

# 天気晴朗 なれど 波高し

西田 雅昭  
NISHIDA, Masaaki



Illustration : Toshiyuki Ido

Visual Basic .NET 奮戦記

第10回

データベース処理-その4-

Level

- 1
- 2
- 3
- 4
- 5

Technology Tools

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

↓  
Visual Studio .NET 2003

我輩の名前は「頑固一徹」。Visual Basic一筋のベテランプログラマーである。ADO.NETのデータベース処理の実態を理解するため、我輩は接続型の処理(直接処理)を勉強することにした。今回は、その勉強の成果として、「ExecuteReader」の使い方や効率的なデータ取得方法、コマンドパラメータ、エラー処理について紹介し、最後に除算に関するちょっとした宿題を出した。今回はまず、この宿題について考えてみよう。



前回の宿題

前回の拙稿をお読みでない方のために、まずはどんな宿題を出したかをお見せしよう。

この除算プログラムをリスト1に再掲しておく(説明の都合上、変数名をプレフィックスを付けない形に変更している)。この宿題を解くヒントは除算に

Homework

リスト5の除算のプログラムで、被序数のデータ型を「Single」にすると、どうなるだろうか。さらに、答えも「Single」にすると、面白いことが起きるはずである。この理由と対処法は?

リスト1: 除算プログラム

```
Private Sub btnCalc_Click( _
    ByVal sender As System.Object, _
    ByVal e As System.EventArgs _
) Handles btnCalc.Click
    Dim x As Integer = 10
    Dim y As Integer = 0
    Dim Answer As Integer

    Try
        Answer = x ÷ y
```

```
Catch ex As DivideByZeroException
    MessageBox.Show(ex.Message)
Catch ex As Exception
    MessageBox.Show(ex.Message)
Finally
    MessageBox.Show( _
        "演算と結果のチェックを終了しました")
End Try
End Sub
```

図1：除算できないことを示すエラーメッセージ



関する例外処理にある。

今回から読み始めた方は、フォームを作成してボタンをひとつ配置し、このイベントプロシージャを記述してみたい。

いうまでもないことだが、このプログラムを実行しボタンをクリックすると図1のようなメッセージボックスが開く。



### 除算でのデータ型について考える

この除算に使われている被除数 (y) や答え (Answer) のデータ型を変更するとどうなるのだろうか。

### 浮動小数点型の計算

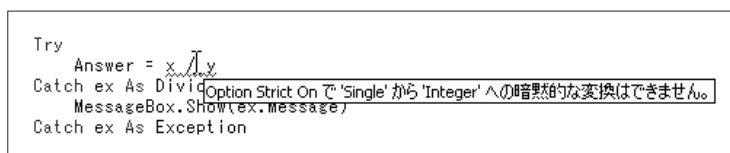
まず被除数を浮動小数点型 (Single) にしてみよう。

```
Dim y As Single = 0
```

これを実行しようとして、ビルドすると「Option Strict Onで'Single'から'Long'への暗黙的な変換はできません」というビルドエラーが出る。

「¥演算子」は、整数型の数値を整数型の数値で除算する場合にしか使うことができないのである。しかし、このエラーメッセージは、なぜ「'Long'への暗黙的な変換はできません」などといっているのだろうか。

図2：ビルドエラーの説明



### Memo

プロジェクトで「Option Strict On」を指定していない場合は、このビルドエラーは発生しない（コラム「Option Strictについて」参照）。また、ここでいう「整数型」とは、Byte、Short、Integer、Longの4つの型を意味している。

### 「¥演算子」と「/演算子」の違い

「¥演算子」のヘルプを見て、その理由がわかった。ヘルプの「解説」部分に、

「Option StrictがOffの場合は、除算を実行する前に、浮動小数点数式は内部処理形式が長整数型 (Long) に丸められます。Option StrictがOnの場合は、コンパイラエラーが発生します。」

という記述があった。

そこで、演算式を、

```
Answer = x / y
```

と訂正する。訂正した瞬間に「x / y」の下に破線が出る。そこをマウスでポイントすると、先ほどと同じく「'Single'から'Integer'への暗黙的な変換はできません」という説明を見ることができ（図2）。

VB.NETは、コードを記述した瞬間に、シンタックス上の細かなミスまで教えてくれる。これも便利な機能である。最近はこの慣れてしまったので、VB6が使いにくくなってしまった。

だが、このメッセージは何を言いたいのだろうか。「¥」を「/」に変えた瞬間に起きたのだから「/演算子」に関係

があると考え、「/演算子」のヘルプを見てみることにした。すると、

「2つの数値の商を計算し、結果を浮動小数点で返します。」

と書いてある。変数「Answer」が「Integer」型なので、結果を代入できないのであろうか。それにしても、わかりにくいメッセージである。

仕方がないので、「Answer」のデータ型を「Single」に変更する。

```
Dim Answer As Single
```

これで、先ほどの破線が消えた。VB.NETのデバッグは非常に楽だ。

ではプログラムを実行してボタンをクリックしてみよう。

驚いたことに、すぐに図3のようなメッセージボックスが出る。例外が発生しないようだ。これはバグなのか。

こうなった理由がどうしてもわからないので、この「Try」構造の後ろに、

```
MessageBox.Show(Answer.ToString)
```

と記述して、計算の結果を表示してみる。

これでプログラムを実行すると、なんと結果は図4のようになった。

図3：問題なく処理された？



図4：除算結果は無限大

