

ツール

プロジェクト解析

<http://www.vector.co.jp/soft/win95/prog/se296488.html>

多機能アドイン

<http://www.mztools.com/v3/download.aspx>

VB6 の一般ルールなど

変数の後に付いている記号

BASIC 言語の初期の時代には、データ型を識別するために、特定の記号を使用している場合があった。例えば、文字列型はドル記号 (\$)、整数型はパーセント記号 (%) といったルールを決めて、変数の名前などの後ろに付けた。この名残りがいまでも残っていて、記号が使用できる。

記号	型
\$(ドル記号)	文字列型
%(パーセント記号)	整数型
&(アンバサンド記号)	長整数型 (Long)

整数のサイズ

型	サイズ	範囲
バイト型 (Byte)	1 バイト	0 ~ 255
ブール型 (Boolean)	2 バイト	真 (True) または偽 (False)
整数型 (Integer)	2 バイト	-32,768 ~ 32,767
長整数型 (Long)	4 バイト	-2,147,483,648 ~ 2,147,483,647
単精度浮動小数点数型 (Single)	4 バイト	-3.402823E38 ~ -1.401298E-45 (負の値)、1.401298E-45 ~ 3.402823E38 (正の値)。
倍精度浮動小数点数型 (Double)	8 バイト	-1.79769313486232E308 ~ -4.94065645841247E-324 (負の値)、4.94065645841247E-324 ~ 1.79769313486232E308 (正の値)。
通貨型 (Currency)	8 バイト	-922,337,203,685,477.5808 ~ 922,337,203,685,477.5807

10 進型 (Decimal)	14 バイト	小数部分を持たない数値の場合、-79, 228, 162, 514, 264, 337, 593, 543, 950, 335 ~ 79, 228, 162, 514, 264, 337, 593, 543, 950, 335 の範囲の値をとります。小数点以下 28 桁の数値の場合、 -7.9228162514264337593543950335 ~ 7.922
日付型 (Date)	8 バイト	西暦 100 年 1 月 1 日 ~ 西暦 9999 年 12 月 31 日
オブジェクト型 (Object)	4 バイト	オブジェクトを参照するデータ型
文字列型 (String) (可変長)	10 バイト	+ 文字列の長さ 0 ~ 2GB
文字列型 (固定長)		文字列の長さ 1 ~ 2GB
バリエーション型 (Variant) (数値)	16 バイト	倍精度浮動小数点数型の範囲と同じ。
バリエーション型 (Variant) (文字列)	22 バイト	+ 文字列の長さ 可変長の文字列型の範囲と同じ。
ユーザー定義型	--	要素に依存 それぞれの要素の範囲はそのデータ型の範囲と同じ。

Load ステートメント

フォームまたはコントロールをメモリにロードします。

構文

Load object

object には、ロードする Form オブジェクト、MDIForm オブジェクト、またはコントロール配列の要素の名前を指定します。

解説

フォームを表示せずにロードするとき以外は、Load ステートメントを使う必要はありません。通常、フォームがまだロードされていないときにフォームを参照すると、そのフォームは自動的にロードされます。ただし、Set ステートメントまたは If...TypeOf ステートメントを使って参照した場合を除きます。たとえば、Show メソッドを使うと、フォームは表示する前にロードされません。フォームがロードされると、フォームが表示されているかどうかにかかわらず、アプリケーションはそのコントロールとプロパティを変更できます。また、初期化のときにすべてのフォームをロードしておき、後で必要に応じて表示することも可能です。

Call の注意

プロシージャを呼び出すとき、キーワード Call は省略できます。キーワード Call を使って、引数が必要なプロシージャを呼び出す場合は、引数リスト (引数 argumentlist) をかっこで囲む必要があります。キーワード Call を省略するときは、引数リストを囲むかっこも省略しなければなりません。Call 構文で組み込み関数またはユーザー定義型関数を呼び出す場合、その関数の戻り値を取得することはできません

Form の扱い方

<http://kamoland.com/wiki/wiki.cgi?VisualBasic%A4%CE%A5%E1%A5%E2%231>

Form を表示する方法

VB において Form ほど基本的かつ難解なものはないと思います。生成した Form を解放し忘れたため、プログラムを終了したつもりでもそれが残っていて、結局タスクマネージャで終了させるはめになるのは、よくあることです。

Form を表示するには、大きく分けて 3 つの方法があります

A . 暗黙の生成による方法

よく使われる方法です。frmMain という名前の Form がプロジェクトにあるとき、

```
frmMain.Show
```

として表示します。破棄するには、

```
Unload frmMain
```

とします。簡単なのですが、frmMain を複数表示させようとするとき、扱いが難しくなります。そのため、frmMain を 1 枚だけしか表示しない場合に用います。

B . 明示的に生成する方法 #1(コンパイル時確定)

```
Dim frmInst As frmMain  
Set frmInst = New frmMain  
frmInst.Show
```

同様に、破棄するには、

```
Unload frmInst  
Set frmInst = Nothing
```

とします。少し手数が増えます。しかし、frmMain を複数表示してそれぞれを管理するには、この方法をとります。

C . 明示的に生成する方法 #2(実行時確定)

```
Dim frmInst As Form  
Set frmInst = Forms.Add("frmMain")  
Load frmInst  
frmInst.Show
```

破棄するには、

```
Unload frmInst  
Set frmInst = Nothing
```

とします。生成する Form 名が、実行時に確定する場合に使用します。かなり特殊な用法です。