

**МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ**

**Харківський національний університет внутрішніх справ**

**Кафедра інформаційних технологій факультету № 4**

**ТЕКСТ ЛЕКЦІЇ**

**з навчальної дисципліни «Алгоритмізація та програмування»**

обов'язкових компонент  
освітньої програми першого(бакалаврського) рівня вищої освіти  
125 «Кібербезпека»  
(«Поліцейська діяльність у кіберсфері», «Безпека інформаційних та  
комунікаційних систем»)

**за темою «Візуальні компоненти - кнопки».**

м. Харків  
2019 р.

## **ЗАТВЕРДЖЕНО**

Науково-методичною радою  
Харківського національного  
університету внутрішніх справ  
Протокол від 24.01.19 № 1

## **СХВАЛЕНО**

Вченою радою факультету № 4  
Протокол від 16.01.19 № 1

## **ПОГОДЖЕНО**

Секцією Науково-методичної ради  
ХНУВС з технічних дисциплін

Протокол від 17.01.19 № 1

Розглянуто на засіданні кафедри інформаційних технологій (протокол від 15.01.19 № 1)

**Розробник:** завідувач кафедри інформаційних технологій, канд.техн.наук,  
доцент Струков В.М.

**Рецензент:** доктор технічних наук, професор Зацеркляний М.М.

## **План лекції**

1. Елементи управління.
2. Командні кнопки.
3. Кнопки-перемикачі.

## **Література:**

### **Основна:**

1. Э. Троелсен. Язык программирования C# 5.0 и платформа .NET 4.5. , 6-е изд. : Пер. с англ. — М. : ООО "И.Д. Вильямс", 2013. — 1312 с. : ил. — Парал. тит. англ.
2. Конспект лекцій.

### **Додаткова:**

3. Программирование на языке C#: учеб. пособие / Л.В. Соловей, Н.Н. Мирошниченко, Н.Г. Пономарёв. – Х. : НТУ «ХПИ», 2016. – 356 с.

## **Текст лекції**

### **1. Елементи управління.**

Елементи управління (командні кнопки і кнопки вибору) є візуальними класами, які отримують введені користувачем команди(вказівки) і можуть ініціювати різні події. Всі елементи управління успадковуються від класу Control і тому мають ряд загальних властивостей:

- BackColor: визначає фоновий колір елемента,
- BackgroundImage: визначає фонове зображення елемента,
- ContextMenu: контекстне меню, яке відкривається при натисканні на елемент правою кнопкою миші. Задається за допомогою елемента ContextMenu,
- Cursor: визначає, як буде відображатися курсор миші при наведенні на елемент,
- Dock: задає розташування елемента на формі,
- Enabled: визначає, чи буде доступний елемент для використання. Якщо ця властивість має значення False, то елемент блокується,
- Font: встановлює шрифт тексту для елемента,
- ForeColor: визначає колір шрифту,
- Location: визначає координати верхнього лівого кута елемента управління,
- Name: ім'я елемента керування у програмному коді,
- Size: визначає розмір елемента,
- Width: ширина елемента,
- Height: висота елемента.

## 2. Командні кнопки.

Кнопка є одним з основних елементів діалогу під час виконання застосування. За допомогою кнопок користувач підтверджує або відмінює рішення про виконання тієї або іншої команди або встановлення того чи іншого режиму. Всім користувачам Windows знайомі такі кнопки, наприклад, як *Ok* і *Cancel*. У Windows Forms передбачена можливість працювати з двома типами кнопок:

- командні кнопки;
- кнопки-перемикачі.

Натиснення командної кнопки (клацання по кнопці лівою кнопкою миші) приводить до негайного виконання деякої операції. Типовим прикладом командної кнопки є кнопка *Ok*.

Для реалізації стандартних командних кнопок в *Windows Forms* використовується компонент *button*. На формі такі кнопки виглядають таким чином:



Найбільш частою подією, що трапляється для кнопки *button*, є *OnClick* – клацання лівою кнопкою миші в області кнопки. Обробник події *OnClick* є найпростішим і зручнішим засобом для розміщення в тексті застосування програмної реалізації Вашого завдання. Для того, щоб потрапити в текст програми заготовки обробника події *OnClick* кнопки *button* потрібно двічі клацнути по кнопці на формі. **Обробник події** – це процедура, яка реалізує дію, що виконується при здійсненні даної події. *Windows Forms* автоматично створює шаблон-заготовку кожного обробника події (початковий текст програми відповідної процедури):

```
private void button6_Click(object sender, EventArgs e)
{
}

```

### Оформлення кнопки.

Щоб керувати зовнішнім відображенням кнопки, можна використовувати властивість *FlatStyle*. Вона може набувати таких значень:

- *Flat* - кнопка має плоский вигляд,
- *Raised* - кнопка набуває об'ємний вигляд при наведенні на неї покажчика, в інших випадках вона має плоский вигляд,

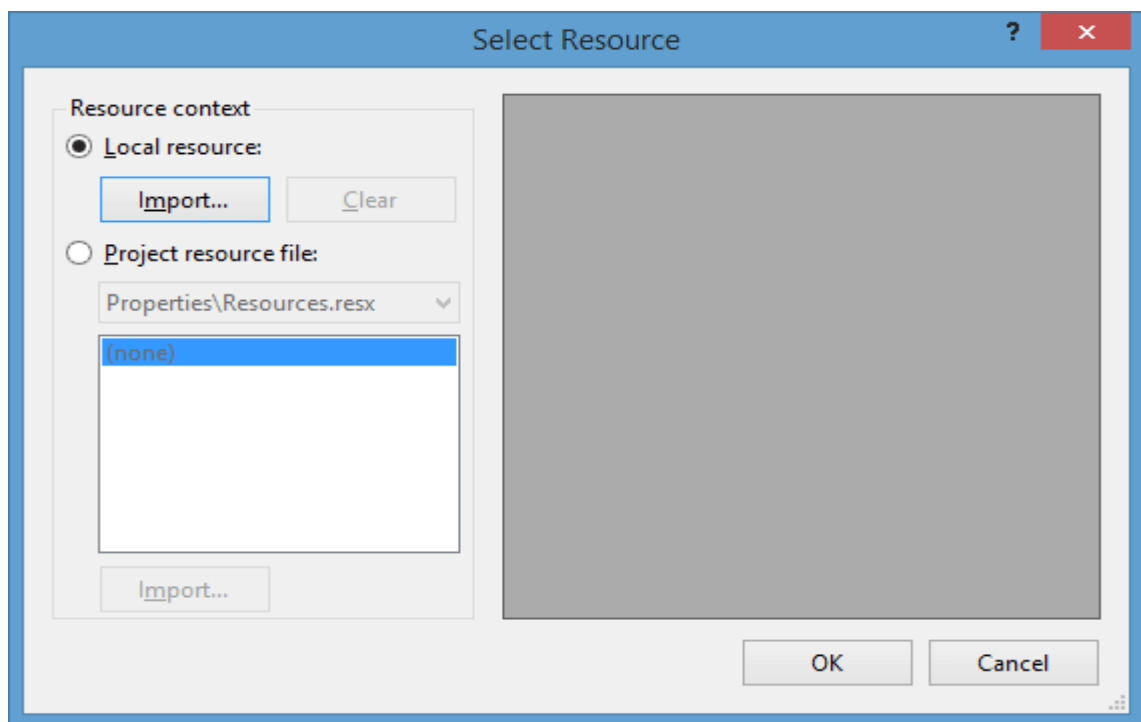
- Standard - кнопка має об'ємний вигляд (використовується за умовчанням),
- System - вид кнопки залежить від операційної системи,

#### Зображення на кнопці.

Як і для багатьох елементів управління, для кнопки можна задавати зображення за допомогою властивості *BackgroundImage*. Однак ми можемо також керувати розміщенням тексту і зображення на кнопці. Для цього треба використовувати властивість *TextImageRelation*. Вона набуває таких значень:

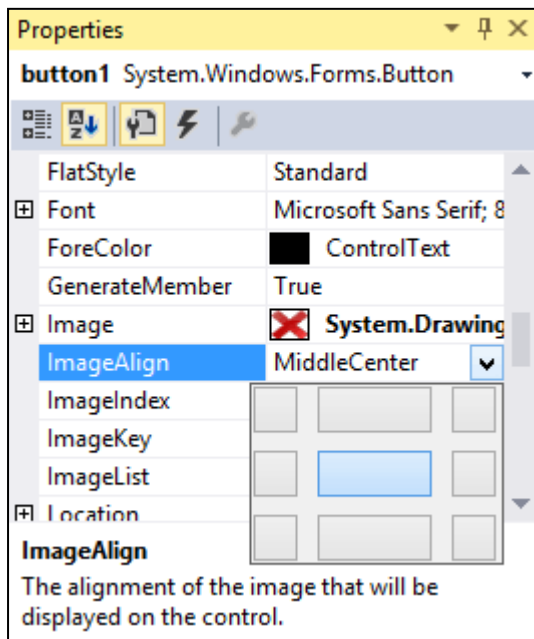
- Overlay: текст накладається на зображення,
- ImageAboveText: зображення розташовується над текстом,
- TextAboveImage: текст розташовується над зображенням,
- ImageBeforeText: зображення розташовується перед текстом,
- TextBeforeImage: текст розташовується перед зображенням.

Наприклад, встановимо для кнопки зображення. Для цього виберемо кнопку і у вікні властивостей натиснемо на поле *Image* (не плутати з *BackgroundImage*). Нам відкриється діалогове вікно встановлення зображення:



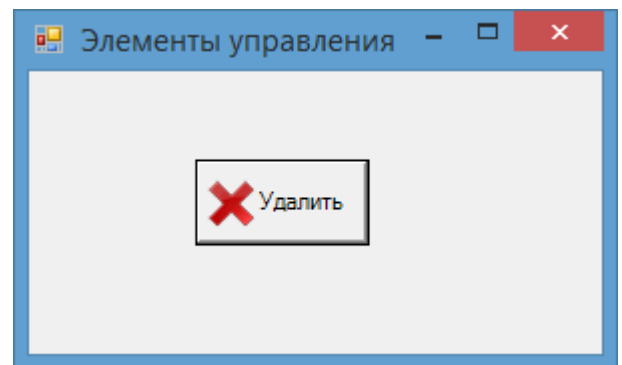
У цьому вікні виберемо опцію *Local Resource* і натиснемо на кнопку *Import*, після чого нам відкриється діалогове вікно для вибору файлу зображення.

Після вибору зображення ми можемо встановити властивість *ImageAlign*, яка управляє позиціонуванням зображення на кнопці:



Нам доступні 9 варіантів, за допомогою яких ми можемо прикріпити зображення до певної сторони кнопки. Залишимо значення за замовчуванням - *MiddleCenter*, тобто позиціонування по центру.

Потім перейдемо до властивості *TextImageRelation* і встановимо для неї значення *ImageBeforeText*. У підсумку ми отримаємо кнопку, де відразу після зображення йде напис на кнопці:



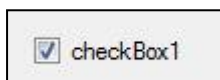
### **3. Кнопки-перемикачі (прапорці).**

Ці кнопки використовуються для формування режимів роботи застосунків. Вони дозволяють обрати одне або декілька значень із запропонованого набору. Типовим прикладом застосування таких кнопок-перемикачів є будь-яка закладка у вікні команди Параметри з меню Сервіс редактора MS Word.

Існують два типи перемикачів:

- *незалежні перемикачі*,
- *залежні перемикачі*.

**Незалежні перемикачі** в *Windows Forms* реалізовані за допомогою компонента *checkBox*. Якщо на формі (або в іншому контейнері) є декілька таких перемикачів, то кожний з них функціонує незалежно від інших. На формі компонент *checkBox* зображається у вигляді квадрата з позначкою (або без неї) і з текстовим позначенням:



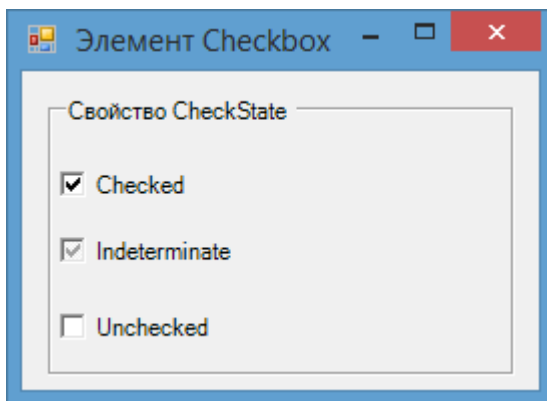
Елемент *checkBox* призначений для установки одного з двох значень: відмічений або не зазначено. Щоб відмітити прапорець, треба встановити у його властивості *checked* значення *true*. Стан незалежного перемикача може

задаватися трьома способами:

- у вікні властивостей об'єктів на етапі візуального конструювання;
- оператором програми, який визначає стан перемикача;
- клацанням миші по самому перемикачу в процесі виконання

застосування.

Крім властивості *Checked* у елемента *checkBox* є властивість *CheckState*, яка дозволяє задати для прапорця один з трьох станів - *Checked* (відмічений), *Indeterminate* (прапорець не відмічений - відмічений, але знаходиться в неактивному стані) і *Unchecked* (не відмічений):




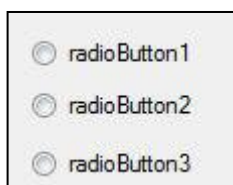
Також слід відзначити властивість *AutoCheck* - якщо вона має значення *false*, то ми не можемо змінювати стан прапорця. За замовчуванням вона має значення *true*.

При зміні стану прапорця він генерує подію *CheckedChanged*. Обробляючи цю подію, ми можемо отримувати змінений прапорець і здійснювати певні дії:

```
private void checkBox_CheckedChanged(object sender, EventArgs e)
{
    CheckBox checkBox = (CheckBox)sender; // приводимо відправника до
                                           // елемента типу CheckBox
    if (checkBox.Checked == true)
    {
        MessageBox.Show("Прапорець " + checkBox.Text + " відмічений");
    }
    else
    {
        MessageBox.Show("Прапорець " + checkBox.Text + " не відмічений");
    }
}
```

**Залежні перемикачі** призначені для реалізації альтернативного вибору - дозволяють вибрати тільки одне із запропонованої множини значень. Вони

реалізовані в *Windows Forms* за допомогою компонента *radioButton*  . Як правило, використовується у складі групи кнопок. Група залежних перемикачів може виглядати на формі, наприклад, таким чином:



У будь-який момент часу з групи залежних перемикачів може бути обраний клацанням миші тільки один. Включення не обраного перемикача негайно приведе до виключення раніше обраного (включеного) перемикача.

Залежні перемикачі зазвичай формуються в групи. Кожна група розміщується в **контейнері**. Контейнером для групи перемикачів може бути інший компонент, наприклад, форма, панель *Panel* або група *GroupBox*. Перемикачі взаємозалежні тільки в межах одного і того ж контейнера.

Основними властивостями компонента *radioButton* є властивості *Text*, *Enabled* і *Checked*, порядок використання яких такий же, як і у компонента *checkBox*.