

Лабораторна робота №9. Проект MyUtilites. Робота з елементом MenuStrip

Тема: Windows Forms. Проект MyUtilites. Робота з елементом MenuStrip.

Мета заняття: навчитись працювати з елементом MenuStrip для створення меню та підменю додатку, навчитись працювати з обробником подій Click елементів MenuStrip з допомогою Windows Forms мовою C# у середовищі Microsoft Visual Studio.

1. Теоретичні відомості

Для створення меню в Windows Forms застосовується елемент MenuStrip. Найбільш важливі властивості елементу MenuStrip:

- Dock: Прикріплює меню до однієї зі сторін форми;
- LayoutStyle: Задає орієнтацію панелі меню на формі. Може мати значення:
 - HorizontalStackWithOverflow: Розташування по горизонталі з переповненням - якщо довжина меню перевищує довжину контейнера, то нові елементи, що виходять за межі контейнера, не відображаються, тобто панель переповнюється елементами;
 - StackWithOverflow: Елементи розташовуються автоматично з переповненням;
 - VerticalStackWithOverflow: Елементи розташовуються вертикально з переповненням⁴
 - Flow: Елементи розміщуються автоматично, але без переповнення - якщо довжина панелі меню менше довжини контейнера, то виходять за межі елементи переносяться;
 - Table: Елементи позиціонуються в вигляді таблиці;
- ShowItemToolTips: Вказує, чи будуть відображатися спливаючі підказки для окремих елементів меню
- Stretch: Дозволяє розтягнути панель по всій довжині контейнера
- TextDirection: Задає напрямок тексту в пунктах меню

MenuStrip виступає свого роду контейнером для окремих пунктів меню, які представлені об'єктом ToolStripMenuItem. Додати нові елементи в меню можна в режимі дизайнера (рис. 1.1):

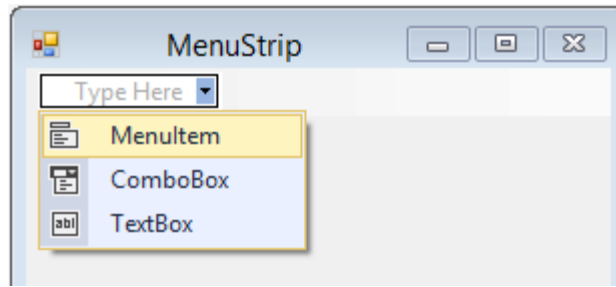


Рисунок 1.1 – MenuStrip. Режим дизайнер

Для додавання є три види елементів: MenuItem (об'єкт ToolStripMenuItem), ComboBox і TextBox. Таким чином, в меню можемо використовувати списки, що випадають і текстові поля, однак, як правило, ці елементи застосовуються в основному на панелі інструментів. Також ми можемо додати пункти меню в коді C # (рис. 1.2):

```
public Form1()
{
    InitializeComponent();

    ToolStripMenuItem fileItem = new ToolStripMenuItem("Файл");

    fileItem.DropDownItems.Add("Создать");
    fileItem.DropDownItems.Add(new ToolStripMenuItem("Сохранить"));

    menuStrip1.Items.Add(fileItem);

    ToolStripMenuItem aboutItem = new ToolStripMenuItem("О программе");
    aboutItem.Click += aboutItem_Click;
    menuStrip1.Items.Add(aboutItem);
}

void aboutItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("О программе");
}
```

Рисунок 1.2 — Додавання пунктів меню у коді програми

ToolStripMenuItem в конструкторі приймає текстову мітку, яка буде використовуватися як текст меню. Кожен подібний об'єкт має колекцію DropDownItems, яка зберігає дочірні об'єкти ToolStripMenuItem. Тобто один елемент ToolStripMenuItem може містити набір інших об'єктів ToolStripMenuItem. І таким чином, утворюється ієрархічне меню або структура у вигляді дерева.

Якщо передати при додаванні рядок тексту, то неявним чином буде створено об'єкт ToolStripMenuItem: fileItem.DropDownItems.Add ("Создать") (рис. 1.3).

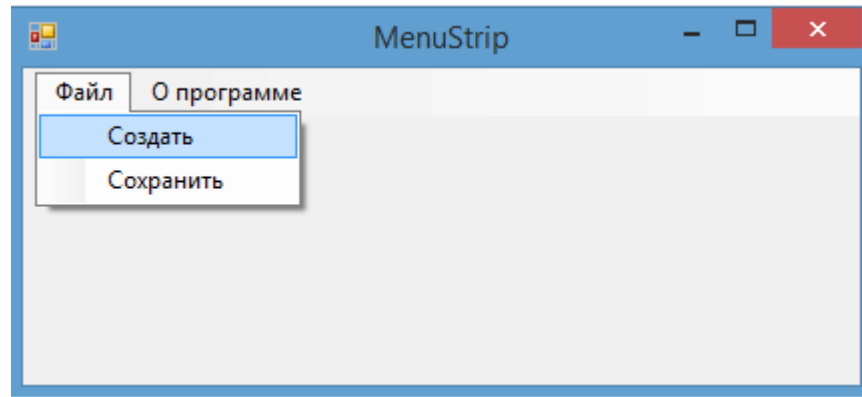


Рисунок 1.3 – Створення елемента «Создать» MenuItem «Файл»

Призначивши обробники для події Click, можемо обробити натискання на пункти меню: `aboutItem.Click += aboutItem_Click`.

Якщо потрібно швидко звернутися до якогось пункту меню, то можемо використовувати клавіші швидкого доступу. Для завдання клавіш швидкого доступу використовується властивість `ShortcutKeys` (рис. 1.4) :

```

1 public partial class Form1 : Form
2 {
3     public Form1()
4     {
5         InitializeComponent();
6
7         ToolStripMenuItem fileItem = new ToolStripMenuItem("Файл");
8
9         ToolStripMenuItem saveItem = new ToolStripMenuItem("Сохранить") { Checked = true, CheckOnClick = true };
10        saveItem.Click+=saveItem_Click;
11        saveItem.ShortcutKeys = Keys.Control | Keys.P;
12
13        fileItem.DropDownItems.Add(saveItem);
14        menuStrip1.Items.Add(fileItem);
15    }
16
17    void saveItem_Click(object sender, EventArgs e)
18    {
19        MessageBox.Show("Сохранение");
20    }
21 }

```

Рисунок 1.4 – Створення клавіш швидкого доступу у коді програми

Клавіші задаються за допомогою перерахування `Keys`. В даному випадку після натискання на комбінацію клавіш `Ctrl + P`, буде спрацьовувати натискання на пункт меню «Зберегти».

За допомогою зображень можемо урізноманітнити зовнішній вигляд пунктів меню. Для цього можемо використовувати такі властивості:

- `DisplayStyle`: Визначає, чи буде відображатися на елементі текст, або зображення, або і те й інше;
- `Image`: Вказує на саме зображення;
- `ImageAlign`: Встановлює вирівнювання зображення щодо елемента;
- `ImageScaling`: Вказує, чи буде зображення розтягуватися, щоб

заповнити весь простір елемента.

- ImageTransparentColor: Вказує, чи буде колір зображення прозорим

Якщо зображення для пункту меню встановлює в режимі дизайнера, то потрібно вибрати у вікні властивість пункт Image, після чого відкриється вікно для імпорту ресурсу зображення в проект (рис. 1.5).

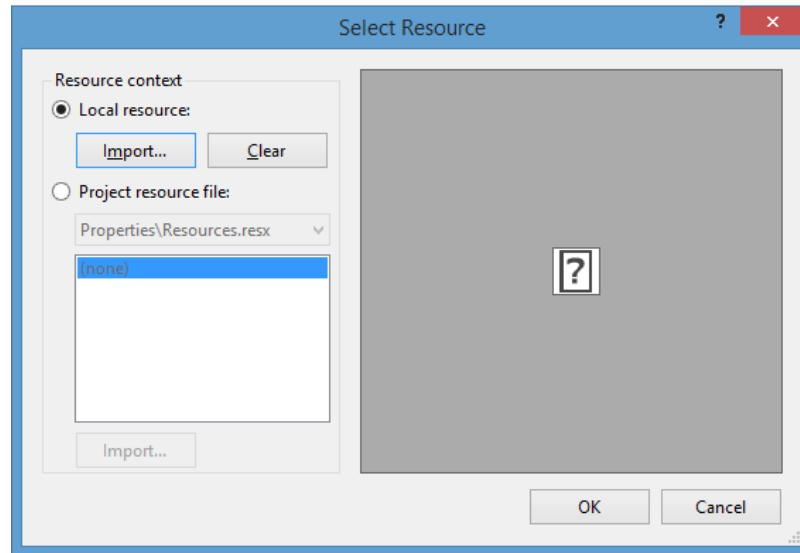


Рисунок 1.5 – Встановлення зображення пункту меню у режимі дизайнера

Щоб вказати, як розмістити зображення, у властивості DisplayStyle треба встановити значення Image. Якщо хочемо, щоб кнопка відображала тільки текст, то треба вказати значення Text, або можна комбінувати два значення за допомогою іншого значення ImageAndText. За замовчуванням зображення розміщується зліва від тексту (рис. 1.6):

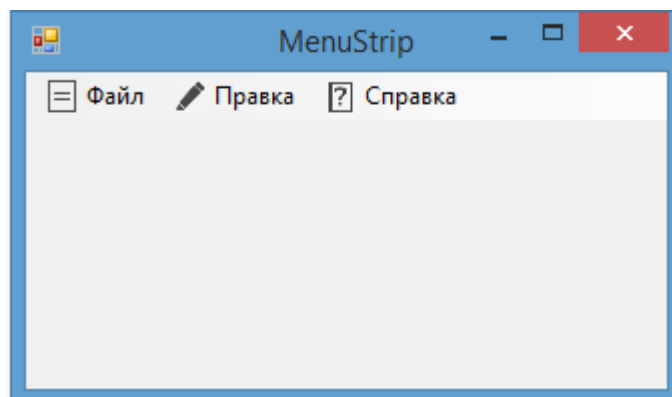


Рисунок 1.6 – Приклад відображення зображення та тексту пункту меню

Також можна встановити зображення пункту меню в коді програми:

```
fileToolStripMenuItem.Image = Image.FromFile(@"D:\Icons\block32.png");
```

2. Практичне завдання



У процесі виконання завдань лабораторної роботи необхідно формувати набори тестових даних для перевірки правильності виконання програмного коду. Створений код і результати перевірки його роботи потрібно помістити у звіт. Тестувати роботу програми рекомендується після додання чи зміни кожного оператора виведення.

Завдання 1. Створити проект для розв'язання задачі.

Створити меню File, Help додатку MyUtilites.

1. Запускаем Visual Studio. Вибираємо створити додаток Windows. Назва проекту MyUtilites.
2. Додаємо Menu. Для цього на панелі інструментів знаходимо MenuStrip (рис. 2.1).

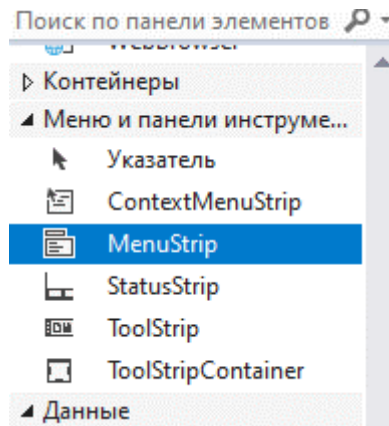


Рисунок 2.1 – Вибір елемента MenuStrip на панелі інструментів

3. Перейменувати назву Form1 на MainForm, Text на «MyUtilites» (рис. 2.2)

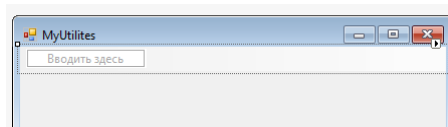


Рисунок 2.2 – Форма «MyUtilites» з елементом MenuStrip

4. Створюємо елементи меню: File, Help (рис. 2.3)

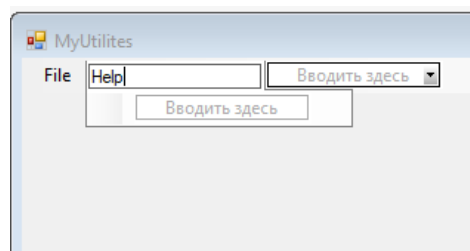


Рисунок 2.3 – Елемент MenuStrip. Режим дизайнера

5. Створюємо підменю File →Exit, Help →About (рис. 2.4)

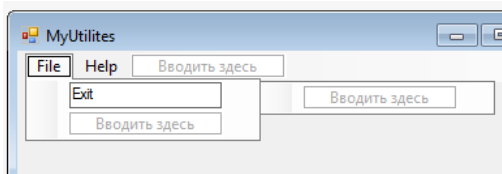


Рисунок 2.4 – Елемент MenuStrip. Режим дизайнера. Створення підменю

6. Перейменувати поле Name підменю Exit на tsmiExit та створюємо для нього подію Click

```
namespace MyUtilites
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();
        }

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

7. У конструктор події додаємо метод this.Close();

```
namespace MyUtilites
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();
        }

        private void tsmiExit_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```

8. Перейменуємо поле Name підменю About на tsmiAbout. Створюємо для елемента подію Click, у якій додаємо метод MessageBox.Show(); в дужках в лапках пишемо текст “Програма MyUtilites містить набір невеликих програм, які можуть стати у нагоді. А головне навчить мене основам програмування на C#. Автор: Іванов Іван ” (рис. 2.5). Змінити прізвище та ім’я автора на власне. Можна додати перенесення тексту на новий рядок через \n.

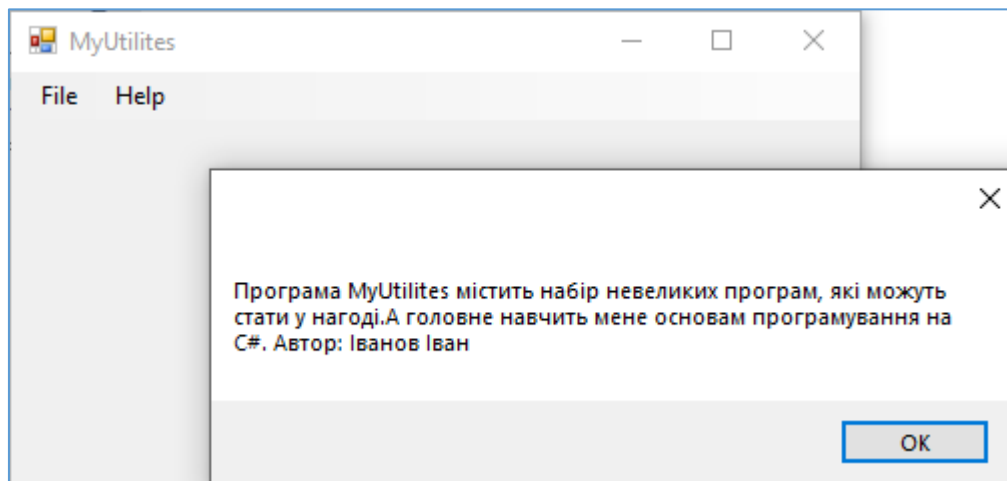


Рисунок 2.5 – Метод `MessageBox.Show` події `Click` підменю `About`

9. Додаємо заголовок вікна (рис. 2.6). Для цього в `MessageBox.Show ()` в дужках після тексту пишемо через кому в лапках заголовок (“.....”, “Про програму”).

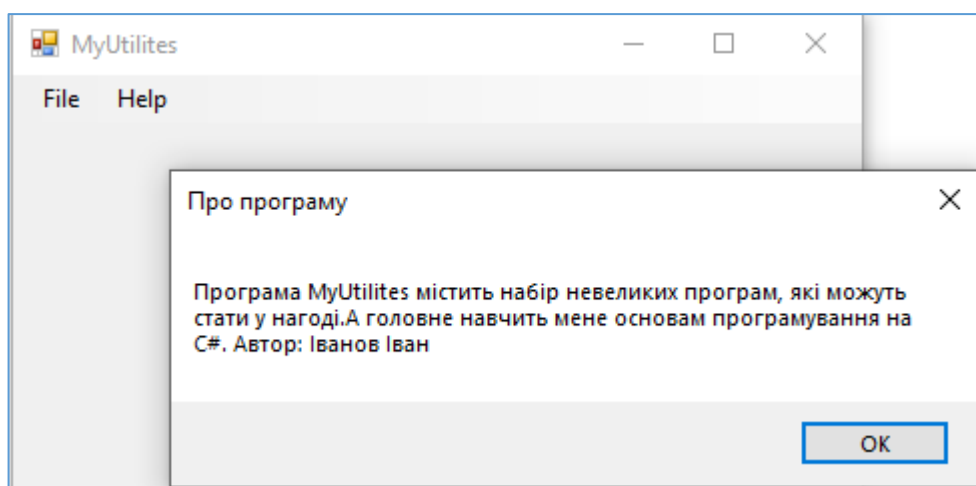


Рисунок 2.6 – Вікно `MessageBox`, яке викликається після натиснення `About`

10. Перевірте роботу елементів меню додатку `MyUtilites` (`Ctrl+F5`). Оформіть звіт.

3. Контрольні запитання

1. Призначення елемента `MenuStrip`.
2. Як створити меню, підменю для форми?
3. Як налаштувати дії при виборі елемента підменю?
4. Які види елементів можна додати в конструкторі для `MenuStrip`?
5. Яке призначення властивостей `Dock`, `LayoutStyle`, `Stretch`?
6. Як додати пункти меню у програмному коді?

7. Як задати клавіші швидкого доступу пунктів меню
8. Як додати зображення (піктограми) до елементів меню?

Література

1. Евдокимов П. В. С# на примерах. СПб.: Наука и Техника, 2019. 320 с.
2. Маки А. Введение в .NET 4.0 и Visual Studio 2010 для профессионалов; пер. с англ. М. : ООО ИД "Вильямс", 2010. 416 с
3. С# 7.0. Справочник. Полное описание языка.: Пер. с англ. СПб.: ООО "Альфа-книга", 2018. 1024 с.
4. Троелсен, Эндрю, Джепикс, Филипп. Язык программирования С# 7 и платформы .NET и .NET Core. СПб. : ООО "Диалектика", 2018. 1328 с.
5. Офіційний сайт компанії Microsoft щодо технологій WPF та Windows Forms [Електронний ресурс]. – Режим доступу : <http://window-sclient.net>.
6. С#. Теорія та практика. URL: https://www.bestprog.net/uk/sitemap_ua/c-3
7. Сажин А. Справочник по языку программирования С#. URL: <https://brainoteka.com/blogs/c-spravochnik>.
8. С# Tutorial URL <https://www.theengineeringprojects.com>.
9. Уроки С#. URL: <https://itproger.com/course/csharp>.
10. Полное руководство по С# 8 и .NET Core. URL: <https://metanit.com/sharp/>
11. С#. Мини-программы. URL: <https://geekbrains.ru/chapters/970>