

# Використання фреймворку Qt для створення застосунку для ведення нотаток

## 1. Які конкретні задачі планували вирішувати за допомогою цієї бібліотеки?

Основною задачею проекту було створення зручного, милого та інтуїтивно зрозумілого десктопного застосунку для нотаток. Користувач (я) мав отримати інструмент для ведення особистих записів з такими функціями:

- Додавання нових нотаток:  
при натисканні кнопки "Додати" створюється нова нотатка із дефолтним заголовком "New note".
- Редагування тексту нотатки:  
використовується текстовий редактор QTextEdit із можливістю змінювати шрифт і колір тексту через QFontDialog та QColorDialog.
- Перейменування нотатки:  
реалізовано через контекстне меню для зручності управління заголовками записів.
- Видалення нотаток:  
при натисканні кнопки "Видалити" з'являється вікно підтвердження дії QMessageBox, після чого нотатка видаляється.
- Пошук нотаток за назвою:  
текст у рядку пошуку автоматично фільтрує список нотаток, залишаючи лише ті, заголовки яких містять введений текст.
- Закріплення нотаток:  
важливі нотатки можна закріпити, щоб вони залишалися на початку списку. При цьому порядок закріплених нотаток не змінюється при їх модифікації.
- Експорт нотаток у .txt формат:  
реалізовано можливість збереження тексту нотатки у файл через діалог збереження QFileDialog.
- Зміна теми інтерфейсу:  
реалізовано перемикання між темною та світлою темами інтерфейсу.

Qt дозволив ефективно реалізувати ці функції завдяки великій кількості готових класів для GUI.

## 2. Чому було обрано саме цю бібліотеку, а не аналоги?

Було обрано саме Qt, оскільки він є кросплатформним, код пишеться вже знайомою мовою C++ з використанням набору інструментів фреймворку: готових класів для роботи з графічним інтерфейсом, обробки подій через систему сигналів і слотів, зручне збереження даних у форматі XML. Оскільки цей фреймворк є досить популярним, це значно збільшувало шанси знайти відповіді на свої питання на таких ресурсах як YouTube чи Stack Overflow. Аналоги були менш гнучкими, вони не мали такого багатого функціоналу для роботи з текстом та кастомізації інтерфейсу.

## 3. Наскільки просто та зрозуміло було отримати, встановити, налаштувати та почати використовувати цю бібліотеку?

Отримати, встановити, налаштувати та почати використовувати фреймворк було просто, адже він у вільному доступі та в мережі було вдосталь інструкцій для цього. Зокрема я використала інформацію з сайту Стенфордського університету, який знайшла за запитом в пошуковику ([https://web.stanford.edu/dept/cs\\_edu/resources/qt/install-mac](https://web.stanford.edu/dept/cs_edu/resources/qt/install-mac)).

Інтеграція Qt із компілятором була автоматичною, що є вагомою перевагою. Документація та короткі відео уроки на YouTube допомогли швидко зрозуміти принципи роботи з основними класами.

На YouTube є великий вибір уроків по Qt, і було легко знайти відео про використання потрібного мені класу чи методу, подивитись як вони працюють

(<https://youtube.com/playlist?list=PL2D1942A4688E9D63&si=VV4hMo-LiLX1Phfz>).

## 4. Наскільки зрозумілою та корисною була документація бібліотеки?

Офіційна документація Qt є чітко структурованою та детальною, що дає змогу легко знайти потрібний клас або метод. До кожного методу є пояснення параметрів, приклади використання та приклади поширених помилок.

Також я використовувала ChatGPT та інші ШІ для швидкого ознайомлення з проблемами та пошуку оптимальних методів їх вирішення. Іноді це дозволяло уникати довготривалого пошуку інформації в інтернеті.

## 5. Наскільки було зрозуміло, як саме використовувати бібліотеку, які класи/методи/функції використовувати для вирішення поставлених задач?

В Qt насправді досить зручний інтерфейс та інтуїтивно зрозумілі класи, наприклад,

для GUI: QPushButton, QListWidget, QTextEdit;

для файлів: QFile, QTextStream;

для збереження даних: QDomStreamReader, QDomStreamWriter.

Основні класи мали логічні назви, що значно полегшило пошук потрібного функціоналу.

Також сам фреймворк дає багато непоганих підказок при виникненні помилок у коді.

#### 6. Наскільки зручно було використовувати бібліотеку, чи не треба було писати багато надлишкового коду?

Використання Qt значно зменшило обсяг написання “рутинного” коду завдяки великій кількості готових класів для графічного інтерфейсу та роботи з файлами. Система слотів і сигналів усунула необхідність ручного зв’язування компонентів коду. Єдиними випадками де мені довелося писати додатковий код були кастомізація списку нотаток (NoteListWidget) та кастомізація віджета самої нотатки (NoteWidget).

#### 7. Наскільки зрозумілою була поведінка класів/методів/функцій з бібліотеки?

Усі класи передбачувано виконували свої функції, тому з цим не виникло якихось особливих проблем. Методи майже завжди видавали зрозумілі повідомлення про помилки, як вже було зазначено.

#### 8. Наскільки зрозумілою була взаємодія між різними класами/методами/функціями цієї бібліотеки, а також взаємодія між бібліотекою та власним кодом?

Як вже було зазначено раніше, взаємодія між компонентами через сигнали та слоти спростила інтеграцію з кастомними класами проекту.

#### 9. Чи виникали якісь проблеми з використанням бібліотеки? Чи вдалось їх вирішити, як саме?

Загалом проблеми виникали лише через мою власну неувважність та недостатню кількість знань :)

Також були деякі складнощі з відображенням списку нотаток (NotesListWidget), які вирішились за допомогою застосування кастомного класу (NoteWidget).

10.Що хорошого можна сказати про цю бібліотеку, які були позитивні аспекти використання бібліотеки?

З позитивних аспектів з легкістю можна виділити легкість роботи з GUI, зручну систему сигналів та слотів, простоту роботи з текстом та файлами, а також наявність великої кількості прикладів та хорошої документації.

11.Що поганого можна сказати про цю бібліотеку, які були негативні аспекти використання бібліотеки?

Qt - досить складний інструмент для новачків, зокрема через те, що є дуже велика кількість нових класів та функцій, розібратись в яких з першого разу досить важко. Іноді мені доводилось багато разів повністю переписувати великі частини коду через його складність та плутанину з зв'язками компонентів. Тож, хоч Qt і здається зрозумілим з першого погляду, на практиці все не так просто.

12.Якби довелось вирішувати аналогічну задачу, але вже враховуючи досвід використання в цій лабораторній роботі, що варто було б робити так само, а що змінити? Можливо, використати інші бібліотеки, чи використати інші можливості цієї бібліотеки, чи інакше організувати код, чи ще щось?

Якби довелося вирішувати аналогічну задачу, враховуючи досвід, отриманий під час написання цього проекту, я б все одно використовувала Qt для GUI, обробки подій та реалізації збереження даних.

Думаю, те, що дійсно корисно було б організувати по-іншому це збереження даних у іншому форматі (наприклад, JSON), а також змінити деякі частини коду з точки зору архітектури проекту. Наприклад, клас Notebook варто було б розділити на два окремі класи: один був би самим місцем зберігання нотаток, а інший використовувався б для управління логікою цих нотаток. Також можна було б використати ще якісь можливості Qt (наприклад, QML) для більш сучасного вигляду інтерфейсу, додати такі функції як зміна порядку нотаток в списку пересуванням, тощо.