

2,3,4 次方程式の解を求める

Copyright 2009 年、佐々木芳

# はじめに

プログラミングが専門でない技術者にとっても Windows アプリケーション (マイクロソフト Excel など)に実装さ れている OLE オートメーションや VBA(Visual Basic for Applications)は仕事の効率や信頼性を向上させ る上で重要な機能です。特に繰り返し行う定型作業の自動化には有効です。 プログラマにとっては Office 系アプリケーションであろうと技術系のアプリケーション(例えば CAD/CAM など)であろうとそれらに搭 載されているオートメーション機能、VBA はなじみやすいかもしれませんが、プログラミングが専門でない技 術者にとって Office 系のアプリケーションでオートメーションや VBA の技術になじむのはなかなか大変だと思 います。オートメーションをうまく利用するためには2つの側面を理解する必要があります。1つ目は、オート メーションに対応した VisualBasic などの言語仕様を理解することです。これはどの用途のアプリケーション でも共通です。2 つ目は、各アプリケーションの機能を公開するオブジェクトモデルを理解することです。これ はアプリケーションごとにまったく違います。 今のところ私の経験した CAD アプリケーションではオートメーション を利用するためのドキュメントはあまり充実していませんでした。それらに比べるとマイクロソフトの Excel など は日本語のドキュメントが付属してそれなりに理解しやすくはなっていますが、Execlを直接使わない場合 にはこの膨大な機能のドキュメントをオートメーションの入門書とするのはあまりいい方法ではないでしょう。 そこで、技術系向けのオートメーションプログラミングの入門的な例を作ってみようと思い立ち、まず2.3.4 次方程式の解を求める電卓風のアプリケーション「Arithmetika」を作りました。 今後、このアプリにオートメ ーションの機能を追加して、オートメーションプログラミングの入門書にしてみようと思います(うまくいけばいい のですが)。ただし、VisualBasic などの各言語仕様についてはそれぞれの解説書を参考にしてください。 (2009/01/10)

オートメーションサーバの機能を追加しました。今まで Office や CAD アプリでオートメーションの機能を使い それなりの効果が期待できることは経験していましたが、自分の作成したアプリケーションにオートメーション 機能を実装したことはありませんでした。そこで今回は VC6+MFC アプリケーションに後からどのようにオート メーション機能を実装したらよいかの実証実験だと思い進めてきました。(2009/04/21)

#### 参考文献

- [1] マイクロソフト MSDN ライブラリ 2001 年 10 月リリース
- [2] OLE オートメーション プログラマーズ リファレンス(アスキー出版)
- [3]**ATL インターナル**(アスキー出版)
- [4]ATL COM プログラミング(翔泳社)

# 目次

1	イン	/ストール	4
2	操作	=方法	5
	2.1	方程式の解を求める	6
	2.2	VB スクリプトにより解を得る	8
3	解詢	拉	8
	3.1	複素数の平方根	8
	3.2	実数係数の2次方程式の解	9
	3.3	実数の立方根	9
	3.4	複素数の立方根	9
	3.5	実数係数の3次方程式の解	9
	3.6	実数係数の4次方程式の解	10
4	才一	-トメーション	10
	4.1	Arithmetika のインスタンスを作成する	10
	4.2	Arithmetika のオブジェクトモデル	10
	4.3	Application オブジェクト	11
	4.4	Complex オブジェクト	12
	4.5	Complexes オブジェクト	12
	4.6	Equation オブジェクト	13
	4.7	Output オブジェクト	14
	4.8	UserEvent オブジェクト	14
5	才一	- トメーションクライアントの例	14
	5.1	Visual Basic 2008 Express Edition	14
	5.2	Excel 2000 VBA	16
	5.3	VBScript	19
	5.4	Arithmetika 組込み VBScript	19
6	修正	<b>一</b> 履酥	20

# 1 インストール

<u>ここから</u>、LZH 形式の圧縮ファイルをダウンロードしてください。適当なフォルダに解凍すると以下の2つのファイルができます。

- Setup\_Arithmetika.msi
- MSVCP60.DLL

Windows XP 以降の OS へのインストールは Setup\_Arithmetika.msi を実行するだけです。 XP 以前の OS(例えば Windows 2000 Professional など)ヘインストールする場合は、最初に MSVCP60.DLL をシステムフォルダにコピーしてから Setup\_Arithmetika.msi を実行してください。 MSVCP60.DLL はマイクロソフト の再頒布可能ファイルです。

MSVCP60.DLL はマイクロソフトサポートオンライン(<a href="http://support.microsoft.com/kb/259403/ja">http://support.microsoft.com/kb/259403/ja</a>)から取得できます。

インストールが完了すると、スタートメニュとデスクトップへショートカットが作成されます。

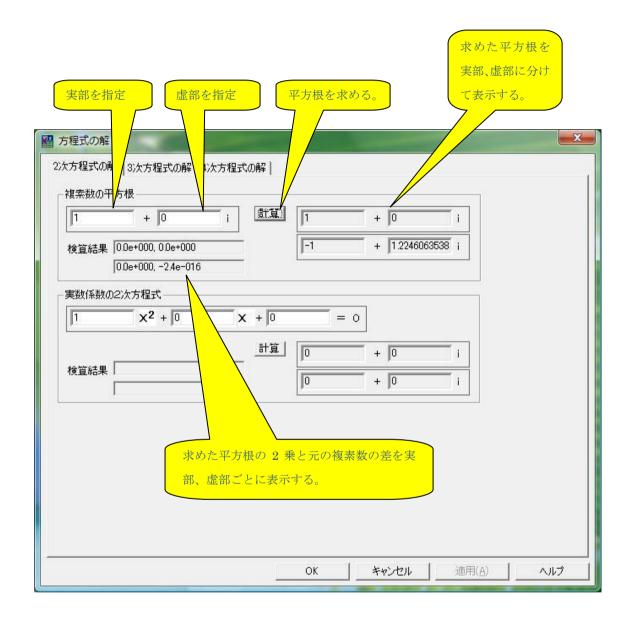
# 2 操作方法

スタートメニュまたはデスクトップのショートカットからArithmetikaを起動すると以下のウインドウが現れます。

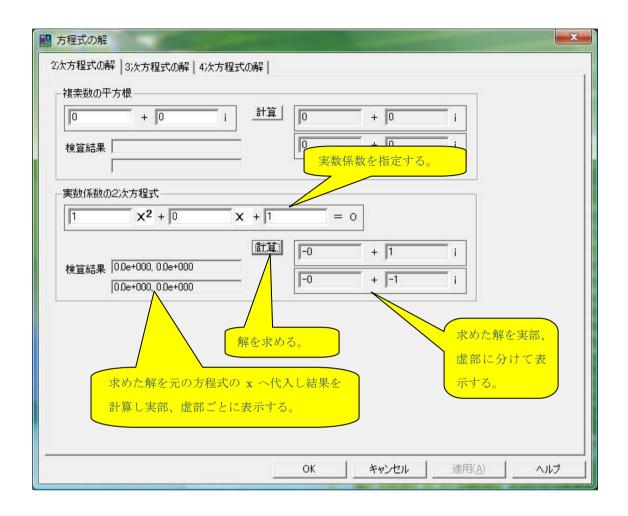


# 2.1 方程式の解を求める

複素数の平方根を求める場合は、このウインドウから以下のように行います。

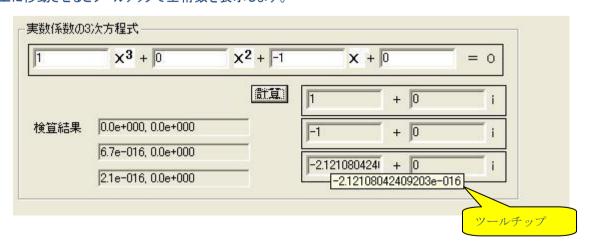


実数係数の2次方程式の解を求める場合は、以下のように行います。



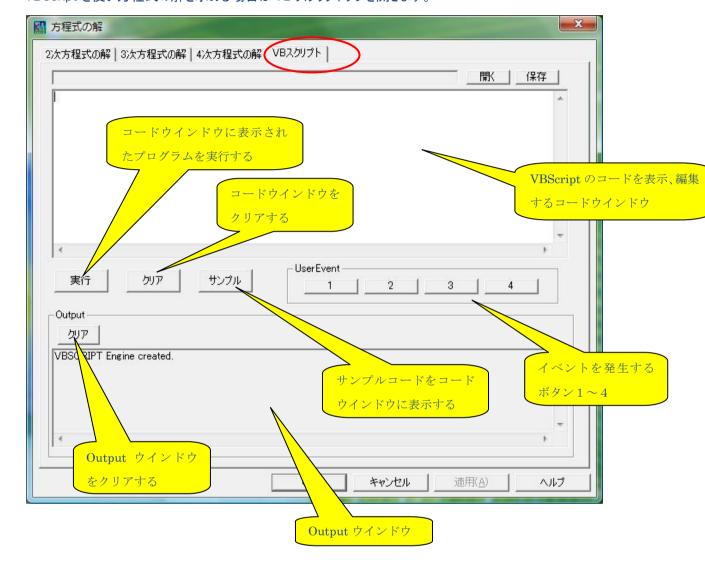
3次、4次方程式の解も同様に求めます。

計算結果の桁数が大きくなると実部、嘘部の数値がすべて表示できない場合がありますが、マウスポインタをその上に移動させるとツールチップで全桁数を表示します。



# 2.2 VB スクリプトにより解を得る

VBScript を使い方程式の解を求める場合は VB スクリプトタブを開きます。



## 3 解説

本プログラムはマイクロソフト VisualStudio6.0 C++コンパイラで作成したアプリケーションです。計算には、C のランタイムライブラリの double 精度の実数平方根、三角関数を使っています。複素数の四則演算には C++の complex<double>を使いました。

以下で、簡単にアルゴリズムの説明をします。

# 3.1 複素数の平方根

複素数 z=a+ib の極形式  $(r, \theta)$ は

$$z = a + ib = r\cos\theta + ir\sin\theta$$

$$= r(\cos\theta + i\sin\theta)$$
$$= re^{i\theta}$$

となるので、複素数 zの平方根は

$$\sqrt{z} = \sqrt{r}e^{i\theta/2} = \sqrt{r}(\cos\theta/2 + i\sin\theta/2)$$

となる。

ただし、

$$a=r\cos\theta$$
,  $b=r\sin\theta$  
$$r=\sqrt{a^2+b^2}, \quad \theta=\arctan(b/a)$$

#### 3.2 実数係数の2次方程式の解

根の公式

$$\chi = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

を使う。

#### 3.3 実数の立方根

実数の立方根は以下の

「逐次計算で平方根、立方根を求める(北海道大学 北村正直)

(http://www.nikonet.or.jp/spring/tikuji/tikuji.htm)]

を参考にさせてもらいました。

#### 3.4 複素数の立方根

複素数の平方根と同様に極形式を使いました。

複素数 zの立方根は

$$\sqrt[3]{z} = \sqrt[3]{re^{i\theta/3}} = \sqrt[3]{r(\cos\theta/3 + i\sin\theta/3)}$$

となる。

#### 3.5 実数係数の3次方程式の解

3次方程式の解を求めるのには以下を

「Cubic Formula(<a href="http://mathworld.wolfram.com/CubicFormula.html">http://mathworld.wolfram.com/CubicFormula.html</a>)」を参考にさせてもらいました。

#### 3.6 実数係数の4次方程式の解

フェラーリの解法を使いました。

#### 4 オートメーション

オートメーションサーバの機能を追加しました。これにより他のアプリケーションから Arithmetika の機能が使用できるようになります。特に VisualBasic、VBScript、JScript、VBA(Visual Basic for Applications)の言語を使用すると簡単に Arithmetika を操作できます。

# 4.1 Arithmetika のインスタンスを作成する

Arithmetikaのインスタンスを作成するにはプログラムID(ProgID)と呼ぶ文字列"Arithmetika.Application" を使って行います。

以下に例を示します。

Visual Basic 2008 Express Edition の場合

Dim A as Arithmetika. Application

A = CreateObject("Arithmetika.Application")

VBA(Visual Basic for Applications)の場合

Dim A as Arithmetika. Application

Set A= CreateObject("Arithmetika.Application")

VBScript の場合、

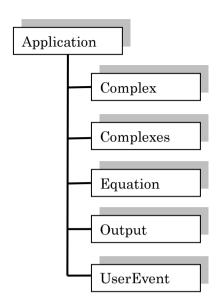
Dim A

Set A= CreateObject("Arithmetika.Application")

CreateObject 関数の詳細は各言語のランゲージレファレンスを参照してください。

#### 4.2 Arithmetika のオブジェクトモデル

まず Arithmetika の公開するオブジェクトモデルを説明します。これらのオブジェクトのインタフェースを定義するにあたって『OLE オートメーション プログラマーズ リファレンス(アスキー出版)』の「標準とガイドライン」を参考にしました。以下に Arithmetika の公開するオブジェクトの階層構造を示します。



「4.1 Arithmetika のインスタンスを作成する」で説明したように、CreateObject 関数はトップレベルオブジェクトである Application (の参照)を返します。以後この Application オブジェクト(の参照)を使い Arithmetika の全機能を呼び出すことができます。

ガイドラインによれば、CreateObject でインスタンスを作成した時点でのユーザインタフェースウインドウは不可視となっていますが、Arithmetika のユーザウインドウは可視状態となります。

# 4.3 Application オブジェクト

Arithmetika アプリケーションのトップレベルオブジェクトです。下位オブジェクトを操作するための方法を提供します。

#### プロパティ一覧

ApplicationApplication オブジェクトを返す。読み取り専用。ParentApplication オブジェクトを返す。読み取り専用。

Name アプリケーション名を返す。読み取り専用。

FullName アプリケーションのフルパス名を返す。読み取り専用。

Visible アプリケーションがユーザに見えるかどうかを返す。または設定する。 読み書

き可能。

Version アプリケーションのバージョンを返す。読み取り専用。

EquationEquation オブジェクトを返す。読み取り専用。OutputOutput オブジェクトを返す。読み取り専用。

#### メソッド一覧

#### Quit ()

説明アプリケーションを終了する。

Help (ByVal HelpFile As String, ByVal helpcontextID As Object) 説明 オンラインヘルプを表示する。

#### 4.4 Complex オブジェクト

方程式の解である複素数を公開するオブジェクトです。

# プロパティ一覧

ApplicationApplication オブジェクトを返す。読み取り専用。ParentApplication オブジェクトを返す。読み取り専用。

Real 複素数の実部を返す。または設定する。読み書き可能。 Imaginary 複素数の嘘部を返す。または設定する。読み書き可能。

Absolute 複素数の絶対値を返す。読み取り専用。 Argument 複素数の偏角を返す。読み取り専用。

Text "<実部>+<嘘部>i"形式の文字列を返す。読み取り専用。

#### メソッド一覧

なし

# 4.5 Complexes オブジェクト

方程式の解である複素数のコレクションを公開するオブジェクトです。

#### プロパティ一覧

ApplicationApplication オブジェクトを返す。読み取り専用。ParentApplication オブジェクトを返す。読み取り専用。

Count コレクションに含まれている項目の数を返す。読み取り専用。

# メソッド一覧

Item (ByVal index As Integer) As Complex

説明 コレクションに含まれている Complex オブジェクトを返す。Index でコレクション内の Complex オブジェクトの位置を指定する。

#### Clear()

説明 コレクションを空にする。

# 4.6 Equation オブジェクト

方程式の解を求めるメソッドとその解を公開するオブジェクトです。

# プロパティ一覧

Application オブジェクトを返す。読み取り専用。
Parent Application オブジェクトを返す。読み取り専用。

Solutions 方程式の解のコレクションである Complexes オブジェクトを返す。 読み取り

専用。

#### メソッド一覧

ComplexSquareRoot (ByVal real As Double, ByVal imag As Double) As Integer

説明 複素数 real+imag\*i の平方根を求める。平方根は Solutions プロパティを使い取得する。

CubicRoot (ByVal x As Double) As Double

説明 実数×の立方根を返す。

ComplexCubicRoot (ByVal real As Double, ByVal imag As Double) As Integer

説明 複素数 real+imag\*i の立方根を求める。立方根は Solutions プロパティを使い取得する。

SolveQuadratic (ByVal A As Double, ByVal B As Double, ByVal C As Double) As Integer

説明 2 次方程式  $Ax^2 + Bx + C = 0$  の解を求める。解は Solutions プロパティを使い取得する。

SolveCubic (ByVal A As Double, ByVal B As Double, ByVal C As Double, ByVal D As Double)
As Integer

説明 3 次方程式  $Ax^3 + Bx^2 + Cx + D = 0$  の解を求める。解は Solutions プロパティを使い取得する。

SolveQuartic (ByVal A As Double, ByVal B As Double, ByVal C As Double, ByVal D As Double, ByVal E As Double) As Integer

説明 4 次方程式  $Ax^4 + Bx^3 + Cx^2 + Dx + E = 0$  の解を求める。解は Solutions プロパティを使い取得する。

#### 4.7 Output オブジェクト

Output ウインドウを処理するメソッドを公開するオブジェクトです。 VBScript からの出力を表示するウインドウです。

#### プロパティー覧

ApplicationApplication オブジェクトを返す。読み取り専用。ParentApplication オブジェクトを返す。読み取り専用。

#### メソッド一覧

TraceLine (ByVal text As String)

説明 Output ウインドウへ文字列 text + 改行を追加する。

Clear ()

説明 Output ウインドウをクリアする。

#### 4.8 UserEvent オブジェクト

ユーザーのマウス操作(ボタンのクリック)をイベントとして処理するオブジェクトです。

#### プロパティ一覧

ApplicationApplication オブジェクトを返す。読み取り専用。ParentApplication オブジェクトを返す。読み取り専用。

Count ボタンの数を返す。読み取り専用。

Name(4) ボタン 1~4 に表示する名前。読み書き可能。

# イベント一覧

OnClick(ByVal index As Integer)

説明 ボタンをクリックしたときに発生するイベント。クリックされたボタンは index(0~3)で識別する。

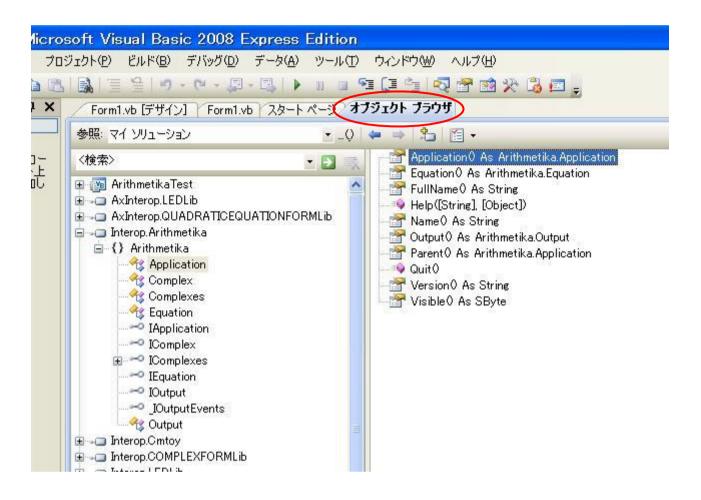
#### 5 オートメーションクライアントの例

# 5.1 Visual Basic 2008 Express Edition

Arithmetika の公開するオブジェクトを操作するために、参照を追加する必要があります。Visual Basic 2008 Express EditionのIDEの「プロジェクト」メニュの「参照の追加」を開くと以下のウインドウが開きます。 ここで COM タブを開き、Arithmetika を選択し、OK ボタンをクリックします。これで Visual Basuic のプロジェクトに Arithmetika の参照が追加されます。



参照が追加されると、オブジェクトブラウザ(「表示」メニュの「オブジェクトブラウザ」で開く)で確認できます。



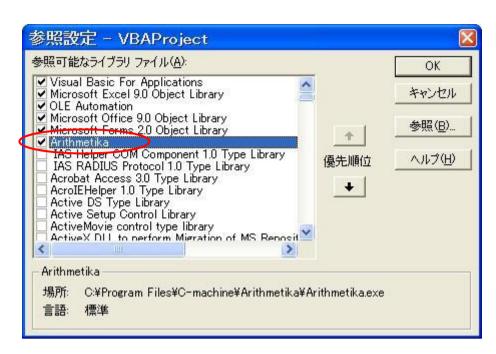
サンプルのプロジェクトファイルはホームページからダウンロードしてください。

Visual Basic 2008 Express Edition は こ ち ら か ら (http://www.microsoft.com/japan/msdn/vstudio/express/) ダウンロードできます。

新しく Visual Basic プロジェクトを作成する場合は、最初に参照を追加します。こうすると、エディターで自動コード補完機能が利用できるので、まず参照の追加を行うことをお勧めします。

#### 5.2 Excel 2000 VBA

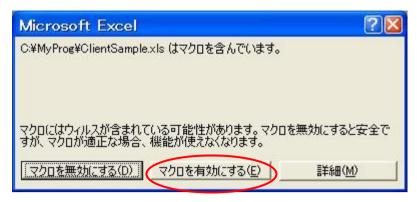
Arithmetika の公開するオブジェクトを操作するために、参照を追加する必要があります。Excel 2000 の「ツール」メニュの「マクロ」→「Visual Basic Editor」を開きます。次に今開いた Visual Basic Editor の「ツール」メニュの「参照設定」を開き、Arithmetika を選択し、OK ボタンをクリックします。これで VBA のプロジェクトに Arithmetika の参照が追加されます。



参照が追加されると、オブジェクトブラウザ(「表示」メニュの「オブジェクトブラウザ」で開く)で確認できます。

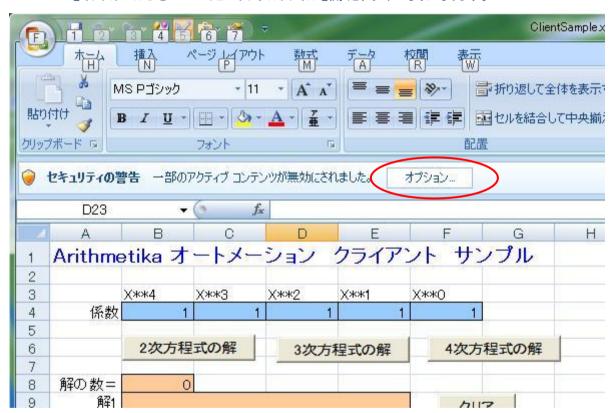


サンプルの Excel ファイルは<u>ホームページ</u>からダウンロードしてください。 サンプルの Excel ファイルを実行すると以下のダイアログボックスが表示されるので、

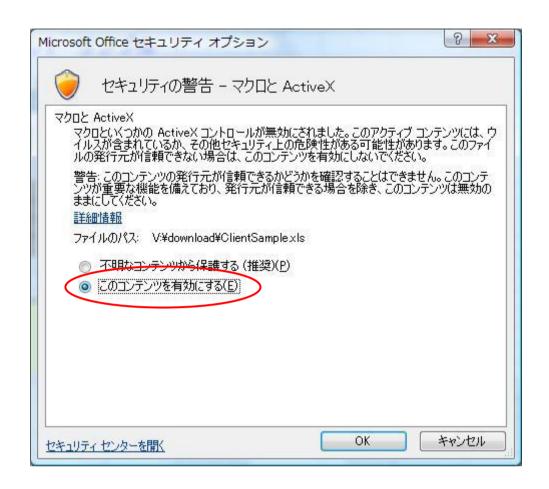


ここで「マクロを有効にする(E)」をクリックします。

Excel2007 をインストールした Vista でこのサンプルファイルを開くと、以下のようになります。



ここで、「オプション」をクリックし、以下のダイアログが表示されたら、「このコンテンツを有効にする(E)」を選択し、「OK」ボタンをクリックしてください。



新しく Excel クライアントを作成する場合は、最初に参照を追加します。こうすると、エディターで自動コード補完機能が利用できるので、まず参照の追加を行うことをお勧めします。

#### 5.3 VBScript

VBScript のプログラムは、通常のテキスとファイルとして作成し、拡張子を vbs にして保存します。 サンプルスクリプトはホームページからダウンロードしてください。

拡張子 vbs のファイルをダブルクリックすると、Windows はそのファイルの内容を VBScript のプログラムとして 実行します。

VBScript については、ユーザーズガイド(<a href="http://msdn.microsoft.com/ja-jp/library/cc392209.aspx">http://msdn.microsoft.com/ja-jp/library/cc392209.aspx</a>)、ランゲージリファレンス(<a href="http://msdn.microsoft.com/ja-jp/library/cc392193.aspx">http://msdn.microsoft.com/ja-jp/library/cc392193.aspx</a>)を参照してください。

#### 5.4 Arithmetika 組込み VBScript

Arithmrtika の「VB スクリプト」タブを開いて VBScript を使用する場合は、参照設定は必要ありません。

以下のオブジェクト名は参照設定せずに既存オブジェクトとして使えます。

- Arithmetika
- Complex
- Complexes
- Equation
- Output
- UserEvent

ここで、Arithmetika は Application オブジェクト(の参照)です。この場合は、VBScrit を実行する前に Arithmetika アプリケーションオブジェクトが存在するので CreateObject 関数を使用する必要はありません。 ボタンをクリックしたときのイベントは以下の関数で処理できます。

Sub UserEvent\_OnClick(index)

'index でクリックされたボタンを識別する

...
End Sub

プログラムは通常のテキスとファイルとして作成します。 拡張子は vbs でもかまいませんが、arivbs などにしておくと、「6.3 VBScript」のプログラムと区別がつけやすくなります。このプログラムは、まず「VB スクリプト」タブのコードウインドウに読み込んでから、「実行」ボタンで実行します。

「VB スクリプト」タブの「サンプル」ボタンをクリックするとサンプルプログラムがコードウインドウに表示されます。

VBScript については、ユーザーズガイド(<a href="http://msdn.microsoft.com/ja-jp/library/cc392209.aspx">http://msdn.microsoft.com/ja-jp/library/cc392193.aspx</a>) を参照してください。

#### 6 修正履歴

V1.0.2 2009,1,10最初の公開

V1.1.2 2009,4,20オートメーション機能の追加(ヘルプを追加) 3 次方程式 $x^3+x^2+x+1=0$ の解を求める時にハングしていたのを修正

<u>V1.1.3</u> 2009,5,24UserEventオブジェクトを追加。