

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Кафедра програмної інженерії

КУРСОВА РОБОТА
ПОЯСНЮВАЛЬНА ЗАПИСКА
з дисципліни “ Об’єктно -орієнтоване програмування”
«Термінологія»

Керівник, Проф.

Бондарєв В.М.

Студент гр. ПЗПІ-22-1

Мальцев Д.В.

Комісія:

Проф. _____ Бондарєв В.М.

Ст. викл. _____ Черепанова Ю.Ю.

Ст. викл. _____ Ляпота В.М.

Харків 2023

Харківський національний університет радіоелектроніки

Кафедра *програмної інженерії*

Рівень вищої освіти *перший (бакалаврський)*

Дисципліна *Об'єктно-орієнтоване програмування*

Спеціальність *121 Інженерія програмного забезпечення*

Освітня програма: *Програмна інженерія*

Курс 1 .

Група ПЗПІ-22 -1 .

Семестр 2 .

ЗАВДАННЯ

на курсовий проект студента

Мальцева Данила Володимировича

1 Тема проекту: «Термінологія»

2 Термін здачі студентом закінченого проекту: “16” - червня - 2023 р.

3 Вихідні дані до проекту:

Специфікація програми, методичні вказівки до виконання курсової роботи.

4 Зміст розрахунково-пояснювальної записки:

Вступ, специфікація програми, проектна специфікація, інструкція користувача, висновки

КАЛЕНДАРНИЙ ПЛАН

<i>№</i>	<i>Назва етапу</i>	<i>Термін виконання</i>
1	Видача теми, узгодження і затвердження теми	13.02.2023 - 14.03.2023 р.
2	Формулювання вимог до програми	15-03-2023 – 30-03-2023 р.
3	Розробка зберігання та пошуку даних	01-04-2023 – 11-04-2023 р.
4	Розробка функцій авторизації, реєстрації	11-04-2023 – 01-05-2023 р.
5	Розробка функцій зберігання та завантаження даних	02-05-2023 – 08-05-2023 р.
6	Тестування і доопрацювання розробленої програмної системи.	09-05-2023 – 18-05-2023 р.
7	Оформлення пояснювальної записки, додатків, графічного матеріалу	25-05-2023 – 27-05-2023 р.
8	Захист	05.06.2023 – 16.06.2023 р.

Студент _____

Мальцев Д.В.

Керівник _____

Бондарев В.М.

РЕФЕРАТ

Пояснювальна записка до курсової роботи: 39 с., 22 рис., 1 додаток, 7 джерел.

ТЕРМІНОЛОГІЯ, МОВА ПРОГРАМУВАННЯ C#, КЛАС, ООП.

Метою роботи є розробка програми “Термінологія ”, яка дозволить користувачам мати базу з термінами, записувати нові, редагувати існуючі та продивлятися зв’язані з ними.

В результаті отримана програма, що дозволяє створювати терміни, редагувати їх, продивлятися, видаляти, та зв’язувати.

Зміст

Вступ

Темою даної роботи є створення додатку для відслідковування та редагування термінів та їх визначень, з додаванням можливості вказувати терміни, пов'язані з кожним визначенням. Робота була виконана з використанням принципів об'єктно-орієнтованого програмування, і в якості основного інструменту було використано мову програмування C# та технологію Windows Forms для створення користувацького інтерфейсу.

Цей проект використовує JSON формат для зберігання даних, що дозволяє з легкістю додавати, редагувати або видаляти терміни, а також відслідковувати їх взаємозв'язки. Крім того, була реалізована функція пошуку, що дозволяє користувачеві легко знайти потрібний термін та його визначення.

Метою роботи є написання програми з можливістю створювати, редагувати та видаляти терміни. Результатом роботи є закріплення вивченого з дисципліни "Об'єктно-орієнтоване програмування", практика розробки користувацького інтерфейсу за допомогою Windows Forms, а також використання формату JSON для зберігання та обробки даних.

1 Специфікація Програми

1.1 Функції програми:

- Додавання термінів
- Редагування термінів
- Зв'язування термінів
- Видалення термінів
- Пошук термінів
- Перегляд термінів

Усі терміни зберігаються у файлі JSON формату „terms.json“

1.2 Інтерфейс користувача:

Після запуску програми користувач бачить стартове вікно, де він може почати шукати терміни, видалити всі існуючі терміни, додати новий термін

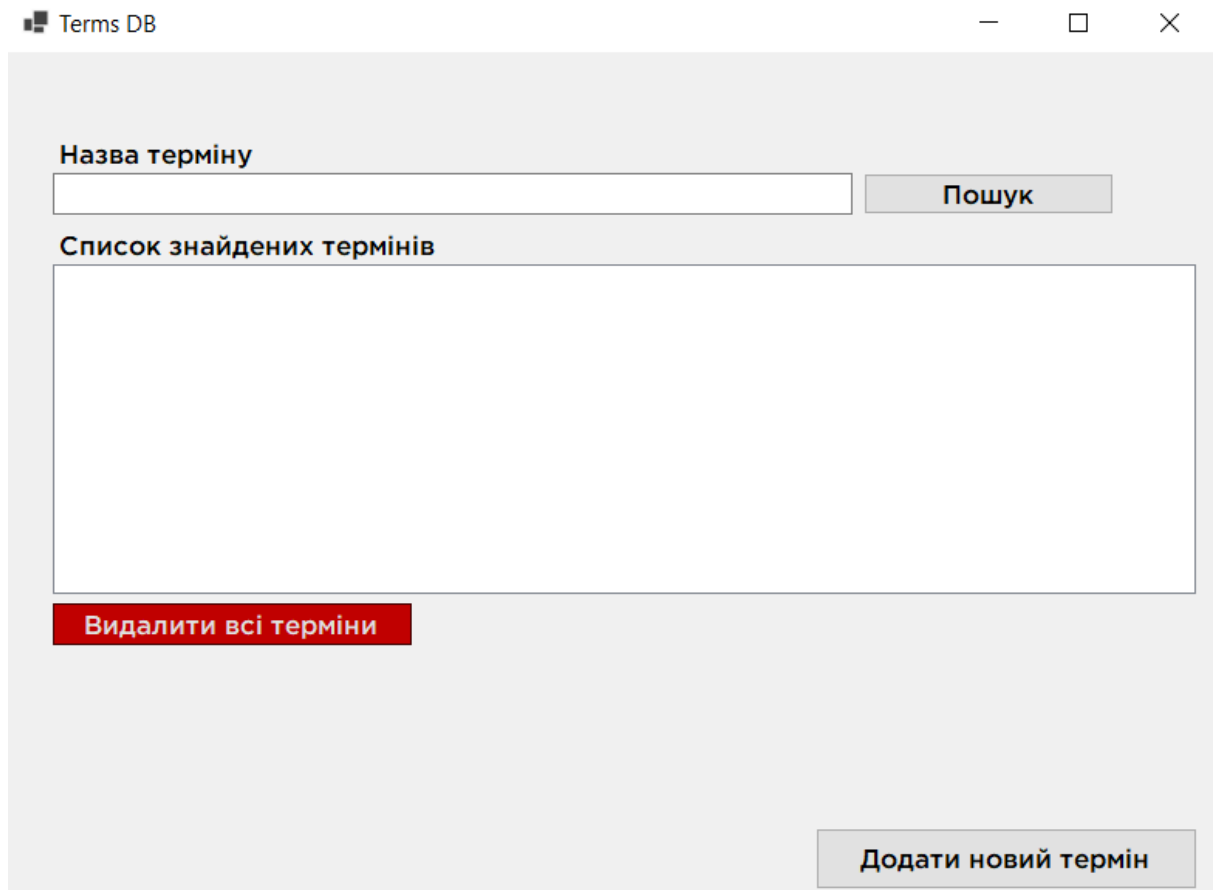


Рис. 1.1 – Вікно пошуку термінів

Для пошуку терміну потрібно ввести у поле «Назва терміну» назву терміну та натиснути кнопку «пошук». Після цього користувач побачить список термінів які існують. Натиснувши користувача перекине на вікно, де він зможе побачити визначення терміну та зв'язані терміни.

Для додавання терміну, користувач може натиснути кнопку «Додати новий термін». Після натискання користувачеві відкриється вікно, де він має змогу додати новий термін.

Для того, щоб видалити всі терміни, користувач може натиснути червону кнопку «Видалити всі терміни»

The screenshot shows a window titled 'FormAddTerm' with standard window controls (minimize, maximize, close). The form contains the following elements:

- A text input field labeled 'Назва терміну' (Term Name).
- A large text area labeled 'Визначення терміну' (Term Definition).
- A button labeled 'Додати термін' (Add Term) at the bottom left.
- A button labeled 'Зв'язати термін' (Link Term) at the bottom right.
- A button labeled 'назад' (Back) at the bottom left.
- A list box at the bottom right containing the following items:
 - Психологія
 - Наука
 - Психічне або душевне
 - Психічні процеси

Рис 1.2 – Вікно додавання нового терміну

Вікно додавання терміну є інтуїтивно зрозумілим та простим у використанні. Щоб додати новий термін, вам потрібно виконати наступні дії:

1. Натисніть на кнопку "Додати термін". З'явиться нове вікно для додавання терміну.
2. Введіть назву терміну у відповідне текстове поле "Назва терміну". Виберіть назву, яка ясно та коротко описує сам термін.
3. У полі "Визначення" введіть визначення терміну. Намагайтеся бути якомога більш точним та зрозумілим.
4. Для додавання термінів, пов'язаних з цим терміном, виберіть потрібний термін із спадного списку і натисніть на кнопку "Додати зв'язок".
5. Після того, як всі поля заповнено, натисніть на кнопку "Зберегти". Ваш термін тепер збережено і доступний для перегляду.
6. Якщо ви хочете скасувати додавання терміну, просто натисніть на кнопку "Назад".

The screenshot shows a web application window titled "FormEditTerm". The main content area is titled "Психологія" (Psychology). Below the title, there is a text box containing the definition: "Наука, що вивчає психічні явища (мислення, почуття, волю) та поведінку людини, пояснення якої знаходимо в цих явищах." (Science that studies mental phenomena (thought, feelings, will) and human behavior, the explanation of which we find in these phenomena). Below the definition, there is a section titled "Зв'язані терміни" (Related terms). This section contains a dropdown menu with two options: "Наука" (Science) and "Психічне або душевне" (Mental or spiritual). To the right of the dropdown menu, there are two buttons: "Редагувати" (Edit) and "Видалити" (Delete). At the bottom left, there is a button labeled "назад" (Back).

Рис 1.3 – Вікно перегляду визначення та редагування терміну

Вікно редагування терміну також використовується для перегляду визначення терміну. Для його використання виконуйте наступні дії:

1. Виберіть термін, який ви хочете переглянути або редагувати, натиснувши на нього у головному вікні програми.
2. Після вибору терміну з'явиться нове вікно, де ви можете переглянути деталі цього терміну. Ви побачите "Назва терміну" поле "Визначення", де ви можете переглянути визначення терміну, та список "Пов'язані терміни", де ви можете переглянути інші терміни, пов'язані з цим терміном.
3. Якщо ви хочете редагувати термін, то натисніть на кнопку «Редагувати» і вам буде доступне редагування терміну у полі «Визначення». Після редагування натисніть на кнопку «Зберегти», яка з'явилася замість кнопки «Редагувати»".
4. Якщо ви не хочете редагувати термін або хочете скасувати внесені зміни, натисніть на кнопку "Назад".
5. Якщо ви хочете видалити термін, натисніть на червону кнопку «Видалити». Вам відкриється вікно підтвердження, натиснувши так ви видалите термін.

2 Проектна Специфікація

Цей розділ описує структуру програми і коду, які були використані для створення програми управління термінами та визначеннями.

2.1 Структура програми

Програма складається з наступних основних частин:

- *Головне вікно*: Це вхідна точка програми, де користувач може переглядати список усіх термінів, вводити пошукові запити і переходити до різних форм.
- *Вікно додавання терміну*: В цьому вікні користувач може додавати нові терміни, вводячи назву терміну та його визначення.
- *Вікно редагування терміну*: В цьому вікні користувач може редагувати існуючі терміни, змінюючи їх назву, визначення або додаючи/видаляючи пов'язані терміни.

2.2 Код програми:

Програма розроблена на мові програмування C#, використовуючи парадигму об'єктно-орієнтованого програмування (ООП). Основні класи програми включають:

- *Клас **Term***: Цей клас представляє термін, з полями для назви, визначення і список пов'язаних термінів.
- *Класи **форм***: Це класи, які представляють окремі вікна програми (**FormMain**, **FormAddTerm**, **FormEditTerm**), з методами для обробки подій користувача та взаємодії з об'єктами **Term**.

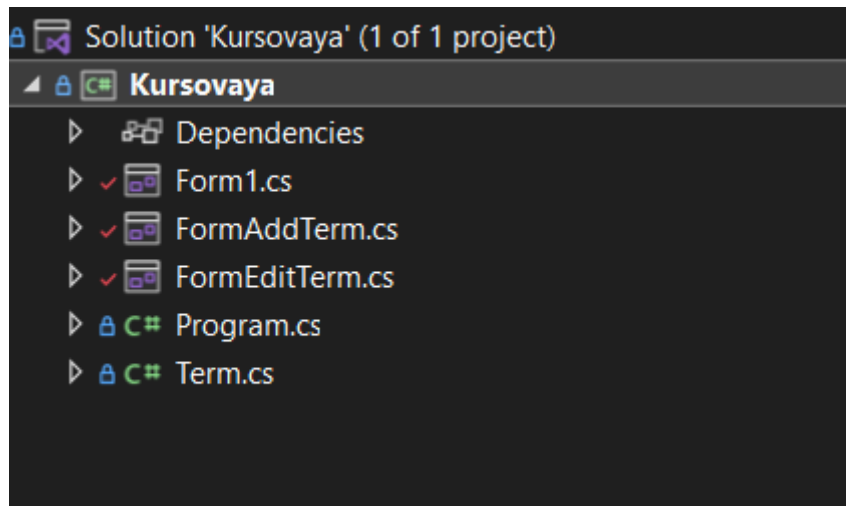


Рис 2.1 – Структура файлів проекту

```
1 reference
private void buttonAddRelation_Click(object sender, EventArgs e)
{
    string selectedTerm = listBoxRelatedTerms.SelectedItem.ToString();

    if (!newTerm.RelatedTerms.Contains(selectedTerm))
    {
        newTerm.RelatedTerms.Add(selectedTerm);
    }
    else
    {
        MessageBox.Show("Термін уже зв'язан.", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}
```

Рис 2.2 – Функція зв'язки термінів

Ця функція викликається при натисканні на кнопку "Зв'язати". Вона додає вибраний термін до списку пов'язаних термінів.

Процес виконання функції такий:

1. Спочатку вибраний термін зі списку "listBoxRelatedTerms" зберігається в змінній "selectedTerm".
2. Потім функція перевіряє, чи вже існує вибраний термін в списку пов'язаних термінів "newTerm.RelatedTerms". Якщо ні, термін додається до списку.

3. Якщо вибраний термін вже існує в списку пов'язаних термінів, користувачу виводиться повідомлення про помилку "Термін уже зв'язан", і новий термін не додається.

Ця функція допомагає уникнути дублювання термінів у списку пов'язаних, адже кожен термін повинен з'являтися в цьому списку тільки один раз.

```
1 reference
private void buttonAddTerm_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(textBoxTermName.Text) || string.IsNullOrEmpty(textBoxTermDefinition.Text))
    {
        MessageBox.Show("Будь ласка, введіть назву та визначення терміну", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
    if (terms.Any(t => t.Name == textBoxTermName.Text))
    {
        MessageBox.Show("Такий термін вже існує", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
    // Створюємо новий термін
    newTerm.Name = textBoxTermName.Text;
    newTerm.Definition = textBoxTermDefinition.Text;
    if (terms == null)
    {
        terms = new List<Term>();
    }
    terms.Add(newTerm);
    foreach (var relatedTermName in newTerm.RelatedTerms)
    {
        var relatedTerm = terms.Find(t => t.Name == relatedTermName);
        if (relatedTerm != null && !relatedTerm.RelatedTerms.Contains(newTerm.Name))
        {
            relatedTerm.RelatedTerms.Add(newTerm.Name);
        }
    }

    textBoxTermName.Clear();
    textBoxTermDefinition.Clear();

    SaveData();
    //ініціалізуємо новий термін
    newTerm = new Term { RelatedTerms = new List<string>() };
}
```

Рис 2.3 – Функція додавання терміну

Ця функція виконується, коли користувач натискає на кнопку "Додати термін". Вона додає новий термін у систему.

Ось, як працює ця функція:

1. Перш за все, вона перевіряє, чи користувач ввів назву терміну і його визначення. Якщо ні, з'являється повідомлення про помилку "Будь ласка, введіть назву та визначення терміну".
2. Далі перевіряється, чи існує термін з такою ж назвою. Якщо так, знову з'являється повідомлення про помилку "Такий термін вже існує".

3. Якщо назва і визначення терміну були введені коректно і такий термін ще не існує, створюється новий термін з введеною назвою та визначенням.
4. Якщо це перший термін, який додається в систему, створюється новий список термінів.
5. Новий термін додається до списку термінів.
6. Для кожного пов'язаного терміну, якщо він існує і вже не містить назву нового терміну у своїх пов'язаних термінах, назва нового терміну додається до його списку пов'язаних термінів.
7. Поле для введення назви та визначення терміну очищається.
8. Дані зберігаються за допомогою функції SaveData().
9. Ініціалізується новий термін для наступного додавання.

Ця функція дозволяє користувачам додавати нові терміни до системи, переконуючись, що вони мають відповідну назву, визначення і що термін з такою назвою ще не існує.

```
↑ reference
private void listBoxRelatedTerms_SelectedIndexChanged(object sender, EventArgs e)
{
    string selectedTermName = listBoxRelatedTerms.SelectedItem.ToString();

    Term selectedTerm = terms.Find(t => t.Name == selectedTermName);
    if (selectedTerm != null)
    {
        FormEditTerm formEditTerm = new FormEditTerm(selectedTerm, terms);
        this.Hide();
        formEditTerm.Show();
    }
}
```

Рис 2.4 – Функція переходу на зв'язаний термін

Ця функція викликається, коли користувач вибирає термін зі списку пов'язаних термінів, відображеного у listBoxRelatedTerms.

Ось, як ця функція працює:

1. Спочатку вибраний користувачем термін зчитується із `listBoxRelatedTerms`.
2. Далі знайти цей термін в списку всіх термінів. Якщо він існує...
3. ... створюється нове вікно редагування терміну (`FormEditTerm`), передаючи вибраний термін та список всіх термінів як параметри.
4. Поточне вікно (`this`) ховається, і нове вікно редагування терміну показується.

Ця функція дозволяє користувачам вибирати термін із списку пов'язаних термінів і переходити до вікна редагування для цього терміну, де вони можуть переглянути або змінити деталі цього терміну.

3 Інструкція Користувача

3.1 Установка програми

Щоб запустити цю програму на своєму комп'ютері, вам знадобляться наступні інструменти та ресурси:

1. **Microsoft Visual Studio:** це основний інструмент для розробки програм на мові C#. Ви можете завантажити його з офіційного веб-сайту Microsoft. Під час встановлення виберіть компоненти ".NET desktop development".
2. **.NET Framework 4.7.2 або вище:** це програмна платформа, яка потрібна для виконання програм на C#. Вона повинна бути вже встановлена на ваш комп'ютер під час встановлення Visual Studio. Якщо ні, ви можете завантажити її з офіційного веб-сайту Microsoft.
3. **Код проекту:** ви повинні мати доступ до коду програми. Він представлений у вигляді репозиторію на GitHub. Вам потрібно відкрити файл рішення (.sln) у Visual Studio.

Після того, як ви встановите всі необхідні інструменти та відкриєте проект у Visual Studio, ви можете запустити програму, натиснувши кнопку "Start" або натиснувши клавішу F5.

3.2 Робота з програмою

3.2.1 Головне вікно

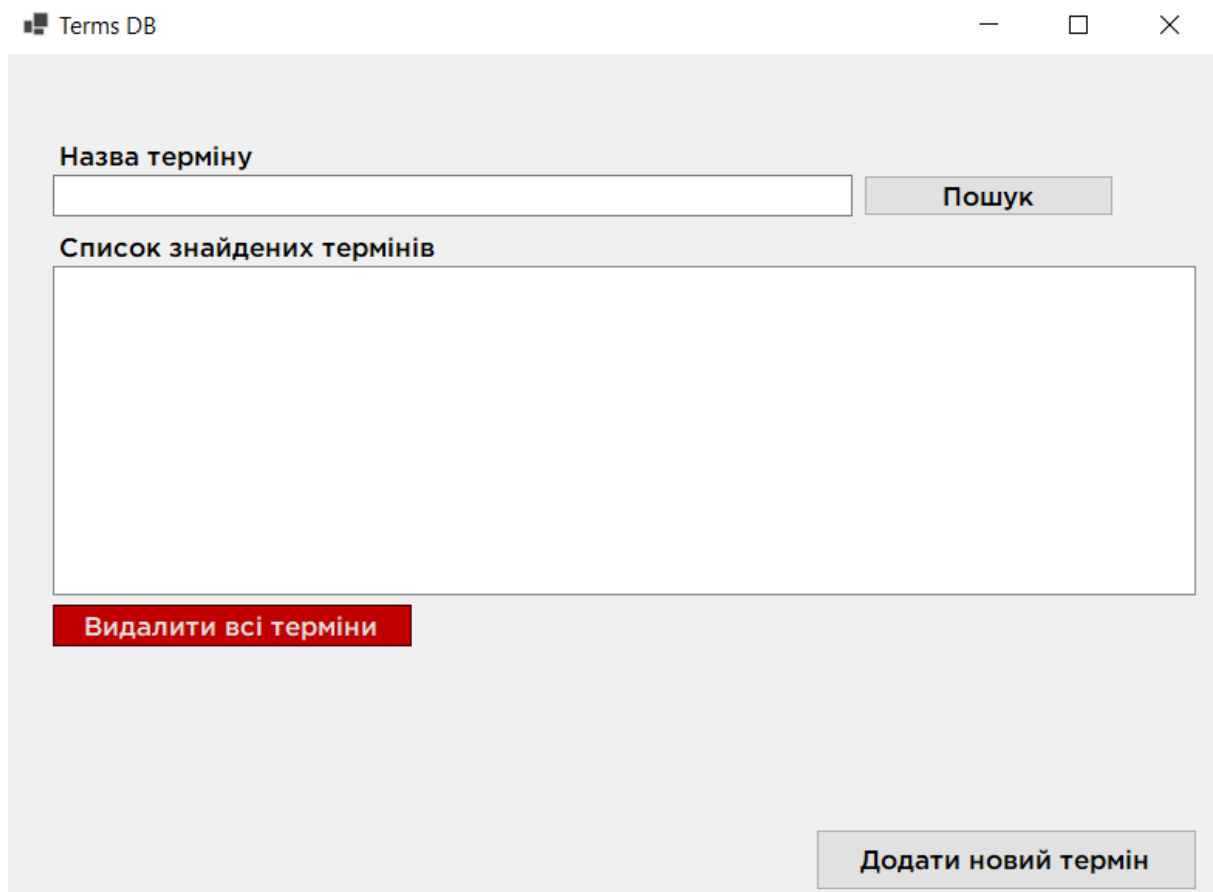


Рис. 3.2.1 – Вікно пошуку термінів

1. **Відкриття програми:** Після запуску програми ви зустрічаєтесь з головною сторінкою, де ви бачите кнопку для пошуку термінів ("Пошук"), поле для введення назви терміна ("Назва терміну"), кнопку для додавання нового терміну ("Додати новий термін") та список термінів ("Список знайдених термінів").
2. **Пошук терміну:** Щоб знайти конкретний термін, ви можете ввести його назву у поле "Назва терміну", розташоване вгорі сторінки, а потім натиснути кнопку "Пошук". Якщо введений термін входить у список термінів, він відобразиться в "Список знайдених термінів".
3. **Додавання нового терміну:** Для додавання нового терміну ви можете натиснути кнопку "Додати новий термін". При цьому відкривається нова форма, де ви можете ввести назву та визначення нового терміну.

4. **Перегляд та редагування терміну:** Щоб переглянути або редагувати термін, ви можете обрати його в списку. Після цього відкривається нова форма, де ви можете переглянути або змінити визначення терміну, а також додати або видалити пов'язані терміни.
5. **Видалення всіх термінів:** Щоб видалити всі терміни, ви можете натиснути кнопку "Видалити всі терміни". Перед видаленням програма запитає вас про підтвердження цієї дії, оскільки видалення є незворотнім.
6. **Збереження даних:** За допомогою функції "SaveData()" програма автоматично зберігає всі внесені зміни, включаючи додані, змінені та видалені терміни.

3.2.2 Вікно додавання

The screenshot shows a Windows-style application window titled "FormAddTerm". It contains two main input fields: "Назва терміну" (Term Name) at the top and "Визначення терміну" (Term Definition) below it. At the bottom left is a "назад" (Back) button. At the bottom right are two buttons: "Додати термін" (Add Term) and "Зв'язати термін" (Associate Term). Below the "Зв'язати термін" button is a list box containing the following items: "Психологія", "Наука", "Психічне або душевне", and "Психічні процеси".

Рис 1.2 – Вікно додавання нового терміну

Вікно додавання терміну є інтуїтивно зрозумілим та простим у використанні. Щоб додати новий термін, вам потрібно виконати наступні дії:

1. Натисніть на кнопку "Додати термін". З'явиться нове вікно для додавання терміну.
2. Введіть назву терміну у відповідне текстове поле "Назва терміну". Виберіть назву, яка ясно та коротко описує сам термін.
3. У полі "Визначення" введіть визначення терміну. Намагайтеся бути якомога більш точним та зрозумілим.
4. Для додавання термінів, пов'язаних з цим терміном, виберіть потрібний термін із спадного списку і натисніть на кнопку "Додати зв'язок".
5. Після того, як всі поля заповнено, натисніть на кнопку "Зберегти". Ваш термін тепер збережено і доступний для перегляду.
6. Якщо ви хочете скасувати додавання терміну, просто натисніть на кнопку "Назад".

3.2.3 Вікно перегляду визначення та редагування терміну

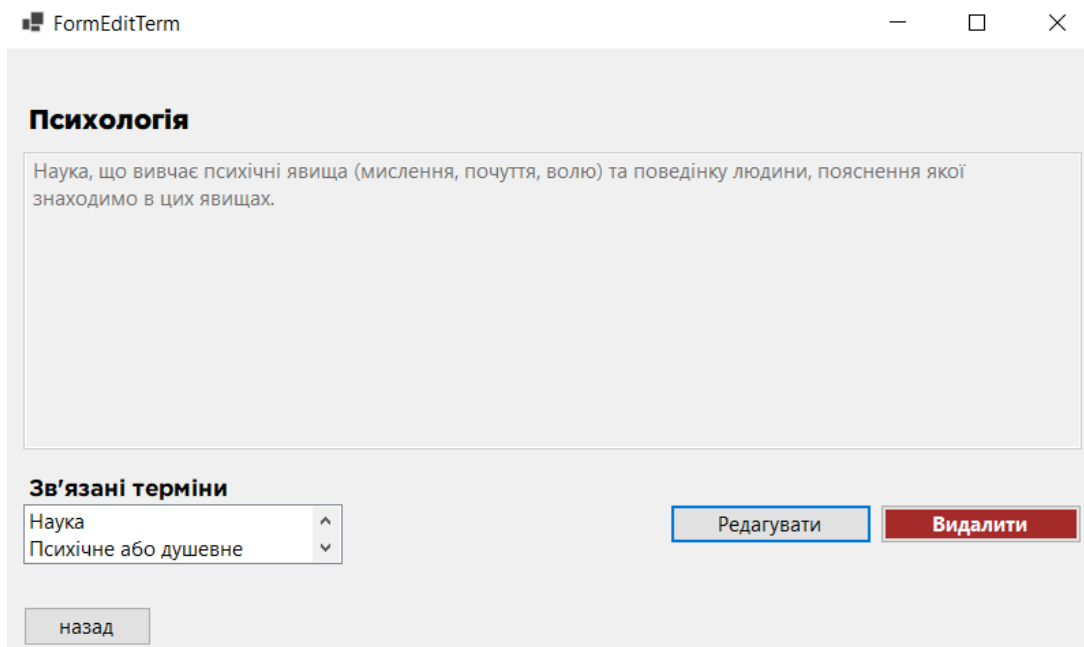


Рис 1.3 – Вікно перегляду визначення та редагування терміну

Вікно редагування терміну також використовується для перегляду визначення терміну. Для його використання виконуйте наступні дії:

1. Виберіть термін, який ви хочете переглянути або редагувати, натиснувши на нього у головному вікні програми.
2. Після вибору терміну з'явиться нове вікно, де ви можете переглянути деталі цього терміну. Ви побачите "Назва терміну" поле "Визначення", де ви можете переглянути визначення терміну, та список "Пов'язані терміни", де ви можете переглянути інші терміни, пов'язані з цим терміном.
3. Якщо ви хочете редагувати термін, то натисніть на кнопку «Редагувати» і вам буде доступне редагування терміну у полі «Визначення». Після редагування натисніть на кнопку «Зберегти», яка з'явилася замість кнопки «Редагувати»".
4. Якщо ви не хочете редагувати термін або хочете скасувати внесені зміни, натисніть на кнопку "Назад".
5. Якщо ви хочете видалити термін, натисніть на червону кнопку «Видалити». Вам відкриється вікно підтвердження, натиснувши так ви видалите термін.

Висновки

У ході виконання курсової роботи була створена програма, що служить як база даних для зберігання термінів та їх визначень. Програма була розроблена за допомогою мови програмування C# і використовує середовище розробки програмного забезпечення Visual Studio.

Основні функції програми включають додавання нових термінів з визначеннями, редагування існуючих термінів, пошук термінів, встановлення зв'язків між термінами та видалення термінів. Кожен термін може мати декілька пов'язаних термінів, що дозволяє створювати складні взаємозв'язки.

Програма використовує JSON для зберігання та відтворення даних, що дозволяє легко зберігати і обмінюватися даними.

Незважаючи на те, що програма вже має багато корисних функцій, є потенційний простір для подальшого вдосконалення. Наприклад, можна додати можливість імпорту та експорту термінів у інші формати, такі як CSV або XML, або розширити пошукові можливості для пошуку термінів не тільки за назвою, але і за визначенням.

У загальному підсумку, ця курсова робота дала можливість поглиблено вивчити та практично застосувати навички програмування на C#, розуміння основ об'єктно-орієнтованого програмування та використання інструментів середовища Visual Studio.

Перелік джерел посилання

1. Бондарев В. М. Основи програмування на JavaScript : навчальний посібник / В. М. Бондарев. – Харків : Коряк С. Ф., 2015. – 182 с.
2. C# / BestProg. Програмування: теорія та практика. URL: https://www.bestprog.net/uk/sitemap_ua/c-3/ (дата звернення: 25.05.2022)
3. ДСТУ 3008:2015 Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. К.: ДП «УкрНДНЦ», 2016. 26 с.
4. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. К.: ДП «УкрНДНЦ», 2016. – 16 с.
5. Документація платформи ASP .NET Core. URL: <https://learn.microsoft.com/en-us/dotnet/> (дата звернення 25.05.2022).
6. Документація мови програмування JavaScript. URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript> (дата звернення 25.05.2022).
7. Документація бібліотеки React. URL: <https://ru.reactjs.org/> (дата звернення 25.05.2022).

Додаток А

MainForm

```
namespace Kursovaya
{
    using System;
    using System.Windows.Forms;
    using System.IO;
    using Newtonsoft.Json;
    using System.Collections.Generic;
    public partial class MainForm : Form
    {
        List<Term> terms;
        public MainForm()
        {
            InitializeComponent();
            LoadData();
        }
        private void LoadData()
        {
            string json = File.ReadAllText("terms.json");
            terms = JsonConvert.DeserializeObject<List<Term>>(json);
        }

        private void Form1_Load(object sender, EventArgs e)
        {
        }

        private void button1_Click(object sender, EventArgs e)
        {
            string query = textBox1.Text;
            if (string.IsNullOrEmpty(query))
            {
                MessageBox.Show("Будь ласка, введіть назву терміну", "Помилка", MessageBoxButtons.OK,
                MessageBoxIcon.Error);
                return;
            }
            var foundTerms = terms.Where(t =>
            t.Name.ToLowerInvariant().Contains(query.ToLowerInvariant())).ToList();

            if (foundTerms.Count == 0)
            {
                MessageBox.Show("Термін не знайдено.", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
                return;
            }
            listBox1.Items.Clear();
            foreach (Term term in foundTerms)
            {
                listBox1.Items.Add(term);
            }
        }

        private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
        {
        }
    }
}
```

```

{
    Term selectedTerm = (Term)listBox1.SelectedItem;

    FormEditTerm formEditTerm = new FormEditTerm(selectedTerm, terms);

    this.Hide();

    formEditTerm.Show();
}

private void button2_Click(object sender, EventArgs e)
{
    // Створюємо екземпляр нової форми
    FormAddTerm formAddTerm = new FormAddTerm();

    // Закриваємо поточну форму
    this.Hide();

    // Показуємо нову форму
    formAddTerm.Show();
}

private void DeleteAll_Click(object sender, EventArgs e)
{
    DialogResult dialogResult = MessageBox.Show("Ви впевнені що хочете видалити всі терміни?",
"Підтвердження видалення", MessageBoxButtons.YesNo);

    if (dialogResult == DialogResult.Yes)
    {
        terms.Clear();
        SaveData();
        listBox1.Items.Clear();
    }
}

private void SaveData()
{
    string json = JsonConvert.SerializeObject(terms, Formatting.Indented);
    File.WriteAllText("terms.json", json);
}
}

```

FormAddTerm

```

using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;

```



```

using System.Threading.Tasks;
using System.Windows.Forms;

namespace Kursovaya
{

    public partial class FormAddTerm : Form
    {
        private Term newTerm;
        private List<Term> terms;
        public FormAddTerm()
        {
            InitializeComponent();
            this.terms = terms;
            LoadData();
        }
        private void LoadData()
        {
            if (File.Exists("terms.json"))
            {
                string json = File.ReadAllText("terms.json");
                terms = JsonConvert.DeserializeObject<List<Term>>(json);
            }
            else
            {
                terms = new List<Term>();
            }
        }

        private void FormAddTerm_Load(object sender, EventArgs e)
        {
            newTerm = new Term { RelatedTerms = new List<string>() };
            foreach (Term term in terms)
            {
                listBoxRelatedTerms.Items.Add(term.Name);
            }
        }

        private void textBoxTermName_TextChanged(object sender, EventArgs e)
        {
        }

        private void textBoxTermDefinition_TextChanged(object sender, EventArgs e)
        {
        }

        private void buttonAddTerm_Click(object sender, EventArgs e)
        {
            if (string.IsNullOrEmpty(textBoxTermName.Text) ||
                string.IsNullOrEmpty(textBoxTermDefinition.Text))
            {
                MessageBox.Show("Будь ласка, введіть назву та визначення терміну", "Помилка",
                    MessageBoxButtons.OK, MessageBoxIcon.Error);
                return;
            }
        }
    }
}

```

```

        if (terms.Any(t => t.Name == textBoxTermName.Text))
        {
            MessageBox.Show("Такий термін вже існує", "Помилка", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
            return;
        }
        // Створюємо новий термін
        newTerm.Name = textBoxTermName.Text;
        newTerm.Definition = textBoxTermDefinition.Text;
        if (terms == null)
        {
            terms = new List<Term>();
        }
        terms.Add(newTerm);
        foreach (var relatedTermName in newTerm.RelatedTerms)
        {
            var relatedTerm = terms.Find(t => t.Name == relatedTermName);
            if (relatedTerm != null && !relatedTerm.RelatedTerms.Contains(newTerm.Name))
            {
                relatedTerm.RelatedTerms.Add(newTerm.Name);
            }
        }

        textBoxTermName.Clear();
        textBoxTermDefinition.Clear();

        SaveData();
        //ініціалізуємо новий термін
        newTerm = new Term { RelatedTerms = new List<string>() };
    }
    private void SaveData()
    {
        string json = JsonConvert.SerializeObject(terms, Formatting.Indented);
        File.WriteAllText("terms.json", json);
    }

    private void button1_Click(object sender, EventArgs e)
    {
        MainForm form1 = new MainForm();
        this.Hide();
        form1.Show();
    }

    private void label1_Click(object sender, EventArgs e)
    {
    }

    private void label2_Click(object sender, EventArgs e)
    {
    }

    private void buttonAddRelation_Click(object sender, EventArgs e)
    {
        string selectedTerm = listBoxRelatedTerms.SelectedItem.ToString();

```

```

        if (!newTerm.RelatedTerms.Contains(selectedTerm))
        {
            newTerm.RelatedTerms.Add(selectedTerm);
        }
        else
        {
            MessageBox.Show("Термін уже зв'язан.", "Помилка", MessageBoxButtons.OK,
                MessageBoxIcon.Warning);
        }
    }
}
}

```

FormEditTerm

```

using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Kursovaya
{
    public partial class FormEditTerm : Form
    {
        public Term currentTerm;
        private List<Term> terms;
        public FormEditTerm(Term selectedTerm, List<Term> termsList)
        {
            InitializeComponent();
            this.currentTerm = selectedTerm;
            this.terms = termsList;
            labelTermName.Text = currentTerm.Name;
            textBoxTermDefinition.Text = currentTerm.Definition;
        }

        private void FormEditTerm_Load(object sender, EventArgs e)
        {
            labelTermName.Text = currentTerm.Name;
            textBoxTermDefinition.Text = currentTerm.Definition;
            listBoxRelatedTerms.Items.Clear();
            foreach (var relatedTerm in currentTerm.RelatedTerms)
            {
                listBoxRelatedTerms.Items.Add(relatedTerm);
            }
        }

        private void labelTermName_Click(object sender, EventArgs e)

```

```

{
}

private void textBoxTermDefinition_TextChanged(object sender, EventArgs e)
{
}

private void back_Click(object sender, EventArgs e)
{
    MainForm form1 = new MainForm();

    this.Hide();

    form1.Show();
}

private void Edit_Click(object sender, EventArgs e)
{
    textBoxTermDefinition.Enabled = true;
    textBoxTermDefinition.Focus();
    Edit.Visible = false;
    saveTermin.Visible = true;
}

private void SaveData()
{
    string json = JsonConvert.SerializeObject(terms, Formatting.Indented);
    File.WriteAllText("terms.json", json);
}

private void saveTermin_Click_1(object sender, EventArgs e)
{
    if (string.IsNullOrWhiteSpace(textBoxTermDefinition.Text))
    {
        MessageBox.Show("Будь ласка, введіть назву та визначення терміну", "Помилка",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
    string name = labelTermName.Text;
    string definition = textBoxTermDefinition.Text;

    Term termToEdit = terms.Find(t => t.Name == name);
    if (termToEdit != null)
    {
        termToEdit.Definition = definition;
        saveTermin.Visible = false;
        Edit.Visible = true;
        textBoxTermDefinition.Enabled = false;
    }
    else
    {

```

```

        Term newTerm = new Term { Name = name, Definition = definition, RelatedTerms = new List<string>() };
        if (terms == null)
        {
            terms = new List<Term>();
        }

        terms.Add(newTerm);
    }

    SaveData();
}

private void deleteTerm_Click(object sender, EventArgs e)
{
    // Удаляем термин из всех связанных терминов
    foreach (var relatedTermName in currentTerm.RelatedTerms)
    {
        var relatedTerm = terms.Find(t => t.Name == relatedTermName);
        if (relatedTerm != null)
        {
            relatedTerm.RelatedTerms.Remove(currentTerm.Name);
        }
    }
    terms.Remove(currentTerm);

    SaveData();

    MainForm form1 = new MainForm();
    this.Hide();
    form1.Show();
}

private void listBoxRelatedTerms_SelectedIndexChanged(object sender, EventArgs e)
{
    string selectedTermName = listBoxRelatedTerms.SelectedItem.ToString();

    Term selectedTerm = terms.Find(t => t.Name == selectedTermName);
    if (selectedTerm != null)
    {
        FormEditTerm formEditTerm = new FormEditTerm(selectedTerm, terms);
        this.Hide();
        formEditTerm.Show();
    }
}
}

Term.cs

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

namespace Kursovaya
{
    public class Term
    {
        public string Name { get; set; }
        public string Definition { get; set; }
        public List<string> RelatedTerms { get; set; }

        public Term()
        {
            RelatedTerms = new List<string>();
        }
        public override string ToString()
        {
            return Name;
        }
    }
}

```

Programm.cs

```

namespace Kursovaya
{
    internal static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            // To customize application configuration such as set high DPI settings or default font,
            // see https://aka.ms/applicationconfiguration.
            ApplicationConfiguration.Initialize();
            Application.Run(new MainForm());
        }
    }
}

```