第6回 C++の基礎知識

長谷川裕恭，小川丈博

SystemCは，C++をベースに開発されたシステム・レベル言語である．今回から2回に分けて，SystemCを記述するうえで必要となるC++の基礎知識について解説する．まず，C言語とC++の違いについて説明する．その次に，C++の特徴であるクラスの考えかたと利用方法について述べる．

（編集部）

SystemCを VHDLや Verilog HDLの RTL（register transfer level）記述と同等の抽象度で表現するだけであれば，C++の知識はさほど必要となりません．しかし，SystemCは，より抽象度の高いシステム・レベルの検証を高速に行えるところに導入のメリットがあります．

記述の抽象度を上げるには，インターフェースやチャネルといった概念を利用します．これらを理解するにはC++の知識が必要となってきます．そこで，今回から2回に分けて，SystemCを記述する際に必要となるC++の基礎知識を解説していきます．

C++は，C言語にオブジェクト指向言語としての機能を追加したものです．C言語の文法をそのまま受け継いでおり，これにオブジェクト指向言語としてのクラス（構造体の拡張）の概念が追加されています．ここでは，まずC言語との細かい違いについて解説し，その後，SystemCの記述を使用しながら，クラスの概念について解説していきたいと思います．

●行コメント // を利用できる

C言語ではコメントを /*～*/ の間に記述します．これに加えて，C++では，// から改行までをコメントとして扱います．

●初期化と代入が明確に分かれている

宣言と同時に値を設定することを「初期化」といいます．**リスト1**の⑤がこの初期化です．このとき，変数を初期化するための = や（ ）は「初期化子」と呼びます．⑤の（ ）はC++で導入された初期化子（演算子）です．C言語でも = を使用して初期化できますが，これは初期化ではなく代入として扱われています．C++では，この初期化と代入を明確に分けて考えています．

●使用しない引き数を省略できる

関数の宣言を行う場合，使用しない仮引き数名を省略することができます．**リスト1**では，main関数の仮引き数の型のみを記述しており，仮引き数名を記述していません．このように，プログラム中で使用しない仮引き数名は省略することができます．C言語の場合は，プロトタイプ宣言に限って仮引き数名を省略することができました．

●参照は値渡しと同じ形式で記述

リスト1の⑩に記述しているswap関数の仮引き数は参照変数です．参照は，宣言時に型名の後に & を付けて記述します．参照変数は，宣言時に別の変数を代入することで，その変数の別名となります．

```
int x;
int& y = x; // 型& 変数名 = 初期値
```

参照変数の考えかたはポインタと似ていますが，ポインタより自然な形で記述することができます．**リスト1**の記述では，同じ動作を行うswap関数をポインタと参照で記述しています．

ポインタの場合，実引き数にアドレス演算子を使用して swap(&x, &y) のようにアドレス渡しであることを明示する必要があります．一方，参照の場合は swap(x, y) のように値渡しと同じ形式で記述できます．内部に隠しポインタを持ち，アドレスを渡す方法をとっています．

参照変数は初期値として代入した変数の別名となります．使用する場合もこの変数と同じ形式で使用することになります．

●constは書き込み禁止の変数

変数宣言の前に const を付加すると，その変数は書き込み禁止の変数（定数）として定義されます．**リスト1**の④の記述がこれにあたります．この場合，int型の変数value は定数となり，例えば，value = 10; といったように宣言後に代入することはできません．

また，const変数は配列のサイズ指定（定数式）に使用することができます．

```
const int size = 10;
char x[size];
```

C言語にもconstはありますが，上記のように配列のサイズを指定することはできません．

●bool型が用意されている

C++では，int型やchar型といった基本型にbool型が追加されました．bool型はif文などの条件式で使用する真/偽の判定に使用します．真のときには true という値を持ち，偽のときには false という値を持ちます．

●同名の関数を複数定義できる

C言語では，同名の関数を定義することができません．C++では，「オーバロード」と呼ばれる機能により，引き数の型か個数が違えば同名の関数を定義することができます．

リスト1には同名のswapという関数があります．一方の仮引き数がintへのポインタ型，もう一方がintへの参照型となっています．C++では，関数名が同じでも引き数の型が異なる関数をいくつも定義することができます．

このように，同名の関数を複数定義することを「オーバロード」と言います．オーバロードされた関数は，呼び出しの際，実引き数の型に合ったものが選択されます．**リスト1**では，⑥のswap関数の呼び出しがポインタ型のswap関数呼び出しになり，⑦の呼び出しが参照型のswap関数の呼び出しになります．

〔リスト1〕C言語との違い

```
#include <stdio.h>

void swap(int* src, int* dst);
void swap(int& src, int& dst);                    ←①オーバロード
int add(int a, int b = 0);                        ←②デフォルト引き数

int main(int, char* [])                           ←③仮引き数名を省略
{
  const int value = 10;                           ←④書き込み禁止（定数）

  int x(100);
  int y = value;                                  ←⑤初期化

  swap(&x, &y);                                   ←⑥ポインタ型引き数のswap関数呼び出し
  printf("x = %d, y = %d\n", x, y);

  swap(x, y);                                     ←⑦参照型引き数のswap関数呼び出し
  printf("x = %d, y = %d\n", x, y);

  printf("%d + %d = %d\n", x, y, add(x,y));

  printf("%d + %d = %d\n", x, y, add(x));

  return 0;                                       ←⑧すべての引き数を渡した記述
}                                                 ←⑨デフォルト引き数を省略して呼び出し

void swap(int* src, int* dst)
{
  int tmp = *src;
  *src = *dst;
  *dst = tmp;
}

void swap(int& src, int& dst)                     ←⑩参照変数
{
  int tmp = src;
  src = dst;
  dst = tmp;
}

int add(int a, int b)                             ←⑪実際の定義部分
{
  return a + b;
}
```

（実行結果）

```
x = 10, y = 100
x = 100, y = 10
100 + 10 = 110
100 + 10 = 100
```

●仮引き数にデフォルト値を与える

関数定義の仮引き数名の後に「＝値」と記述すると，この値を仮引き数のデフォルト値とすることができます．**リス**