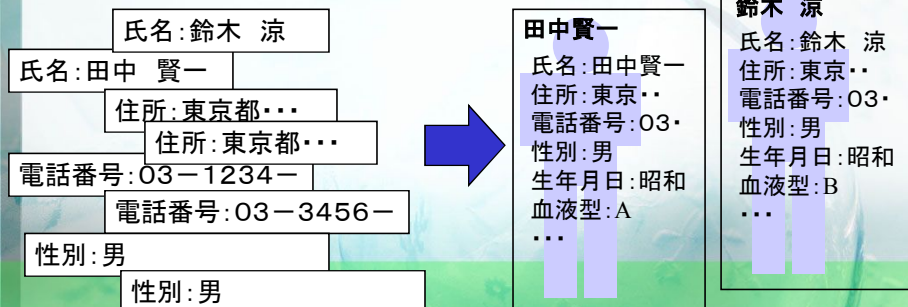


構造体 (Structure)

- 例えばプログラムで人の情報を管理する場合、氏名や住所をばらばらに変数として宣言するのでは無く、まとめて扱った方が便利である。
 - ・ 配列は同じ種類のデータをまとめて扱うことができるが、構造体は異なる性質のデータも扱うことができる
 - ・ 所属する変数をメンバという。



構造体の定義・宣言

- 構造体の定義:

```
struct 構造体名{
    型名   メンバ名;;
    .
    .
    .
};
```
- 構造体の定義例: (構造体名をStrNameとした場合)

```
struct StrName{
    int    data1;
    char   str[64];
};
```
- 構造体の宣言:

```
StrName   dat;
(構造体名) (構造体変数名)
```

構造体を用いた例題

日付情報を、ひとまとめにした構造体を考えましょう。
そして今日の日付を代入して表示してみましょう。

```
#include <stdio.h>
struct Date {
    int year ;
    int month ;
    int day ;
};
void main(void){
    Date d = {2004, 12, 1};
    printf("year = %d ¥n", d.year);
    printf("month = %d ¥n", d.month);
    printf("day = %d ¥n", d.day);
}
```

Date(日付)は年、月、日の情報が必要
→これらを別々にint型変数で定義するのではなく、まとめて管理したい

例えば、Date構造体 から day という変数を宣言すると、その年を知るにはd.year、その月はd.month とすれば良い。

構造体変数の生成と共に初期値を代入している

別の書き方をすると以下ようになる。

```
Date d ;
d.year = 2004;
d.month = 12;
d.day = 1;
```

構造体を用いた課題12-1

プロジェクト名: studentInfo 下記の2名のすべての情報が表示されるようにしなさい。

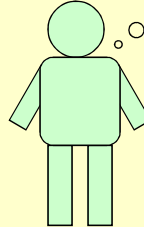
```
#include <stdio.h>
struct Student {
    int no ;
    char name[20]; /* Name */
    char phone[11]; /* Phone */
    char address[20]; /* Address */
};
void main(void){
    Student aoki = {66001, "青木聡", "09034560987", "習志野市泉町1-2-1"};
    Student akasaka= {66002, "赤坂太郎", "09035560122", "習志野市泉町2-2-3"};
    printf("学生番号 = %d ¥n", aoki.no);
    printf("名前 = %s ¥n", aoki.name);
}
```

構造体を用いた例題

「本の情報を管理する構造体」を考えてみましょう。
(本には、タイトルや著者名などが必要です。)

プログラム名: structStudent.c

```
#include <stdio.h>
#include <string.h>
struct StructBook{
    char title[20]; /* Book Title */
    char name[20]; /* Author Name */
};
void main(void){
    StructBook wagahai ={"吾輩は猫である", "夏目漱石"};
    StructBook arashigaoka ={"嵐が丘", "エミリーブロンテ"};
    printf("title= %s Name = %s %n", wagahai.title, wagahai.name);
    printf("title= %s Name = %s %n", arashigaoka.title, arashigaoka.name);
}
```



本という構造体には、タイトルや著者...が必要。

構造体を配列にして利用する

サッカーゲームの選手情報を管理する構造体を考えてみましょう。
(選手にはゼッケン番号と登録名などが必要です。)

プロジェクト名: player

```
#include "stdafx.h"
#include <string.h>
struct Player{
    int no; /* player No */
    char name[20]; /* Name */
};
int _tmain(int argc, _TCHAR* argv[])
{
    Player man_u[100]; int i = 0;
    man_u[0].no = 1; strcpy(man_u[0].name, "Edwin Sar");
    man_u[1].no = 2; strcpy(man_u[1].name, "Gary Neville");
    man_u[2].no = 3; strcpy(man_u[2].name, "Patrice Evra");
    for(i = 0; i <= 2; i++){
        printf("No= %d Name = %s %n", man_u[i].no, man_u[i].name);
    }
    getchar();
}
```

文字(1文字)では無く、文字列(複数文字)を、文字配列に代入するにはstrcpy(代入先の変数, "文字列")という関数を利用: (#include <string.h>が必要)

課題12-2

前例の「サッカー選手の構造体」のサンプルプログラムを参考にして、自分の好きなスポーツクラブのゼッケン番号とポジション名、選手名の1軍全員の一覧を表示させて下さい。

プロジェクト名: player

