SKILL UP!

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7回

C#でXML文書を扱う（その1）

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。

¥VALIDXML
今回のサンプルプログラム

※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、XML文書を扱うプログラムをC#で作成してみましょう。C#のプログラムとXML文書には、とても重要な概念で共通することがあります。今回の記事は、その共通する概念に焦点を当てて、C#のプログラムとXML文書の両方を理解してしまおうという、とてもオイシイ内容です。

XML基礎の基礎

最初に、この記事を読むにあたって最低限必要なXML文書（XMLドキュメント、XML Document）の基礎のそのまた基礎を押さえておきましょう。

XML（eXtensible Markup Language）は、日本語で「拡張可能なマーク付け言語」の意味ですが、どのように拡張可能なのか、というような難しいことはさておいて、XML文書の姿を眺めてみましょう[★1]。

XML文書はタグ付きテキストとして記述します。それぞれのタグ付きテキストは要素ともいいます（図1）。

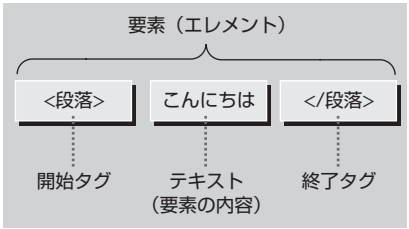
図1の要素を含む、最も単純なXML文書の例を次に示します。

```
<?xml version="1.0"?>
<段落>こんにちは</段落>
```

最初の行は、これがXMLバージョン1.0の仕様に従ったXML文書であることを示しています。XML文書の先頭には、このようなXML文書であることを示す情報を記述します。この情報をXML宣言といいます。

次の行の「<段落>こんにちは</段落>」は、まさに図1を実際のXML文書に記述した例です。これは完璧とはいえないものの、これもXML文書です。

図1：単純なタグ付きテキスト



★1　XMLのやや難しい考え方をやさしく知りたいという場合は、記事末の参考リソースを参照してください。
★2　厳密にいえば、プログラミングでは普通はクラスのインスタンス、つまり「オブジェクト」を使います。しかし、インスタンスを作成しないで使う場合もありますから、ここでは初心者にもわかりやすいように「クラス」という言葉を使います。
★3　厳密には、XMLの場合も、要素をクラスと単純に言い換えるのは必ずしも正確であるとはいえませんが、ここではわかりやすいように「クラス」という言葉を使います。なお、XMLとその関連仕様（英語の原文）の中でもクラス（Class）という言葉がよく使われています。

続いてもう少し複雑なXML文書の例を次に示します。

```
<?xml version="1.0"?>
<楽器リスト>
  <楽器>ピアノ</楽器>
  <楽器>チェロ</楽器>
</楽器リスト>
```

これは、<楽器リスト>という要素の中に<楽器>という要素が2個ある例です。さらに複雑な例はあとで示しますが、XML文書では、ある要素（親要素）に属する要素（子要素）は、親の開始タグと終了タグではさまれた中に記述しなければならないという点に注意してください。

さらにいえば、XML文書には次のような決まりがあります。

- ルート要素（この場合は<楽器リスト>）は1個でなければならない
- 開始タグに対応する終了タグが必要（<タグ/>の形式の空の要素は例外）
- 開始タグに対応する終了タグが終わる前に、その要素の子要素でないほかの要素の開始タグを記述すること（終了タグをまたぐこと）はできない

これらは、XML文書の最も重要な規則です。そして、このような規則があるから、XML文書やその中の要素をオブジェクトとして扱うことができるのですが、そのことをC#のプログラムと比べながら考えてみましょう。

C#とXMLの微妙なカンケイ

C#は「オブジェクト指向のプログラミング言語」といわれています。Visual BasicもVisual Basic .NETになったら、オブジェクト指向になりました。そして、いまやプログラミングに不可欠なXMLも、オブジェクト指向しているそうです。

どうしてこのように何もかもがオブジェクト指向になったのか、ということはあとで考えることにして、ここでC#プログラムとXML文書の構造について少し考えてみましょう（図2）。

C#のプログラムは、たとえば、コンソールアプリケーションの場合、アプリケーションのクラスがあって、その中にmainがあり、そのクラスの中でデータや入出力のクラスを使います★2。アプリケーションのクラスの中で使うクラスのオブジェクトは、アプリケーションの中の要素です。

図2：C#プログラムとXML文書（概念図）

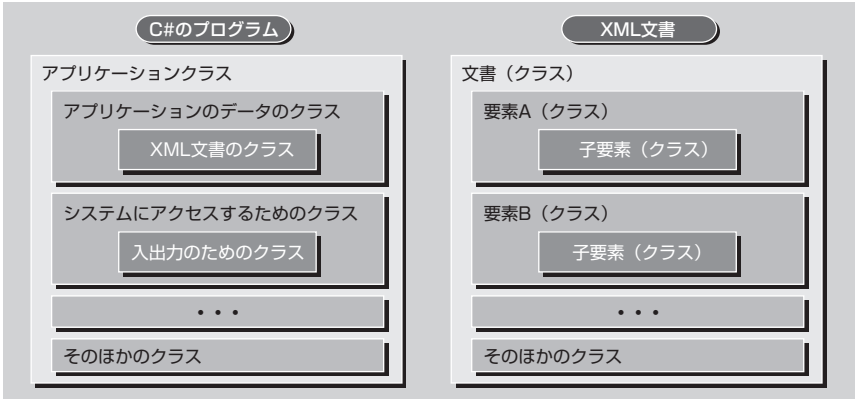
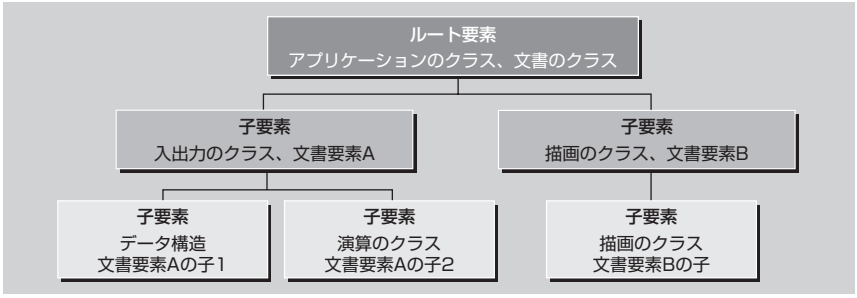


図3：ツリー状の階層構造



一方、XML文書も、ルート要素である文書クラスがあって、その中に子要素があります★3。

別のいいかたをするなら、C#のプログラムもXML文書も、ツリー状の階層構造を持ちます（図3）。

ところが、従来のプログラムや文書は、必ずしもツリー状の階層構造ではありません。たとえば、Goto文でジャンプするようなプログラムの構造はツリー状の階層構造としては表現できず、フローチャートで表現します（図4）。また、HTML文書の場合は、終了タグがなかったり、あるタグとその終了タグをまたいでほかのタグを書いても、たいていエラーになりませんが、これもツリー状の構造として表現することはできません（図5）。

これでわかることは、HTMLや従来