

プログラミングII

第5回: 構造体

2018年12月19日(水)

筑波大学 情報メディア創成学類

三河 正彦

構造体

- 型を組み合わせて新しい型を作り出すもの
- 復習
 - ♦ `int a; /* 整数型の値を一つだけ入れることができる */`
 - ♦ `int a[10]; /* 整数型の値を複数入れることができる */`
 - ▶ 全ての要素は同じ型でなければならない
- 複数の異なる型のデータの組をまとめて操作したい.

- ♦ 構造体

- ♦ 例:

車の { 名称
色
重量
馬力
サイズ

生徒の { 氏名
学籍番号
国語の成績
算数の成績
平均点

構造体の宣言と構造体変数の宣言

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

```
/* 構造体の宣言 はじめ */
```

```
struct SEITO {  
    char name[20];  
    int  kokugo;  
    int  suugaku;  
    int  eigo;  
    double hensachi;  
};
```

```
/* 構造体の宣言 おわり */
```

構造体名

メンバ

```
int main(void)
```

```
{  
    struct SEITO student; /* 構造体変数の宣言 */  
    struct SEITO students[10]; /* 配列も使える */  
    ...  
}
```

構造体の使い方 (メンバへのアクセス)

```
student.name[0] = 'a';  
student.kokugo = 80;  
student.suugaku = 100;  
student.eigo = 90;  
student.hensachi = 62.5;  
...
```

```
scanf("%d", &(student.kokugo));  
printf("%d=\n", student.kokugo);
```

構造体型変数名.メンバ

という形で**ドット演算子**を使って、
構造体の各メンバにアクセスする。

構造体の使い方 (メンバへのアクセス: 配列の場合)

```
students[0].name[0] = 'a';  
students[0].kokugo = 80;  
students[0].suugaku = 100;  
students[0].eigo = 90;  
students[0].hensachi = 62.5;  
...  
  
printf("%d=\n", students[0].kokugo);
```

typedef

- 既存の型に別名を付ける.
- 書式 `typedef 型名 識別子;`

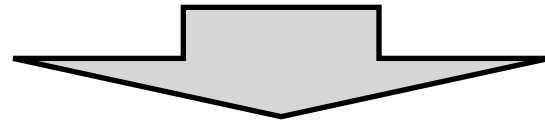
```
/* 構造体の宣言 はじめ */
typedef struct SEITO {
    char name[20];
    int  kokugo;
    int  suugaku;
    int  eigo;
    double hensachi;
} SEITO;
/* 構造体の宣言 おわり */

int main(void)
{
    SEITO student; /* 構造体変数の宣言 */

    ...
}
```

構造体の初期化

```
SEITO student;  
  
student.name[0] = 'a';  
student.kokugo = 80;  
student.suugaku = 100;  
student.eigo = 90;  
student.hensachi = 62.5;
```



```
SEITO student = {"mikawa", 80, 100, 90, 62.5};
```

小さなサイズのメンバ (ビットフィールド)

```
struct BITS{  
    unsigned int a:1; /* aは1ビット */  
    unsigned int b:3; /* bは3ビット */  
    unsigned int c:4; /* cは4ビット */  
};
```

- 小さいサイズのメンバを宣言できる.
- ビットフィールドに使える型
 - ✦ int
 - ✦ unsigned int
 - ✦ signed int

1 byte = 8bit

おわり