ご購入はこちら

≫ 文法の曖昧さを理解して確実性と再利用性を高める

マイコンC言語 転ばぬ先のつえ

第17回　演算子⑧…条件演算子とカンマ演算子

鹿取 祐二

本連載では，C言語の言語仕様（文法）の曖昧な部分と，それにより起こる問題を解説します．再利用性と効率が高く，安全かつ安心して使えるソフトウェアが開発できるようになることを目指します．

今回は，条件演算子とカンマ演算子を解説します．式と文の知識を前提とするので，不安がある方は，まずコラムを参照することをお勧めします．

＜編集部＞

28 条件演算子…if文と使い方が異なることに注意

● 制御の流れはif文に似ている

条件演算子は絶対値を求めるプログラムにたびたび登場します．条件演算子を文法的に紹介すると，次のように3つの式を？と：で区切って記述するものです．この形式で記述された式を条件式と呼びます．

```
式1　?　式2　:　式3
```

制御の流れはif文と似ており，式1を評価し，真であれば結果は式2，偽であれば結果は式3となります．従って，何度も登場した次の関数はint型の絶対値を求める関数となります．

```
int abs(int x)
{
      return x >= 0 ? x : -x;
}
```

● if文と決定的に違う部分

▶条件式は式の仲間，if文は文の仲間

ほとんどのC言語の書籍では，「条件演算子を使った条件式はif文と同じことができる」と説明されています．しかし，筆者は条件式とif文は異なるものと考えた方が良いと思っています．その理由は，条件式は式の仲間であり，if文は文の仲間であるからです．両者は式と文という違いがあるので，記述できる場所が違います．もっとも，条件式の方は式なので，；を付ければ式文の文とできます．これに対してif文の方は扱いを式に変更することはできません．

▶例…絶対値の表現方法

例えば，関数からの返却値を記述するreturn文は次のような構文となっています．

```
return　式 ;
```

returnの後には式を記述します．言い換えれば，式は記述できても，文は記述できないのです．結果，returnの後には条件式なら記述できますが，if文は記述できません．もし，if文を使うなら，次のようにあらかじめ絶対値を求めてから，それをreturn文で返すしかありません．

```
int abs(int x)
{
      if( x < 0 )
          x = -x;
      return x;
}
```

何となく格好が悪いですよね．それと同じ理由で条件式なら，次のような記述が可能です．

```
// 絶対値を引数で渡す
      sub( x >= 0 ? x : -x );
```

これは関数呼び出しの実引数には式が記述できるからです．同じように次のwhile文も記述できます．

```
// 絶対値が5と等しい間繰り返す
      while( (x >= 0 ? x : -x) == 5 )
```

これもwhile文の小括弧の中には式を記述するからです．このように条件式は式の仲間なのでif文よりは，はるかに記述可能な場所が多いです．；を付ければ式文の文とすることも可能なので，記述可能な場所は圧倒的にif文より多いと考えて構いません．

● 弱点（注意事項）も存在する

if文よりも応用力が高い条件式ですが，弱点と言うか，注意事項も存在します．それは1つの式には1つの評価結果（型）しか存在しないことです．

▶if文の利点…評価結果の型は複数あってもよい

例えば，if文の場合，次のように真と偽の処理には異なる型の評価結果となる文が記述可能です．

第1回　整数型①…値の範囲と負の内部表現（2021年1月号）
第2回　整数型②…表現できない値を代入したときの規則（2021年2月号）
第3回　整数型③…整数の格上げと算術変換規則（2021年3月号）