

4

目的のオブジェクトを
探し出せ!

TypeFinderの使い方と拡張のためのヒント

豊田 孝 TOYOTA, Takashi

はじめに

私たちの周りではいろいろな変化が発生しています。一部の方は「とてもついていけない……」と弱気になっているかもしれません。本稿を最後まで

読めば、その弱気はかなり消えるはずです。.NET時代を生き抜くための、確かな学習方法を身に付けることができるからです。

私は、現在発生している大きな変化を次の2つに大別できると考えています。

- Visual Studio 6.0からVisual Studio .NETへの変化
- COMシステムエンジンから.NETランタイムエンジンへの変化

これは現象面から捉えた変化ですが、日々プログラムコードを忙しく書いているプログラマの視点に立つと、すべての変化には、次のような共通点があるといえるでしょう。

- 「関数とサブルーチンの世界」から「クラスの世界」への変化

本日は、Visual Studio.NET開発環境に付属している「TypeFinder」というユーティリティプログラムの使い方とその拡張を通して、.NET時代の主役である「クラス」について考えてみます。本論に入る前に、本稿の構成を示しておきます。

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 .NET Framework 1.0 Ver 1.0.3705 (SP2) Internet Explorer 6.0 Windows Scripting Host

初級

中級

上級

この記事で解説したサンプルプログラムは、付録CD-ROMの¥DMAG¥F01_04フォルダ以下に収録しています

¥FINDTYPEVB	: FindTypeの拡張版
¥WHATISTYPE	: 「WHATISTYPEPEEX.VBS」の.NET検証プロジェクト
• WHATISTYPE.VBS	: プロトタイプスクリプト
• WHATISTYPEPEEX.VBS	: 拡張スクリプト

- TypeFinderの概要
- Type (型) とは何か
- TypeFinderの使い方
- TypeFinder拡張上のヒント

ご覧のように「型」という用語が含まれているため、「もしかするとこの記事はオブジェクト指向に密接に関連しているのではないかと」、一部の読者は気が重くなっているかもしれません。そのような方に申し上げますが、本稿では抽象的な議論は一切行いません。COMから.NETへの変化を具体的なサンプルコードで示します。オブジェクト指向という角度から眺めれば、COMから.NETへの技術変化は難関ではなく、たいへん興味深い変化なのです。また、ここで誤解のないように述べておきますが、.NETの時代に入ったからといって、COMの時代が過ぎ去ったということでもありません。さらにいえば、COMの知識が不要になるということはまったくありません。詳細は別の機会に譲りますが、Visual Studio .NETの世界はCOMの発想が支えているといっても過言ではないのです。

それでは、本日の第1課題に入りましょう。

TypeFinderの概要

TypeFinderは、和訳すれば「型を見つけるプログラム」ということになるでしょう。問題は「型とは何か」ですが、それは次節で考えます。

このユーティリティプログラムは、Visual Studio.NETをデフォルトでイン

ストールすると、「C:\Program Files\Microsoft Visual Studio .NET\FrameworkSDK\Samples\Applications\TypeFinder」に格納されています。嬉しいことに、Visual Basic.NETとVisual C#.NET用の2種類の言語のソースコードが用意されています。関連するユーティリティとしては「C:\Program Files\Microsoft Visual Studio .NET\FrameworkSDK\Tool Developers Guide\Samples」に格納されている「meta-info」というサンプルもあります。こちらは、一部の方はご承知でしょうが、ILDASM.EXE (中間言語逆アセンブラ)機能の一部を紹介するものです。meta-infoはC++で記述されています。

ここであえて一言述べさせていただきますが、C#やC++という他のプログラミング言語名を目にするとひるんでしまうVisual Basicプログラマの方がいらっしゃるかもしれませんが、それはよくありません。Visual Basicは立派なプログラミング言語です。ひとつのプログラミング言語を知っていれば、他の言語を理解する基礎は十分もっているのです。後でC#とVisual Studio.NETを併用したサンプルをお見せしますが、十分理解できるはずですよ。

本論に戻しましょう。「TypeFinder」はすでにふれたように、「型を見つける」という機能を提供しています。しかし、

リスト1: プロトタイプスクリプト (WhatIsType.vbs)

```
Sub WhatIsType()
  Dim wss
  Set wss = CreateObject("WScript.Shell.1")
  wss.Popup "WhatIsType?", 30, "ShoeishaAndToyota"
  Set wss = Nothing
End Sub

WhatIsType()
```

型を見つけるとはいっても、型とはそもそも何なのか、なぜ型が用意されたのか、型を知るメリットとは何なのか、こうした疑問への回答が用意されない限り、みなさんも私も積極的に前に進むことなどできません。そこで、すでに枯れた技術となっているCOMベースの単純なVBScriptサンプルプログラムを紹介しながら、この型について考えてみます。みなさんの親しみ慣れたVBScriptサンプルから入りますから、肩の力を抜いて以降の文章をお読みください。

Type (型) とは何か

それでは、VBScriptサンプルプログラム (リスト1) を見ていただきましょう。

一部の方は、このサンプルプログラムと、私たちが今知りたいと考えている「型」の間にはいったいどのような関係があるだろう? と首をひねっていることでしょう。実は、多いに関係があるのです。このサンプルプログラムの一部をリスト2のように修正してみます。

修正後のサンプルプログラムを実行すると、図1のような画面が表示されます。

修正サンプルプログラムには次のよ



目的のオブジェクトを
探し出せ!

Visual Basic.NET