Створення компонент користувача Windows Forms

# Особливості компонента

Компоненти бібліотеки Windows Forms суттєво спрощують написання десктопних застосунків. Ви бачите зовнішній вигляд майбутньої програми вже на етапі проектування. Налаштування властивостей візуального компонента відразу відображається на екрані. Саме тому в компонента є три інтерфейси: один - для взаємодії з середовищем програмування, другий - з програмістом на етапі проектування, третій - з програмним кодом застосунку на етапі виконання. Написання власного компонента - справжній приклад об’єктно-орієнтованого програмування, оскільки доводиться використовувати функціонал базових компонент, перевизначати його, проектувати поля, методи, властивості і події, передбачати дії користувачів та ще багато всього.

## Що таке “компонент”

* Складова частина чогось більшого
* Незалежний модуль програмного забезпечення, придатний для повторного використання через строго визначений інтерфейс
* Двійковий файл, чий інтерфейс визначено властивостями, методами та подіями
* Екземпляр класу з ієрархії Component
* Об’єкт, частина програмного забезпечення, з яким можна взаємодіяти на етапі проектування засобами середовища програмування

## Навіщо будувати нові компоненти

* Надати розробникові додаткову функціональність
  + удосконалити наявний компонент (наприклад, кнопка могла звучати при клацанні, напис міг би мати першу літеру особливого вигляду)
  + зробити схожий до наявного (круглу кнопку)
  + створити цілком новий (перемикач-слайдер, групу залежних перемикачів, підписане поле введення)
* Підтримати повторне використання готового програмного коду
  + компонентні класи “завжди під рукою”, використання готових компонент у рази підвищує швидкість
* Збільшити продуктивність праці
  + власний компонент виконує частину налаштувань (наприклад, встановлює потрібні розміри, шрифти, кольори: “велика червона кнопка з золотим написом”)
* Досягти узгодженості
  + однакові компоненти – однакові способи програмування, однакові інтерфейси готових програм

## Особливості побудови компонент

* Користувачами компонент є програмісти
* Створення компонент є невізуальним процесом
* Створення компонент використовує всі засоби ООП
* Компоненти створюють з дотриманням багатьох домовленостей
* Компоненти мають бути гнучкими
* Компоненти мають три різних інтерфейси: *часу виконання*, *часу проектування* та *розробника*

## Етапи побудови компонента

* Створення проекту для компонента і тестової програми
* Створення каркасу компонента
  + вибір базового класу і/або складових компонент
  + візуальне компонування видимих частин, налаштування
* Програмування поведінки
* Тестування інтерфейсу часу виконання
  + написати тестову програму, що створює компонент на етапі виконання і випробовує всі його можливості
* Встановлення компонента в палітру
* Тестування інтерфейсу часу проектування.

# Навчальні матеріали

Підручник [Visual C# 2008. Базовий курс] містить цілком доступний приклад, з якого можна почати. Відповідний параграф цього підручника додано до [папки лекції](https://drive.google.com/drive/folders/1MoDc5i0YiBUu7c9QR987_NfHdIZehx0d?usp=sharing) як окремий документ.

Є ціла книга [C# Розробка компонент в MS Visual Studio], присвячена проектуванню компонент користувача. Її додано до папки [Книги C#](https://drive.google.com/drive/folders/1xzGQ4md83Mhu3WsBC8_epoKkDP5PRylX?usp=sharing). Це для тих, хто хотів би розібратися в питанні глибоко.

Книжки 2008/09 років, але інформація стосовно Windows Forms актуальна.

## Рядок введення з назвою

Створіть компоненту *LabeledTextbox*, як описано в підручнику, використайте її для того, щоб модифікувати проект TrafficLights в частині введення тривалостей світіння ламп світлофора.

## Група залежних перемикачів

Компонента Radiobutton – “істота” соціальна, одна не ходить. Вона “вміє” знаходити поруч собі подібних і діяти злагоджено з усіма іншими: клацання на кнопці змінює і її стан, і увімкненого сусіда. Таке вміння починає шкодити, коли ми намагаємося в одному вікні розташувати дві групи перемикачів. Доводиться їх якось розділяти, розміщувати в окремих контейнерах. У цьогорічних студентських програмах уже помічені проблеми такого роду :) .

Якщо у вікні треба розташувати декілька груп перемикачів, то робота стає схожою на каторгу: купа одноманітних дій з налаштування, розпізнавання тощо. І текст програми не назвеш елегантним. Після такого навіть одну групу кнопок хочеться створювати так само легко, як спадний список чи меню.

Компоненти користувача для того і придумані, щоб складну роботу втілити в життя один раз і потім використовувати, скільки доведеться. Проблему з групою залежних перемикачів вирішує “авторська” компонента *RadioGroup*. З’ясувалося, що набагато легше уявити собі проект такого компоненту, ніж реалізувати. Він на порядок складніший, ніж *LabeledTextbox* з підручника. Спробуйте розібратися.

Розробка описана в двох статтях. Одну знайдете на [сайті рідного Вісника](http://visnyk-ami.lnu.edu.ua/db/313/15.pdf), або в папці цієї лекції. Інша – на [CodeProject](https://www.codeproject.com/Articles/1204564/Construction-and-Design-Time-Support-of-the-RadioG). Візьміть за звичку відвідувати цей сайт. Тут можна знайти дуже багато цікавого. (І код розробленої компоненти :)

Завантажте собі компоненту і використайте її для переробки проекту SimpleTester. У ньому є дві групи залежних перемикачів, створені вручну. Замініть їх екземплярами

*RadioGroup* і порівняйте, як програмувати зручніше.

## Проекти компонент

Розробіть і опишіть свій власний компонент

* Напис, що демонструє поточний час, оновлюється щосекунди, чи що п’ять секунд, чи щохвилини.
* Кнопку, що звучить при клацанні, звук можна налаштовувати
* Велику червону кнопку з золотим написом, напис налаштовують
* Підписаний ProgressBar, що змінює написи залежно від ступеня виконання
* Власна пропозиція.