



ウィンドウ表示とクラスの 関係を考察してみる

SDKでも、ここまでできる

初音 玲 HATSUNE, Akira

はじめに

Visual Basic.NETでは、IDE（統合開発環境）でフォーム上にコントロー

ルと呼ばれる部品を配置してゆくだけで、その配置した通りの画面を簡単に作れる。画面を表示するだけで、計算したり入力した情報を使わなくてもよいならば、開発言語のことをまったく

知らなくてもよいくらいだ。この特徴は、Visual BasicがWindows上の開発言語のデファクトスタンダード（事実上の標準）になるくらい成功した要因のひとつであるので、どれくらい便利なものであるかは周知の事実だと思う。

しかし、IDEでフォーム上にコントロールを配置したものが、どのように画面として表示されるかを理解している人は意外と少ないかもしれない。難しそうだと敬遠している人もいるだろう。でも、フォームとコントロールの関係は、クロスプログラミングを知るための身近な好例でもあるし、ひとつひとつ追ってみると、思ったほど難しくないのでこの機会に知識のひとつに加えてほしい。

本稿で前提となるもの

- O S Windows 2000 Professional (SP3) 以降
- 開発環境 Visual Studio.NET (IDE) V7.0.9486
- Visual Basic .NET Compiler 7.00.9466
- .NET Framework 1.0 Ver 1.0.3705 (SP2)
- Internet Explorer 6.0.2600.0000 (q319182; Q323759)

初級

中級

上級

本記事で解説したサンプルプログラムは、付録CD-ROMの¥DMAG¥F01_03ディレクトリ以下に収録しています

- ¥SIMPLE : 単純なフォーム
- ¥TXTCMD : 単純な入力フォーム
- ¥DYNACTRL : 動的にコントロールを作成する例
- ¥OPTSAMP : グループ化フォーム
- ¥SDKHELLO : SDKによるコンパイル
- ¥SDKSIMP : SDKによる単純なフォーム
- ¥SDK_IDE : SDKとIDEの両用ソース
- ¥IDE_SDK : 両用ソースのソリューション

単純なフォームを表示する

まずは、図1（サンプルSimple）のよ

図1：単純なフォーム

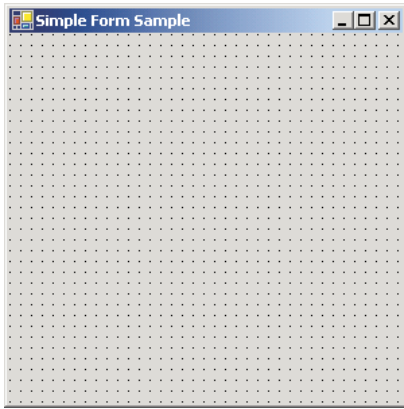


表1：サンプルSimpleのプロパティ値

プロパティ	値
Name	frmSimple
Text	Simple Form Sample

うにウィンドウのみの画面としたときに、どのように画面を表示しているのかを見てゆこう。ウィンドウのみの画面をIDE上で作成するのは簡単だ。IDEを起動して新規プロジェクトで「Windowsアプリケーション」を選択すれば、自動的にフォームがひとつ作成される。

自動的に作成された状態でもよいが、今回はプロパティウィンドウを使い、フォームに対して表1のような内容を設定している。なお、フォームのファイル名もfrmSimple.vbとした。

それでは、frmSimple.vbの内容を見よう（リスト1）。

リスト1：簡単なフォームの表示（サンプルSimpleより抜粋）

```
Public Class frmSimple
    Inherits System.Windows.Forms.Form

    #Region " Windows フォーム デザイナで生成されたコード "

    Public Sub New()
        MyBase.New()

        ' この呼び出しは Windows フォーム デザイナが必要です。
        InitializeComponent()

        ' InitializeComponent() 呼び出しの後に初期化を追加します。

    End Sub

    ' Form は dispose をオーバーライドしてコンポーネント一覧を消去します。
    Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub

    ' Windows フォーム デザイナが必要です。
    Private components As System.ComponentModel.IContainer

    ' メモ：以下のプロシージャは、Windows フォーム デザイナが必要です。
    ' Windows フォーム デザイナを使って変更してください。
    ' コード エディタは使用しないでください。
    <System.Diagnostics.DebuggerStepThrough> Private Sub InitializeComponent()
        '
        ' frmSimple
        '
        Me.AutoScaleBaseSize = New System.Drawing.Size(5, 12)
        Me.ClientSize = New System.Drawing.Size(292, 273)
        Me.Name = "frmSimple"
        Me.Text = "Simple Form Sample"

    End Sub

#End Region

End Class
```

- ① IDEで作成したフォームでは、1フォームが1クラスとなる。よって、frmSimpleフォームは、frmSimpleクラスとして定義する
- ② Inheritsを使ってFormクラスを基底クラスとして継承し、frmSimpleクラスが、Formとしての基本的な機能を備えるようにする。この行により、

- frmSimpleクラスのプロパティやメソッドは、Formのプロパティやメソッドとなる
- ③パブリックプロシージャNewは、frmSimpleクラスがオブジェクト化されるときに自動的に呼び出されるプロシージャだ。この中では、MyBase

.Newを実行して基底クラス（Formクラス）のNewメソッドを呼び出し、フォーム表示の基本的な処理を実行する。その後、⑥として定義したInitializeComponentサブプロシージャを呼び出し、プロパティウィンドウで設定した値を反映する