

2000年に.NETとC#に関する情報とプレビュー版が公開され、2002年には、Visual Studio .NET 2012製品群の1ラインナップに Visual C#が加わる形で正式リリースされました（この当時は、プログラミング言語ごとに製品が分かれていました。現在はそういう分け方はしていません）。

● C#と.NET

C#を説明するにあたって付随するのが.NETの存在です。C#の話に入る前に、.NETとの関係性について説明しておきます。

一般に、プログラミング言語で書いたプログラムを動かすためには、次のようなものが必要になります。

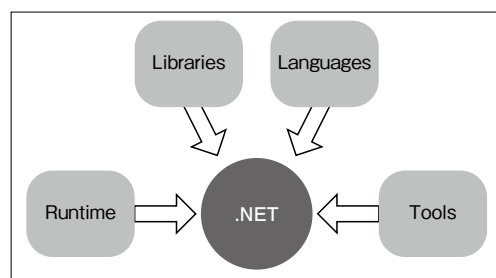
- 実行時環境 (Runtime)
- ライブラリ (Libraries)
- 言語コンパイラ (Languages)

また、プログラミングを補助するためのツール (Tools) も、ほぼ必須と言えます。図2に示すように、「.NET」というものはこれら Runtime + Libraries + Languages + Tools の4つを含むものです。

言語コンパイラ

C#はこのうちの一部、Languagesに含まれる言語コンパイラの1つです。

○図2 .NETの成り立ち



C#と.NETというレイヤーが分かれているのは、.NETという枠組みの中で、C#以外にもさまざまなプログラミング言語を実装して、相互にやり取りできるようにするためです。たとえば、VB（バージョン7以降）やF#などのプログラミング言語は.NET上で動き、C#と簡単に相互運用できます。.NETの目的の1つに、複数のプログラミング言語の間のギャップをなくすことも含まれています。

実行環境

プログラムを実行するのに最低限必要なものがあります。これを実行環境 (Runtime) と呼びます。

ほとんどのプログラミング言語で、まずメモリ管理などの機能が必須です。また、C#の場合は、一度IL (Intermediate Language) と呼ばれる形式にしたものを、実行時にネイティブコード化してプログラムを動かすので、その解釈プログラムも必要になります。このような、C#を動かすのに必要な実行環境は.NET Runtimeなどと呼ばれます。

ライブラリ

最近のプログラミング環境では、多くの機能を言語機能としてではなく、ライブラリとして提供しています。タッチやマウスなどのユーザーからの入力を受け付け、文字や絵として結果を出力するのも、ライブラリを通してです。

「一度書いたらどこでも動く」を実現するためには、実行環境の規格に加えて、標準で使えるライブラリが何かも定まっている必要があります。.NETの場合は、この実行環境+標準ライブラリの規格を指す言葉が.NET Framework（当初からあるWindows向けのもの）や.NET Core（2015年世代で

新たに作られたクロスプラットフォーム向けのもの）になります。

他のプログラミング言語では、これらをあまり区別しなかったりもします。たとえばJavaの場合、実行環境も標準ライブラリも言語コンパイラもすべてで「規格どおり」に実装し、認定を受けて初めてJavaを名乗れることになります。

● C# 1.0：最初のC#

.NETのような大掛かりなものを一から作っていたわけですから、それだけで相当な労力がかかったはずですが、最初のC#、つまり、C# 1.0は、リリースに必要な最低限に絞ったものと言えます。その結果、「既存のプログラミング言語をきれいにまとめたおしたのもの」という印象が強いです。

特に、出自が出自なので、J++に近いプログラミング言語でした。つまり、「COM相互運用とデリゲートを持ったJava拡張」から出発しています。

それに加えて、ユーザー定義の値型 (struct)、プロパティ、属性、foreachなどの機能が追加されています。主に、実行性能面への配慮と、Javaへの不満の解消となっています。

● 開発ツールとの連携

「C#といえばVisual Studio」というくらい、C#は開発ツールとの連携を考えることで生産性を高めています。

図3に示すように、C#はコードを書いた直後から常にコンパイルが行われ、リアルタイムにエラーや警告

を知ることができます。また、図4に示すように、今カーソルがある場所においてどういコードが書けるのか、補完候補を出してくれます。

リアルタイムな解析を行うために、C#はコンパイルの速度にも気を使って言語設計・実装されています。また、補完候補が出しやすいような語順で文法が決められていたりします。

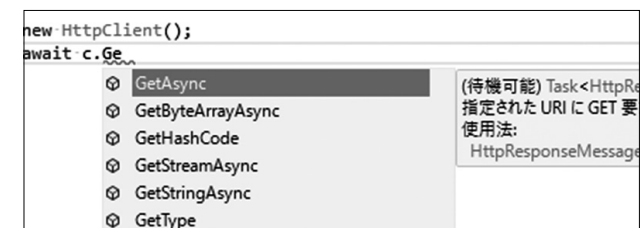
また、図5に示すように、Visual StudioにはGUIをデザインするためのデザイナーなども付属しています。

このGUIデザイナーのようなツールがC#コードを自動生成する場合があります(実際、

○図3 リアルタイムにエラーを検知



○図4 コードの補完



○図5 GUIのデザイナー

