



[6주차 2강] 구조체(2)

학습 내용

📍 11.2 구조체의 정의, 선언, 사용 (2)

📍 11.3 구조체 배열





학습 목표

- 📍 11.2 구조체의 정의, 선언, 사용 방법을 익힌다.
- 📍 11.3 구조체와 배열의 관계를 이해한다.





1. 구조체 개요
2. 구조체의 정의, 선언, 사용 (2)
3. 구조체 배열
4. 구조체 포인터
5. 구조체와 함수
6. 중첩 구조체 및 자기참조 구조체
7. typedef 사용자 형정의

2. 구조체의 정의, 선언, 사용



구조체에 사용 가능한 연산자

- 구조체는 사용자가 만든 자료형이기 때문에 기본 자료형인 int, char, double 등에 비해 사용 가능한 연산자가 제한적임
 - ✓ 예) 산술 연산, 비교 연산 등은 지원 안 됨
 - ✓ $st1 + st2$: 구조체끼리 더하라? 의미적으로 불명확
 - ✓ $st1 < st2$: 구조체끼리 비교하라? 의미적으로 불명확
- 구조체 변수에 사용 가능한 연산자
 - ✓ 대입연산자(=), 주소연산자(&), 간접참조 연산자(*), sizeof 연산자 정도...



2. 구조체의 정의, 선언, 사용



구조체 변수의 대입 연산

- 모든 멤버 변수에 대해, 대입 연산이 수행됨

```
struct student st1 = { 10, "Tom", 3.2};  
struct student st2;  
  
st2 = st1;  
  
printf("id: %d\n", st2.id);  
printf("name: %s\n", st2.name);  
printf("grade: %.2f\n", st2.grade);
```

실행 결과

```
id: 10  
name: Tom  
grade: 3.20
```

st1	10	"Tom"	3.2
st2	10	"Tom"	3.2

멤버끼리
단순 대입



배열의 경우에도 name[0]부터 name[7]까지,
각 원소마다 단순 대입 (strcpy가 호출되는 것이 아님)



※ 실습하기



[예제 11.2]

예제 11.1의 런치 박스 구조체 사용하여 다음 프로그램을 작성하시오.

- ① 두 개의 런치 박스 A와 B선언
- ② 런치 박스 A의 각 항목의 가격을 사용자로부터 입력 받기
- ③ 런치 박스 A의 정보를 B에 복사
- ④ B의 주메뉴 가격을 사용자로부터 입력 받아 바꾸기
- ⑤ A와 B의 가격 정보를 화면에 출력



2. 구조체의 정의, 선언, 사용

구조체에 할당되는 메모리 크기

- 구조체 크기는 구조체 멤버 크기의 합? 아닐 수도 있다.
 - ✓ 4 (int) + 8 (char 배열) + 8 (double) = 20 bytes 일까?

```
struct student{
    int id;   char name[8];   double grade;
};

void main( )
{   printf("%d ", sizeof( int ) );
    printf("%d ", sizeof( char[8] ) );
    printf("%d ", sizeof( double ) );
    printf("%d ", sizeof( struct student ) );
}
결과:
4 8 8 24
```

→ 구조체의 크기가 필요한 경우
sizeof 연산자를 사용하자.
(다른 자료형도 마찬가지)



1. 구조체 개요
2. 구조체의 정의, 선언, 사용 (2)
- 3. 구조체 배열**
4. 구조체 포인터
5. 구조체와 함수
6. 중첩 구조체 및 자기참조 구조체
7. typedef 사용자 형정의

3. 구조체 배열

구조체가 배열, 포인터, 함수 등과 결합되어 확장될 수 있음

- 구조체 배열(3절), 구조체 포인터(4절), 구조체와 함수(5절)
- 대부분 기존에 학습한 문법을 그대로 적용하여 확장

구조체 배열

- 구조체가 원소로 사용된 배열
 - ✓ 단, 같은 구조체끼리의 묶음만 허용
- 선언, 접근, 초기화 등 구조체 배열에 대한 문법
 - ✓ 일반 배열과 동일



3. 구조체 배열



구조체 배열 선언과 접근: [] 사용

```
struct student ast[3]; // 구조체 배열 선언
```

```
ast[0].id = 10;
```

```
strcpy(ast[0].name, "Tom");
```

```
ast[0].grade = 3.2;
```

```
ast[1] = ast[0]; // 구조체 대입
```

```
ast[1].name[0] = 'M';
```

ast[0]			ast[1]			ast[2]		
id	name[8]	grade	id	name[8]	grade	id	name[8]	grade
10	Tom	3.20	10	Mom	3.20			



3. 구조체 배열



구조체 배열 초기화: 중괄호 { } 이용

```
int i;  
struct student ast[3]  
    = { { 10, "Tom", 3.2},  
        { 20, "Alice"} };  
// 생략된 부분은 모두 0으로 초기화  
  
for( i= 0 ; i < 3 ; ++i )    // 반복문을 이용한 배열 출력  
    printf("%d: %d, %s, %.2f\n",  
           i, ast[i].id, ast[i].name, ast[i].grade);
```

실행 결과

```
0: 10, Tom, 3.20  
1: 20, Alice, 0.00  
2: 0, ,0.00
```



※ 실습하기



[예제 11.3] 복소수 구조체 배열

- 복소수를 구조체로 표현하고, 크기가 3인 복소수 구조체 배열을 선언한 후 (0번, 1번 복소수는 아래와 같이 초기화), 0번과 1번 복소수의 덧셈 결과를 2번 복소수에 대입하는 프로그램을 작성하시오.

- ✓ 0번 복소수: $1.2 + 2.0i$
- ✓ 1번 복소수: $-2.2 - 0.3i$

```
struct complex {    // 구조체 자료형 선언
    double real, imag;
};
```

실행 결과

```
0: 1.2 + 2.0i
1: -2.2 + -0.3i
2: -1.0 + 1.7i
```



※ 실습하기



[예제 11.4] 예제 11.1에서 정의한 구조체'의 '배열'을 이용하여 2개의 런치 박스의 정보를 사용자로부터 입력 받고 출력하는 프로그램을 작성하시오.





학습 정리

- 구조체 변수의 **대입 연산**은 각 멤버 변수에 대한 대입 연산을 의미
- 구조체의 크기가 필요한 경우 sizeof 연산자를 사용
- **구조체 배열**은 동일한 구조체 자료형의 변수를 묶은 배열
- 선언, 접근, 초기화 등 구조체 배열에 대한 문법은 일반 배열과 동일

