

構造体

前回は一度に大量の変数を宣言する際に必要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ますが、異なる型の変数をまとめることもできます。

➤ 構造体について

基本となる形を見てみましょう。

(構造体を宣言するとき)

```
struct 構造体タグ名{  
    データの型メンバ名;  
    ...;  
};
```

(宣言した構造体を使用するとき)

```
struct タグ名変数 1,変数 2
```

構造体の型自身の名前を構造体タグと呼び、その構成要素をメンバと呼びます。構造体を宣言するときは最後の「}」のあとの「;」を忘れないようにしてください。main 文の前に構造体を宣言しておく、main 文の中で他の型の変数を宣言するときと同じように使うことができます。実際のプログラムの例を見てみましょう。

```
#include <stdio.h>  
struct student{  
    int height;  
    double bmi;  
};  
int main(void){  
    struct student a,b;  
    a.height=150; a.bmi=22.5;  
    b.height=175; b.bmi=19.2;  
    printf("A さん%d センチメートル、BMI : %f\n",a.height,a.bmi);  
    printf("B さん%d センチメートル、BMI : %f\n",b.height,b.bmi);  
    return 0;  
}
```

上の例では、構造体タグ名：`student`、メンバ：`height,bmi`、
構造体変数名：`a,b` で、2 人の生徒の身長と BMI のデータをまとめています。
今までは整数と実数をまとめて使うことができませんでしたが、上の例を見ればわかるように、構造体を使えば型の違う変数もまとめて宣言できます。

宣言後にメンバにアクセスするときは、“`a.height`”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。この時には、メンバや変数を指定するときに、“`a[1].height,b.weight[2],c[3].bmi[4]`”のように通常の配列を宣言する場合と同様に[]を用いて表します。

➤ 練習問題

1. A さん・B さん・C さんの出席番号と 100m 走のタイムを入力し、3 人の平均タイムを出力するプログラムを作成してください。
2. 1～3 日目の日付(○日)・製品 A の売り上げ・製品 B の売り上げを入力し、最も売り上げの多かった日の日付と売り上げの合計を出力するプログラムを作成してください。
3. 問 1 の生徒 3 人が握力測定を行い、それぞれ 2 回測定値を出した。出席番号とそれぞれの 2 回分の測定値を入力し、そのうち記録がいい方と 3 人の平均記録を出力するプログラムを作成してください。

以上です。お疲れ様でした。