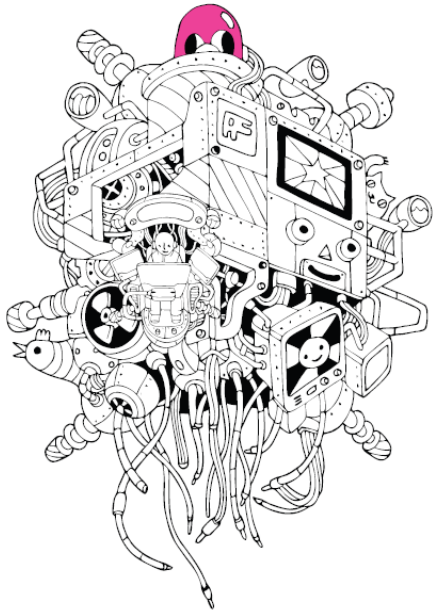


C 프로그래밍 개정판

Chapter 23. 구조체와 사용자 정의 자료형2

Chapter 23-1. 구조체의 정의와 typedef 선언

실습 I

```
typedef int INT;
typedef int * PTR_INT;

typedef unsigned int UINT;
typedef unsigned int * PTR_UINT;

typedef unsigned char UCHAR;
typedef unsigned char * PTR_UCHAR;

int main(void)
{
    INT num1 = 120;             // int num1 = 120;
    PTR_INT pnum1 = &num1;      // int * pnum1 = &num1;

    UINT num2 = 190;            // unsigned int num2 = 190;
    PTR_UINT pnum2 = &num2;     // unsigned int * pnum2 = &num2;

    UCHAR ch = 'Z';             // unsigned char ch = 'Z';
    PTR_UCHAR pch = &ch;        // unsigned char * pch = &ch;

    printf("%d, %u, %c \n", *pnum1, *pnum2, *pch);
    return 0;
}
```

typedef 선언

```
typedef  int  INT;
```

자료형의 이름 int에 INT라는 이름을 추가로 붙여줍니다.

위의 typedef 선언으로 인해서!!!

```
INT num;
INT * ptr;
```

int num; 과 동일한 선언

int * ptr; 과 동일한 선언

새로 부여된 이름	대상 자료형
INT	int
PTR_INT	int *
UINT	unsigned int
PTR_UINT	unsigned int *
UCHAR	unsigned char
PTR_UCHAR	unsigned char *

정의되는 이름들

```c
typedef int INT;
typedef int * PTR_INT;

typedef unsigned int UINT;
typedef unsigned int * PTR_UINT;

typedef unsigned char UCHAR;
typedef unsigned char * PTR_UCHAR;

int main(void)
{
    INT num1 = 120;             // int num1 = 120;
    PTR_INT pnum1 = &num1;      // int * pnum1 = &num1;

    UINT num2 = 190;            // unsigned int num2 = 190;
    PTR_UINT pnum2 = &num2;     // unsigned int * pnum2 = &num2;

    UCHAR ch = 'Z';             // unsigned char ch = 'Z';
    PTR_UCHAR pch = &ch;        // unsigned char * pch = &ch;

    printf("%d, %u, %c \n", *pnum1, *pnum2, *pch);
    return 0;
}
```

실행결과

```
120, 190, Z
```

구조체 정의와 typedef 선언

```
struct point
{
    int xpos;
    int ypos;
};

typedef struct point Point;
```

구조체 point 정의 후

struct point에 Point라는 이름을 부여하기 위한 typedef 선언 추가!

합친 형태

```
typedef struct point
{
    int xpos;
    int ypos;
} Point;
```

구조체 point의 정의와

Point에 대한 typedef 선언을 한데 묶은 형태

실습 2

```
struct point
{
    int xpos;
    int ypos;
};

typedef struct point Point;

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

int main(void)
{
    Point pos={10, 20};
    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;
}
```

* 아래 예제를 변형하여,
건물이름의 구조체를 만들고 건물의
이름, 번지수, 층수, 가격을 넣으시오

구조체 정의와 typedef 선언 관련 예제

```
struct point
{
    int xpos;
    int ypos;
};

typedef struct point Point;
```

구조체 point의 정의와 typedef 선언

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

구조체 person의 정의와

Person이라는 이름의 typedef 선언을 하나로!

```
int main(void)
{
    Point pos={10, 20};
    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;
}
```

실행결과

```
10 20
이승기 010-1212-0001 21
```

구조체의 이름 생략

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

typedef 선언으로 인해서 새로운 이름 Person이 정의되었으니, 구조체의 이름 persons은 큰 의미가 없다.

이름이 생략된 형태

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

따라서 이렇듯 구조체의 이름을 생략하는 것도 가능하다.

Chapter 23-2. 함수로의 구조체 변수 전달과 반환

C 프로그래밍 개정판

실습 3

```
typedef struct point
{
    int xpos;
    int ypos;
} Point;

void ShowPosition(Point pos)
{
    printf("[%d, %d] \n", pos.xpos, pos.ypos);
}

Point GetCurrentPosition(void)
{
    Point cen;
    printf("Input current pos: ");
    scanf("%d %d", &cen.xpos, &cen.ypos);
    return cen;
}

int main(void)
{
    Point curPos=GetCurrentPosition();
    ShowPosition(curPos);
    return 0;
}
```

* 아래 예제를 변형하여,
(I) 학생의 성적 정보를 추가하시오

함수의 인자로 전달되고 return문에 의해 반환되는 구조체 변수1

```
typedef struct point
{
    int xpos;
    int ypos;
} Point;

void ShowPosition(Point pos)
{
    printf("[%d, %d] \n", pos.xpos, pos.ypos);
}

Point GetCurrentPosition(void)
{
    Point cen;
    printf("Input current pos: ");
    scanf("%d %d", &cen.xpos, &cen.ypos);
    return cen;
}

int main(void)
{
    Point curPos=GetCurrentPosition();
    ShowPosition(curPos);
    return 0;
}
```

구조체 변수 cen이 통째로 반환된다.

ShowPosition 함수의 매개변수에 curPos에 저장된 값이 통째로 복사된다.

실행결과

```
Input current pos: 2 4
[2, 4]
```

실습 4

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

* 아래 예제를 변형하여,
(I) 건물이름의 구조체를 만들고 건물의 이름, 번지수, 층수, 가격을 넣으시오

```
void ShowPersonInfo(Person man)
{
    printf("name: %s \n", man.name);
    printf("phone: %s \n", man.phoneNum);
    printf("age: %d \n", man.age);
}

Person ReadPersonInfo(void)
{
    Person man;
    printf("name? "); scanf("%s", man.name);
    printf("phone? "); scanf("%s", man.phoneNum);
    printf("age? "); scanf("%d", &man.age);
    return man;
}

int main(void)
{
    Person man=ReadPersonInfo();
    ShowPersonInfo(man);
    return 0;
}
```

배열까지도 통째로 복사

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

구조체의 멤버로 배열이 선언된 경우

구조체 변수를 인자로 전달하거나 반환 시

배열까지도 통째로 복사가 이뤄진다.

실행결과

```
name? Jung
phone? 010-12XX-34XX
age? 22
name: Jung
phone: 010-12XX-34XX
age: 22
```

```
void ShowPersonInfo(Person man)
{
    printf("name: %s \n", man.name);
    printf("phone: %s \n", man.phoneNum);
    printf("age: %d \n", man.age);
}

Person ReadPersonInfo(void)
{
    Person man;
    printf("name? "); scanf("%s", man.name);
    printf("phone? "); scanf("%s", man.phoneNum);
    printf("age? "); scanf("%d", &man.age);
    return man;
}

int main(void)
{
    Person man=ReadPersonInfo();
    ShowPersonInfo(man);
    return 0;
}
```

실습 5

```
typedef struct point
{
    int xpos;
    int ypos;
} Point;

void OrgSymTrans(Point * ptr)    // 원점대칭
{
    ptr->xpos = (ptr->xpos) * -1;
    ptr->ypos = (ptr->ypos) * -1;
}

void ShowPosition(Point pos)
{
    printf("[%d, %d] \n", pos.xpos, pos.ypos);
}

int main(void)
{
    Point pos={7, -5};
    OrgSymTrans(&pos);    // pos의 값을 원점 대칭이동시킨다.
    ShowPosition(pos);
    OrgSymTrans(&pos);    // pos의 값을 원점 대칭이동시킨다.
    ShowPosition(pos);
    return 0;
}
```

* 아래 예제를 변형하여,

(1) X축 및 Y축 대칭 좌표 계산 기능을 추가하시오

(2) 학생의 국어, 영어, 수학 성적 정보를 받아서 평균을 내는 함수를 작성하시오. 단, 반드시 포인터 변수로 구현해야 함.

구조체 기반의 Call-by-reference

```
typedef struct point
{
    int xpos;
    int ypos;
} Point;

void OrgSymTrans(Point * ptr)    // 원점대칭
{
    ptr->xpos = (ptr->xpos) * -1;
    ptr->ypos = (ptr->ypos) * -1;
}

void ShowPosition(Point pos)
{
    printf("[%d, %d] \n", pos.xpos, pos.ypos);
}

int main(void)
{
    Point pos={7, -5};
    OrgSymTrans(&pos);    // pos의 값을 원점 대칭이동시킨다.
    ShowPosition(pos);
    OrgSymTrans(&pos);    // pos의 값을 원점 대칭이동시킨다.
    ShowPosition(pos);
    return 0;
}
```

구조체 변수 대상의 Call-by-reference는 일반변수의 Call-by-reference와 동일하다.

실행결과

```
[-7, 5]
[7, -5]
```

실습 6

```
typedef struct point
{
    int xpos;
    int ypos;
} Point;

int main(void)
{
    Point pos1={1, 2};
    Point pos2;
    pos2=pos1;  // pos1의 멤버 대 pos2의 멤버간 복사가 진행됨

    printf("크기: %d \n", sizeof(pos1));   // pos1의 전체 크기 반환
    printf("[%d, %d] \n", pos1.xpos, pos1.ypos);
    printf("크기: %d \n", sizeof(pos2));   // pos2의 전체 크기 반환
    printf("[%d, %d] \n", pos2.xpos, pos2.ypos);
    return 0;
}
```

* 아래 예제를 변형하여,
(I) 학생의 정보를 추가하시오

구조체 변수를 대상으로 가능한 연산1

```
typedef struct point
{
    int xpos;
    int ypos;
} Point;

int main(void)
{
    Point pos1={1, 2};
    Point pos2;
    pos2=pos1;  // pos1의 멤버 대 pos2의 멤버간 복사가 진행됨

    printf("크기: %d \n", sizeof(pos1));   // pos1의 전체 크기 반환
    printf("[%d, %d] \n", pos1.xpos, pos1.ypos);
    printf("크기: %d \n", sizeof(pos2));   // pos2의 전체 크기 반환
    printf("[%d, %d] \n", pos2.xpos, pos2.ypos);
    return 0;
}
```

구조체 변수간 대입연산의 결과로 멤버 대 멤버 복사가 이뤄진다는 사실을 확인하자!

실행결과

```
크기: 8
[1, 2]
크기: 8
[1, 2]
```

실습 7

```
typedef struct point
{
    int xpos;
    int ypos;
} Point;
```

```
Point AddPoint(Point pos1, Point pos2)
{
    Point pos={pos1.xpos+pos2.xpos, pos1.ypos+pos2.ypos};
    return pos;
}
Point MinPoint(Point pos1, Point pos2)
{
    Point pos={pos1.xpos-pos2.xpos, pos1.ypos-pos2.ypos};
    return pos;
}
int main(void)
{
    Point pos1={5, 6};
    Point pos2={2, 9};
    Point result;

    result=AddPoint(pos1, pos2);
    printf("[%d, %d] \n", result.xpos, result.ypos);
    result=MinPoint(pos1, pos2);
    printf("[%d, %d] \n", result.xpos, result.ypos);
    return 0;
}
```

* 아래 예제를 변형하여,
(I) 두 점의 거리구하는 계산을 추가하시오

구조체 변수를 대상으로 가능한 연산2

```
typedef struct point
{
    int xpos;
    int ypos;
} Point;
```

구조체 변수를 대상으로는 덧셈 및 뺄셈 연산이 불가능하다.

따라서 필요하다면 덧셈함수와 뺄셈함수를 정의해야 한다.

```
Point AddPoint(Point pos1, Point pos2)
{
    Point pos={pos1.xpos+pos2.xpos, pos1.ypos+pos2.ypos};
    return pos;
}
```

구조체 Point의 덧셈 함수

```
Point MinPoint(Point pos1, Point pos2)
{
    Point pos={pos1.xpos-pos2.xpos, pos1.ypos-pos2.ypos};
    return pos;
}
```

구조체 Point의 뺄셈 함수

```
int main(void)
{
    Point pos1={5, 6};
    Point pos2={2, 9};
    Point result;

    result=AddPoint(pos1, pos2);
    printf("[%d, %d] \n", result.xpos, result.ypos);
    result=MinPoint(pos1, pos2);
    printf("[%d, %d] \n", result.xpos, result.ypos);
    return 0;
}
```

실행결과

```
[7, 15]
[3, -3]
```

Chapter 23-3. 구조체의 유용함에 대한 논의와 중첩 구조체

C 프로그래밍 개정판

실습 8

* 아래 예제를 변형하여,
(I) 건물 정보를 추가하시오

```
typedef struct student
{
    char name[20];       // 학생 이름
    char stdnum[20];     // 학생 고유번호
    char school[20];     // 학교 이름
    char major[20];      // 선택 전공
    int year;            // 학년
} Student;

void ShowStudentInfo(Student * sptr)
{
    printf("학생 이름: %s \n", sptr->name);
    printf("학생 고유번호: %s \n", sptr->stdnum);
    printf("학교 이름: %s \n", sptr->school);
    printf("선택 전공: %s \n", sptr->major);
    printf("학년: %d \n", sptr->year);
}
```

```
int main(void)
{
    Student arr[7];
    int i;

    for(i=0; i<7; i++)
    {
        printf("이름: "); scanf("%s", arr[i].name);
        printf("번호: "); scanf("%s", arr[i].stdnum);
        printf("학교: "); scanf("%s", arr[i].school);
        printf("전공: "); scanf("%s", arr[i].major);
        printf("학년: "); scanf("%d", &arr[i].year);
    }

    for(i=0; i<7; i++)
        ShowStudentInfo(&arr[i]);
    return 0;
}
```

구조체를 정의하는 이유

- 연관 있는 데이터를 하나로 묶을 수 있는 자료형을 정의할 수 있다. 구조체의 정의 이유!
- 연관 있는 데이터를 묶으면 데이터의 표현 및 관리가 용이해진다.
- 데이터의 표현 및 관리가 용이해지면 그만큼 합리적인 코드를 작성할 수 있다.

```
typedef struct student
{
    char name[20];      // 학생 이름
    char stdnum[20];    // 학생 고유번호
    char school[20];    // 학교 이름
    char major[20];     // 선택 전공
    int year;           // 학년
} Student;
```

인자 전달 시 용이하다.

```
void ShowStudentInfo(Student * sptr)
{
    printf("학생 이름: %s \n", sptr->name);
    printf("학생 고유번호: %s \n", sptr->stdnum);
    printf("학교 이름: %s \n", sptr->school);
    printf("선택 전공: %s \n", sptr->major);
    printf("학년: %d \n", sptr->year);
}
```

```
int main(void)
{
    Student arr[7];
    int i;

    for(i=0; i<7; i++)
    {
        printf("이름: "); scanf("%s", arr[i].name);
        printf("번호: "); scanf("%s", arr[i].stdnum);
        printf("학교: "); scanf("%s", arr[i].school);
        printf("전공: "); scanf("%s", arr[i].major);
        printf("학년: "); scanf("%d", &arr[i].year);
    }

    for(i=0; i<7; i++)
        ShowStudentInfo(&arr[i]);
    return 0;
}
```

하나의 배열 선언으로 종류가 다른 데이터들을 한데 저장할 수 있다.

실습 9

```
typedef struct point
{
    int xpos;
    int ypos;
} Point;

typedef struct circle
{
    Point cen;
    double rad;
} Circle;
```

* 아래 예제를 변형하여,
(I) 학생의 정보를 추가하시오
  A. 학생의 이름
  B. 학생의 학번
  C. 학생의 국어, 영어, 수학 성적

```
void ShowCircleInfo(Circle * cptr)
{
    printf("[%d, %d] \n", (cptr->cen).xpos, (cptr->cen).ypos);
    printf("radius: %g \n\n", cptr->rad);
}

int main(void)
{
    Circle c1={{1, 2}, 3.5};
    Circle c2={2, 4, 3.9};
    ShowCircleInfo(&c1);
    ShowCircleInfo(&c2);
    return 0;
}
```

중첩된 구조체의 정의와 변수의 선언

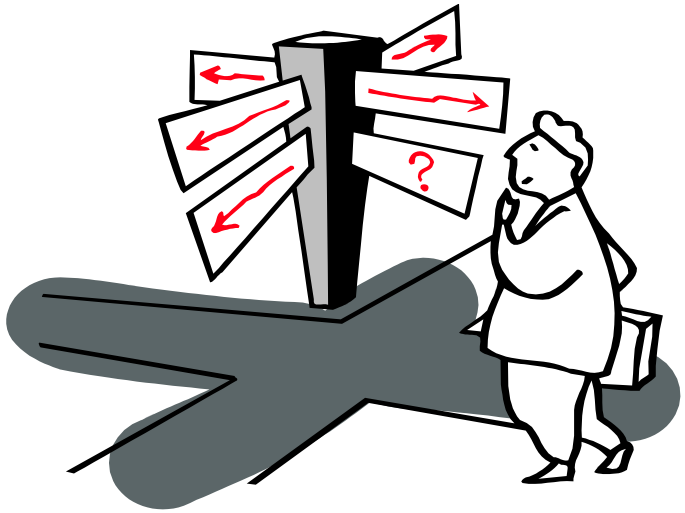
```
typedef struct point
{
    int xpos;
    int ypos;
} Point;

typedef struct circle
{
    Point cen;
    double rad;
} Circle;
```

```
void ShowCircleInfo(Circle * cptr)
{
    printf("[%d, %d] \n", (cptr->cen).xpos, (cptr->cen).ypos);
    printf("radius: %g \n\n", cptr->rad);
}

int main(void)
{
    Circle c1={{1, 2}, 3.5};
    Circle c2={2, 4, 3.9};
    ShowCircleInfo(&c1);
    ShowCircleInfo(&c2);
    return 0;
}
```

앞서 정의한 구조체는 이후에 새로운 구조체를 선언하는데 있어서 기본 자료형의 이름과 마찬가지로 사용이 될 수 있다.

실행결과

```
[1, 2]
radius: 3.5

[2, 4]
radius: 3.9
```



Chapter 23이 끝났습니다. 질문 있으신지요?