

将来にわたって使えるスキルを身につけるために

オブジェクト指向で はじめる プログラミング

日向 俊二
HYUGA, Shunji

第 3 回

C#のメソッドとコンストラクタ

Technology Tools

- Visual Basic .NET
- Visual C# .NET
- SQL Server 2000
- Oracle 9i
- Access 2002
- ASP.NET
- Internet Information Services
- Other:
.NET Framework SDK

Level



Samples

・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、付録CD-ROMの¥DOTNET¥CSHARPディレクトリに収録しています。

¥INTEGRAL
今回作成したサンプル

※この記事では.NET Framework SDKを利用して解説していますが、C#Builder、SharpDevelopでも利用できます（C#コンパイラならすべて利用できます）。



はじめに



今回は、文字、空白、行、コメントなど、C#のプログラムの書き方における、もっとも基本的な決まりについて説明しました。基本的な決まりがわかったら、次に、ステートメントや関数をいくつか覚えてプログラムを作ってみるというのが、プログラミングをマスターする従来の標準的な方法でした。

C#でも、構文をいくつか覚え、.NET Frameworkを使って、自分がいちばん興味をもてる分野のプログラムを作ってみるというのはとても良い方法です。ただし、C#のプログラミングには、従来の非オブジェクト指向の言語を使ったプログラミングとは違って、特に意識する必要がある点がひとつだけあります。それは、

オブジェクトのことを常に意識してプログラムを作る

ということです。これを心がけるだけ

で、オブジェクト指向プログラミングの特性を容易に理解できるようになり、その利点を生かしてよいプログラムを楽に開発できるようになります。

プログラムを作るうえで最初に理解しておきたいことは、「オブジェクトのもととなるクラスの作り方」です。今回は、そのうち、とくにコンストラクタとメソッドに焦点を当てます。



オブジェクトの 作成と操作



はじめに、オブジェクトの作成方法と使い方を復習しておきましょう。

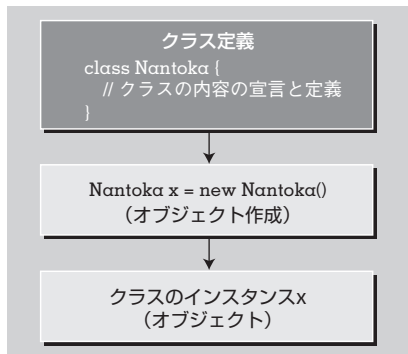
◆インスタンスの作成

クラスのインスタンスであるオブジェクトは、通常、演算子newを使って、クラス名と同じ名前のメソッドを呼び出して作ります（図1）。

```
クラス名 オブジェクトの変数
= new クラス名();
```

オブジェクト指向で はじめる プログラミング

図1：クラスとオブジェクト



newのあとのクラス名と同じ名前のメソッドを「コンストラクタ」といいます【*1】。

◆メソッドやフィールドの呼び出し

いったんインスタンスを作ったら、次の形式で、クラスのメソッド（サブルーチン、プロシージャ）を呼び出したり、公開されているクラスのフィールドにアクセスしたりできます。

オブジェクトの変数.メソッド名();
オブジェクトの変数.フィールド名;

★1 プログラマや研究者の中には、コンストラクタを一種のメソッドとみなさなかったり、「コンストラクタを“呼び出す”」という表現を避ける人たちもいます。しかし、ここではMicrosoftのドキュメントに準拠して、厳密さよりもわかりやすさを優先した表現を使います。

メソッドやフィールドがstatic（静的）で宣言されている場合は、次の形式でアクセスできます（staticについては後述します）。

クラス名.メソッド名();
クラス名.フィールド名;

このように、通常のメソッドは、「オブジェクトの変数名またはクラス名」＋「ピリオド(.)」＋「メソッド名()」または「フィールド名」という組み合わせだけで使うことができます。

これに対し、コンストラクタを使ってオブジェクトを作るときは、new演算子を使わなければなりません。そして、new演算子のあとに指定できるのは、クラス名と同じ名前のコンストラクタだけです。

上記のように、書式をみてみただけでも、メソッドとコンストラクタは異なっています。以降で、この意味や、「メソッドを呼び出すこと」と、「コンス

A T T E N T I O N !

C#のコンストラクタには、3種類の異なる性質のコンストラクタがあります。しかし、通常、もっとも頻繁に使われるのは、newと共に使ってインスタンスを作成する「インスタンスコンストラクタ」です。この連載では特に明示した場合を除いて、「コンストラクタ」という言葉はインスタンスコンストラクタを指します。コンストラクタの種類についてはコラム「C#の3種類のコンストラクタ」を参照してください。

トラクタを使ってオブジェクトを作ること」の違いについて、詳しく考えてみましょう。



今回作成するクラス

ここでは、ひとつの例として、整数を表現するクラス「IntegralValue」を作って使ってみることにします。

実は、C#には整数を表わすプリミテ

C#の3種類のコンストラクタ

C#には以下のコンストラクタがあります。

インスタンスコンストラクタ

もっともよく使われるコンストラクタです。これはクラスのインスタンスを作成して初期化するために使います。後掲のIntegralValueクラスに定義したコンストラクタは、いずれもインスタンスコンストラクタです。

プライベートコンストラクタ

クラスの内部からアクセスできる特別な種類のインスタンスコンストラクタを作ることができます。これを「プライベート（private）コンストラクタ」といいます。このコンストラクタはクラスの外部からはアクセスできません。注意すべき点は、プライ

ベートコンストラクタを宣言したクラスは「インスタンスを作成できない」ということです。したがって、このコンストラクタは、後述のstatic（静的）メンバだけを使う、または特別なメソッドでオブジェクトを作成してその参照を返すような特別なクラスに限って定義することがあります

静的コンストラクタ

最初のインスタンスが作成される前、またはstatic（静的）メンバが参照される前に自動的に呼び出されて初期化を行なう「静的（static）コンストラクタ」というものもあります。このコンストラクタは、インスタンスを作成し、かつstatic（静的）メンバも使うようなクラスで定義することがあります。



ちょっと一服