

ソフトゼミ A 第 5 回 構造体

構造体は複数のデータを一つにまとめて扱う時に使うものです。配列と違う点は、異なるデータの型をまとめることができることです。配列ほど重要な存在ではありませんが、複雑なデータの処理を簡潔に、わかりやすく書くのに適しています。

使い方

(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時)

```
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}のあとに「;」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ことで、main 文の中で他の型の変数 (int, double 等) を宣言するときと同じように使うことができます。

具体例

a05_1.c

```
#include <stdio.h>

struct Student {
    int number;
    double average;
};

int main(void) {
    struct Student a, b;

    a.number = 18; a.average = 85.7;
    b.number = 25; b.average = 79.5;

    printf("Aさん %d番、平均点 : %f¥n", a.number, a.average);
    printf("Bさん %d番、平均点 : %f¥n", b.number, b.average);

    return 0;
}
```

この例では構造体タグ : Student、メンバ : number, average となり、生徒の番号と平均点をまとめています。

各メンバを指定するときは a.number のように変数名とメンバ名の間に「.(ドット)」を用いて表します。

また、メンバに配列を持ったり、変数を配列で宣言したりすることもできます。この時にはメンバや変数を指定するときに
a[1].age, b.score[2], c[2].score[5]
のように[]を忘れないようにしてください。

演習問題

1. 二つの点の x 座標, y 座標を入力すると、その 2 点を結ぶ線分の midpoint の座標を出力するプログラムを書いてください。座標は `double` 型にしてください。
2. 生徒 3 人分の番号、数学、英語の点数をそれぞれ入力すると 2 教科の合計点が 1 番高い人の番号とその合計点を出力するプログラムを書いてください。