## 構造体と関数

以前に「関数」を学んだとき、関数間でデータを受け渡す方法として引数や返却値について習った。この引数や返却値に構造体を利用することができる。関数間では上位の関数（呼ぶ側）から下位の関数（呼ばれる側）へ引数を利用することでデータを渡すことができるが、その方法には、「データの値」を渡す方法と、「データのアドレス」を渡す方法があった。構造体も通常の変数のように「値による呼び出し」と「参照による呼び出し」の２つの方法で引数を利用できる。

## 構造体と値による呼び出し

値による呼び出しでは、上位の関数で定義した構造体変数をコピーして下位の関数に渡すことになる。下位の関数で構造体のメンバ引数に値を代入したとしても上位の関数の構造体のメンバに影響がでない。これを関数の独立性が高いと言う。独立性の高い関数は再利用がしやすい。返却値の利用と組み合わせることでプログラムのどこで値を変更しているかを限定できるのでプログラムの管理がしやすくなる。大規模なプログラムを作成する場合には有効な手段である。ただし、注意点があり、この場合には対象となる構造体の内容をすべてコピーすることになるため、構造体のサイズが大きい場合にはメモリ量や処理時間が多くなる。それでは具体的な使用例を以下に示す。

プログラム例１

```
#include <stdio.h>

// 構造体のテンプレート（枠型）の宣言
typedef struct record                       // 成績
{
        int number;                         // 番号
        char name[20];                      // 名前
        double average;                     // 平均点
}RECORD;

// 関数のプロトタイプ宣言
void disp_struct(RECORD stu);

int main(void)
{
        int i;

        // 構造体の宣言と初期化
        RECORD student1 = { 1, "Satou", 90.2 };
        RECORD student2[3] = {
                { 2, "Suzuki", 77.3 },
                { 3, "Takahashi", 59.1 },
                { 4, "Tanaka", 64.9 }
        };

        // 構造体メンバの表示
        disp_struct(student1);

        for(i = 0; i < 3; i++)
                disp_struct(student2[i]);
```

```
        return 0;
}

// 構造体メンバを表示する関数
void disp_struct(RECORD stu)
{
        printf("%d %s %.1f¥n", stu.number, stu.name, stu.average);
}
```

このプログラムでは、構造体メンバを表示する関数に値渡し（値による呼び出し）でデータを渡している。構造体変数と構造体配列の要素のそれぞれが実引数（このプログラムでいう関数 main()での関数 disp_struct()の引数）として用いることができるように関数 disp_struct()を定めた。

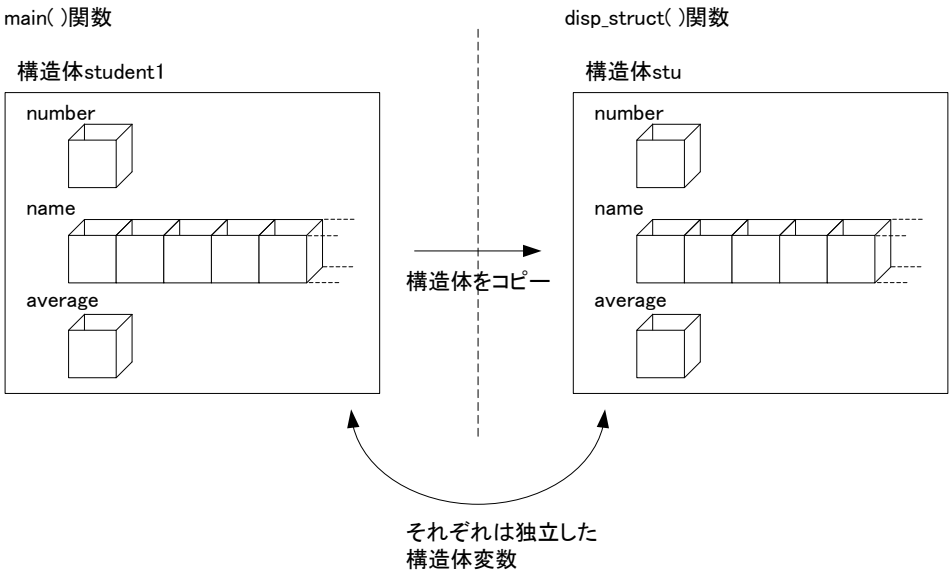


図１　構造体と値による呼び出し

### 構造体と参照による呼び出し

参照による呼び出しでは構造体をコピーせず、構造体の先頭アドレスを引数として渡すため、下位の関数における値の変更が上位の関数に反映される。また、構造体のサイズに関わらず、アドレス分のメモリだけを使うため、プログラム実行時のメモリ量と処理速度を軽減できる。それでは具体的な使用例を以下に示す。

プログラム例２

```
#include <stdio.h>

// 構造体のテンプレート（枠型）の宣言
typedef struct record              // 成績
{
        int number;                // 番号
        char name[20];             // 名前
        double average;            // 平均点
}RECORD;
```

```
// 関数のプロトタイプ宣言
void input_struct(int i, RECORD *stu);
void disp_struct(RECORD stu);

int main(void)
{
        int i;

        // 構造体の宣言
        RECORD student1, student2[3];

        // 構造体メンバへの入力
        input_struct(1, &student1);

        for(i = 0; i < 3; i++)
                input_struct(i+2, &student2[i]);


        // 構造体メンバの表示
        disp_struct(student1);

        for(i = 0; i < 3; i++)
                disp_struct(student2[i]);

        return 0;
}

// 構造体メンバに値を入力する関数
void input_struct(int i, RECORD *stu)
{
        printf("学生%d¥n", i);
        printf("学籍番号:");
        scanf("%d", &stu->number);
        printf("名前:");
        scanf("%s", stu->name);
        printf("平均値:");
        scanf("%lf", &stu->average);
}

// 構造体メンバを表示する関数
void disp_struct(RECORD stu)
{
        printf("%d %s %.1f¥n", stu.number, stu.name, stu.average);
}
```

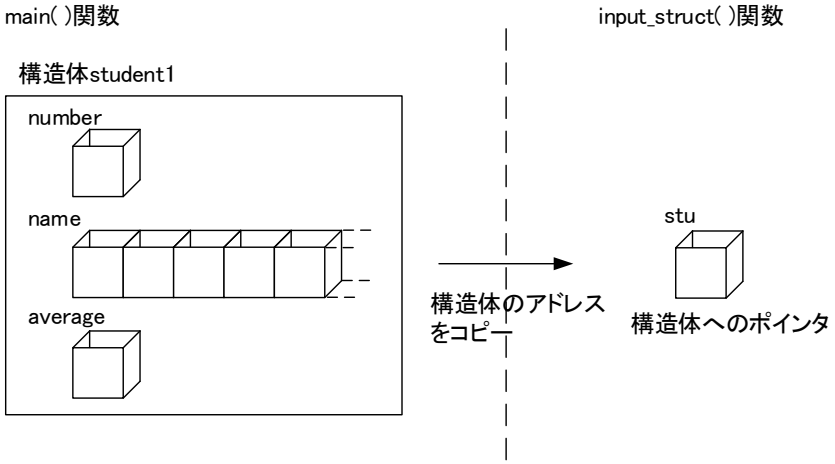


図２　構造体と参照による呼び出し

**構造体と返却値**

通常の変数と同じように return 文を使用して関数の返却値に構造体を用いることができる。関数 input_struct()の終了時に return 文で構造体 stu を関数 main()に返却している部分がこれにあたる。

プログラム例３

```
#include <stdio.h>

// 構造体のテンプレート（枠型）の宣言
typedef struct record    // 成績
{
        int number;      // 番号
        char name[20];   // 名前
        double average;  // 平均点
}RECORD;

// 関数のプロトタイプ宣言
RECORD input_struct(int i);       //返却値で行う
void disp_struct(RECORD stu);

int main(void)
{
        int i;

        // 構造体の宣言
        RECORD student1, student2[3];

        // 構造体メンバへの入力
        student1 = input_struct(1);

        for (i = 0; i < 3; i++)
                student2[i] = input_struct(i + 2);


        // 構造体メンバの表示
        disp_struct(student1);
```

```
        for (i = 0; i < 3; i++)
                disp_struct(student2[i]);

        return 0;
}

// 構造体メンバに値を入力する関数
RECORD input_struct(int i)
{
        RECORD stu;

        printf("学生%d¥n", i);
        printf("学籍番号:");
        scanf("%d", &stu.number);
        printf("名前:");
        scanf("%s", stu.name);
        printf("平均値:");
        scanf("%lf", &stu.average);

        return stu;
}

// 構造体メンバを表示する関数
void disp_struct(RECORD stu)
{
        printf("%d %s %.1f¥n", stu.number, stu.name, stu.average);
}
```

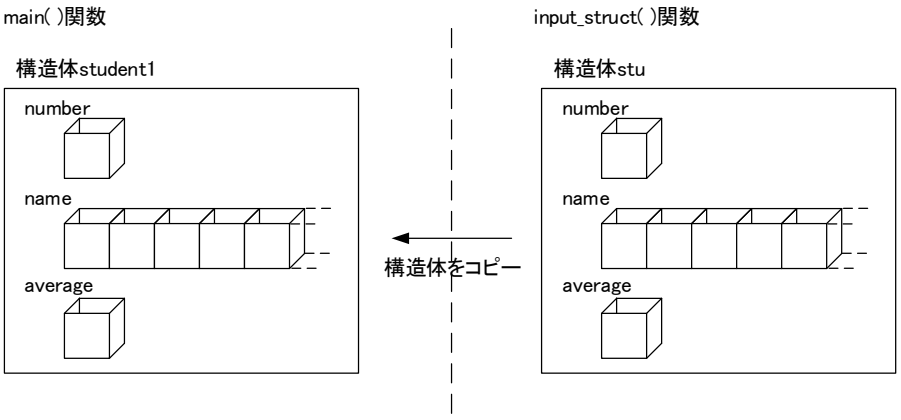


図３　構造体と返却値

演習　プログラム例１～３を作成し、実行結果を確認しなさい。

課題４の続き
Kadai4-3　プログラム例３を参考とする。次の仕様に従って、3 教科の点数から学生毎の平均点と評価を求め、実行結果例のように出力するプログラムを作成しなさい。

仕様
(1)構造体のメンバは以下の項目とする。

| | |
|---|---|
| int 型 | 学籍番号、英語、数学、物理 |
| double 型 | 平均点 |
| char 型 | 評価 |

(2)キーボードから学籍番号、英語・数学・物理の点数を入力する関数を作成すること。
(3)英語・数学・物理の 3 教科の平均点を求め、この平均点に対応する評価を設定する関数を作成すること。評価は以下の 4 段階とする。

| 平均点 | 評価 |
|---|---|
| 80 点以上 | A |
| 60 点以上 80 点未満 | B |
| 50 点以上 60 点未満 | C |
| 50 点未満 | D |

(4)学籍番号、平均点、評価を表示する関数を作成すること。

**実行結果例**

```
学生のデータを入力します
学籍番号:1
英語:67
数学:46
物理:52
学籍番号:2
英語:65
数学:56
物理:78
学籍番号:3
英語:98
数学:87
物理:92
学籍番号:4
英語:53
数学:36
物理:45

評価の結果を出力します
学籍番号:1 平均点:55.0 評価:C
学籍番号:2 平均点:66.3 評価:B
学籍番号:3 平均点:92.3 評価:A
学籍番号:4 平均点:44.7 評価:D
```