11강

구조체와 공용체

동양미래대학교 강환수 교수

한국방송통신대학교 프라임칼리지

본 강의 사용 및 참조 자료

- Perfect C, 3판, 강환수 외 2인 공저, 인피니티북스, 2021

13장 구조체와 공용체

목차

01

구조체

구조체 개념 1/3

선물셋트

- 인기가 있거나 관련 있는 상품들을 묶어
  하나의 구성제품으로 판매하는 것

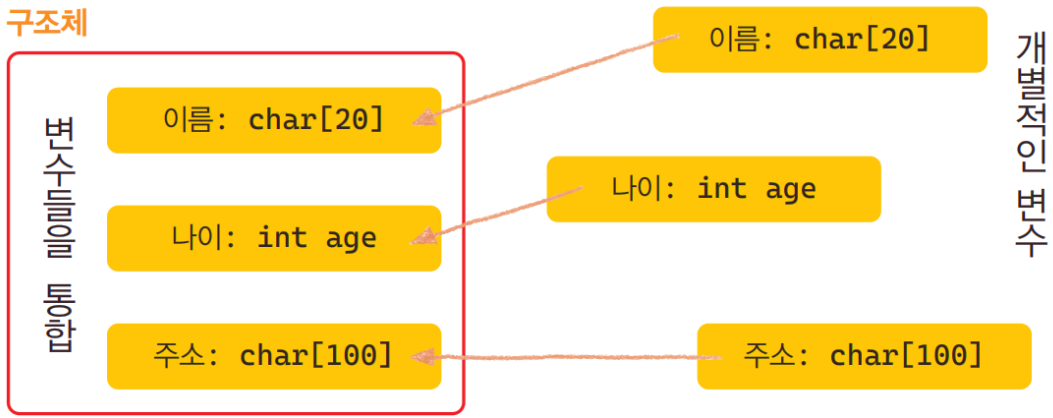
구조체

- 정수, 문자, 실수나 포인터 그리고 이들의 배열 등을 묶어
  하나의 자료형으로 이용하는 것
- 서로 관련 있는 정보들을 하나로 묶어 처리하는 경우가 흔히 발생
  - 차에 대한 정보, 계좌에 대한 정보, 책에 대한 정보,
    학생, 교수, 강좌에 관한 정보

구조체 개념 2/3

유도 자료형(derived data types)

- 연관된 멤버로 구성되는 통합 자료형으로 대표적인 유도 자료형
- 기존 자료형에서 새로이 만들어진 자료형



[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 552쪽

구조체 개념 3/3

학생정보

교수정보

여러 자료형의 통합체인 학생, 교수, 강좌 등을 새로운 하나의 자료형인 구조체로 정의

강좌정보

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 553쪽

구조체 정의 개념

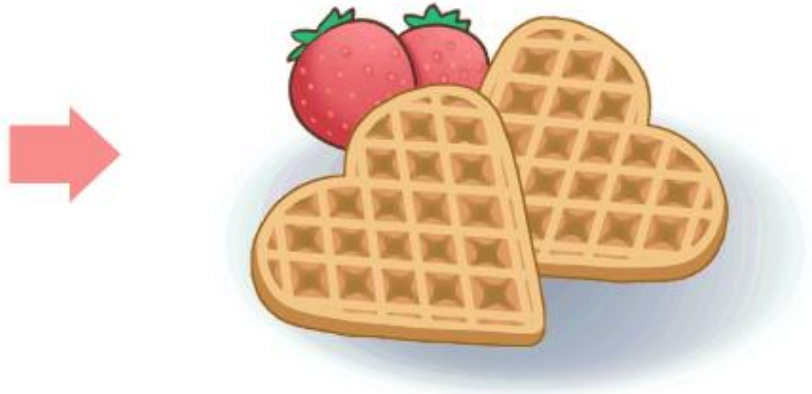
- 와플이나 붕어빵을 만들려면
  - 와플 기계나 붕어빵 기계가 필요하듯이
- 구조체를 자료형으로 사용하려면
  - 먼저 구조체를 정의
    - 구조체를 만들 구조체 틀(template)을 정의

구조체 정의 방법 1/2

구조체 틀을 만드는 구조체 정의 방법

- 키워드 struct 다음에 구조체 태그이름을 기술
  - 중괄호를 이용하여
    원하는 멤버를 여러 개의 변수로 선언하는 구조
- 구조체 멤버(member) 또는 필드(field)
  - 구조체를 구성하는 하나 하나의 항목

구조체 정의 방법 2/2



구조체 틀: 정의

```
struct lecture
{
    char name[20]; //강좌명
    int credit;    //학점
    int hour;      //시수
};
```

구조체 정의 없이는 자료형
struct lecture를 사용할 수 없다.

구조체를 자료형으로 사용

```
struct lecture datastructure;
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 553쪽

대학의 강좌정보를 처리하는 구조체

- struct lecture
  - 구조체 정의는 변수의 선언과는 다름
    - 변수선언에서 이용될 새로운 구조체 자료형을 정의하는 구문
  - 반드시 세미콜론으로 종료
    - 모든 멤버 선언에 반드시 세미콜론 삽입, 마지막 멤버에도 삽입
  - 각 구조체 멤버의 초기값 대입 불가능
    - int credit; int hour;
    - Int credit, hour;로도 가능

구조체 멤버의 이름은 모두 유일

- 멤버로는 다양한 자료형,
  다른 구조체 변수 및 구조체 포인터도 허용

문자열 입출력 함수: stdio.h

구조체 구성요소(struct member)라 한다.
초기값을 설정할 수 없다.

```
struct 구조체태그이름
{
    자료형 변수명1;
    자료형 변수명2;
    ...
};
```

세미콜론은 반드시 필요하다.

```
struct lecture
{
    char name[20];    //강좌명
    int credit;       //학점
    int hour;         //시수
};
```

마지막 멤버 hour에도 반드시 ;이 필요하다.

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 554쪽

구조체 태그이름: account

- struct account
  - 계좌정보를 표현하는 구조체

- 계좌주이름, 계좌번호, 잔고 정보를 하나의 단위로 처리하는 자료형을 정의

```
struct account
{
    char name[10];      //계좌주이름
    int actnum;         //계좌번호
    double balance;     //잔고
};
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 554쪽

구조체 변수 선언 1/2

구조체가 정의되었다면

- 구조체형 변수 선언이 가능
  - 구조체 struct account가 새로운 자료유형으로 사용 가능
- 새로운 자료형 struct account 형 변수 mine을 선언 구문
  - struct account mine;
- 구조체 자료형 변수 선언 및 초기화 구문

```
struct   구조체태그이름   변수명;
struct   구조체태그이름   변수명1, 변수명2, 변수명3, ... ;
```

여러 변수의 선언도 가능하다.

```
struct account yours;
struct account act1, act2, act3;
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 555쪽

구조체 변수 선언 2/2

구조체 정의와 변수 선언을 함께하는 방법

- 이 문장 이후 struct account도 새로운 자료형으로 사용 가능

```
struct account
{
    char name[12];        //계좌주이름
    int actnum;           //계좌번호
    double balance;       //잔고
} myaccount;

struct account youraccount;
```

변수 myaccount는 struct account형 변수로 선언된다.

변수 youraccount도 struct account형 변수로 선언된다.

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 556쪽

구조체 변수의 초기화 1/2

- 변수 선언 시 중괄호를 이용한 초기화 지정이 가능
  - 초기화 값은 중괄호 내부에서
    각 멤버 정의 순서대로 초기값을 쉼표로 구분하여 기술
  - 기술되지 않은 멤버값은 자료형에 따라
    기본값인 0, 0.0, '₩0' 등으로 저장

구조체 변수의 초기화 2/2

```
struct 구조체태그이름    변수명 = {초기값1, 초기값2, 초기값3, ... };
```

```
struct account
{
    char name[12];          //계좌주이름
    int actnum;             //계좌번호
    double balance;         //잔고
};

struct account mine = {"홍길동", 1001, 300000 };
struct account mine = {"한국인", 1001};
```

멤버 balance에는 double형 기본값인 0.00이 저장

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 557쪽

구조체의 멤버 접근 연산자 .

선언된 구조체형 변수에서 멤버 접근 방법

- 접근연산자 .를 사용하여 멤버를 참조
  - yours.actnum=1002;
    - 변수 yours의 멤버 actnum에 1002를 저장하는 기능을 수행
    - 접근연산자는 .는 참조연산자라고도 부름

```
구조체변수이름.멤버
mine.actnum = 1002;    mine.balance = 300000;
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 557쪽

실습예제 1/2

Prj02 02nestedstruct.c 구조체 멤버로 다른 구조체 형 포함 난이도: ★

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

//날짜를 위한 구조체
struct date
{
    int year;   //년
    int month;  //월
    int day;    //일
};

//은행계좌를 위한 구조체
struct account
{
    struct date open;     //계좌 개설일자
    char name[12];        //계좌주 이름
    int actnum;           //계좌번호
    double balance;       //잔고
};
```

구조체 멤버로 다른 구조체 변수를 포함

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 561쪽

실습예제 2/2

```
21  int main(void)
22  {
23      struct account me = { { 2022, 3, 9 }, "홍길동", 1001, 300000 };
24
25      printf("구조체 크기: %zu\n", sizeof(me));
26      printf("[%d. %d. %d]\n", me.open.year, me.open.month, me.open.day);
27      printf("%s %d %.2f\n", me.name, me.actnum, me.balance);
28  }
```

변수 open을 위한 {}는 생략 가능

중첩된 구조체를 접근하려면 접근연산자를 2번 사용

```
구조체 크기: 40
[2022. 3. 9]
홍길동 1001 300000.00
```

산술적인 크기인 36보다 큼

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 561쪽

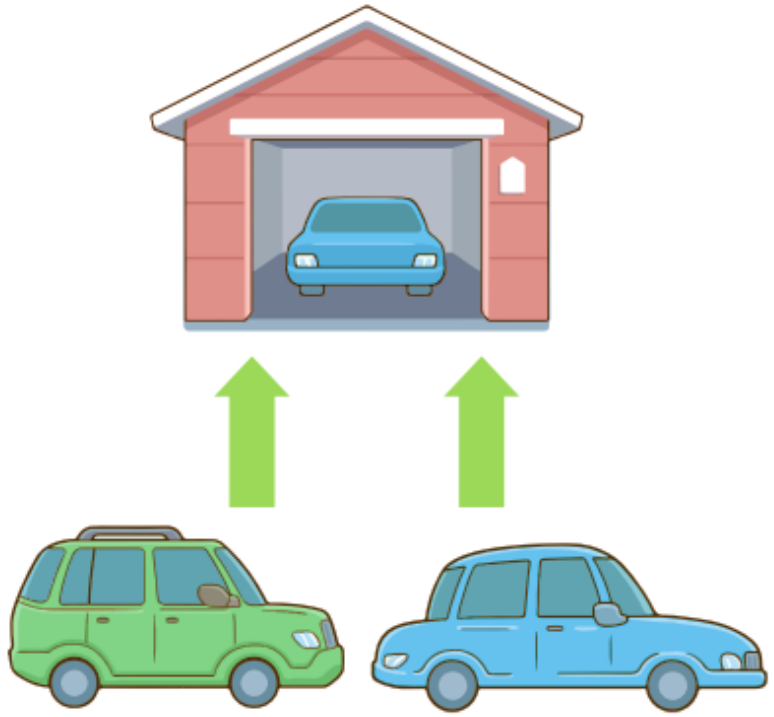
02

공용체

공용체 개념 1/2

- 하나의 차고에 일반 세단과 SUV를 각각 주차한다고 생각
  - 공간이 하나이므로 어느 시간에 한 대의 주차는 가능
- 공용체
  - 동일한 저장 장소에 여러 자료형을 저장하는 방법
    - 이러한 겸용 주차장과 비슷한 개념
  - 멤버에 한번에 한 종류만 저장하고 참조 가능

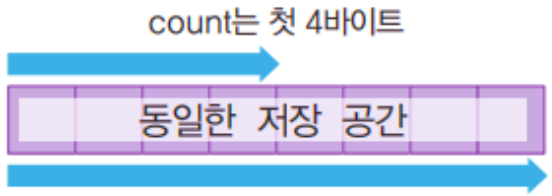
공용체 개념 2/2



공용체

```
union share
{
    int count;
    double value;
};

union share a;
```




```
a.count = 55;
```

```
a.value = 243.5;
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 566쪽

union을 사용한 공용체 정의 및 변수 선언

- 공용체(union)
  - 서로 다른 자료형의 값을 동일한 저장공간에 저장하는 자료형
- 공용체 선언 방법
  - union을 struct로 사용하는 것을 제외하면
    구조체선언 방법과 동일

공용체 정의 및 변수 선언 구문

공용체 정의 및 변수 선언 구문

```
union 공용체태그이름
{
    자료형 멤버변수명1;
    자료형 멤버변수명2;
    ...
} [변수명1] [,변수명2];
```

공용체 구성요소인 멤버(struct member)이다.

세미콜론은 반드시 필요하다.

```
union data
{
    char ch;        //문자형
    int cnt;        //정수형
    double real;    //실수형
} data1;
```

```
union udata
{
    char name[4];   //char형 배열
    int n;          //정수형
    double val;     //실수형
};
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 567쪽

공용체 변수의 크기

- union data의 변수 data1은
  멤버 중 가장 큰 자료형의 크기로 정해짐
- 동시에 여러 멤버의 값을 동시에 저장하여 이용 불가능
  - 마지막에 저장된 단 하나의 멤버 자료값만을 저장
- 구조체와 같이 typedef를 이용하여
  새로운 자료형으로 정의 가능

공용체 초기화

- 처음 선언한 멤버의 초기 값으로만 저장이 가능
- 만일 다른 멤버로 초기값을 지정하면
  - 컴파일 시 경고가 발생
  - 초기값으로 동일한 유형의 다른 변수의 대입도 가능

```
typedef union data uniondata;

    uniondata data2 = {'A'};        //첫 멤버인 char형으로만 초기화 가능
    //uniondata data2 = {10.3};     //컴파일 시 경고 발생
```

warning C4244: '초기화중' : 'double'에서 'char'(으)로 변환하면서 데이터가 손실될 수 있습니다.

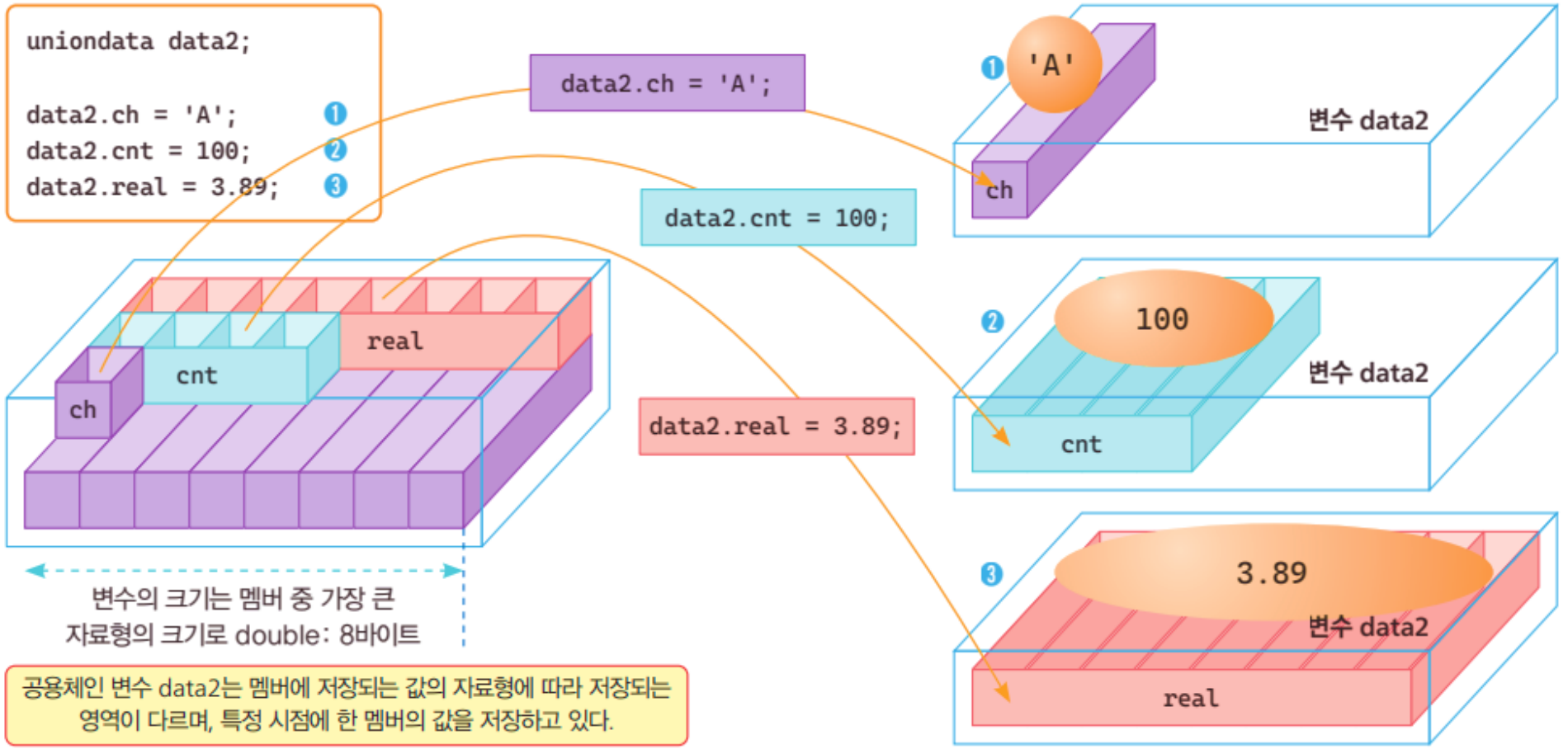
```
    uniondata data3 = data2;        //다른 변수로 초기화 가능
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 567쪽

공용체 멤버 접근

- 구조체와 같이 접근연산자 .를 사용
  - data2.ch = 'A';
    - 위 문장 이후에 멤버 cnt나 real의 출력은 가능하나 의미는 없음
  - 유형이 char인 ch를 접근하면 8바이트 중에서 첫 1바이트만 참조
    - int인 cnt를 접근하면 전체 공간의 첫 4바이트만 참조
    - double인 real을 접근하면 8바이트 공간을 모두 참조
  - 항상 마지막에 저장한 멤버로 접근해야 원하는 값을 얻을 수 있음
    - 공용체에서 정확한 멤버를 사용하는 것은 프로그래머의 책임

공용체 변수의 저장공간과 참조



[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 568쪽

03

자료형 재정의

typedef 구문

- 이미 사용되는 자료 유형을
  다른 새로운 자료형 이름으로 재정의
- typedef int profit;
  - profit을 int와 같은 자료형으로 새롭게 정의하는 문장
  - 자료형 재정의 typedef 구문

```
typedef   기존자료유형이름   새로운자료형1, 새로운자료형2, ... ;
```

```
typedef int profit;
typedef unsigned int budget;
typedef unsigned int size_t;
typedef unsigned __int64 size_t;
```

여러 이름으로도 재정의할 수 있다.

새로운 이름 size_t도 자료형 unsigned int와 동일하게 이용 가능하다.

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 573쪽

프로그램의 시스템 간 호환성을 위해 필요

- 터보 C++ 컴파일러에서 자료유형 int는 저장공간 크기가 2바이트
  - 비주얼 스튜디오에서는 4바이트
- 비주얼 스튜디오에서 작성한 프로그램은 터보 C++에서는 문제가 발생
  - 2 바이트로는 2,000,000을 저장할 수 없기 때문

Visual C++: 4바이트

```
int salary = 2000000;
```

Turbo C++: 2바이트

```
int salary = 2000000;
```

오버플로 발생(데이터 손실)

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 573쪽

개발환경마다 다른 자료형의 크기 대처 방안

- Visual C++에서는 다음과 같이 int를 myint로 재정의
  - 모든 int 형을 myint형으로 선언하여 이용
- 만일 이 소스를 터보 C++에서 컴파일 한다면
  - typedef 문장에서 int를 long으로 수정
  - 아무 문제 없이 다른 소스는 수정 없이 그대로 이용 가능

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 574쪽

여러 이름의 자료형

- 자료형 int를 여러 개의
  새로운 자료형 이름 integer와 word로 재정의

```
typedef int integer, word;

integer myAge;   //int myAge와 동일
word yourAge;    //int yourAge와 동일
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 574쪽

실습예제 1/2

Prj05 05typedef.c 자료형 재정의 키워드 typedef 이용 난이도: ★

```
01  #include <stdio.h>
02
03  //함수 외부에서 정의된 자료형은 이후 파일에서 사용 가능
04  typedef unsigned int budget;
05
06  int main(void)
07  {
08      budget year = 24500000; //새로운 자료형 budget 사용
09
10      //함수 내부에서 정의된 자료형은 함수 내부에서만 사용 가능
11      typedef int profit;
12      profit month = 4600000; //새로운 자료형 profit 사용
13      printf("올 예산은 %d, 이달의 이익은 %d 입니다.\n", year, month);
14
15      return 0;
```

budget은 int와 같은 자료형으로 변수 year를 budget으로 선언하면서 초기값 대입

실습예제 2/2

```
16  }
17
18  void test(void)
19  {
20      budget year = 24500000; //새로운 자료형 budget 사용
21
22      //profit은 이 함수에서는 사용 불가, 컴파일 오류 발생
23      //profit year;
24  }
```

```
올 예산은 24500000, 이달의 이익은 4600000 입니다.
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 575쪽

struct를 생략한 새로운 자료형 정의 1/2

- 구조체 자료형은 struct date 처럼 항상 키워드 struct를 써야 하나?
  - typedef 사용하여 구조체 struct date를 date로 재정의
    - 물론 date가 아닌 datetype 등 다른 이름으로도 재정의가 가능

```
struct date
{
    int year;     //년
    int month;    //월
    int day;      //일
};

typedef struct date date;
```

자료형인 date는 struct date와 함께 동일한 자료유형으로 이용이 가능하다.

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 575쪽

struct를 생략한 새로운 자료형 정의 2/2

- typedef 구문에서 새로운 자료형으로 software 형 정의
  - 이 구문 이후에는 software를 구조체 자료형으로 변수 선언에 사용
  - 구조체 태그이름은 생략 가능
- 구조체 software 형
  - 멤버로 구조체 date형 변수 release

```
typedef struct
{
    char title[30];      //제목
    char company[30];    //제작회사
    char kinds[30];      //종류
    date release;        //출시일
} software;
```

software는 변수가 아니라 새로운 자료형이다.

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 576쪽

실습예제

lab2structmovie.c 난이도: ★

```
01  #include <stdio.h>
02
03  int main(void)
04  {
05      //영화 정보 구조체
06      typedef struct movie
07      {
08          char* title;        //영화제목
09          long long profit;   //흥행수익
10      } movie;
11
12      movie parasite;
13      parasite.title = "기생충";
14      parasite.profit = 310000000000;//전 세계에서 3,100억 수익
15
16      printf("[%s] 총수익: %lld\n", parasite.title, parasite.profit);
17
18      return 0;
19  }
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 578쪽

정 리 하 기

정리하기

- 구조체는 정수, 문자, 실수나 포인터 그리고
  이들의 배열 등을 묶어 하나의 자료형으로 이용하는 것이다.
- 공용체는 동일한 저장 장소에 공용으로 여러 자료형을 저장한다.
- 구조체와 공용체에서 접근연산자 .를 사용하여 멤버를 참조한다.
- 키워드 typedef로 이미 사용되는 자료 유형을
  다른 새로운 자료형 이름으로 재정의한다.
- 키워드 typedef를 사용해
  struct가 생략된 간단한 자료형을 다시 정의해 사용한다.

12강

함수와 포인터 활용

다음시간안내