

将来にわたって使えるスキルを身につけるために

オブジェクト指向で はじめる プログラミング

日向 俊二
HYUGA, Shunji

第 **6** 回

ウィンドウアプリケーションを作る (その2)

Technology Tools

- Visual Basic .NET
- Visual C# .NET
- SQL Server 2000
- Oracle 9i
- Access 2002
- ASP.NET
- Internet Information Services
- Other:
 .NET Framework SDK

Level



Samples

・この記事で取り上げたソースコードおよびサンプルプログラムは、付録CD-ROMの¥DOTNET¥CSHARPディレクトリに収録しています。

¥EDITOR
今回のサンプルプログラム

※この記事では.NET Framework SDKを利用して解説していますが、C#Builder、SharpDevelopでも利用できます（C#コンパイラならすべて利用できます）。



はじめに

今回は、ウィンドウに文字を表示するきわめて単純なC#のプログラムを使って作ってみました。.NET Framework SDK+テキストエディタで作ったプログラムコードをよく検討すると、プログラムの仕組みや動作がより明確に見えてきます。今回はもう少し複雑で実用的なプログラムを作ってみるとともに、使えばとても便利なIDEをあえて使わずに.NET Framework SDK+テキストエディタでプログラムを制作することのメリットについてもよく考えてみましょう。



テキストエディタで テキストエディタを作る！

ここで作成するサンプルアプリケーションは、図1に示すテキストエディタです。このプログラムの[ファイル]メニューには、[新規作成] [開く] [保存] [名前を付けて保存] [終了]という、テキストエディタとして必要不可欠な一連のメニュー項目を作成します(図2)。さらに、[バージョン]というメニュー項目がある[ヘルプ]メニューを追加して、いわゆるアバウトボックス(バージョン情報ダイアログボックス)も表示できるようにします(図3)。このとき表示するアバウトボックスはきわめて単純なものですが、今回はあえて以下の2点を示すため、Formクラス

図1：テキストエディタアプリケーション

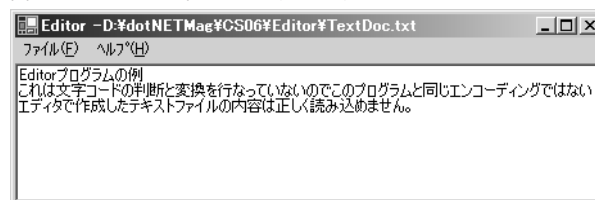
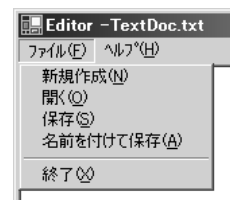


図2：テキストエディタの[ファイル]メニュー



オブジェクト指向で はじめる プログラミング

図3：[ヘルプ]メニューの[バージョン]を選択すると、アバウトボックスを表示



(フォーム) から派生させたクラスのオブジェクトとして作成します。

- IDEを使わなくてもフォームベースのダイアログボックスをメインウィンドウと同様に作成できる
- 複数のソースファイルで構成されるプログラムを容易に開発できる

まずはメインウィンドウ

メインウィンドウの作り方は、前回の「ウィンドウ版Hello C#」と基本的にはまったく同じです。

普通、ウィンドウはFormsクラスの派生クラスとして定義し、メインウィンドウにはプログラムの開始点(アプリケーションのメインエントリーポイント)である「Main()」が必要ですから、このサンプルのメインウィンドウのクラスの骨組みはリスト1のようになります。このリスト1のコードの各部分は前回説明したサンプルと同じ働きをします。

これで、実行時にはウィンドウとして表示される、「frmMain」という名前のフォームのクラスができ、プログラムが起動するとstatic void Main()の中の「Application.Run(new frmMain());」が自動的に実行されてウィンドウが表示されます。

ウィンドウの内容は? ～メニューとテキストボックスの作成

さて、骨組みはできましたが、これだけではウィンドウには何も表示されません。そこで、.NET FrameworkライブラリにあるGUIコンポーネントをウィンドウに配置してウィンドウの内容を構成しましょう。

このサンプルの場合、ウィンドウの中には、メニューコンポーネントとテキストボックスコンポーネントを配置したいので、使用するコンポーネントの型とその変数を宣言します(リスト2)。

最後の「System.ComponentModel.Container」は、あとでコンポーネント全体を一括して管理できるようにするための、コンポーネントの論理的なコンテナです。

前回のサンプルでは、コンポーネントの初期化を、メソッド「InitializeComponent()」の中で実行したので、このサンプルでも同様にしましょう。リスト3のように、テキストボックス(textBox1)、メインメニュー(mainMenu1)とメニュー項目(menuXxxx)のインス

リスト2：使用するコンポーネントの宣言 (Main.cs)

```
// 使用するコンポーネントの宣言
private System.Windows.Forms.TextBox textBox1;
private System.Windows.Forms.MainMenu mainMenu1;
private System.Windows.Forms.MenuItem menuFile;
private System.Windows.Forms.MenuItem menuFileNew;
private System.Windows.Forms.MenuItem menuFileOpen;
private System.Windows.Forms.MenuItem menuFileSave;
private System.Windows.Forms.MenuItem menuFileSaveAs;
private System.Windows.Forms.MenuItem menuFileSepa;
private System.Windows.Forms.MenuItem menuFileExit;
private System.Windows.Forms.MenuItem menuHelp;
private System.Windows.Forms.MenuItem menuAbout;

// コンポーネントのコンテナ
private System.ComponentModel.Container components = null;
```

リスト1：メインウィンドウの骨組み (Main.cs)

```
public class frmMain :
    System.Windows.Forms.Form
{
    // 使用するコンポーネントと変数の宣言
    (略)

    // コンストラクタ
    public frmMain()
    {
        // コンポーネントの初期化
        InitializeComponent();
    }

    // コンポーネントの初期化
    private void InitializeComponent()
    {
        // コンポーネントの初期化コード
    }

    // アプリケーションのメインエントリーポイント
    static void Main()
    {
        Application.Run(new frmMain());
    }
}
```

タンスを作成したあとで、それぞれのプロパティに値を設定します。とても複雑で面倒なように見えますが、それぞれの行の意味を一度理解してしまえば、コードを記述するのも理解するのも変更したり追加するのも簡単です。

コンポーネントをフォームに追加する際のポイントは、