

## Лабораторна робота №15. Проект MyUtilites. View Picture

**Тема:** Windows Forms. Проект MyUtilites. View Picture

**Мета заняття:** навчитись працювати з класом OpenFileDialog для створення діалогу відкриття графічних файлів, напрацювати навички з обробником подій Click елементів Button та виводом інформації у елемент PictureBox з допомогою Windows Forms мовою C# у середовищі Microsoft Visual Studio.

### 1. Теоретичні відомості

Вікна відкриття та збереження файлу представлені класами **OpenFileDialog** та **SaveFileDialog**. Вони мають ряд загальних властивостей, серед яких можна виділити наступні:

- **DefaultExt:** Встановлює розширення файлу, яке додається за замовчуванням, якщо користувач ввів ім'я файлу без розширення.
- **AddExtension:** При значенні true додає до імені файлу розширення при його відсутності розширення береться із властивості DefaultExt або Filter.
- **CheckFileExists:** Якщо має значення true, то перевіряє наявність файлу з вказаним ім'ям.
- **CheckPathExists:** Якщо має значення true, то перевіряє існування шляху до файлу з вказаним ім'ям.
- **FileName:** Повертає повне ім'я файлу, обраного в діалоговому вікні.
- **Filter:** Задає фільтр файлів, завдяки чому в діалоговому вікні можна відфільтрувати файли з розширення. Фільтр задається у такому форматі *Назва\_файлів* */\*. розширення*. Наприклад, текстові файли (\*.txt)|\*.txt. Можна задати відразу кілька фільтрів, для цього вони поділяються вертикальною лінією |, наприклад, Bitmap files (\*.bmp) |\*.bmp|Image files (\*.jpg)|\*.jpg
- **InitialDirectory:** Встановлює каталог, який відображається під час першого виклику вікна
- **Title:** Заголовок діалогового вікна

Окремо у класу SaveFileDialog можна ще виділити пару властивостей:

- **CreatePrompt:** При значенні true в разі, якщо зазначений неіснуючий файл, то буде відображатися повідомлення про його створення.
- **OverwritePrompt:** При значенні true в разі, якщо зазначений існуючий файл, то буде відображатися повідомлення про те, що файл буде перезаписано

Щоб відобразити діалогове вікно, треба викликати метод ShowDialog(). Додамо на форму текстове поле textBox1 і дві кнопки button1 і button2. Також перетягнемо з панелі інструментів компоненти OpenFileDialog і SaveFileDialog.

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();

        button1.Click += button1_Click;
        button2.Click += button2_Click;
        openFileDialog1.Filter = "Text files (*.txt)|*.txt|All files (*.*)|*.*";
        saveFileDialog1.Filter = "Text files (*.txt)|*.txt|All files (*.*)|*.*";
    }
    // збереження файлу
    void button2_Click(object sender, EventArgs e)
    {
        if (saveFileDialog1.ShowDialog() == DialogResult.Cancel)
            return;
        // отримаємо вибраний файл
        string filename = saveFileDialog1.FileName;
        // зберігаємо текст в файл
        System.IO.File.WriteAllText(filename, textBox1.Text);
        MessageBox.Show("Файл збережений");
    }
    // відкриття файлу
    void button1_Click(object sender, EventArgs e)
    {
        if (openFileDialog1.ShowDialog() == DialogResult.Cancel)
            return;
        // отримаємо вибраний файл
        string filename = openFileDialog1.FileName;
        // читаємо файл в рядок
        string fileText = System.IO.File.ReadAllText(filename);
        textBox1.Text = fileText;
        MessageBox.Show("Файл відкрит");
    }
}
```

При натисканні на першу кнопку буде відкриватися вікно відкриття файлу. Після вибору файлу він буде зчитуватися, а його текст буде відображатися в текстовому полі. Клік на другу кнопку відобразить вікно для збереження файлу, в якому треба встановити його назву. І після цього відбудеться збереження тексту з текстового поля в файл.

## 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. Створити проект «View Picture».

1. Запускаем Visual Studio. Відкриваємо проект MyUtilites (продовження проекту, створеного у лабораторних роботах 10-14).

2. Додаємо ще одну сторінку ViewPicture. Вибираємо елемент tabControl1 та властивість TabPages (рис. 1).

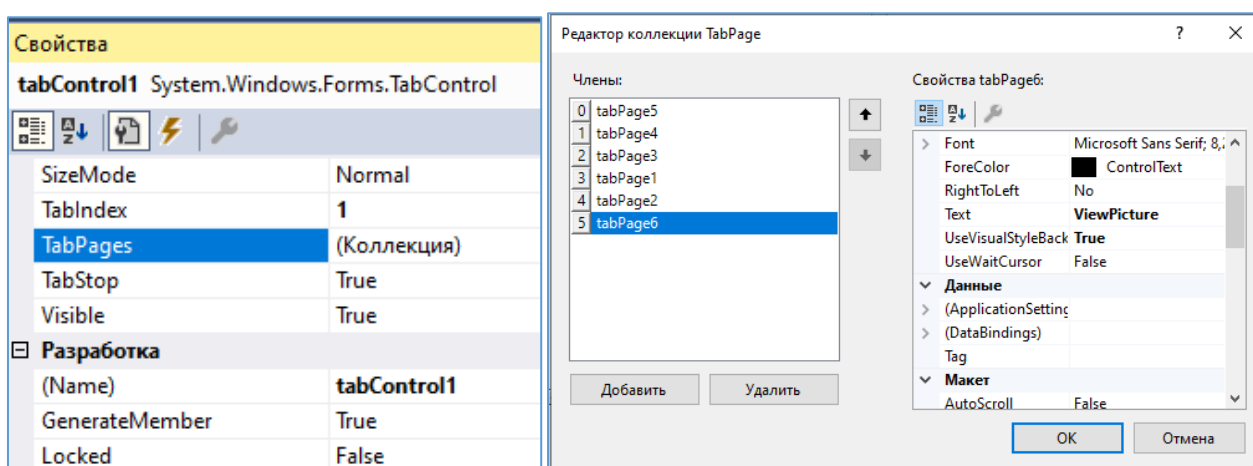


Рисунок 1 – Додавання TabPage 6 (сторінка ViewPicture) до елемента tabControl1

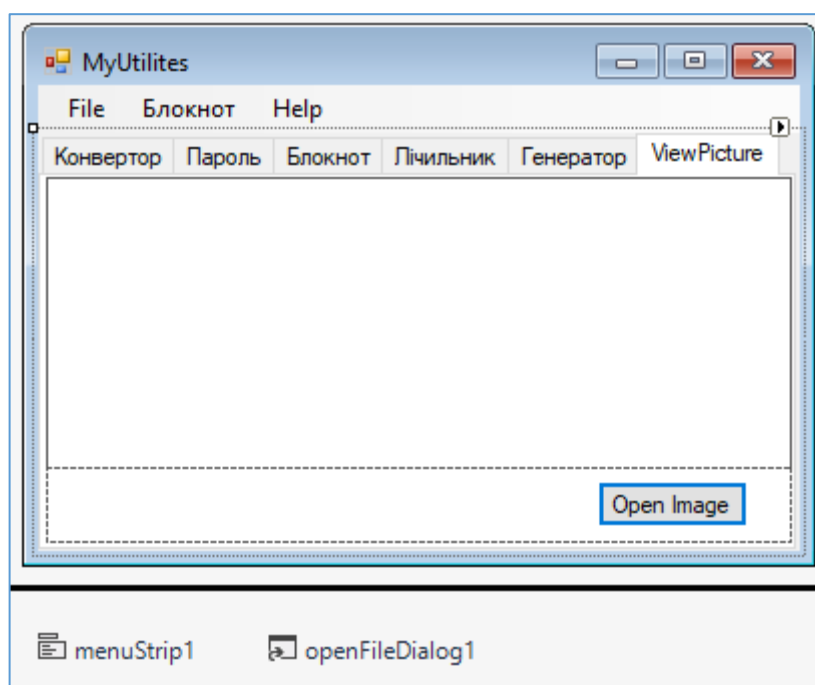


Рисунок 2 – Дизайн форми. Додаток «ViewPicture»

3. Додаємо на форму 2 елемент OpenFileDialog. Після додавання він буде відображатись внизу дизайнера форми (рис. 2).

4. Додаємо елемент PictureBox та розтягуємо його по краям, залишивши місце для кнопки внизу. Додаємо до форми кнопку. Встановити властивість елементів за таблицею 1

Таблиця 1 – Властивості елементів форми. Додаток «ViewPicture»

Елемент	Поле	Значення
PictureBox	SizeMode	Zoom
	Anchor	Top, Bottom, Left, Right
Button	Name	btOI
	Text	Open Image
	Anchor	Top, Right

Значення Zoom властивості SizeMode використовується для того, щоб зображення, що вибирається, змінювала розміри під розмір елементу PictureBox.

Значення «Top, Bottom, Left, Right» властивості Anchor робить кругову прив'язку до форми по колу.

Значення «Top, Right» властивості Anchor елемента Bottom прив'язує кнопку до правого нижнього краю.

5. Створюємо обробник події Click для кнопки. У конструктору форми після методу InitializeComponent(), додаємо такий код (рис. 3):

```
openFileDialog1.FileName="";  
openFileDialog1.Filter = "Image Files(*.BMP;*.JPG;*.GIF)|*.BMP;*.JPG;*.GIF|All files (*.*)|*.*";
```

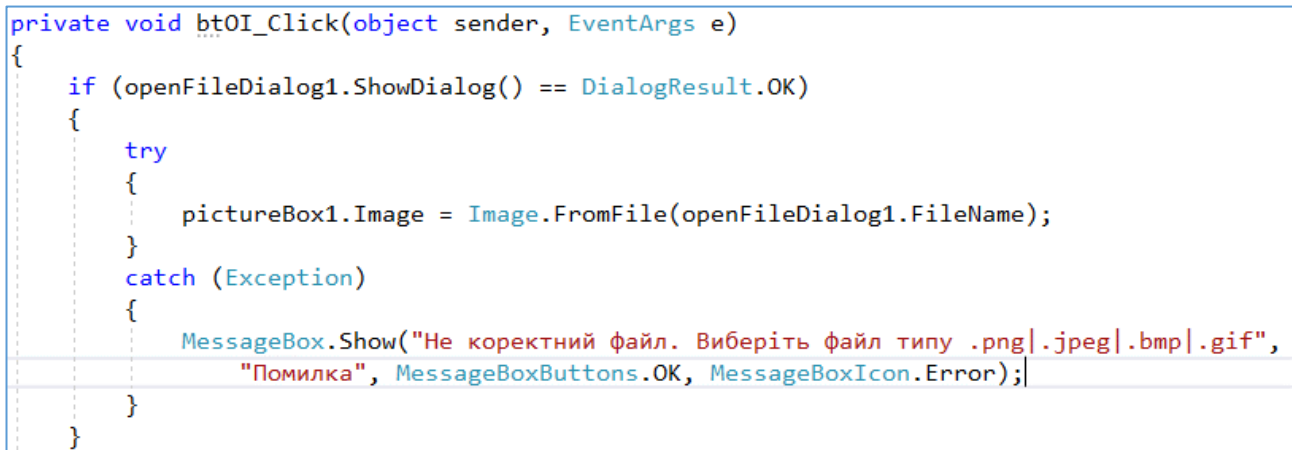
```
public MainForm()  
{  
    InitializeComponent();  
    rnd = new Random();  
    metrica = new Dictionary<string, double>();  
    metrica.Add("mm", 1);  
    metrica.Add("cm", 10);  
    metrica.Add("dm", 100);  
    metrica.Add("m", 1000);  
    metrica.Add("kl", 1000000);  
    metrica.Add("mile", 1609344);  
    openFileDialog1.FileName = "";  
    openFileDialog1.Filter = "Image Files(*.BMP;*.JPG;*.GIF)|*.BMP;*.JPG;*.GIF|All files (*.*)|*.*";  
}
```

Рисунок 3 – Конструктор форми. Додаємо openFileDialog1

6. В обробнику події Click для кнопки робимо перевірку, що файл дійсно вибраний та якщо вибраний, то передати шлях до файлу елемента PictureBox,

якщо файл не є зображенням, то видаємо помилку. Для цього використовуємо конструкцію try...catch (рис. 4)

```
if (openFileDialog1.ShowDialog() == DialogResult.OK)
{
    try
    {
        pictureBox1.Image = Image.FromFile(openFileDialog1.FileName);
    }
    catch (Exception)
    {
        MessageBox.Show("Не коректний файл. Виберіть файл типу.png|.jpeg|.bmp|.gif", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```



```
private void btOI_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        try
        {
            pictureBox1.Image = Image.FromFile(openFileDialog1.FileName);
        }
        catch (Exception)
        {
            MessageBox.Show("Не коректний файл. Виберіть файл типу .png|.jpeg|.bmp|.gif",
                "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
```

Рисунок 4 – Код обробника подій Click для кнопки «Open Image»

7. Перевірте роботу програми «View Picture» додатку MyUtilites (Ctrl+F5). Оформіть звіт.

### 3. Контрольні запитання

1. Для чого призначений елемент OpenFileDialog? Як додати його до проекту?
2. Для чого використовується метод ShowDialog() у коді програми?
3. Яке призначення властивостей FileName та Filter у програмі?
4. Як задати типи графічних файлів, що будуть відкриватись у програмі?
5. Що буде робити програма, якщо вибраний не графічний файл ?
6. Як програма перевіряє тип файлу, що вибраний для відкриття?
7. Як забезпечити, щоб розмір зображення, що відкривається, відповідав розмірам вікна програми?

## 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](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. Программа для просмотра изображений на С#. URL: <https://cutt.ly/EyyFARo>.