



情報処理演習II

No. 11

2023. 10. 31

芝浦工業大学 システム理工学部 機械制御システム学科

担当：桑原

第9、10回講義のまとめ

- ポインタ変数も、他の変数と同様に関数の引数として使用できる

a) 一般の変数が引数： 関数には変数の"値"が渡される

(Call by value)

→ 呼出し側の変数には一切影響を与られない

b) ポインタ変数が引数： 関数には"ポインタ"（変数の先頭アドレス）が渡される

(Call by reference)

→ 呼出し側の変数に変化を与えることができる

- ある関数の中の変数の値を別の関数の中から変えるには、ポインタを引数として受渡す

(復習) 変数を関数にポインタ渡し (Call by reference) する方法

サンプルプログラム 9.3

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

```
void swap(int *x, int *y)
{
    int tmp;
    tmp = *x;
    *x = *y;
    *y = tmp;
}
```

引数はアドレスなので、**ポインタ変数で受ける**

「*x」はポインタxの先にある変数を指すが、メモリ上ではmain関数での変数x そのもの
(*y)でも同様

```
int main(void)
{
    int x = 1, y = 2;

    printf("x = %d, y = %d\n", x, y);
    swap(&x, &y);
    printf("x = %d, y = %d\n", x, y);

    return 0;
}
```

関数swapには変数xとyの**ポインタを渡す**

戻り値以外で関数から値を複数取り出せる

(復習) 配列を関数の引数に使用方法

- 配列を渡される関数の引数はポインタ変数で定義する。

配列を関数に渡す側

```
#include <stdio.h>
#define N 5

int main(void) {
    int i, total;
    int array[N];
    printf("%d 個の整数を入力して下さい。 ¥n", N);
    for (i = 0; i < N; i++) {
        scanf("%d", &array[i]);
    }

    total = sum(array); 配列渡し

    printf("合計は %d です。 ¥n", total);

    return 0;
}
```

配列を渡される側 **ポインタ変数**

```
int sum(int *array) {
    int i, sum = 0;

    for (i = 0; i < N; i++) {
        sum = sum + array[i];
    }

    return(sum);
}
```

**渡された配列を参照、
値の代入が可能**

構造体

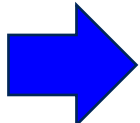
- **構造体**：複数のデータ型の変数を、1つにまとめて操作できるようにしたもの

- 例：学生の情報

- 学籍番号（下5桁）：23123
- 氏名：T.S.
- GPAスコア：3.5

} すべて、ある個人（T.S.さん）の情報
3種類の変数/人

- Q学科に100人の学生がいて、学生情報を管理する場合、
3種類×100人 = 300個の変数を用意する？

 **学生1人のデータをまとめる（構造体にする）と変数の管理が楽**

1. 構造体の定義

```
struct タグ名{  
    データ型 メンバ変数1;  
    データ型 メンバ変数2;  
    データ型 メンバ変数3;  
    ....  
};
```

2. 構造体型の変数の宣言

```
struct タグ名 構造体型の変数名;
```

3. 構造体のメンバ変数へのアクセス

構造体型の変数名.メンバ変数

ドット演算子

サンプルプログラム 各自で書いて実行してみてください。

サンプルプログラム 11.1

```
#include <stdio.h>

// 構造体を定義
struct student{
    int id;
    char name[128];
    double gpa;
};
```

```
int main(void) {
    // 構造体型の変数を作成
    struct student A;

    // 構造体に値をセット
    A.id = 23123;
    strcpy(A.name, "T.S.");
    A.gpa = 3.5;

    printf("No. %t  Name %t GPA %n");
    printf("%d %t  %s %t %.1f %n",
           A.id, A.name, A.gpa);

    return 0;
}
```

構造体型変数の使い方（１）

重要

・構造体型変数の一括初期化

```
struct student A = {23123, "T.S.", 3.5};
```

構造体型変数は、配列と同様に、**{ }**を使って初期化できるが、
構造体の宣言時と同じ順番で初期値を指定しなくてはならない。

・構造体型変数の代入

```
struct student A = {23123, "T.S.", 3.5};  
struct student B = A;
```

構造体型変数は、変数と同様に代入可能
student型変数Bのメンバ変数には、Aのメンバ変数がコピーされる。

・ポインタを経由して操作する場合

```
struct student A;  
// ポインタ変数に構造体へのポインタを渡す  
struct student *pA = &A;  
  
// ポインタ経由で構造体にアクセス  
printf("No.:%d, Name:%s", pA->id, pA->name);
```

- ・構造体型変数は、変数と同様に'**&**'を使ってポインタを渡せる。
- ・構造体型ポインタ変数のメンバ変数にアクセスするときは、**アロー演算子**を用いる。

・構造体の配列

```
struct student students[3] = {  
    {23123, "T.S.", 3.5},  
    {23124, "A.B.", 3.3},  
    {23125, "C.D.", 3.8}  
};  
  
for(int i = 0; i < 3; i++){  
    printf("No:%d, Name:%s", students[i].id, students[i].name);  
}
```

- ・構造体型変数は、変数と同様に配列として扱える。

・関数への値渡し

```
struct student A = {23123, "T.S.", 3.5};  
FuncA(A); // 値渡し
```

- ・値渡しでは、**変数Aが持つ構造体をコピーして関数FuncAに渡す。**
- ・関数内で引数のデータを操作しても、コピーのデータが変わるだけで、**変数Aが持つ構造体には影響しない。**

・関数へのポインタ渡し（参照渡し）

```
struct student A = {23123, "T.S.", 3.5};  
FuncB(&A); // ポインタ渡し
```

- ・ポインタ渡しでは、**ポインタを経由して変数Aが持つ構造体のデータそのものを操作できる。**

サンプルプログラム 各自で書いて実行してみてください。

サンプルプログラム 11.2

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

```
// 構造体を定義
```

```
struct student{  
    int id;  
    char name[128];  
    double gpa;  
};
```

```
void print_student_id(student s) {  
    printf("Id : %d¥n", s.id);  
}
```

```
void change_id(int id, student* s) {  
    s->id = id;  
}
```

```
int main(void) {  
    // 構造体型の変数を作成  
    struct student A = {23123, "T.S.", 3.5};  
  
    print_student_id(A);  
  
    change_id(23999, &A);  
  
    print_student_id(A);  
  
    return 0;  
}
```

課題 1 1

- 今回作成したソースプログラムをGoogleDriveの「課題提出¥課題11」に提出してください。
本日、1限終了時刻を〆切とします。
- 課題11-1：以下の機能を満たすプログラムを作成せよ。
 - 小数型の入力inと出力outをメンバ変数とする構造体Dataを定義する。
 - Data型変数による配列data[5]を定義する。
dataの各要素のメンバ変数inを右表1列目 θ 、outを0で初期化する。
 - data[i] ($i=0\sim 4$)を入力とし、メンバ変数inから出力outを計算する関数Updateを作成する。
Updateの引数：data[i]
処理：data[i]のinから、out(sin出力)を求める。
 - main関数でfor文を使い、関数Updateを呼び出し、data[i]の入力と出力を表示する。

入力in θ (rad)	出力out $\sin\theta$
0	
$1/2 \pi$	
π	
$3/2 \pi$	
2π	

* #include <math.h>を使ってsin(theta)を計算する。ただし、thetaはラジアン単位。

課題 1 1

- ・課題11-2：以下の機能を満たすプログラムを作成せよ。
 - ・小数型の入力inと出力outをメンバ変数とする構造体Dataを定義する。
 - ・t=0秒から0.01秒ごとインクリメントさせ t =1.0秒まで繰り返し処理を行う。
 - ・Data型配列data[101]を定義する。全てのメンバ変数を0で初期化する。
*なぜ要素数を「101」にしたかは各自考えてください。
 - ・tをdata[i]のメンバ変数inに代入し、出力outを計算する
関数Updateを作成する。
Updateの引数：t, data[i]
処理：tを入力とし、振幅 1、周期 1 秒のsin波の値を計算する。
data[i]のinにtを代入、outに計算値を代入する。
 - ・main関数でfor文を使い、関数Updateを呼び出し、更新されたdata[i]の入力と出力を表示する。
- * #include <math.h>を使ってsin(theta)を計算する。ただし、thetaはラジアン単位。

入力in 時間t	出力out sinθ
0	0
0.01	
0.02	
...	
1.00	0