

«RAKE» особливо корисний у випадках, коли необхідно зберегти ключові словосполучення для подальшого пошуку файлів, де ці словосполучення зустрічаються. Його застосування включає можливість зменшити текст, для подальшої індексації, що полегшує і прискорює процес пошуку файлу.

Список використаних джерел

1. Що таке Azure index [Електронний ресурс] – Режим доступу: <https://learn.microsoft.com/ru-ru/azure/search/search-what-is-an-index>
2. Інші алгоритми [Електронний ресурс] – Режим доступу: <https://medium.com/international-school-of-ai-data-science/the-keyword-quest-exploring-automatic-keyword-extractors-db553c6ac229>
3. Що таке RAKE [Електронний ресурс] – Режим доступу: <https://ovchinnikov.cc/writing/rake/>
4. Реалізація програми [Електронний ресурс] – Режим доступу: <https://github.com/Haker53/azure-search-dotnet.git>

УДК 004.42.4

СТВОРЕННЯ ДОДАТКІВ НА WINDOWS FORMS: АРХІТЕКТУРА ТА ФУНКЦІОНАЛЬНІ МОЖЛИВОСТІ

Клячко М.М., Лебединський А.В.

Харківський національний автомобільно-дорожній університет, Харків

У сучасному світі, де Windows залишається однією з найпоширеніших операційних систем, розробка додатків для неї не втрачає своєї актуальності. Існує кілька різних платформ, які розробники можуть використовувати для створення програм для Windows. С# є відмінним вибором для створення програм для Windows завдяки тісній інтеграції з платформою .NET та інструментами розробки Microsoft. Ось деякі ключові аспекти використання С# для розробки програм під Windows:

- інтеграція з .NET Framework/.NET Core: С# щільно інтегрований з платформою .NET, що забезпечує широкі можливості розробки різноманітних додатків під Windows. Залежно від вимог проекту, ви можете вибрати .NET Framework [1] для традиційних настільних додатків або .NET Core/.NET 5+ для більш сучасних і кросплатформових додатків;

- Windows Presentation Foundation (WPF): WPF - це потужний фреймворк для створення настільних програм під Windows із використанням С#. WPF надає широкі можливості для створення інтерактивних інтерфейсів користувача за допомогою XAML (eXtensible Application Markup Language) і підтримує різні функції, такі як стилі, шаблони, анімації та багато іншого;

- Windows Forms (WinForms): Для створення класичних настільних програм під Windows також можна використовувати Windows Forms. Це більш

старий, але все ще широко використовуваний фреймворк, який надає простий і швидкий спосіб розробки програм із використанням C#;

- зручна інтеграція з Visual Studio: Visual Studio – основне середовище розробки для C#, і вона має багатий набір інструментів для створення програм під Windows. Вона пропонує широкі можливості для налагодження, дизайну інтерфейсу, створення звітів та багато іншого;

- доступ до бібліотек Windows API: C# дозволяє легко взаємодіяти з різними компонентами операційної системи Windows, такими як реєстр, файлова система, служби Windows та багато іншого завдяки доступу до багатого набору бібліотек Windows API [2];

- обробка подій та асинхронне програмування: C# має зручні засоби для обробки подій та асинхронного програмування, що робить розробку інтерактивних і чуйних додатків для Windows більш простою та ефективною.

Плюси та мінуси використання. Використання Windows Forms у C# забезпечує простоту та зручність у розробці настільних програм для Windows.

Плюси:

- простота: Створення інтерфейсу користувача за допомогою WinForms відбувається швидко і легко завдяки візуальному редактору і простоті використання компонентів;

- швидка розробка: Розробка програм на Windows Forms займає менше часу, ніж при використанні деяких інших технологій, завдяки зручному інструментарію та простоті мови програмування C#;

- інтеграція з .NET: WinForms без проблем інтегрується з іншими технологіями та бібліотеками до .NET Framework, що забезпечує доступ до широкого спектру можливостей для розробки;

- широке поширення: Використання WinForms дозволяє створювати програми, які можуть бути запущені на широкому діапазоні пристроїв під керуванням операційної системи Windows [3, 4].

Мінуси:

- обмежені можливості інтерфейсу: WinForms пропонує базові засоби для створення інтерфейсу, що може обмежувати можливості в галузі візуалізації та досвіду користувача, особливо в порівнянні з більш сучасними технологіями, такими як WPF.

- не підходить для мобільних пристроїв: Програми WinForms призначені в основному для настільних комп'ютерів під керуванням Windows і не можуть бути легко адаптовані для мобільних пристроїв або інших платформ.

- старіння: З появою нових технологій, таких як WPF та UWP, WinForms може розглядатися як застаріла технологія, хоча й залишається затребуваною для розробки певного типу програм.

Архітектура Windows Forms. **Форми (Forms):** Основний елемент інтерфейсу в WinForms. Форми представляють вікна додатку, на яких можуть розміщуватися інші елементи інтерфейсу, такі як кнопки, тексти, таблиці тощо. **Контролі (Controls):** Елементи інтерфейсу, що розміщуються на формах, такі як кнопки, тексти, полоси прокрутки, таблиці і т. д. Контролі можуть взаємодіяти з користувачем і виконувати різні функції. **Події (Events):** WinForms викорис-

товує модель подій для обробки взаємодії користувача з додатком. Контроли генерують події при взаємодії з користувачем (наприклад, натискання кнопки), і програміст може прив'язувати обробники подій для реагування на ці події. Модель розділення (Separation of Concerns): Хороша практика при розробці додатків полягає в розділенні логіки програми на різні шари або компоненти. Наприклад, розділення між логікою додатку і його графічним інтерфейсом.

Функціональні можливості Windows Forms. Графічний інтерфейс користувача (GUI): WinForms надає широкий набір контролів для створення і налаштування графічного інтерфейсу, що відповідає потребам додатку. Можливості макетування (Layout Capabilities): Ви можете використовувати різні менеджери макетування для організації розташування контролів на формі, наприклад, FlowLayoutPanel або TableLayoutPanel. Валідація введених даних (Data Validation): WinForms дозволяє встановлювати правила для валідації даних, що вводяться користувачем в тексти, наприклад, обов'язкові поля або правильний формат даних. Можливості друку (Printing Capabilities): Додатки, розроблені на WinForms, можуть використовувати можливості друку для надання звітів або іншої інформації у друкованій формі. Мультимедійні можливості (Multimedia Capabilities): Ви можете інтегрувати зображення, звуки і відео в додаток за допомогою контролів, таких як PictureBox або Windows Media Player. Мережеві можливості (Networking Capabilities): WinForms дозволяє взаємодіяти з мережевими ресурсами, використовуючи класи .NET для роботи з мережевими протоколами. Можливості доступу до баз даних (Database Access Capabilities): Ви можете підключати ваші додатки до різних типів баз даних, таких як SQL Server, MySQL або SQLite, використовуючи ADO.NET або Entity Framework.

Об'єднавши наявну інформацію, розглянемо застосування Windows Forms практично [5]. Була створена форма (рис. 1), яка включає реєстраційні поля для клієнтів.

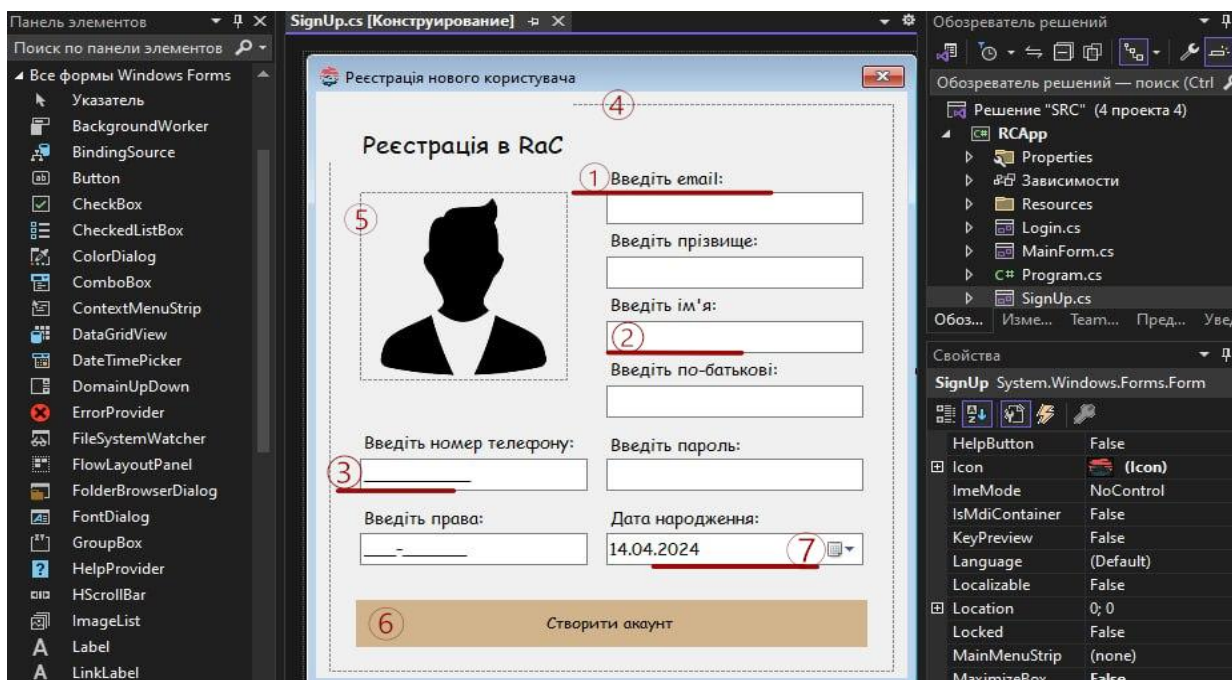


Рисунок 1 – Шаблон вікна реєстрації користувача

На формі були розміщені такі компоненти:

1. Label – компонент для написання тексту;
2. TextBox – компонент для введення тексту;
3. MaskedTextBox – поле (маска) для введення даних специфічного формату;
4. Panel – призначений для компонування об'єктів у вікні;
5. PictureBox – компонент для відображення зображень;
6. Button – кнопка;
7. DateTimePicker - календар, що розкривається за натисканням, у якому можна вибрати дату.

У висновку можемо сказати, що створення програм на Windows Forms представляє цікаву тему, яка охоплює як архітектурні аспекти, так і функціональні можливості цієї платформи.

Архітектура Windows Forms дозволяє розробникам швидко створювати графічні інтерфейси за допомогою компонентів, які легко розташовувати і керувати ними. Функціональні можливості включають широкий спектр можливостей для створення різноманітних додатків, від простих програм з однією формою до складних програм з багатьма вікнами, діалоговими вікнами і вбудованими елементами управління.

Список використаних джерел

1. Ефективний пошук в Інтернеті. *Ділова розвідка в Інтернеті*. URL: <http://isearch.kiev.ua/uk/news/programs/o-appl/1593-create-your-first-windows-forms-application-in-c-for-example-the-game-qtic-tac-toeq> (дата звернення: 10.04.2024).
2. Troelsen A., Japikse P. Pro C# 10 With .NET 6: Foundational Principles and Practices in Programming. Apress L. P., 2022.
3. C# Guide – .NET managed language. *Microsoft Learn: Build skills that open doors in your career*. URL: <https://learn.microsoft.com/uk-ua/dotnet/csharp/> (date of access: 10.04.2024).
4. Середовище програмування сі шарп та його переваги. *FoxmindEd*. URL: <https://foxminded.ua/seredovyshche-prohramuvannia-si-sharp/> (дата звернення: 10.04.2024).
5. C# та Windows Forms. Перший додаток. URL: <https://itproger.com/course/csharp-app> (дата звернення: 10.04.2024).