ソフトゼミ A 2016 第 5 回

構造体

前回は、一度に大量の変数を宣言するのに有用な配列について解説しました。今回扱う「構造体」も複数のデータを同時に扱うことができ、さらには異なる型の変数をまとめて扱うこともできます。

➢ 構造体について

構造体の基本的な形を見てみましょう。2 つのステップに分かれています。

```
/*構造体の型を宣言するとき*/
struct 構造体タグ名{
    データの型　メンバ名；
    データの型　メンバ名；
    ・・・；
};

/*宣言された構造体を使うとき*/
struct 構造体タグ名　変数名；
```

構造体の型自体の名前を構造体タグと呼び、構造体の構成要素をメンバと呼びます。構造体の型を宣言するときには「}」のあとの「；」を忘れないようにしてください。

main 文の前で構造体の型を宣言しておくことで、main 文の中でほかの型の変数を宣言するときと同じように使えます。

```
struct student x；

int          y；
```

では、プログラムの例を見てみましょう。

```
#include<stdio.h>
struct student{
      int age;
      double height;
};
int main(void){
    struct student a,b;
    a.age=18;
    a.height=157.8;
    b.age=19;
    b.height=175.5;
    printf("Alice %d 才、身長:%f cm¥n",a.age,a.height);
    printf("Bob %d 才、身長:%f cm¥n",b.age,b.height);
    return 0;
}
```

この例では、構造体タグ:student、メンバ:age, height、構造体変数名:a, b、で 2 人の学生の年齢と身長のデータをまとめています。構造体を使うと型の異なる変数をまとめて宣言することができます。

宣言後にメンバにアクセスするときには、「a.age」のように「変数名.メンバ名」とドット(.)を用いて表します。

また、メンバや変数に配列を用いることもできます。この場合には、メンバや変数を指定するときに、通常の配列と同様に[ ]を用いて表します。

```
struct student2{
    int age;
    double height[2];
};


sturuct student2 c,d[2];
```

例えば上のように宣言した場合、以下の 9 つの変数が使えます。

"c.age"　"c.height[0]"　"c.height[1]"
"d[0].age"　"d[0].height[0]"　"d[0].height[1]"
"d[1].age"　"d[1].height[0]"　"d[1].height[1]"

➢ 文字列(string)について

今まで、変数に保存できたデータは数値だけでした。しかし、変数には文字がいくつか並んだ「文字列」を保存することもできます。扱えるのが数値だけではつまらないので、今回サラッとだけ触れておきます（詳しくはゼミ C で解説する予定です)。

文字列を格納するのに使われる変数のひとつが、char 型の配列です。変数 1 個あたりにつき半角英数字 1 文字を格納することができます。全角の場合は変数 2 個分が必要になります。

また、文字列の最後にはヌル文字(ナル文字:¥0)という特殊な文字（変数 1 個分）が自動で格納されます。これは文字列の「終わりのしるし」です

プログラムの例を見てみましょう。

```
#include<stdio.h>
int main(void){
    char name[10] = “Alice” ;

    printf(“私の名前は%s です。¥n”,name);

    return 0;
}
```

3 行目で char 型の配列を宣言し、初期化しています。int 型や double 型の数値を初期化するときと異なり、二重引用符(“ ”)をつけます。このとき、用意した配列の大きさよりも多くの文字を格納しようとすると重大なエラーになるので、配列は余裕をもって宣言しておきましょう。ヌル文字分も考えることを忘れずに。

このとき配列 name には次のように文字が入っています.

name

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
A	l	i	c	e	¥0				

変数内に格納した文字列を printf 文で表示するときは、「%s」を使い(５行目)、ヌル文字¥0 が見つかるまで連続で、配列の中身を表示します。後ろの数値の部分には配列名を書きますが、この時[ ]はつけないことに注意です。
「配列 name(文字列)をヌル文字の前までまとめて表示」のようにとらえてくれれば今のところは OK です。

もちろん scanf 文を使ってキーボードから文字列を読み込むこともできます。

```
scanf("%s",name);
```

このとき、char 型の配列名の前には「&」をつけない、という「約束」になっています。その理由は「ポインタ」を学べば明らかになります(ポインタについては第 7 回で解説します)。ポインタと配列名には切っても切れない関係があるのです…。

➢ 練習問題

1. 学生 3 人の英語とドイツ語のテストの点数を入力し、その平均点を出力するプログラムを作成してください。

2. 学生 5 人の名前と身長を入力して、身長が 180 ㎝以上の学生の名前とその身長を出力するプログラムを作成してください。ただし該当する学生がいない場合は「該当する生徒はいません。」と出力してください。

3. 学生 5 人の出席番号と 50m走のタイムを入力し、最もタイムのよい学生の出席番号とタイムを出力するプログラムを作成してください。