

C言語勉強会

～構造体～

構造体とは？

- 複数の型を入れることの出来る箱
- 配列では同じ型(int型など)しか格納できない
- 例えば成績を管理するために
 - 学籍番号 int 型
 - 名前 char型
 - 点数 int 型これらを同じ箱で管理できる



書き方

struct+タグ名

構造体の宣言

```
struct student {  
    int id;           //学籍番号  
    char name[20];   //名前  
    int tensu;       //点数  
};
```

int main

構造体の変数

構造体の変数

```
struct student s;  
s.id = 10;  
strcpy(s.name, "Sato");  
s.tensu = 100;  
printf("%2d %s %3d¥n", s.id, s.name, s.tensu);  
return 0;
```

中身の参照

}

書いてみよう

```
#include <stdio.h>

struct student{
    int id;           //学籍番号
    char name[20];   //名前
    int age;         //年齢
};

int main(void)
{
    struct student s;
    s.id = 自分の番号;
    strcpy(s.name, "自分の名前");
    s.tensu = 自分の年齢;
    printf("%2d %s %3d¥n", s.id, s.name, s.age);
    return 0;
}
```

実行結果

```
C:\¥WINDOWS¥system32¥cmd.exe
10 Sato 100
続行するには何かキーを押してください . . .
```

- 初期化の書き方はもう一つある

```
struct student s = { 10, "Sato", 100 };
```

- こちらの書き方も覚えておこう

typedef

- 構造体はstruct+タグ名で型名(intとか)のように扱っていると考えられる。
- より分かりやすくするためにtypedefで名前を変えることができる。

もとの型名

新しい型名

```
typedef int TEN;
```

```
typedef struct student STUDENT;
```

練習問題

名前、学年、年齢を入力するとそれを表示するプログラムを作りたい。ただし、これらの情報を一括りで管理したいので、構造体を使う。



解答

```
#include <stdio.h>
```

```
struct student{  
    char name[20];  
    int grade;  
    int age;  
};
```

```
int main(void)
```

```
{
```

```
    struct student s;
```

```
    printf("名前を入力してください\n");
```

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

```
    printf("学年を入力してください\n");
```

```
    scanf("%d", &s.grade);
```

```
    printf("年齢を入力してください\n");
```

```
    scanf("%d", &s.age);
```

```
    printf("名前:%s\n学年:%2d年\n年  
齢:%2d歳\n", s.name, s.grade, s.age);
```

```
    return 0;
```

```
}
```


構造体と配列

- 構造体を配列にして扱うこともできる
- 複数の生徒のデータを扱うときなどに便利

```
struct student data[] = {  
    { 10, "kokoa", 15 },  
    { 11, "tino", 13 },  
    { 12, "rize", 16 },  
};
```

構造体と関数

- 構造体を関数の引数や戻り値に出来る

```
struct student input(void){
    struct student s = { 10, "kokoa", 15 };
    return s;
}
void output(struct student s){
    printf("%d:%s:%d\n", s.id, s.name, s.age);
}
```

構造体を構造体の中を使う

宣言

```
struct kyuryo{  
    int yon;  
    int go;  
    int roku;  
};  
  
struct student{  
    int id;  
    char name[20];  
    struct kyuryo k;  
};
```

参照

```
struct student s  
printf("%d:%d:%d¥n",  
s.k.yon,s.k.go,s.k.roku);
```

問題

新選組の隊長らの給料を一括りで管理したい。
構造体を使って、彼らの給料を次のように

隊:名前:4月:5月:6月

と表示させたい。

shinsen.csvファイルからデータを読み込んで
それを構造体に格納し、それを表示するプログラムを作る。

データの読み込み

- 前回やったファイル操作の復習

```
char s[20];  
int a;  
FILE *fp;
```

ファイル読み
取りモード

```
fp = fopen("shinsen.csv", "r");
```

ファイルから
読み込む

```
fscanf(fp, "%[^,],%d", s, &a);
```

ファイルポイ
ンタ以外は
scanfと同じ
ように使える

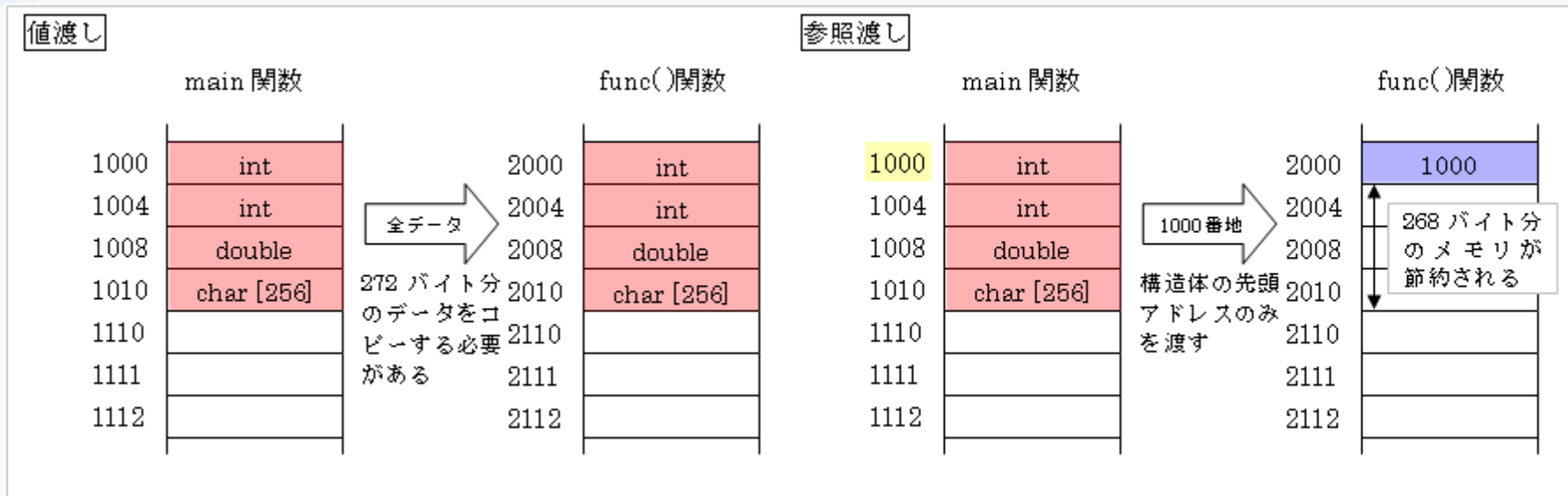
```
fclose(fp);
```

“,”までの文
字読み取る

解答

- 添付したsinsengumiプロジェクトにコードがある
- あくまで一例なので各々で考えて書いてみよう

追記:値渡しと参照渡し



値渡しは全ての値をコピーして関数に渡すので、コピーする時間が生じ、処理が遅くなります。

一方、参照渡しは配列や構造体の先頭アドレスを渡すだけで良いので、値渡しに比べて格段に処理が早く、メモリ消費も少なくなります。

ポインタを使うことで参照渡しを行える
気になったら調べてみよう