

開発者の要望に答える Visual Basic

●My / ジェネリック / Using / パーシャル / 演算子オーバーロード

大澤 文孝 OSAWA, Fumitaka

はじめに

Visual Basic 2005には、多彩な新機能 / 支援機能が追加され、効率良くプログラミングできます。

新機能には、「使っても使わなくてもよく、使えば便利になる」ものと、「仕

様が変わっており、今後は積極的に使うべきもの」の2種類があります。後者は、コーディングのスタイルを大きく変えるものであり、開発者は、必ず知っておかなければなりません。

そこで本稿では、この「今後は積極的に使うべきもの」に焦点を当て、

- ・ パーシャルタイプ
- ・ Myオブジェクト
- ・ Usingステートメント
- ・ ジェネリック
- ・ 演算子のオーバーロード

の意味と使い方を説明します。

自動生成されるコードを分離する「パーシャルタイプ」

まずコーディングするうえで知っておきたいのが、Visual Basic 2005では、

「自動生成されたコードのソースファイルが分離される」

という点です。

従来のVisual Studio .NET 2003では、たとえばフォームに新しいButton

コントロールを配置すると、そのButtonコントロールを配置するためのコードが、次々とソースファイルに書かれました。

しかしVisual Studio 2005の場合には、フォームにコントロールを置いたとしても、そのようなコードは書かれませんが、

正確には、書かれないのではなくて、「隠された別ファイル」に自動的に書かれるようになっているのです。

開発者が記したコードと自動生成されるコードとの関係

たとえばForm1.vbというフォームのソースを作ると、Form1.Designer.vbというファイルが自動生成されます。これが隠されたファイルです。

隠された別ファイルは、「ソリューションエクスプローラ」で、「すべてのファイルを表示」ボタンをクリックすると、表示されます(図1)。

フォームのプロパティを操作したり、フォームにコントロールを配置したりすると、このForm1.Designer.vbファイルは修正されますが、開発者から通常見えるForm1.vbのほうが変更されることはありません。

この機能を実現したのが、「パーシャ

Level

1 2 3 4 5

Technology Tools

- Visual Basic
- Visual C#
- Visual C++
- SQL Server
- Oracle
- Access
- Excel
- ASP.NET
- Other:
↓
Visual Studio 2005

図1：ソリューションエクスプローラで隠されたファイルを確認する



ルタイプ」です。

ソースコードには、次のような記述が含まれます。

Form1.vb

```
Public Class Form1
    ' 開発者はここにコードを書いていく
End Class
```

Form1.Designer.vb

```
Partial Public Class Form1
    Inherits System.Windows.Forms.Form
    ' 自動生成されるコードが書かれる
End Class
```

ここでの「Partial」がパーシャルタイプを示す特別なキーワードです。

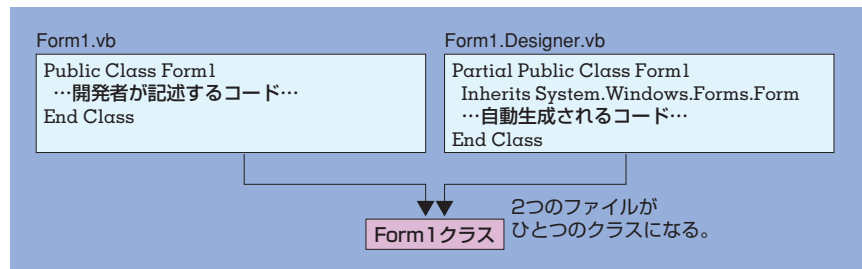
「Partial」を指定すると、分断された複数のファイルが、ひとつのクラスとしてコンパイルされます (図2)。

パーシャルタイプのメリットと注意点

このように、Visual Studio 2005では、パーシャルタイプによって、「開発者が記述するコード」と「Visual Studio 2005が自動生成するコード」が別ファイルとなり、相互に混じり合わなくなりました。

これにより、開発者が記述するソー

図2：パーシャルタイプの役割



スファイルは、Visual Studio 2005によって勝手に書き換えられることがないので、ソースが見やすく、また、トラブルの原因も減ります。

その反面、「隠されたファイルに自動生成されたコードが書かれている」ということを知らないと、「自分が記述していないはずの動作をする」と悩むこととなります。

もしそのような現象に悩んだなら、ソリューションエクスプローラで、「すべてのファイルを表示」にして、自動生成されたコードを確認してください^[註1]。

ちなみにパーシャルタイプは、開発者が明示的に使うこともできます。

たとえばひとつのクラスを3つとか4つとかのソースファイルにわたる目的でパーシャルタイプを使ってもかまいません。そうすると、複数の開発者でチーム開発するときに、「ある人はここからここまでを担当」「別の人はここからここまでを担当」といったことができるようになります。

しかしそのようなパーシャルタイプの使い方は、推奨されません。

なぜなら、複数のソースファイルに分割すると、「いったいそのクラスには、

どのようなプロパティやメソッドが実装されているのか」という全体像がわかりにくくなってしまいうからです。

Partialを使う場面は、まちまちですが、少なくとも、「大きなクラスを複数のファイルにわけて、最終的にひとつのクラスとして構成する」ということは、どこかで設計が間違っています。もしそうなってしまったら、パーシャルタイプを使うべきものではなく、クラス自身を分離すべき問題です。

このような理由から、特別な意図がない限り、「パーシャルタイプは自動生成されるコードのためのものであり、開発者が明示的に使うことはない」としておくのが無難です。

ファイルやレジストリ、リソースにアクセスできる「Myオブジェクト」

Visual Basic 2005で追加された大きな拡張機能が「Myオブジェクト」です。

Myオブジェクトは、アプリケーションやフォーム、設定情報、そして、リソースなどへアクセスするための7つのオブジェクトで構成されています(表1)^[註2]。

注1) あくまでも中身の確認に留め、自動生成されたファイルを修正するのは避けてください。自動生成されたファイルは、修正しても、上書きされる恐れがあります。

注2) ASP.NETアプリケーションの場合には、表1に示した以外にも、「My.Log」「My.Request」「My.Response」の3つのオブジェクトを使うことができます。