

Автор:

**Омельчук Ніка Романівна**  
Спеціальність 121 «Інженерія  
програмного забезпечення»  
студентка 32 ІІЗ групи

Науковий керівник:

кандидат педагогічних наук, доцент,  
доцент кафедри комп'ютерної та  
програмної інженерії  
Галицький Олександр Вадимович

## **СТВОРЕННЯ ВЛАСНОЇ МОВИ ПРОГРАМУВАННЯ ТА КОМПІЛЯТОРА ДЛЯ НЕЇ**

**Анотація.** Створення власної мови програмування та компілятора для неї – це цікавий напрямок досліджень. Саме створення власної мови програмування є досить цікавим завданням, яке потребує неабияких вмінь, навичок, а також самого розуміння, як працює мова, як правильно писати код і як його трактувати (розшифрувати).

**Ключові слова:** мова програмування, компілятор, WPF, XAML, C#, розробка, дизайн.

**Вступ.** Мета полягає в розробці мови та інструменту, який перетворює програми на цій мові в машинний код. Це дозволяє нам краще розуміти процес компіляції та покращує навички програмістів.

Програми-компілятори застосовуються у сфері ІТ для роботи з кодом. Даний програмний продукт використовуватимуть для розробки простих консольних програм, які можуть виконувати прості математичні дії, та загалом цікаві функції.

Дана мова програмування актуальна тим, що команди писатимуться українською мовою, тим самим будучи інтуїтивно зрозумілішими українцям.

**Постановка задачі.** Оскільки, завдання роботи полягає у розробці власної мови програмування та створення програми, яка буде компілювати даний код, то варто звернутися до дослідження, щодо проектування мови програмування, які будуть виконуватися базові логічні, математичні функції та їх компілювання. В результаті дослідження маємо отримати розроблену програму-компілятор на основі власної мови програмування.

**Метою дослідження** є розробка власної мови програмування, а також створення програми, яка буде компілювати створений код.

**Основна частина.** Насамперед, новосторена програма та мова програмування повинна складатися з таких функцій, як: зчитування написаного тексту; перетворення його на машинний код; надання користувачам підказок як правильно написати код; пошук помилок та наголошення на них; збереження написаного коду на пристрої. Обчислювальні функції програмного продукту повинні бути реалізовані мовою програмування C# у середовищі Visual Studio за допомогою технології WPF. Експлуатаційне призначення програмного продукту – допомога в розробці консольної програми, та забезпечення того, щоб вона працювала правильно та безпечно. Щоб розробити прототип дипломного проекту використовувалась графічна підсистема WPF. WPF (Windows Presentation Foundation) - технологія від Microsoft для роботи з GUI, для платформи .NET framework. Щоб мати змогу нею користуватись потрібно спочатку дослідити та вивчити її основні функції, та дізнатись як створювати дизайн програми за допомогою вбудованої туди мови програмування XAML. Сам код програми реалізовувати на мові програмування C#.

Розробка дизайну компілятора для власної мови програмування включає в себе створення логотипу та вікон з елементами всередині. Назва мови програмування – UaNiKi (скорочено UN).

Проект складається з 4 вікон, в яких знаходяться такі елементи як: кнопки (Button), картинки (Image), текст (TextBlock), звичайні та форматовані текстові поля (RichTextBox, TextBox) та меню. У першому вікні – меню можна побачити такі кнопки як: «Відкрити

локальну папку», яка буде відкривати діалогове вікно, в якому згодом можна буде відкрити готовий проект та «Створити проект», яка буде відкривати наступне вікно. Також на формі є картинка з логотипом мови програмування, тексти-підказки «Корисні файли» та «Початок роботи», та текстові кнопки, при натисканні яких будуть відкриватись посібники з мови програмування та документація до самої програми.

У другому вікні можна побачити такі кнопки як: «Назад», яка переносить нас на початкову форму; «Створити», яка буде створювати проект за введеними даними у текстові поля (ім'я проекту та розташування), та відкриває головне вікно проекту; та «...», яка відкриває діалогове вікно браузера папок, в якому можна обрати шлях розташування проекту. На формі також є тексти-підказки «Налаштувати новий проект», та текст над текстовими полями, що насправді являють собою властивість TextHint.

У вікні створеного проекту можна побачити 2 форматованих текстових поля, перший з яких слугує для написання туди коду (біля нього ще розташоване текстове поле з нумерацією рядків), а у другий програма записуватиме помилки, чи загалом стан компіляції програми. Також у вікні є меню, в якому є такі поля як відкриття, збереження, та компіляція проектів. Останнє ж вікно – консоль, відкриватиметься при компіляції коду.

Для того, щоб розробити функціонал програми потрібно створити взаємодію між формами та бібліотеку класів, в якій буде знаходитись 2 класи UaNiKi та Redacture, перший з яких це клас де будуть всі функції мови програмування, а друга буде відслідковувати правильність написання коду. Також є клас який не буде входити у бібліотеку, а буде знаходитись в проекті безпосередньо та відповідати за зовнішній вигляд коду.

У клас UaNiKi додаються методи з різноманітними логічними та математичними функціями. Наприклад найпростішою функцією є команда «написати», яка буде записувати в консоль заданий текст. Таким чином поступово додаються й інші функції зза допомогою яких виконуватимуться команди при компіляції коду.

**Висновок.** За результатами виконаної роботи було розроблено власну мову програмування та компілятор для неї. Цей проект вимагав розуміння теорії мов програмування та компіляції, а також навичок в розробці програмного забезпечення. Власна мова може використовуватися для написання програм та розв'язання різноманітних завдань. Компілятор перетворює вихідний код на мові програмування в машинний код або інший виконуваний формат, що дозволяє виконувати програми на цій мові. Дана мова актуальна, оскільки команди писатимуться українською мовою, що робить їх інтуїтивно зрозумілими для українців. Програма-компілятор буде використовуватись для розробки консольних програм та забезпечення їх правильної та безпечної роботи.

### Список використаних джерел

1. 'Enter' - WPF .NET Framework. Microsoft Learn: Build skills that open doors in your career. URL: <https://learn.microsoft.com/ru-ru/dotnet/desktop/wpf/advanced/how-to-detect-when-the-enter-key-pressed?view=netframeworkdesktop-4.8> (дата звернення: 29.03.2024).
2. Lapinskyi V.V., Mykytenko P.V., Halytskyi O.V. (2021). Design of medical information systems user interface. *Information Technologies and Learning Tools*, 85(5), 1-13. <https://doi.org/10.33407/itlt.v85i5.44072>
3. Windows Presentation Foundation - WPF .NET. Microsoft Learn: Build skills that open doors in your career. URL: <https://learn.microsoft.com/ru-ru/dotnet/desktop/wpf/overview/?view=netdesktop-8>. (дата звернення: 28.03.2024).
4. WPF .NET Framework. Microsoft Learn: Build skills that open doors in your career. URL: <https://learn.microsoft.com/ru-ru/dotnet/desktop/wpf/controls/change-selection-in-a-richtextbox-programmatically?view=netframeworkdesktop-4.8> (дата звернення: 29.03.2024).

5. WPF в порівнянні з WinForms - The complete WPF tutorial. Welcome - The complete WPF tutorial. URL: [https://wpf-tutorial.com/uk/2/про-wpf/wpf-в-порівнянні-з-winforms/#google\\_vignette](https://wpf-tutorial.com/uk/2/про-wpf/wpf-в-порівнянні-з-winforms/#google_vignette) (дата звернення: 30.03.2024).
6. Що таке WPF? - The complete WPF tutorial. Welcome - The complete WPF tutorial. URL: <https://wpf-tutorial.com/uk/1/про-wpf/що-таке-wpf/> (дата звернення: 30.03.2024).