

# C言語10

構造体

# 構造体の記述

## 【例】

```
#include<stdio.h>

typedef struct Soseki{
    char title[30];
    int year;
}Soseki;

int main(void){
    Soseki soseki1 = {"吾輩は猫である", 1911};

    printf("題名 : %s\n", soseki1.title);
    printf("発行年 : %d\n", soseki1.year);

    return 0;
}
```

## 【実行結果】

```
題名 : 吾輩は猫である
発行年 : 1911
```

# 構造体の配列

```
#include<stdio.h>

typedef struct Soseki{
    char title[30];
    int year;
}Soseki;

int main(void){

    int i;
    Soseki soseki[3]={{ "吾輩は猫である", 1905},
                      {"坊ちゃん", 1906},
                      {"草枕", 1906}};

    for(i=0; i<3; i++){
        printf("題名 : %s¥n", soseki[i].title);
        printf("発行年 : %d¥n", soseki[i].year);
    }

    return 0;
}
```

## 【実行結果】

題名 : 吾輩は猫である  
発行年 : 1905  
題名 : 坊ちゃん  
発行年 : 1906  
題名 : 草枕  
発行年 : 1906

# 構造体の配列, 代入

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

```
typedef struct Soseki{
    char title[30];
    int year;
}Soseki;

int main(void){

    int i;

    Soseki soseki[3];
    strcpy(soseki[0].title, "吾輩は猫である"); soseki[0].year = 1905;
    strcpy(soseki[1].title, "坊ちゃん");    soseki[1].year = 1906;
    strcpy(soseki[2].title, "草枕");        soseki[2].year = 1906;

    for(i=0; i<3; i++){
        printf("題名 : %s\n", soseki[i].title);
        printf("発行年 : %d\n", soseki[i].year);
    }

    return 0;
}
```

## 【実行結果】

```
題名 : 吾輩は猫である
発行年 : 1905
題名 : 坊ちゃん
発行年 : 1906
題名 : 草枕
発行年 : 1906
```

# 構造体と関数

- 平面上の点として、整数の組(x, y)からなる構造体 struct Point を定義する。
- 2つの点pt1とpt2を加える関数addPointを定義する。このとき、2点を加えた新しい点pt3は、
$$pt3.x = pt1.x + pt2.x$$
$$pt3.y = pt1.y + pt2.y$$
を満たすものとする。
- 最初の例では、関数の引数には、値を渡します。2番目の例では、関数の引数には、ポインタを渡します。

# 値渡し

```
#include<stdio.h>
typedef struct Point{
    int x;
    int y;
}Point;

struct Point addPoint(Point pt1, Point pt2){
    struct Point pt3;
    pt3.x = pt1.x + pt2.x;
    pt3.y = pt1.y + pt2.y;
    return pt3;
}

int main(void){
    Point pt1={2, 3};
    Point pt2={5, 7};
    Point pt3;
    pt3 = addPoint(pt1, pt2);

    printf("pt1 = (%d, %d)\n", pt1.x, pt1.y);
    printf("pt2 = (%d, %d)\n", pt2.x, pt2.y);
    printf("pt3 = (%d, %d)\n", pt3.x, pt3.y);

    return 0;
}
```

## 【実行結果】

```
pt1 = (2, 3)
pt2 = (5, 7)
pt3 = (7, 10)
```

# ポインタ渡し

```
#include<stdio.h>
typedef struct Point{
    int x;
    int y;
}Point;

void addPoint(Point *pt1, Point *pt2, Point *pt3){
    pt3 -> x = pt1->x + pt2->x;
    pt3 -> y = pt1->y + pt2->y;
}

int main(void){
    Point pt1={2, 3};
    Point pt2={5, 7};
    Point pt3;
    addPoint(&pt1, &pt2, &pt3);
    printf("pt1 = (%d, %d)\n", pt1.x, pt1.y);
    printf("pt2 = (%d, %d)\n", pt2.x, pt2.y);
    printf("pt3 = (%d, %d)\n", pt3.x, pt3.y);

    return 0;
}
```

## 【実行結果】

pt1 = (2, 3)

pt2 = (5, 7)

pt3 = (7, 10)

# 構造体と配列と関数

```
#include<stdio.h>

typedef struct Novel{
    char title[30];
    int year;
}Novel;

struct Novel Soseki(int i){
    Novel soseki[3]={{ "吾輩は猫である", 1905},
                    {"坊ちゃん", 1906},
                    {"草枕", 1906}};
    return soseki[i];
}

int main(void){
    int i;

    for(i=0; i<3; i++){
        printf("題名 : %s¥n", Soseki(i).title);
        printf("発行年 : %d¥n", Soseki(i).year);
    }

    return 0;
}
```

## 【実行結果】

題名 : 吾輩は猫である

発行年 : 1905

題名 : 坊ちゃん

発行年 : 1906

題名 : 草枕

発行年 : 1906



# 構造体と並べ替え

```
#include<stdio.h>

typedef struct Pref{
    char name[30];
    int area;
    double pop;
}Pref;

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

int main(void){
    int n=3, i, j;

    Pref t3[3] = {"青森", 9645,133.5}, {"岩手", 15279,129.5}, {"秋田", 11636, 105.0}};

    printf("北東北3県\n");
    for(i=0; i<n ; i++)
        printf("%6s %6d km^2 %6.1f万人\n", t3[i].name, t3[i].area, t3[i].pop);

    for(i=0; i<n-1;i++){
        for(j=i+1; j<n ;j++){
            if(t3[i].area<t3[j].area)
                swap(&t3[i], &t3[j]);
        }
    }

    printf("面積の広い順\n");
    for(i=0; i<n ; i++)
        printf("%6s %6d km^2 %6.1f万人\n", t3[i].name, t3[i].area, t3[i].pop);

    return 0;
}
```

## 【実行結果】

北東北3県

青森 9645 km<sup>2</sup> 133.5万人

岩手 15279 km<sup>2</sup> 129.5万人

秋田 11636 km<sup>2</sup> 105.0万人

面積の広い順

岩手 15279 km<sup>2</sup> 129.5万人

秋田 11636 km<sup>2</sup> 105.0万人

青森 9645 km<sup>2</sup> 133.5万人

# 文字コードと並べ替え

```
#include<stdio.h>
typedef struct Pref{
    unsigned char  name[30];
    unsigned char  kana[30];
}Pref;
void swap(Pref *x, Pref *y){
    Pref tmp = *x;
    *x = *y;
    *y = tmp;
}

int main(void){
    int n=3, i, j;
    Pref t3[3] = {"青森", "あおもり"}, {"岩手", "いわて"}, {"秋田", "あきた"};
    printf("北東北3県\n");
    printf(" 漢字 ひらがな: ひらがな最初2文字のコード\n");
    for(i=0; i<n ; i++)
        printf("%6s %8s: %d %d %d %d\n", t3[i].name,  t3[i].kana,
            t3[i].kana[0], t3[i].kana[1],
            t3[i].kana[2], t3[i].kana[3]);

    for(i=0; i<n-1;i++){
        for(j=i+1; j<n ;j++){
            if(t3[i].kana[1] > t3[j].kana[1]){
                swap(&t3[i], &t3[j]);
            }
            else if(t3[i].kana[1] == t3[j].kana[1]){
                if(t3[i].kana[3] > t3[j].kana[3])
                    swap(&t3[i], &t3[j]);
            }
        }
    }

    printf("あいうえお順\n");
    for(i=0; i<n ; i++)
        printf("%6s %6s %n", t3[i].name, t3[i].kana);

    return 0;
}
```

## 【実行結果】

北東北3県

漢字 ひらがな: ひらがな最初2文字のコード

青森 あおもり: 130 160 130 168

岩手 いわて: 130 162 130 237

秋田 あきた: 130 160 130 171

あいうえお順

青森 あおもり

秋田 あきた

岩手 いわて

【注】 0番目と2番目のコードの値は同じなので、  
まず、1番目を比較し、次に3番目を比較する。

# 構造体のネスト(入れ子)

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

```
typedef struct Point{  
    int x;  
    int y;  
}Point;
```

```
typedef struct Triangle{  
    Point p1;  
    Point p2;  
    Point p3;  
}Triangle;
```

```
int main(void){
```

```
    // 構造体の初期化  
    Point point1={1, 2};  
    Point point2={4, 3};  
    Point point3={2, 5};
```

```
    Triangle t1;
```

```
    //構造体への代入  
    t1.p1=point1;  
    t1.p2=point2;  
    t1.p3=point3;
```

```
    printf("%d, %d\n", t1.p1.x, t1.p1.y);  
    printf("%d, %d\n", t1.p2.x, t1.p2.y);  
    printf("%d, %d\n", t1.p3.x, t1.p3.y);
```

```
    return 0;  
}
```

**【実行結果】**

1, 2

4, 3

2, 5

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

## 構造体のネスト(2) 構造体の配列

```
typedef struct Laureate{ // 受賞者  
    char name[30];  
    char cat[30]; //cat = category  
}Laureate;
```

```
typedef struct NobelPrize{  
    int year;  
    Laureate lt[30];  
}NobelPrize;
```

```
int main(void){  
    int i;
```

```
    Laureate l1[2] = {"大村智", "生理学医学賞"}, {"梶田隆章", "物理学"};  
    NobelPrize np1;
```

```
    np1.year=2015;
```

```
    for(i=0; i<2; i++)  
        np1.lt[i] =l1[i];
```

```
    printf("ノーベル賞%d年 ¥n", np1.year);  
    for(i=0; i<2; i++){  
        printf(" %8s %s¥n", np1.lt[i].name, np1.lt[i].cat);  
    }
```

```
    return 0;
```

```
}
```

### 【実行結果】

ノーベル賞2015年

大村智 生理学医学賞

梶田隆章 物理学

# 構造体のネスト(3)構造体の多重配列

```
#include<stdio.h>

typedef struct Laureate{
    char name[30];
    char cat[30]; //cat = category
}Laureate;

typedef struct NobelPrize{
    int year;
    int n; //受賞者数
    Laureate lt[30]; //受賞者
}NobelPrize;

int main(void){
    int i, j;

    Laureate nl[4][3]={ {"根岸英一", "化学賞"}, {"鈴木章", "化学賞"}, {"END", ""},
                        {"山中伸弥", "生理学・医学賞"},
                        {"赤崎勇", "物理学賞"}, {"天野浩", "物理学賞"}, {"中村修二", "物理学賞"},
                        {"大村智", "生理学・医学賞"}, {"梶田隆章", "物理学賞"};

    NobelPrize np[4];

    np[0].year=2010; np[0].n=2;
    np[1].year=2012; np[1].n=1;
    np[2].year=2014; np[2].n=3;
    np[3].year=2015; np[3].n=3;

    for(i=0; i<4; i++){
        for(j=0; j<3; j++){
            np[i].lt[j] =nl[i][j];
        }
    }

    for(i=0; i<4; i++){
        printf("ノーベル賞%d年 %n", np[i].year);
        for(j=0; j< np[i].n; j++){
            printf(" %8s %s%n", np[i].lt[j].name, np[i].lt[j].cat);
        }
    }

    return 0;
}
```

## 【実行結果】

```
ノーベル賞2010年
  根岸英一 化学賞
  鈴木章 化学賞
ノーベル賞2012年
  山中伸弥 生理学・医学賞
ノーベル賞2014年
  赤崎勇 物理学賞
  天野浩 物理学賞
  中村修二 物理学賞
ノーベル賞2015年
  大村智 生理学・医学賞
  梶田隆章 物理学賞
```