

## Лабораторна робота 3. Створення елементів управління

**Вправа 1. Створення складеного елемента управління (3 бали)**

**Вправа 2. Створення спеціалізованого елемента управління (3 бали)**

**Вправа 3. Створення розширених елементів управління (3 бали)**

**Мета роботи:** Вивчення способів розробки елементів управління та отримання навичок з їх налаштування і застосування в подальшій роботі.

### **Вправа 1. Створення складеного елемента управління (3 бали)**

На додаток до вже існуючих елементів управління можна розробляти власні, щоб забезпечити для своїх застосувань спеціалізовану функціональність.

Існує три види елементів управління, розроблюваних користувачем:

- **складені** (composite) - створені при об'єднанні інших елементів управління Windows Forms;
- **спеціалізовані** (custom) - створені з нуля і надають власний код для промальовування;
- **розширені** (extended) - додають функціональність до вже існуючого елемента управління Windows Forms.

Складені елементи управління спадкуються від класу **UserControl**. Він надає базовий рівень функціональності, який забезпечує додавання інших елементів управління, а також властивостей, методів і подій. Клас **UserControl** має власний конструктор, який дозволяє використовувати у Visual Studio IDE перетягання додаткових елементів управління з **Toolbox** на поверхню конструктора і налаштувати їх.

У цій вправі ви створите складений елемент управління, який діє як цифровий годинник. У нього ви додасте елемент управління **Label**, який відображає правильний час, і компонент **Timer**, який кожену секунду оновлює **Label**. Надавши властивість **Enabled** елемента управління **Timer**, ви дасте користувачам можливість включати і виключати годинник.

### **Розробка складеного елемента управління**

1. Створіть у Visual Studio нове застосування Windows Forms. Назвіть його **WinTimer1**.
2. У полі **Text** створеної форми **Form1** введіть **Timer**.
3. Для властивості **Font** форми встановимо наступні параметри: **Размер – 26, Начертание – полужирный, Шрифт – Segoe UI**.
4. Властивості **Size** елемента **Form1** встановимо значення **260; 140**.
5. У меню **Project** виберіть **Add User Control** і клацніть **Add** в діалоговому вікні **Add New Item**. Вкажіть ім'я **UserControlTimer** і натисніть **Додати**. До вашого проекту буде доданий порожній елемент управління користувача, який відкриється в конструкторі.
6. З **Toolbox** перетягніть **Label** в призначений елемент управління користувача.
7. Властивості **AutoSize** елемента **Label** встановимо значення **False**.
8. Властивості **Size** елемента **Label** встановимо значення **157; 55**.
9. Видаліть дані з властивості **Text** елемента управління **Label**.
10. Змініть розміри елемента управління користувача так, щоб він приблизно дорівнював розміру елемента управління **Label**.
11. З **Toolbox** перетягніть в елемент управління користувача компонент **Timer**.
12. У вікні **Properties** компоненту **Timer** присвойте властивості **Interval** значення **1000** і властивості **Enabled** значення **True**.
13. Двічі клацніть компонент **Timer**, щоб відкрити у вікні коду обробник події **Timer.Tick** за замовчуванням і додайте наступний рядок програми:

```
label1.Text = DateTime.Now.ToLongTimeString();
```

14. У вікні коду додайте наступне оголошення властивості:

```
public bool TimeEnabled
{
    get { return timer1.Enabled; }
    set { timer1.Enabled = value; }
}
```

15. У меню **File** виберіть **Save All**, щоб зберегти ваше рішення.

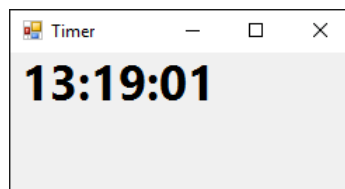
16. У меню **Build** виберіть **Build Solution**.

**Увага:** Зверніть увагу на пункт **15**. Виконуємо тільки побудову проекту і не компілюємо його.

### Застосування складеного елемента управління

17. Виберіть вкладку конструктора **Form1**. Зверніть увагу на **Панель елементів**. Над розділом **Все форми Windows Forms** з'являється розділ **Компоненти WinTimer1** з елементом управління користувача **UserControlTimer**.
18. Перетягнемо створений елемент управління користувача **UserControlTimer** на форму.
19. Зверніть увагу на те, що ви можете призупинити роботу таймера, присвоївши властивості **TimeEnabled** елемента управління користувача **UserControlTimer** значення **False** у вікні **Properties**.
20. Побудуйте і запустіть застосування. Результат роботи зображений на рисунку. Зверніть увагу, що елемент управління користувача під час виконання діє так же, як і в конструкторі.

**Примітка:** Таку послідовність кроків потрібно виконати для версії Visual Studio 2012 Ultimate. Можливо для інших версій вона буде не такою.



### Розробка бібліотеки класів елементів управління

21. Щоб розробити бібліотеку класів елементів управління, виконайте наступні дії:
  - a. Створіть у Visual Studio нове застосування – **Бібліотека елементів управління Windows Forms** і дайте йому ім'я **WinControlLibrary**.
  - b. За замовчуванням проект містить складений елемент управління.
  - c. Додайте елементи управління і код, як описано у **Вправі 1**.
  - d. Виберіть елемент управління, який не повинен змінюватися при спадкуванні класів (у цьому випадку це і **label1**, і **timer1**), і задайте для властивості **Modifiers** (Модифікатори) цього елемента управління значення **Private**.
  - e. Створіть бібліотеку **DLL**, побудувавши проект (**Послідовність дій з оголошення і використання класу. Див. лекцію** )

## Вправа 2. Створення спеціалізованого елемента управління (3 бали)

Для спеціалізованих елементів управління не існує інтерфейсу користувача за замовчуванням, ці елементи повинні створити увесь код, потрібний для відображення свого графічного зображення. Елемент управління розробляється для того, щоб отримати точнісінько таке саме візуальне представлення, яке вам потрібне, і закодувати будь-яку потрібну функціональність для взаємодії з користувачем

Спеціалізовані елементи управління спадкують від класу **Control**. Цей клас забезпечує функціональність, потрібну для елемента управління. Він надає базову функціональність, потрібну для взаємодії елемента управління з іншою частиною застосування.

Ключовим завданням при розробці спеціалізованого елемента управління є реалізація видимого інтерфейсу користувача. Його можна створити реалізацією методу **OnPaint**, який викликається кожного разу, коли елемент управління відображається на екрані.

Виконавши цю вправу, ви навчитеся створювати спеціалізований елемент управління, який також є цифровим годинником. Подібно до елемента управління з попередньої вправи, вони включають компонент **Timer** для оновлення інтерфейсу користувача на регулярній основі. Проте тепер ви самі створите зображення для цього елемента управління замість того, щоб використати елемент управління **Label**.

### Розробка спеціалізованого елемента управління

1. Створіть у Visual Studio нове застосування Windows Forms. Назвіть його **WinTimer2**.
2. У меню **Project** виберіть **Add New Item (Добавить новый Элемент)**. Виберіть **Custom Control (Пользовательский элемент управления)** в діалоговому вікні **Add New Item**. Вкажіть ім'я **UserControlTimer2** і натисніть **Add (Додати)**. До проекту буде доданий новий спеціалізований елемент управління. Відкрийте і перегляньте його код.
3. Перейдіть на вкладку конструктор. У конструкторі **UserControlTimer2** перетягніть компонент **Timer** з **Toolbox** на поверхню конструктора.
4. У вікні **Properties** присвойте властивості **Interval** компонента **Timer** значення **1000** і властивості **Enabled** значення **True**.
5. Двічі клацніть компонент **Timer** щоб відкрити вікно коду обробника події **Timer.Tick** за замовчуванням і додайте наступний рядок програми:

```
this.Refresh();
```

6. Перевизначте метод **OnPaint** і вкажіть код для відображення прямокутника, залитого синім кольором, який заповнює увесь елемент управління:

```
protected override void OnPaint(PaintEventArgs pe)
{
    base.OnPaint(pe);
    Graphics g = pe.Graphics;
    g.FillRectangle(Brushes.Blue, 0, 0, this.Width, this.Height);
}
```

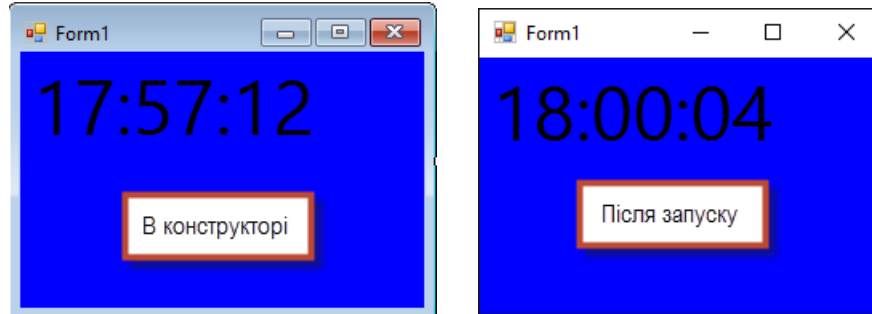
7. Для відображення часу додайте наступний код до методу **OnPaint** нижче визначення прямокутника :

```
pe.Graphics.DrawString(DateTime.Now.ToLongTimeString(),
    this.Font, new SolidBrush(this.ForeColor), 0, 0);
```

8. У меню **File** виберіть **Save All**, щоб зберегти ваше рішення.
9. У меню **Build** виберіть **Build Solution**.

### Застосування спеціалізованого елемента управління

10. Виберіть вкладку конструктора **Form1**. З **Toolbox** перетягніть **UserControlTimer2** у форму. До неї буде доданий екземпляр вашого спеціалізованого елемента управління, який почне відраховувати кожну секунду.
11. Побудуйте і запустіть застосування. Зверніть увагу, що елемент управління користувача діє під час виконання так само, як в конструкторі.



### Вправа 3. Створення розширених елементів управління (3 бали)

**Розширені елементи управління** - це розроблені користувачем елементи управління, які розширюють вже існуючий елемент управління .NET Framework. Зберігається вся функціональність існуючих елементів управління, але при цьому додаються властивості і методи, а при потребі змінюється і зовнішнє представлення елемента управління.

#### Розробка розширеного елемента управління

1. Створіть у Visual Studio нове застосування Windows Forms. Назвіть його **WinButNum**.
2. У меню **Project** виберіть **Add Class**. Назвіть цей клас **ClickButton** і клацніть **Add**.
3. Змініть оголошення класу, щоб **ClickButton** успадковував клас **Button**:

```
public class ClickButton: System.Windows.Forms.Button
```

4. Додайте наступне поле і властивість у вікно коду з метою створення властивості **Clicks**:

```
int mClicks;  
public int Clicks  
{  
    get { return mClicks; }  
}
```

5. Перевизначте метод **OnClick**, щоб інкрементувати закриту змінну **mClicks** кожного разу, коли клацає кнопка:

```
protected override void OnClick(EventArgs e)  
{  
    mClicks++;  
    base.OnClick(e);  
}
```

6. Перевизначте метод **OnPaint**, щоб відобразити число клацань в правому нижньому кутку елемента управління:

```
protected override void OnPaint(System.Windows.Forms.PaintEventArgs pevent)
```

```
{  
    base.OnPaint(pevent);  
    System.Drawing.Graphics g = pevent.Graphics;  
    System.Drawing.SizeF stringsize;  
    stringsize = g.MeasureString(Clicks.ToString(), this.Font, this.Width);  
    g.DrawString(Clicks.ToString(), this.Font,  
        System.Drawing.SystemBrushes.ControlText,  
        this.Width - stringsize.Width - 3, this.Height - stringsize.Height - 3);  
}
```

7. Збережіть і побудуйте рішення.

### Застосування розширеного елемента управління

8. Виберіть вкладку конструктора **Form1**.
9. З **Toolbox** перетягніть екземпляр **ClickButton** у форму і змініть його розміри у бік збільшення.
10. Побудуйте і запустіть застосування.
11. У формі клацайте **ClickButton1**. Зверніть увагу, що число клацань відображається в правому нижньому кутку.

