

StarSuite Basic v. s Visual Basic

Prepared by NARAHARA @Naoki
Last Modified on Sep/27/2006

以前は BASIC はその内使われなくなってしまうだろうと考えていた。
近頃 BASIC に対する考えが変わった。BASIC は当面スクリプト言語として生き残りそうだ。プログラム言語としては BASIC はやはり使われなくなってしまうだろう。

身近には Excel の VBA マクロを Calc に移植する機会が出てくるであろう。そのとき、一から書き直すのでなければ StarSuite Basic (OOo Basic) を選ぶであろう。また、最初に学ぶ言語が Fortran や Pascal であった世代が現役である内は BASIC を選ばなければならないかも知れない。

考えを新たにぼちぼちと MS Excel ファイルを OOo Calc ファイルに移植してみることにした。移植している時にもった私的な興味を備忘のために残す。

1. BASIC

SB と VB は同じではない。"StarSuite Basic には、Visual Basic と共通する部分が多くあります。" とガイド(*1)にある。

VB から SB への移植を考慮してコーディングしておくに越したことはない。

1.1 Not

Not 演算子の演算の順位が違う。以下のコードは等価である。

```
If (Not iA) < iB Then '* in VB  
If Not iA < iB Then '* in SB
```

意図をはっきり示す意味からも、VB でも () を使うようにする。

```
If (Not iA) < iB Then '* if -iA is less than iB then  
If Not (iA < iB) Then '* if iA is not less than iB then
```

ガイド(*1) には演算の順位が示されていない。

1.2 CStr

CStr の言語仕様が異なるようだ。 --- 仕様調査、確認要

```
MsgBox CStr(0.02) '* Show "0.02" in VB  
MsgBox CStr(0.02) '* Show "2E-2" in SB
```

1.3 ParamArray

VB で使える ParamArray キーワードは SB では使えない。ガイド(*1) に "注" がある。SB, VB とも Array 関数はある。それに同等なシート関数はない。VB で ParamArray は使うな。

```
vA = Array("SB", "VB", "QB")  
MsgBox vA(0) '* Show "SB"
```

1.4 If 文

ガイド(*1) に明記ないが、単純 If 文が許される。

```
If iA < iB Then iX = iB
```

ガイド(*1) に示されている以下のコードは誤りと考えられる。

```
If I = 5 Then Exit For End If '* Syntax Error  
  
If I = 5 Then '* Correct block if statement  
Exit For  
End If
```

(*6) は単純及びブロック If 文の両方の構文を示している。

1.5 Public 大域変数

VB では Global は旧版との互換のためにのみ残されており、Public と同じ意味である。VB では Global は使うな。

SB で VB の Public にあたるのは Global である。Public と Global は同じではない。

SB を VB に移植することは現実には殆どないであろう。特に問題とはならないであろう。

1.6 Private

SB では Private/Public 指定は変数に対するもので、手続き及び関数では無効のようだ。指定しても文法エラーにはならず、読み飛ばしているようだ。SB では手続き及び関数名の範囲は大域である。(*2) に同様な記述あり。

VB では有効範囲を明示しておくべきである。VB では大域スコープの同名の手続き/関数は許されない。SB では許される。

```
RetVal = Module1.SameNameFoo() '* Refer SameNameFoo in Module1  
RetVal = Module2.SameNameFoo() '* Refer SameNameFoo in Module2
```

1.7 ByRef

SB には ByRef キーワードはない。VB では ByRef を明示するべきだが、移植を考えるのであれば省略すべきであろう。VB でも参照渡しは既定渡しである。

1.8 ハンガリアン記法

趣味である。使うのであれば VB でも接頭辞は一文字にしておこう。

```
strText ==> sText, objObject ==> oObject, lngIndex ==> lIndex, etc.
```

1.9 Mid 関数

SB の Mid 関数には置換機能が拡張されている。VB ==> SB 移植では特に問題にならないだろう。VB には同様な機能の Mid ステートメントがある。

```
MyString = "This was my text"  
Mid(MyString, 6, 3, "is") '* Now MyString is "This is my text"  
  
Mid(MyString, 6, 3) = "is" '* Mid statement in VB
```

MS Office 97 VBA には Replace 関数がない。SB にも Replace 関数はない。VB の文字列処理関数には Mid と Mid\$ のようにバリエーション型と文字列型があることに留意すること。(VB では必ず Mid\$ を使え。)
なお、文字列処理関数に \$ をつけていても SB は文句は言わない。

ガイド(*1) に記載のユーザ定義関数 Replace には誤りがある。(“プログラミング作法” に採り上げて欲しいのかもしれない。)

```
CurrentPos = InStr(StartPos, Search, Source) '* Wrong  
CurrentPos = InStr(StartPos, Source, Search) '* Correct
```

1.10 文字列連結

ガイド(*1) には明示ないが文字列連結演算子に & が許されるようだ。VB が推奨するように & を使おう。

1.11 行長さ

VB では

行継続文字のスペースとアンダースコア () を使用して、物理的に複数の行を結合して、長い論理行を作成できます。行継続文字を使用して、最大 10 行の物理的な行を 1 行の論理行に結合できます。したがって、論理行には合計 10,230 文字まで格納できます。

SB では

その内、目にするだろう。

1.12 For Each ... Next

SB には For Each ... Next ループはない。移植でこのループが生きることはまずない。コードの見直しの良い機会と考えよう。(SB にはコレクションの概念はない。)

1.13 Declare

VB では DLL 内でのプロシージャの登録番号でプロシージャを指定できるが SB では出来ない。

```
Declare Sub Nick "StdCall.dll" Alias "Ext1" (... ) '* for VB/SB
Declare Sub Nick "StdCall.dll" Alias "#1" (... ) '* for VB
```

MS Excel 97 では DLL のパスは 'ActiveWorkbook.Path' に通っていない。
 'ActiveWorkbook.Path' に DLL があっても DLL はフル パスで指定する必要がある。しかるに Calc では通っている。
 FreeLibrary 関数は SB にはあるが VB にはない。

2. Macro

2.1 セル アドレス

アドレスのインデックスが異なる。

Calc			Excel		
A1	B1	(2, 0)	A1	B1	(1, 3)
A2	B2	(2, 1)	A2	B2	(2, 3)
(0, 2)	(1, 2)	(c, r)	(3, 1)	(3, 2)	(r, c)

2.2 入力規制

Excel で言う所の入力規制で選択肢リストをマクロで設定する際の文字列に違いがある。例えば、10 インチ (10"), 20 インチ (20") のリストを与える場合は、(あえて Chr(34) == "" を使わないと)

```
sList = "10"" 20"" '* in VB
sList = ""10"";""20"" '* in SB
```

Excel では再設定にはシートの保護解除が必要である。Calc ではシートに保護がかかっている場合でも設定できる。

2.3 セルの名前

Calc ではセル(範囲)の名前のスコープはドキュメントである。シート スコープの名前は付けることは出来ない。

--- * ---

- (*1) "StarSuite 8 Basic プログラミング ガイド"
<<http://docs.sun.com/app/docs/doc/819-1332?l=ja>>
- (*2) <<http://oooug.jp/compat/basic/pukiwiki.php?VBA%20%A4%C8%20OpenOffice.org%20Basic%20%A4%CE%B8%DF%B4%B9%CO%AD>>
- (*3) "How to Use BASIC Macros in OpenOffice.org"
<http://documentation.openoffice.org/HOW_TO/various_topics/How_to_use_basic_macros.sxw>
- (*4) "VBA to StarBasic Cross Reference"
<http://documentation.openoffice.org/HOW_TO/various_topics/VbaStarBasicXref.pdf>
- (*5) "VBA to StarBasic Cross Reference Example Files"
<http://documentation.openoffice.org/HOW_TO/various_topics/VbaStarBasicExampleFiles.zip>
- (*6) "Andrew Pitonyak's Macro Document"
<http://documentation.openoffice.org/HOW_TO/various_topics/AndrewMacro.sxw>
"OpenOffice.org の有用なマクロ情報"
<http://openoffice-doc.j.sourceforge.jp/tr/temp/AndrewMacro_jp.sxw>

--- * ---

- (A) "がんばれ!! ゲイツ君" から名言
 "なんかプログラミングできますとか言って実際はVBしかやったことないとかいう人もいますけれど、申し訳ありませんがそれはプログラミングでなくてただのコーディングですからご承知頂きますと幸いです。"
 --- そうですね。実験しながらのコーディングです。

また、C に関して “本当に良いプログラムというのはそのコードも溜息が出るほど美しいのだ、ということが如実に解るのもこの言語です。”
— ただ、出来の悪いプログラマだと勘違いしてパズルになる。

<END>