

構造体の復習

アルゴリズムとデータ構造B

第2回

構造体，構造体の配列，構造体へのポインタ

- 構造体の利用法の整理
 - ✓ 表や線形リスト等のデータ構造で構造体を利用するため

演習

- ex01.c : ポインタ
- ex02.c : 構造体

教科書 pp.80-83

- 任意のデータ型を自由に組み合わせて作られるデータ構造

例) 身体測定データ (氏名, 身長, 体重) を管理

- 三つの配列で
 - ✓ **暗黙的**に配列0番目の要素が「太郎」のもの, 1番目が「次郎」, 2番目が「三郎」
 - ✓ 各データ間の関係を直接的に表現できていない

	氏名	身長	体重
0	高専太郎	174	64.2
1	高専次郎	163	58.1
2	高専三郎	168	70.3

氏名 : 高専太郎
身長 : 174
体重 : 64.2

氏名 : 高専次郎
身長 : 163
体重 : 58.1

氏名 : 高専三郎
身長 : 168
体重 : 70.3

- 構造体の宣言

構造体であることを示す

構造体タグ名
(任意, 省略可能な時がある)

構造体の型名

```
struct body {
```

```
    char name[20];    // 氏名
```

```
    int height;       // 身長
```

```
    double weight;    // 体重
```

```
};
```

struct body 型の
各メンバ

氏名：高専太郎
身長：174
体重：64.2

- 構造体変数の宣言

- ✓ 実際に使用する変数を宣言

```
struct body a; // struct body 型の構造体変数 a
```

- 構造体メンバへのアクセス

- ✓ ドット演算子（.）を使用
- ✓ 例）構造体変数 a のメンバ int height;（身長）にアクセス

```
a.height = 174;  
printf("身長： %d cm\n", a.height); // 身長： 174 cm
```

文字列：文字の配列

```
char a = 'a';           // 文字
char a[10] = "abcde";    // 文字列
printf("%s\n", a);       // abcde
```

- 変数に文字列を入力するとき,

- ✓ 初期化と一緒に `char a[10] = "abcde";`

ok

- ✓ 初期化の後に `char a[10]; a = "abcde";`

ng

- ポインタだったら `char *a; a = "abcde";`

ok

- ✓ `strcpy`, `scanf` 等

```
strcpy(a, "abcde");
scanf("%s", a);      // 標準入力から入力
```

- 構造体変数の宣言
 - ✓ 実際に使用する変数を宣言

```
struct body a; // struct body 型の構造体変数 a
```

氏名：高専太郎
身長：174
体重：64.2

- 構造体メンバへのアクセス：メンバに値を入力

```
struct body a = {"高専太郎", 174, 64.2};
```

もしくは

```
strcpy(a.name, "高専太郎");  
a.height = 174;  
a.weight = 64.2
```

- 構造体の宣言,
変数の宣言を同時に

```
struct body {  
    char name[20];  
    int height;  
    double weight;  
} a;
```

- 「struct タグ名」にtypedef で別名を

```
struct body a;  
-----  
typedef struct body BODY;  
           型名      別名
```

BODY a; // struct body a; と同じ

```
typedef struct body {  
    ...  
} BODY;
```


身体測定データを一括して扱いたい → 配列の利用

氏名：高専太郎

身長：174

体重：64.2

氏名：高専次郎

身長：163

体重：58.1

氏名：高専三郎

身長：168

体重：70.3

- 3人分なので、`struct body a, b, c;` でもいけるっちゃいけるけど、

`struct body a[3];` の方が扱いやすい

```
a[0].height = 174; // 太郎の身長  
a[1].height = 163; // 次郎の身長  
a[2].height = 168; // 三郎の身長
```

通常の変数と同様，構造体もポインタで指すことが可能

```
struct body *p;  
p = &a;
```

- ポインタを使用したメンバへのアクセス
 - ✓ ドット演算子に加え，アロー演算子が使用可能

```
a.height = 180;  
printf("身長： %d cm\n", (*p).height);  
printf("身長： %d cm\n", p->height);  
// どちらも 身長： 180 cm
```

- Teams の「第〇回講義演習」から ex〇.c をダウンロード
- 指示に従って編集, コンパイル, 実行
 - ✓ 環境はお任せします ➡ みんなの環境を教えてください
- 出力例と比較
- 完成したら, 出力結果をソースコードに直接貼り付け, 編集済みの ex〇.c を提出
 - ✓ 特にフィードバックの予定等はない
- 講義内の演習は**課題点に関係しません**

ex01.c：ポインタのまとめ

- ポインタが指すアドレス，ポインタが指すアドレスの値を，通常の変数，配列，関数の場合に正しく出力する
 - ✓ printf()で，ポインタが指すアドレスを表示するためには **%p** を使ってください
- 配列中の二つの値を入れ替える**swap**関数を作成して，正しく呼び出す
 - ✓ 参照渡しに注意

- 設問 7 はやらなくて大丈夫です

```
> ./ex1
```

変数 a のアドレス, 値を a を用いて出力

```
&a : 0x16b143150, a : 5
```

設問 1 : 変数 a のアドレス, 値をポインタ p を用いて出力

```
&a : 0x16b143150, a : 5
```

配列 a の 2 番目, 4 番目の要素のアドレス, 値を a を用いて出力

```
&a[2] : 0x16b143188, a[2] : 3
```

```
&a[4] : 0x16b143190, a[4] : 5
```

設問 2 : 配列 a の 2 番目, 4 番目の要素のアドレス, 値を p1, p2 を用いて出力

```
&a[2] : 0x16b143188, a[2] : 3
```

```
&a[4] : 0x16b143190, a[4] : 5
```

配列 a の先頭 (0 番目), 2 番目の要素のアドレス, 値を a を用いて出力

```
&a[0] : 0x16b143160, a[0] : 1
```

```
&a[2] : 0x16b143168, a[2] : 3
```

設問 3 : 配列 a の先頭 (0 番目) のアドレス, 値を p を用いて出力

```
a : 0x16b143160, a[0] : 1
```

設問 4 : 配列 a の 2 番目の要素のアドレス, 値を p を用いて出力

```
&a[2] : 0x16b143168, a[2] : 3
```

設問 5 : 配列 a の 2 番目の要素のアドレス, 値を p を用いて配列風の記述で出力

```
&a[2] : 0x16b143168, a[2] : 3
```

設問 6 : 配列 a の 2 番目の要素のアドレス, 値を a を用いてポインタ風の記述で出力

```
&a[2] : 0x16b143168, a[2] : 3
```

関数 func のアドレス : 0x104cbf660, func(5) : 5

設問 7 : 関数ポインタ p_func を用いて, 関数 func のアドレスを出力, fun を呼び出す

```
func のアドレス : 0x104cbf660, func(5) : 5
```

設問 8 ~ 10 : 配列の要素の値を入れ替える swap 関数を作成

```
a[0] = 5, a[1] = 3
```

swap 後

```
a[0] = 3, a[1] = 5
```

ex02.c：構造体のまとめ

- 身体測定データの管理を例題に，
構造体，構造体の配列，構造体へのポインタの使い方を確認

```
> ./ex2
```

設問 3 : aを用いて「氏名：高専太郎，身長：174 cm 体重：64.2 kg」と出力

氏名：高専太郎，身長：174 cm 体重：64.2 kg

設問 4 : 構造体ポインタ p を用いて，構造体変数 a の情報を設問 3 と同様に出力

氏名：高専太郎，身長：174 cm 体重：64.2 kg

設問 6 : 構造体配列の情報を設問 3 と同様に，for文を用いて出力

氏名：高専太郎，身長：174 cm 体重：64.2 kg

氏名：高専次郎，身長：163 cm 体重：58.1 kg

氏名：高専三郎，身長：168 cm 体重：70.3 kg