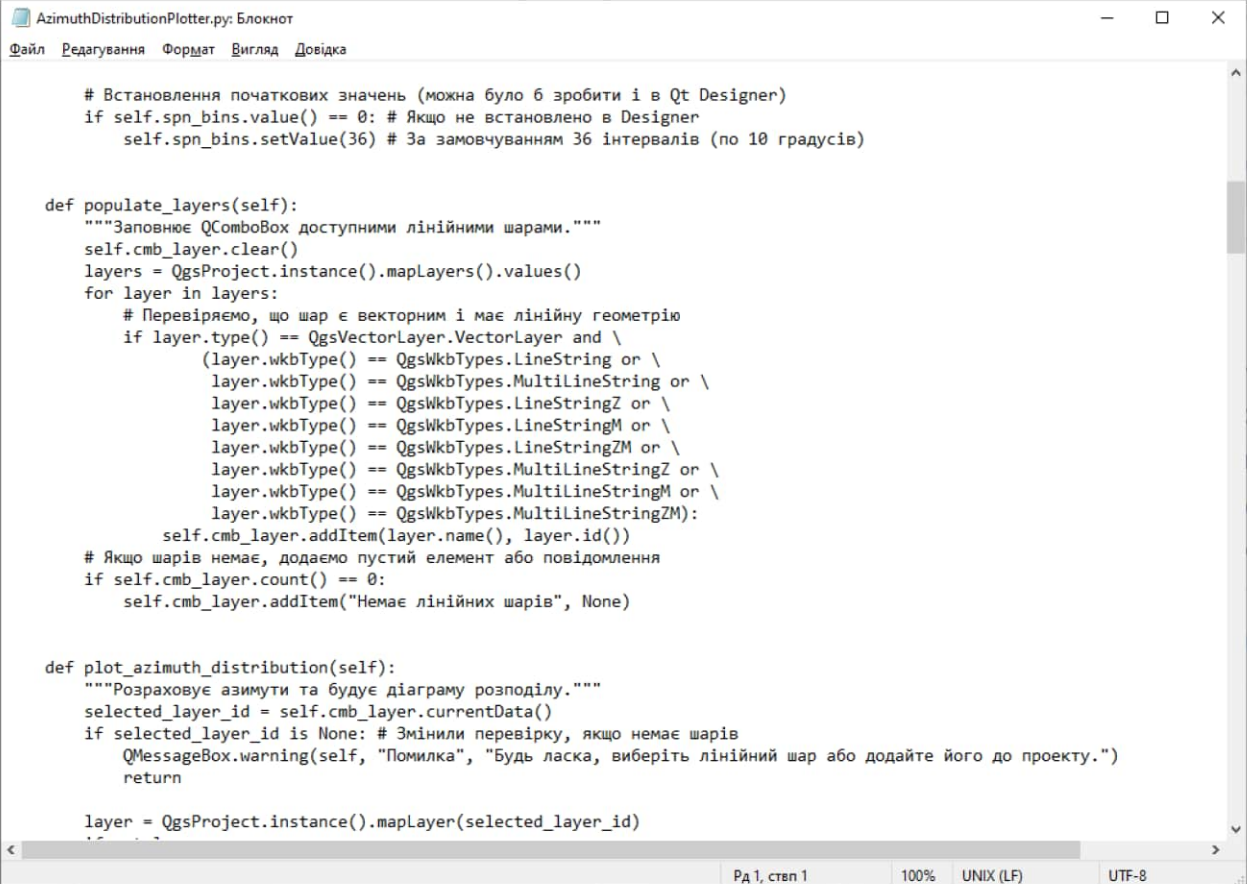
# =====
# КЛАС ДІАЛОГОВОГО ВІКНА
# =====
class AzimuthDistributionPlotterDialog(QDialog, Ui_AzimuthDistributionPlotterDialogBase):
    def __init__(self, parent=None):
        super().__init__(parent)
        self.setupUi(self)

        # Підключення віджетів з UI за їх objectName
        self.cmb_layer = self.findChild(QComboBox, 'cmb_layer')
        self.spn_bins = self.findChild(QSpinBox, 'spn_bins')
        self.btn_plot = self.findChild(QPushButton, 'btn_plot')
        self.plot_widget = self.findChild(QWidget, 'plot_widget') # Це має бути ваш контейнер для графіка

        # Ініціалізація Matplotlib
        self.figure = plt.Figure(figsize=(5, 4), dpi=100) # Розмір може бути скоригований
  
```

Рисунок 3.3 – Фрагмент файлу `azimuth_distribution_plotter.py`

Однією з ключових функцій плагіну є `populate_layers`, яка відповідає за заповнення випадального списку доступними лінійними шарами. У ній відбувається ітерація по всіх шарах проекту QGIS, і перевіряється, чи є шар векторним (`QgsVectorLayer.VectorLayer`) та чи його тип геометрії відповідає лінійним об'єктам. Для цього використовуються такі константи, як `QgsWkbTypes.LineString` та `QgsWkbTypes.MultiLineString` (а також їх 3D/М варіанти), перевіряючи їх через `layer.wkbType()` для коректного розпізнавання. Якщо лінійних шарів не знайдено, в комбо-боксі відображається повідомлення «Немає лінійних шарів» (рис.3.4).



```

AzimuthDistributionPlotter.py: Блокнот
Файл  Редагування  Формат  Вигляд  Довідка

# Встановлення початкових значень (можна було б зробити і в Qt Designer)
if self.spn_bins.value() == 0: # Якщо не встановлено в Designer
    self.spn_bins.setValue(36) # За замовчуванням 36 інтервалів (по 10 градусів)

def populate_layers(self):
    """Заповнює QComboBox доступними лінійними шарами."""
    self.cmb_layer.clear()
    layers = QgsProject.instance().mapLayers().values()
    for layer in layers:
        # Перевіряємо, що шар є векторним і має лінійну геометрію
        if layer.type() == QgsVectorLayer.VectorLayer and \
            (layer.wkbType() == QgsWkbTypes.LineString or \
             layer.wkbType() == QgsWkbTypes.MultiLineString or \
             layer.wkbType() == QgsWkbTypes.LineStringZ or \
             layer.wkbType() == QgsWkbTypes.LineStringM or \
             layer.wkbType() == QgsWkbTypes.LineStringZM or \
             layer.wkbType() == QgsWkbTypes.MultiLineStringZ or \
             layer.wkbType() == QgsWkbTypes.MultiLineStringM or \
             layer.wkbType() == QgsWkbTypes.MultiLineStringZM):
            self.cmb_layer.addItem(layer.name(), layer.id())
    # Якщо шарів немає, додаємо пустий елемент або повідомлення
    if self.cmb_layer.count() == 0:
        self.cmb_layer.addItem("Немає лінійних шарів", None)

def plot_azimuth_distribution(self):
    """Розраховує азимуті та будує діаграму розподілу."""
    selected_layer_id = self.cmb_layer.currentData()
    if selected_layer_id is None: # Змінили перевірку, якщо немає шарів
        QMessageBox.warning(self, "Помилка", "Будь ласка, виберіть лінійний шар або додайте його до проекту.")
        return

    layer = QgsProject.instance().mapLayer(selected_layer_id)

```

Рисунок 3.4 – Фрагмент коду для перевірки наявності та якості лінійних шарів

Основні обчислення та побудова діаграми виконуються у функції `plot_azimuth_distribution`. Вона отримує обраний лінійний шар, виконує додаткову перевірку його геометрії (також використовуючи `layer.wkbType()`), а потім ітерується по кожному об'єкту шару. Для кожного лінійного об'єкта (або

частини мультілінійного об'єкта) розраховується азимут кожного сегмента за допомогою `QgsDistanceArea().bearing()`, який повертає значення в радіанах. Отримані азимуті конвертуються в градуси та нормалізуються до діапазону $[0, 360]$. Після цього дані готуються для побудови полярної гістограми (розидіаграми) за допомогою `numpy.histogram` та `matplotlib.pyplot`. Графік налаштовується (наприклад, розташування півночі, напрямок за годинниковою стрілкою) та відображається на полотні `FigureCanvas`. Лістинг програми наведений у Додатку А.

Щоб скористатися плагіном «Побудова Діаграм Азимутів» у QGIS, необхідно виконати кілька кроків:

1) Завантажити у програмі QGIS необхідні дані, а сама шейп-файл з лінійними об'єктами `UKR_roads.shp`, рис.3.5:

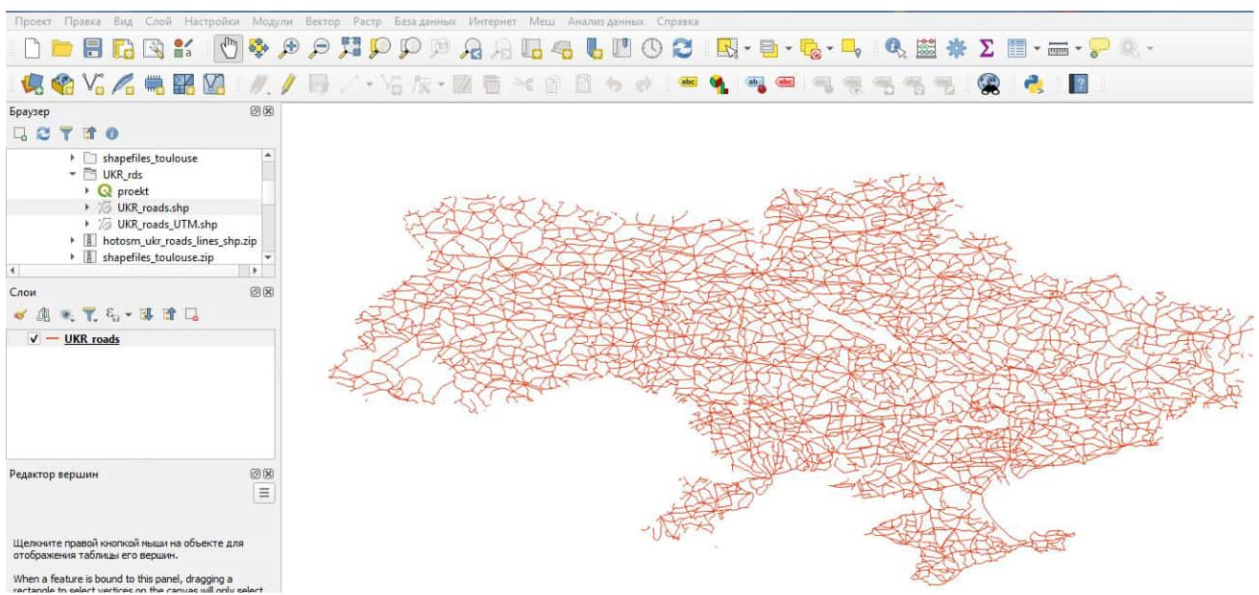


Рисунок 3.5 – Вхідний шейп-файл з лінійними об'єктами `UKR_roads.shp`

2) У Властивостях шару перевірте систему координат. У вхідному файлі вона географічна, рис. 3.6:

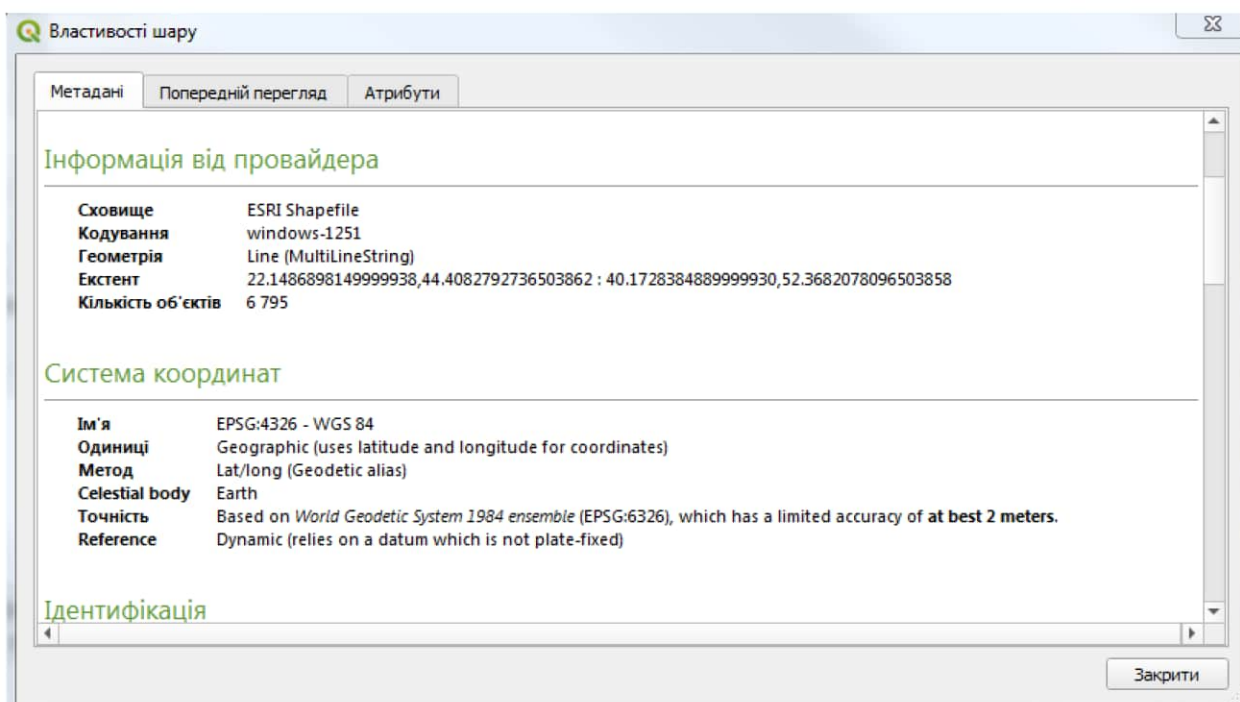


Рисунок 3.6 – Визначення системи координат шейп-файлу з лінійними об'єктами UKR_roads.shp

3) Здійснить перепроєкцію даних, для цього натисніть правою кнопкою миші на нашому шарі в панелі "Шари" вибираємо «Експортувати» (Export) – «Зберегти об'єкти як...». В налаштуваннях збереження виберіть необхідну вам проєкцію, для України підходять зони UTM 35N (EPSG:32635), 36N (EPSG:32636), 37N (EPSG:32637), рис. 3.7:

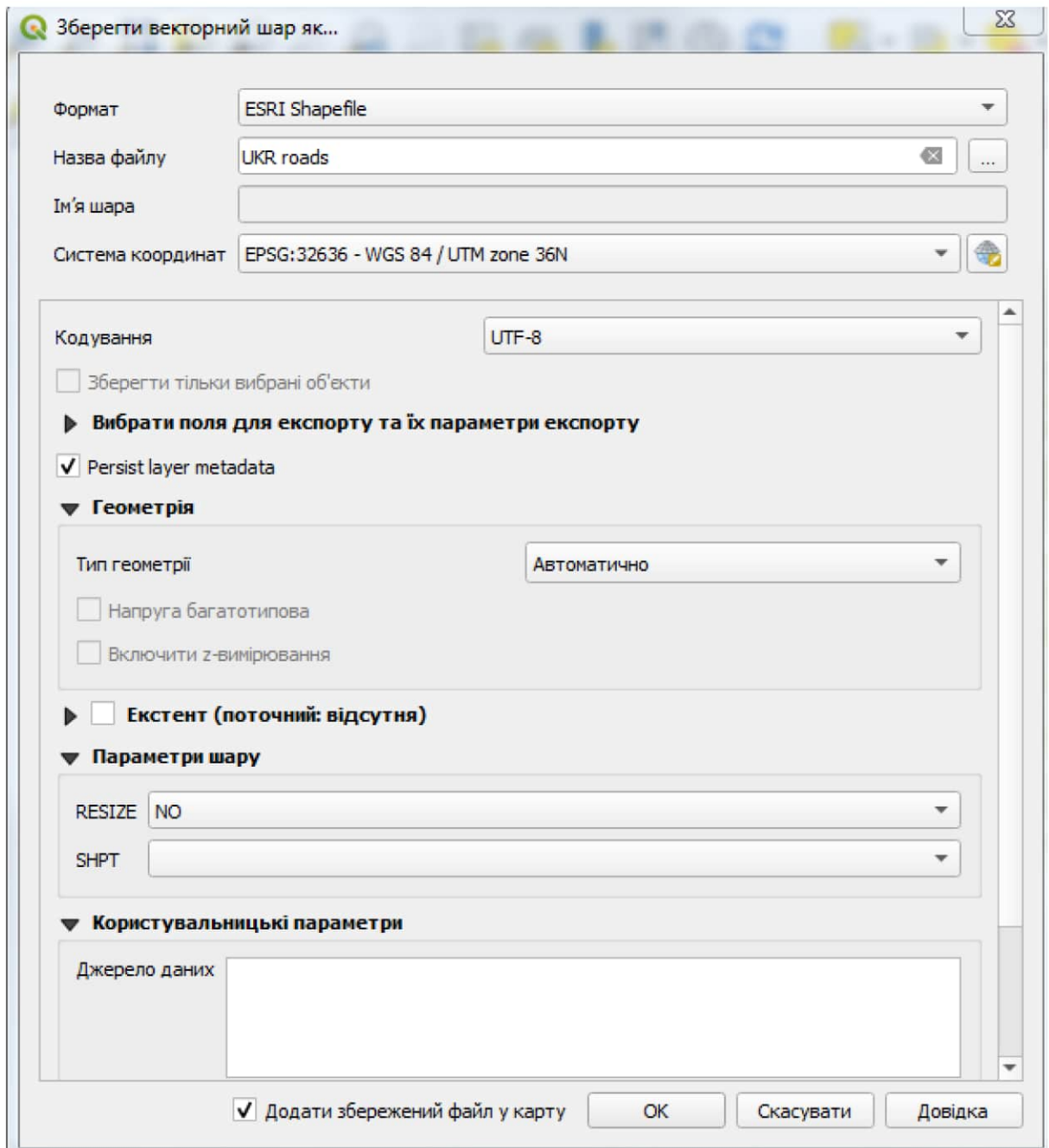


Рисунок 3.7 – Зміна проєкції вхідного шейп-файлу

4) Переконайтеся, що у нашому проєкті QGIS завантажений лінійний векторний шар. Це може бути, наприклад, шар доріг, річок або будь-яких інших лінійних об'єктів. Плагін розроблений для аналізу азимутів саме лінійних геометрій. Важливо, щоб тип геометрії шару був визначений як Line або MultiLineString. В нашій роботі – це карта доріг України.

5) Відкрийте меню «Плагіни» у верхній панелі QGIS. Знайдіть у списку ваш плагін, який називається «Побудова Діаграм Азимутів». Клацніть на

ньому, а потім виберіть опцію «Побудувати діаграму азимутів». Це відкриє діалогове вікно плагіна, рис. 3.8:

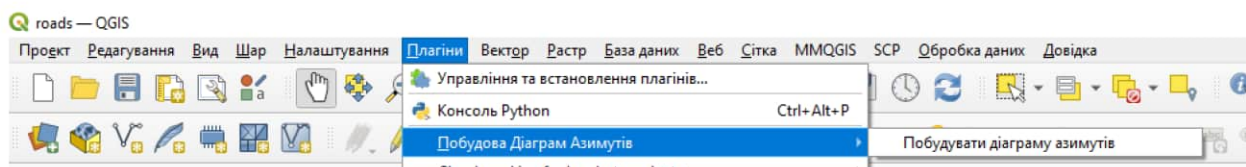


Рисунок 3.8 – Відкриття діалогового вікна плагіна

б) Відкриється діалогове вікно під назвою «diagram azimut», рис. 3.9:

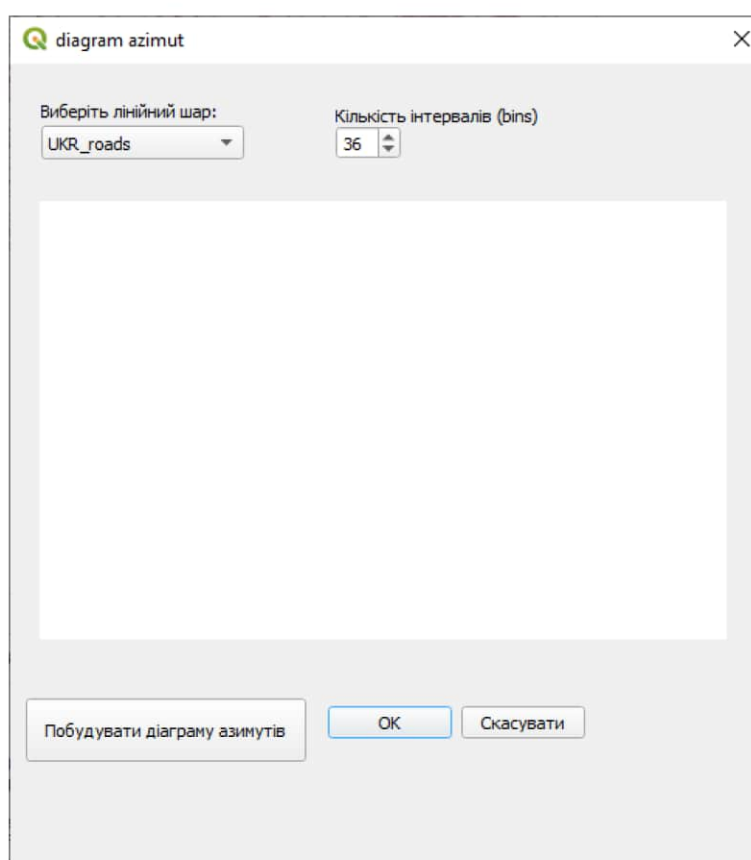


Рисунок 3.9 – Діалогове вікно плагіна

У цьому вікні основні елементи управління: «Виберіть лінійний шар» - це випадаючий список, який повинен автоматично заповнитися всіма лінійними векторними шарами, завантаженими у вашому поточному проекті QGIS. Якщо у списку відображається «Немає лінійних шарів», це означає, що плагін не розпізнав жодного лінійного шару, або ви не завантажили їх.

«Кількість інтервалів (bins)» – це поле для введення чисел, де ви можете вказати бажану кількість інтервалів (секторів) для побудови рози-діаграми азимутів. За замовчуванням може бути встановлено 36 інтервалів, що відповідає кроку в 10 градусів.

Після вибору лінійного шару зі списку та, за необхідності, налаштування кількості інтервалів, натисніть кнопку «Побудувати діаграму азимутів». Плагін обробить геометрію обраного шару, розрахує азимуту всіх лінійних сегментів та побудує полярну гістограму (розу-діаграму), яка відображатиме розподіл орієнтації лінійних об'єктів у просторі. Діаграма буде відображена безпосередньо у вікні плагіна, рис. 3.10, рис. Б1-Б3:

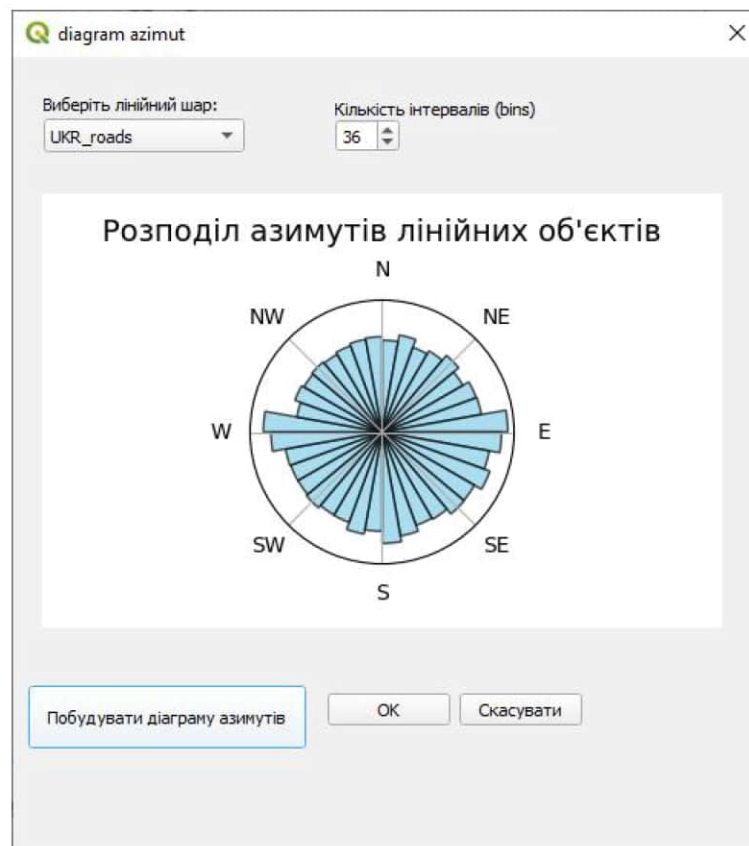


Рисунок 3.10 – Діаграма розподілу азимутів

3.4 Висновки

У третьому розділі було детально розглянуто процес реалізації та тестування додатку (плагіну) для побудови діаграм розподілу азимутів. Описано середовище розробки, що базується на програмному забезпеченні

QGIS, мові програмування Python та необхідних бібліотеках, таких як PyQt5 для графічного інтерфейсу, Matplotlib для візуалізації та вбудований модуль math. Підкреслено, що QGIS постачається з інтегрованим Python та PyQt5, що спрощує налаштування середовища. Окремо відзначено використання `pip` у консолі OSGeo4W Shell для встановлення додаткових бібліотек, таких як Matplotlib, та описано повний процес налаштування робочого середовища, включаючи створення структури папок плагіна та його розміщення в директорії модулів QGIS.

Далі, було змодельовано бізнес-процес функціонування додатку за допомогою контекстної діаграми. Деталізовано сутності, процеси та інтерфейсні дуги, що описують потік інформації та взаємодії.

Також у розділі описано безпосереднє функціонування інструменту. Реалізація плагіну деталізована через опис основної логіки, що міститься у файлі `azimuth_distribution_plotter.py`.

У цьому розділі також описано покроковий процес використання плагіна, починаючи від завантаження вхідного шейп-файлу, і закінчуючи запуском плагіна, вибором параметрів та отриманням кінцевої діаграми розподілу азимутів.

Таким чином, у розділі продемонстровано повний цикл розробки, від теоретичного обґрунтування до практичної реалізації та тестування інструменту, що підтверджує його готовність до використання.

ВИСНОВКИ

У даній кваліфікаційній роботі було підтверджено, що програмне забезпечення QGIS є потужною та гнучкою платформою для розробки спеціалізованих інструментів геопросторового аналізу. В результаті виконання кваліфікаційної роботи було розроблено функціонал для аналізу та візуалізації розподілу азимутів лінійних об'єктів у середовищі QGIS. Попри те, що QGIS є потужною та гнучкою платформою для геопросторового аналізу, інструмент саме візуалізації розподілу азимутів лінійних об'єктів у стандартній поставці QGIS відсутній.

Розроблений плагін, що базується на Python, PyQt5 та Matplotlib, ефективно вирішує поставлену задачу, надаючи користувачам QGIS можливість автоматично розраховувати азимути безпосередньо з геометрії лінійних об'єктів та візуалізувати їхній розподіл у вигляді інтерактивної кругової гістограми (рози вітрів). Цей інструмент є цінним доповненням до арсеналу ГІС-фахівців, дозволяючи їм здійснювати більш глибокий аналіз орієнтації лінійних об'єктів, що важливо у таких галузях, як геологія, гідрологія, планування інфраструктури та інших, тим самим сприяючи кращому розумінню просторових закономірностей та прийняттю обґрунтованих рішень.

ПЕРЕЛІК ПОСИЛАНЬ

1. Бондур В.Г., Зверев А.Т., Гапонова Е.В. Сучасні проблеми дистанційного зондування Землі із космосу 2012. Т. 9.№4. С. 322.
2. Бусігін Б.С., Гаркуша І.М., Середінін Е.С., Гаєвенко А.Ю.. Інструментарій геоінформаційних систем. - Київ: ІРГ «СБ», 2000. - 172 с.
3. NumPy v1.20 Manual. [Електронний ресурс] – Режим доступу до ресурсу: <https://numpy.org/doc/stable/>
4. Map Projection. [Електронний ресурс] – Режим доступу до ресурсу: https://www.usna.edu/Users/oceano/pguth/md_help/html/mapb0iem.htm
5. QGIS Development – Plugins. [Електронний ресурс] – Режим доступу до ресурсу: <https://qgis.org/pyqgis/3.28/>
6. Neteler, M., & Mitasova, H. (2024). Open Source GIS: A Grass GIS Approach (4th ed.). Springer.
7. McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython (2nd ed.).
8. PyQt Documentation (актуальні версії онлайн). [Електронний ресурс] – Режим доступу до ресурсу: <https://doc.qt.io/qtforpython/>
9. Matplotlib Documentation. [Електронний ресурс] – Режим доступу до ресурсу: <https://matplotlib.org/stable/contents.html>
10. Longley, P. A., Goodchild, M. F., Maguire, D. J., & Rhind, D. W. (2015). Geographic Information Science and Systems. Wiley.
11. Harris, R., & Brunson, C. (2020). Spatial Statistics: GeoComputation and Spatial Social Science. SAGE Publications.

Додаток А. Фрагмент лістингу програми

```
import os

from PyQt5.QtWidgets import QDialog, QComboBox, QPushButton,
QVBoxLayout, QLabel, QSpinBox, QMessageBox, QWidget # Додали QWidget
from PyQt5.QtCore import QSettings, QTranslator, QApplication, QSize
from PyQt5.QtGui import QIcon

from qgis.gui import QgisInterface

import matplotlib
matplotlib.use('Qt5Agg') # Важливо для інтеграції Matplotlib з PyQt
import matplotlib.pyplot as plt
from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg as
FigureCanvas
import numpy as np
import math

from .azimuth_distribution_plotter_dialog_base import
Ui_AzimuthDistributionPlotterDialogBase

class AzimuthDistributionPlotterDialog(QDialog,
Ui_AzimuthDistributionPlotterDialogBase):
    def __init__(self, parent=None):
        super().__init__(parent)
        self.setupUi(self)

        self.cmb_layer = self.findChild(QComboBox, 'cmb_layer')
        self.spn_bins = self.findChild(QSpinBox, 'spn_bins')
        self.btn_plot = self.findChild(QPushButton, 'btn_plot')
```

```
self.plot_widget = self.findChild(QWidget, 'plot_widget') # Це має бути ваш
контейнер для графіка
```

```
self.figure = plt.Figure(figsize=(5, 4), dpi=100) # Розмір може бути
скоригований
```

```
self.canvas = FigureCanvas(self.figure)
```

```
# Перевіряємо, чи існує вже макет для plot_widget, якщо ні - створюємо
```

```
self.plot_widget_layout = self.plot_widget.layout()
```

```
if not self.plot_widget_layout:
```

```
    self.plot_widget_layout = QVBoxLayout(self.plot_widget)
```

```
self.plot_widget_layout.addWidget(self.canvas)
```

```
self.populate_layers()
```

```
self.btn_plot.clicked.connect(self.plot_azimuth_distribution)
```

```
if self.spn_bins.value() == 0: # Якщо не встановлено в Designer
```

```
self.spn_bins.setValue(36) # За замовчуванням 36 інтервалів (по 10
градусів)
```

```
def populate_layers(self):
```

```
    self.cmb_layer.clear()
```

```
    layers = QgsProject.instance().mapLayers().values()
```

```
    for layer in layers:
```

```
        if layer.type() == QgsVectorLayer.VectorLayer and \
        (layer.geometryType() == QgsWkbTypes.LineGeometry or \
        layer.geometryType() == QgsWkbTypes.MultiLineGeometry):
            self.cmb_layer.addItem(layer.name(), layer.id())
```

```
if self.cmb_layer.count() == 0:
```

```

self.cmb_layer.addItem("Немає лінійних шарів", None)

def plot_azimuth_distribution(self):
    selected_layer_id = self.cmb_layer.currentData()
    if selected_layer_id is None: # Змінили перевірку, якщо немає шарів
        QMessageBox.warning(self, "Помилка", "Будь ласка, виберіть лінійний шар або додайте його до проекту.")
        return

    layer = QgsProject.instance().mapLayer(selected_layer_id)
    if not layer:
        QMessageBox.warning(self, "Помилка", "Вибраний шар не знайдено.")
        return

    if not (layer.geometryType() == QgsWkbTypes.LineGeometry or \
            layer.geometryType() == QgsWkbTypes.MultiLineGeometry):
        QMessageBox.warning(self, "Помилка", "Вибраний шар не є лінійним.")
        return

    azimuths = []
    distance_area = QgsDistanceArea()
    distance_area.setEllipsoid('WGS84') # Використовуємо стандартний еліпсоїд WGS84

    for feature in layer.getFeatures():
        geom = feature.geometry()

        if geom.type() == QgsWkbTypes.LineGeometry:
            parts = [geom.asPolyline()]
        elif geom.type() == QgsWkbTypes.MultiLineGeometry:

```

```

    parts = geom.asMultiPolyline()
else:
    continue

for part in parts:
    for i in range(len(part) - 1):
        p1 = QgsPointXY(part[i])
        p2 = QgsPointXY(part[i+1])

# Перевіряємо, чи точки відрізняються, щоб уникнути помилок з нульовою
# довжиною
        if p1 == p2:
            continue

        # QgsDistanceArea.bearing повертає азимут у радіанах
        # від Півночі за годинниковою стрілкою
        azimuth_rad = distance_area.bearing(p1, p2)

# Переводимо радіани в градуси і нормалізуємо до діапазону [0, 360)
        azimuth_deg = (math.degrees(azimuth_rad) + 360) % 360
        azimuths.append(azimuth_deg)

if not azimuths:
    QMessageBox.information(self, "Інформація", "Вибраний шар не
містить лінійних об'єктів або сегментів для розрахунку азимутів.")
    return

self.figure.clear()
ax = self.figure.add_subplot(111, projection='polar')

```

```

bins = self.spn_bins.value()
if bins <= 0:
    bins = 1

azimuths_rad = np.radians(azimuths)

counts, bin_edges = np.histogram(azimuths_rad, bins=bins, range=(0, 2 *
np.pi))

angles = (bin_edges[:-1] + bin_edges[1:]) / 2
width = (2 * np.pi) / bins # Ширина кожного сектора
ax.bar(angles, counts, width=width, bottom=0.0, align='center',
color='skyblue', edgecolor='black', alpha=0.7)

ax.set_theta_zero_location('N') #
ax.set_theta_direction(-1) # Напрямок за годинниковою стрілкою
ax.set_title("Розподіл азимутів лінійних об'єктів", va='bottom',
fontsize=14)

ax.set_xticks(np.radians([0, 45, 90, 135, 180, 225, 270, 315]))
ax.set_xticklabels(['N', 'NE', 'E', 'SE', 'S', 'SW', 'W', 'NW'])

ax.set_yticks([])
self.figure.tight_layout()
self.canvas.draw() # Перемальовуємо полотно

class AzimuthDistributionPlotter:
    def __init__(self, iface):
        """

```

Constructor.

```

:param iface: An interface instance that will be passed to this class
    instance (see https://qgis.org/api/classQgisInterface.html)
    """
self.iface = iface
self.plugin_dir = os.path.dirname(__file__)

# initialize locale
locale = QSettings().value('locale/userLocale')[0:2]
locale_path = os.path.join(
    self.plugin_dir,
    'i18n',
    'AzimuthDistributionPlotter_{}.qm'.format(locale))

if os.path.exists(locale_path):
    self.translator = QTranslator()
    self.translator.load(locale_path)
    QApplication.installTranslator(self.translator)

# Declare instance attributes
self.actions = []
self.menu = self.tr(u'Побудова Діаграм Азимутів')

iface_actual = self.iface.mainWindow()
if iface_actual is None:
    from qgis.utils import iface
    self.iface = iface

```

```
self.toolbar = self.iface.addToolBar(u'AzimuthDistributionPlotter')
self.toolbar.setObjectName(u'AzimuthDistributionPlotter')
```

```
def tr(self, message):
```

```
    """
```

```
    Get the translation for a string using Qt translation API.
```

```
    Controls how the translator is chosen.
```

```
    (e.g. self.tr('Hello World'))
```

```
    :param message: String for translation.
```

```
    :type message: str
```

```
    :returns: Translated string.
```

```
    :rtype: str
```

```
    """
```

```
    return QApplication.translate('AzimuthDistributionPlotter', message)
```

```
def add_action(
```

```
    self,
```

```
    icon_path,
```

```
    text,
```

```
    callback,
```

```
    enabled_by_default=True,
```

```
    add_to_menu=True,
```

```
    add_to_toolbar=True,
```

```
    status_tip=None,
```

```
    whats_this=None,
```

parent=None):

"""

Add a toolbar icon to the toolbar.

:param icon_path: Path to the icon for this action. Can be a resource path (e.g. `!:/plugins/myplugin/icon.png`) or a full path to an image file.

:type icon_path: str

:param text: Text that should be shown in the menu and as button tooltip.

:type text: str

:param callback: Function to be called when the action is triggered.

:type callback: function

:param enabled_by_default: Specify if the action should be enabled by default.

:type enabled_by_default: bool

:param add_to_menu: Flag whether the action should also be added to the menu.

:type add_to_menu: bool

:param add_to_toolbar: Flag whether the action should also be added to the toolbar.

:type add_to_toolbar: bool

:param status_tip: Optional text to show in the status bar when the mouse hovers over the action.

```
:type status_tip: str

:param whats_this: Optional text for "What's This?" help.
:type whats_this: str

:param parent: Parent widget for the new action.
:type parent: QWidget

:returns: The action that was created.
:rtype: QAction
"""

icon = QIcon(icon_path)
action = QAction(icon, text, parent)
action.triggered.connect(callback)
action.setEnabled(enabled_by_default)

if add_to_menu:
    self.iface.addPluginToMenu(
        self.menu,
        action)

if add_to_toolbar:
    self.toolbar.addAction(action)

self.actions.append(action)

return action
```

```

def initGui(self):
    """
    Create the plugin's toolbar icon and menu entry.
    """
    icon_path = '/plugins/AzimuthDistributionPlotter/icon.png'
    self.add_action(
        icon_path,
        text=self.tr(u'Побудувати діаграму азимутів'),
        callback=self.run,
        parent=self.iface.mainWindow())

def unload(self):
    """
    Removes the plugin menu item and icon from QGIS GUI.
    """
    for action in self.actions:
        self.iface.removePluginMenu(
            self.tr(u'Побудова Діаграм Азимутів'),
            action)
        self.iface.removeToolBarIcon(action)
    # remove the toolbar
    del self.toolbar

def run(self):
    """
    Runs the plugin, showing its dialog.
    """
    # Create the dialog with the appropriate parent

```

```
self.dlg = AzimuthDistributionPlotterDialog() # self.iface вже не  
передається в AzimuthDistributionPlotterDialog
```

```
# Show the dialog
```

```
self.dlg.show()
```

```
# or if you want it modal (blocking QGIS interaction)
```

```
# self.dlg.exec_()
```

Додаток Б. Тестування створеного додатку (плагіну)

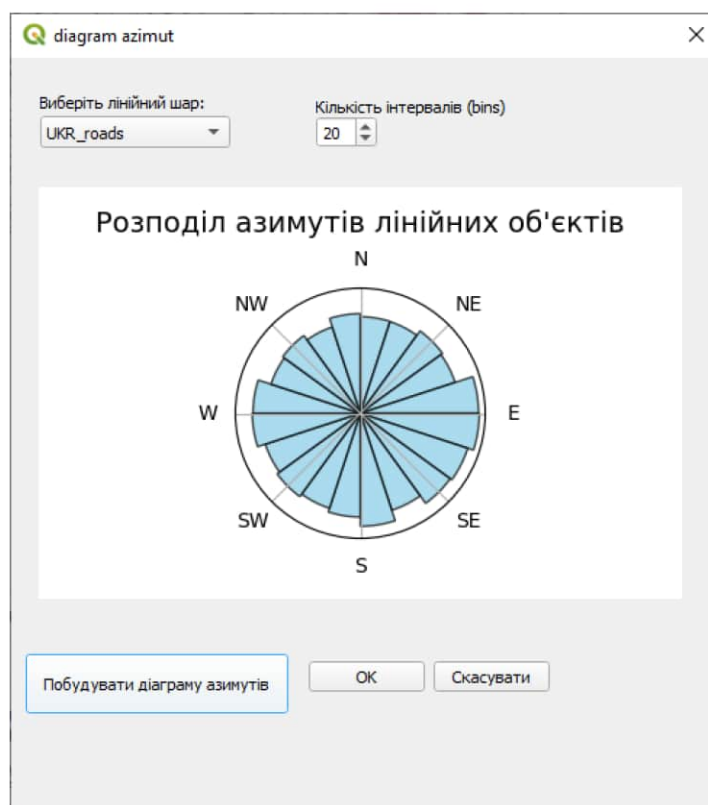


Рисунок Б.1 – Діаграма розподілу азимутів, кількість інтервалів 20

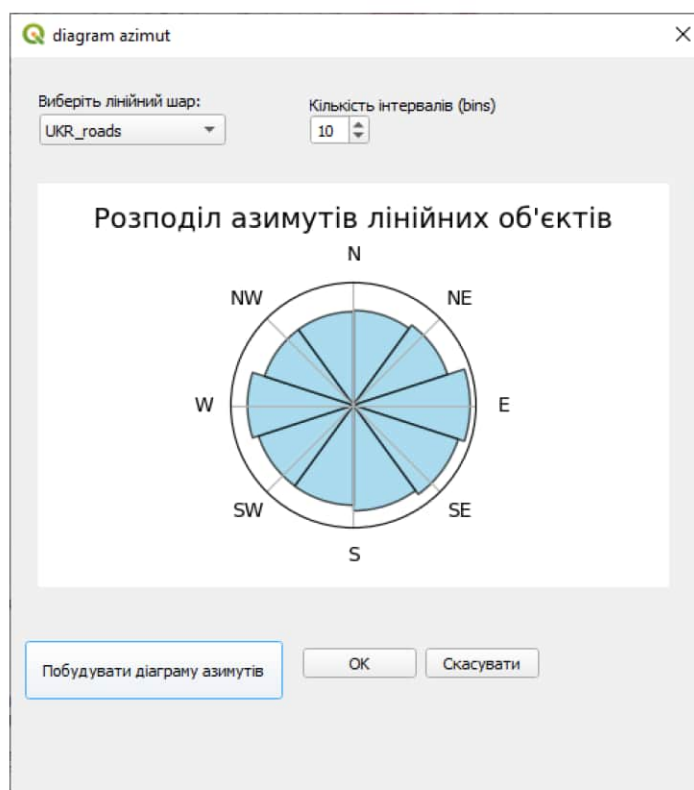


Рисунок Б.2 – Діаграма розподілу азимутів, кількість інтервалів 10

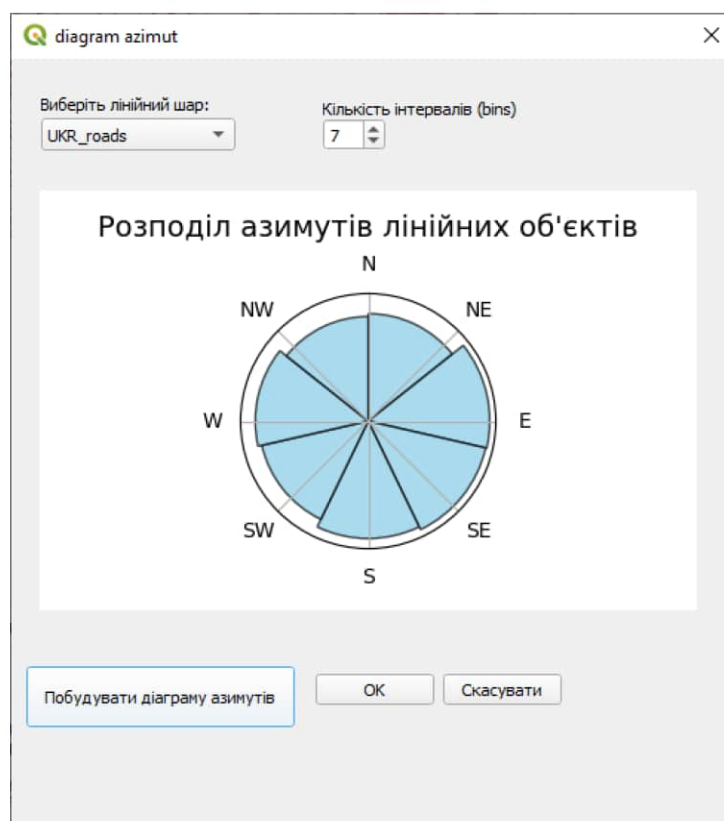


Рисунок Б.3 – Діаграма розподілу азимутів, кількість інтервалів 7