

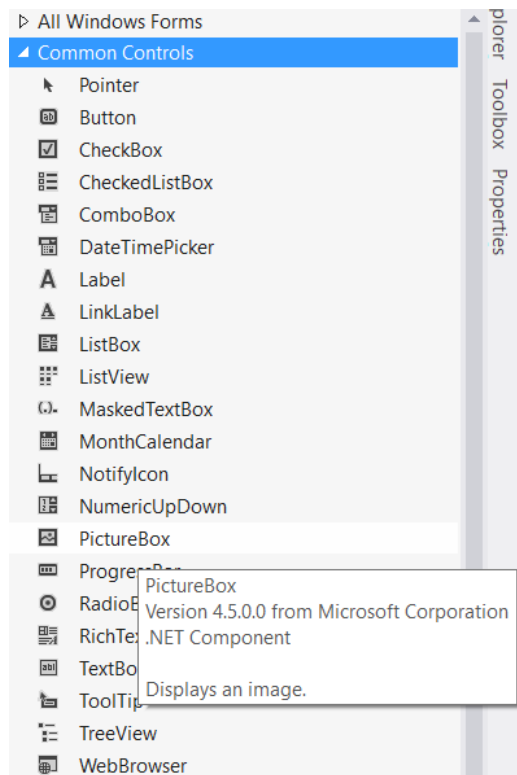
## Лаб. 2. Завантаження та відображення зображень у MS VS2012 Windows Form Application

**Мета:** Засвоїти основні принципи роботи із зображеннями у Windows Form Application у середовищі Microsoft Visual Studio 2012. Навчитись використовувати компоненти PictureBox та діалоги відкривання файлів, редагувати зображення.

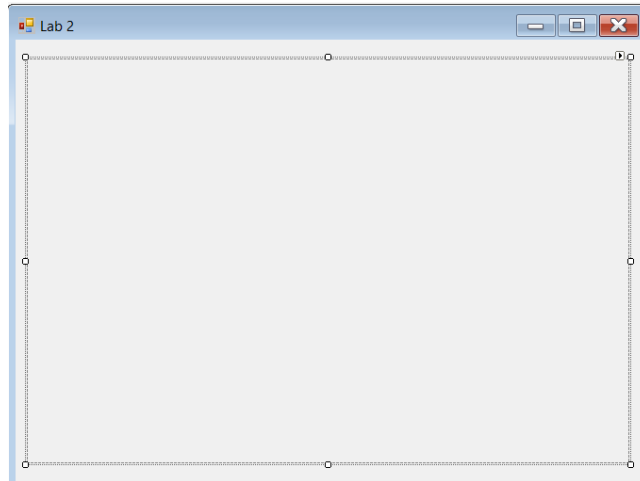
**Завдання:** Створити віконний проєкт WF у середовищі MS VS2012 із компонентою PictureBox, у якій навчитись відображати рисунки, завантажені з файлів та редагувати ці зображення.

### 1. PictureBox та Bitmap

Для відображення рисунків у компонентах Windows Form передбачений засіб PictureBox, який можна знайти серед Common Controls:



Такий він має вигляд на нашій формі:



Для відображення медіа даних на цьому засобі, до нього треба прив'язати об'єкт класу, у якому містяться дані зображення. Для цього використовують клас **Bitmap**:

```
Bitmap^ imgData; // змінна, у якій будуть зберігатись дані зображення
```

Клас має складну структуру та велику кількість методів для перетворення, декодування медіа даних та доступу до них.

Для створення об'єкту класу **Bitmap** у нових версіях Microsoft Visual Studio замість `new` використовується ключове слово `gcnew`:

```
imgData = gcnew Bitmap("D:\\Fig_1.jpg"); // ініціалізація файлу із зображенням
```

Цей оператор виділяє для об'єкту область пам'яті у керованій купі, що дозволяє збирачу сміття його знищувати автоматично, тобто програмісту нема потреби знищувати об'єкт за допомогою команди `delete`.

Один з конструкторів класу **Bitmap** приймає адресу медіа даних та перетворює внутрішній формат.

Дані зберігаються у вигляді матриці розміром із зображення, кожна точка зображення має формат **RGB**, тобто Червоний-Зелений-Синій.

Для доступу до даних пікселя на зображенні використовують клас **Color**, наприклад наступним чином:

```
Color pixelColor = imgData->GetPixel( x, y );
```

Тут ми передали у клас дані про колір пікселя з координатами на нашому зображенні **X** та **Y**. Нумерація пікселів у зображенні ведеться від нуля. Початок координат знаходиться у верхньому лівому куті (тобто вісь **X** напрямлена вниз, а не вгору, як зазвичай у математиці).

Конкретний колір ми можемо отримати тепер через нашу змінну:

`pixelColor.R`.

У цьому полі зберігається інформація про інтенсивність червоного кольору нашого пікселя.

Задати колір пікселя можна наступним чином:

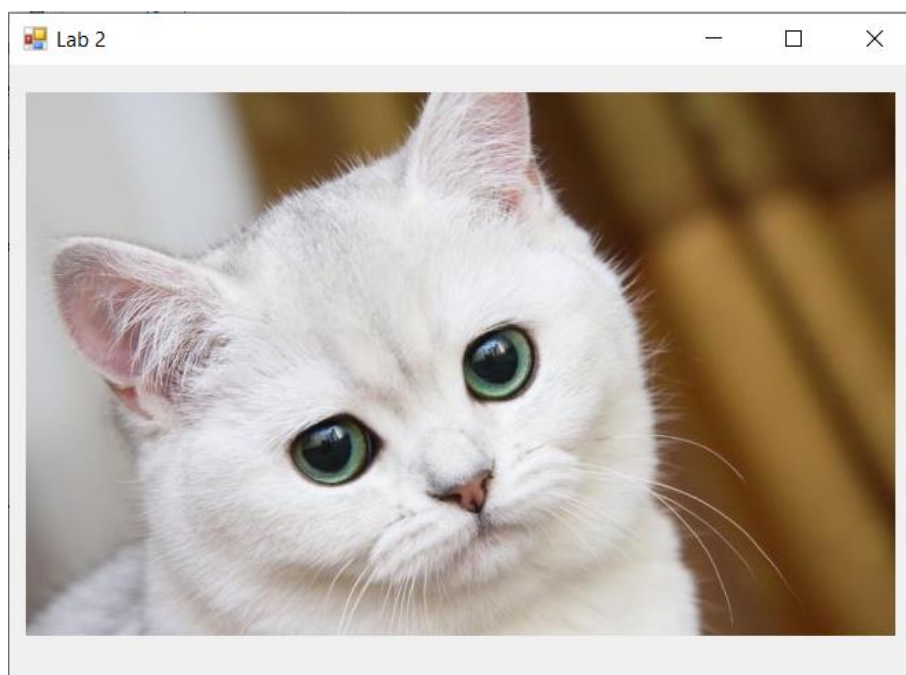
```
Color newColor = Color::FromArgb( pixelColor.R, 0, 0 );  
imgData->SetPixel( x, y, newColor );
```

Тут ми залишили червоний канал пікселя та надали синьому та зеленому каналам значення 0 (тобто прибрати їх).

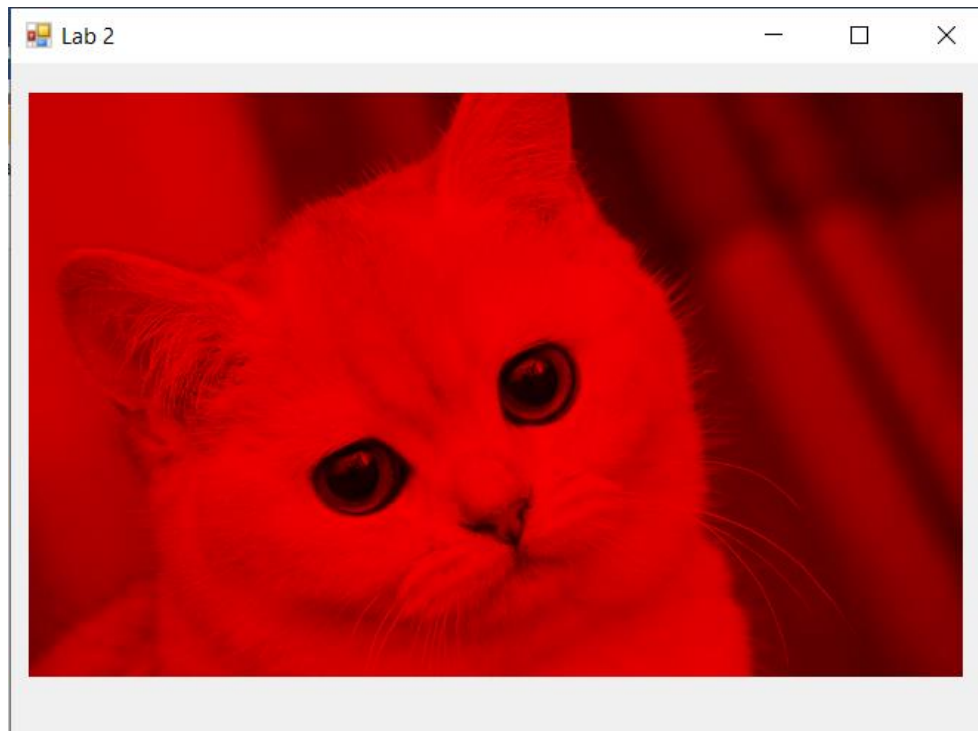
Щоб відобразити наше зображення на формі у PictureBox, треба передати дані нашого зображення:

```
pbMainPic->Image = imgData; //Передача даних на відображення у компоненті PictureBox
```

Тоді ми отримаємо зображення на формі (приклад із зображенням котика):



Якщо ж залишити лише червоний канал для кожного пікселя зображення, отримаємо наступне:

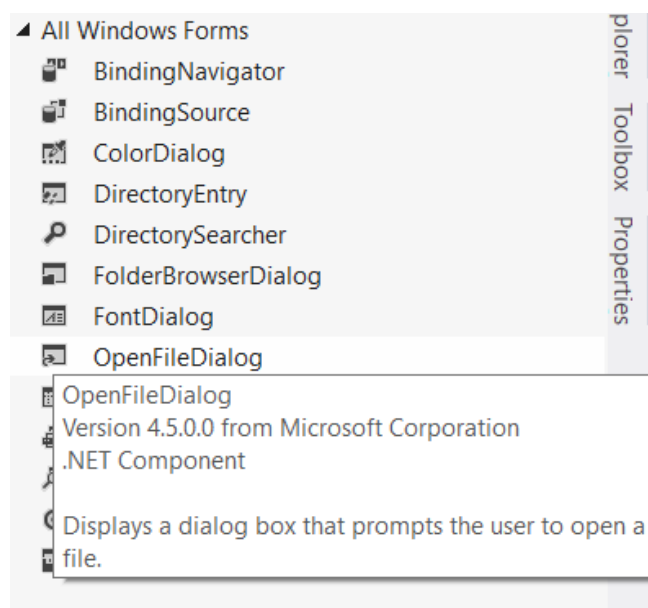


Повна інформація про клас `Bitmap` та його використання як завжди представлена у довідці MSDN:

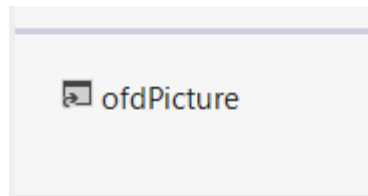
<https://docs.microsoft.com/en-us/dotnet/api/system.drawing.bitmap?redirectedfrom=MSDN&view=netframework-4.5>

## 2. Стандартний діалог відкривання файлу

У каталозі компонент передбачені діалоги відкривання та збереження файлів із стандартним інтерфейсом користувача.



При додавання компоненти на форму, внизу дизайнера він буде відображатись наступним чином:



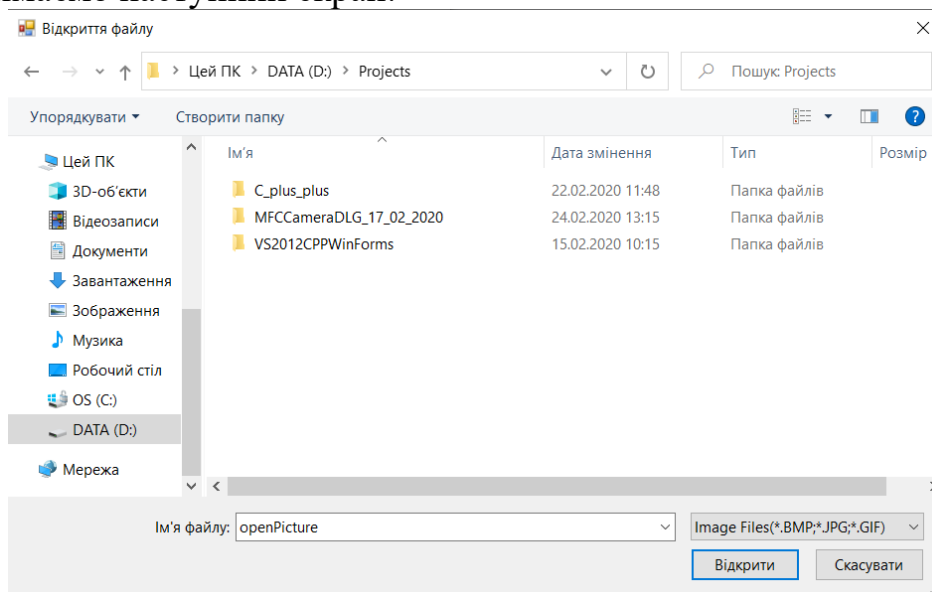
Для передачі шляху до файлу треба створити змінну, у якій цей шлях буде зберігатись:

```
String^ fileName; //Змінна, у якій буде шлях до файлу
```

Для цього є клас `String`.

Для старту діалогу використовується метод `ofdPicture->ShowDialog()`

Отримаємо наступний екран:



Однак, отримати дані про шлях до файлу можна лише після того, як користувач зробить свій вибір, тому доцільно використати наступну умову:

```
if (System::Windows::Forms::DialogResult::OK == ofdPicture->ShowDialog())  
{  
    fileName = ofdPicture->FileName;  
}
```

Тобто, якщо користувач натиснув клавішу ОК (у нашому випадку «Відкрити»), програма отримала результат `:DialogResult::OK`

тоді можна отримати шлях до файлу (інакше виникне помилка).

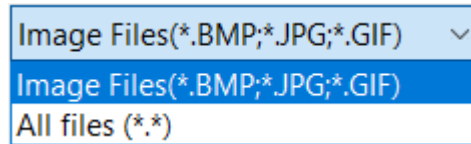
Для налагодження вибору файлів лише певного типу у класі діалогу передбачене поле `Filter` стрічкового типу:

```
ofdPicture->Filter = "Image Files(*.BMP;*.JPG;*.GIF)|*.BMP;*.JPG;*.GIF|All files (*.*)|*.*";
```

Формат стрічки є наступним:

“Назва фільтру1|Розширення файлів через ; | Назва фільтру2 |  
Розширення файлів через ;”

У нашому випадку ми отримаємо:



Діалог відкривання файлів у MSDN:

[https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-6.0/aa279103\(v%3dvs.60\)](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-6.0/aa279103(v%3dvs.60))

### Додаткове завдання

1. Перед відображенням на формі змінити кольори рисунку на градації сірого. Це просто зробити, якщо замість кожного кольору пікселя задати середнє між його кольорами на кольоровому зображенні.
2. Перетворити сіре зображення у чорно-біле (чорному відповідає (0,0,0), а білому (255,255,255) залежно від порогу (якщо інтенсивність більша за поріг, то колір білий, інакше – чорний). Значення порогу вводити через TextBox.