

モデル記述のためのプログラミング 4

プログラミングの基礎 (2)

07186 井原 智彦*

平成 14 年 5 月 31 日

1 手続き型プログラミングひとめぐり (3)

1.1 関数における引数の受け渡し (2)

前回, 参照渡しという概念について説明したが, 参照渡しというのは, C/C++では比較的新しい概念である. 従来は, 次のようにポインタを使って同様のことを表現していた^{*1}. ポインタ渡しの場合は, まず次のように Swap3 関数を作成する.

```
1 void Swap3( int* a, int* b ) {
2     int temp;
3     temp = *a;
4     *a = *b;
5     *b = temp;
6 }
```

そして, この関数を以下のようにして呼ぶ.

```
1 void TestSwap2() {
2     int value1, value2;
3     value1 = 3;
4     value2 = 4;
5     cout << "交換前の value1 は " << value1 << " 、 "
6         << "value2 は " << value2 << " です。" << endl;
7     Swap3( &value1, &value2 );
8     cout << "交換後の value1 は " << value1 << " 、 "
9         << "value2 は " << value2 << " です。" << endl;
10 }
```

* 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} 現在でもポインタ渡しの方が広く用いられている. 過去から使われているという点が大きいが, 加えて, 参照渡しの場合, 一見, その変数が, 参照 (int&型) なのか実体 (int 型) なのか分からないので, 書き間違いが発生し, ひいてはバグの原因になる, と嫌う人もいたためである.

ここで、見慣れない演算子が2つほど見受けられる。*演算子と&演算子である。これらの演算子も含め、ポインタについて解説する。

まず、int*型とは、int型へのポインタという意味である。上記で説明した参照(int&型)と異なるのは、

- 一番の違いは、参照は、実体と同じように使えるのに対し、ポインタはそうではないということである。参照型はあたかも実値が格納されているようにふるまうのに対し、ポインタ型には指し示した実体が存在するメモリ上のアドレスが格納されているに過ぎない。
- 参照は必ず実体が必要であるのに対し、ポインタは実体がなくとも存在しうる。つまり、参照は必ず定義されるのに対し、ポインタは宣言だけでも構わない。

である。

そのため、ポインタはそのままでは使えない。しかし、*演算子を使うと、実体に格納されている値を呼び出すことができる。これで、参照と同じように使える。一方、実体をポインタに渡すにあたっては、実体のアドレスを渡してやる必要がある。&演算子を用いれば、変数(実体)の存在する(メモリ上の)アドレスを呼び出すことができる。

ちなみに、参照は、ポインタを隠しているだけで、実際には、上記のような作業をコンパイラがやっているのである(つまりポインタを使おうが参照を使おうが同じ、ということ)。Visual Basicでも同様である(ただし、Visual BasicではC++のようないわゆるポインタは存在しない)。

なお、参照やポインタを使用した場合は、一連のTestSwap関数で示したように、値の変更が可能である。しかし、場合によっては、値の変更をする必要がないこともある。その際、コーディングのミスで不要に変更してしまうのを防ぐ場合には、引数の前に、constを付加してやればよい。

```

1 void Swap4( const int& a, int& b ) {
2     int temp;
3     temp = a;
4     a = b;                               // エラー!
5     b = temp;
6 }
```

```

1 void Swap5( const int* a, int* b ) {
2     int temp;
3     temp = *a;
4     *a = *b;                               // エラー!
5     *b = temp;
6 }
```

Visual Basicの場合 Visual Basicの場合、関数における引数の受け渡しにおいて、ポインタは表には出てこない。参照渡しのみである。