≫ 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18回　静的変数領域の削減①…static記憶クラスの基礎知識

鹿取 祐二

表1　C言語の変数の分類…静的変数と動的変数
局所変数と大域変数とは異なり，記憶クラスと密接に関わっている

宣言場所 / 記憶クラス	局所変数（関数，複文の先頭）	大域変数（関数外部）
なし	動的変数①	静的変数②
auto	動的変数⑤	文法違反
static	静的変数③	静的変数④
register	動的変数⑥	文法違反

リスト1　局所変数の宣言場所とスコープ
複文の先頭で宣言した変数は，その複文内でしか使えない

```
void func(void)
{
int a;
        // 変数aのみ使用可能
    {
    int b;
        // 変数aとbが使用可能
    }
        // 変数aのみ使用可能
}
```

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．

（編集部）

今回からは，静的変数領域の削減について紹介します．「静的変数」の一言で，どの変数であるかが理解できれば良いのですが，宣言場所や記憶クラスも絡んでおり，多少複雑です．そこで静的変数の文法的な説明も含めて，その削減方法を紹介します．

● 変数は静的変数と動的変数に分かれる

C言語の変数は，その性質により静的変数と動的変数の2つに分かれます．これは，局所変数や大域変数と異なる意味の言葉であり，両者を混同してはいけません．これらの言葉は記憶クラスとも密接に関わっています．内訳は**表1**の通りです．

▶局所変数と大域変数のおさらい

表1の横軸は，変数宣言が行われている場所の内訳です．関数や複文の先頭で宣言されたものが局所変数です．関数外部で宣言されたものが大域変数です．

局所変数の宣言場所には，{　}の複文の先頭も含まれます．局所変数は，**リスト1**の変数aのように関数の先頭で宣言するのが一般的ですが，変数bのように複文の先頭でも宣言できます．

ただし，複文の先頭で宣言された変数のスコープは，その複文内に限られます．**リスト1**に示す通り，変数bは複文から外れると使えません．変数bのように，同じ関数内であっても使える場所と使えない場所が混在するので，一般的にはあまり使われません．

▶覚えるべきは4通りの組み合わせ

表1の縦軸は，記憶クラスの指定です．文法上，記憶クラス指定子には**表1**に記載したもの以外にもexternとtypedefがありますが，どちらも記憶領域を確保するものではないので除外しています．また，このうち記憶クラスがautoとregisterのものは覚えなくても良いと思います．理由はコラム1に示します．

結果，覚えるべき内容は，局所変数，大域変数ともに記憶クラスの指定なしとstaticの4カ所です．変数の特性としては，**表1**に示す①記憶クラスの指定なしの局所変数のみが動的変数，②～④が静的変数となります．⑤と⑥は①と同じですが，使う機会がないので覚える必要はありません．

▶本稿でやること…static記憶クラスの意味を解説

筆者は，static記憶クラスをライブラリのソースコード以外で利用する必要はないと考えています．しかしMISRA-Cの考え方は真逆で，static記憶クラスの積極的な利用を推奨しています．

従ってMISRA-Cを採用する場合は，static記憶クラスの意味を覚える必要があります．ところが，static記憶クラスには2つの意味があるので厄介で

第9回　派生型④…移植性は無いが可読性バツグン！ビット・フィールド（2021年11月号）
第10回　演算子①…「関数呼び出し」に使う小括弧（2021年12月号）
第11回　演算子②…[ ]と*の意味，++aとa++の違い（2022年1月号）