

## 19. 共用体

見た目は構造体と似ているが、構造体とは違い、すべてのメンバを同一先頭アドレスで保持する共用体というデータ構造がある。領域が重なるので、一度に一種類のデータだけを保持することになる。

### 共用体宣言

```
union foo
{
    int    i;
    char   c;
    double f;
};
```

共用体は `union` を使って宣言する。構造体と同様に、`union` に続く部分をタグ名と呼ぶ。上の例では `foo` がタグ名である。タグ名に続く中括弧の中には、その共用体のデータとなりえる共用体メンバを書き並べる。各メンバは任意のデータ型が宣言できる。宣言の一般的な書き方は次のようになる。

```
union タグ名
{
    第 1 メンバの宣言;
    第 2 メンバの宣言;
    .
    .
    .
};
```

なお、共用体変数の宣言時の初期化は、第 1 メンバの型で行える。

### 共用体への代入と参照

共用体メンバは、構造体変数と同様に、共用体名とメンバ名をピリオド (.) でつなげて参照する。

共用体変数名.メンバ名

また、共用体へのポインタの場合には構造体同様 `->` の記法が使える。

### 例

```
foo.i=1;
printf("foo=%d", foo.i);
foo.f=1.0;
printf("foo=%f", 5.5*foo.f);
foo.c=getchar();
putchar(foo.c);
```

**例題**

整数変数と単精度浮動小数点変数はともに 4 バイトの領域を必要とする。共用体を使うことで 2 つの変数を同一領域に共有させ、小数を入力しその内部表現を 16 進数で出力するプログラムを以下に示す。なお整数を 16 進数で表示する際、最左ビットは符号ではないので `unsigned` で宣言してある。IEEE754 では、単精度浮動小数点の内部形式を「最左ビットは仮数の符号、指数部は 8 ビット (127 のゲタばき)、仮数部 23 ビット」と定めているが、これがあなたの環境でも実現されているかどうかいろいろな数を入力して確かめなさい。

```
#include <stdio.h>

main()
{
    union fdump {
        float d;
        unsigned int i;
    } a;

    printf("Input (float) = ");
    scanf("%f", &a.d);

    printf("%f = %x (16 進数)\n", a.d, a.i);
}
```