

Практична робота 3

Тема: Введення атрибутивних даних.

Мета: ознайомлення з інструментами програми QGIS по роботі з атрибутивними даними..

Короткі теоретичні відомості

В кожному шарі використовуються дані двох типів. Просторові дані описують розташування і форму географічних об'єктів. Атрибутивні дані інформують про інші характеристики об'єктів.

Створення і редагування об'єктів зазвичай тягне за собою створення і редагування атрибутів цих об'єктів, на доповнення до зміни їх форми.

QGIS надає можливість зручного перегляду і оновлення атрибутів в базі даних. Можна редагувати атрибути об'єктів двома способами: використовуючи діалогове вікно атрибутів або використовуючи таблицю атрибутів. В діалоговому вікні атрибутів можна переглянути атрибути вибраних об'єктів карти; додавати, видаляти або модифікувати атрибути окремого об'єкту або кількох об'єктів одночасно; копіювати і вставляти окремі атрибути або набори атрибутів об'єкта.

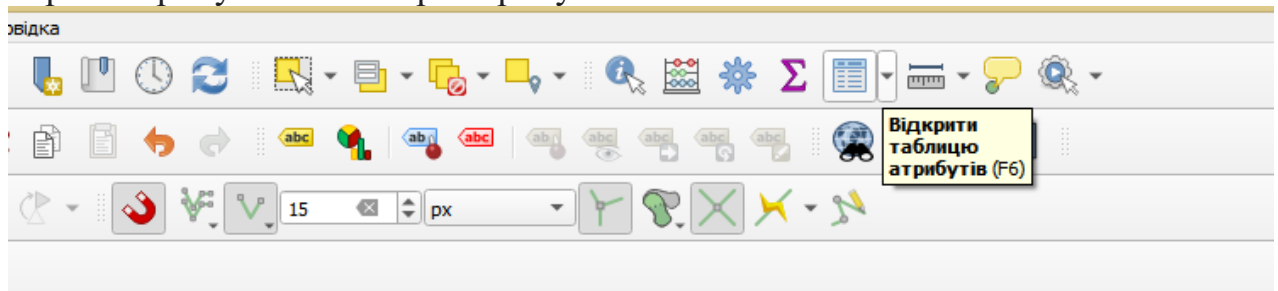


Рис. 3.1. Спосіб відкриття таблиці атрибутів (піктограма на панелі інструментів)

Ті ж функції можна виконати, використовуючи таблицю атрибутів шару. Однак в таблицях можна виконувати також обчислення та інші функції, такі як додавання і сортування записів.

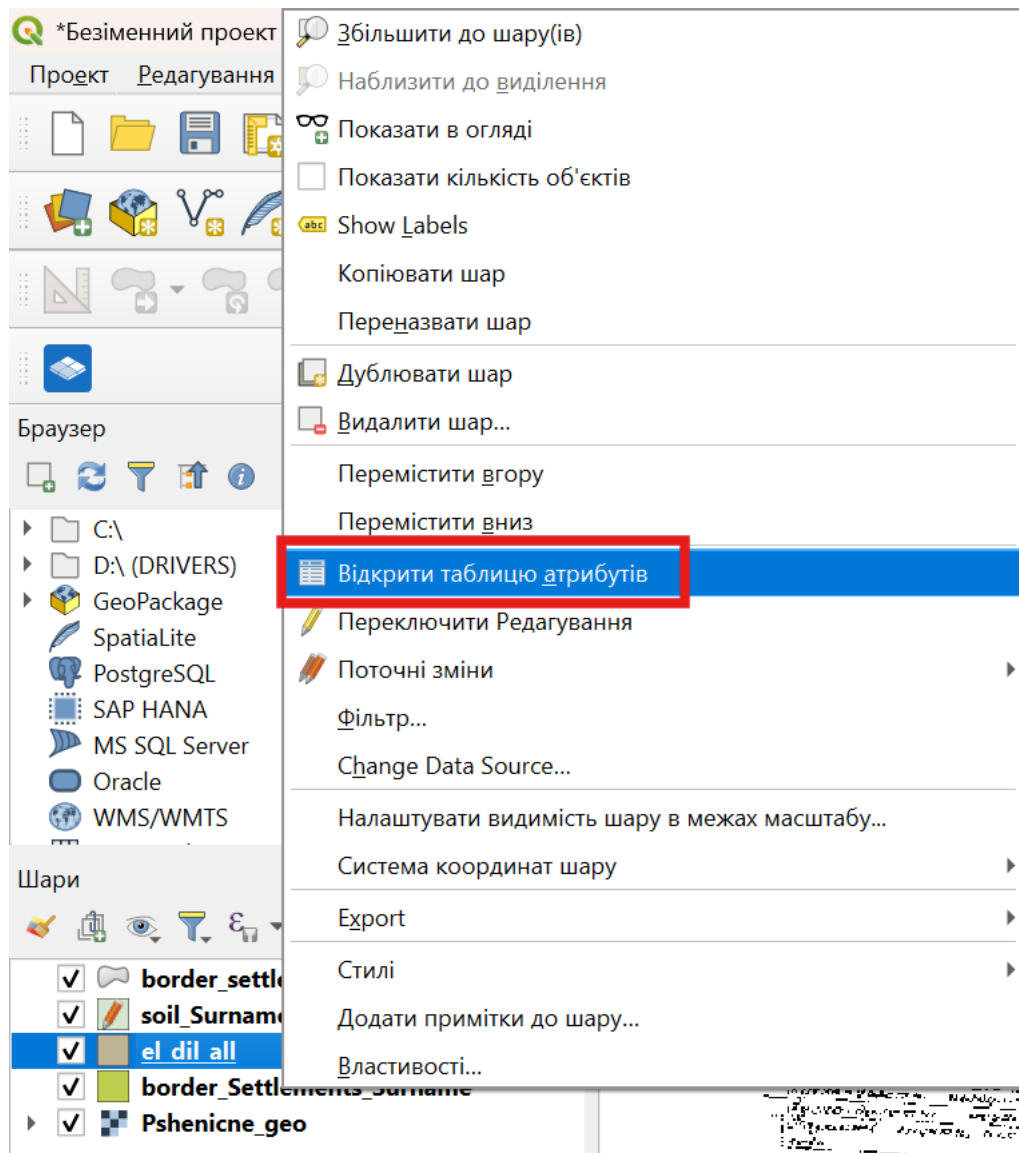


Рис. 3.2. Спосіб відкриття таблиці атрибутів (контекстне меню шару)

Методичні вказівки

Перегляд атрибутів. Атрибут – це додаткова характеристика, властивість, яка описує деяку характеристику об'єкта.

Для перегляду таблиці атрибутів шару необхідно в таблиці змісту клацнути правою кнопкою мишки на шарі, для якого потрібно переглянути таблицю атрибутів. У контекстному меню вибрати опцію відкриття атрибутивної таблиці – **Open Attribute Table (Відкрити таблицю атрибутів)**. У відповідь на команду відкриється атрибутивна таблиця шару (рис. 3.3.).

OBJECTID	Id	Shape_Leng	Shape_Area	N	P	K	Hum	pH
1	63	1269,84252819...	96095,6978336...	204,2000000000...	205,0000000000...	110,0000000000...	5,090000000000	7,600000000000
2	65	1870,47198901...	119297,784615...	204,2000000000...	205,0000000000...	80,000000000000	5,090000000000	7,500000000000
3	64	1237,33635686...	71513,0852408...	204,2000000000...	205,0000000000...	74,000000000000	5,090000000000	7,500000000000
4	66	1449,83022134...	68163,5306411...	204,2000000000...	205,0000000000...	84,000000000000	5,090000000000	7,400000000000
5	62	1248,23305449...	94964,7546020...	189,0000000000...	205,0000000000...	120,0000000000...	5,270000000000	7,300000000000
6	64	686,273299705...	27793,2786860...	204,2000000000...	205,0000000000...	74,000000000000	5,090000000000	7,500000000000
7	35	1186,96849592...	81168,5729834...	166,2000000000...	205,0000000000...	100,0000000000...	3,220000000000	7,400000000000
8	34	1470,86998324...	132224,102744...	170,2000000000...	205,0000000000...	110,0000000000...	4,610000000000	7,400000000000
9	33	1180,29661053...	80808,8776472...	170,2000000000...	205,0000000000...	130,0000000000...	4,610000000000	7,400000000000
10	36	1444,21034357...	102245,275385...	162,6000000000...	205,0000000000...	100,0000000000...	3,670000000000	7,200000000000
11	38	786,890284031...	34400,9683621...	182,4000000000...	205,0000000000...	90,000000000000	4,500000000000	7,600000000000
12	38	1182,95516222...	76437,1138275...	182,4000000000...	205,0000000000...	90,000000000000	4,500000000000	7,600000000000
13	37	1162,47269293...	73992,2826932...	162,6000000000...	205,0000000000...	80,000000000000	3,670000000000	7,400000000000
14	40	1572,26119638...	110625,589467...	182,4000000000...	205,0000000000...	140,0000000000...	4,500000000000	7,500000000000

Рис. 3.3. Вигляд атрибутивної таблиці шару «Елементарні ділянки»

Колонки в атрибутивній таблиці, можна називати полями, а рядки – записами. Перше відображене поле містить ідентифікаційну характеристику об’єкта. За замовчуванням цим ідентифікатором є наскрізне нумерування всіх оцифрованих полігонів. Друге поле вказує на присвоєний номер полігону. Наступні поля можна видалити, додати, перейменувати, приховати тощо.

Внизу таблиці атрибутів містяться кнопки переходу на режим перегляду форми чи на режим порегляду таблиці а також меню фільтрування та відображення даних таблиці.

Для зміни ширини поля таблиці потрібно помістити вказівник мишки на границю комірки із назвою колонки, розмір якої потрібно змінити. При зміні зображення вказівника натиснути на ліву кнопку мишки та потягнути її в потрібному напрямку. Крайня сіра лінія вказує, де буде знаходитись нова границя колонки. Якщо розташування границі відповідає вимозі користувача, кнопку мишки потрібно відпустити. Ширину колонки змінено.

Порядок записів в таблиці атрибутів за потреби можна змінити – відсортувати. Сортування записів в таблиці дозволяє ефективно досліджувати дані. Для сортування даних потрібно навести курсор миші на чорний трикутник в правому нижньому кутку комірки з назвою колонки. Натиснути на ліву кнопку миші можна здійснити сортування за зростанням або спаданням.

Щоб переглянути, які дані в таблиці відповідають графічному об’єкту, потрібно вибрати на панелі інструментів **Визначити об’єкт за областю або одним натисканням кнопки** та натиснути на зображення об’єкту. Запис атрибутів вибраного на карті об’єкту у відкритій таблиці виділяться блакитним кольором (рис. 3.4.). Цей запис в таблиці можна знайти або користуючись прокруткою.

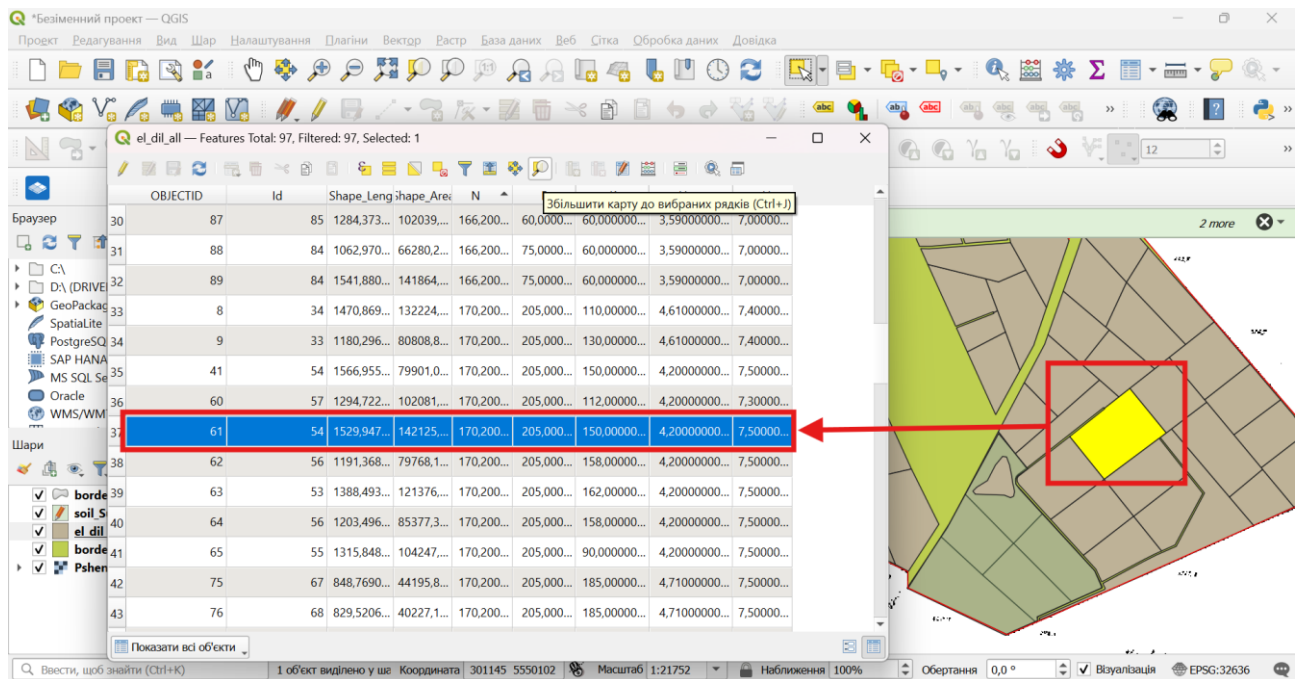


Рис.3.4. Виділення обраного об'єкту на карті синім кольором в таблиці

Можна виконати і зворотній вибір – за атрибутами визначити графічний об'єкт. Для цього потрібно в таблиці атрибутів виділити запис, натиснувши ліворуч від першої колонки вибраного запису – весь рядок виділяється блакитним кольором, а також жовтим кольором виділяється об'єкт на карті.

Щоб закрити таблицю атрибутів треба натиснути на кнопку закрити (x) в правому верхньому куті таблиці.

Додавання і редагування атрибутів. Найчастіше атрибутивна таблиця не містить потрібної інформації про об'єкти. Потрібні дані часто знаходяться в інших таблицях або на паперових носіях.

При необхідності легко можна додати або видалити поле з таблиці QGIS.

Додавання поля до атрибутивної таблиці можна виконати в такій послідовності:

1. Відкрити таблицю атрибутів.
2. Вверху на панелі інструментів таблиці атрибутів натиснути

Додати поле.

3. Ввести ім'я поля в стрічку **Name (Ім'я)**.
4. Вибрати тип даних в стрічці **Type (Тип)**.
5. Задати інші властивості поля, якщо потрібно.
6. Натиснути **ОК (рис. 3.5.)**.

УВАГА! Ім'я поля може містити максимум 10 символів.

Видалення поля атрибутивної таблиці виконується наступним чином:

1. У вікні атрибутивної таблиці клацнути правою кнопкою миші на заголовку поля, що необхідно видалити.
2. Натиснути на **Delete field (Видалити поле)** вгорі на панелі інструментів таблиці атрибутів.
3. Натисніть **Yes (Так)**, щоб підтвердити.

УВАГА! Видалення поля не можна відмінити.

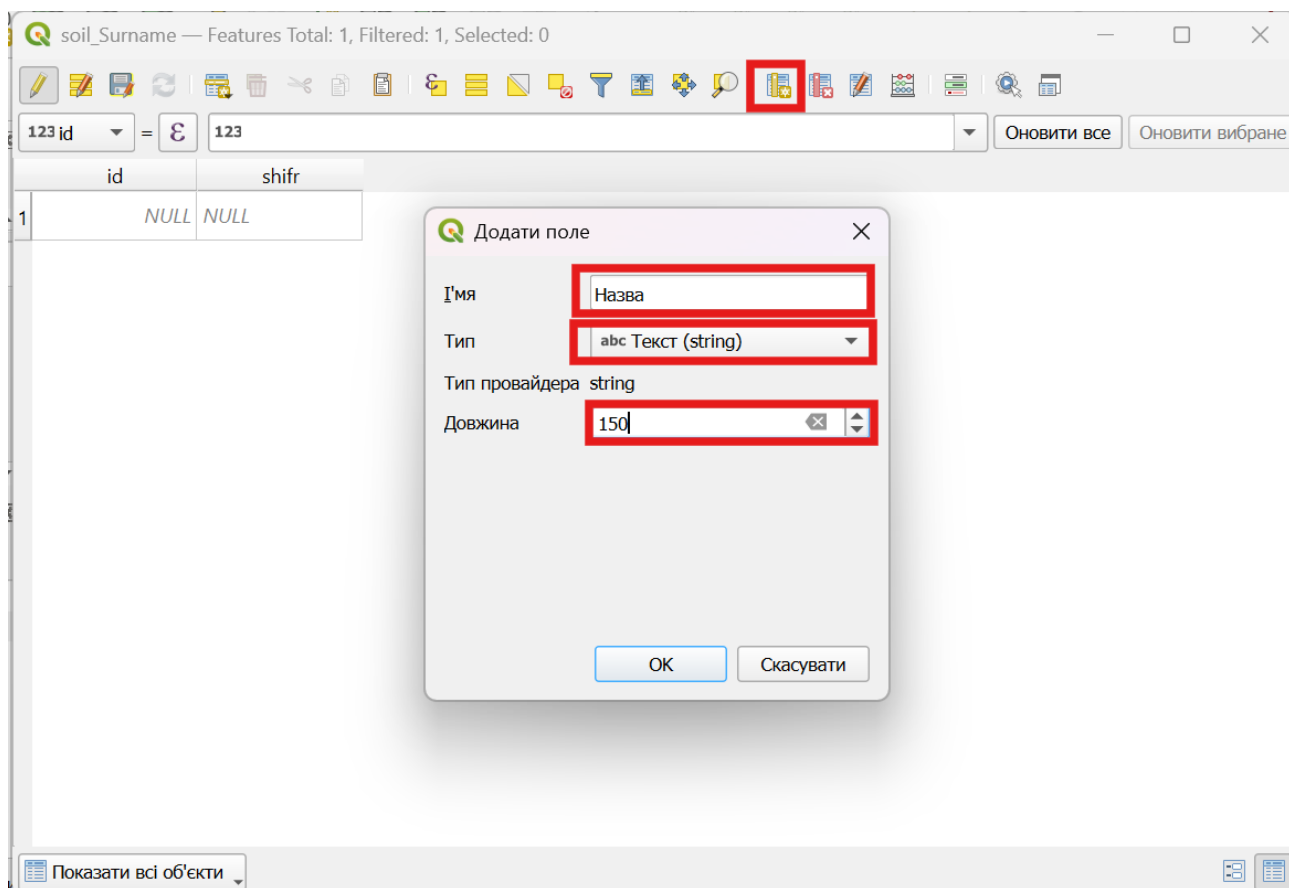


Рис.3.5. Додавання полів в таблицю атрибутів

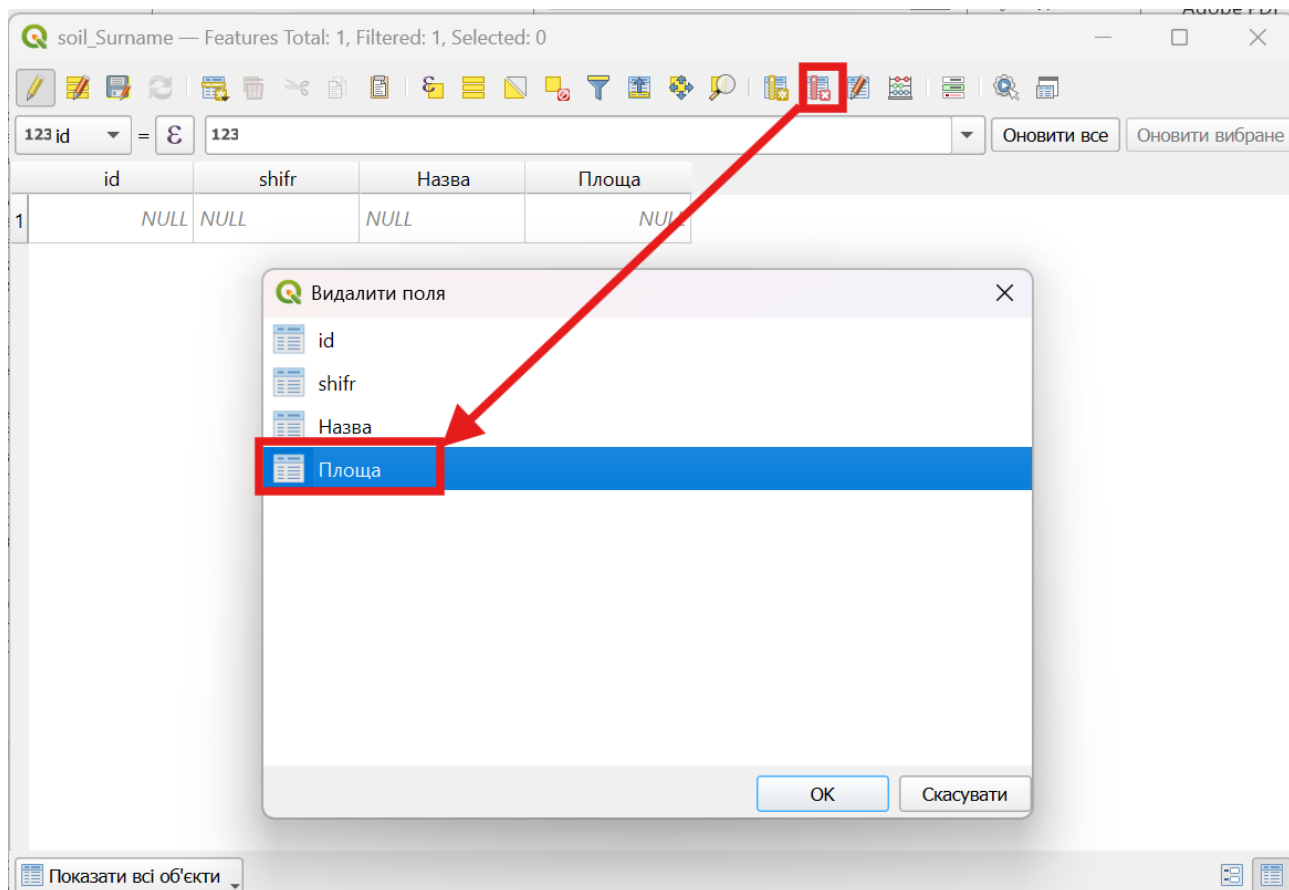


Рис.3.6. Видалення полів з таблиці атрибутів

Для введення даних до таблиці потрібно включити редагування даних, натиснувши на  **Переключити режим редагування** (вверху на панелі інструментів таблиці атрибутів):

1. Відкрити атрибутивну таблицю, дані якої потрібно відредагувати.
2. Клацнути на комірку, в яку потрібно ввести інформацію.
3. Натиснути **Переключити режим редагування**.
4. Ввести значення та натиснути **Enter** на клавіатурі.
5. Після внесених необхідних значень натиснути знову

Переключити режим редагування.

6. У вікні що з'явиться підтвердити збереження змін.
7. В атрибутивну таблицю додано інформацію.

З метою редагування даних потрібно виконати такі ж дії з використанням інструменту  **Переключити режим редагування**.

Коли всі потрібні зміни внесено, їх можна зберегти та завершити редагування аналогічно до завершення при завершенні шифрування.

Зміна послідовності полів. Послідовність полів в таблиці атрибутів можна змінювати використовуючи інструмент  **Organize Columns (Упорядкувати колонки таблиці)**. Цей інструмент дає змогу відключити непотрібні поля та виставити потрібну послідовність полів.

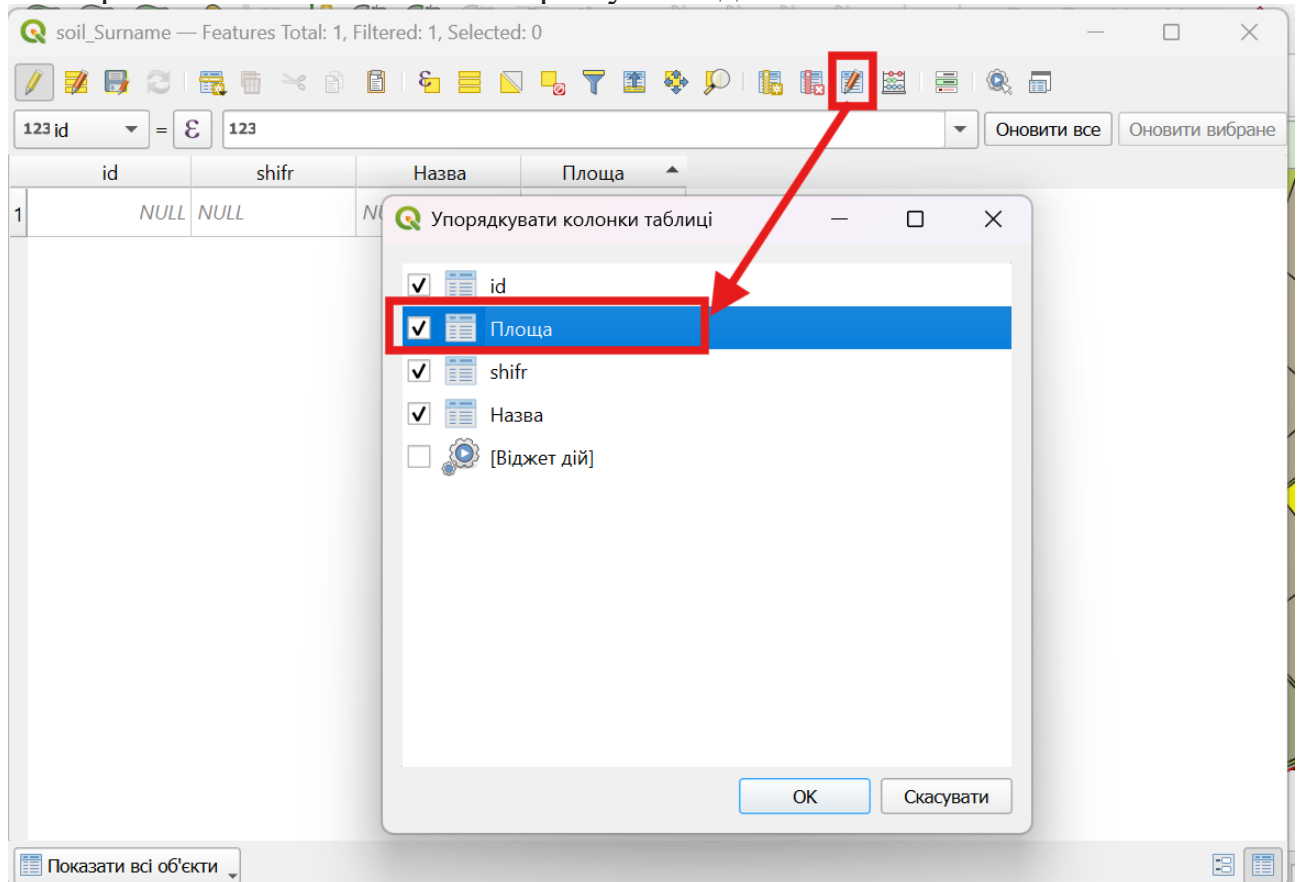


Рис. 3.7. Діалогове вікно Упорядкувати колонки таблиці в QGIS

Для заповнення поля розрахунковими значеннями, наприклад, площі об'єкта в таблиці атрибутів потрібно використати інструмент Калькулятор відкритого поля та використати функцію \$area, рис. 3.8.

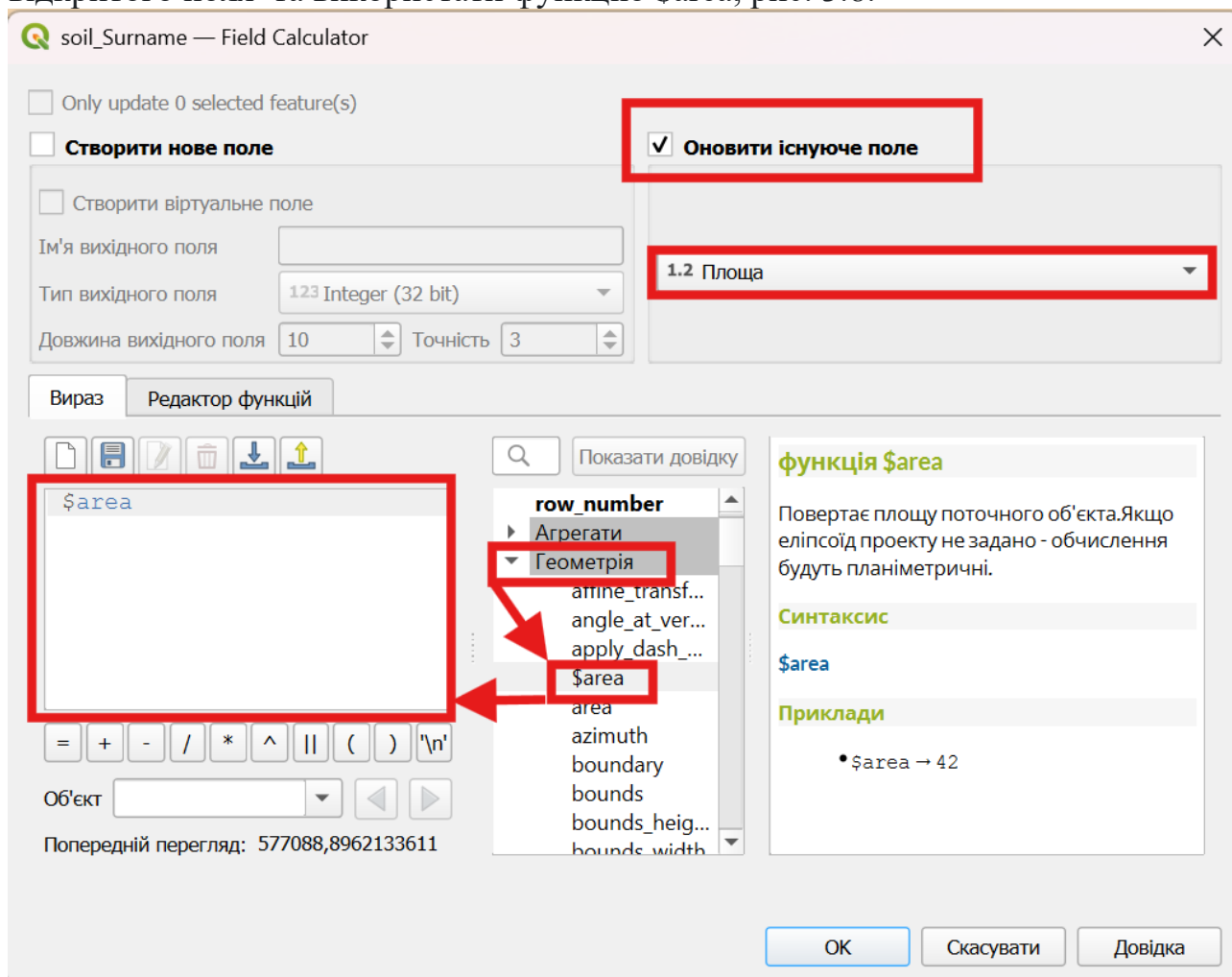


Рис. 3.8. Діалогове вікно Калькулятор відкритого поля в QGIS

Вікно **Властивості шару (Layer Properties)** для векторного шару в QGIS – це надає можливості керування виглядом, даними, поведінкою та метаданими. Для переходу в цей розділ достатньо зробити подвійний клік на назві шару в панелі шарів або натиснути правою кнопкою миші та обрати **Властивості (Properties)** (рис. 3.9).

У вікні **Властивості (Properties)** можна переглянути Загальну інформацію та джерела: шлях до файлу або бази даних, кількість об'єктів, тип геометрії, систему координат.

Також у цьому вікні можна змінити візуалізацію, налаштувати відображення текстів або значень.

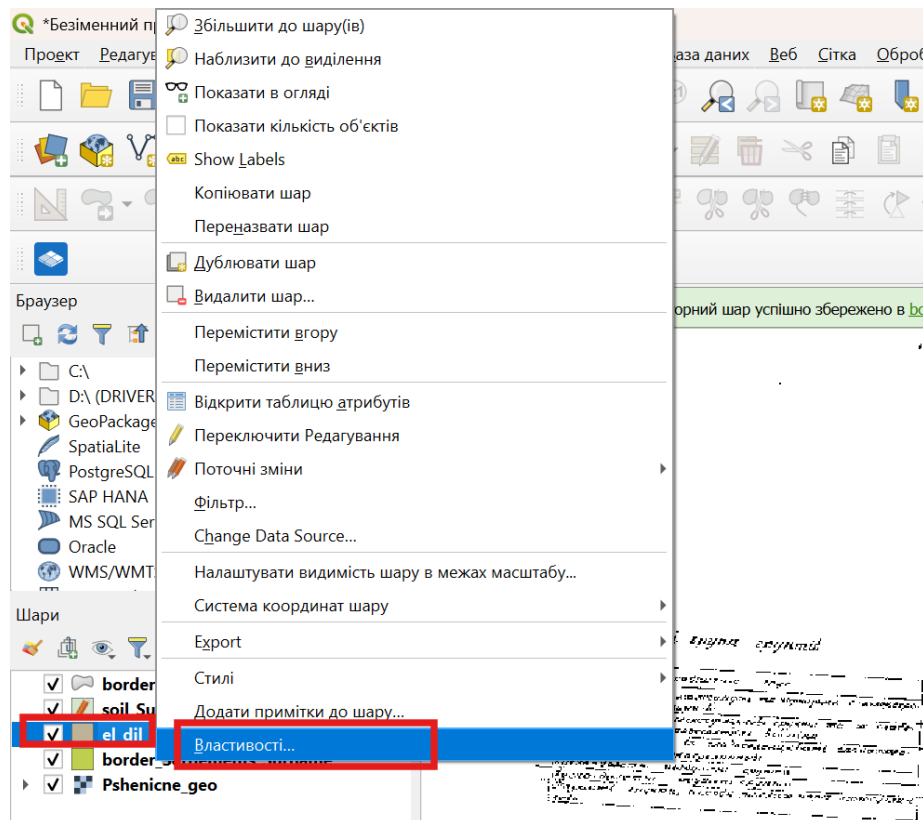


Рис. 3.9. Відкриття властивостей шару в QGIS

Для виведення підписів номерів елементарних ділянок необхідно обрати **Підписи** та визначити поле, що відображатиметься, встановити шрифт та розмір (рис.3.10)

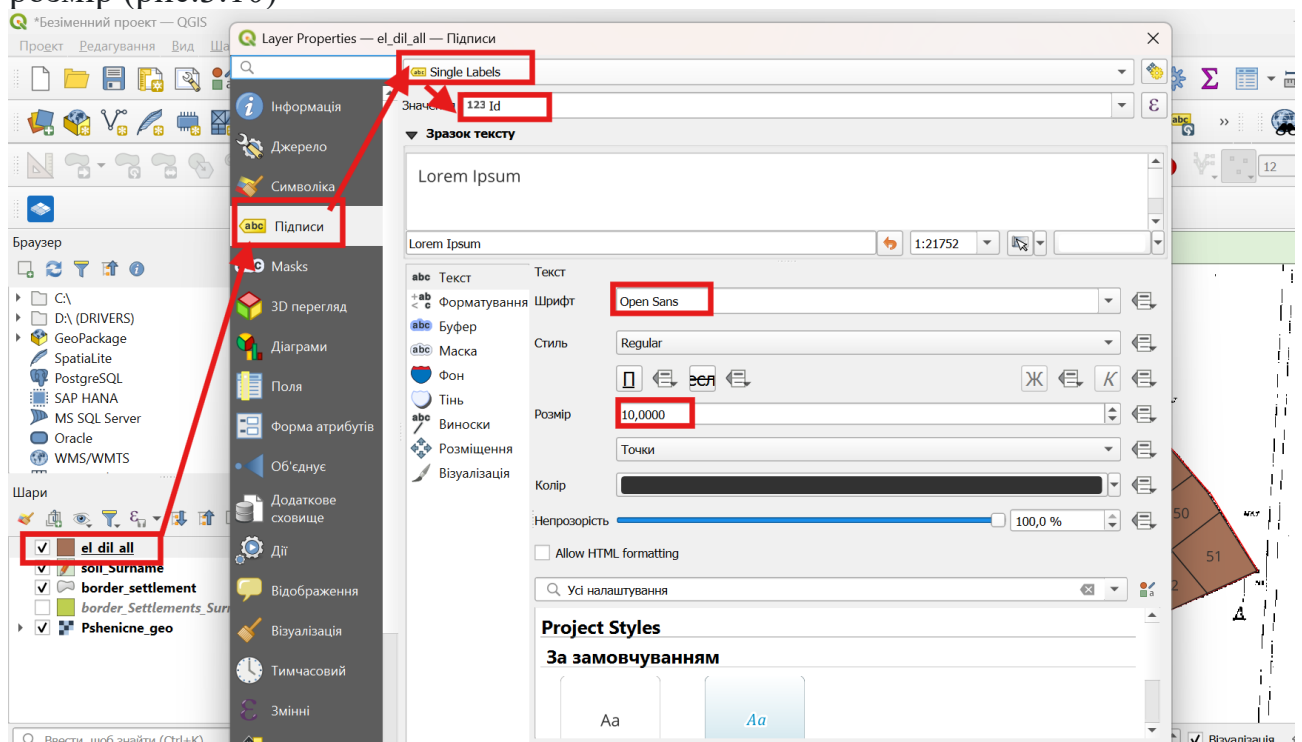


Рис. 3.10. Встановлення підписів для шару в QGIS

Перегляд списку атрибутивних полів, їхніх типів даних (число, текст, дата), довжини та коментарів здійснюється через розділ **Поля** і меню

Властивості (рис. 3.11.). Тут також можна додавати нові або вилучати існуючі поля.

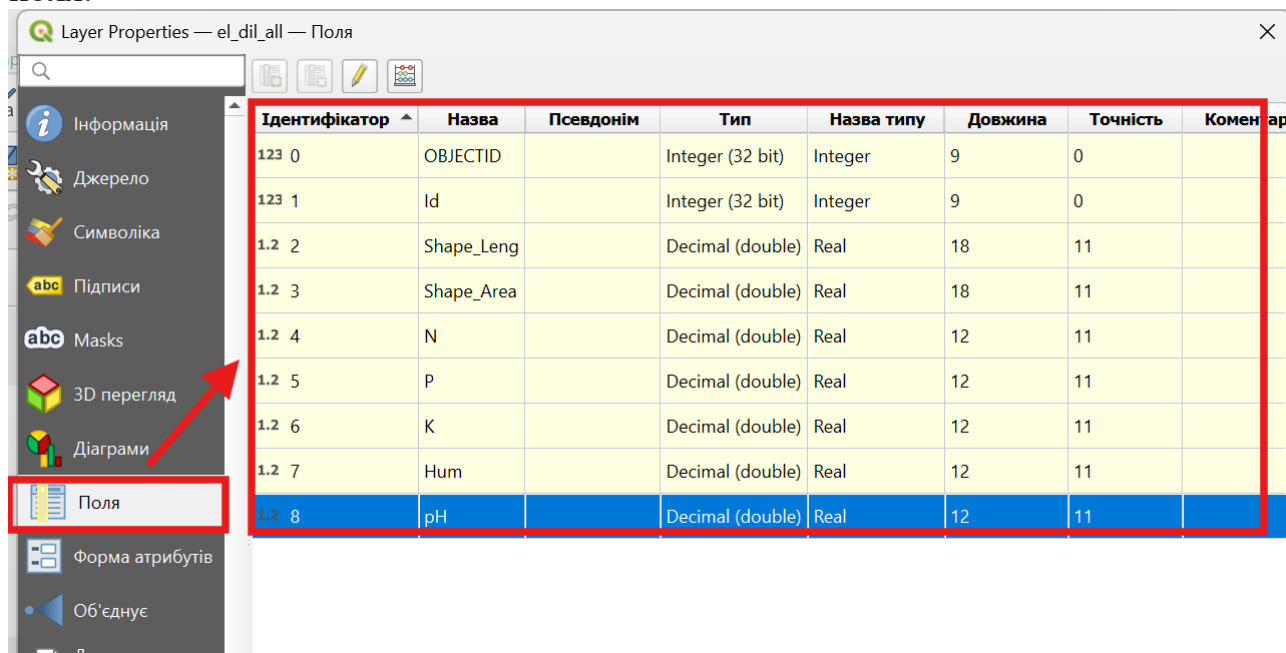


Рис. 3.11. Властивості Полів для шару в QGIS

Щоб виконати автозаповнення таблиці атрибутів одного шару атрибутами іншого, потрібно:

В головному меню вибрати vector – Data Management Tools – Join Attributes by Location (рис 3.12).

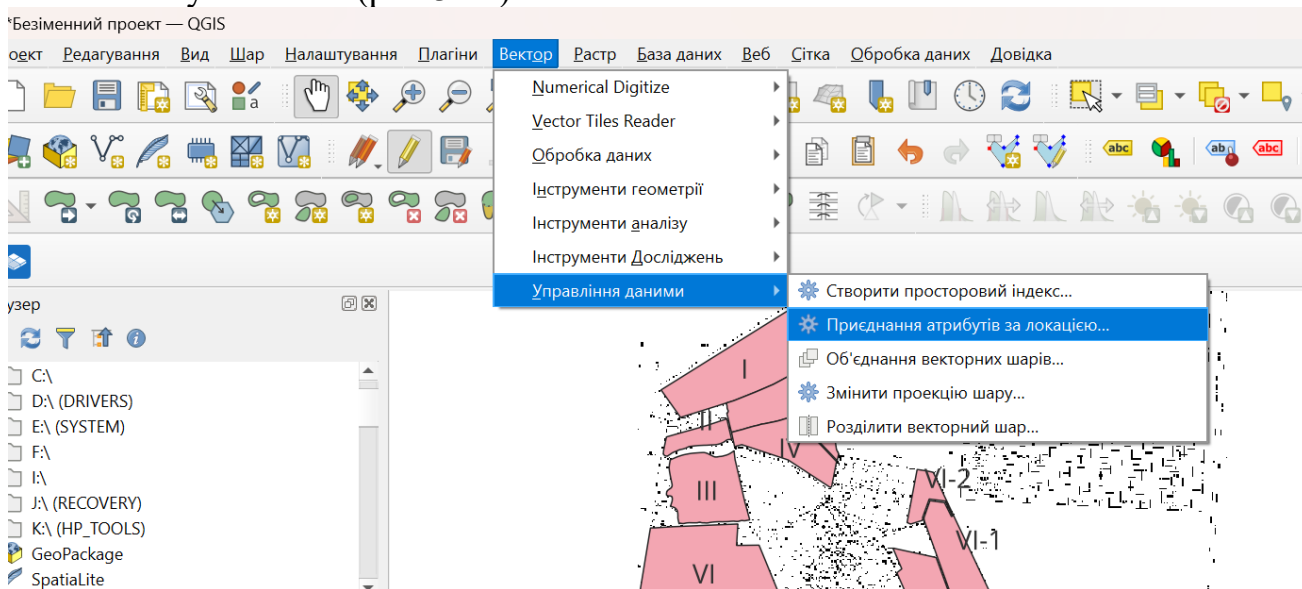


Рис. 3.12. Вибір програмного модулю для автозаповнення атрибутів шейп-файлу по попаданню в межу іншого шейпфайлу

У вікні команди обрати шейп-файл, таблицю атрибутів якого потрібно заповнити, спосіб перетинання входить повністю (are within) та шейп-файл, з якого будуть додаватися атрибути, рис. 3.13.

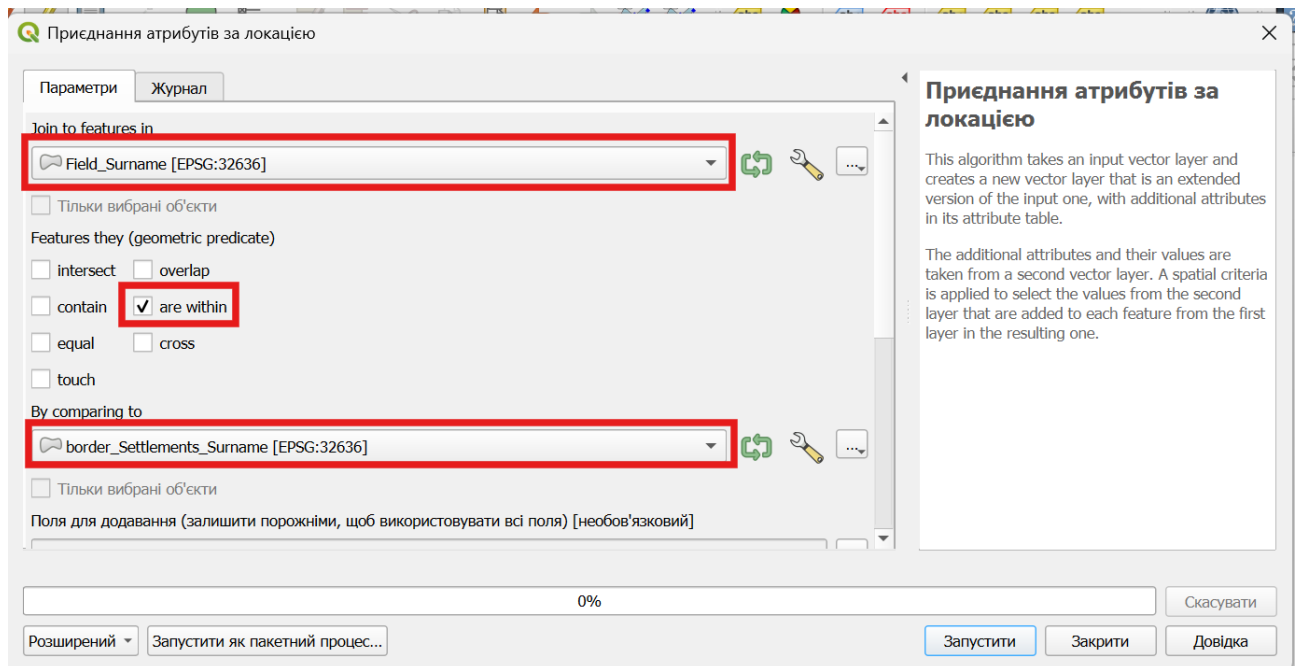


Рис. 3.13. Налаштування модуля автозаповнення

У випадку навіть мінімальної накладки рядки не заповняться, дивіться рядок №5 (рис. 3.14.).

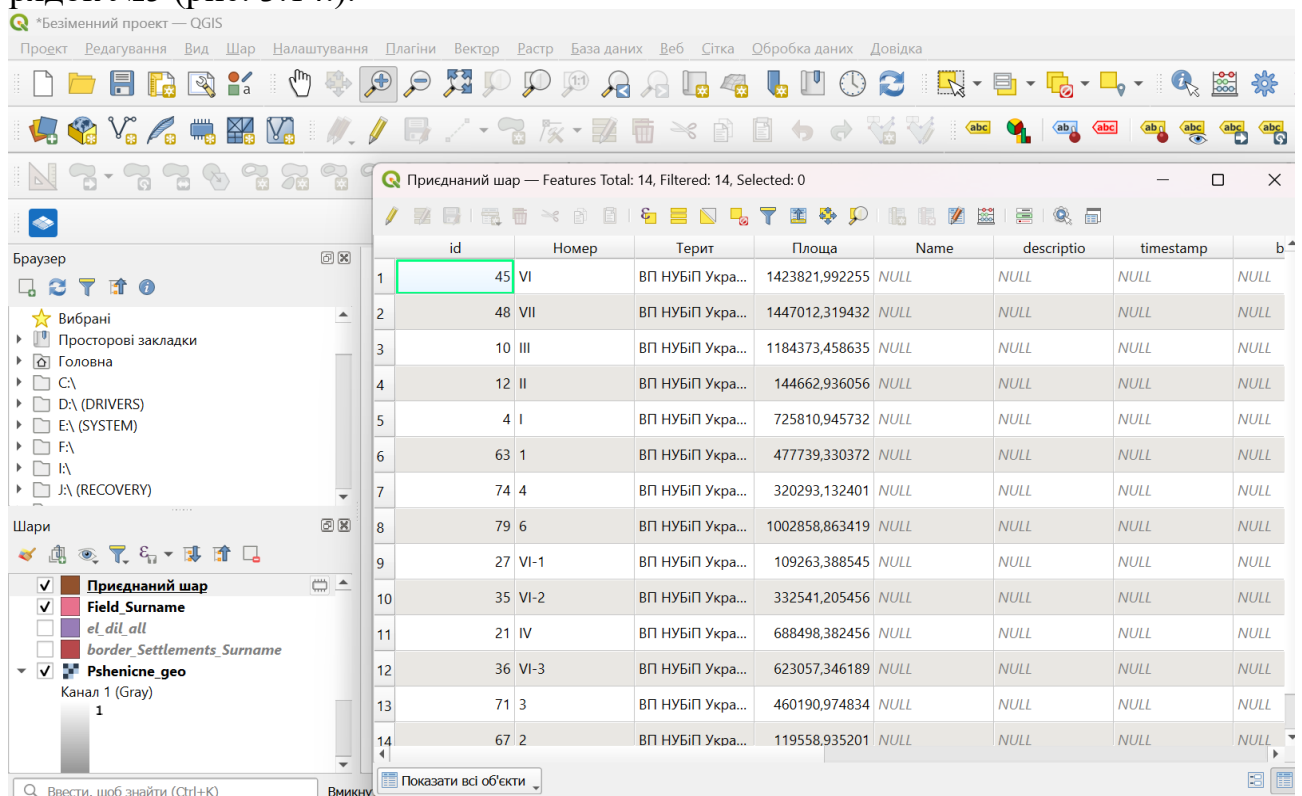
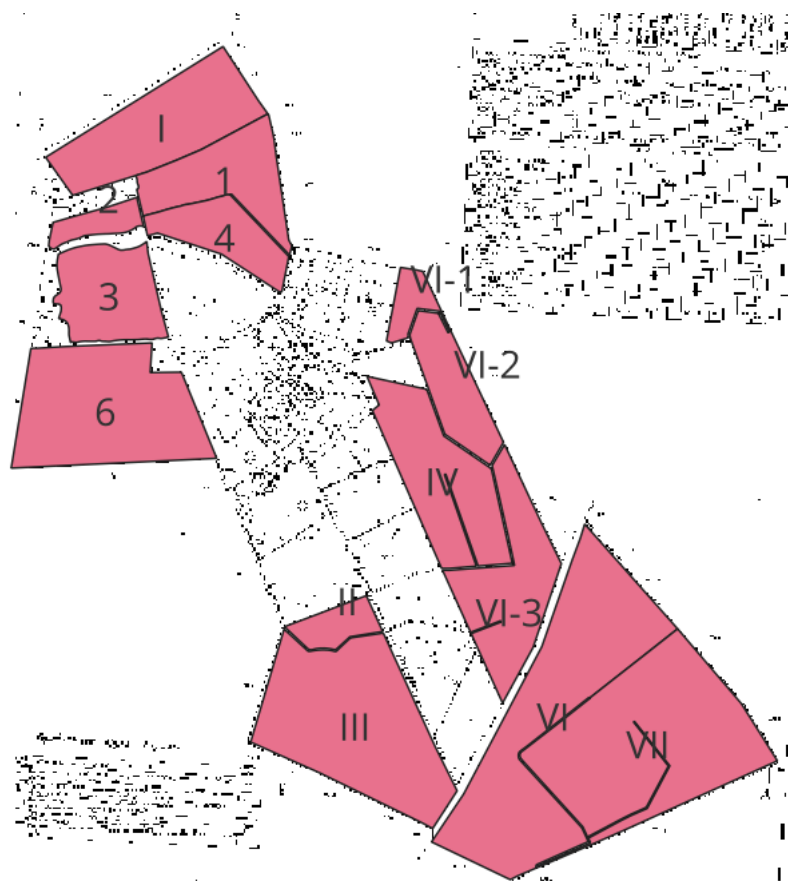


Рис. 3.14. Результат автозаповнення атрибутів шейп-файлу по попаданню в межу іншого шейпфайлу

Хід роботи

1. Ознайомтесь з Методичними вказівками до виконання практичної роботи.
2. Створіть шаблон звіту в MS Word та збережіть його під назвою Work03_Surname.docx. Хід виконання наступних пунктів роботи повинен бути підтверджений наявністю скріншотів у звіті.
3. Запустіть програмний засіб QGIS.
4. Відкрийте створений в минулій лабораторній роботі проект Soil_#групи_Surname.
5. Знайдіть у вікні Шари шейпфайл з елементарними ділянками (el_dil_all.shp).
6. Відкрийте таблицю атрибутів. Які дані містить ця таблиця?
7. Ознайомтеся з властивостями полів. Які типи даних використовуються для полів агрохімічних показників та для площі? Які типи даних використовуються для полів з кодами? (Поле Id – містить номер елементарної ділянки)
8. Знайдіть у вікні Шари шейпфайл з полями (Field_Surname.shp). Відкрийте таблицю атрибутів. Які дані містить ця таблиця?
9. Знайдіть у вікні Шари шейпфайл з ґрунтами (Soil_Surname.shp). Відкрийте таблицю атрибутів. Які дані містить ця таблиця?
10. Знайдіть у вікні Шари шейпфайл з межею (border_Settlements_Surname.shp). Відкрийте таблицю атрибутів. Які дані містить ця таблиця?
11. До атрибутивної таблиці шейпфайлу з ґрунтами (Soil_Surname.shp) додайте поля (колонки): Назва_гр (Тип – Текст, довжина – 180 символів) Площа_гр (Тип – Десяткове число, Довжина 10, Точність 3.)
12. Внести атрибутивні дані до шейпфайлу Soil_Surname.shp (дані шифрів агровиробничих груп ґрунтів отримати з растрового зображення, площу обчислити)
13. До атрибутивної таблиці шейпфайлу з полями (Field_Surname.shp) додайте поля (колонки): Номер_поля (Тип – Текст, довжина – 10 символів) Площа_Поля (Тип – Десяткове число, Довжина 10, Точність 3.)
14. Внести атрибутивні дані до шейпфайлу Field_Surname.shp (Номерів полів взяти з скріна карти, площу обчислити)



15. До атрибутивної таблиці шейпфайлу з полями (border_Settlements_Surname.shp) додайте поля (колони): Територія (Тип – Текст, довжина – 100 символів) та заповніть ВП НУБіП України «Агрономічна дослідна станція»

16. Автозаповніть поля шейп-файлу Field_Surname.shp полями шейп-файлу межі проекту border_Settlements_Surname.shp.

17. Якщо створився тимчасовий файл приєднання – експортуйте його в шейпфайл Field_Surname2.shp

2. Збережіть файл звіту у форматі PDF: Work03_Surname.pdf та разом з файлами шейп-файлів межі проекту, земельних ділянок, агровиробничих груп ґрунтів завантажте його на оцінювання одним архівом.

3. Дайте відповіді усно на контрольні запитання.

Контрольні запитання.

1. Що таке атрибутивні дані?
2. Які є інструменти по роботі з атрибутивними даними в QGIS?
3. Як можна переглянути атрибутивні дані в QGIS?
4. Чи можна автозаповнити поля?
5. Чи можна відновити видалені поля в QGIS

Практична робота 4

Тема: Моделювання бази даних.

Мета роботи: вивчення основних методів та набуття навичок проектування бази даних з використанням UML

Короткі теоретичні відомості

Проектування бази даних складається з кількох етапів: створення концептуальної моделі, логічної моделі та фізичної реалізації.

Під час концептуального проектування здійснюється побудова семантичної моделі предметної області, тобто інформаційної моделі найбільш високого рівня абстракції. Така модель створюється без орієнтації на якусь конкретну СКБД і модель даних. Терміни «семантична модель», «концептуальна модель» і «інфологічна модель» є синонімами. На цьому етапі визначаються об'єкти, зв'язки між об'єктами, атрибути, ключові атрибути. Конкретний вид і зміст концептуальної моделі бази даних визначається обраним для цього формальним апаратом. Зазвичай використовуються графічні нотації, подібні ER-діаграм або UML.

Найчастіше концептуальна модель бази даних включає в себе:

- опис інформаційних об'єктів або понять предметної області та зв'язків між ними.
- опис обмежень цілісності, тобто вимог до допустимих значень даних і до зв'язків між ними.

Під час логічного проектування виконується створення схеми бази даних на основі конкретної моделі даних, наприклад, об'єктно-орієнтованої PostgreSQL. На думку авторів цих методичних вказівок – найбільш зручною є подання логічної моделі через діаграму класів UML. На етапі логічного проектування враховується специфіка конкретної моделі даних, але може не враховуватися специфіка конкретної СКБД.

Перетворення концептуальної моделі в логічну модель, як правило, здійснюється за формальними правилами. Цей етап може бути в значній мірі автоматизований.

На етапі фізичного проектування відбувається реалізація створеної схеми бази даних для конкретної СКБД. Специфіка конкретної СКБД може включати в себе обмеження на іменування об'єктів бази даних, обмеження на підтримувані типи даних та інші. Крім того, специфіка конкретної СКБД при фізичному проектуванні включає вибір рішень, пов'язаних з фізичним середовищем зберігання даних (вибір методів управління дисковою пам'яттю, поділ бази даних по файлам і пристроям, методів доступу до даних), створення індексів та інші.

UML (англ. Unified Modeling Language) – уніфікована мова моделювання, використовується в області об'єктно-орієнтованого програмування. Для розуміння мови UML необхідно засвоїти її концептуальну

модель, яка включає в себе три складові частини: основні будівельні блоки мови; правила їх поєднання; деякі загальні для всієї мови механізми.

Словник мови UML включає три види будівельних блоків: сутності; відносини; діаграми. Сутності – це абстракції, які є основними елементами моделі. Відносини пов'язують різні сутності. Діаграми групують, представляють інтерес сукупності сутностей.

В UML є чотири типи сутностей: структурні; поведінкові; групуючі; анотаційні. Сутності є основними об'єктно-орієнтованими блоками мови. З їх допомогою можна створювати коректні моделі.

Структурні сутності як правило представляють собою статичні частини моделі, відповідні концептуальним або фізичним елементам системи. Існує кілька різновидів структурних сутностей.

Клас (Class) - це опис сукупності об'єктів із загальними атрибутами, операціями, відносинами і семантикою. Клас реалізує один або кілька інтерфейсів.

Інтерфейс (Interface) - це сукупність операцій, які визначають сервіс (набір послуг), що надається класом або компонентом.

Прецедент (Use case) - це опис послідовності виконуваних системою дій.

Кооперація (Collaboration) визначає взаємодію; вона являє собою сукупність ролей і інших елементів, які, працюючи спільно, роблять деякий кооперативний ефект, що не зводиться до простої суми доданків.

Компонент (Component) - це фізична замінна частина системи, яка відповідає деякому набору інтерфейсів і забезпечує його реалізацію.

Поведінкові сутності (Behavioral things) є динамічними складовими моделі UML. Це дієслова мови: вони описують поведінку моделі в часі і просторі. Взаємодія (Interaction) - це поведінка, суть якої полягає в обміні повідомленнями (Messages) між об'єктами в рамках конкретного контексту для досягнення певної мети.

Існує всього два основних типи поведінкових сутностей.

- згруповані сутності - це блоки, на які можна розкласти модель. Пакети (Packages) представляють собою універсальний механізм організації елементів в групі. У пакет можна помістити структурні, поведінкові і навіть інші сутності.

- анотаційні сутності - пояснювальні частини моделі UML. Це коментарі для додаткового опису, роз'яснення або зауваження до будь-якого елементу моделі. Є тільки один базовий тип анотаційних елементів - примітка (Note).

База даних, яку ми проектуємо, повинна за запитом робити вибірку даних:

- чи є в певного громадянина земельна ділянка в базі даних створеної нами ГІС системи; якщо так, які земельні ділянки перебувають у власності;
- сумі заборгованості щодо сплати земельного податку (звільнення від сплати земельного податку);

- історія сплати земельного податку громадянином за певний період тощо.

Перспективи діаграми класів.

Вибір точки зору залежить від того, наскільки далеко ви просунулися в процесі розвитку. Під час формулювання моделі домену, наприклад, ви рідко проходите далі концептуальної точки зору. Моделі аналізу зазвичай містять поєднання концептуальних і специфікаційних точок зору. Розробка моделі дизайну зазвичай починається з великого акценту на перспективі специфікації та розвивається в перспективі впровадження.

Діаграму можна інтерпретувати з різних точок зору:

- Концептуальний: представляє концепції в домені;
- Специфікація: фокус зосереджений на інтерфейсах абстрактного типу даних (ADT) у програмному забезпеченні;
- Реалізація: описує, як класи реалізовуватимуть свої інтерфейси;
- Перспектива впливає на кількість деталей, які потрібно надати, і на види стосунків, які варто представити. Як ми вже згадували вище, ім'я класу є єдиною обов'язковою інформацією.

Модель UML якщо її концептуалізувати у вигляді класів, дає гарне уявлення про взаємозв'язок сутностей, його легше збагнути.

Діаграма класів (class diagram) – статичне представлення структури моделі та є найпопулярнішою діаграмою UML. Відображає статичні елементи, такі як: класи, типи даних, їх зміст та відношення.

Клас в мові UML служить для позначення множини об'єктів, які мають однакову структуру, поведінку і відносини з об'єктами інших класів. Графічно клас зображується у вигляді прямокутника, який додатково може бути розділений горизонтальними лініями на розділи або секції. У цих розділах можуть зазначатися ім'я класу, атрибути (ознаки, змінні) та операції (методи) (Рис .4.1).

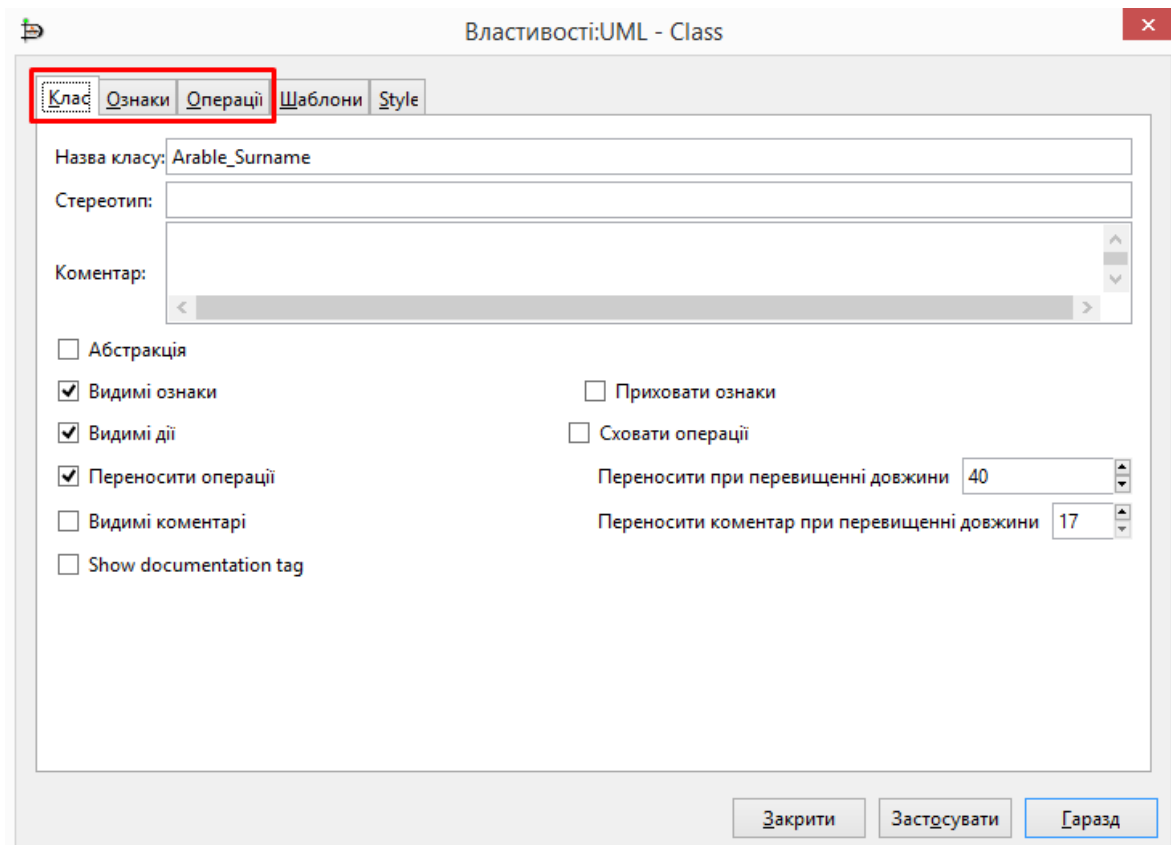


Рис. 4.1. Властивості класу в мові UML

Діаграма класів UML — це графічна нотація, яка використовується для побудови та візуалізації об'єктно-орієнтованих систем. Діаграма класів в уніфікованій мові моделювання (UML) — це тип статичної структурної діаграми, яка описує структуру системи, показуючи системні:

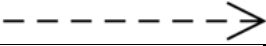
- класи,
- їхні атрибути,
- операції (або методи),
- і зв'язки між об'єктами.

Відношення між класами. Крім внутрішньої будови або структури класів на відповідній діаграмі вказуються відносини між класами. залежності; асоціації; узагальнення; реалізація. Базовими відносинами в мові UML є залежності, асоціації, узагальнення та реалізація. Кожне з цих відношень має власне графічне представлення на діаграмі, яке відображає взаємозв'язки між об'єктами відповідних класів (Таблиця 4.1.).

Таблиця 4.1.

Відношення між класами

	Асоціація
	Спрямована асоціація
	Успадкування
	Реалізація (Імплементация)

	Залежність
	Агрегація
	Композиція

Відношення залежності графічно зображуються пунктирною лінією між відповідними елементами зі стрілкою, направленою від класу-клієнту залежності до незалежного класу або класу-джерела. Для відносин залежності зумовлені ключові слова, які позначають деякі спеціальні його види:

«access» - для позначення доступності відкритих атрибутів і операцій класу-джерела для класів-клієнтів;

«bind» - клас-клієнт може використовувати деякий шаблон для своєї подальшої параметризації;

«derive» - атрибути класу-клієнту можуть бути обчислені по атрибутах класу-джерела;

«import» - відкриті атрибути і операції класу-джерела стають частиною класу-клієнту, так, ніби вони були оголошені безпосередньо в ньому;

«refine» - вказує, що клас-клієнт служить уточненням класу-джерела в силу причин історичного характеру, коли з'являється додаткова інформація в ході роботи над проектом.

Відношення асоціації відповідає наявності деякого відношення між класами. Дане відношення позначається суцільною лінією з додатковими спеціальними символами, які характеризують окремі властивості конкретної асоціації. Ім'я асоціації є необов'язковим елементом її позначення. Якщо воно задане, то записується з великої букви поруч з лінією відповідної асоціації. Для асоціації може позначатися кількість екземплярів об'єктів кожного класу, які беруть участь у зв'язку (0 - якщо жодного, 1 - якщо один, * - якщо багато). Можуть вказуватися мінімальна й максимальна кількість, наприклад, 0,1...* означає, що на відповідному кінці асоціації може не бути жодного екземпляра, бути один або багато.

Бінарна асоціація – служить для зображення довільного відношення між двома класами. Вона зв'язує в точності два різних класи й може бути ненаправленим або направленим відношенням. Окремий випадок бінарної асоціації - рефлексійна асоціація, що зв'язує клас із самим собою. Ненаправлена асоціація зображується лінією без стрілки. Як простий приклад ненаправленої бінарної асоціації можна розглянути відношення між двома класами – класом Компанія та класом Співробітник. Вони зв'язані між собою бінарною асоціацією «працює». Для даного відношення визначений наступний порядок читання - співробітник працює в компанії.



Рис. 4.2. Приклад бінарної асоціації

Спрямована бінарна асоціація зображується суцільною лінією із простою стрілкою на одній з її кінцевих крапок. Напрямок цієї стрілки вказує на те, який клас є першим, а який - другим. Як простий приклад спрямованої бінарної асоціації можна розглянути відношення між двома класами - класом Клієнт і класом Рахунок. Вони пов'язані між собою бінарною асоціацією з ім'ям Має, для якої визначено порядок проходження класів. Це означає, що конкретний об'єкт класу Клієнт завжди повинен вказуватися першим при розгляді взаємозв'язку з об'єктом класу Рахунок. Іншими словами, ці об'єкти класів утворюють кортеж елементів, наприклад, клієнт, рахунок_1, рахунок_2, ..., рахунок_n>.



Рис. 4.3 Приклад спрямованої бінарної асоціації

Узагальнення – Відносини між більше загальним поняттям (предком) і менш загальним поняттям (нащадком). Менш загальний елемент моделі повинен бути позгоджений з більш загальним елементом і може містити додаткову інформацію. Дане відношення використовується для подання ієрархічних взаємозв'язків між різними елементами мови UML, такими як пакети, класи, варіанти використання. Наслідування (Inheritance) - Спеціальний концептуальний механізм, за допомогою якого більш спеціальні елементи включають в себе структуру і поведінку більш загальних елементів. клас-нащадок має всі властивості і поведінкою класу-предка, а також має власні властивості і поведінку, які можуть бути відсутніми у класу-предка.

Реалізація (Realization) - це семантичне відношення між класифікаторами, при якому один класифікатор визначає сутність, а інший гарантує її виконання. Відносини реалізації в діаграмах класів зустрічаються між інтерфейсами і класами, що реалізують їх. Зображається відношення реалізації у вигляді пунктирної лінії з незафарбованою стрілкою, як щось середнє між відносинами узагальнення та залежності.

Відношення Агрегація. Як приклад відносини агрегації можна розглянути взаємозв'язок типу "частина-ціле", яка має місце між класом Системний блок персонального комп'ютера і його складовими частинами: Процесор, Материнська плата, Оперативна пам'ять, Жорсткий диск і Гнучкий диск. Використовуючи позначення мови UML, компонентний склад системного блоку можна представити у вигляді відповідної діаграми класів, яка в даному випадку ілюструє ставлення агрегації.

Композиція (Composition) - Різновид відносини агрегації, при якій складові частини цілого мають такий же час життя, що й саме ціле. Ці частини знищуються разом зі знищенням цілого. Графічно відношення композиції зображується суцільною лінією, один з кінців якої являє собою зафарбований

усередині ромб. Цей ромб вказує на той клас, який представляє собою клас-комполит. Решта класи є його "частинами".

Методичні вказівки

Для оцінювання якості ґрунтів база даних повинна містити дані про господарство, поля, ґрунтовий покрив та дані агрохімічної паспортизації.

- 1.
2. Завантажити програмний засіб Dia з free кодом за посиланням: <http://dia-installer.de/>.

3. Запустити файл встановлення  dia-setup-0.97.2-2-unsigned.exe
4. Встановлення програмного засобу триває до 1хв.

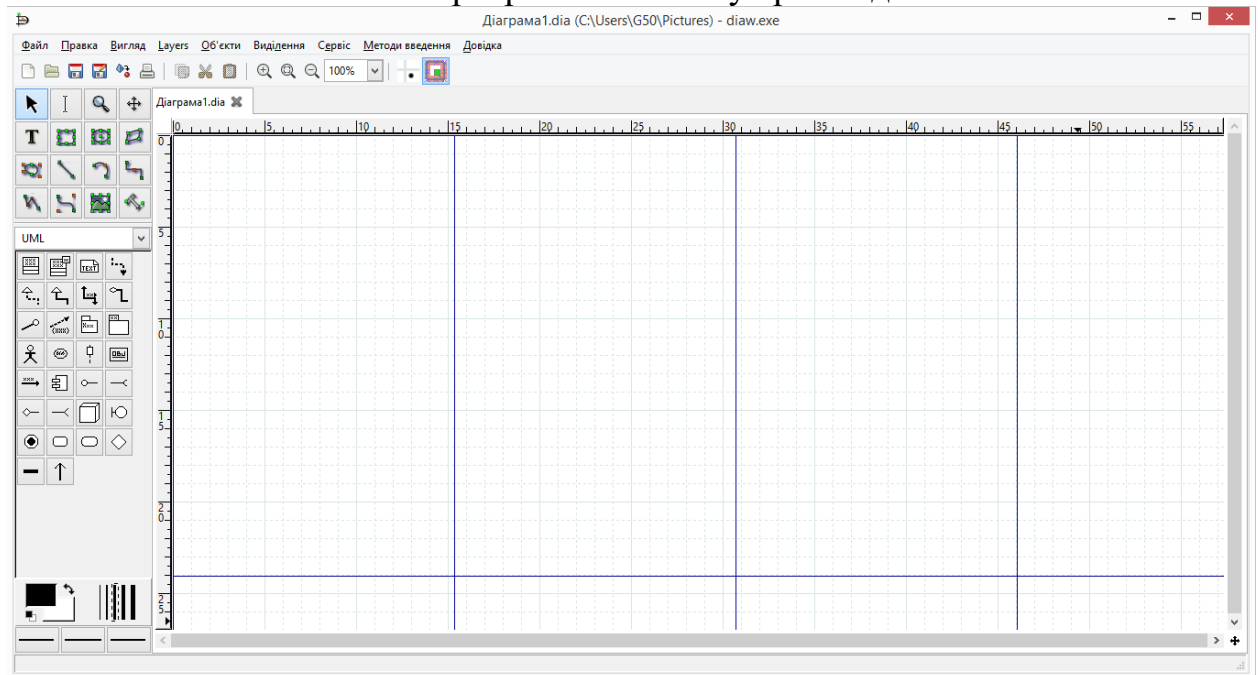
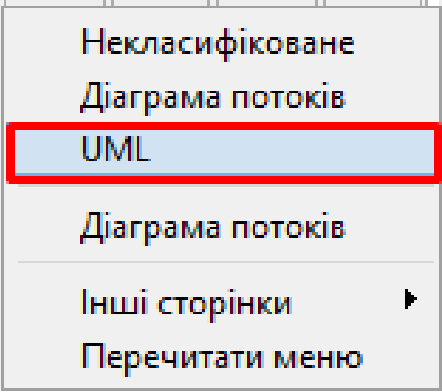


Рис. 4.4. Інтерфейс програми Dia

5. На лівій панелі обрати тип нотації для створення моделей – UML.

	
<i>Рис. 4.5. Обрання типу нотації</i>	<i>Рис. 4.6. Вибір методу створення схеми бази даних</i>

6. При обранні нотації нижче відкривається панель з шаблонами для створення моделі (рис. 4.6.).

7. Створити новий клас, ввести назву класу та додати ознаки.

!!!Всі поля мають бути заповнені або англійською мовою або латиницею (рис. 11.4. – 11.6.).

6.1. Обрати шаблон клас (рис. 4.7.);

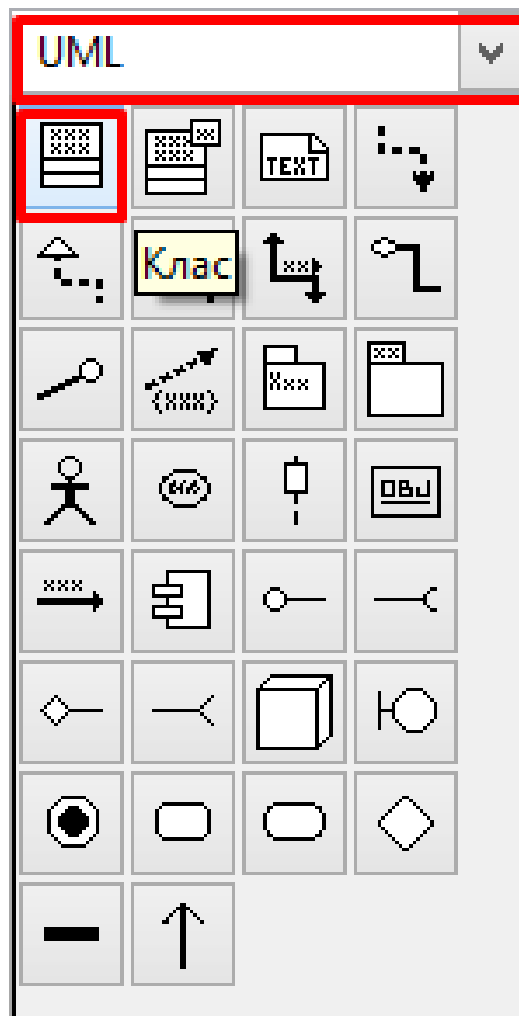


Рис. 4.7. Створення нового класу

- 6.2. Викреслити в робочому просторі шаблон Клас;
- 6.3. Натиснути правою кнопкою мишки на щойно створеному шаблоні та обрати властивості;
- 6.4. На закладці Клас внести назву класу (з великої літери), рис. 4.8.;
- 6.5. На закладці Ознаки внести назву ознаки (з малої літери літери) , рис. 11.6.;
- 6.6. Заповнити Тип;
- 6.7. Заповнити Значення;
- 6.8. Звстосувати зміни.
8. Зберегти зміни

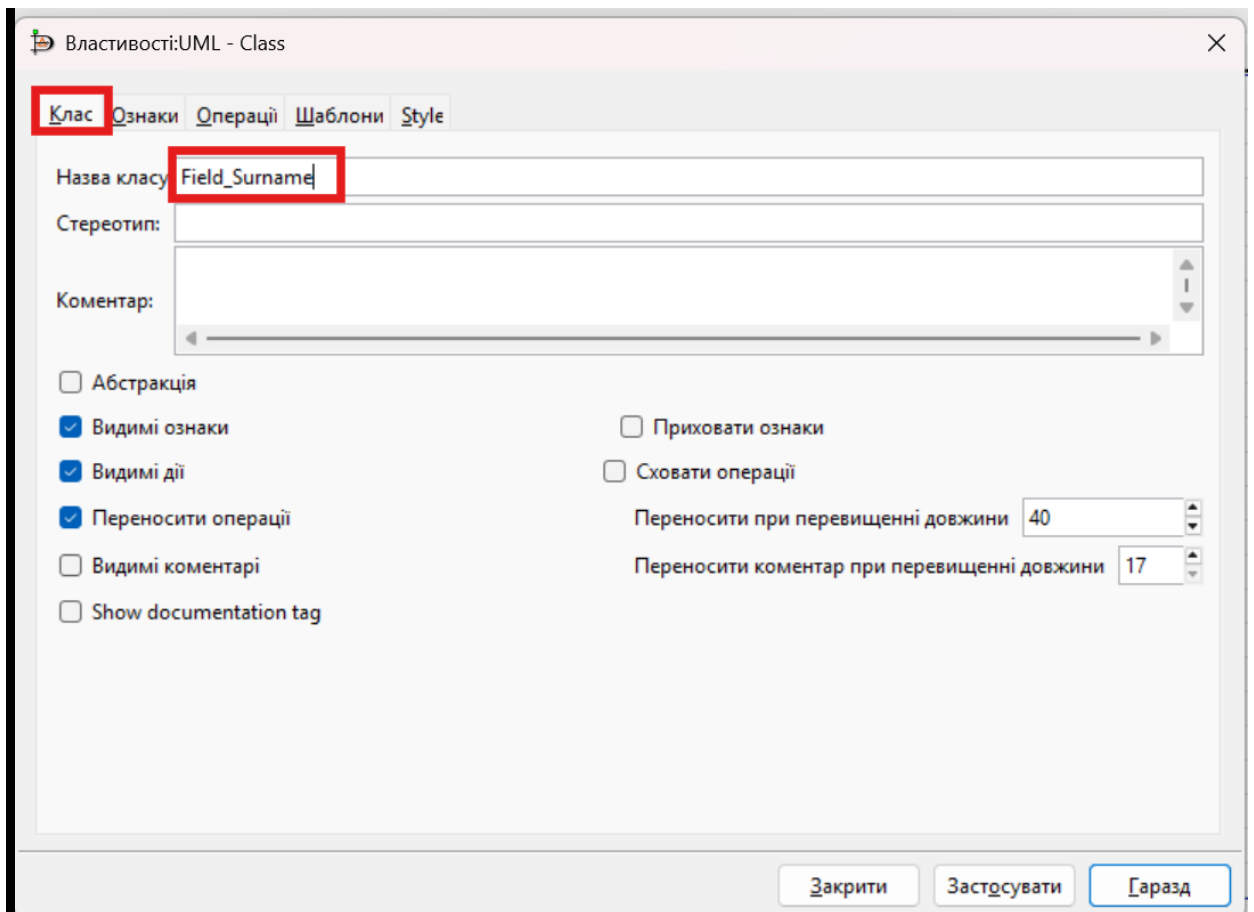


Рис. 4.8. Закладка **Клас** у вікні **Властивості**

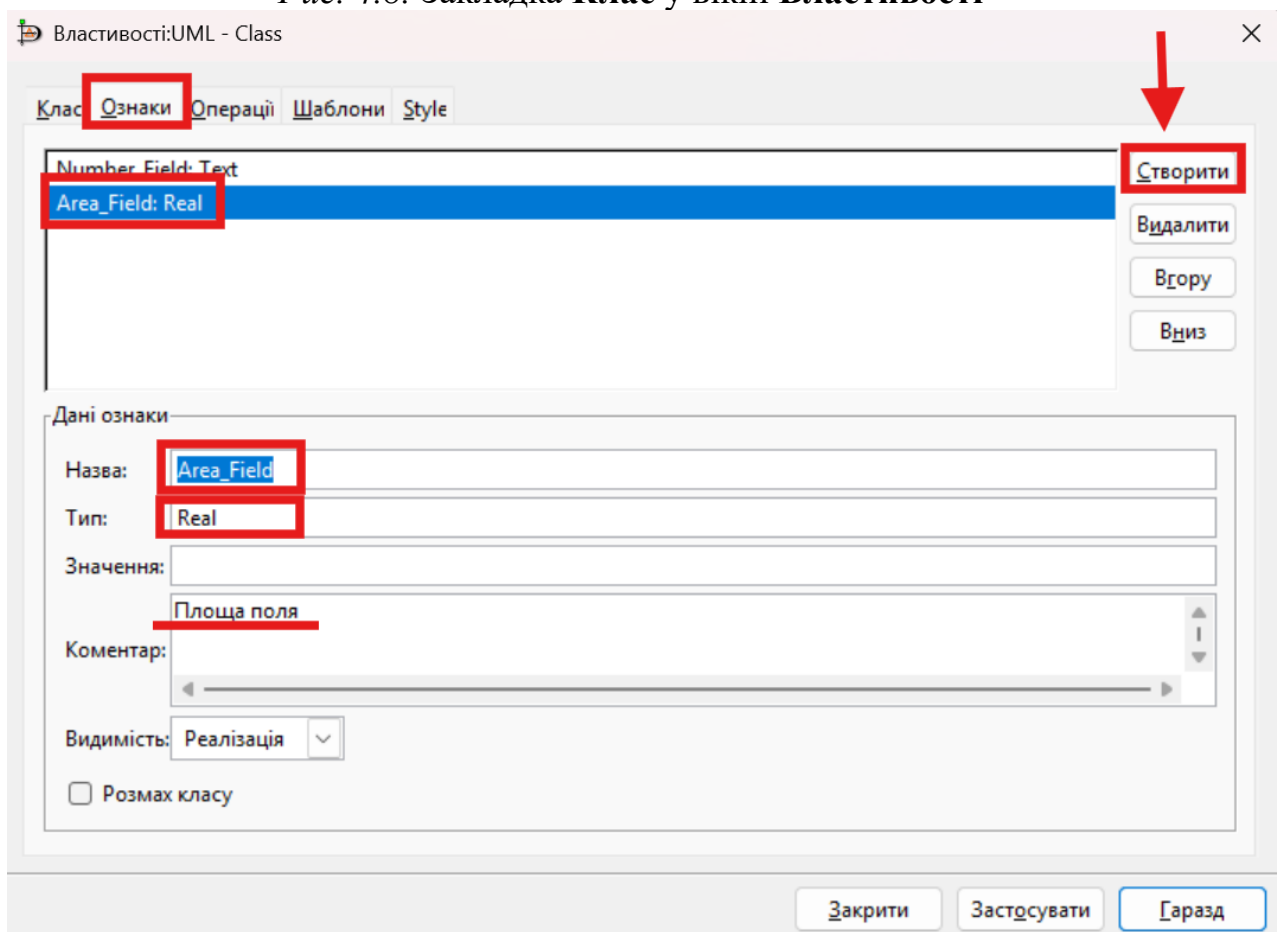


Рис. 4.9. Закладка **Ознаки** у вікні **Властивості**

9. На першому етапі визначається Назва класу:
Ім'я класу з'являється в першому розділі.
10. На другому етапі визначаються Атрибути класу:
 - 10.1. Атрибути показані у другому розділі.
 - 10.2. Тип атрибута вказується після двокрапки.
 - 10.3. Атрибути відображаються на змінних-членах (членах даних) у коді.
11. На третьому етапі визначаються Операції класу (методи):
 - 11.1. Операції показані в третьому розділі. Це послуги, які надає клас.
 - 11.2. Тип повернення методу відображається після двокрапки в кінці підпису методу.
 - 11.3. Повернений тип параметрів методу відображається після двокрапки після назви параметра. Операції відображаються на методи класу в коді
12. Зберегти набір класів, рис. 4.10.

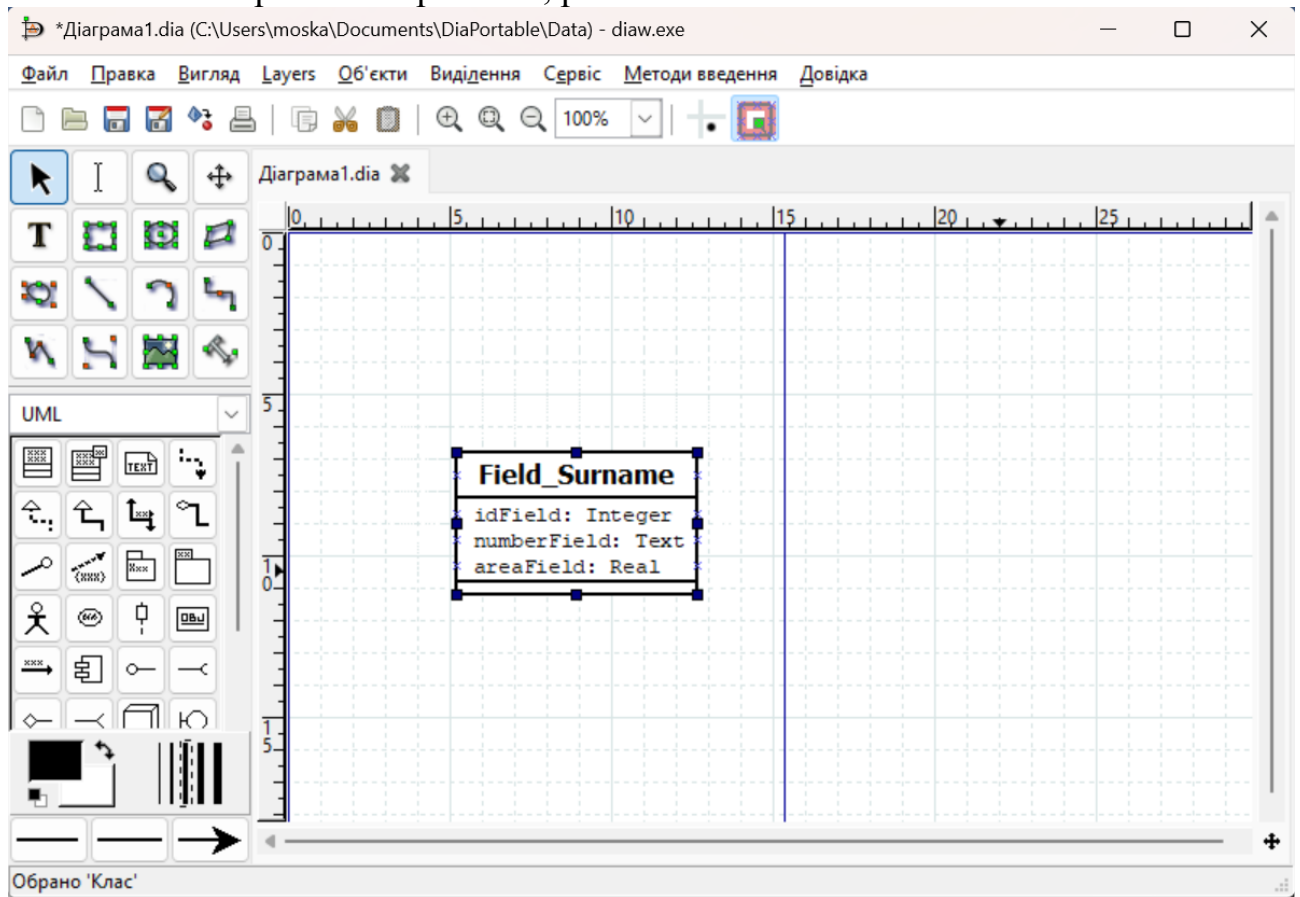
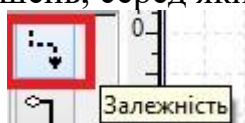


Рис. 4.10. Приклад класу в Dia

Користуючись панеллю інструментів в Dia можна встановити ряд відношень, серед яких:





Між класами мають бути встановлені відношення, а між окремими ознаками - бінарні зв'язки.

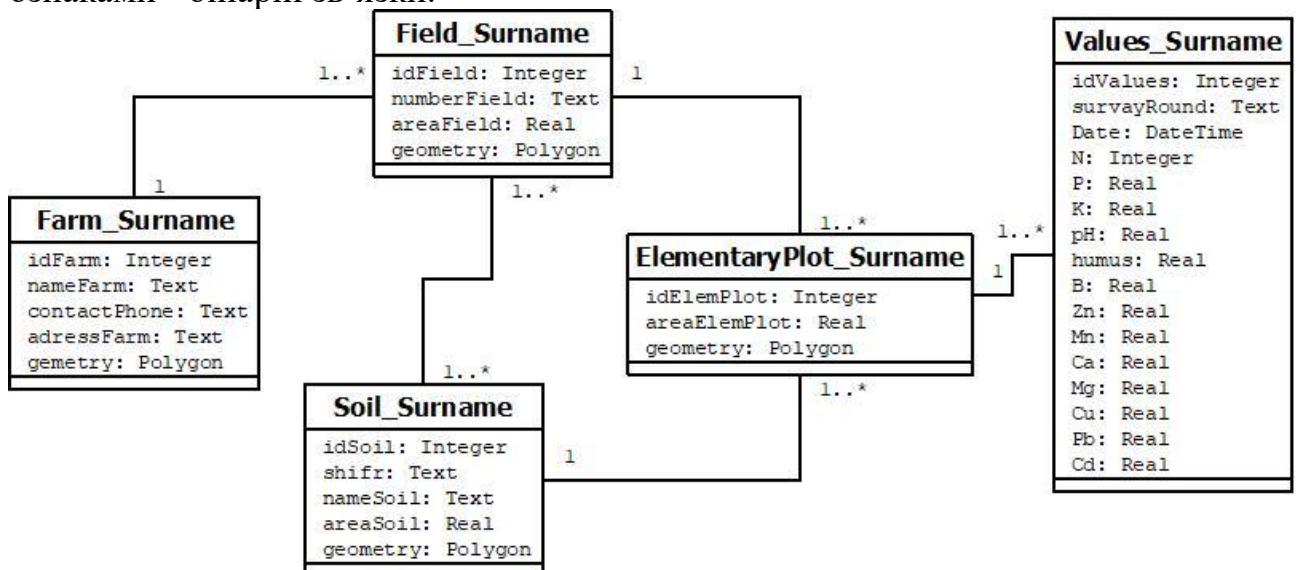


Рис. 4.11. Приклад визначення відношень між класами та сутностями в Dia

Хід роботи

1. Ознайомтесь із методичними вказівками до виконання практичної роботи.
2. Створіть шаблон звіту у MS Word під назвою Work4_Surname.docx.
3. Усі кроки виконання фіксуйте чіткими та читабельними скріншотами у звітах.
4. Проаналізуйте модель бази даних подану на рис. 4.11. Що потрібно додати? Чи всі типи даних коректно подані?
5. Запустіть векторний графічний редактор Dia (або його portable-версію). ПРИМІТКА! Можна застосовувати проектування «на папері»
6. Створіть початкову концептуальну модель бази даних, що складається з діаграм полів та агровиробничих груп ґрунтів.
7. Додайте до діаграм класифікацію об'єктів згідно з проектом Soil_#групи_Surname та збережіть їх в окремих файлах.
8. Створіть у моделі нові класи з назвами та атрибутами латиницею (ті що не були описані в проекті Soil_#групи_Surname).
9. Встановіть відносини та зв'язки між сутностями всіх класів концептуальної моделі.
10. Збережіть підсумкові звіти у форматі PDF - Work04_Surname.pdf.
11. Сформууйте архіви з файлами проектів Dia та відповідними PDF-звітами для завантаження на eLearn.
12. Усно дайте відповіді на контрольні запитання.

Контрольні запитання.

1. Що таке концептуальна модель бази даних?
2. Що є логічна модель бази даних?
3. Що таке UML модель?
4. Які є типи сутностей у мові UML?
5. Назвіть та охарактеризуйте структурні сутності.
6. Що таке діаграми UML моделі бази даних?
7. Що таке класи в UML моделі?
8. Що таке бінарне відношення? З якою метою встановлюється?
9. Які є типи зв'язків у мові UML?
10. Яка кратність зв'язку може бути встановлена при зв'язку-асоціації?

