

西田雅昭
の

新

Visual Basic!

西田 雅昭
NISHIDA, Masaaki

基本からの
業務アプリケーション構築技法

「ボ」

第5回 オブジェクトとクラス



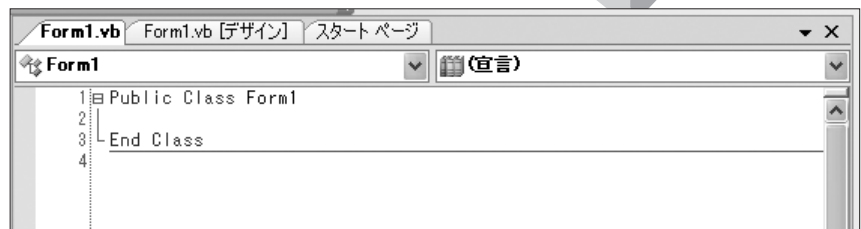
オブジェクトとは

Visual Basic 2005 (以下VB2005) で Windows アプリケーションのプロジェクトを作成した直後にコードエディタを開くと、

```
Public Class Form1
```

```
End Class
```

図1: 「Windows アプリケーション」プロジェクトのコードエディタ



となっています(図1)。そして、この中にいろいろなコードを記述していきます。これは、「Form1」というクラスを作成(定義)していることになりま

す。VB2005のプログラミングとは、独自のクラスを定義することなのです。

ですから、VB2005のプログラミングを理解するためには、クラスとオブジェクトを理解しなければなりません。

オブジェクトという言葉を知っただけで、いやな顔をする方も少なくないでしょうが、少し我慢して読み進めてみてください。

一般に「オブジェクト(Object)」という何を連想するでしょうか。思えば中学、高校と苦しめられた英文法では、Objectは「目的語」でした。Objectは「動詞が働きかける対象」と理解し

レベル >>> Level

1

2

3

4

5

サンプル >>> Sample

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、<http://www.shoeisha.com/mag/windev/>からダウンロード可能です。

言語 >>> Language

▪ Visual Basic

ツール >>> Tool

▪ Visual Studio 2005 Professional

*) 本連載ではVisual Studio 2005 Professionalを使用していますが、解説内容は特に明記しない限り、Visual Basic 2005 Expressでも利用できます。Express Editionをお持ちの方もぜひ挑戦してみてください。



ていたのを思い出します。また、美術館に行くと、彫刻によく「オブジェ (objets)」という名前が付いています。Objectをフランス語でobjetsと言い、「物」や「対象」という意味になります。

そこで、あまり難しく考えず、オブジェクトとは単純に“物”である、と思ひましょう。たとえば、「戦争」や「平和」などの抽象的な概念も、「コンピュータ」や「自転車」などの具体的な物体も、みなオブジェクトなのです。

手続きとオブジェクト

では、プログラミングを行なううえでの「オブジェクト」とは何のことなのでしょう。

プログラミングの世界で、「オブジェクト」と対照的な言葉は「手続き」です。プログラムを作成する際には、いろいろな「処理」を書いて、それを組み合わせていきます。サブルーチンやプロシージャという言葉はすでにご存知でしょう。

例として、以下のような直線を引くプロシージャを考えてみましょう。

```
Sub WriteLine(_
  開始点X, 開始点Y, 方向, 長さ, 色)
```

このプロシージャを呼び出す際には、

```
Call WriteLine(12, 16, 190, 32, Red)
```

のように記述します。

このようなプログラミングの方法を「手続き指向プログラミング (Procedure Oriented Programming)」と呼びます。

このようなプログラミングでは、「処理」と「データ」を別のものとして捉

えていました。しかしプログラムが大きくなると、処理の内容は複雑になり、全体構造を理解するのが困難になります。

そこで、小さく独立した手続きを構造的に組み合わせて大きな処理を作ろう、という考え方が生まれました。ひとつひとつの手続きが、別の手続きと独立していれば、処理を理解するのが楽になります。このような考え方を「構造化プログラミング (structured Programming)」と呼びます。

先ほどの例では、プロシージャ「WriteLine」の内部がどのようなになっているかを知らなくても、複雑な図形が描けます。しかし、もし複雑な図形を描くとすると、大量の「Call WriteLine」を記述することになり、どれがどの線なのかわかりにくくなります。

さらにシステムが大きくなると、「処理」と「データ」の組み合わせが煩雑になり、「処理」中心の考え方では、「データ」の流れを理解するのが難しくなります。しかも、ちょっとしたデータ処理のミスがシステム全体に影響するので、大きなシステムを構築するのは非常に難しいでしょう。皆さんの中にも、変数のおっかけっこで苦労したという経験がある方もいると思います。

そこで、処理の対象を「オブジェクト (物)」として捉える、もう少し詳しく言うと、「データ」を中心として「手続き」をその回りに配置してひとつのパック (オブジェクト) とする、という考え方が生まれました。これが「オブジェクト指向プログラミング (Object Oriented Programming)」と呼ばれるものです。データとそれに対する処理が一体となっているので、理解しやす

いのです。

先ほどの直線の例を考えてみましょう。まず「Line」という名前の直線のクラスを定義します。

```
Class Line
```

```
'5つのプロパティを定義
  開始点X
  開始点Y
  方向
  長さ
  色
```

```
'2つのメソッドを定義
  描く
  消す
```

```
End Class
```

そして、三角形を描く場合は、

```
Dim 高さ As New Line(0, 0, 90, 10, Black)
Dim 底辺 As New Line(...)
Dim 斜辺 As New Line(...)
```

のようにオブジェクトを宣言しておいて、

```
高さ.描く
```

のように記述するのです。もし「高さ」の色を変えたいときは、

```
高さ.色 = Red
```

のように、簡単に記述できます。

この方法に、どのような利点があるのかについては、おいおい説明していきます。