

新

Java ユーザーのための C# 入門

乗り換えをもくろむあなたに

第 1 回 C# プログラミングの基礎知識

矢嶋 聡 YAJIMA, Satoshi
NRIラーニングネットワーク株式会社
MSDN Regional Director

はじめに

これから6回に渡って、C#プログラミングを使った基本的な実装方法について扱っていかうと思います。

主に対象者としてJavaプログラマの初級者を想定してはいますが、C#を説明するのが主なテーマでもあり、Javaプログラマでない方にも、役立つ内容にしました。Javaプログラマを含む初心者レベルの開発者全般を対象にした、C#プログラミングの記事であると考えてよいでしょう。

今回の連載では、C#を使って「何が作れるのか」「どう作るのか」という点

に重きを置いて、.NETにおけるC#の実装技術を中心に解説します。そのため、全6回の構成は「C#の基本構文→制御構文→オブジェクト構文」というような構文主体の構成にはせず、右のように実装技術ごとの構成にする予定です（もちろん、必要に応じてC#の構文説明も行ないます）。

また、今回の連載では開発ツールとしてVisual Studio.NETを使うことにします。無償で配布されている.NET Framework SDKを使って、コマンドプロンプトからC#プログラムをコンパイルすることもできますが、やはり企業レベルの本格的な開発となれば、Visual Studio.NETを使うことになるでしょう。

連載内容

- 第1回：C#プログラミングの基礎知識
- 第2回：コンソールアプリケーションを使った基本的なプログラミング
- 第3回：Windows フォーム
- 第4回：ADO.NET（データベースアクセス）
- 第5回：さまざまなソフトウェア部品（コンポーネントの開発など）
- 第6回：C#実装のまとめ+ASP.NET入門

う。

今回は「どう作るのか」という点については、「Visual Studio.NETでどう作るのか」を主に説明します。たとえば、第4回のADO.NETでは、単にADO.NETのクラスライブラリの解説だけではなく「Visual Studio.NETではこんな風に簡単にADO.NET対応プログラムが作成できます」というような、Visual Studio.NETにおける作り方も扱います。

ただし、Visual Studio.NETが手元に

本稿で前提となるもの

OS Windows 2000 Professional (SP3) 以降

開発環境 Visual Studio.NET
Visual C# .NET Compiler 7.00.9466
.NET Framework 1.0 Ver 1.0.3705 (SP2)
Internet Explorer 6.0

初級

中級

上級

ない方でも理解できるように、実際の Visual Studio.NET の操作画面も掲載し、Visual Studio.NET での生産性が高い開発手法を実感できるようにしました。また、無償の .NET Framework SDK でも対応できるように、できるだけ代替手段も提示しようと考えています。

それでは、まず第1回となる今回は、.NET 向けプログラムの特徴について見てゆきましょう。

今回のサンプル

第1回では、簡単なサンプルを題材に C# プログラムの特徴や、開発手順を確認してみます。このサンプルプログラムは、C# で記述した Windows フォームと呼ばれる種類のアプリケーションです。プログラムを実行すると、フォームと呼ばれるひとつのウィンドウが表示されます。ウィンドウには [OK] ボタンがついていて、このボタンをクリックすると、メッセージボックスが表示されます (図1)。

この C# プログラムのソースコードをリスト1に示します。コードの詳細は「第3回：Windows フォーム」で改めて説明しますが、今回は、リスト1を用いて C# の特徴を考えてみましょう。

C# とは何か

C# とは、もともと .NET の実行環境である .NET Framework 向けのアプリケーション開発のために、マイクロソフトによって考案されたプログラミング言語です。C# の特徴としては「C++ のオブジェクト指向言語としての特徴と、開発生産性が高い Visual Basic の特徴をあわせもつ」とよく言われます。

前述のサンプルプログラムにも、C++ の構文を受け継いだ特徴が見られ、括弧「{」から「}」までがひとつのブロックを表わしており、ひとつのステートメント (文) がセミコロンで終わるのも C++ と同様です。ただし、C# の文法は C++ に比べると洗練されており、冗長性のあると思われる部分は削除され、C++ よりも簡素化されています。また、

図1：今回作成するサンプル



C++ と同様にオブジェクト指向プログラミング言語であるので、オブジェクトを記述するために「クラス」を用いています。このサンプルプログラムも、ひとつのフォームというオブジェクトが登場します。そのフォームオブジェクトを表現している部分が、リスト1-①の「class Form ...」から、リスト1-⑥の「}」までであり、ひとつのまとまった「クラス」と呼ばれるブロックになっています。リスト1-①の「Form1」

リスト1：C# サンプルプログラム1 (Form1.cs)

```
using System;
using System.Windows.Forms;

// フォームを表わすクラス
public class Form1 : System.Windows.Forms.Form ①
{
    // ボタンコントロールを表わす変数
    private Button Btn1; ②
    // 初期処理
    public Form1() ③
    {
        Btn1 = new Button();
        Btn1.Text = "OK";
        Btn1.Click += new EventHandler(Button1_Click);

        Text = "Form1";
        Controls.Add(Btn1);
    }
    // Main メソッド
    static void Main(string [] args) ④
    {
        Application.Run( new Form1() );
    }
    // ボタンクリック時のイベントハンドラ
    private void Button1_Click(object s, EventArgs e) ⑤
    {
        MessageBox.Show("Hello, C#!");
    }
} ⑥
```