2013 年 7 月 25 日（木）実施

構造体と typedef

typedef 宣言によって，struct 構造体タグ名という表記を再定義し，データ型名のように扱うことができる。構文は

```
typedef struct 構造体タグ名
{
・・・
} 再定義名;
```

となり，この場合の構造体変数の宣言は，再定義名を用いて行うことができる。なお，ここでは構造体タグ名は省略可能である。

構造体を指すポインタ

ポインタ変数の解説の際（第 9 回教材）に，構造体を指すポインタ変数が利用できることを述べた。これは，

```
struct 構造体タグ名 *ポインタ変数名;

または

再定義名 *ポインタ変数名;
```

という宣言によって利用可能である。

ポインタ変数が構造体を指すには，例えばポインタ変数名を sp，構造体変数名を str とすると，sp = &str; という代入を行えばよい。このとき，str にはメンバが存在するが，例えばメンバの一つが x であるとすると，str.x をポインタ変数で表すには，(*sp).x となる。これを*sp.x と書くと，*(sp.x)を意味することとなり，誤りである。なお，(*sp).x は簡潔に sp->x と表記することができる。

例題 1（構造体の再定義名；三角形及び四角形を扱う構造体）

次のソースプログラムをテキストエディタで入力して，prog15-1.c の名前を付けて保存する。それを翻訳・編集して実行形式のファイルを作成し，実行せよ。(3 回実行し，ガイドコメントに従って，それぞれ 0，1，2 を入力する。また，長さや高さには実数値を入力する。)

```
/* prog15-1.c */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```
typedef struct
{
    char   n[7];      /* name (名称)*/
    double    s;      /* side (辺)*/
    double    h;      /* height (高さ)*/
    double    a;      /* area (面積)*/
} TRIRECT;            /* triangle/rectangle (三角形/四角形)*/

TRIRECT getArea(TRIRECT, int);

int main(void)
{

    TRIRECT figure = {"", 0.0, 0.0, 0.0};
    int type;

    printf("図形が三角形であれば 0, 長方形であれば 1, "
     "正方形であれば 2 を入力してください : ");
    scanf(" %d", &type);

    if (type == 0)
    {
        printf("三角形の底辺の長さ(実数値)を入力してください : ");
        scanf(" %lf", &figure.s);
        printf("三角形の高さ(実数値)を入力してください : ");
        scanf(" %lf", &figure.h);
    }
    else if (type == 1)
    {
        printf("長方形の一辺 X の長さ(実数値)を入力してください : ");
        scanf(" %lf", &figure.s);
        printf("長方形の一辺 Y の長さ(実数値)を入力してください : ");
        scanf(" %lf", &figure.h);
    }
    else if (type == 2)
    {
        printf("正方形の一辺の長さ(実数値)を入力してください : ");
        scanf(" %lf", &figure.s);
        figure.h = figure.s;
    }
    else
    {
        printf("入力する数値は 0, 1, 2 のいずれかです。");
        exit(1);
    }

    figure = getArea(figure, type);

    printf("%s の面積は%f です。¥n", figure.n, figure.a);

    return 0;
}

TRIRECT getArea(TRIRECT fig, int t)
{
    if (t == 0)
```

```
    {
        strncpy(fig.n, "三角形", 6);
        fig.a = fig.s * fig.h / 2;
    }
    else
    {
        fig.a = fig.s * fig.h;

        if (t == 1)
            strncpy(fig.n, "長方形", 6);
        else
            strncpy(fig.n, "正方形", 6);
    }

    return fig;
}
```

【解説】

1. TRIRECT figure は，宣言の際に初期化を行っている。

2. strncpy は，第 2 引数に書かれた文字列を第 3 引数に書かれた文字数分だけ，第 1 引数に複写する文字列操作のライブラリ関数である。

例題 2（**構造体を指すポインタ変数**）

次のソースプログラムをテキストエディタで入力して，prog15-2.c の名前を付けて保存する。それを翻訳・編集して実行形式のファイルを作成し，実行せよ。

```
/* prog15-2.c */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX 5
#define MAXBUF 256

typedef struct
{
    char         id[8];
    char      name[21];
    char      class[9];
    char      point[4];
    char       eval[2];
} STUDENT;

void dispdata(STUDENT *, int);

int main(void)
{
```

```
    STUDENT prog[MAX];

    int i;
    char temp[MAXBUF];
    char headln[MAXBUF];
    FILE *fpr;

    if (NULL == (fpr = fopen("stprog.csv","r")))
    {
        printf("stprog データファイルが開けません。¥n");
        exit(1);
    }

    fgets(headln, sizeof headln, fpr);

    printf("   %s", headln);

    for (i=0; i<MAX; i++)
    {
        if (fgets(temp, sizeof temp, fpr) == NULL) break;

        strcpy(prog[i].id, strtok(temp,","));
        strcpy(prog[i].name, strtok(NULL,","));
        strcpy(prog[i].class, strtok(NULL,","));
        strcpy(prog[i].point, strtok(NULL,","));
        strcpy(prog[i].eval, strtok(NULL,"¥n"));

        dispdata(&prog[i], i);
    }

    fclose(fpr);

    return 0;
}

void dispdata(STUDENT *data, int i)
{
        printf("%d) %s,%s,%s,%s,%s¥n", i+1,
          data->id, data->name, data->class, data->point, data->eval);
}
```

【解説】

prog14-2.c では，strtok で指定した文字列を",¥n"として，各項目がカンマで区切られている場合と項目ごとに改行されている場合，1 レコードの最後の項目の後ろにもカンマが付いている場合に対応していたが，このプログラムでは prog14-1.exe を正しく実行して作成されたファイル stprog.csv の存在を前提としている。

演習 1

100 人以内の住所録データを入力し，出力用のファイル"add.csv"を開いて書き込むプログラムを考える。この空欄 1)　　，2)　　，3)　　を埋めてソースプログラムを完成させ，テキストエディタで入力して，ex15-1.c の名前を付けて保存する。それを翻訳・編集して実行形式のファイルを作成し，実行せよ。なお，このプログラムでは，データの番号を入力する際に空打ちをする(文字を打たずに Enter キーを打つ)ことで，入力を途中で打ち切ることが可能である。

```
/* ex15-1.c */

#include <stdio.h>
#include <stdlib.h>

#define MAX 100

1)      struct
{
    char bango[4];
    char namae[21];
    char furigana[31];
    char yubin[9];
    char add1[51];
    char add2[41];
    char tel[13];
} ADDRESS;

int key_input(2)      *);

void w_file(ADDRESS *, int);

int main(void)
{
    ADDRESS jusho[MAX];
    ADDRESS *ptr = jusho;
    int i;

    for(i=0; i<MAX; i++, ptr++)
        if(!key_input(ptr)) break;

    w_file(jusho, i);

    return 0;
}

int key_input(ADDRESS *in_data)
{
    printf("データの番号を入力してください。:");
    gets(in_data->bango);

    if (in_data->bango[0] == '¥0') return 0;

    printf("氏名を入力してください。:");
    gets(in_data->namae);
    printf("ふりがなを入力してください。:");
```

```
    gets(in_data->furigana);
    printf("郵便番号を入力してください。:");
    gets(in_data->yubin);
    printf("住所を入力してください。:");
    gets(in_data->add1);
    printf("アパート名・部屋番号等を入力してください。:");
    gets(in_data->add2);
    printf("電話番号を入力してください。:");
    gets(in_data->tel);

    return 1;
}

void w_file(ADDRESS *out_data, int wi)
{
    FILE *fout;
    int j;

    if(NULL == (fout=fopen("3)      ","w")))
    {
        printf("ファイルが開けません。");
        exit(1);
    }

    for(j=0; j<wi;j++, out_data++)
    {
        fprintf(fout,"%3s,%-s,%-s,%-s,%-s,%-s,%-s¥n",
          out_data->bango, out_data->namae, out_data->furigana,
          out_data->yubin, out_data->add1, out_data->add2, out_data->tel);
    }

    fclose(fout);
}
```

演習 2（**余裕のある人向け**）

ダウンロードした Prog1_15ex.pdf に記述された課題を解き，ダウンロードした解答用ファイル Prog1_15ex_Ans.docx に解答を書き込んだもの及び csv ファイル programs.csv を提出せよ。

提出物：

1) 例題 1，2 の出力結果をコピーして貼り付けたテキストファイル res15.txt

2) 演習 1 のソースプログラムのファイル ex15-1.c の完成版

3) csv ファイル add.csv