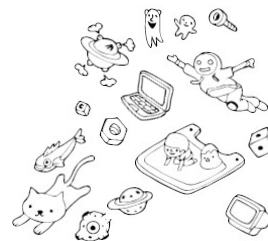


Chapter 22. 구조체 part 1



구조체란 무엇인가?

구조체의 필요성

- ▶ 구조체 (struct) ← “사용자 정의 자료형”
 - ▶ 서로 다른 자료형의 변수들을 묶어서 만든 하나의 새로운 자료형
 - ▶ 구조체 정의 방법
 - ▶ 키워드 **struct**를 이용하여 구조체 태그 이름을 기술하고 중괄호 사이에 변수들을 선언
 - ▶ 구조체를 정의하는 구문도 C 언어의 문장이기 때문에 중괄호 다음 마지막에 **세미콜론(;)을 반드시 기술해야 한다.**
 - ▶ 구조체 정의: 변수 선언이 아니므로 메모리에 방이 만들어지지 않음 → 구조체 변수 선언은 어떻게??

```
struct book {
```

```
    char title[50]; //제목
    char author[50]; //저자
    char publish[50]; //출판사
    int pages; //페이지수
    int price; //가격
};
```

book은 구조체 태그 이름으로 변수 선언 시 이 용할 새로운 자료형이다.

구조체를 구성하는 구조체의 멤버이며, 일반 변수의 선언과 같다.

```
int a;
char b;
float c;
struct book s;
```

3

구조체의 정의

```
int xpos; // 마우스의 x 좌표
int ypos; // 마우스의 y 좌표
```

마우스의 좌표정보를 저장하고 관리하기 위해서는 x좌표와 y좌표를 저장할 수 있는 두 개의 변수가 필요하다.

xpos와 ypos는 서로 독립된 정보를 표현하지 않고 하나의 정보를 표현한다. 따라서 이 둘은 늘 함께한다.

```
struct point // point라는 이름의 구조체 정의
{
    int xpos; // point 구조체를 구성하는 멤버 xpos
    int ypos; // point 구조체를 구성하는 멤버 ypos
};
```

구조체를 이용해서 xpos와 ypos를 하나로 묶었다. 이 둘을 묶어서 point라는 이름의 새로운 자료형을 정의!

int가 자료형의 이름인것 처럼 point도 자료형의 이름이다.

단, 프로그래머가 정의한 자료형이기에 ‘사용자 정의 자료형(user defined data type)’이라 한다.

```
struct person
{
    char name[20]; // 이름 저장
    char phoneNum[20]; // 전화번호 저장
    int age; // 나이 저장
};
```

개인의 이름과 전화번호 나이 정보를 person이라는 구조체 정의를 통해서 묶고 있다.

배열도 구조체의 멤버로 선언이 가능!

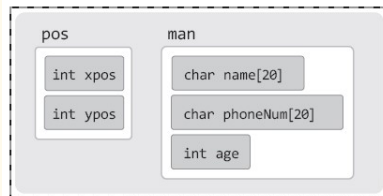
구조체 변수의 선언과 접근

구조체 변수선언의 기본 형태

```
struct type_name val_name;
```

```
struct point pos;
struct person man;
```

↓ 구조체 변수선언의 예



구조체 변수선언의 결과

구조체 멤버의 접근방식

구조체 변수의 이름. 구조체 멤버의 이름

```
pos.xpos=20;
printf("%s\n", man.name);
```

구조체 변수 pos의 멤버 xpos에 20을 저장
man의 멤버 name에 저장된 문자열 출력

구조체 변수의 선언과 접근관련 예제1

```

struct point // 구조체 point의 정의
{
    int xpos;
    int ypos;
};

int main(void)
{
    struct point pos1, pos2;
    double distance;
    fputs("point1 pos: ", stdout);
    scanf("%d %d", &pos1.xpos, &pos1.ypos);
    fputs("point2 pos: ", stdout);
    scanf("%d %d", &pos2.xpos, &pos2.ypos);

    /* 두 점간의 거리 계산 공식 */
    distance=sqrt((double)((pos1.xpos-pos2.xpos) * (pos1.xpos-pos2.xpos)+
        (pos1.ypos-pos2.ypos) * (pos1.ypos-pos2.ypos)));

    printf("두 점의 거리는 %g 입니다. \n", distance);
    return 0;
}
  
```

TwoPointDistance.c

이 예제에서 호출하는 함수 sqrt는 제곱근을 반환하는 함수로써
헤더파일 **math.h**에 선언된 수학관련 함수이다.

실행결과

```

point1 pos: 1 3
point2 pos: 4 5
두 점의 거리는 3.60555 입니다.
  
```

구조체 변수의 선언과 접근관련 예제2

```
struct person
```

```
{
    char name[20];
    char phoneNum[20];
    int age;
};
```

```
int main(void)
```

```
{
    struct person man1, man2;
    strcpy(man1.name, "안성준");
    strcpy(man1.phoneNum, "010-1122-3344");
    man1.age=23;
    printf("이름 입력: "); scanf("%s", man2.name);
    printf("번호 입력: "); scanf("%s", man2.phoneNum);
    printf("나이 입력: "); scanf("%d", &(man2.age));
    printf("이름: %s \n", man1.name);
    printf("번호: %s \n", man1.phoneNum);
    printf("나이: %d \n", man1.age);
    printf("이름: %s \n", man2.name);
    printf("번호: %s \n", man2.phoneNum);
    printf("나이: %d \n", man2.age);
    return 0;
}
```

TelephoneInfo.c

구조체의 멤버라 하더라도 일반적인 접근의 방식을 그대로 따른다. 구조체의 멤버로 배열이 선언되면 배열의 접근방식을 취하면 되고, 구조체의 멤버로 포인터 변수가 선언되면 포인터 변수의 접근방식을 취하면 된다.

```
이름 입력: 김수정
번호 입력: 010-0001-0002
나이 입력: 27
이름: 안성준
번호: 010-1122-3344
나이: 23
이름: 김수정
번호: 010-0001-0002
나이: 27
```

실행결과

구조체 정의와 동시에 변수 선언하기

```
struct point
{
    int xpos;
    int ypos;
} pos1, pos2, pos3;
```

point라는 이름의 구조체를 정의함과 동시에 point 구조체의 변수 pos1, pos2, pos3를 선언하는 문장이다.

```
struct point
{
    int xpos;
    int ypos;
};
struct point pos1, pos2, pos3;
```

위와 동일한 결과를 보이는 구조체의 정의와 변수의 선언이다.

구조체를 정의함과 동시에 변수를 선언하는 문장은 잘 사용되지 않는다.
그러나 문법적으로 지원이 되고 또 간혹 사용하는 경우도 있다.

구조체 변수의 초기화

```

struct point
{
    int xpos;
    int ypos;
};
struct person
{
    char name[20];
    char phoneNum[20];
    int age;
};
int main(void)
{
    struct point pos={10, 20};
    struct person man={"이승기", "010-1212-0001", 21};
    printf("%d %d \n", pos.xpos, pos.ypos);
    printf("%s %s %d \n", man.name, man.phoneNum, man.age);
    return 0;
}

```

InitStructVal.c

초기화 방식이 배열과 유사하다.

초기화 할 데이터들을 중괄호 안에 순서대로 나열하면 된다. .

실행결과

```

10 20
이승기 010-1212-0001 21

```

구조체 정의 구문 위치

- ▶ 구조체 정의 구문은 변수의 범위와 비슷
 - ▶ 다음 구문은 **메인 함수 내부에서 구조체를 정의**했으므로 구조체 자료형 **struct book**은 **메인 함수 내부에서만 이용**이 가능
 - ▶ 구조체 정의 구문의 위치에 따라 구조체 선언 시 자료형 이용의 범위가 정해짐

```
int main(void)
```

```
{
```

```

    struct book {
        char title[50]; //제목
        char author[50]; //저자
        char publish[50]; //출판사
        int pages; //페이지수
        int price; //가격
    };

```

```
    struct book mybook
```

```
    ...
```

```
}
```

구조체의 이용은 구조체 정의 문장 이후, 함수 main()에서만 가능하다.

구조체의 저장공간

▶ 구조체 저장 공간 모양

- ▶ 구조체 `book`의 변수인 `mybook`은 멤버의 여러 자료형이 하나로 묶인 변수로서 다음과 같은 구조로 메모리에 할당

변수 `mybook`

<code>title[50]</code>	0	1	2	...	48	49
<code>author[50]</code>	0	1	2	...	48	49
<code>publish[50]</code>	0	1	2	...	48	49
<code>pages</code>						
<code>price</code>						

- ▶ 위 변수의 크기는 각 멤버의 크기의 합($50+50+50+4+4 = 158$)으로 158 바이트 정도
- ▶ 연산자 `sizeof`를 이용하여 변수의 크기 또는 구조체의 크기를 알 수 있음
 - ▶ `sizeof (struct book)`을 `printf`하면 그 값이 실제 할당 크기임
 - ▶ `sizeof` 연산자를 이용하면 예상한 구조체의 크기보다 큰 값이 나올 수 있으며 시스템마다 차이가 있을 수 있음

11

실습문제

- ▶ 다음 사항을 프로그래밍 하시오.
 - ▶ 이름, 나이, 키, 몸무게로 구성된 구조체 형을 하나 선언하시오. (`name`, `age`, `height`, `weight`)
 - ▶ `name`은 `char` 배열, `age`는 `int`, `height` 및 `weight`는 `float`로 선언
 - ▶ 위 구조체형 변수로 `my` 및 `you`를 선언
 - ▶ keyboard로부터 이름, 나이, 키, 몸무게를 입력 받아 `my`에 저장
 - ▶ 나이, 키, 몸무게의 값에 10씩 더해서 `you`에 저장 후 `you`의 내용을 출력
 - ▶ 입출력 결과는 아래와 같음

```
이름을 입력하세요 : 홍길동
나이를 입력하세요 : 21
키를 입력하세요 : 175
몸무게를 입력하세요 : 60
이름 : 홍길동
나이 : 31
키 : 185
몸무게 : 70
```

12

구조체 배열의 선언과 접근

```

struct point
{
    int xpos;
    int ypos;
};

int main(void)
{
    struct point arr[3];
    int i;
    for(i=0; i<3; i++)
    {
        printf("점의 좌표 입력: ");
        scanf("%d %d", &arr[i].xpos, &arr[i].ypos);
    }
    for(i=0; i<3; i++)
        printf("[%d, %d] ", arr[i].xpos, arr[i].ypos);
    return 0;
}

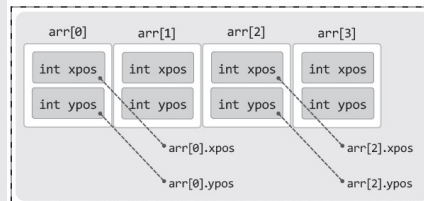
```

StructArray.c

```
struct point arr[4];
```

길이가 4인 구조체 배열의 선언방법

선언된 배열의 형태



실행결과

```

점의 좌표 입력: 2 4
점의 좌표 입력: 3 6
점의 좌표 입력: 8 9
[2, 4] [3, 6] [8, 9]

```

구조체 배열의 초기화

```
struct person man={"이승기", "010-1212-0001", 21};
```

구조체 변수의 초기화

구조체 변수 하나를 초기화하기 위해서 하나의 중괄호를 사용하듯이...

```

struct person arr[3]={
    {"이승기", "010-1212-0001", 21}, // 첫 번째 요소의 초기화
    {"정지영", "010-1313-0002", 22}, // 두 번째 요소의 초기화
    {"한지수", "010-1717-0003", 19}  // 세 번째 요소의 초기화
};

```

구조체 배열의 초기화

구조체 배열을 초기화하기 위해서 배열요소 각각의 초기화 값을 중괄호로 묶어서 표현한다.

구조체 배열의 초기화 예제

```

struct person
{
    char name[20];
    char phoneNum[20];
    int age;
};

int main(void)
{
    struct person arr[3]={
        {"이승기", "010-1212-0001", 21}, // 첫 번째 요소의 초기화
        {"정지영", "010-1313-0002", 22}, // 두 번째 요소의 초기화
        {"한지수", "010-1717-0003", 19}  // 세 번째 요소의 초기화
    };

    int i;
    for(i=0; i<3; i++)
        printf("%s %s %d \n", arr[i].name, arr[i].phoneNum, arr[i].age);

    return 0;
}

```

InitStructArray.c

실행결과

```

이승기 010-1212-0001 21
정지영 010-1313-0002 22
한지수 010-1717-0003 19

```

구조체 변수와 포인터

```

struct point pos={11, 12};
struct point * pptr=&pos;
    구조체 point의 포인터 변수 선언
(*pptr).xpos=10;
    pptr이 가리키는 구조체 변수의 멤버 xpos에 접근
(*pptr).ypos=20;
    pptr이 가리키는 구조체 변수의 멤버 ypos에 접근

```

```

(*pptr).xpos=10; ↔ pptr->xpos=10;
(*pptr).ypos=20; ↔ pptr->ypos=20;

```

“* 연산과 . 연산” 대신에 “->연산” 사용

구조체 변수와 포인터 관련 예제

```

struct point
{
    int xpos;
    int ypos;
};

int main(void)
{
    struct point pos1={1, 2};
    struct point pos2={100, 200};
    struct point * pptr=&pos1;

    (*pptr).xpos += 4;
    (*pptr).ypos += 5;
    printf("[%d, %d] \n", pptr->xpos, pptr->ypos);

    pptr=&pos2;
    pptr->xpos += 1;
    pptr->ypos += 2;
    printf("[%d, %d] \n", (*pptr).xpos, (*pptr).ypos);
    return 0;
}

```

StructPointer.c

프로그래머들이 주로 사용하는 연산자이니
-> 연산자의 사용에 익숙해지자.

실행결과

```

[5, 7]
[101, 202]

```

포인터 변수를 구조체의 멤버로 선언하기1

```

struct point
{
    int xpos;
    int ypos;
};

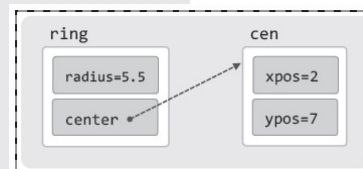
struct circle
{
    double radius;
    struct point * center;
};

int main(void)
{
    struct point cen={2, 7};
    double rad=5.5;
    struct circle ring={rad, &cen};
    printf("원의 반지름: %g \n", ring.radius);
    printf("원의 중심 [%d, %d] \n", (ring.center)->xpos, (ring.center)->ypos);
    return 0;
}

```

StructPointerMem.c

구조체 변수의 멤버로 구조체 포인터 변수가 선언될 수 있다!



실행결과

```

원의 반지름: 5.5
원의 중심 [2, 7]

```

포인터 변수를 구조체의 멤버로 선언하기2

```

struct point
{
    int xpos;
    int ypos;
    struct point * ptr;
};

int main(void)
{
    struct point pos1={1, 1};
    struct point pos2={2, 2};
    struct point pos3={3, 3};

    pos1.ptr = &pos2; // pos1과 pos2를 연결
    pos2.ptr = &pos3; // pos2와 pos3를 연결
    pos3.ptr = &pos1; // pos3를 pos1과 연결

    printf("점의 연결관계... \n");
    printf("[%d, %d]와(과) [%d, %d] 연결 \n",
        pos1.xpos, pos1.ypos, pos1.ptr->xpos, pos1.ptr->ypos);
    printf("[%d, %d]와(과) [%d, %d] 연결 \n",
        pos2.xpos, pos2.ypos, pos2.ptr->xpos, pos2.ptr->ypos);
    printf("[%d, %d]와(과) [%d, %d] 연결 \n",
        pos3.xpos, pos3.ypos, pos3.ptr->xpos, pos3.ptr->ypos);
    return 0;
}

```

PointerRelation.c

type형 구조체 변수의 멤버로 type형 포인터 변수를 둘 수 있다.

실행결과

점의 연결관계...

[1, 1]와(과) [2, 2] 연결

[2, 2]와(과) [3, 3] 연결

[3, 3]와(과) [1, 1] 연결

구조체 변수와 첫 번째 멤버의 주소 값

```

struct point
{
    int xpos;
    int ypos;
};

struct person
{
    char name[20];
    char phoneNum[20];
    int age;
};

int main(void)
{
    struct point pos={10, 20};
    struct person man={"이승기", "010-1212-0001", 21};

    printf("%p %p \n", &pos, &pos.xpos);
    printf("%p %p \n", &man, man.name);
    return 0;
}

```

StructValAddress.c

구조체 변수의 주소 값과 구조체 변수의 첫 번째 멤버의 주소 값은 일치한다.
응용 프로그램 분야에서는 이 사실을 이용해서 프로그램을 작성하기도 한다.

실행결과

003EF7B8 003EF7B8

003EF784 003EF784

실습 문제 (Lab 1) → 생략

- ▶ (1) 다음과 같은 employee 구조체를 만들 것
 - ▶ name: 종업원 이름 저장을 위한 문자열 배열
 - ▶ jumin: 종업원 주민번호 저장을 위한 문자열 배열
 - ▶ salary: 급여 정보 저장을 위한 정수형 변수
- ▶ (2) 길이가 3인 employee 배열을 선언
- ▶ (3) 세 명의 정보를 프로그램 사용자로부터 입력 받아서 저장한 다음, 배열에 저장된 데이터를 순서대로 출력하는 예제를 작성할 것

23

실습 문제 (Lab 2)

기존 숙박 관리 프로그램을 **구조체**를 사용하는 형태로 확장 → **Version 4!**

- ▶ (1) 다음과 같은 hotel 구조체를 만들 것
 - ▶ name: 고객 이름 저장을 위한 문자열 배열
 - ▶ juso: 고객 주소 저장을 위한 문자열 배열
 - ▶ pnum: 고객 연락처 저장을 위한 문자열 배열
- ▶ (2) 메뉴는 다음과 같이 확장
 - ▶ 1. 입실, 2. 퇴실, 3. 전체 조회, 4. 객실 조회, 5. 종료
- ▶ (3) 기능 설명
 - ▶ "3. 전체 조회": 고객 이름만 출력
 - ▶ "4. 객실 조회": 고객 정보를 출력
- ▶ [주의]
 - ▶ 문자입력: scanf() → gets()
 - ▶ 초기화: strcpy(room[i][j].name, "W0")

```

***** 숙박관리 프로그램 *****
1.입실, 2.퇴실, 3.전체조회, 4.객실조회, 5.종료: 1
출 번호 입력(1-3출): 2
방 번호 입력(1-5호): 2
고객이름: 장재신
주소: 경남 김해시 인제로 197 인제대학교
전화번호: 010-9658-8138

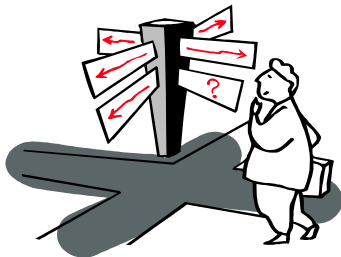
1.입실, 2.퇴실, 3.전체조회, 4.객실조회, 5.종료: 3
O              O              O              O              O
O              O              O              O              O
O              O              O              O              O
1.입실, 2.퇴실, 3.전체조회, 4.객실조회, 5.종료: 4
출 번호 입력(1-3출): 2
방 번호 입력(1-5호): 2
***** 2호 고객정보 *****
고객 성명: 장재신
고객 주소: 경남 김해시 인제로 197 인제대학교
고객 연락처: 010-9658-8138
=====
1.입실, 2.퇴실, 3.전체조회, 4.객실조회, 5.종료:

```

24

실습 시간 (2019년 11월 5일)

- ▶ **예제 (22장):** InitStructArray.c, InitStructVal.c, PointRelation.c, StructArray.c, StructPointer.c, StructPointMem.c, StructValAddress.c, TelephoneInfo.c, TwoPointDistance.c (9개)
- ▶ **Lab 문제:** Lab 1, Lab 2 (숙박관리 프로그램 Ver. 4)



Chapter 22가 끝났습니다. 질문 있으신지요?