完全掌握 SQL Server プログラミング再入門

DBアプリケーションなんてカンタンだ！

第6回 データセットを使いやすくしてみよう

株式会社システムインテグレータ
湯尾 守 *YUO, Mamoru*
http://www.sint.co.jp/

Technology Tools

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

Level

★★½☆☆

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¥SQLServerディレクトリに収録しています。

¥ORIGINAL
今回作成したクラス（CompanyDataSet、DepartmentsTable、DepartmentRow）

¥VSSOURCE
VS.NETで自動生成したDataSetクラス

¥SQL
テーブルとサンプルデータを作成するSQLスクリプトの収められたファイル

はじめに

DataTableオブジェクトは列を定義するときにその列の型を指定して定義します。しかし、これらの型のチェックは実行時に行なわれるため、整数型の列に文字列をセットするコードを記述をしてもコンパイル時にこのエラーを発見できません。また、DataRow内の整数型の列のデータを取得するときも取得時のデータはすべてObject型なので、取得後に整数型にキャストしなくては使用できません。

これではせっかく型を指定して列を作成してもそのありがたみはあまりないと言えます。そこで今回のテーマは、

型指定のあるデータセットを作成する

です。

データのカスタマイズと継承

オブジェクト指向言語

そもそもオブジェクト指向とは「データ中心」の考え方です。データを関数に渡して操作するのではなく、“データ自身に操作を依頼する”というのがオブジェクト指向的な考え方です。データ自身が何をすべきか知っていなくてはこのような発想でプログラミングをすることはできないので、データ自身が状況に応じた振る舞いをすべて知っている必要があります。このデータのことを「オブジェクト」と呼んでいます。クラスはそのデータの内容とデータ操作の手法が書かれたオブジェクト設計図です。

ところが、ライブラリで提供されているクラスに、各アプリケーションに必要な動作についてすべて記述されているかというともちろん答えは否です。どのようなアプリケー

ションが作成されるかわからないのにそのような便利なライブラリを作成できるはずがありません。つまり、アプリケーションを作成するときにライブラリのクラスをそのまま使用することは基本的にはできないのです。

しかし、イチからすべてのデータの中身と振る舞いを記述するのは面倒です。そこで、オブジェクト指向言語ではライブラリとして、よくある定番といえるデータ（クラス）を提供し、このクラスを基にして自分の作成するアプリケーションに合わせたデータの内容や振る舞いを定義するといった手法をとります。

DataTableクラスのカスタマイズ

前回までDataSetオブジェクトやDataTableオブジェクトをそのまま使用してきましたが、今回は今まで作ってきたアプリケーション専用のオブジェクトを作成してみましょう。まずは部門データ（表1）を例にとって考えてゆきましょう。

表1：「Departments」テーブルの定義

列名	データ型	長さ	備考
DepartmentID	nchar	3	主キー
DepartmentName	nvarchar	50	Null を許容しない
Prefectures	nvarchar	10	
City	nvarchar	20	

部門のデータはリレーショナルデータベースから取得するのでテーブルと同じデータ構造を持つDataTableクラスを継承して作成します。テーブルには「部門ID」「部門名」「都道府県」「市区町村」という列があります。それを記述するとリスト1のようになります。

各列を表わすDataColumnオブジェクトを読み取り専用のプロパティとして用意しておくのがポイントです。このプロパティはいたるところで使用されます。

リスト1：DepartmentsTableクラス

```
Public Class DepartmentsTable
  Inherits DataTable

  Private columnDepartmentID As DataColumn
  Private columnDepartmentName As DataColumn
  Private columnPrefectures As DataColumn
  Private columnCity As DataColumn

  Public Sub New()
    MyBase.New("Departments")
    ' 列の生成
    columnDepartmentID = _
      New DataColumn("DepartmentID", GetType(String))
    Columns.Add(columnDepartmentID)
    columnDepartmentName = _
      New DataColumn("DepartmentName", GetType(String))
    Columns.Add(columnDepartmentName)
    columnPrefectures = _
      New DataColumn("Prefectures", GetType(String))
    Columns.Add(columnPrefectures)
    columnCity = New DataColumn("City", GetType(String))
    Columns.Add(columnCity)
    ' 主キーの設定
    PrimaryKey = New DataColumn() {columnDepartmentID}
    ' サイズの指定
    columnDepartmentID.MaxLength = 3
    columnDepartmentName.MaxLength = 50
    columnPrefectures.MaxLength = 10
    columnCity.MaxLength = 20
  End Sub

  ' 列「部門ID」
  Friend ReadOnly Property DepartmentIDColumn() As DataColumn
    Get
      Return columnDepartmentID
    End Get
  End Property

  ' 列「部門名」
  Friend ReadOnly Property DepartmentNameColumn() As DataColumn
    Get
      Return columnDepartmentName
    End Get
  End Property

  ' 列「都道府県」
  Friend ReadOnly Property PrefecturesColumn() As DataColumn
    Get
      Return columnPrefectures
    End Get
  End Property

  ' 列「市区町村」
  Friend ReadOnly Property CityColumn() As DataColumn
    Get
      Return columnCity
    End Get
  End Property
End Class
```