

# Windows プログラミング

第 5 回

## ADO.NET 2.0 –その2–

こだか かおる  
KODAKA, Kaoru

### Level

1 2 3 4 5

### Technology Tools

- Visual Basic
- Visual C#
- Visual C++
- SQL Server
- Oracle
- Access
- ASP.NET
- Other:

Visual Studio 2005 ベータ2  
SQL Server 2005  
Express Edition April CTP

### Samples

### はじめに

新しい開発ツールが登場すると、ついつい、その進化のインパクトに身構えてしまうものです。覚えなければならないことは増えますし、今まで身につけた技術が無駄になってしまうのではないかと、という恐れもあります。とはいえ、「新しいもの」が「過去のもの」の改良である以上、その間には何らかのつながりがあるものです。逆に言えば、過去のツールの問題点を理解しているからこそ、新しいツールの恩恵がより理解しやすいのかもしれませんが。

過去の資産への互換性という意味合いもあって、新しいツールでありながら、古い手法をサポートすることも少なくないでしょう。しかし、その古い手法にこだわり続けるのは得策ではありません。 .NET Frameworkにおけるデータベース関連の機能は、DAO、ODBC、RDO、ADOと脈々と受け継がれて来た部分と、最新のテクノロジーから成り立っています。たしかに、「××という

新機能を使わなくても、○○機能を使えばいいや」というのもひとつの判断です。とはいえ、新しい機能には新しい機能なりの存在理由があるはず。その理由を確認しつつ、どちらを使っていくのか、じっくり検討してみることをお勧めします。

### データセット? O/Rマッピング?

今回も、前回に引き続きデータアクセス関連、すなわちADO.NET 2.0に関する機能について見ていくことにしましょう。ADO.NET 2.0が具体的な形を現わす以前、ひとつのキーワードが目を集めていました。そのキーワードは「ObjectSpace」です。いわゆるO/Rマッピングツールですが、結局ADO.NET 2.0への搭載は見送られました。O/Rマッピングは、実際にデータを格納するデータベースとオブジェクトをつなぐ仕組みです。

一般的に用いられているデータベースは、リレーショナルデータベースで

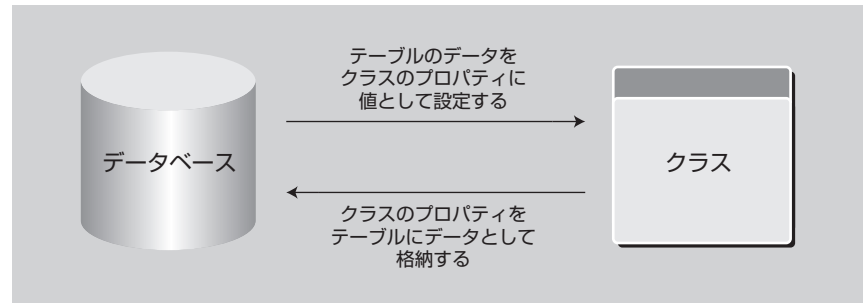
す。列と行からなるテーブルがあり、テーブルは、他のテーブルとキーによって関係（リレーション）付けられます。この仕組みにより、重複するデータが最小限に抑えられるのが、リレーショナルデータベースの特徴です。また、データを操作する際は、INSERT、UPDATE、DELETE、SELECTといったSQLを利用します。

これに対し、オブジェクト指向プログラミングで利用されるオブジェクトは、プロパティやメソッドといったものを持っています。データを取り扱うクラス、たとえば「カスタマー」といったオブジェクトを考えた場合、カスタマーは「顧客ID」「顧客名」といったプロパティを持ちます。当然、それらの値がデータベースに格納されるデータとなります。

オブジェクト指向プログラミングの観点から言えば、オブジェクトそのものをデータストアに格納できるのがベストな形でしょう。そのような方向性で考えられたODBMS（Object Database Management System）、もしくは、リレーショナルな要素も取り込んだ ORDBMS（Object Relational Database System）といったものも検討されていますが、まだあまり一般的なものにはなっていません。

オブジェクトをそのままデータストアに格納することができない以上、データストアから取り出したデータをオブジェクトに設定する、もしくは、オブジェクトのプロパティをデータストアに格納できる形式に変換する、といった処理が必要になります。これが、O/Rマッピング（図1）で行なわれるこ

図1：O/Rマッピング



とです。しかし、.NET環境について見てみれば、O/Rマッピングはあまり使われていません。海外のサイトを巡ってみると、いくつかのO/Rマッピングツールが公開されていますが、少なくとも日本では、.NET Frameworkで用意されているデータセットを使うのが一般的のようです。

それにはさまざまな理由があるのですが、たとえば、O/Rマッピング時に発生するパフォーマンス上のコスト増が考えられます。いったん取得したデータをオブジェクトに設定しなおす。このような処理には、当然のことながらそれなりの時間がかかるからです。また、データセットを使えば、さほどRDBMSを意識しなくても済むということもあるでしょう。データアダプタの設定さえ済んでしまえば、あとはFill()メソッド、Update()メソッドでデータの取得、格納が行なえます。

ということで、O/Rマッピングについて検討してきましたが、少なくとも現時点では「データセットを使う」というのが、Microsoftの指し示すADO.NET 2.0での方向性です。このあたりは、今後ObjectSpaceが.NET Framework上で提供されるようになれば、また変わっていくことでしょう。あるい

は、ADO.NET 2.0でもデファクトスタンダードなO/Rマッピングツールが登場すれば、一気に大きな変化が訪れるかもしれません。機会があれば、O/Rマッピングツールの比較などもやってみたいところです。

## レイヤー

ADO.NET 2.0では、データを扱う際にデータセットを使う。これをとりあえずの結論として、話を先に進めることにします。そのデータセットの使いどころですが、アプリケーションの層（レイヤー）から見れば、すべてのレイヤーが対象になります（図2）。データベースに近い位置から見て、「データストア」「ビジネスロジック」「ユーザーイ

図2：アプリケーションのレイヤー

