

13주차 3차시 구조체 배열, 공용체 실습

【학습목표】

1. 실습을 통하여 구조체 배열을 학습하고 실행할 수 있다.
2. 실습을 통하여 공용체를 학습하고 실행할 수 있다.

학습내용1 : 구조체 배열 실습

- * 구조체 배열의 선언과 접근 예제 실습(교재 13주차 1차시 예제 참조)
 - 도서 source: StructArray.c

```
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;
}
```

- * 구조체 변수와 포인터 예제 실습(교재 13주차 1차시 예제 참조)
 - 도서 source: StructPointer.c

```

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;
}

```

- * 문자열 형태의 '종업원 이름'과 문자열 형태의 '주민등록번호' 그리고 정수 형태의 '급여정보'를 저장할 수 있는 employee라는 이름의 구조체를 정의해 보자. 그리고 나서 employee 구조체 변수를 하나 선언한 다음, 프로그램 사용자가 입력하는 정보로 이 변수를 채우자. 그리고 마지막으로 구조체 변수에 채워진 데이터를 출력해 보자

* typedef 선언 예제 실습(교재 13주차 1차시 예제 참조)

- 도서 source: TypeNameTypedef.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;
}
```

학습내용2 : 공용체 실습

- * 구조체 변수를 대상으로 가능한 연산 예제 실습(교재 13주차 2차시 예제 참조)
 - 도서 source: StructAddMin.c

```
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;
}
// 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;
}
```

* 구조체 vs 공용체 예제 실습(교재 13주차 2차시 예제 참조)

- 도서 source: UnionMemAlloc.c

```
typedef struct sbx
{
    int mem1;
    int mem2;
    double mem3;
} SBox;

typedef union ubox
{
    int mem1;
    int mem2;
    double mem3;
} UBox;
```

```
int main(void)
{
    SBox sbx;
    UBox ubx;
    printf("%p %p %p \n", &sbx.mem1, &sbx.mem2, &sbx.mem3);
    printf("%p %p %p \n", &ubx.mem1, &ubx.mem2, &ubx.mem3);
    printf("%d %d \n", sizeof(SBox), sizeof(UBox));
    return 0;
}
```

【학습정리】

1. 구조체 배열을 초기화하기 위해서 배열요소 각각의 초기화 값을 중괄호로 묶어서 표현한다.
2. 구조체 변수의 주소값과 구조체 변수의 첫 번째 멤버의 주소값은 일치한다.
3. 열거형 상수의 값은 명시되지 않으면 0부터 시작해서 1씩 증가한다. 만약 값이 명시되지 않는 상수는 앞에 정의된 상수 값에서 1이 증가한다.