

Особливості використання візуальних компонентів для введенням та виведення даних

Мешко Ю.І.

студентка

ПНПУ імені В.Г. Короленка

meshkojulia2002@gmail.com

Візуальна складова кожної програми має велике значення як для програміста, так і для користувача. Одне з важливих правил при створенні тієї чи іншої програми є те, що інтерфейс повинен бути зрозумілим і зручним у використанні. Більшість сучасних програм для повноцінного функціонування потребують «діалогу» з користувачем, тобто обміну інформацією з введенням різних даних у спеціально відведені для цього поля. Прикладом такого поля у середовищі Visual C++ є компонент TextBox.

Компонент TextBox використовується для отримання вхідних даних від користувача або для виведення тексту. За замовчуванням в текстове поле можна ввести до 2048 символів, але якщо для властивості компоненту Multiline задати значення True, то це відкриє доступ до багаторядкового введення даних (до 32 КБ тексту) [2].

Усі дані, які вводяться у поле компоненту TextBox, сприймаються програмою як рядкові дані. Щоб перевести їх у числовий формат використовують спеціальні функції (методи): дійсний тип використовується метод ToDouble; цілочисловий тип – ToInt32 [1].

Перехід до технології візуального програмування вимагав створення нової бібліотеки класів. Головною особливістю даної бібліотеки полягає в тому, що в її основі лежить концепція властивостей, методів та подій ToInt32 [1].

Властивості є атрибутами компоненту, що визначають його зовнішній вигляд і поведінку. До основних властивості компоненту TextBox відносять: Text (повідомлення у полі), Location (положення компоненту на формі), MaxLength (максимальна допустима кількість символів), Multiline (багаторядковість), Lines (масив рядків), ScrollBars (смуги прокрутки) та ін. Також однією з властивостей цього компоненту є ReadOnly, яка при значенні True забороняє можливість введення даних у поле та внесення будь-яких змін користувачем, а робить текстове поле доступним лише для читання та інші.

На відміну від властивостей, які представляють собою змінні чи об'єкти, методи представляють собою функції класу (об'єкту), які можна викликати ззовні для управління об'єктом. Методи компоненту TextBox: AppendText() (додавання тексту до поточного), Clear() (очищення поля),

Copy() (копіювання рядка), Focus() (встановлення фокусу), Hide() (приховування компоненту), Show() (робить компонент видимим) та ін.

Зазвичай основна робота з цим компонентом здійснюється подією TextChanged, яка виникає при зміні вмісту текстового поля користувачем. При зміні тексту програмним способом дана подія не виникає.

Для кращого розуміння роботи з компонентом розглянемо приклад програми, яка виводить у текстове поле значення кубів цілих чисел в діапазоні від 1 до N з кроком 2 (рис.1):

```
private: System::Void button1_Click(...)
{
    int n, s,i;
    //перетворення змінної до цілочислового типу
    n=Convert::ToInt32(textBox2->Text);
    for(i=1; i<=n; i+=2)
    {
        s=Math::Pow(i,3);
        //виведення на екран результату роботи програми
        textBox1->AppendText(i+"^3="+s+"\r\n");
    }
}
```

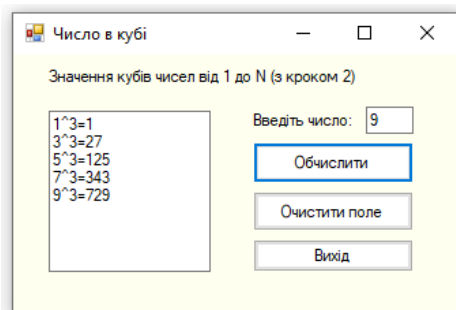


Рис. 1. Зовнішній вид форми розробленого додатку

Проаналізувавши можливості середовища програмування Visual C++ можемо відзначити, що візуальний компонент TextBox простий у використанні і може забезпечувати зручне виведення даних та введення повідомлень і результатів роботи програми..

Список використаних джерел

1. Кривцова О.П. Програмування мовою C++. Технологія візуального програмування : навч. посіб. Полтава: ПНПУ імені В.Г. Короленка, 2020. 144 с.

2. Общие сведения об элементе управления TextBox (Windows Forms). URL:<https://docs.microsoft.com/ru-ru/dotnet/desktop/winforms/controls/textbox-control-overview-windows-forms?view=netframeworkdesktop-4.8>.