

モデル記述のためのプログラミング 1

プログラミングに入る前に

07186 井原 智彦*

平成 14 年 5 月 10 日

1 はじめに

工学や経済学の数値計算に用いるプログラミング言語としては、FORTRAN, Pascal, Object Pascal, C, C++, C#, Java, Visual Basic などがあるが、いずれも手続き型言語（もしくはオブジェクト指向言語）であり、基本的なことに関しては、その文法が若干異なるだけで、記述方法はほぼ同じである。

今回は、研究室で多く用いられている C++ および Visual Basic のサンプルコードを示しながら、基本的なことについて解説したい。

また、最後に、プログラミングを勉強していくために、どういった書籍が必要となるか、簡単に説明する。

2 コードを書く前に

2.1 C++

UNIX を記述するために AT&T Bell 研究所（現 Lucent Technologies Bell 研究所）の Brian W. Kernighan や Dennis M. Ritchie らが開発した C に対し、C++ は、同じく Bell 研究所の Bjarne Stroustrup らが C にオブジェクト指向を持たせるために作成した言語である。しかし、現在の C++ は、C に対して単にクラス^{*1}が付け加わっただけではない。最近になっても、例外処理、テンプレート、そして STL と、新しい要素を次々に取り入れ、未だに拡張を続けている言語である^{*2}。

この演習では、最新の ANSI C++ に則って進めるのではなく、ANSI C++ のうち、研究室でよく用いられている Visual C++ 6.0 や C++ Builder 5.0 に採用されている部分について採り上げるようにしたい。

2.2 Visual Basic / Visual Basic for Applications

Visual Basic は BASIC より派生したプログラミング言語であり、Visual Basic for Applications は Visual Basic を真似て作られた Microsoft Office アプリケーション^{*3}用のマクロ記述言語である。

* 東京大学大学院工学系研究科地球システム工学専攻博士課程， E-mail ihara@globalenv.t.u-tokyo.ac.jp

*1 クラス (class) についてはいずれ説明する予定。

*2 実は、よく用いられる bool 型も、最近になって加わった型である。

*3 VBA は、Microsoft 社の製品、特に Microsoft Office 用に開発された言語ではあるが、他社製品でも採用されていることがある。代表的な例が、Microsoft 社に買収される前の Visio 社の Visio。

BASIC 生まれで、かつ、初心者でも容易に扱える言語として設計されたので、以下に示すような特徴がある。

- 変数を宣言しなくとも、変数が使用できる。
- 変数の宣言と同時に初期化がおこなわれる。
- 非常に扱いやすい Visual Basic Editor。
- 高機能なソースコード解釈。
- 豊富な組み込みデータ型。
- 実行時エラーへの対処。
- その他いろいろ。

ただし、変数を宣言しないで使用できる、というのは、変数名を誤って記述しても、コンパイラはそれをエラーと見なさずにコンパイルしてしまう、という欠点につながる。そこで、ソースの冒頭で Option Explicit と記述して、コンパイラが宣言していない変数はエラーとみなせるような環境にした方が望ましい。

また、変数名に関してだが、Visual Basic は大文字も小文字も同じように認識する特徴があり、Visual Basic Editor では、存在する変数に関しては、(大文字と小文字とを誤って) 記述した途端、自動的に正しい大文字小文字を揃えてくれるという特徴がある。これを利用して、変数や関数、サブプロシージャで名前を付ける際は、適度に大文字を混ぜてやると、エラーの発見が容易となる。

3 手続き型プログラミングひとめぐり (1)

3.1 変数

C++では、表 3.1 にある基本型が存在する。表中、signed とあるのは符号付き、unsigned とあるのは符号なしである。

なお、C++の場合、処理系およびコンパイラによって、基本型のサイズ(必要な記憶容量)が異なる。通常、int 型が最も実行効率がよくなるように設計されており、32bit OS では、int 型は 32 ビット(つまり 4 バイト)である場合がほとんどである。表には、参考に Visual C++ 6.0 における基本型のサイズも付記する。

Visual Basic の場合 Visual Basic(VB) および Visual Basic for Applications(VBA)^{*4}の場合、表 3.2 にあるような組み込みデータ型が存在する。

Win32API Win32API (32bit OS で用いられる Windows API) では、言語に依存しないように、また別の型が用いられている。これに関しては、必要に応じて、自分で調べて欲しい。

3.2 変数・定数の使い方

3.2.1 変数・定数の宣言・定義・初期化

C++では、int i; とすることで変数の宣言がおこなえる。その場で初期化されないが^{*6}、int i = 0; とすれば、宣言と同時に初期化される(つまり i を定義したことになる)。

^{*4} Visual Basic Script Edition(VBS) も VB や VBA とほぼ同様。

^{*6} static 変数を除く。

表 3.1 C++の基本型

分類	型	説明
整数	bool	真偽を表すのに使い、値は true もしくは false のみを取る。すべての条件式は bool 型の値を返す。たとえば、i の値によって <code>i!=0</code> は true または false を返す。VC++6.0 の場合、bool のサイズは 1 バイトである。
	char	通常、実行文字セットの要素を格納する。VC++6.0 では 1 バイト。なお、C++コンパイラは、char、signed char、unsigned char はそれぞれ別々の型として扱う。
	short	short int と同。サイズが char 型以上、int 型以下の整数型で、VC++6.0 では 2 バイト。unsigned short や signed short も宣言できるが、signed short は short と同じ意味である。
	int	サイズが short int 型以上、long 型以下の整数型。VC++6.0 では 4 バイト。unsigned int や signed int も宣言できるが、signed int は int と同じ意味である。
	<code>_intn</code>	サイズ付きの整数。n は整数変数のサイズをビット数で表す。n の値は 8、16、32、64 のいずれか。
	long	long int と同。サイズが int 型以上の整数型。VC++6.0 では 4 バイト。unsigned long や signed long も宣言できるが、signed long は long と同じ意味である。
浮動小数点	float	float 型は最小の浮動小数点型。VC++6.0 では 4 バイト。
	double	サイズが float 型以上、long double 型以下の浮動小数点型。VC++6.0 では 8 バイト。
	long double	サイズが double と等しい浮動小数点型。VC++6.0 では 8 バイト。

自動変数（キーワードは auto だが、関数内で宣言すると自動的に自動変数となる）は、その自動変数が宣言されている関数内でのみ有効である。変数が有効な範囲をスコープという。クラスのメンバ変数としてはパブリック変数（public）とプライベート変数（private）とがあり、プライベート変数は宣言したクラス内でのみ有効であり、パブリック変数はクラス外からも利用できる。

ただし、パブリック変数を宣言しても、該当ファイルを読み込まない限り、その変数が読み込まれることはない。これに関しては、関数を説明する際に、詳しく説明する。

なお、`const double SOLAR_CONST = 1367.4` と、const を付加してやれば定数が作成できる。定数に関しては宣言のみをおこなうことはできず、同時にその場で定義してやる必要がある。定数に関しても public や private を付加してやることにより、それぞれパブリック定数、プライベート定数が作成できる。

Visual Basic の場合 `Public intValue As Integer` とすることでパブリック変数が、`Private intValue As Integer` とすることでプライベート変数が宣言できる^{*7}。いずれもモジュールレベルで使用する。一方、プロシージャレベルで `Dim intValue As Integer` と宣言すると、プロシージャ変数が宣言できる。Visual Basic では変数を宣言した途端、初期化も同時におこなわれる。たとえば、Integer 型の変数は自動的に 0 に初期化される。

プライベート変数は、宣言したモジュールのすべてのプロシージャから参照可能である。そして、C++と異なり、パブリック変数は、宣言したモジュールだけではなく、他のモジュールも含めて、すべてのプロシージャ

^{*7} ただし、オブジェクトモジュールでは、Public キーワードを付加してもプライベート変数になってしまう。Option Private Module と宣言した標準モジュールでも同様。

表 3.2 Visual Basic の組み込みデータ型

分類	型	説明
真偽	ブール型 (Boolean)	2 バイトの変数。真 (True) または偽 (False) のみ格納できる。他の数値型をブール型の値に変換すると、0 は偽 (False) になり、0 以外の値はすべて真 (True) となる。一方、ブール型の値を他のデータ型に変換すると、偽 (False) は 0、真 (True) は -1 になる。
整数	バイト型 (Byte) 整数型 (Integer) 長整数型 (Long)	1 バイトの変数。0 ~ 255 の範囲の単精度の正の数値を格納する。 2 バイトの変数。-32,768 ~ 32,767 の範囲の値をとる。 4 バイトの変数。-2,147,483,648 ~ 2,147,483,647 の範囲の値をとる。
固定小数点	通貨型 (Currency)	8 バイトの変数。整数形式で表され、10,000 倍されて整数として記憶された、15 桁の整数部分と 4 桁の小数部分を持つ固定小数点数。-922,337,203,685,477.5808 ~ 922,337,203,685,477.5807 の範囲の値をとる。
浮動小数点	単精度浮動小数点数型 (Single) 倍精度浮動小数点数型 (Double) 日付型 (Date)	IEEE 32 ビット (4 バイト) の浮動小数点数の変数。負の値は -3.402823E38 ~ -1.401298E-45、正の値は 1.401298E-45 ~ 3.402823E38 の範囲の値をとる。 IEEE 64 ビット (8 バイト) の浮動小数点数の変数。負の値は -1.79769313486232E308 ~ -4.94065645841247E-324、正の値は 4.94065645841247E-324 ~ 1.79769313486232E308 の範囲の値をとる。 IEEE 64 ビット (8 バイト) の浮動小数点数の変数。100/01/01 ~ 9999/12/31 の範囲の日付と、0:00:00 ~ 23:59:59 の範囲の時刻を表すことができる。日付型の変数には、日付リテラルとして認識可能な任意の値 ^{*5} を代入できる。
その他	文字列型 (String) オブジェクト型 (Object)	文字列には、可変長文字列と固定長文字列の 2 種類があり、可変長文字列には約 2GB (2 ³¹) まで、固定長文字列には 1 ~ 約 2GB (2 ³¹) の文字を格納することができる。 4 バイトの変数。オブジェクトを参照するためのアドレスを格納する。
バリエーション型	バリエーション型 (Variant)	バリエーション型は、Private ステートメント、Public ステートメント、または Static ステートメントを使用して明示的にデータ型が宣言されていない、すべての変数に割り当てられる。特殊なデータ型であり、固定長の文字列型 (String) データとユーザー定義型を除く、あらゆる種類のデータを格納することができる。また、特殊な値 Empty 値、Null 値、およびエラー値や Nothing なども格納することができる。バリエーション型のデータがどのように処理されるかは、VarType 関数または TypeName 関数を使って調べることができる。

ジャから参照可能である。

スコープが大きいとどこでも使えて便利なようであるが、思わぬミスを招きかねない。一般には、できるだけスコープを短くした方が望ましい。

なお、Const SOLAR_CONST As Double = 1367.4 とすると、SOLAR_CONST は定数となる。

3.2.2 Variant 型

Visual Basic の場合 Visual Basic の場合、変数を宣言せずに用いたり、宣言しても As 節による型指定をおこなわない*⁸と、その変数は自動的に Variant 型となる。Variant 型はどの組み込みデータ型にも置き換えられる便利なデータ型ではあるが、その代わりに、Variant 型を用いると処理が低速になってしまう。また、本来なら、型エラーとしてコンパイラに発見される部分が、Variant 型の場合、どんな型にも化けるため、エラーとして発見されなくなってしまう。そこで、必要以上に Variant 型を用いるのは推奨できない。

3.3 関数

実行時に 1 つの単位として処理されるコードの集まりを関数と呼ぶ。関数には、値を返す関数と値を返さない関数 (void 型関数) とがある。実行可能なコードは、すべて何らかの関数に含める必要がある。

値を返す関数とは、数学でいう関数 (例えば、 $y = f(x_1, x_2)$) のようなもので、0 個以上の引数に対して 1 つの値を返す関数である。また、標準ライブラリに組み込まれている関数もこれらの一種とみなすことができる (例えば、sin 関数は double 型を返す関数であると考えられる)。

これに対し、void 型関数は、値を返さない。

値を返さない void 型関数は、ひとかたまりの処理をまとめるときによく使われる。一方、値を返す関数は、上記で述べたように関数を作成する際に使われる。

なお、変数と同じく、public や private キーワードを付加して、スコープを設定することが可能である。

C++ では、プロシージャを他のプロシージャ内にネストすることはできない。

いずれも例は後述するサンプルプログラムで紹介する。

Visual Basic の場合 C++ でいうところの関数は、Visual Basic ではプロシージャと呼ばれる。プロシージャには、Function プロシージャ、Property プロシージャ*⁹、Sub プロシージャの 3 種類がある。

Function プロシージャは C++ での値を返す関数、Sub プロシージャは C++ での void 型関数に相当する。

これに関しても、次回以降に説明するサンプルプログラムで紹介する。

4 情報源

プログラミングを勉強するにあたって、情報源となるものをいくつか挙げておく。

*⁸ たとえば、Dim i As Integer とせずに、Dim i のように As 節なしに宣言すると、i は自動的に Variant 型になる。また、Dim i, j As Integer とすると j は Integer 型になるが、i は Variant 型になってしまう。このような場合は、Dim i As Integer, j As Integer と記述する。

*⁹ Class の Property を記述する際に用いられる。ゆえに Class を説明しない今回は、説明を省略。

4.1 書籍

入門書 C++の入門書としては、次の本がいいと言われている。ただし、自分自身は使ったことがなく、読んだことすらないので詳細は不明。

- Skinner, M. T. (春木良且訳).
C++基礎講座.
インプレス, 改訂新版, 1999.
- Skinner, M. T. (春木良且訳).
C++実用講座.
インプレス, 改訂新版, 1999.

Visual Basic (もしくは Visual Basic for Applications) に関しては全く不明である。研究用途で VBA を使う例はあまりないと考えられ、ゆえにその類の参考書も見ることがない。

基本的に、自分にあう本を選べばいいのではないのでしょうか？

C++の本は、C を解説してから説明している本が多く、そうではなくとも、我々のようなモデル記述のためだけにプログラミングをおこなっている者にとって不要な箇所を先に説明している本が少なくない。そういった煩わしさを避ける本として次の本がある（個人的に薦められるとは思いますが、先日購入したばかりで未読）。

- Andrew Koenig, Barbara E. Moo. (小林健一郎訳).
Accelerated C++.
ピアソン・エデュケーション, 初版, 2001.

一次資料 実は、かなりの本がそうなのだが、間違った記述を含んでいることが多い。そんなとき、一次資料なるものがあれば、それは絶対の正確さが保証できる（誤植を除いて）ので、手元にあると便利。プログラミング言語の場合、それは設計者自身の手で書かれたものがそれに相当する。

- Bjarne Stroustrup (株式会社ロングテール/長尾高弘訳).
プログラミング言語 第3版.
アジソン・ウェスレイ・パブリッシャーズ・ジャパン, 初版, 1998.

残念ながら、実は、この本でさえ間違いがあるが、それはロングテールや Stroustrup 自身のサイトに正誤表があるので、それで補うことができる。

Visual Basic の場合は、Microsoft から出ているガイドブックがそれに相当するが、ちょっと高めかも。

おすすめ 個人的に薦められる本は、次の2冊。

- Harvey M. Deitel, Paul J. Deitel (小嶋隆一訳).
C++プログラミング 第2版, Vol.3.
プレントイスホール出版, 初版, 1998.
- 堀江郁弥, 望月うさぎ.
A 級 B 型 C++入門.
ソフトバンク, 初版, 1995.

前者は Vol.1 から Vol.3 までの 3 部作であるが、そのうち上記に挙げた Vol.3 では、STL^{*10}を採り上げており、豊富な例とともに、STL のうち代表的なライブラリをわかりやすく解説している。

後者は、何らかの手続き型プログラミング言語（著者は C を念頭に置いているが）を知っている人間が、C++を理解するのに打ってつけの本。ただし、大分前に出版された本なので、在庫があるかどうかは微妙。

4.2 雑誌

以下の 3 誌が有名かも。いずれも月刊。

- 翔泳社。
Visual Basic Magazine.
- ソフトバンク パブリッシング。
C MAGAZINE.
- 日経 BP 社。
日経ソフトウェア。

VB Magazine は、Windows アプリケーション作成に関する記事がかなり多い。Windows アプリケーション作成に関係なく、ちょっと VB を使う人にとっては、あまり意味のない雑誌かも。

C MAGAZINE は、Windows に特化しておらず、アルゴリズムの紹介などもあって、興味があれば楽しいはず。ただし、難しい。ちなみに、C MAGAZINE の C は computer の C であるらしいが、大半の記事が C で書かれている。

日経ソフトウェアは、この 3 誌のなかでもっともお手軽。プログラミング言語も、C/C++だけではなく、Visual Basic、Delphi、Java、C#と広く採り上げている。半面、自分の興味のある記事が雑誌全体の一部になってしまうという嫌いはある。

どの雑誌も、6 月号あたりから初心者向けに連載記事を一新しているので、購読するなら今から始めるとちょうどいいかも。

4.3 ヘルプ

プログラミングそのものについては全く参考にならないが、例えば、あるオブジェクトのプロパティを調べたいなど、Windows やアプリケーションのコントロールに特化した内容を調べたいときは、実は、開発ソフト付属のヘルプが一番役に立つ（一次資料だから？）。特に、Windows API を調べたい場合は、msdn Library を抜きにはできない。他も、例えば、Microsoft Excel のコントロールの扱い方を調べるのならば、Microsoft Excel 付属の VBA ヘルプ（本体のヘルプではなくて）に一通りのリファレンスが掲載されている。

4.4 インターネットリソース

探せば、いくらでもインターネットリソースは存在する。特に Tips の類は、WWW 上で公開されているものを少し拾ってくるだけでも十分助けになる。一方で、体系立てて、しっかり勉強しようという場合には、ちょっと不向き。

^{*10} Standard Template Library。標準テンプレートライブラリのこと。

他に、メーリングリストやネットニュースでも、プログラミング言語を取り扱っているものがあるので、高度な質問に関しては助けになるかも^{*11}。また、高度な質問でなくとも、過去ログをダウンロードできる場合は、それをダウンロードするとよい。過去ログに対して検索をかけることで、自分が持っている疑問のうち、かなりの疑問に対する回答が記載されているはず。

^{*11} メーリングリストやネットニュースでは、自分で調べて解決が付く質問は、基本的に質問しないものである。また、過去に同じ質問が出ていないのか、事前に過去ログを読んでおくといよい。