モデル記述のためのプログラミング 5

プログラミングの基礎（3）

07186　井原 智彦*

平成 14 年 6 月 7 日

1　手続き型プログラミングひとめぐり（4）

1.1　外部ファイルの読み込み

変数の項で説明を省略したが，ここでは変数・関数について，別のファイルに記述したものをどのようにすれば，他のファイルでそれらの変数や関数を用いることができるか，について説明する．

例えば，半径を与えて円の面積を求める関数 CalcCircleArea なるものを作成する必要があったとする．そして，その関数を他のファイルから呼び出したいとする．その場合，まず外部に公開する必要がある関数や変数の宣言のみを記述したファイル（これを通常ヘッダファイルという）を作成する．例えば，次のような sample501.h を作成したとする[*1]．

```
1  #if !defined(SAMPLE501_H__INCLUDED_)
2  #define SAMPLE501_H__INCLUDED_
3
4  const double PI = 3.1415926535897932;
5
6  double CalcCircleArea( double radius );
7
8  #endif
```

sample501.h はあくまでも関数や変数の宣言がなされているだけなので，これとは別に，それらの実体を記述したファイル（通常ソースファイルという）を作成する．ここでは，sample501.cpp を作成するとする．ただし，通例，ソースファイル名は，さきほど記述したヘッダファイルと同じファイル名で拡張子のみを cpp に変更したものとする．なお，#include "sample501.h"として，sample501.h を読み込むので，改めて宣言する必要はなく，実体の記述のみおこなえばよい．すると，以下のようになる．

* 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 ここで，sample501.h のファイルが#if〜#endif で括られているのに気づくかと思う．これは，条件付きディレクティブと呼ばれるもので，ここでは，コンパイルする際，1 度このファイルを読み込んだら，2 度目以降は読み込まない，という指令である．今回の例は単純なのでそのようなことはないが，いくつものヘッダファイルを読み込む場合，それぞれのヘッダファイル自身がまた別のヘッダファイルを読み込んでいることが多く，それらをどんどんたどっていくと，結果として同じヘッダファイルをいくつも読み込んでいたということになりかねない．当然，同名の変数や関数は 2 度以上，宣言できないので，エラーとなってしまう．その場合，このように条件付きディレクティブを用いて，それを回避してやる．

```
 1  #include "sample501.h"
 2  #include <cmath>
 3
 4  double CalcCircleArea( double radius ) {
 5      return PI * pow( radius, 2 );
 6  }
```

こうすることにより，外部に対して，変数（定数）PI と関数 CalcCircleArea を公開する準備ができた．これらの変数や関数を利用するには，それらを利用するソースファイルの冒頭で，#include "sample501.h" として sample501.h を読み込むだけでよい．たとえば，次のようになる． **［プログラム sample502.dsw 参照］**

```
 1  #include <iostream>
 2  #include "sample501.h"
 3
 4  using namespace std;
 5
 6  int main() {
 7      cout << "円周率は、 " << PI << " である。¥n";
 8      cout << "¥n";
 9      cout << "半径 3 の円の面積は、 " << CalcCircleArea( 3 ) << " である。"
10          << endl;
11      return 0;
12  }
```

なお，#include <cmath>は，標準ライブラリの 1 つである cmath ライブラリ（math ライブラリとも）を読み込めという命令であり，#include "sample501.h"と同じように考えてもらってよい[*2]．

ちなみに，紙面冒頭からおまじないのように用いている#include <iostream>という命令も同様であり，標準ライブラリの 1 つである iostream ライブラリを読み込むものである．iostream ライブラリには，ストリーム系の標準入出力を取り扱う関数群が含まれている．

■Visual Basic の場合　Visual Basic の場合は，同じプロジェクトに含まれているすべてのモジュールが自動的に読み込まれる．一見便利なように思えるかもしれないが，この仕様は変数名の衝突を招きやすい．これを避けるため，モジュール変数は Private キーワードを付加して，宣言・定義した方が好ましい．

[*2] C++に名前空間（namespace）の概念が導入される以前は，#include <cmath>ではなく，#include <math.h>と記述するのが普通であった（要するに C と同じ）．math.h には多くの数学関数が宣言されているが，その math.h を読み込めという命令であることが分かる．#include "sample501.h"との対比では，こちらの方が分かりやすいかもしれない．