

はじめに

.NET フレームワークによるウィンドウズアプリケーションを作成する。

準備

1. 「スタート」 → 「Microsoft Visual Studio 2010」 → 「Microsoft Visual Studio 2010」を起動
2. 「ファイル」 → 「新規作成」 → 「プロジェクト」
3. (C#が選択されていなければ) 「他の言語」 → 「Visual C#」 を選択
4. 「Windows フォームアプリケーション」 を選択
5. 名前の欄に 「ImageViewer」 と入力 (名前は任意) → 「OK」
6. 緑色の三角ボタンを押して実行する。ウィンドウの×印で終了する。

画面の設計

1. 「デザイン」 タブが開かれていることを確認する
2. 「ツールボックス」 が表示されていることを確認する。ツールボックスのピン留めをしておくと便利
3. Form1 を選択した状態で、「プロパティ」 ウィンドウの「Text」 欄に 「Image Viewer」 と入力する
4. 同じく「プロパティ」 ウィンドウの「Size」 欄に 「600, 400」 と入力する
5. 「ツールボックス」 から 「MenuStrip」 を左クリックし、Form1 上で再度左クリック
6. 「ここへ入力」 欄をダブルクリックして、「File」 と入力する
7. 下側に表示された「ここへ入力」 欄に 「Open」、さらに下側に表示された「ここへ入力」 欄に 「Exit」 を入力
8. 「ファイル」 の右側の「ここへ入力」 欄に 「Edit」 と入力し、下側の「ここへ入力」 欄に 「binarize」 と入力する
9. 「ツールボックス」 から 「PictureBox」 を選択して、フォーム上に設置する
10. 「プロパティ」 の欄に 「pictureBox1」 と表示されていることを確認したうえで 「Location」 を 「12,29」、「Size」 を 「250, 250」、「BackColor」 を任意の色に設定する
11. 「ツールボックス」 から 「PictureBox」 を選択して、フォーム上に設置する
12. 「プロパティ」 の欄に 「pictureBox2」 と表示されていることを確認したうえで 「Location」 を 「289,29」、「Size」 を 「250, 250」、「BackColor」 を任意の色に設定する
13. 「ツールボックス」 から 「ProgressBar」 を選択して、フォーム上に設置する
14. 「プロパティ」 の欄に 「pogressBar1」 と表示されていることを確認したうえで 「Location」 を 「12,311」、「Size」 を 「526, 23」 に設定する

15. 緑色の三角ボタンを押して実行する。ウィンドウの×印で終了する。

ファイルオープン処理

1. 「ツールボックス」から「OpenFileDialog」を左クリックし、その状態で Form を左クリックすると「openFileDialog1」が追加される
2. Form の「File」→「Open」をダブルクリックする
3. 「openToolStripMenuItem_Click」関数が自動で生成されたことを確認する
4. 関数内に以下のコードを入力する

```
int i, j;
if (openFileDialog1.ShowDialog() == DialogResult.OK)
{
    Bitmap bmp = new Bitmap(openFileDialog1.FileName);
    pictureBox1.Image = bmp;
    Bitmap bmpout = new Bitmap(bmp.Size.Width, bmp.Size.Height);
    progressBar1.Maximum = bmp.Size.Height;

    for (j = 0; j < bmp.Size.Height; j++)
    {
        for (i = 0; i < bmp.Size.Width; i++)
        {
            bmpout.SetPixel(i, j, bmp.GetPixel(i, j));
        }
        progressBar1.Value++;
    }
    pictureBox2.Image = bmpout;
    progressBar1.Value = 0;
}
```

5. 緑色の三角ボタンを押して実行する。「File」から「Open」を選択し画像ファイルを開く

アプリケーション終了処理

1. 「Form1」のデザインから「File」→「Exit」をダブルクリックする
2. `exitToolStripMenuItem_Click`関数が開かれるので、以下のコードを追加する

```
Environment.Exit(0);
```

3. 緑色の三角ボタンを押して実行する。「File」から「Exit」をクリックするとアプリケーションが終了することを確認する

二値化処理

1. 「Form1」のデザインから「Edit」→「Binarize」をダブルクリックする
2. binarizeToolStripMenuItem_Click関数が開かれるので、以下のコードを追加する

```
Bitmap bmp = new Bitmap(pictureBox1.Image);
int i, j;
Bitmap bmpout = new Bitmap(bmp.Size.Width, bmp.Size.Height);
progressBar1.Maximum = bmp.Size.Height;
for (j = 0; j < bmp.Size.Height; j++)
{
    for (i = 0; i < bmp.Size.Width; i++)
    {
        //赤 (R) のみで判別
        if (bmp.GetPixel(i, j).R < 128)
        {
            bmpout.SetPixel(i, j, Color.FromArgb(0, 0, 0));
        }else{
            bmpout.SetPixel(i, j, Color.FromArgb(255, 255, 255));
        }
    }
    progressBar1.Value++;
}
pictureBox2.Image = bmpout;
progressBar1.Value = 0;
```

3. 緑色の三角ボタンを押して実行する。「File」→「Open」によって画像ファイルを開く
4. 「Edit」から「Binarize」をクリックすると二値化された画像が右側に表示されることを確認する

.演算子 (C# リファレンス) MSDN より

ドット演算子 (.) は、メンバー アクセスに使用します。ドット演算子は、型または名前空間のメンバーを指定します。たとえば、ドット演算子を使用して、.NET Framework クラス ライブラリ内の特定のメソッドにアクセスします。

C#

```
// The class Console in namespace System:  
System.Console.WriteLine("hello");
```

たとえば、次のクラスを考えます。

C#

```
class Simple  
{  
    public int a;  
    public void b()  
    {  
    }  
}
```

C#

```
Simple s = new Simple();
```

変数 `s` には、`a` と `b` という 2 つのメンバーがあります。それらのメンバーにアクセスするためにドット演算子を使用します。

C#

```
s.a = 6;    // assign to field a;  
s.b();     // invoke member function b;
```

ドットは修飾名にも使用します。修飾名とは、属している名前空間やインターフェイスなどを示す名前のことです。

C#

```
// The class Console in namespace System:  
System.Console.WriteLine("hello");
```

`using` ディレクティブを使用すると、名前の修飾を省略できます。

C#

```
namespace ExampleNS
```

```

{
    using System;
    class C
    {
        void M()
        {
            System.Console.WriteLine("hello");
            Console.WriteLine("hello");    // Same as previous line.
        }
    }
}

```

ただし、識別子があいまいな場合は、修飾する必要があります。

C#

```

namespace Example2
{
    class Console
    {
        public static void WriteLine(string s){}
    }
}
namespace Example1
{
    using System;
    using Example2;
    class C
    {
        void M()
        {
            // Console.WriteLine("hello");    // Compiler error. Ambiguous reference.
            System.Console.WriteLine("hello"); //OK
            Example2.Console.WriteLine("hello"); //OK
        }
    }
}

```

<http://msdn.microsoft.com/ja-jp/library/vstudio/6zhxzbds.aspx>